

Министерство образования и науки РФ

Центр теории и методики обучения математике и информатике
Института стратегии развития образования РАО

ФГБОУ ВО Тульский государственный педагогический университет
им. Л. Н. Толстого

**А. Р. ЕСАЯН, Н. М. ДОБРОВОЛЬСКИЙ,
Е. А. СЕДОВА, А. В. ЯКУШИН**

ДИНАМИЧЕСКАЯ МАТЕМАТИЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ СРЕДА

GeoGebra

Учебное пособие

Часть I

Тула
Издательство ТГПУ им. Л. Н. Толстого
2017

УДК 519.68
ББК 22.18я73
Д46

Рецензенты:

доктор физико-математических наук, профессор *И. М. Буркин*
(Тулский государственный университет);
доктор физико-математических наук, профессор *И. В. Денисов*
(Тулский государственный педагогический университет им. Л. Н. Толстого)

Динамическая математическая образовательная среда *GeoGebra*:
Д46 Учеб. пособие / А. Р. Есаян, Н. М. Добровольский, Е. А. Седова, А. В. Якушин. – Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2017. – Ч. I. – 417 с.

ISBN 978-5-9500201-0-0

Замечательным свойством динамической математической образовательной среды *GeoGebra* является опора на двуединство алгебраического и геометрического, проявляющееся в том, что каждому создаваемому алгебраическому объекту сразу же ставится в соответствие некоторый зримый геометрический образ, и наоборот – по каждому строящемуся геометрическому образу формируется его алгебраическое описание. Тем самым реально существующая тесная связь между алгеброй и геометрией получает в *GeoGebra* оперативное визуальное воплощение, что существенно облегчает как обучение различным разделам математики, так и их освоение на любых уровнях образования на качественно новом уровне. Именно этот факт вместе с динамическими возможностями представления геометрических объектов и определяет дидактический потенциал *GeoGebra*, позволяя использовать ее для организации учебного процесса в соответствии с лично-стно ориентированной парадигмой образования, и интенсифицировать учебный процесс за счет активизации познавательной и учебно-исследовательской деятельности обучающихся. В первой части пособия обсуждается среда разработки приложений *GeoGebra*, достаточно подробно рассказывается об основных треугольных центрах, списках, 2D-графике, ряде задач дискретной математики, электронных таблицах, а также о создании новых пользовательских инструментов.

Пособие ориентировано на студентов, аспирантов, учителей, преподавателей вузов и может быть использовано на семинарских занятиях по специальным и факультативным курсам, связанным с компьютерными методами решения задач, и в частности с построением динамических моделей и экспериментальной проверкой правильности утверждений. Большая часть материала пособия доступна ученикам старших классов профильной школы.

УДК 519.68
ББК 22.18я73

ISBN 978-5-9500201-0-0

© А. Р. Есаян, Н. М. Добровольский,
Е. А. Седова, А. В. Якушин, 2017
© ТГПУ им. Л. Н. Толстого, 2017

Предисловие

Дидактический потенциал популярной в мире математической программной среды *GeoGebra* определяется ее динамическими возможностями представления геометрических объектов. Использование *GeoGebra* в средней школе и в вузе способствует созданию творческой атмосферы на занятиях и дает возможность:

1) достаточно просто, быстро и качественно создавать статичные геометрические чертежи, а также формировать динамические модели, позволяющие в реальном времени визуализировать и изменяемые данные, и результаты действий над ними;

2) проводить экспериментальную проверку правильности сформулированных утверждений на динамических моделях до проведения соответствующих доказательств. Практика показывает, что при правильно построенной модели и известных теоретических результатах они всегда совпадают с экспериментальными результатами;

3) проводить экспериментальную проверку выдвинутых гипотез, доказательства которых еще неизвестны или их не удалось найти в источниках. Такая проверка значительно повышает правдоподобность любой выдвинутой гипотезы, хотя и не отменяет необходимость ее доказательства;

4) быстро знакомить обучающихся как на основных, так и на факультативных занятиях с интересными геометрическими фактами, не входящими в ту или иную основную программу;

5) использовать замену доказательств экспериментальной проверкой правильности утверждений при реализации различных коррекционных программ, а также на дополнительных занятиях с плохо успевающими учениками;

6) создавать качественные наглядные пособия и чертежи по геометрии.

В предлагаемом учебном пособии обсуждаются основные возможности программного обеспечения *GeoGebra*, которое позволяет достаточно просто создавать разнообразные динамические объекты для их использования на занятиях по математике в средней школе на любых уровнях образования или в вузе. При этом, под динамичностью понимается возможность ручного или автоматического “проигрывания” этапов построения модели или возможность оперативного изменения готовой модели при изменении значений ее конкретных параметров.

Использование динамичности для проигрывания этапов построения модели может использоваться учителем для предварительного создания сложных геометрических чертежей. Последующая работа по поэтапному воссозданию чертежа перед учениками уже не требует от него каких-либо усилий, кроме нажатия кнопок, позволяя полностью сосредоточиться на объяснении самого построения и доказательстве тех или иных утверждений, если прово-

дить их требуется. В качестве примера можно рассмотреть, скажем, построение сечений многогранников методом следов. Заранее подготовленный чертеж исключит потери времени на оперативную правку возможных неточностей при его создании непосредственно на уроке.

Использование динамичности второго типа, когда модель может изменяться вместе с теми или иными параметрами, помогает учителю подводить учеников к самостоятельному формулированию того или иного утверждения. В данном случае построение чертежа может быть лабораторным заданием и для самого ученика. Изменяя параметры модели, он наверняка обратит внимание на имеющиеся инварианты, то есть остающиеся неизменными свойства модели, а это уже прямая и несложная дорога к формулированию некоторого утверждения (теоремы). Остальное в руках учителя, проводить доказательство или не делать этого зависит от реализуемой им программы обучения. Рассмотрим пример. Пусть около окружности описан правильный шестиугольник. Если соединить диагоналями его противоположные вершины, то получим, что все они пересекаются в одной точке, и в данном случае – это центр окружности. Пусть ученик описал вокруг окружности некоторый шестиугольник (убрали слово “правильный”). Соединив противоположные вершины диагоналями, он наверняка обратит внимание, что они снова пересекаются в одной точке. Тем самым он “перепоткрыл” теорему Бриансона (для окружности). Используя динамичность модели, можно легко убедиться, что теорема остается справедливой и в случаях, когда шестиугольник вырождается в пятиугольник или четырехугольник. С продвинутым учеником или студентом можно продолжить исследования и дальше, предложив заменить в рассмотренной задаче окружность эллипсом или другим коническим сечением и т. д.

Пособие ориентировано на школьников, студентов, аспирантов, учителей, преподавателей вузов и может быть использовано на семинарских занятиях по специальным и факультативным курсам, связанным с динамической математикой. Большая часть материала пособия доступна ученикам средних и старших классов школ, которые обучаются как по общеобразовательным, так и по специализированным программам.

Пособие создано в рамках выполнения исследований по грантам РФФИ № 15-41-03262_р_центр_а, № 16-41-710194_р_центр_а, и издано при финансовой поддержке этого гранта.

Авторы признательны всем людям, которые ознакомились с различными частями рукописи и внесли полезные предложения по их улучшению. Особенно хотелось бы поблагодарить коллег по работе: Монахова В. М., Абдуразакова М. М., Шулюпова В. А., Реброву И. Ю., Титова А. В., Ванькову В. С., Мартынюк Ю. М., Ваныкину Г. В., Сундукову Т. О., Лапицкую Л. П., Торину Е. Г., Даниленко С. В. и др. С авторами пособия можно связаться по электронной почте:

esayanalbert@mail.ru

elena-sedova@yandex.ru

dobrovol@tspu.tula.ru

yakushin@tspu.tula.ru

Введение

Многие системы компьютерной графики дают возможность создавать статичные геометрические чертежи (графики функций, конструкции, схемы, диаграммы состояний и т. п.), которые и являются конечным продуктом соответствующих действий. Но существуют и такие математические среды, в которых при создании чертежей в памяти и в файле сохраняется не только сам сформированный чертеж, но также автоматически записываемый алгоритм его построения и исходные данные. При этом данные легко доступны для изменений, а уже существующий алгоритм построения тут же применяется к ним заново. Все это позволяет в реальном времени видеть преобразования чертежа вместе с изменением начальных данных. Именно эту триаду из исходных данных, алгоритма и чертежа и называют динамической моделью. А математические среды, которые позволяют создавать динамические модели, называют “системами динамической геометрии”, “интерактивными геометрическими системами” или “средами динамической математики”. Стоит заметить, что идея использования интерактивных геометрических сред в обучении возникла задолго до появления компьютеров и принадлежит Ф. Клейну и А. Пуанкаре. Ф. Клейн в “Эрлангенской программе” писал об изучении геометрии через движение, а А. Пуанкаре говорил об использовании динамических образов при решении геометрических задач.

Система *GeoGebra* относится к новому поколению учебного программного обеспечения. Она используется для визуализации математических объектов и создания их динамических моделей. Читатели, которые знакомы с приложением *Mathtype* фирмы *Design Science* для вставки сложных математических выражений в документы, хорошо понимают всю простоту и мощь этого специализированного редактора формул. Так вот, одна из возможностей приложения *GeoGebra* по простоте и мощи напоминает *MathType*, только ее ниша не формулы, а чертежи геометрических задач. Но *GeoGebra* позволяет создавать не только статичные чертежи, но и формировать сложные динамические (анимационные) модели, строить графики функций, символично или приближенно решать ряд математических задачи, работать с электронными таблицами, проводить компьютерные эксперименты и т. п. Поэтому *GeoGebra* часто называют виртуальной математической лабораторией.

Основные понятия *GeoGebra* – объекты и инструменты. Объекты могут быть алгебраическими и геометрическими. Наборами специальных инструментов создаются те или иные объекты, причем одновременно, как в форме алгебраического описания, так и в виде соответствующего ему геометрического образа. Объекты обладают свойствами, которые достаточно просто можно изменять. Роль инструментов выполняют встроенные программные модули, приводящиеся в действие или щелчками левой кнопки мыши по кнопкам пиктографического меню или с помощью ввода специальных команд. Допускается создание пользовательских инструментов. Из объектов и

формируются динамические модели – чертежи, графики, рисунки, конструкции, схемы, диаграммы состояний и т. п. и их описания.

Слово “динамическая” в сочетании с *GeoGebra* предполагает наличие динамичности двух типов [53]. Суть первого типа динамичности такова. Если некая геометрическая модель уже создана, то всегда возможно пошаговое (покадровое) ручное или автоматическое последовательное развертывание процесса построения модели. Причем делать это можно многократно, с любого места и в любом направлении. Более того, просмотр процесса построения модели может быть организован как шагами, которыми модель изначально формировалась, так и более крупными шагами, объединяющими в себе по несколько последовательных исходных шагов. Заметим, что первый тип динамичности в моделях *GeoGebra* всегда присутствует. Суть второго типа динамичности такова. Если в модели некоторые параметры являются переменными (позиция точки, длина отрезка, величина угла и т. п.), то их изменение легко проводится управляющими элементами мыши или с помощью клавиатурных ключей, а результат этих изменений тут же виден и на модели, и в ее алгебраическом описании.

Первая версия *GeoGebra* появилась совсем недавно – в 2002 г. Ее автором является австрийский математик, профессор университета Зальцбурга Маркус Хохенвартер (*Markus Hohenwarter*) [54, 55]. *GeoGebra* была задумана в качестве средства, объединяющего возможности систем интерактивной геометрии типа *Cabri Geometry*, *The Geometer's Sketchpad* и т. п. и систем компьютерной алгебры типа *Derive*, *Maple* и т. п. в единую интегрированную систему обучения и изучения математики. Программа написана на языке *Java*, является свободно распространяемой, кроссплатформенной (*Windows*, *Linux*, *Mac OS* и др.), представлена более чем на 60 языках, имеет обширное сетевое сообщество, работает на персональных компьютерах, планшетах, смартфонах. Странное слово *GeoGebra* – это аббревиатура типа мопед (мотоцикл+велосипед). Составлено оно из начальной и конечной частей слов *Geometry* и *Algebra*. С самого начала система позиционируется в качестве “Динамической математики для всех”, то есть для любого уровня образования. В настоящее время *GeoGebra* непрерывно развивается и совершенствуется М. Хохенвартером и большой группой его почитателей и последователей со всего мира. Обновления системы выходят практически каждую неделю. Последняя вышедшая версия системы для платформы *Windows* – *GeoGebra 5.0-279.0*.

Замечательным свойством *GeoGebra* является опора на двуединство алгебраического и геометрического, проявляющееся в том, что каждому создаваемому алгебраическому объекту сразу же ставится в соответствие некоторый зримый геометрический образ и наоборот, по каждому строящемуся геометрическому образу формируется его алгебраическое описание. Тем самым, реально существующая тесная связь между алгеброй и геометрией получает в *GeoGebra* оперативное визуальное воплощение, что существенно облегчает как обучение различным разделам математики, так и их освоение на любых уровнях образования на качественно новом уровне. Именно этот факт вместе

с динамическими возможностями представления геометрических объектов и определяет дидактический потенциал *GeoGebra*, позволяя использовать ее для организации учебного процесса в соответствии с личностно-ориентированной парадигмой образования, и интенсифицировать учебный процесс за счет активизации познавательной и учебно-исследовательской деятельности обучающихся. С помощью *GeoGebra* можно проводить экспериментальную проверку правильности сформулированных утверждений, в том числе и проверку выдвинутых гипотез, доказательства которых еще неизвестны или их не удалось найти в источниках. Подобная проверка значительно повышает правдоподобность любой выдвинутой гипотезы, хотя и не отменяет необходимость ее доказательства.

Нельзя сказать, что интерфейс *GeoGebra* с пользователем является традиционным или современным ленточным – он весьма специфичен. Но его оригинальность не есть плод фантазии или прихоти разработчиков, а вытекает из своеобразия решаемых системой проблем. В нем много неожиданных, изящных и полезных находок, которые не просто облегчают работу с системой, а делают ее интересным и увлекательным занятием, не требующим отвлечения на технические детали, мало связанные или совсем не связанные с решаемой задачей. Одно из неоспоримых достоинств интерфейса *GeoGebra* – это наличие не только пиктографических меню инструментов и обычных контекстных меню, но и контекстных пиктографических меню объектов. При этом традиционные характеристики хорошего интерфейса, такие как понятность, удобство, дружелюбность и легкость запоминания в полной мере относятся и к интерфейсу *GeoGebra*.

Наряду с *GeoGebra* имеется много других продолжающих развиваться систем серии “Динамическая математика”. Среди них можно отметить: *GeoNext* (Германия, коммерческая), *The Geometer's Sketchpad* (Англия, коммерческая), *Cabri Geometry* (Франция, коммерческая), а также *KSEG* (свободная), *Kig* (свободная), *Desmos* (графический онлайн-калькулятор) и т. д. Обширный список подобных систем и их сравнительные характеристики можно найти по адресу: https://en.wikipedia.org/wiki/List_of_interactive_geometry_software. Отметим также серию коммерческих программных сред для поддержки уроков математики и информатики в средней школе, разработанных в Институте Новых Технологий (г. Москва): “АвтоГраф”, “Живая математика” (вариант *The Geometer's Sketchpad*), “Интерактивная стереометрия”, “Живая статистика”, “Логомиры” (творческая среда и вероятности) и т. п. Получить информацию по ним можно по адресу: <http://www.int-edu.ru/content/matematika-i-informatika-0>.

Справочные материалы, многочисленные примеры и последняя версия *GeoGebra* доступны по адресу: <http://www.geogebra.org>, установка системы не требует каких-либо ухищрений, а установка ее новых версий может проводиться автоматически. Остается пожелать удачи в освоении *Geogebra* тем, кто решился на этот шаг, и уверить их в том, что сожалеть об этом не придется.

1. Среда разработки приложений

1.1. Основные панели

Документы *GeoGebra* создаются и редактируются в основном окне системы. Там же формируются и геометрические формы (модели, конструкции), то есть создаются чертежи. В этом окне, по мере надобности, могут открываться и закрываться разнообразные взаимосвязанные панели для осуществления того или иного специализированного интерфейса с пользователем. Эти панели можно не только открывать и закрывать, но и превращать в отдельные окна и снова встраивать в основное окно системы. На рис. 1 представлены следующие часто используемые панели [83]:

- панель заголовка. Эта неименованная панель является частью основного окна системы и состоит из строки заголовка, основного меню системы и пиктографического меню инструментов с дополнительными управляющими кнопками. Пиктографическое меню инструментов является контекстным, то есть меняется вместе с изменением текущей панели инструментов (“Полотно” и “Полотно 2”, “Полотно 3D”, “Таблица” и “CAS”);

- панель “Объекты” (*Algebra*);
- панель “Полотно” (*Graphics*);
- панель “Таблица” (*Spreadsheet* – электронная таблица);

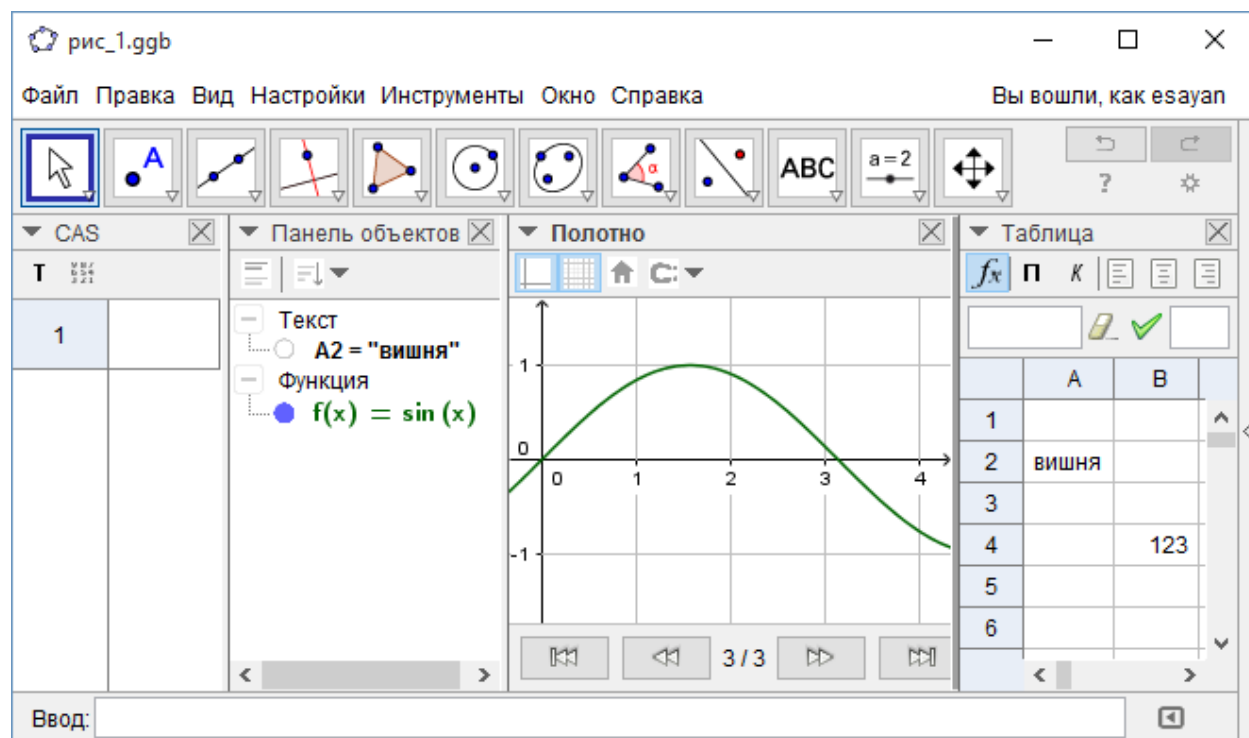



Рис. 1. Некоторые панели *GeoGebra*

- панель *CAS* или панель символьных вычислений (*Computer Algebra System* – система компьютерной алгебры);
- строка ввода команд “Ввод” (*Input Bar*) с кнопкой “

Имеются и другие панели для работы в системе *GeoGebra*.

- панель “Полотно 2” (*Graphics 2*);
- панель “Полотно 3D” (*3D Graphics*);
- панель “Калькулятор вероятностей” (*Probability Calculator*);
- панель “Протокол” (*Construction Protocol*);
- панель “Виртуальная клавиатура” (*Virtual Keyboard*).

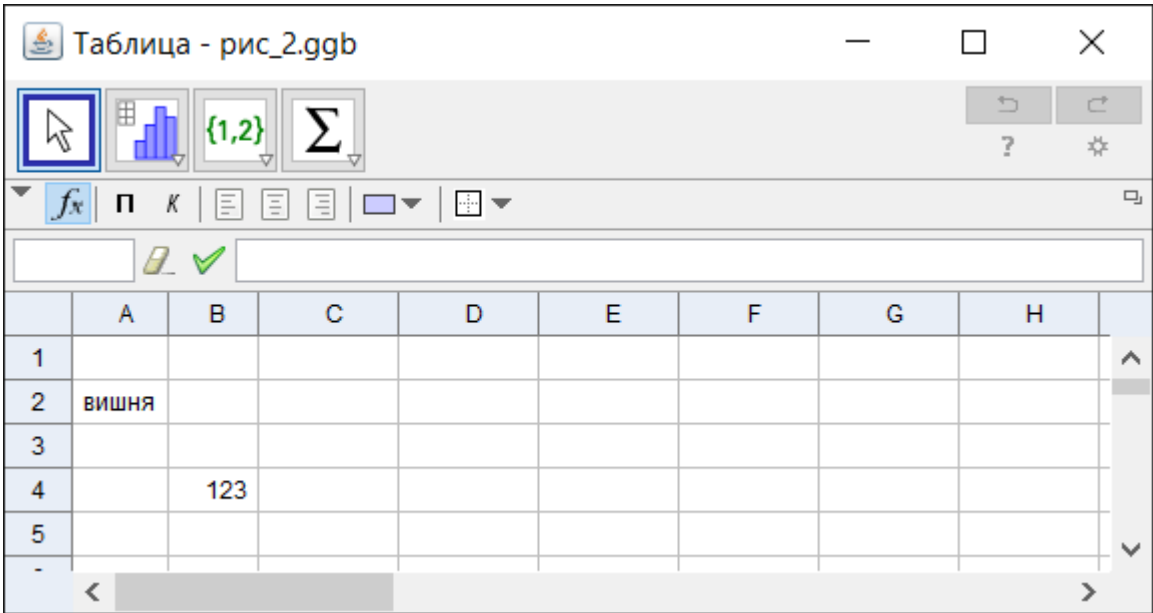


Рис. 2. Вид панели “Таблица”

1.2. Элементы интерфейса

Опишем некоторые общие действия, связанные с интерфейсом пользователя и системы. Задача освоения интерфейса в целом, то есть изучение совокупности правил, средств и методов взаимодействия пользователя и системы, может быть решена лишь в процессе знакомства с объектами *GeoGebra* и инструментами для их создания и управления. Сейчас лишь отметим, что традиционные характеристики хорошего интерфейса, такие как понятность, удобство, дружелюбность и легкость запоминания в полной мере относятся и к интерфейсу *GeoGebra*.

A. Вид/Панель Name

Открытие панелей
А) Команда A выполняется через основное меню и открывает или закрывает панель с именем Name.





А. Настройки/Язык/R-Z>/...



Установка языка



А) Командой А устанавливается язык общения с системой, то есть проводится ее локализация. Заметим, что при установке русского языка во встроенных командах системы, записываемых в строке ввода или на панели CAS, можно использовать и русские, и английские имена. Например, правильно будет понято и выражение *Определитель*[...], и выражение *Determinant*[...]. В каждом из этих вариантов записи есть свои плюсы и минусы. Не вдаваясь в детальное обсуждение этой темы отметим лишь то, что написание английских имен не только избавляет нас от казусов локализации, а они есть, но и способствует усвоению этих имен, а значит упрощает понимание многочисленных примеров использования системы *GeoGebra*, щедро разбросанных по Интернету. Если установлен английский язык локализации системы, то можно писать только английские имена команд. В дальнейшем мы будем работать с системой, локализованной для России (*Options/Language/R-Z>/Russian*-Русский язык), но использовать англоязычные имена команд. Впрочем, *online*-справки по большей части команд *GeoGebra* пока можно получать только при англоязычной локализации.

- A. Click()
- B. Click()
- C. Click()

Некоторые общие действия

А) На каждой открываемой панели в правом верхнем углу располагается кнопка “” для закрытия этой панели. Щелчком по кнопке “”, появляющейся при наведении курсора на соответствующее место левее кнопки “”, текущая панель перемещается из основного окна *GeoGebra* в отдельное независимое окно. Например, щелчок по кнопке “” на панели “Таблица” в окне, представленном на рис. 1, выделит электронную таблицу в отдельное окно (см. рис. 2). Подобную операцию можно осуществить с любой панелью, кроме панели “Виртуальная клавиатура” и дополнительной панели “Список команд”;


В) Кнопка “” находится в правой части второй строки панели, открытой в отдельном окне. Щелчок по кнопке “” приводит к встраиванию панели в главное окно *GeoGebra*.


С) Команда С выполняется щелчком по кнопке “”, находящейся справа от строки ввода. Эта кнопка работает как двухпозиционный переключатель, вставляя в главное окно системы и удаляя из него панель “Список команд”, состоящую из трех частей. Верхняя часть панели содержит перечень имен всех команд и всех категорий команд в виде раскрывающихся списков. При выборе категории, а затем имени конкретной команды, в средней части панели индицируется синтаксис этой команды. В нижней части панели находятся три управляющих кнопки: “Вставить”, “Справка (Online)” и “”.

Кнопкой “Вставить” в строку ввода вставляется шаблон текущей команды. Это упрощает ввод команды, хотя ее можно формировать и иными способами, например, прямым набором символов с клавиатуры. В этом случае набору помогает всплывающий список подходящих шаблонов. Щелчком по




любому шаблону его можно вставить в строку ввода. Альтернативно вставка шаблона организуется двойным щелчком по имени команды на панели “Список команд”. Кроме того, система формирует список всех команд, которые уже вводились в текущей сессии. Раскрывается этот список ключом “↑”. По списку можно перемещаться ключами “↑” и “↓”. Ключом *Enter* текущая команда дублируется в строке ввода для дальнейшего использования. При активном поле ввода в его правом конце имеется кнопка “a”. Щелчком по ней открывается панель латинских букв и некоторых специальных символов, которые также можно использовать при вводе команд.

Кнопка “Справка (Online)” позволяет обратиться к Интернет-ресурсам за помощью по синтаксису текущей команды, если он не совсем ясен.

Третья кнопка “” не столь важна – по ней сворачивается полностью или частично раскрытый список команд.

A. Click()





Локальные пиктографические меню панелей

А) На многих панелях, в том числе и на панелях “Объекты” и “Полотно”, в верхней строке слева от заголовка присутствует кнопка “”. Щелчок мышью по этой кнопке раскрывает локальное пиктографическое меню, а кнопка принимает вид “”. Щелчок по кнопке “” закрывает локальное меню.

Локальное меню для панели “Объекты” состоит из двух выборов: кнопки, по которой прячется или выводится информация о так называемых вспомогательных объектах, и списка, позволяющего сортировать объекты по различным признакам. По умолчанию они сгруппированы по категориям (*Числа, Отрезки, Точки, Функции, ...*), хотя возможны и другие основания для их представления:

- по типу зависимости объектов (свободные и зависимые);
- по графическим слоям, в которых нарисованы геометрические образы объектов;
- по порядку ввода-построения объектов.



Локальное меню панели “Полотно” является контекстным. Иными словами, оно всегда относится к текущему объекту и служит для установки и изменения его свойств, то есть стиля вывода. В начале работы с документом показывается меню для системы координат, состоящее из 4 выборов: кнопки, позволяющей прятать и показывать координатные оси; кнопки, позволяющей прятать и показывать координатную сетку; кнопки для возврата чертежа на позицию по умолчанию; списка для задания типа привязки объектов к сетке.

<p>A. Click()</p> <p>B. Click()</p>	<p>C. Click()</p> <p>D. Click()</p>
---	---

Полезные кнопки

Часто приходится выполнять действия с помощью малоприметных кнопок, расположенных с правой стороны

пиктографического меню (см. рис. 1).

А, В) При создании чертежа не стоит забывать про возможность отката, то есть восстановления предыдущих его состояний (кнопка  или ключ $Ctrl+z$), а также движения по этим состояниям в обратном направлении (кнопка  или ключ $Ctrl+y$).

С) При использовании конкретного инструмента полезной может быть краткая справка по работе с ним, вызываемая кнопкой помощи “?” или ключом $Ctrl+F1$. Эта справка появляется на панели “*Описание*”, на которой имеется кнопка “*Справка (Online)*”. По ней можно перейти на сайт *GeoGebra* для получения более развернутой справки по данному инструменту (если она там есть). Краткую справку по инструменту в виде всплывающей подсказки также можно получить, просто подведя курсор к соответствующей кнопке.

Д) Работа с системой протекает при наличии ряда установок, действующих по умолчанию или заданных пользователем. Скажем объект “*Точка*” имеет определенную форму, цвет, размеры и т. д. Установки можно изменять разными способами и наиболее общий из них – это действовать через панель “*Настройки*”. Командой *D* эта панель может быть открыта на любой из закладок: “*Объекты*”, “*Полотно*”, “*Разметка*”, “*Настройки по умолчанию*”, “*Дополнительно*”. Наиболее часто приходится работать с многостраничной закладкой “*Объекты*”. Через локальное пиктографическое меню любой закладки можно переходить к другим закладкам.

Через основное меню панель “*Настройки*” открывается на закладке “*Дополнительно*” (*Настройки/Дополнительно*). Через контекстное меню любого объекта панель “*Настройки*” открывается на закладке “*Объекты*”, причем текущим становится именно этот объект, то есть сразу можно приступить к редактированию его свойств (*RClick(объект)/Свойства*).

2. Панели “*Объекты*” и “*Полотно*”

2.1. Связь алгебры и геометрии

Панель “*Объекты*” служит для учета формируемых алгебраических объектов любого типа и иных действий с ними. Создаются такие объекты пользователем через строку ввода или системой как алгебраические описания геометрических объектов “*рисуемых*” на панели “*Полотно*”.

Панель “*Полотно*” служит для формирования разнообразных геометрических объектов. Создаются такие объекты пользователем с помощью многочисленных встроенных и, возможно, пользовательских инструментов или системой как геометрические образы алгебраических объектов, вводимых на панель “*Объекты*” через строку ввода.

Замечательным свойством *GeoGebra* является опора на двуединство алгебраического и геометрического, проявляющееся в том, что алгебраическим

объектам панели “Объекты” соответствуют некоторые геометрические образы панели “Полотно” и наоборот, геометрическим формам панели “Полотно” соответствуют их алгебраические описания на панели “Объекты”. Тем самым, реально существующая тесная связь между алгеброй и геометрией получает в *GeoGebra* оперативное и наглядное воплощение.

На рис. 1 представлен пример панели “Объекты”, на которой через строку ввода сформированы: число 2, точка (2, 1), уравнение прямой $x=-2$, и уравнения двух конических сечений – окружности и параболы. Каждый созданный алгебраический объект на панели “Полотно” имеет свой геометрический образ. Для числа 2 этот образ невидим (радиокнопка слева от $a = 2$ выключена). В дальнейшем мы научимся создавать геометрические объекты непосредственно на панели “Полотно” и при этом автоматически получать соответствующие им алгебраические объекты на панели “Объекты”.

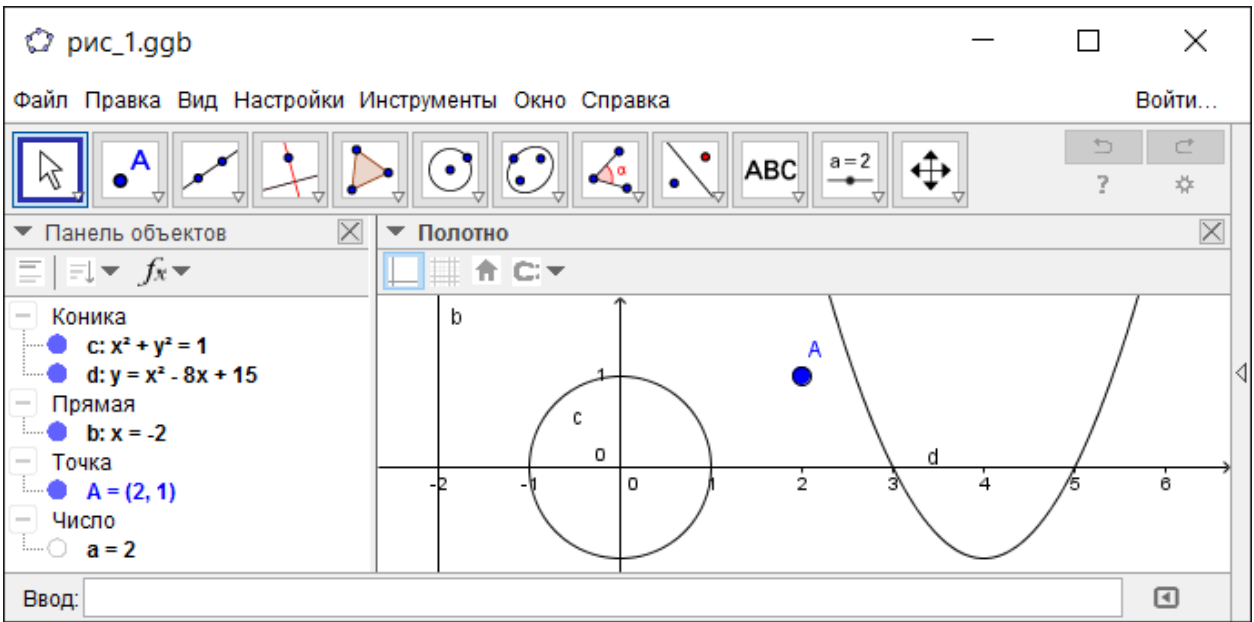


Рис. 1. Вид панелей “Объекты” и “Полотно” с некоторыми объектами

2.2. Объекты

Пришла пора поговорить об объектах более обстоятельно. Как мы уже поняли, каждый объект *GeoGebra* одновременно представляется в двух ипостасях – алгебраической и геометрической. В любом случае системой формируется список всех существующих объектов, который полностью или частично выводится на панели “Объекты”. По умолчанию этот список является отсортированным (сгруппированным) по отдельным именованным категориям (типам) объектов. Имена этих категорий: Числа, Точки, Отрезки, Прямые, Лучи, Углы, Треугольники, Четырехугольники, ..., Списки, Тексты, Функции, Неравенства, Конические сечения, Конус, Пирамида, Цилиндр и т. д.

Через строку ввода объекты задаются выражениями, которые могут содержать только их значения или имена и значения. Вот примеры задания объектов значениями: 7 (число), Div(12, 4) (число), (1, 2) (точка), $x=3$ (прямая), $\sin(x)$ (функция), $x^2+y^2=4$ (окружность), $x^3+y^2=3$ (неявная кривая) и т. д. и их задания именами и значениями: $a=7$, $b=\text{Div}(12, 4)$, $P=(1, 2)$, $t:x=3$, $g(x)=\sin(x)$, $m:x^2+y^2=4$, $q:x^3+y^2=3$ и т. д. При вводе объектов значениями их имена система формирует автоматически по своему усмотрению. Хотя впоследствии их всегда можно изменить, причем даже на уже существующее имя. Если, скажем, был объект с именем x и появился новый объект с этим же именем, то имя прежнего объекта автоматически приобретет индекс и станет x_1 .




Метки (обозначения) объектов на панели “Полотно”


Каждый объект имеет такие характеристики, как *имя*, *значение* и *заголовок*. Из указанных характеристик на панели “Полотно” для объекта формируется *метка* или, по-другому, *обозначение*, которое может быть: *именем*, *значением*, *именем и значением*, *заголовком*. Например:

- n – имя объекта “переменная”, 5 – значение объекта n , $n=5$ – имя и значение объекта n , *вес* – заголовок объекта n ;
- P – имя объекта “точка”, (1, 2) – значение объекта P , $P=(1, 2)$ – имя и значение объекта P , *фокус* – заголовок объекта P и т. д.

Имя и значение для объекта обязательны. Заголовок необязателен и фактически является вторым именем или псевдонимом объекта. Он может быть любым текстом и даже формулой. Создаются заголовки через панель “Настройки” (вкладка “Объекты (Свойства)”, страница “Основные”, поле ввода “Заголовок”). Там же устанавливаются и другие свойства объектов, например, тип выводимого обозначения. Заголовки могут появляться только на панели “Полотно”. Если установлен вывод заголовка, а он не сформирован, то по умолчанию показывается имя объекта. Можно вообще отменить вывод обозначения объекта и даже самого объекта. В последнем случае вывод обозначения отменяется автоматически. Впрочем, отмену вывода обозначения и (или) объекта проще выполнять через его контекстное меню. На этой же вкладке независимыми переключателями для текущего объекта дополнительно можно включить режим оставления следа при перемещениях или зафиксировать позицию на полотне, запрещая перемещения. Разрешается также объявить объект вспомогательным. В этом случае в списке панели “Объекты” он не будет виден, хотя через панель “Настройки” останется доступным. Некоторые объекты система автоматически делает вспомогательными.

- A. Click() / Объекты / Настройки по умолчанию
 B. Настройки / Сбросить настройки
 C. Click(объект) / ...

Стиль вывода объектов на панели “Полотно”

А) Вид выводимых на панель “Полотно” объектов зависит от ряда установленных для них свойств, действующих по умолчанию. Просмотреть эти установки, а при необходимости и изменить их, можно на страницах вкладки “Настройки по умолчанию (🔧)” панели “Настройки”. Открывается вкладка командой *A* (кнопка  находится с правой стороны в строке пиктографического меню).

В) Команда *B* выполняется через меню и сбрасывает все изменения установок объектов, сделанные в текущей сессии.

С) Для установки свойств любого выделенного (текущего) объекта можно использовать локальное пиктографическое меню. При этом выделять объект можно как щелчком по нему на панели “Полотно”, так и щелчком по его описанию на панели “Объекты”. Напомним, что пиктографическое меню объектов открывается щелчком по кнопке “▶”, расположенной перед словом “Полотно”.

А. Настройки/Обозначения/ф

Установки для вывода обозначений объектов

Обозначения объектов на панели “Полотно” выводятся в автоматическом режиме и делается это для различных объектов по-разному, то есть так, как это предписано делать по умолчанию. Скажем, над управляющим элементом “Ползунок” обозначение выводится в виде имени и значения ($a=2$, $x=7$, ...); около точки появляется ее имя (A , P , ...), около отрезка прямой – также индицируется имя отрезка (b , e , ...) и т. д. Однако на страницах вкладки “Настройки по умолчанию” панели “Настройки” можно изменить действующие установки для вывода объекта любого типа, в том числе и установку для вывода его обозначений.

Кроме того, и это мы уже отмечали, на страницах вкладки “Объекты” панели “Настройки” вывод типа обозначения любого уже существующего объекта можно изменять индивидуально, в том числе и отменять этот вывод. Это же самое можно делать и через локальное пиктографическое меню текущего объекта.

А) Командой *A* можно изменять установки для вывода обозначений. При этом для уже существующих объектов обозначения не изменяются. Возможны такие случаи:

- ф = *Автоматически* – выводятся предписанные обозначения;
- ф = *Вводить* – устанавливается вывод предписанных обозначений для всех вновь создаваемых объектов;
- ф = *Не вводить* – отменяется вывод обозначений для всех вновь формируемых объектов;
- ф = *Только для точек* – выводятся предписанные обозначения для точек. Для остальных объектов вывод обозначений блокируется.



Свободные и зависимые объекты

Объекты *GeoGebra* могут быть *свободными* или *зависимыми*. Свободные объекты создаются независимо от других объектов и никак с ними не связаны. Их значения, а также позиции, если только они не зафиксированы, можно изменять в любой момент времени. Например, можно создать свободный объект “точка” (инструмент “Точка”) или свободный объект “прямая, проходящая через две точки” (инструмент “Прямая”) и т. д. Значения и (или) позиции зависимых объектов вольно изменяться не могут. Они в той или иной мере связаны с другими объектами. Например, если создать объект “точка на прямой” (инструмент “Точка на объекте”), то перемещение полученной точки возможно только по этой прямой, и это уже зависимый объект. Хотя другие свойства точки (размер, цвет, форма, ...) мы можем изменять по своему усмотрению. Другой пример. Если создать прямую, проходящую через заданную точку P перпендикулярно другой прямой m (инструмент “Перпендикулярная прямая”), то получим объект, жестко связанный с P и m . Перемещение сформированной прямой возможно лишь при перемещении P или m , хотя другие ее свойства изменять можно независимо от свойств исходных объектов.


2.3. Выделение объектов

Наведение курсора на объект выделяет его, то есть делает текущим, но только на то время, пока курсор находится над объектом. Внешний вид выделенного объекта немного отличается от вида его невыделенных собратьев. Чтобы выделение сохранялось при смещении курсора, требуется выполнить ту или иную команду. О них и идет речь ниже.

- A. Click(объект)
- B. Tab
- C. Shift+Tab

Выделение объектов

А) Командой *A* выделяется конкретный объект. Щелчок левой кнопкой мыши проводится по объекту или на панели “Объекты”, или на панели “Полотно”. В

последнем случае это надо делать в режиме “Перемещать” (). С выделенным (текущим) объектом можно совершать различные операции, в том числе менять свойства, а если он свободен и не зафиксирован, то и без ограничений перемещать по полотну. Фактически работаем мы всегда с одним или несколькими выделенными объектами.

В) Считается, что объекты упорядочены в порядке времени их создания. Если выделен один объект, то по ключу *B* выделение переносится на следующий объект. С последнего объекта выделение переносится на первый объект. Таким образом ключами *B* можно циклически менять текущий объект, а также отслеживать последовательность формирования объектов

С) Если выделен один объект, то по ключу *C* выделение переносится на предыдущий объект. С первого объекта выделение переносится на последний объект.

A. <i>Ctrl</i> -технология	C. <i>Shift+Ctrl</i> -технология
B. <i>Shift</i> -технология	D. <i>Click</i> (категория)

Выделение группы объектов

Если выделить несколько объектов, то с ними можно обращаться как с единым объектом, одновременно выполняя те или иные операции. И хотя выделять группы объектов можно как на панели “Полотно”, так и на панели “Объекты”, последняя предоставляет большие возможности для выполнения этой операции.

А) Командой *A* группа объектов выделяется по так называемой *Ctrl*-технологии, суть которой в следующем. Командой *Click*(объект) выделяется начальный объект, затем нажимается клавиша *Ctrl* и, не отпуская *Ctrl*, проводятся щелчки по другим невыделенным объектам, которые становятся выделенными. При нажатой клавише *Ctrl* можно проводить щелчки и по уже выделенным объектам, снимая с них выделение.

В) Командой *B* группа объектов выделяется по так называемой *Shift*-технологии. Делать это можно только на панели “Объекты”. Суть технологии в следующем. Командой *Click*(объект) выделяется начальный объект, затем нажимается клавиша *Shift* и, не отпуская *Shift*, проводится щелчок по другому объекту, причем совсем не обязательно из той же самой категории, что и начальный объект. Все объекты между начальным и конечным объектами становятся выделенными.

С) Командой *C* группа объектов выделяется по так называемой смешанной технологии. Делать это можно только на панели “Объекты”. Суть технологии в следующем. Сначала по *Shift*-технологии выделить некоторую группу объектов, а затем продолжить добавлять к ней и (или) удалять из нее объекты по *Ctrl*-технологии.

Д) Командой *D*, выполняемой на панели “Объекты” щелчком левой кнопки мыши по заголовку категории объектов, выделяются сразу все объекты этой категории (точки, отрезки, ...).

A. <i>Ctrl+Shift+j</i>	C. <i>Ctrl+j</i>
B. <i>Правка/Выбрать потомков</i>	D. <i>Правка/Выбрать предков</i>

Выделение потомков и предков

А) Ключом *A* выделяются потомки всех выделенных объектов. Иными словами, выделяются все те объекты, которые зависят от выделенных объектов.

В) Команда *B* выполняется аналогично *A*, но делается это через основное меню.

С) Ключом *C* выделяются предки всех выделенных объектов. Иными словами, выделяются все те объекты, от которых зависят выделенные объекты.

Д) Команда *D* выполняется аналогично *C*, но делается это через основное меню.

A. <i>Ctrl+a</i> B. <i>Правка/Выбрать все</i>	C. <i>Ctrl+l</i> D. <i>Правка/Выбрать текущий слой</i>
--	---

Выделение всех объектов и всех объектов текущего слоя

- A) Ключом *A* выделяются все имеющиеся объекты.
- B) Команда *B* выполняется аналогично *A*, но делается это через основное меню.
- C) По умолчанию объекты визуализируются в графическом слое 0. Всего может быть 10 слоев с номерами: 0, 1, ..., 9. Сменить текущий слой можно на панели “*Настройки*”. Например, это делается командой “*Click(☒)/Объекты/Дополнительно/Слой...*”.
- Пусть выделен некоторый объект *φ*. Ключом *C* выделяются все имеющиеся объекты того слоя, которому принадлежит *φ*. Ключ *C* работает и в том случае, когда предварительно выделено несколько объектов, но одного слоя. Если они в разных слоях, то по *C* производится лишь снятие выделений.
- D) Команда *D* выполняется аналогично *C*, но делается это через основное меню.

A. <i>Ctrl+i</i> B. <i>Правка/Инвертировать выбор</i>
--

Инвертирование выделения

- A) Ключом *A* все выделенные объекты делаются невыделенными, а невыделенные объекты – выделенными.
- B) Команда *B* выполняется аналогично *A*, но делается это через основное меню.

2.4. Некоторые операции с объектами

A. <i>Delete</i> B. <i>BackSpace</i>	C. <i>Правка/Удалить</i> D. <i>RClick(объект)/Удалить</i>
---	--

Удаление выделенных объектов

- A-B) Ключами *A* и *B* удаляется выделенный объект или группа выделенных объектов. Вне зависимости от того, на какой из панелей “*Объекты*” или “*Полотно*” проводилось выделение объектов, их удаление происходит одновременно с обеих панелей.
- C) Команда *C* выполняется аналогично *A*, но делается это через основное меню.
- D) Команда *D* выполняется аналогично *A*, но делается это через контекстное меню.

A. <i>Ctrl+c</i> B. <i>Правка/Копировать</i>

Копирование выделенных объектов в буфер

- A) Клавиатурным ключом *A* выделенные тем или иным способом объекты со всеми их свойствами копируются в буфер памяти.


В) Команда В выполняется аналогично А, но делается это через основное меню.

A. Click/Ctrl+v/Click
B. Click/Правка/Вставить/Click

Вставка объектов из буфера

А) Командой А объекты, находящиеся в буфере памяти, вставляются на панель “Полотно” с позиции, по которой производился первый щелчок мышью. Вставленный объект получает имя копируемого объекта, но с добавлением индекса. Если имя уже было с индексом, то индекс изменяется.

В) Команда В выполняется аналогично А, но делается это через основное меню.

A. Click() /Тащи и бросай/Click	D. ω ($\omega \in \{\rightarrow, \leftarrow, \downarrow, \uparrow\}$)
B. Alt+ ω ($\omega \in \{\rightarrow, \leftarrow, \downarrow, \uparrow\}$)	E. Shift+ ω ($\omega \in \{\rightarrow, \leftarrow, \downarrow, \uparrow\}$)
C. Ctrl+ ω ($\omega \in \{\rightarrow, \leftarrow, \downarrow, \uparrow\}$)	

Перемещение выделенных объектов по панели “Полотно”

А) Командой А выделенные свободные объекты и их потомки перемещаются по панели “Полотно”. Позиции этих объектов на панели “Объекты” не меняются.

В) Команды В выполняются аналогично А, но перемещение осуществляется на конкретный (большой) шаг по указанному направлению.



С) Команды В выполняются аналогично А, но перемещение осуществляется на конкретный (средний) шаг по указанному направлению.

Д) Команды С выполняются аналогично А, но перемещение осуществляется на конкретный (небольшой) шаг по указанному направлению.

Е) Команды D выполняются аналогично А, но перемещение осуществляется на 1 пиксель по указанному направлению. Это дает возможность проводить “тонкую” настройку позиций объектов относительно друг друга.

A. Click()
B. Click()

Закрепление и открепление свободных объектов

Свободные объекты могут быть закрепленными на полотне или не закрепленными. У первых из них, пока закрепление не снято, изменить позиции нельзя ни мышью ни каким-либо иным способом, у вторых – изменить позиции можно. Определить, является или нет текущий свободный объект закрепленным, можно по виду значка на крайней правой кнопке локального пиктографического меню панели “Полотно” ( – не закреплен,  – закреплен). Команды А и В выполняются щелчками левой кнопки мыши по этой кнопке.

А) По команде А закрепляется:


- текущий свободный незакрепленный объект;
- все выделенные свободные объекты, если первый из них не закреплен.

В) По команде В открепляется:

- текущий закрепленный объект;

- все выделенные объекты, если первый из них закреплён.

Замечания. 1. Изначально свободный объект создается незакрепленным. И делается это потому, что отключен независимый переключатель “Закрепить объект” на странице “Basic” вкладки “Настройки по умолчанию” панели “Настройки”. При необходимости включить этот переключатель можно через контекстное меню.



2. Закрепление объектов происходит относительно друг друга и потому изменение масштаба к искажению модели не приводит. Менять масштаб можно клавишами “Ctrl++”, “Ctrl+-” и иными способами. Для некоторых типов объектов (тексты, ползунки и т. д.) можно задавать и абсолютные позиции на экране ()” локального пиктографического меню панели “Полотно”.

3. Когда некоторая динамическая модель создана для автоматического выполнения, обычно часть или все ее свободные объекты целесообразно за-
крепить.


2.5. Изменение свойств объектов

Мы уже отмечали, что на страницах вкладок панели “Настройки” можно изменять свойства уже существующих объектов, а также проводить установки свойств для вывода последующих объектов. Некоторые свойства можно изменять и (или) устанавливать через локальные пиктографические меню и (или) контекстные меню объектов. В любом случае изменения свойств существующих объектов касаются всех выделенных объектов. Причем выделять их разрешается как на панели “Полотно”, так и на панели “Объекты”.

Панель “Настройки” имеет несколько вкладок, каждая из которых содержит по несколько страниц с разнообразными управляющими элементами и полями ввода. Именно с этими элементами и полями нам и приходится работать. Находясь в любом месте панели “Настройки”, можно переходить не только к другим ее вкладкам и страницам, но и изменять наборы выделенных объектов. Открывается панель “Настройки” разными способами. Ознакомление с возможностями, предоставляемыми этой панелью, дело не быстрое. Но оно происходит постепенно в процессе освоения инструментов и команд *GeoGebra*, которыми и создаются разнообразные объекты.

A. Click()/ф...	D. Click()/...
B. RClick(объект)/Свойства/...	E. DClick(объект)
C. RClick(позиция)/Полотно/...	


Открытие панели “Настройки” и локальных пиктографических и контекстных меню

А) Командой *A* панель “Настройки” открывается на той странице вкладки φ ($\varphi \in \{“Объекты (Свойства)”, “Полотно”, “Разметка”, “Настройки по умолчанию” и “Дополнительно”\}$), на которой она была открыта в последний раз. Кнопка “” расположена на правой стороне строки инструментов.

В) Командой *B*, выполняемой через контекстное меню, панель “Настройки” открывается на вкладке “Объекты (Свойства)”.

С) Командой *C*, выполняемой через контекстное меню, панель “Настройки” открывается на вкладке “Полотно”. Щелчок левой кнопкой мыши должен проводиться по позиции полотна, свободной от объектов.


Д) Командой *D* открывается локальное пиктографическое меню. Если меню открыто на панели “Полотно” и есть выделенные объекты, то это меню для данных объектов. Если меню открыто на панели “Полотно” и нет выделенных объектов, то это меню для панели “Полотно”. Если меню открыто на панели “Объекты”, то в любом случае можно лишь скрыть или показать вспомогательные объекты, или отсортировать объекты по разным основаниям.

Е) Команда *E* выполняется двойным щелчком левой кнопкой мыши по объекту (при активном инструменте “”). По *E* объект делается текущим и одновременно открывается панель “Переопределить” для правки определения объекта. Если объект был текстовый, то для правки открывается панель “Текст”.




A. *RClick(объект)/Переименовать*
B. *DClick(объект)/...*

Переименование объектов

А) Командой *A* через контекстное меню можно изменить имя объекта.

Меню можно вызывать и на панели “Полотно”, и на панели “Объекты”. Переименование происходит на специально открываемой для этой цели панели “Переименовать” со строкой ввода/редактирования. В конце строки ввода имеется кнопка “”, щелчок по которой открывает панель греческих букв и иных дополнительных символов. При редактировании объекта их можно использовать наряду с символами обычной или виртуальной клавиатуры. Последняя открывается через основное меню командой “Вид/Клавиатура”.

В) Командой *B*, выполняемой через панель “Объекты”, можно отредактировать имя объекта непосредственно в его описании или сделать это на специально открываемой панели “Переопределить”. Отметим, что данная команда позволяет не только переименовать объект, но и переопределить его.

A. *RClick(объект)/* *Показывать объект*
B. *RClick(объект)/* *Показывать обозначение*
C. *Click(* *)/...*

Визуализация объектов и их обозначений (меток)

А) На панели “Объекты” около каждого объекта имеется радиокнопка — небольшой кружок, называемым кнопкой визуализации геометрического образа объекта. Внутренность кнопки раскрашена синим или белым цветом,

означающими, что объект на панели “Полотно” соответственно виден или не виден. На рис. 2 у четырех алгебраических объектов имеются геометрические образы, а у числа 2 образа нет – кнопка не закрашена. Щелчок по кнопке переключает цвет кнопки и одновременно прячет или показывает соответствующий ей геометрический образ. На рис. 3 показана визуализация числа a (2) в виде ползунка – конструкции из отрезка прямой и перемещаемой по нему кнопки. По умолчанию отрезок выводится горизонтально, значения на нем соответствуют промежутку $[\min(-5, a), \max(5, a)]$, а кнопка устанавливается в позицию a . Перемещениями кнопки по отрезку можно изменять значение алгебраического объекта a в указанных пределах.

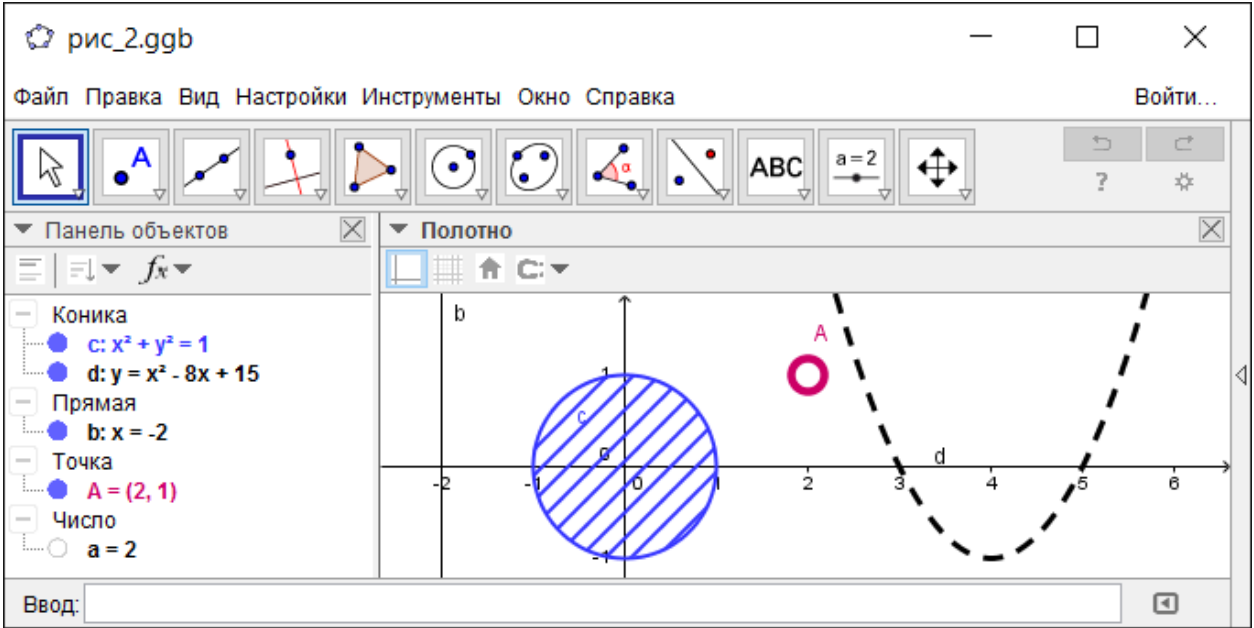


Рис. 2. Объекты рис. 1 с некоторыми измененными свойствами

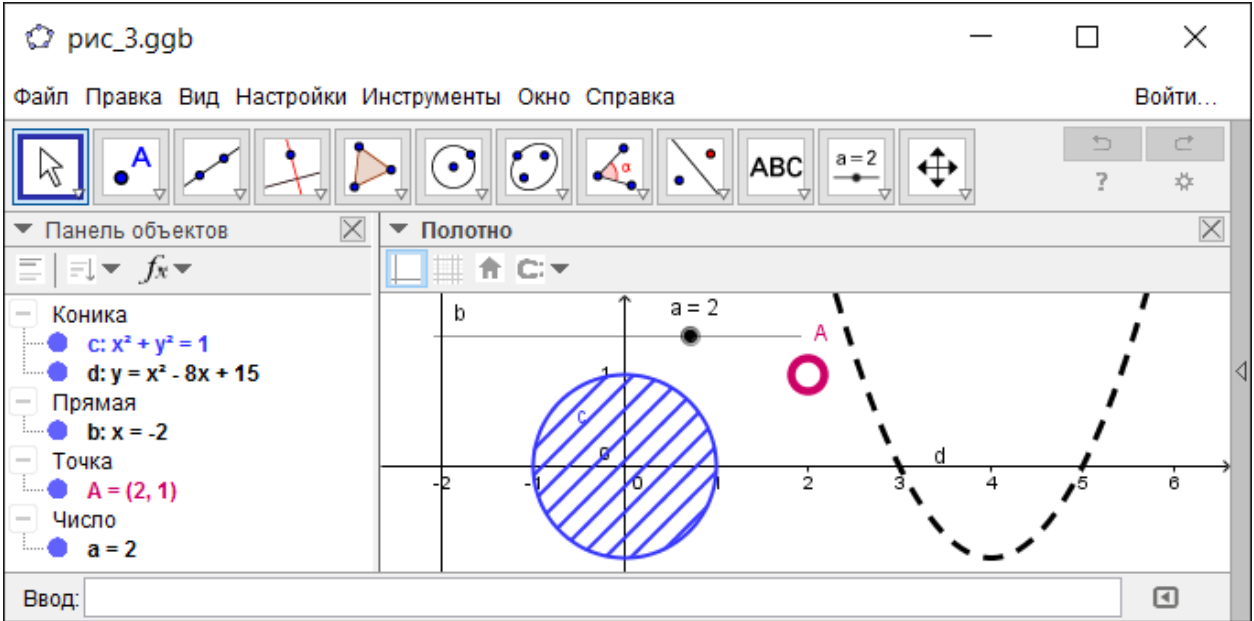


Рис. 3. Визуализация объекта a (числа 2) в виде ползунка

В) Если алгебраический объект визуализирован на панели “Полотно”, то по умолчанию выведено и его обозначение. Команда *В* действует как двухпозиционный переключатель, пряча и восстанавливая обозначение.

С) Обозначения объектов могут иметь разный вид (*имя, имя и значение, значение, заголовок*). По команде *С*, выполняемой через контекстное пиктографическое меню, обозначения можно прятать и снова показывать и, кроме того, менять их вид. Разумеется, все это можно делать и через меню “Настройки”.

Мы уже имели дело с алгебраическими объектами “Коника”, “Прямая”, “Точка”, “Число”. Еще три важных объекта – “Неравенство”, “Текст” и “Функция” – представлены на рис. 4. Дадим краткие пояснения к этому рисунку. Вводя в строке ввода объект “Неравенство” в виде логического выражения $((x - 1)^2 + (y - 1)^2 \leq 2) \wedge (y < \text{abs}(x))$, мы фактически предлагаем системе *GeoGebra* графически решить систему алгебраических неравенств:

$$\begin{cases} (x - 1)^2 + (y - 1)^2 \leq 2 \\ y < |x| \end{cases} \tag{1}$$

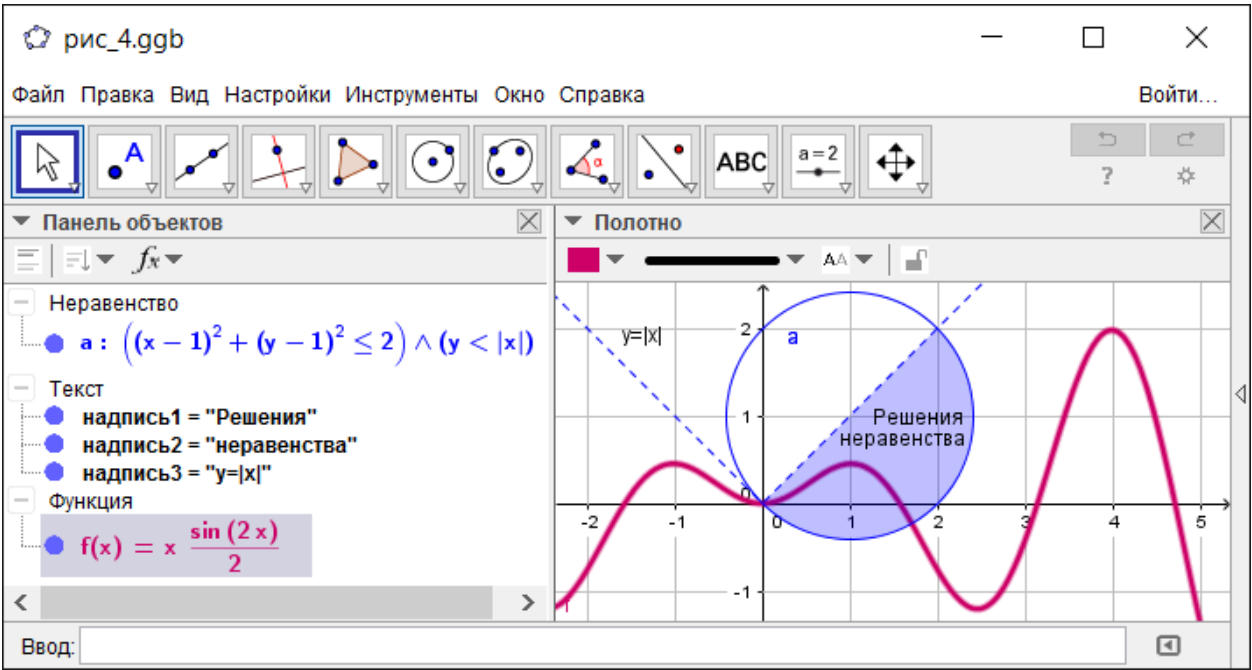



Рис. 4. Примеры объектов “Неравенство”, “Текст” и “Функция”

Оператор отношения “≤” и оператор конъюнкции “∧” следует вводить с панели дополнительных символов, раскрываемой кнопкой “”, которая расположена в конце строки ввода. Впрочем знак “≤” можно вводить и последовательностью “<=”. Геометрический образ для нашего неравенства строится в виде кривых, задаваемых уравнениями $y = |x|$ и $(x - 1)^2 + (y - 1)^2 = 2$, и залив-

кой области точек, удовлетворяющих системе (1). В нашем случае использована стандартная заливка, которую через контекстное меню можно сменить на любую из 7 дополнительных заливок (*Штриховка*, *Cross-hatching* (перекрестная штриховка), *Шахматная*, *Dots* (точки), *Honeycomb* (медовые соты), *Bricks* (кирпичи), *Weaving* (плетение)). У выведенного объекта “Функция” заменен цвет и толщина кривой. У выведенных объектов “Тест” протяжкой мыши изменены позиции на графике.

Преобразуем объекты на рис. 4 так. Через контекстное меню удалим объект “Функция”. Добавим объект “Число” (6), изменим на месте его метку на k и через контекстное меню установим для k диапазон от -1 до 10. Фрагмент $y < |x|$ объекта “Неравенство” заменим на месте на $y < k \cdot |x|$, а у его геометрического образа через контекстное меню изменим заливку на “перекрестную штриховку”. Включением кнопки числового объекта k , если она еще не включена, индицируем на полотне образ k , то есть ползунок. Протаскиванием ползунка подберем для него подходящее место. Перемещая кнопку ползунка, мы фиксируем k и сразу же будем иметь актуализацию решений параметрической системы неравенств

$$\begin{cases} (x-1)^2 + (y-1)^2 \leq 2 \\ y < k|x| \end{cases} \tag{2}$$

(см. рис. 5). Через контекстное меню для ползунка можно устроить и анимацию решений (2) (*RClick(ползунок)/Анимировать*).

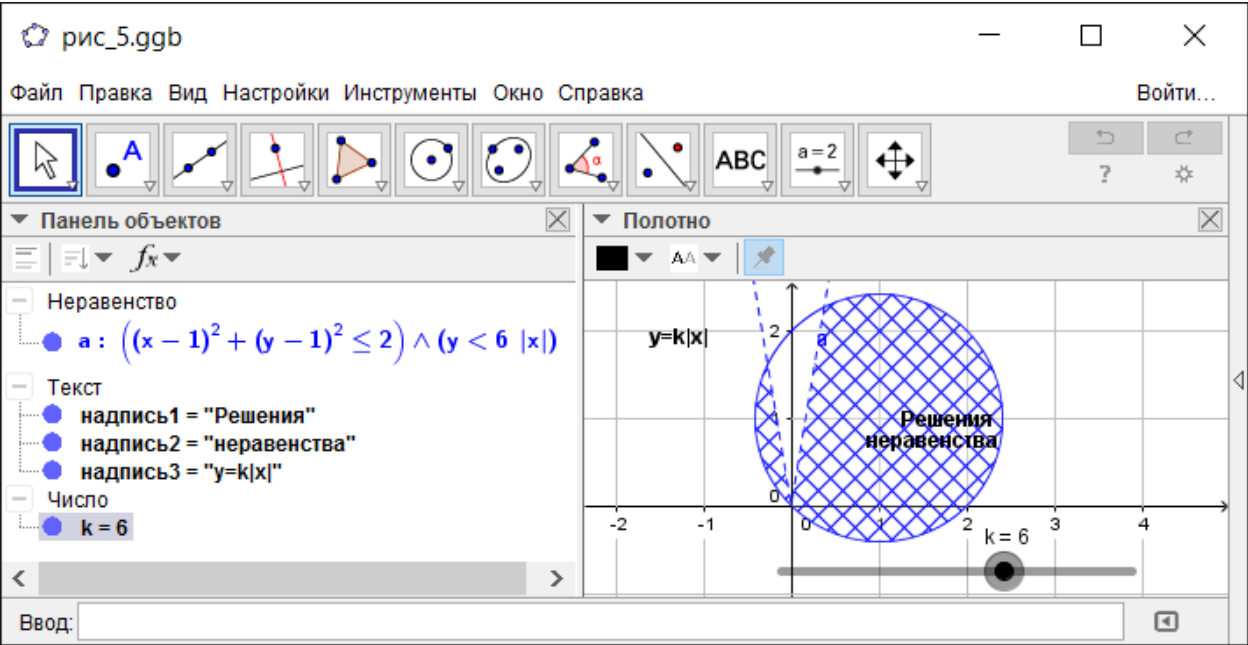


Рис. 5. Графическое решение параметрической системы неравенств (2) при $-1 \leq k \leq 10$

2.6. Создание динамических моделей

Продemonстрируем, как можно использовать для создания динамических моделей только алгебраические средства – команды или только геометрические средства – инструменты. Конечно, в реальных ситуациях при решении задач алгебраические и геометрические средства применяются совместно. Но монопольное их использование способствует более быстрому и эффективному обучению основам *GeoGebra*.

В качестве примера будем создавать модель под именем *sincos* построения и вычисления тригонометрических функций $\sin(x)$ и $\cos(x)$ ($0 \leq x \leq 2\pi$). На рис. 6 на панели “Полотно” представлен статичный вариант этой модели, состоящей из 29 объектов. Функционировать модель должна так. Перемещение движка вдоль основы ползунка должно плавно изменять значение a ($0 \leq a \leq 2\pi$) и всех объектов, связанных с a . Текущему значению a должно соответствовать и выводимое на рис. 6 изображение. Аналогично должен вести себя рисунок и при протягивании точки $D=(a, 0)$ по оси абсцисс в пределах от 0 до 2π . Кнопкой “Анимация” должен запускаться процесс автоматического изменения a от 0 до 2π , от 2π до 0 и т. д. с актуализацией соответствующих объектов на рисунке. Этой же кнопкой анимация должна останавливаться.

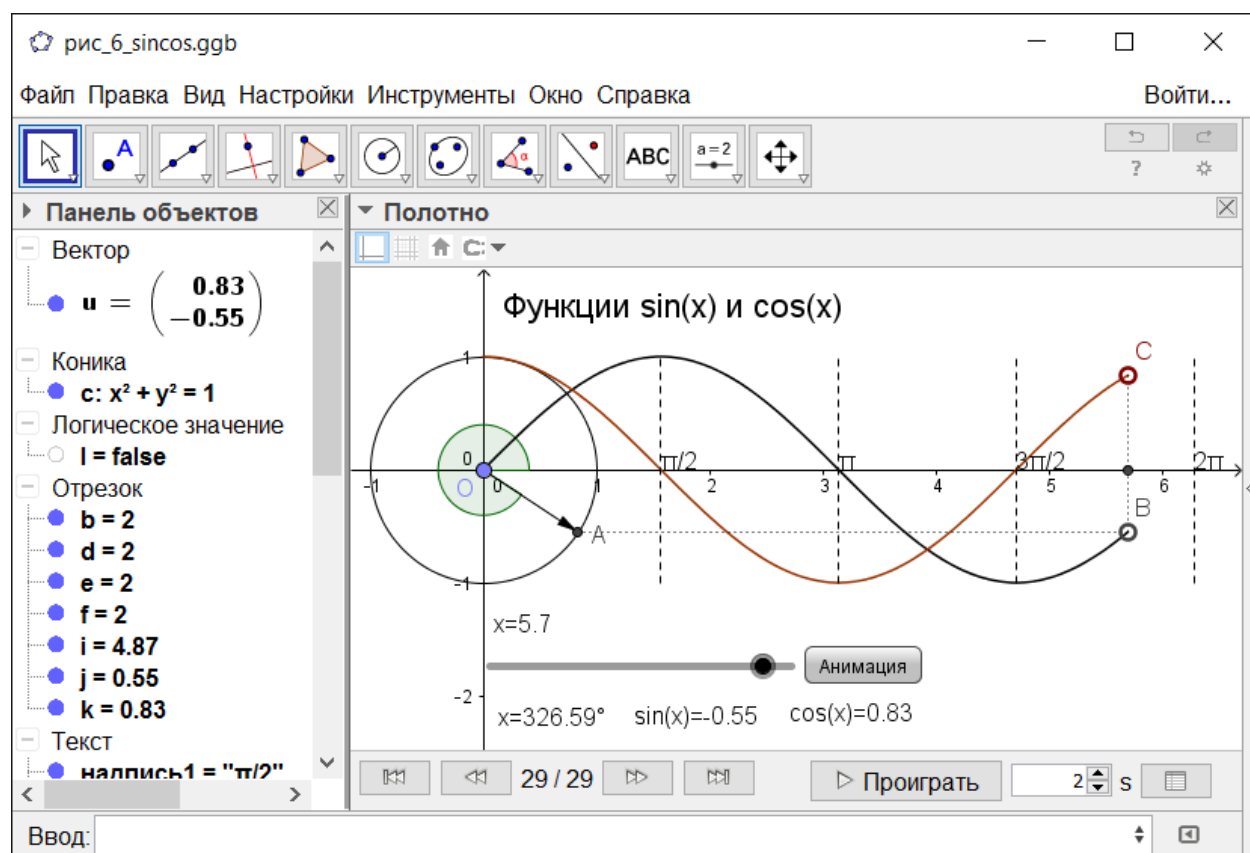



Рис. 6. Модель построения графиков функций $\sin(x)$ и $\cos(x)$

Описание функционирования модели вместе со списком всех составляющих ее (элементарных) объектов и является основой или базой для создания этой модели. Но формировать список объектов мы не будем, а сразу приступим к построению модели *sincos*. Будем считать, что панели “Объекты”, “Полотно” и “Ввод” (*Вид/Строка ввода*) открыты.


2.6.1. Использование алгебраических средств

Будем формировать модель в основном с помощью встроенных команд, вводимых со строки ввода. Ввод любой такой команды создает на панели “Объекты” описание некоторого алгебраического объекта, а по нему на панели “Полотно” формируется соответствующий геометрический образ этого объекта. Создавать отдельные объекты модели будем в приведенном ниже порядке их перечисления:

1) если в строке ввода набрать текст “ $O=(0, 0)$ ” (без кавычек), то на панели “Объекты” получим описание объекта “ $O=(0, 0)$ ”, а на панели “Полотно” – геометрический образ объекта в виде реальной точки O в начале координат. Однако такая точка является свободной и ее можно перемещать мышью по полотну на любое расстояние и в любом направлении. Нам же нужна точка O , которая не может быть смещена из начала координат. Закрепить или наоборот открепить O можно разными способами, например, через контекстное меню установкой *RClick($O=(0, 0)$)\Свойства\Основные\Закрепить объект*. Но можно и непосредственно создать закрепленную точку O , причем такую, которую открепить нельзя. Для этого через строку ввода следует выполнить команду “ $O=Pointin(x^2+y^2\leq 0)$ ”, в которой аргумент определяет разрешенную область для перемещения O . В нашем случае такая область состоит лишь из одной точки $(0,0)$;

2) если ввести уравнение $x^2+y^2=1$, то на панели “Объекты” получим объект “ $c:x^2+y^2=1$ ”, где c – имя объекта, а на панели “Полотно” – окружность c с центром в начале координат радиуса 1. Однако такая окружность является свободной и ее можно будет перемещать мышью по полотну в любом направлении. Нам же нужна “закрепленная” окружность с центром в O радиуса 1. Создать ее можно командой “ $Circle(O,1)$ ”. Заметьте, что в отличие от имени точки O , имя окружности мы сами не задавали. Оно было сформировано системой. Но на чертеже имя c показывать не требуется, поэтому отменим его вывод через контекстное меню (*RClick(объект)/Показывать обозначение*) или через локальное пиктографическое меню *Click(▶)/Click(объект)/Click()/Не показывать*);

3) введем любое число, например, 5.7. Получим объект “ $a=5.7$ ”. Через контекстное меню откроем на панели “Настройки” вкладку “Объекты (Свойства)” и на странице “Ползунок” установим для a интервал изменения и шаг: “мин.:0 макс.:6.28 шаг:0.05”. Закроем панель “Настройки” и щелкнем по белой радиокнопке слева от объекта “ $a=5.7$ ”. Кнопка станет синей и на панели

“Полотно” появится ползунок. Спрячем его обозначение и перетащим ползунок, если требуется, на другое место (*Click*()/*Тащи и бросай*);

4) введем точку $(\cos(a), \sin(a))$. Получим объект “ $A=(\cos(a), \sin(a))$ ” при текущем значении a и на панели “Полотно” его образ в виде точки A на единичной окружности;

5) командой $Vector(A)$ введем вектор. Получим объект “ $u = \begin{pmatrix} \cos(a) \\ \sin(a) \end{pmatrix}$ ” при текущем значении a и на панели “Полотно” его образ в виде радиус-вектора \overrightarrow{OA} . Имя u вектора \overrightarrow{OA} на панели “Полотно” сделаем невидимым;

6) командой $Angle(u)$ введем угол. Получим объект “ $\alpha=значение$ ” (значение угла в градусах) и на панели “Полотно” его образ в виде угла между положительным направлением оси абсцисс и радиусом-вектором \overrightarrow{OA} . Через панель “Настройки” изменим стиль вывода угла и спрячем его обозначение;

7-10) командами $Segment((\pi/2, -1), (\pi/2, 1))$, $Segment((\pi, -1), (\pi, 1))$, $Segment((3\pi/2, -1), (3\pi/2, 1))$ и $Segment((2\pi, -1), (2\pi, 1))$ создадим объекты: “ $b=2$ ”, “ $d=2$ ”, “ $e=2$ ” и “ $f=2$ ” и на панели “Полотно” соответствующие им 4 отрезка прямых между указанными точками. Число 2 в записи объектов – это длины отрезков. Через страницу “Основные” вкладки “Объекты (Свойства)” панели “Настройки” запретим вывод имен этих объектов, для чего удалим флажки в кнопках “Показывать обозначение”, и на странице “Стиль” изменим стиль вывода отрезков линий на штриховой;

11-14) командами $Text[\pi/2, (\pi/2, 0)]$, $Text[\pi, (\pi, 0)]$, $Text[3\pi/2, (3\pi/2, 0)]$, $Text[2\pi, (2\pi, 0)]$ создадим объекты: “надпись1= $\pi/2$ ”, “надпись2= π ”, “надпись3= $3\pi/2$ ” и “надпись4= 2π ”. Соответствующие им значения $\pi/2$, π , $3\pi/2$ и 2π на панели “Полотно” появятся около позиций, указанных вторыми аргументами функций;

15-16) командами $Function(\sin(x), 0, a)$ и $Function(\cos(x), 0, a)$ сформируем объекты “ $g(x)=\sin(x) (0 \leq x \leq a)$ ” и “ $h(x)=\cos(x) (0 \leq x \leq a)$ ”. Их образы на панели “Полотно” – это графики функций $\sin(x)$ и $\cos(x)$, заданные на отрезке $[0, a]$. Через панель “Настройки” цвет и другие свойства выведенных кривых можно изменить;

17-19) введем точки $(a, \sin(a))$, $(a, \cos(a))$ и $(a, 0)$. Получим объекты “ $B=(a, \sin(a))$ ”, “ $C=(a, \cos(a))$ ” и “ $D=(a, 0)$ ” при текущем значении a . На панели “Полотно” им соответствуют точки B , C и D . Через панель “Настройки” изменим размер и форму точек C и B и спрячем имя точки D ;

20-22) командами $Segment(A, B)$, $Segment(B, D)$ и $Segment(C, D)$ сформируем объекты “ $i=длина1$ ”, “ $j=длина2$ ” и “ $k=длина3$ ” и их образы на панели “Полотно” в виде 3 отрезков прямых. Здесь: $длина1$, $длина2$ и $длина3$ – длины соответствующих отрезков. Через панель “Настройки” изменим начертание выведенных отрезков и спрячем их имена;

23-26) командами $Text(x=\sin(a))$, $Text(x=\cos(a))$, $Text(\sin(x)=\sin(a))$, $Text(\cos(x)=\cos(a))$ сформируем четыре объекта: “надпись5= $x=a$ ”, “над-

пись6=“ $x=\alpha^\circ$ ””, “надпись7=“ $\sin(x)=\sin(a)$ ”” и “надпись8=“ $\cos(x)=\cos(a)$ ””. Образами объектов на полотне являются тексты с постоянной и переменной частью, указывающие соответственно текущие значения угла в радианах, угла в градусах, синуса и косинуса. Переместим надписи, появляющиеся около начала координат, поближе к ползунку, разместив первую из них над ползунком, а остальные – под ним;

27) введем значение *false*. Это приведет к созданию объекта “ $l=false$ ”, то есть булевой переменной *l* со значением *false*. Эта переменная будет использована для написания простого скрипта, управляющего анимацией по параметру *a* с помощью кнопки “Анимация” (см. ниже). Если у объекта *l* включить радиокнопку, то будет видно, что этому объекту на панели “Полотно” соответствует управляющий элемент “Флажок” (независимый переключатель). Мы этого делать не будем;

28) командой *Button*(“Анимация”) создадим объект “кнопка1” с реальной управляющей кнопкой “Анимация” на панели “Полотно”. Перетащим кнопку в требуемое место и на странице “Сценарий” вкладки “Объекты (Свойства)” панели “Настройки” для события “По щелчку” напомним *GeoGebra*-скрипт из двух строк:





$$l=If(l==false, true, false)$$



$$StartAnimation(a, l)$$

Приведенный скрипт будет выполняться при каждом щелчке левой кнопки мыши по кнопке “Анимация”. И это обеспечивает следующее поведение модели. Пусть анимации по *a* нет. Тогда $l=false$ и по щелчку начинает выполняться скрипт. По команде *If* логическое значение *l* изменяется на противоположное, то есть получим $l=true$. С учетом этого вторая команда *StartAnimation* запускает анимацию по *a*. Пусть теперь анимация по *a* проводится. Тогда $l=true$ и по щелчку снова начинает выполняться скрипт. По команде *If* получим $l=false$ и с учетом этого вторая команда *StartAnimation* приостанавливает анимацию на текущем значении *a*.

29) командой *Text*(“Функции $\sin(x)$ и $\cos(x)$ ”) сформируем объект “надпись9=“Функции $\sin(x)$ и $\cos(x)$ ””, а соответствующую надпись на панели “Полотно” сместим в требуемое место и через панель “Настройки” изменим для нее размер шрифта.

Замечания. 1. На панель “Полотно” можно вывести “Навигационный бар” с управляющими элементами для проигрывания этапов построения модели (см. рис. 6). Для этого на странице “Основные” вкладки “Полотно” панели “Настройки” в разделе “Шаги построения” достаточно включить независимый переключатель “Показать”, а также указать, что показывать. После этого мы получаем такие возможности:

- с помощью кнопок “  29 / 29  ” вручную пошагово перемещаться по всем 29 этапам построения модели от начала и до конца или в обратном направлении;


- кнопками “ Пройграть”  2 s” запускать анимацию построения модели, задав временной интервал между индикацией соседних кадров.


Показать или спрятать “Навигационный бар” можно также командой *RClick(пустое место)/Шаги построения*.


2. Количество шагов при построении модели можно было бы сократить, если дополнительно использовать команду *Sequence*. Например, шаги 7-10 построения вертикальных отрезков можно реализовать командой *Sequence[Segment[($k \cdot \pi/2, -1$), ($k \cdot \pi/2, 1$)], k , 1, 4]*, а шаги 11-14 создания надписей около этих отрезков – командой *Sequence[Text[Element[{" $\pi/2$ ", " π ", " $3\pi/2$ ", " 2π "}, k], ($k \cdot \pi/2$, 0)], k , 1, 4]*.


2.6.2. Использование геометрических средств


Теперь создадим ту же самую модель *sincos* с помощью (почти) только геометрических средств – специальных инструментов визуального построения объектов, приводимых в действие с помощью кнопок пиктографического меню. В данном случае на панели “Полотно” будем создавать геометрические объекты, а по ним на панели “Объекты” будут формироваться их алгебраические описания. Действия по блокированию вывода обозначений объектов упоминать не будем. Итак, шаги по построению модели *sincos* могут быть следующими:




1) создадим точку $A(0, 0)$ (инструмент “ Точка на объекте”). Заменяем имя A в O ;


2) сформируем единичную окружность по точке O и радиусу, равному 1 (инструмент “ Окружность по центру и радиусу”);

3) создадим ползунок с установкой минимального и максимального значений, а также шага (инструмент “ Ползунок”);


4) разместим в любом месте полотна точку A (инструмент “ Точка”) и через панель “Настройки” подправим координаты A на $(\cos(a), \sin(a))$;

5) проведем радиус-вектор \overrightarrow{OA} (инструмент “ Вектор”);


6) сформируем точку B на положительном направлении оси абсцисс (инструмент “ Точка”), создадим лучи OB и OA (инструмент “ Луч”), построим угол между OB и OA (инструмент “ Угол”). Спрячем точку O и лучи OA и OB . Изменим стиль вывода угла, добавив к дуге стрелочку;


7-10) создадим где-нибудь на полотне отрезок длины 2 (инструмент “ Отрезок фиксированной длины”), скопируем его в буфер (ключ *Ctrl+c*), а затем скопируем 3 раза содержимое буфера на полотно (ключ *Ctrl+v*). У полученных 4 отрезков через панель “Настройки” поменяем координаты концевых точек соответственно на $((\pi/2, -1), (\pi/2, 1))$, $((\pi, -1), (\pi, 1))$, $((3\pi/2, -1), (3\pi/2, 1))$ и $((2\pi,$


-1), $(2\pi, 1)$) и изменим стиль вывода полученных отрезков на штриховой. Через контекстное меню удалим с чертежа лишние точки;


11-14) на оси абсцисс около середин отрезков, построенных в предыдущем пункте, создадим надписи $\pi/2$, π , $3\pi/2$ и 2π (инструмент  *Текст*). Делается это на специальной панели *“Текст”* желательнее при включенной кнопке *“LaTeX”*;


15-16) командами $Function(\sin(x), 0, a)$ и $Function(\cos(x), 0, a)$ через строку ввода сформируем объекты $g(x)=\sin(x) (0 \leq x \leq a)$ и $h(x)=\cos(x) (0 \leq x \leq a)$. Конечно, можно и не использовать строку ввода, а поступить, например, так. Создать на панели *“Полотно”* прямую и через панель *“Настройки”* изменить ее определение на $If(0 \leq x \leq a, \sin(x))$, создать еще одну прямую и изменять ее определение на $If(0 \leq x \leq a, \cos(x))$ и т. д. Но этот способ более трудоемкий;

17-19) создадим на полотне 3 произвольные точки (инструмент  *Точка*) и изменим их координаты на $(a, \sin(a))$, $(a, \cos(a))$ и $(a, 0)$. Получим точки $C=(a, \sin(a))$, $D=(a, \cos(a))$ и $E=(a, 0)$ при текущем значении a . Через панель *“Настройки”* изменим размер и форму точек C и D ;

20-22) проведем отрезки AC , CE и DE (инструмент  *Отрезок*) и через панель *“Настройки”* изменим их начертание. Спрячем точку E ;

23-26) создадим надписи с постоянными и переменными частями вида: $x=a$, $x=\alpha$, $\sin(x)=\sin(a)$, $\cos(x)=\cos(a)$ (инструмент  *Текст*, a – значение в радианах, α – значение в градусах). Для этого ввод на открывающейся панели *“Текст”* следует осуществлять соответственно так: $x=\boxed{a}$, $x=\boxed{\alpha}$, $x=\boxed{\sin(a)}$, $x=\boxed{\cos(a)}$, причем переменную часть вводить в пустую рамку из списка *“Объекты”*. Переместим надписи поближе к ползунку, разместив первую из них над ползунком, а остальные – под ним;

27) создадим переменную l с логическим значением *false* (инструмент  *Флажок*). Для этого сформируем объект *“Флажок”* и получим его имя l . Далее, через панель *“Настройки”* спрячем флажок, предварительно изменив его значение на *false*;

28) создадим управляющую кнопку с надписью *“Анимация”* для запуска и остановки анимации по параметру a (инструмент  *Кнопка*). Перетащим кнопку на требуемое место. На странице *“Сценарий”* вкладки *“Объекты (Свойства)”* панели *“Настройки”* для события *“По щелчку”* напишем GeoGebra-скрипт из двух строк:

$$l=If(l==false, true, false)$$
$$StartAnimation(a, l)$$

Это тот же самый скрипт, который использовался в 2.6.1 и его действие уже подробно описано.

29) сформируем надпись-заголовок “*Функции $\sin(x)$ и $\cos(x)$* ” (инструмент “Текст”), изменим через панель “Настройки” размер используемого шрифта и разместим надпись на требуемом месте.

2.6.3. Панель “Протокол”

Все команды, вводимые в текущей сессии через строку “Ввод”, заносятся в список команд текущей сессии. При активной строке ввода этот список можно раскрыть ключом “↑”. Перемещаясь по списку, ключом *Enter* можно продублировать требуемую команду в строку ввода для дальнейшего редактирования и использования. При следующем открытии документа этот список будет пуст и станет формироваться заново. Кроме списка команд текущей сессии система ведет и более полную, важную и информативную таблицу данных, называемую протоколом и выводимую на панель “Протокол”. Среди прочих данных в нем присутствует список команд, по которым выведены объекты панели “Полотно” вне зависимости от того, когда и как они создавались.

Через панель “Протокол” можно совершать разнообразные действия с существующей моделью: изменять свойства ее объектов, удалять объекты, представлять шаги построения (строки таблицы), осуществлять анимацию процесса построения с помощью клавиатуры или специальных управляющих элементов, расположенных внизу панели. Эти элементы аналогичны соответствующим элементам панели “Полотно”, но там их можно удалять, а в панель “Протокол” они встроены.


A. Вид/Протокол
B. *Ctrl+Shift+l*

Открытие панели “Протокол”

A-B) Командой A, выполняемой через основное меню, или ключом B открывается панель “Протокол”, на которой представлена таблица с различными данными, связанными с построением текущей модели панели “Полотно”.



Перечень выводимых столбцов протокола

Имена столбцов протокола таковы: “№”, “Имя”, “Иконка”, “Определение”, “Команда”, “Значение”, “Заголовок”, “Метка”. Можно показывать и прятать вывод любого из указанных столбцов, кроме “№”, выводимого принудительно. Делается это включением/выключением независимых переключателей в выборе-списке “ Столбцы” контекстного пиктографического меню.



Перемещение по строкам протокола и анимация построения

По строкам протокола можно перемещаться с помощью ключей “↓”, “↑”, *Home*, *End*. При этом текущая строка является выделенной, а на панели “Полотно” всегда видна та часть конструкции, которая соответствует строкам протокола от начальной и до текущей включительно. Любую строку протокола можно сделать текущей не последовательными перемещениями, а

сразу. Для этого достаточно выполнить команду *DClick(строка)*. Команда *DClick(заголовок)* равносильна действию ключа *Home*.



Перестановка в протоколе строк

Каждому объекту панели “Полотно” в протоколе соответствует одна или несколько строк. Переместить строку протокола вверх или вниз можно с помощью технологии “Тащи и бросай”. Такое перемещение приводит к изменению последовательности шагов построения модели и поэтому не всегда осуществимо.



Изменения свойств объектов

Командой *RClick(строка)* через контекстное меню можно изменить свойства объекта, соответствующего указанной строке, или удалить его. Объект текущей строки протокола можно удалить и ключом *Delete*. Заметим, что удаление строки может привести к удалению связанных с ней строк.



Вставка нового объекта в конструкцию

Если через панели “Объекты” или “Полотно” к конструкции добавить новый объект, то соответствующие ему строки протокола окажутся последними. Затем, как мы уже поняли, их можно перетащить в любые другие позиции. Однако можно сделать так, чтобы формируемые строки протокола сразу же оказались на требуемом месте. Для этого нужно выделить в протоколе строку, после которой должна быть вставка, и перейти к созданию объекта. Когда объект будет создан за выделенной строкой появится одна или несколько строк, соответствующих новому объекту.



Создание точек прерывания анимации построения модели

























По умолчанию при анимации последовательно с небольшим временным интервалом выполняются команды строк протокола от первой и до последней. Однако можно организовать анимацию и более крупными шагами, объединяющими в себе по несколько последовательных исходных шагов. Сделать это можно так. В столбце “Метка” (*Breakpoint*) тех строк протокола, с которых будут начинаться новые, более крупные шаги анимации, включить независимые переключатели. Далее, в списке “Опции  ” локального пиктографического меню включить независимый переключатель “Показывать только помеченные”. После этих действий строки протокола с невключенными переключателями окажутся спрятанными, а анимация будет совершаться новыми укрупненными шагами, каждый из которых реализуется командой строки, видимой в протоколе, и командами последующих скрытых строк до очередной видимой строки. Для того, чтобы вернуться к прежнему режиму анимации, достаточно выключить независимый переключатель “Показывать только помеченные”.

Таблица 1. Модель sincos

No.	Name	Toolbar Icon	Command	Value	Caption	Breakpoint
1	Point O		PointIn[$x^2 + y^2 \leq 0$]	$O = (0, 0)$		false
2	Circle c		Circle[O, 1]	$c: x^2 + y^2 = 1$		false
3	Number a			$a = 5.7$		true
4	Point A		(cos(a), sin(a))	$A = (0.83, -0.55)$		false
5	Vector u		Vector[A]	$u = (0.83, -0.55)$		false
6	Angle α		Angle[u]	$\alpha = 326.59^\circ$		true
7	Segment b		Segment[($\pi / 2, -1$), ($\pi / 2, 1$)]	$b = 2$		false
8	Segment d		Segment[($\pi, -1$), ($\pi, 1$)]	$d = 2$		false
9	Segment e		Segment[($3\pi / 2, -1$), ($3\pi / 2, 1$)]	$e = 2$		false
10	Segment f		Segment[($2\pi, -1$), ($2\pi, 1$)]	$f = 2$		true
11	Text надпись1		Text[" $\pi/2$ ", ($\pi / 2, 0$)]	$\pi/2$		false
12	Text надпись2		Text[" π ", ($\pi, 0$)]	π		false
13	Text надпись3		Text[" $3\pi/2$ ", ($3\pi / 2, 0$)]	$3\pi/2$		false
14	Text надпись4		Text[" 2π ", ($2\pi, 0$)]	2π		true
15	Function g		$g(x) = \text{If}[0 \leq x \leq a, \sin(x)]$	$g(x) = \text{If}[0 \leq x \leq 5.7, \sin(x)]$		false
16	Function h		$h(x) = \text{If}[0 \leq x \leq a, \cos(x)]$	$h(x) = \text{If}[0 \leq x \leq 5.7, \cos(x)]$		true
17	Point B		(a, sin(a))	$B = (5.7, -0.55)$		false
18	Point C		(a, cos(a))	$C = (5.7, 0.83)$		false
19	Point D		(a, 0)	$D = (5.7, 0)$		true
20	Segment i		Segment[A, B]	$i = 4.87$		false
21	Segment j		Segment[B, D]	$j = 0.55$		false
22	Segment k		Segment[C, D]	$k = 0.83$		true
23	Text надпись5		Text["x=" + a]	$x=5.7$		false
24	Text надпись6		Text["x=" + α]	$x=326.59^\circ$		false
25	Text надпись7		Text["sin(x)=" + (sin(a))]	$\sin(x)=-0.55$		false
26	Text надпись8		Text["cos(x)=" + (cos(a))]	$\cos(x)=0.83$		true
27	Boolean Value l			$l = \text{false}$		false
28	Button кнопка1			кнопка1	Анимация	true
29	Text надпись9		Text["Функции sin(x) и cos(x)"]	Функции sin(x) и cos(x)		true

Created with [GeoGebra](https://www.geogebra.org/m)

А. Экспортировать как Web-страницу ()

В. Печатать ()

С. Получить online-помощь ()

Действия с протоколом

Команды А-С выполняются через локальное пиктографическое меню.

А) Командой А прото-

кол экспортируется в Web-страницу в том виде, в каком он представлен. А именно, по А открывается панель “Экспорт: Протокол (html)”. На ней можно задать заголовок протокола, указать автора соответствующей модели, записать дату создания файла, сообщить, нужно ли к протоколу добавить рисунок с панели “Полотно” и т. п. Нажатие на кнопку “Экспорт” записывает протокол и сопутствующие данные в html-файл, который может быть просмотрен в любом браузере.

В) Командой В можно распечатать протокол.

С) Команда С позволяет получить online-справку по работе с протоколом, если она существует.

Вернемся к разделу 2.6.1, откроем созданный там файл *sincos.ggb*, укажем в протоколе некоторые точки прерывания для анимации и создадим html-файл протокола. Результат, представленный в браузере *Microsoft Edge*, показан в табл. 1. Заметим, что в протоколе фактически содержатся все данные для построения модели, кроме скрипта для кнопки “Анимация”.

3. Средства для работы с панелями “Объекты” и “Полотно”

3.1. Пиктографические меню панелей

Остановимся, прежде всего, на устройстве и кратком описании пиктографических меню встроенных инструментов для панелей: “Полотно” (*Graphics*), “Полотно 2” (*Graphics 2*), “Символьные вычисления” (*CAS*), “Таблица” (*Spreadsheet*), “Полотно 3D” (*3D Graphics*). Каждая из этих панелей имеет свое иерархическое пиктографическое меню, являющееся основным средством графического интерфейса пользователя. Меню инструментов состоит из меню-бара – постоянно находящейся на панели строки изменяющихся выборов, и связанных с этими выборами раскрывающихся меню (подменю), также состоящих из выборов. При этом, выборы меню-бара задаются пиктограммами инструментов, а выборы всплывающих подменю – пиктограммами инструментов и краткими поясняющими к ним текстами. Выборы меню-бара не являются самостоятельными – в любой момент времени каждый из них своей пиктограммой представляет один из выборов соответствующего всплывающего подменю. Щелчок левой кнопкой мыши по любому выбору подменю дублирует его в виде пиктограммы в соответствующую позицию меню-бара. В любой конкретный момент времени выборы меню-бара соответствуют тем выборам всплывающих подменю, которые были активны в последний раз. Меню-бар называют также главным меню или меню верхнего уровня (*main menu, top level menu*), а всплывающие подменю – раскрывающимися подменю (*pop up menu*). Отдельные выборы всплывающих меню и подменю, называют также кнопками или инструментами, поскольку они соответствуют определенным действиям на текущей панели.

В любых ситуациях только одна из кнопок главного меню обрамлена квадратом с жирной границей и является активной. Щелчок по активной кнопке запускает на выполнение последовательность действий, связанных с ее инструментом. Щелчок по неактивной кнопке главного меню делает ее активной. Щелчок по кнопке всплывающего подменю дублирует ее пиктограмму на соответствующую кнопку главного меню и делает ее активной. Как работать с конкретным инструментом зависит от его назначения. Описанием этого мы и будем заниматься в последующих разделах. Инструменты позволяют пошагово и наглядно строить на панелях “Полотно” (*Graphics*), “Полотно 2” и “Полотно 3D” как простые, так и весьма сложные динамические геометрические объекты. При этом на панели “Объекты” автоматически формируются соответствующие им алгебраические описания этих объектов или их совокупностей.

а) Меню-бар панели “Полотно” и “Полотно 2”



б) Меню-бар панели “Таблица”



с) Меню-бар панели “Символьные вычисления” (GAS)



д) Меню-бар панели “Полотно 3D”



Рис. 1 (а-д). Меню-бары пиктографических меню для разных панелей

Панели “Полотно”, “Полотно 3D”, “Символьные вычисления” и “Таблица” имеют свои собственные пиктографические меню, для панели “Полотно 2” используется то же самое меню, что и для панели “Полотно”. Меню-бары пиктографических меню для разных панелей приведены на рис. 1. На рис. 2 для панелей рис. 1 приведены всплывающие пиктографические меню инструментов в раскрытом виде. В действительности одновременно развернуть каждое всплывающее меню невозможно. Разворачивание одного меню делает его активным (текущим) и закрывает предыдущее открытое меню.

3.2. Пиктографическое меню панели “Полотно”

На рис. 2 приведено пиктографическое иерархическое меню инструментов панелей “Полотно” и “Полотно 2” с раскрытыми всплывающими подменю, где на кнопках приведены пиктограммы без поясняющих текстов. В последующих пунктах кратко описывается работа с каждым инструментом каждого отдельного подменю.

Многие действия можно выполнять не только инструментами, но и конкретными командами через строку ввода. Однако точного соответствия между инструментами и командами не существуют. В ряде случаев результаты выполнения “родственных” инструментов и команд могут незначительно отличаться друг от друга. А для некоторых команд соответствующие им встроенные инструменты вообще отсутствуют, хотя создать их можно. Например, существует команда проведения внешних и внутренних касательных к двум окружностям, но встроенного инструмента для выполнения таких действий

нет. Везде ниже описание работы со встроенными инструментами будет сопровождаться рассмотрением примеров их использования в конкретных ситуациях и завершаться кратким описанием тем или иным образом связанных с ним команд. За редким исключением примеры для команд не приводятся.

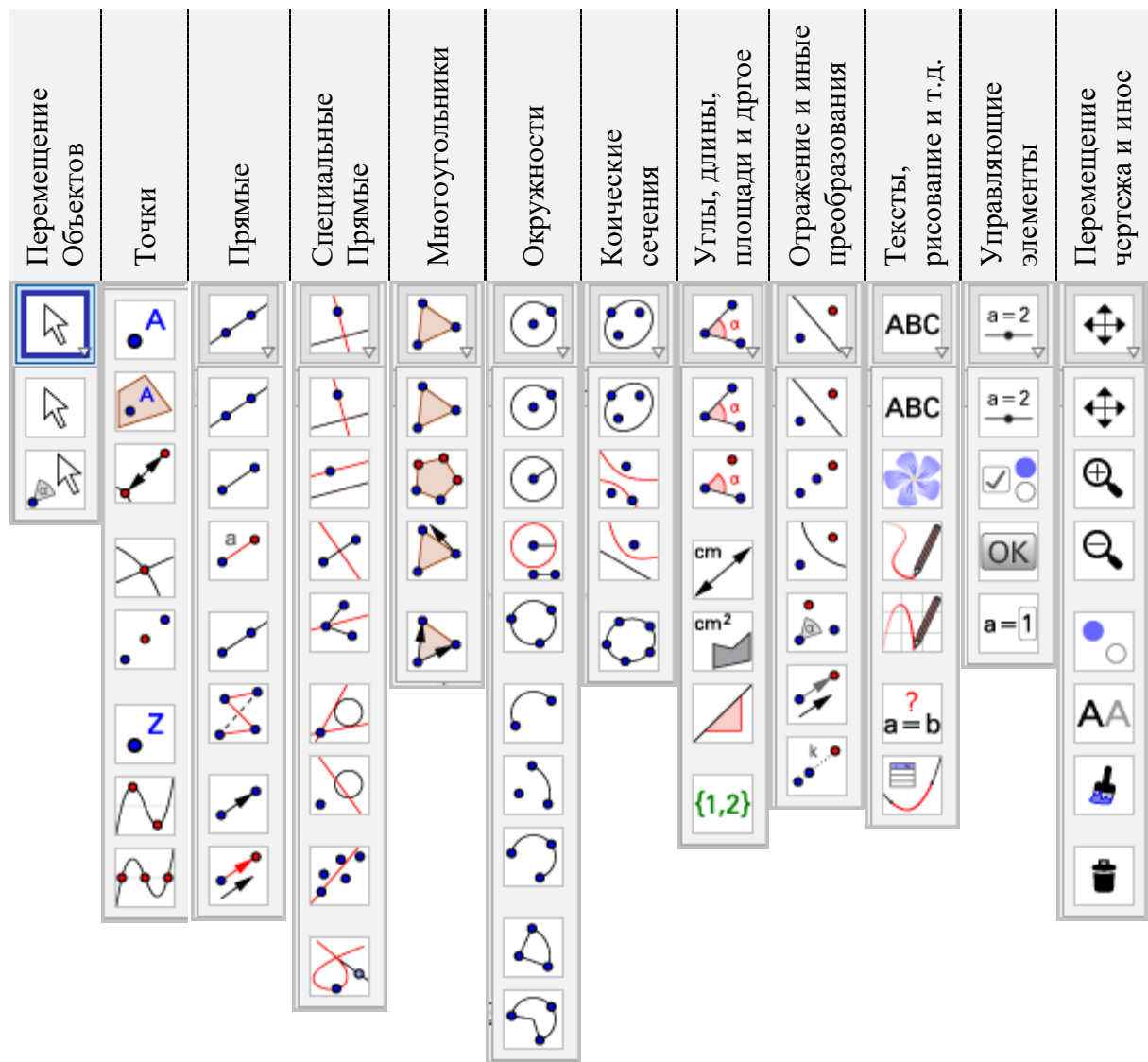



Рис. 2. Пиктографическое меню панелей “Полотно” и “Полотно 2” с раскрытыми всплывающими подменю

3.3. Инструменты и команды


3.3.1. Перемещение объектов

Для перемещения объекта необходимо прежде всего активировать инструмент “ Перемещать” щелчком левой кнопки мыши или клавишей *Esc*. Далее требуется выделить объект, щелкнув по нему левой кнопкой мыши, и, не отпуская ее, протягивать курсор по полю. При отпускании кнопки операция перемещения завершается. Эта операция называется “*Тащи и бросай*”


(*Drag and Drop*). Перемещать объекты можно не только мышью, но и клавишами “ ω ” и “ $key+\omega$ ”, где $\omega \in \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$, $key \in \{Shift, Alt, Ctrl\}$. Разрешается перемещать по полотну сразу несколько объектов. Но для этого их необходимо предварительно выделить. Делается это так:



- производится щелчок левой кнопкой мыши по первому из объектов;
- нажимается клавиша *Ctrl* и не отпуская ее, производятся последовательные щелчки по другим объектам;
- клавиша *Ctrl* отпускается;
- щелчок по любому из выделенных объектов перемещает всю их совокупность по полотну как единый объект. Эту операцию можно совершать повторно, начиная с другого выделенного объекта;
- щелчок по полотну вне выделенных объектов завершает групповое перемещение объектов.




Перемещение объекта. Инструмент “ *Перемещать*” используется для перемещения объектов Ω по полотну способом “*Тащи и бросай*”. Если Ω является частью составного объекта, то перемещения Ω могут проводить к различным преобразованиям связанных с ним объектов. Вместе с Ω перемещается и его обозначение. Однако, обозначение в определенных пределах можно перемещать и отдельно, не меняя позиции самого объекта.

Отметим, что любой объект Ω можно закрепить на полотне, то есть запретить любые его перемещения и удаление. Для этого для данного объекта через панель “*Настройки*” следует включить независимый переключатель “*Закрепить объект*”. При этом остается возможность изменения Ω через связанные с ним объекты. Скажем, если закрепить отрезок, то изменить его местоположение и размеры не удастся, но через концевые точки отрезка это сделать можно. Чтобы объект Ω нельзя было подвергнуть изменениям, требуется закрепить все его свободные объекты. Чтобы на чертеже вообще ничего нельзя было изменить, следует выделить все его объекты ($Ctrl+a$) и через панель “*Настройки*” включить независимый переключатель “*Закрепить объект*”.

Замечание. Очень важно! При создании чертежа работу с любым инструментом лучше всего завершать переходом к инструменту “*Перемещать* ”. Это позволяет снять имеющиеся выделения объектов и избежать последующего случайного нанесения на чертеж дополнительных объектов.

Пример 1. Инструментами “ *Точка*” и “ *Отрезок*” постройте составной объект $\triangle ABC$ из 6 элементов: 3 точек и 3 отрезков (см. рис. 3). При этом позаботьтесь о том, чтобы против вершины A находилась сторона a , против вершины B – сторона b и против вершин C – сторона c . Поэкспериментируйте с треугольником, перемещая каждый из 6 его элементов по полотну.

Пример 2. Инструментом “ Многоугольник” постройте объект $\triangle DEF$. Обратите внимание, что здесь имена сторон треугольника автоматически создаются такими же, как и у противоположных вершин (но малыми буквами). Убедитесь, что возможно перемещение всех его точек и сторон, а также и самого треугольника, выделяя его щелчком по любой внутренней точке.

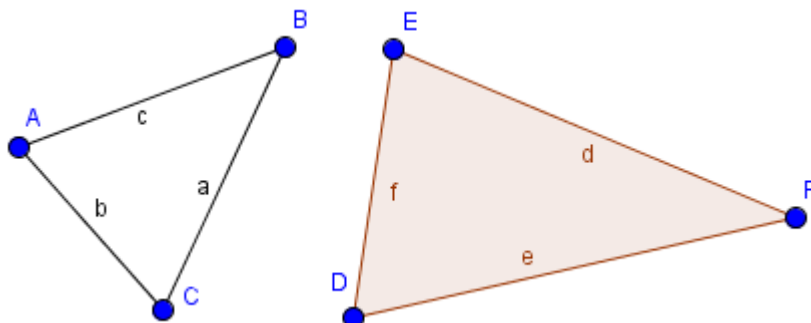




Рис. 3. Объекты для перемещений

 Вращение *относительно некоторого центра*. Инструмент “ Движение относительно точки” используется для вращения объектов вокруг некоторой неподвижной точки A . Для выполнения операции сначала необходимо щелкнуть по точке A , а затем щелкать по другим объектам и вращать их. При этом каждая точка B из вращаемого объекта Ω будет перемещаться по окружности с центром в точке A и радиуса $r=AB$. Чтобы сменить точку, вокруг которой реализуется вращение, необходимо переключиться на другой инструмент и снова вернуться к данному инструменту.

Пример 3. Поэкспериментируйте с объектами рис. 3, вращая любые 6 элементов $\triangle ABC$ (3 вершины и 3 стороны) вокруг точки E . Поэкспериментируйте с объектами рис. 3, вращая любые 6 элементов $\triangle EDF$ (3 вершины и 3 стороны) и сам $\triangle EDF$ вокруг точки B .

A. $Rotate[obj, \alpha]$
 B. $Rotate[obj, \alpha, A]$
 C. $RotateText[“текст”, \alpha]$




Вращение объектов

Вращать объекты можно с помощью команды $Rotate$, выполняемой через командную строку. Вращение происходит на некоторый угол α против часовой стрелки. Если у угла α единицы измерения не заданы, то считается, что это радианы. Градусы должны быть указаны явно: $30^\circ, 90^\circ, \dots$, а вводить их можно ключом $Alt+o$.

A) Командой A создается новый объект, полученный поворотом объекта obj вокруг начала координат. При этом объект obj свою позицию не изменяет. Если obj – это вектор, то вращение происходит не вокруг начала координат, а вокруг начальной точки вектора. Если obj – это текст, то для вращения obj следует использовать команду C .

В) Команда B выполняется аналогично A , но вращение происходит не вокруг начала координат, а вокруг заданной точки A . Если obj – это вектор, то вращение происходит не вокруг точки A , а вокруг начальной точки вектора.

С) Команда C выполняет вращения объекта “текст”. Первый аргумент команды – это некоторый текст обязательно в кавычках или обозначение (имя) текста без кавычек. Вращение происходит вокруг левой верхней точки текстового бокса. Вывод созданного объекта реализуется где-то рядом с началом координат, а затем его можно перетащить в любое другое место полотна.

Примеры 4. Инструментом “ Многоугольник” постройте $\triangle ABC$ (он получит имя $poly1$). Инструментом “ Вектор” добавьте на полотно вектор \overrightarrow{DE} , а инструментом “ Текст” – фразу “Это текст” (см. рис. 4). Текст обычно появляется в левом верхнем углу полотна и его следует перетащить на нужное место. Последовательными командами $Rotate[poly1, 30^\circ]$, $Rotate[u, 30^\circ]$ и $RotateText[“Это текст”, 30^\circ]$, выполняемыми через строку ввода, осуществим вращение каждого из трех выведенных объектов на 30° против часовой стрелки. Вывод по $RotateText$ обычно появляется в нижнем левом углу полотна и его следует перетащить на нужное место. Результат выполнения проведенных действий показан на рис. 4. Поэкспериментируйте далее с вводом последовательных команд $Rotate[poly1, \pi/3]$, $Rotate[u, \pi/3]$ и $RotateText[“Это текст”, \pi/3]$.

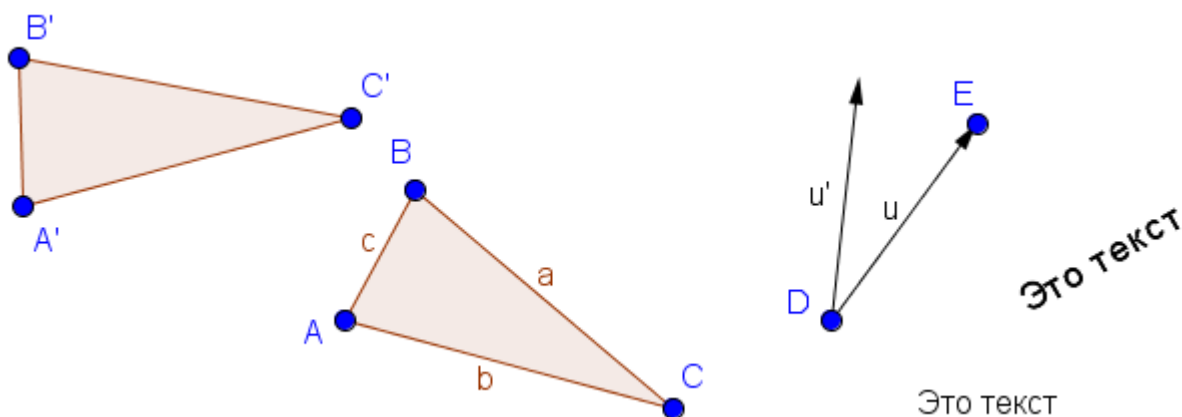





Рис. 4. Вращение объектов


3.3.2. Перемещение и масштабирование чертежа


Здесь рассмотрены инструменты, позволяющие перемещать чертеж по полотну, изменять масштаб по осям координат, скрывать и показывать объекты и их обозначения, а также удалять объекты и копировать их стили.




Переместить чертеж. При активации инструмента “ *Переместить чертеж*” курсор приобретает вид “” и включается режим, который позволяет выполнять такие операции:

1) *Переместить чертеж по полотну* вместе с системой координат. Для выполнения операции необходимо щелкнуть по полотну левой кнопкой мыши (вне осей координат) и, не отпуская кнопку, протягивать курсор по полотну в требуемую позицию. Заметим, что эту операцию можно проводить и инструментом “ *Перемещать*”;


2) *Изменить масштаб по оси абсцисс.* Подвести курсор к горизонтальной оси вне начала координат и, когда он примет вид “”, щелкнуть левой кнопкой мыши и, не отпуская ее, протягивать курсор вдоль оси в ту или в другую сторону. При протяжке вправо масштаб по оси абсцисс будет увеличиваться, а при протяжке влево – уменьшаться. При этом рядом с курсором выводится текущее отношение масштабов по осям в виде “ $x:y=\alpha:1$ ” (α – число);

3) *Изменить масштаб по оси ординат.* Подвести курсор к вертикальной оси вне начала координат и, когда он примет вид “”, щелкнуть левой кнопкой мыши и, не отпуская ее, протягивать курсор вдоль оси в ту или в другую сторону. При протяжке вверх масштаб по оси ординат будет увеличиваться, а при протяжке вниз – уменьшаться. При этом рядом с курсором выводится текущее отношение масштабов по осям в виде “ $x:y=1:\beta$ ” (β – число).


Замечание. Перечисленные выше операции перемещения и изменения масштаба можно выполнять не только при активном инструменте “ *Переместить чертеж*”, но и при других активных инструментах, если при этом нажата клавиша *Shift* или *Ctrl*.

Пример 5. Поэкспериментируйте с чертежом рис. 3, перемещая его по полотну и изменяя масштаб по отдельным осям.



Увеличить. Инструмент “ *Увеличить*” используется для увеличения масштаба по обеим осям координат. Операция проводится щелчком левой кнопкой мыши в некоторой позиции *P* полотна. При выполнении операции не только увеличивается масштаб, но и происходит смещение чертежа так, чтобы точка *P* оставалась на экране неподвижной.







Уменьшить. Инструмент “ *Уменьшить*” используется для уменьшения масштаба по обеим осям координат. Операция проводится щелчком левой кнопкой мыши по некоторой позиции *P* полотна. При выполнении операции не только уменьшается масштаб, но и происходит смещение чертежа так, чтобы точка *P* оставалась на экране неподвижной.

Другой способ увеличения или уменьшения масштаба сразу по двум осям координат – это вращение в ту или иную сторону колесика мыши. И это возможно при любом активном инструменте.


Пример 6. Поэкспериментируйте с чертежом, представленным на рис. 3, изменяя его масштаб одновременно по обеим осям.



Показать/Скрыть объект. Инструмент “ *Показать/Скрыть объект*” применяется для скрытия выделенных объектов или временного показа всех ранее скрытых объектов чертежа. При этом выделенные объекты скрываются при смене инструмента “” на любой другой инструмент. А для того, чтобы ранее скрытые объекты стали видимыми, достаточно активировать рассматриваемый инструмент “”. Однако при смене инструмента “” они опять окажутся спрятанными. Сделать эти объекты снова постоянно видимыми можно или через контекстное меню, или включением соответствующих радиокнопок на панели “Объекты”. Отметим, что при скрытии объекта его обозначение также скрывается.


Пример 7. Поэкспериментируйте с чертежом, представленным на рис. 3, показывая и скрывая его объекты. Сделайте невидимыми все вершины $\triangle EDF$.



Показать/Скрыть обозначение. Инструмент “ *Показать/Скрыть обозначение*” применяется для скрытия и показа обозначений видимых объектов. Для выполнения операции в данном режиме необходимо просто щелкать по требуемым объектам. Если имя объекта скрыто, то оно появится, а если видимо, то исчезнет. Показывать и скрывать обозначения конкретных объектов удобно также через контекстное меню.


Пример 8. Поэкспериментируйте с чертежом, представленным на рис. 3, показывая и скрывая обозначения его объектов.



Копировать стиль. Инструмент “ *Копировать стиль*” используется для копирования стиля вывода или некоторых элементов стиля вывода одного объекта на другие объекты не обязательно такого же типа. Работать с этим инструментом надо так: щелкнуть по объекту, являющемуся донором стиля, а затем щелчками по иным объектам переносить стиль на них.

Пример 9. Скопируйте на чертеже рис. 3 коричневый цвет с одной из сторон $\triangle DEF$ на все вершины $\triangle ABC$.






Удалить. Инструмент “ **Удалить**” применяется для установки режима удаления с чертежа ненужных объектов щелчками по ним левой кнопкой мыши. Выделенные объекты можно удалять также ключом *Delete*. Восстанавливаются удаленные объекты последовательными нажатиями ключа *Ctrl+z* или через меню командой *Edit/Undo* (откаты назад).

Пример 10. Поэкспериментируйте с чертежом, представленным на рис. 3, удаляя и восстанавливая на нем те или иные объекты.





3.3.3. Точки

Здесь рассмотрены инструменты, которые позволяют создавать точки, являющиеся свободными или связанными с другими объектами; преобразовывать свободные точки в связанные и наоборот; фиксировать точки пересечения линий; находить “центральные” (срединные) точки; а также устанавливать точки, соответствующие комплексным числам. Отметим, что для всех формируемых точек, впрочем, как и для других объектов, автоматически создаются обозначения, которые можно изменять, а также прятать, то есть делать их невидимыми на чертеже. Выполняются эти операции через контекстное меню или через панель “*Настройки*”. На последней можно изменять и иные свойства объектов, например, у точек изменить размер, стиль вывода (форму), цвет и т. п.



Точка. Инструмент “ **Точка**” используется для формирования точки в любой позиции полотна. Для выполнения операции требуется просто щелкнуть мышью в нужном месте. Точка может быть свободной или связанной. Точка, установленная на пустом месте полотна, внутри многоугольника, круга или эллипса по умолчанию является свободной. Точка, установленная на отрезке, прямой линии, стороне многоугольника, окружности, луче, векторе графике функции и т. п., является связанной с этим объектом. Инструментом “ **Прикрепить/Снять точку**” свободные и связанные точки можно превращать друг в друга. По умолчанию связанные точки имеют более бледную расцветку по сравнению со свободными точками. Свободную точку можно переместить в любую позицию полотна, и она сохраняет свою позицию при изменении других объектов. Связанная точка не может быть перемещена за пределы объекта, на котором она расположена, и может автоматически изменять свою позицию при преобразованиях этого объекта. Точка, установленная на пересечении нескольких объектов, связана с каждым из них (см. инструмент “ **Пересечение**”).

Через панель “Настройки” и связанные, и свободные точки могут быть зафиксированы (*fix*) на полотне. Зафиксированную точку напрямую переместить в другую позицию нельзя. Однако эта позиция может автоматически измениться при изменении объектов, связанных с этой точкой. Снятие фиксации с точек происходит тем же путем, что и ее установка.

Пример 11. Инструментами “ Точка”, “ Отрезок”, “ Многоугольник” и “ Окружность по центру и точке” создайте чертеж, представленный на рис. 5а, и поэкспериментируйте с ним, перемещая точки A, B, C, D, H, E, F . Убедитесь, что точки K, L и M являются связанными (зависимыми), а точки I и J – свободными. Проведите закрепление точки I внутри четырехугольника и открепление точек L и M соответственно от отрезка и окружности. Через панель “Настройки” зафиксируйте на полотне точки A, B и C . Убедитесь, что теперь четырехугольник можно изменять только протаскиванием точки D . Зафиксируйте точку L на отрезке EF . Убедитесь, что теперь позицию L напрямую изменить нельзя, но протаскиванием концов отрезка это сделать можно.

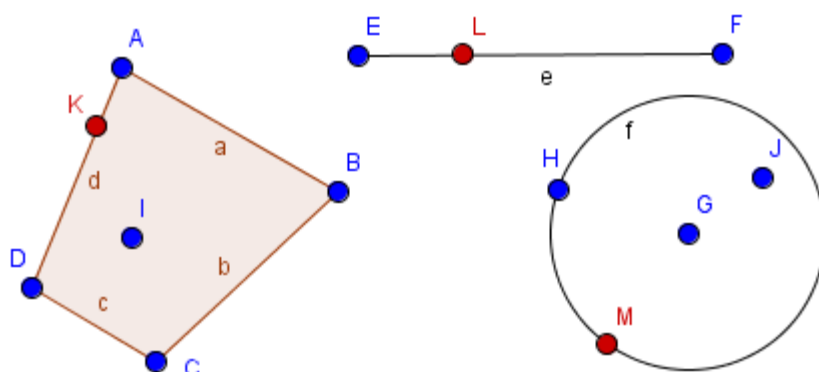







Рис. 5а. Работа с точками различных типов

Замечание. На панели “Полотно” точка обычно представляется своим именем, а на панели “Объекты” в виде $P=(a, b)$, где P – имя, a и b – абсцисса и ордината (a, b – действительные числа). Доступ к координатам точки осуществляется с помощью команд $x[P]$ и $y[P]$, возвращающих соответственно a и b . Эти команды могут выполняться в командной строке непосредственно или быть аргументами других команд.



Точка на объекте. Инструмент “ Точка на объекте”, как и инструмент “ Точка”, позволяет сформировать точку в любой позиции полотна. Разница в том, что в данном случае любая точка, установленная на объекте, является связанной, то есть прикрепленной к нему. Переместить такую точку за пределы объекта нельзя ни смещениями самой точки, ни изменениями формы или позиции объекта.

Замечание. По умолчанию непрозрачность (*opacity*) круга и эллипса равна нулю, что можно видеть для этих объектов на панели “*Настройки*”. Прикрепить точку к границе круга или эллипса можно всегда. Но прикрепить ее внутри круга или эллипса можно лишь при его непрозрачности, большей нуля.

Пример 12. На рис. 5а удалите через контекстное меню свободную точку *I*, сформированную инструментом “ Точка” и на этом месте создайте точку инструментом “ Точка на объекте”. Убедитесь, что теперь точку *I* переместить за пределы многоугольника нельзя. Через панель “*Настройки*” сделайте непрозрачность круга больше 0. Удалите через контекстное меню свободную точку *J* и на этом месте создайте точку инструментом “ Точка на объекте”. Убедитесь, что теперь переместить точку *J* за пределы круга нельзя.

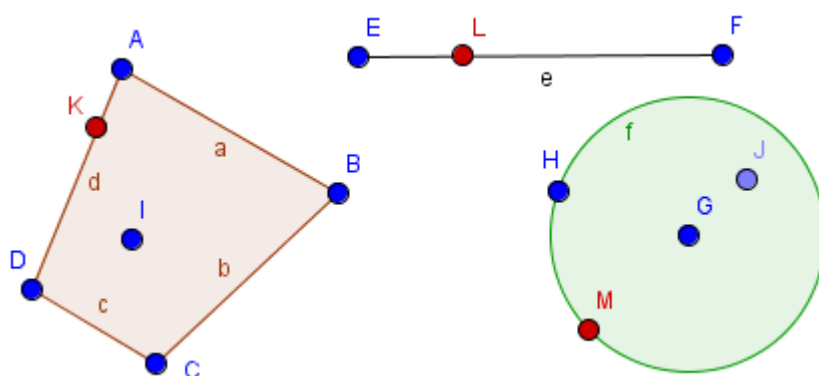




Рис. 5b. Точки на объектах



Прикрепить/Снять точку. Инструмент “ Прикрепить/Снять точку” позволяет щелчком по связанной точке сделать ее свободной и наоборот, свободную точку сделать связанной с конкретным объектом. Для выполнения последней операции необходимо щелкнуть по точке, а затем по объекту, к которому точка прикрепляется.

Пример 13. На рис. 5b сформируйте точки, связанные с отрезком, с окружностью и с кругом. Убедитесь, что переместить эти точки за пределы соответствующих объектов нельзя. Сделайте созданные точки свободными и убедитесь, что теперь их можно переместить в любую позицию полотна.



Пересечение. Инструмент “ Пересечение” используется для построения одной или нескольких точек пересечения двух кривых. Для выполнения операции необходимо указать мышью позицию пересечения кривых или указать поочередно эти кривые. В первом случае создается одна точка пересечения кривых, а во втором – по возможности все точки пересечения этих кривых.

A. <i>Intersect</i> [<i>obj1</i> , <i>obj2</i>] B. <i>Intersect</i> [<i>obj1</i> , <i>obj2</i> , <i>n</i>]	C. <i>IntersectPath</i> [<i>obj</i> , <i>poly</i>] D. <i>IntersectPath</i> [<i>poly1</i> , <i>poly2</i>]
---	---

Нахождение точек пересечения двух объектов


A) По команде A выводятся объекты *obj1* и *obj2* и, по возможности, находятся и выводятся точки их пересечения. Например, пусть через строку ввода введена окружность $x^2+y^2=5$ (объект *c*) и функция $y=x^2-3$ (объект *d*). Тогда по команде *Intersect*[*c*, *d*] получим 4 точки пересечения указанных кривых. Для получения точек пересечения этих кривых можно также использовать команду *Intersect*[$x^2+y^2=5$, $y=x^2-3$] и без вывода самих кривых.


B) Команда B выполняется как A, но выводится лишь *n*-я из найденных точек пересечения объектов *obj1* и *obj2*. Нумерация точек проводится натуральными числами. Если требуемой точки нет, то на панели “Объекты” появляется точка со статусом *undefined* (не определена), которой на панели “Полотно” ничего не соответствует.

C) По команде C выводится отрезок, являющийся пересечением объекта *obj* (прямая, отрезок, луч) и многоугольника *poly*. Если пересечение пусто, то выводится сегмент длины *undefined*.

D) По команде D выводится многоугольник, являющийся пересечением многоугольников *poly1* и *poly2*. Если пересечение пусто, то выводится многоугольник с площадью *undefined*.



Середина или центр. Инструмент “ *Середина или центр*” используется для построения точки в следующих случаях: посередине между двумя указанными точками, по центру выделенного отрезка, по центру выделенной плоской кривой, являющейся коническим сечением (коникой) – пересечением прямого кругового конуса плоскостью.

Пример 15. Используя инструмент “ *Середина или центр*” найдите точку посередине между точками A и B, середину отрезка CD, центр окружности, центр эллипса и центр гиперболы.

A. <i>Midpoint</i> [<i>A</i> , <i>B</i>] B. <i>Midpoint</i> [<i>seg</i>]	C. <i>Midpoint</i> [<i>a</i> < <i>x</i> < <i>b</i>]	D. <i>Midpoint</i> [коника] E. <i>Center</i> [коника]
---	---	--

Нахождение точек пересечения двух объектов


A) По команде A выводится точка, находящаяся посередине между точками A и B.

B) По команде B выводится точка, являющаяся серединой отрезка *seg*.

C) По команде C на панели “Объекты” формируется переменная со значением $(a+b)/2$, где *a* и *b* – действительные числа.


D-E) По командам D-E формируется центр коники (окружности, эллипса, гиперболы).




Комплексное число. Инструментом “ *Комплексное число*”, как и инструментом “*Точка*”, точки создаются щелчками мыши на полотне. Но в данном случае делается это так. Если точка формируется вне границ многоугольника, окружности и т. д., то она становится свободной, иначе – связанной. Обозначения связанных точек формируются обычным образом, а обозначения свободных точек – с индексами в виде z_1, z_2, \dots , хотя в дальнейшем их можно изменить. Наконец, на панели объектов данные точки записываются не в категорию “*Точка*” в форме “ $имя=(a, b)$ ”, а в отдельную категорию – “*Комплексное число*” в форме “ $имя=a+bi$ ”.

Пример 16. Создайте несколько свободных и связанных точек, соответствующих комплексным числам.



Extremum (экстремум). Пусть выведен график некоторой функции. Для указания прямо на нем локальных экстремальных точек можно использовать инструмент “ *Extremum*”. Если выведен график полинома, то определяются все локальные экстремальные точки. В противном случае вычисляются локальные экстремумы на некотором открытом интервале.

Пример 17. Выведем на полотно графики функций $f(x)=x \cdot \sin(2x+3)/3$ и $g(x)=0.1x \cdot (x-2)^3 \cdot (x-4)^2$. Для этого через строку ввода последовательно надо ввести выражения $x \cdot \sin(2 \cdot x+3)/3$ и $0.1 \cdot x \cdot (x-2)^3 \cdot (x-4)^2$. Далее, активируем инструмент “ *Extremum*” и щелкнем левой кнопкой мыши сначала по графику первой функции, а затем по графику второй функции. Результат этих действий можно видеть на рис. 7. На втором графике изменен стиль вывода экстремальных точек.

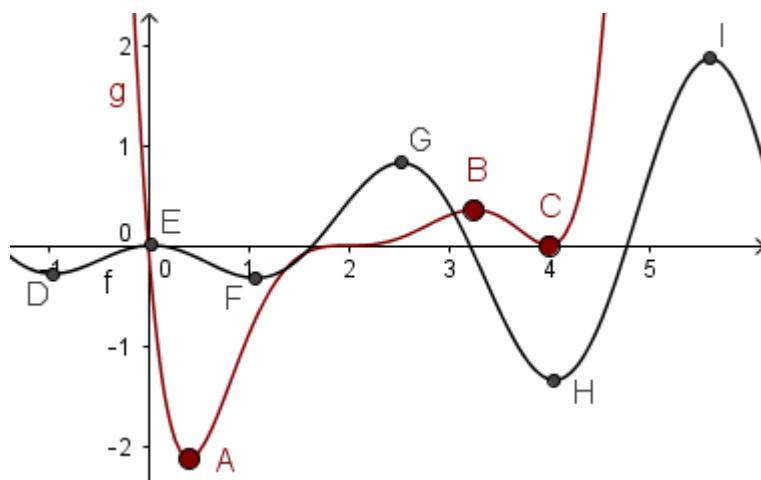


Рис. 7. Нахождение локальных экстремумов функций

A. *Extremum*[*fun*]
 B. *Extremum*[*fun*, *a*, *b*]

Вывод экстремальных точек
 А) Если выведен график некоторой функции с именем *fun*, то по команде *A* прямо на графике выводятся экстремальные точки *fun*. Если график функции не выведен, а *fun* – выражение, определяющее функцию ($x \cdot \cos(x)$, $x^2 + \sqrt{x}$, ...), то по *A* выводятся только экстремальные точки.

В) Команда *B* выполняется как *A*, но экстремальные точки отыскиваются и выводятся на заданном интервале (*a*, *b*).



Roots (корни). Пусть выведен график некоторой функции. Для указания прямо на нем всех корней этой функции, то есть точек пересечения ее с осью абсцисс, можно использовать инструмент “ *Roots*”. Если выведен график полинома, то определяются все корни. В противном случае корни вычисляются на некотором открытом интервале.

Пример 18. Вводом выражения $0.1 \cdot (x-1) \cdot (x-3)^3 \cdot (x-5)^2$ выведем на полотно график функции $g(x) = 0.1(x-1) \cdot (x-3)^3 \cdot (x-5)^2$. Далее, активируем инструмент “ *Roots*” и щелчком левой кнопкой мыши по графику. Результат этих действий можно видеть на рис. 8.

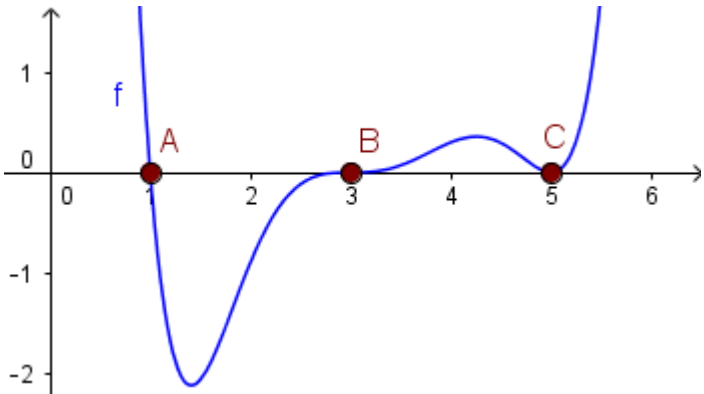


Рис. 8. Нахождение корней функций


A. *Roots*[*fun*, *a*, *b*]

Нахождение корней функции
 А) Пусть *fun* – функция одной переменной *x*. По команде *B* выводится график *fun* и, возможно, один или более корней *fun* на [*a*, *b*] (*a* < *b*).

3.3.4. Прямые

Здесь под условным именем “*Прямые*” описаны инструменты построения таких объектов как: прямая, отрезок, отрезок фиксированной длины, луч, ломанная, произвольный вектор и вектор с началом в заданной точке, коллинеарный другому вектору и равный ему по длине.



Прямая. Инструмент “ Прямая” используется для построения прямой линии по двум позициям или точкам. Выполняется операция последовательными щелчками по этим позициям или точкам. Если щелчки производились по позициям, то в них появляются точки.

В приводимых ниже примерах прямые линии (но не отрезки!) практически всегда будут выводиться тонкими и в точечном стиле: *толщина* – 1.5, *стиль* – “.....”. Чтобы избежать непрерывной корректировки стиля прямых, эти установки можно сделать действующими по умолчанию, то есть зафиксировать их на странице “Установки по умолчанию” панели “Настройки”. При выполнении дополнительной команды меню “Настройки\Сохранить настройки” эти значения будут действовать и в новых сессиях. Возвратить системные установки к принятым по умолчанию значениям можно командой меню “Настройки\Сбросить настройки”.

Замечание. На панели “Объекты” созданные прямые представляются в виде “ $f:ax+by=c$ ”, где f – имя прямой, $ax+by=c$ – уравнение прямой (a , b и c – действительные числа). При $b \neq 0$ и заданной абсциссе x_0 имя f можно использовать для определения ординаты y_0 любой точки $P(x_0, y_0) \in f$, то есть $f(x_0)$ возвращает y_0 . Кроме того, имя f с помощью команд $x[f]$, $y[f]$ и $z[f]$ дает доступ к коэффициентам a , b и c уравнения прямой, то есть $x[f]$, $y[f]$ и $z[f]$ возвращают величины a , b и $-c$. Выполняться команды $f(x_0)$, $x[f]$, $y[f]$ и $z[f]$ могут в командной строке непосредственно или как аргументы других команд.

A. $Line[A, B]$
B. $Line[A, lin]$
C. $Line[A, vec]$


Вывод прямых

A) Командой A выводится прямая, проходящая через заданные точки A и B.

B) Командой B выводится прямая, проходящая через точку A параллельно прямой lin .


C) Командой C выводится прямая, проходящая через точку A по направлению вектора vec .



Отрезок. Инструмент “ Отрезок” применяется для построения отрезка прямой между двумя точками или позициями на полотне. Выполняется операция последовательными щелчками по этим точкам или позициям, которые и соединяются отрезком прямой линии. Если щелчки производились по позициям, то в них появляются точки. Имя отрезка имеет значение, равное текущей длине этого отрезка, то есть если a – имя отрезка, то a всегда равно длине AB . В вычислениях для длины отрезка кроме a альтернативно можно использовать и команду $Distance[A, B]$.


Если решается задача о точках пересечения каких-либо объектов с данным отрезком, то по умолчанию ищутся точки пересечения именно с отрезком.

Однако можно задать режим, при котором такие точки будут находиться как на самом отрезке, так и на его продолжениях. Для этого на панели “*Настройки*” независимым переключателем следует включить режим “*Разрешить внешние пересечения*” (*Allow Outlying Intersection*).

Пример 19. Пусть на отрезке AB необходимо найти точку C , делящую его в отношении λ ($0 \leq \lambda \leq 1000$, $AC/CB = \lambda$). Решить задачу можно так. Инструментом “ *Отрезок*” вывести отрезок AB . В строке ввода ввести сначала число λ ($0 \leq \lambda \leq 1000$), а затем выражение $(A + \lambda \cdot B) / (1 + \lambda)$. После этого на отрезке AB появится требуемая точка C .


Пример 20. Постройте точку пересечения медиан треугольника (центр тяжести, центр тяжести).




Отрезок фиксированной длины. Инструмент “ *Отрезок фиксированной длины*” используется для построения горизонтального отрезка прямой фиксированной длины. Делается это так. Производится щелчок в некоторой позиции или точке полотна. Если это была позиция, то в ней образуется точка. В строке ввода появившейся панели задаем длину a отрезка, и после щелчка по кнопке “*Ok*” появляется требуемый отрезок с точкой на конце. Длину a созданного отрезка изменить нельзя ни перемещениями его концевых точек, ни через панель “*Настройки*”. Протаскиванием по полотну первой точки изменяется позиция отрезка, а протаскиванием второй точки или отрезка реализуется вращение отрезка вокруг его первой точки. Иными словами, вторая точка может перемещаться только по окружности радиуса a с центром в первой точке.



$A. \text{Segment}[A, B]$ $B. \text{Segment}[A, \text{len}]$

Вывод отрезков

А) Командой A выводится отрезок с заданными концевыми точками A и B (см. инструмент “ *Отрезок*”).

В) Командой B выводится отрезок с концевой точкой A длины len (см. инструмент “ *Отрезок фиксированной длины*”).



Луч. Инструмент “ *Луч*” применяется для построения луча последовательными щелчками по двум позициям или точкам на полотне. Если щелчки производились по позициям, то в них создаются точки. Луч строится от первой точки в направлении второй точки. Чтобы не загромождать чертежи, выводить лучи по умолчанию желательно так же, как и прямые линии, то есть тонкими и в точечном стиле: *толщина* – 1.5, *стиль* – “.....” (см. инструмент “ *Прямая*”).

Если решается задача о точках пересечения каких-либо объектов с лучом, то по умолчанию ищутся точки пересечения именно с лучом. Задание режима,

при котором такие точки будут находиться как на самом луче, так и на его продолжении, реализуется так же, как и для отрезка.


A. $Ray[A, B]$ B. $Ray[A, vec]$

Вывод лучей

A) Командой A выводится луч со стартовой точкой A , проходящий через точку B .

B) Командой B выводится луч со стартовой точкой A , проходящий в направлении вектора vec .



Ломанная. Инструмент “ Ломанная” используется для построения незамкнутой ломанной линии щелчками мышью по некоторым позициям или точкам на полотне. Если щелчки производились по позициям, то в них создаются точки. Пары соседних точек соединяются отрезками прямых линий. При повторном щелчке на начальной точке операция завершается, причем последняя и начальная точки отрезком не соединяются. Точек должно быть не меньше 3 (сегментов не меньше 2). Если при построении ломанной держать нажатой клавишу *Alt*, то получаемые углы будут кратны 15° . Имя ломанной возвращает ее длину.


A. $Polyline[\{(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)\}]$ B. $Polyline[(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)]$
--

Вывод ломанной

A) По команде A выводится ломанная линия с начальной точкой (a_1, b_1) и конечной точкой (a_k, b_k) . При совпадении начальной и конечной точек ломанная оказывается замкнутой. Среди точек может присутствовать объект вида $(?, ?)$. Это приводит к разрыву ломанной в соответствующем месте.

B) Команда B по выполнению от A не отличается, но в ней точки задаются не списком, а отдельными аргументами.



Вектор. Инструмент “ Вектор” применяется для построения вектора последовательными щелчками по двум позициям или точкам на полотне. Если щелчки производились по позициям, то в них создаются точки. Вектор строится от первой точки по направлению ко второй точке. Оба конца вектора можно перемещать по полотну.

A. $Vector[B]$ B. $Vector[A, B]$

Вывод векторов

A) Командой A выводится вектор со стартовой точкой в начале координат и концевой точкой B .

B) Командой B выводится вектор со стартовой точкой A и концевой точкой B .




Отложить вектор. Инструмент “ Отложить вектор” используется для построения вектора с началом в конкретной точке, коллинеарного другому

вектору и равного ему по длине. Для выполнения операции щелчками мыши требуется указать на полотне требуемую позицию или точку и некоторый вектор. Если щелчок производился по позиции, то в ней создается точка. В конце вектора также появляется точка. Фактически, по точке A и некоторому вектору $vec = \begin{pmatrix} a \\ b \end{pmatrix}$ формируется вектор $Vector[A, A+(a,b)]$.


3.3.5. Специальные прямые


В этом пункте описаны инструменты для решения ряда задач таких, как: проведение прямой, проходящей через заданную точку перпендикулярно или параллельно другой прямой, лучу, отрезку или вектору; проведение биссектрисы угла, заданного тремя точками или двумя объектами, которые могут быть прямыми, лучами или отрезками; проведение касательных к коническим сечениям, а также к кривым, определяемым функциями одной переменной, неявными функциями или параметрическими функциями. Кроме того, рассмотрены инструменты для построения: поляры к коническим сечениям; прямой, аппроксимирующей экспериментальные данные методом наименьших квадратов; а также локуса, то есть траектории, описываемой некоторой точкой жестко связанной с другой перемещающейся точкой.





Перпендикулярная прямая. Инструмент “ *Перпендикулярная прямая*” используется для проведения прямой, проходящей через заданную точку A перпендикулярно некоторому объекту t или его продолжению. Объект t может быть прямой, лучом, отрезком или вектором. Реализуется операция щелчками мышью по двум заданным объектам – сначала по точке, а затем по второму объекту. Заметим, что точка A может находиться и на самом объекте t , то есть речь может идти о восстановлении в точке $A \in t$ прямой, перпендикулярной к t .

Использованием данного инструмента решается и родственная задача – “Провести перпендикуляр (отрезок, а не прямую) из заданной точки A к объекту t ”. Сделать это можно, например, так:

- проведем перпендикулярную прямую p из A к t ;
- если p и t пересекаются, то перейти к выполнению следующего пункта. В противном случае инструментом “ *Прямая*” провести прямую q так, чтобы объект t оказался расположенным на q ;

- инструментом “ *Пересечение*” создадим точку пересечения B прямых p и t (или p и q);

- инструментом “ *Отрезок*” проведем отрезок AB ;
- удалим перпендикулярную прямую p , а также прямую q , если она проводилась.

Пример 21. Пользуясь инструментом “ Перпендикулярная прямая”, найти точку пересечения перпендикуляров треугольника (ортоцентр), опущенных из его вершин на противоположные стороны или их продолжения. Рассмотреть случаи наличия и отсутствия тупого угла в треугольнике.


A. *PerpendicularLine*[*A*, *obj*]
B. *PerpendicularVector*[*A*, *obj*]

Вывод перпендикулярной прямой или вектора






A) Командой A выводится прямая, проходящая через точку *A*, перпендикулярно прямой, отрезку или вектору *obj*.

B) Командой B выводится вектор, проходящий через точку *A*, перпендикулярно прямой, отрезку или вектору *obj*.




Параллельная прямая. Инструмент “ Параллельная прямая” используется для проведения прямой, проходящей через заданную точку параллельно другой прямой, лучу, отрезку или вектору. Реализуется операция щелчками мышью по двум заданным объектам – сначала по точке, а затем по второму объекту.

Решим, например, такую задачу. Построить параллелограмм *P*, у которого заданы 3 вершины *A*, *B* и *C*. Сделать это можно так:

- инструментом “ Точка” сформируем точки *A*, *B* и *C*;
- инструментом “ Отрезок” проведем отрезки *AB* и *AC*;
- инструментом “ Параллельная прямая” через точку *A* проведем прямую *p*, параллельную *BC*, а через точку *C* – прямую *q*, параллельную *AB*;
- инструментом “ Пересечение” создадим точку пересечения *D* двух проведенных прямых;
- инструментом “ Отрезок” проведем отрезки *AD* и *CD*;
- через контекстное меню спрячем, то есть сделаем невидимыми обе проведенные прямые *p* и *q*. Требуемый параллелограмм построен.

Пример 22. Решить рассмотренную выше задачу каким-либо иным способом.



Срединный перпендикуляр. Инструмент “ Срединный перпендикуляр” используется для проведения перпендикуляра через середину отрезка, вектора или через середину воображаемого отрезка, заданного двумя точками. Реализуется операция щелчком мыши по отрезку, вектору или последовательными щелчками по двум заданным точкам.

Пример 23. Задан треугольник ABC . Построить центр описанной окружности.

A. *PerpendicularBisector*[*seq*]
B. *PerpendicularBisector*[*A*, *B*]


Вывод срединного перпендикуляра


A) Командой *A* выводится прямая, проходящая через середину отрезка *seq*

перпендикулярно *seq*.

B) Командой *B* выводится прямая, проходящая через середину отрезка с концами в точках *A* и *B*, перпендикулярно этому отрезку. Отрезок может быть воображаемым, то есть заданы только точки.



Биссектриса угла. Инструмент “ *Биссектриса угла*” применяется для проведения биссектрисы угла, заданного тремя точками или двумя объектами, которые могут быть прямыми, лучами или отрезками. В первом случае операция выполняется последовательными щелчками мышью по трем точкам, средняя из которых является вершиной угла, а выводится одна биссектриса в виде прямой, делящей указанный угол пополам. Во втором случае операция выполняется последовательными щелчками по двум сторонам угла, а выводятся две биссектрисы для каждой пары вертикальных углов, формируемых этими сторонами или их продолжениями.

Пример 24. Используя инструмент “ *Биссектриса угла*” найти точку пересечения биссектрис треугольника, заданного тремя вершинами. Иными словами, требуется найти центр вписанной в треугольник окружности (*инцентр*).

Пример 25. Постройте для треугольника *ортоцентр* (точку пересечения перпендикуляров, опущенных из вершин на противоположные стороны), *центроид* (точку пересечения медиан) и центр описанной окружности. Проведите через полученные точки прямую. Убедитесь (визуально), что все три построенные точки лежат на одной прямой. Называется она прямой Эйлера.

A. *AngleBisector*[*obj1*, *obj2*]
B. *AngleBisector*[*A*, *B*, *C*]

Вывод биссектрисы

A) Командой *A* выводятся биссектрисы углов со сторонами, заданными объектами




obj1 и *obj2*, которые могут быть прямыми, лучами или отрезками.

B) Командой *B* выводится биссектриса $\angle ABC$ (проходит через точку *B*).








Касательная. Инструмент “ *Касательная*” используется для построения касательных к кривым. Для выполнения операции необходимо щелкнуть

мышью сначала по некоторой точке, отрезку, прямой или лучу, а затем по объекту, к которому проводятся касательные. Рассмотрим возможные варианты задания кривых и особенности построения касательных:

- задана точка P и коническая кривая. Касательные к конике проводятся из точки P . На рис. 9 слева показана гипербола, построенная инструментом  “Гипербола” по двум фокусам A и B и точке C . К гиперболе из точки D инструментом  “Касательная” штрихпунктирными линиями проведены две касательные f и g . Их точки касания E и F с гиперболой сформированы инструментом  “Пересечение”. Перемещая D , можно наблюдать изменение положения точек касания;


- задан объект obj (прямая, отрезок или луч) и коническая кривая. Касательные к конике проводятся параллельно obj ;

- задана точка P и кривая, введенная через строку ввода функцией одной переменной $h(x)$. Касательная проводится к кривой $h(x)$, но не в точке P , а в точке $(x[P], h(x[P]))$. На рис. 9 справа показана парабола $h(x)=(x-10)^2-1$, введенная через строку ввода обычным образом – $(x-10)^2-1$. Инструментом  “Касательная” к параболе h в точке $(x[G], x[G]-10)^2-1$ проведена штрихпунктирная касательная. Заметим, что если бы парабола была выведена инструментом  “Парабола”, то касательная к ней проходила бы через точку G ;

- задана точка P и кривая, введенная через строку ввода неявной функцией $F(x, y)$. Касательные проводятся из точки P к кривой. На рис. 10 слева показана кривая $x^3 - 4xy + y^3 = 0$ и две штрихпунктирные касательные к ней, проведенные из точки A инструментом  “Касательная”. Уравнение кривой введено через строку ввода обычным образом – $x^3-4*x*y+y^3=0$. Точки касания B и C созданы инструментом  “Пересечение”. Что касается записи уравнения кривой на полотне, то она сформирована инструментом  “Текст” на одноименной панели при включенном независимом переключателе “*Latex-формула*”, и сделано это вводом выражения “ $x^3-4x\cdot y+y^3=0$ ” (без кавычек);

- задана точка P , лежащая на параметрической кривой $x=\varphi(t)$, $y=\psi(t)$ ($a \leq t \leq b$), которая введена через строку ввода функцией $\text{Curve}[\varphi(t), \psi(t), t, a, b]$. Касательная проводится через точку P . Заметим, что на такой кривой точка, соответствующая параметру $t=t_0$ – это $P=(\varphi(t_0), \psi(t_0))$. На рис. 10 справа по-

казана кривая $\begin{cases} 3+t+\sin(t) \\ \cos(t) \end{cases} \quad (0 \leq t \leq 6)$ и штрихпунктирная касательная к ней в

точке D , соответствующая значению $t=2$. Кривая сформирована через строку ввода с помощью функции $\text{Curve}[3+t+\sin(t), \cos(t), t, 0, 6]$. Точка $D=(3+2+\sin(2), \cos(2))$ также создана через строку ввода. Запись уравнения кривой на полотне сформирована с помощью инструмента  “Текст” на одноименной

панели при включенном независимом переключателе “*Latex-формула*”. При-
чем ее ввод действительно осуществлен на языке *LaTeX* в виде:

$$\begin{cases} 3+t+\sin(t) \\ \cos(t) \end{cases} \quad (0 \leq t \leq 5)$$
(1)

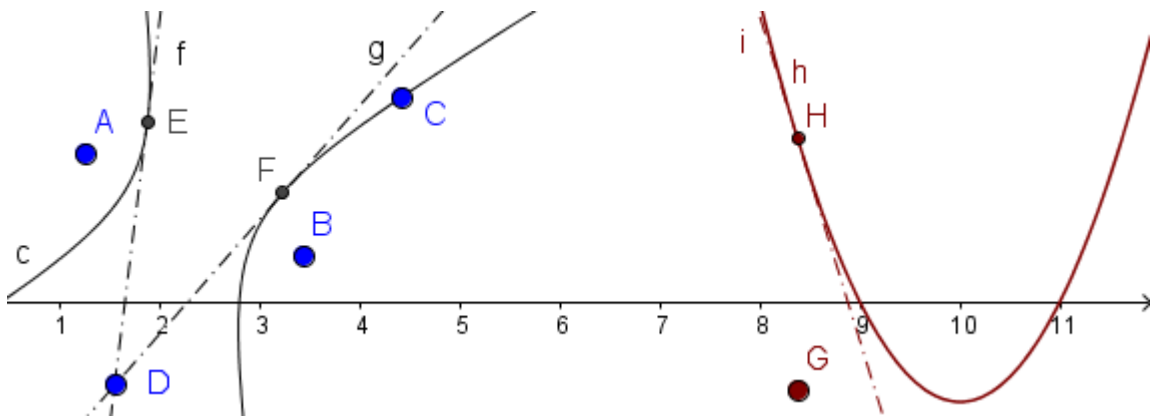


Рис. 9. Две касательные *f* и *g* к гиперболе из точки *D*;
касательная *i* к параболе $(x-10)^2-1$ в точке $(x[G], (x[G]-10)^2-1)$

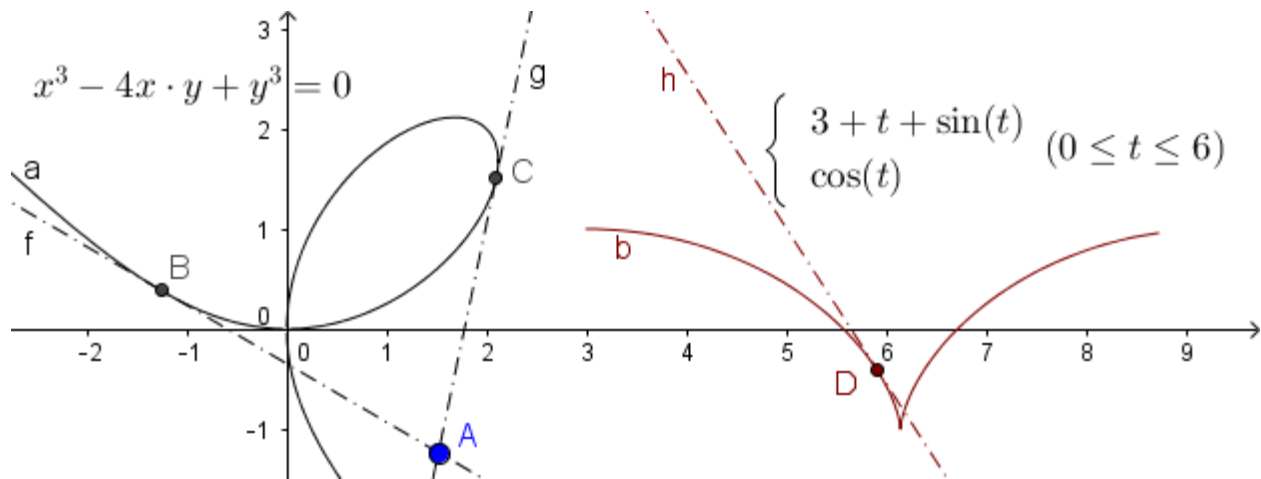



Рис. 10. Касательные *f* и *g* к неявной кривой $x^3 - 4xy + y^3 = 0$ из точки *A*;
касательная *h* к параметрической кривой *b* в точке *D*

Замечание. Качественное оформление надписей на чертежах, содержащих математические формулы, проводится на специальном скриптовом языке программирования научной издательской системы *LaTeX*. Надписи с простыми выражениями обычно удается создавать без знания синтаксиса и семантики этого языка. Дело в том, что, используя инструмент “ABC Текст” при включенном независимом переключателе “*Latex-формула*”, мы делаем доступными три раскрывающихся списка дополнительных инструментов, с помощью которых можно формировать различные элементы *LaTeX*-выражений

на интуитивном уровне. Это нечто подобное системе мини-*LyX*¹. Кроме того, при формировании надписей образцы *LaTeX*-текстов можно использовать в качестве шаблонов. Скажем, исходя из (1), нетрудно понять, что надпись функции “ $x=\varphi(t), y=\psi(t) \ (a\leq t\leq b)$ ” можно получить по *LaTeX*-выражению: $\begin{cases} \varphi(t) \\ \psi(t) \end{cases} \ (a\leq t\leq b)$, где $\varphi(t)$ – это $\varphi(t)$, а $\psi(t)$ – это $\psi(t)$.

Пример 26. Используя инструмент “*Касательная*”:

- провести касательную в точке $x=x_0$ к функции $g(x)=e^{-x\sin(x)/5}$, задав на полотне точку $P=(x_0, y_0)$ и в строке ввода – выражение $\exp(-x*\sin(x)/5)$. Проследить за изменениями касательной при перемещении точки P по полотну;
- провести касательную (или касательные) из точки P к неявно заданной функции $x^5 - y^4 + 7x^2y = 5 \ (-3\leq x\leq 5, -1\leq y\leq 3)$. На рис. 11 показан график этой функции, состоящий из двух частей, и возможная позиция точки A , при которой имеется 4 касательных, проведенных штрихпунктирными линиями. Три точки касания B, C и D сформированы инструментом “ *Пересечение*”, четвертая точка находится в 4 квадранте вне приведенного изображения.

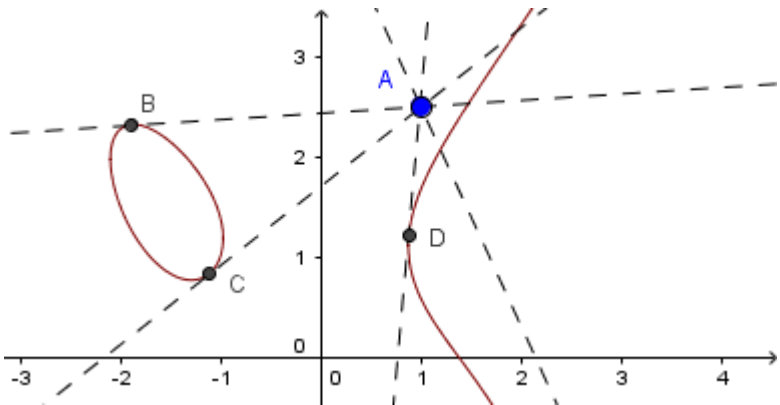


Рис. 11. Касательные из точки A к кривой $x^5 - y^4 + 7x^2y = 5$

Перемещая точку A мышью, выясните, имеются ли позиции A с 0, 1, 2 и 3 касательными? Сколько касательных можно провести из точки, находящейся внутри замкнутой ветви кривой?

- для параметрической кривой: $x=\cos(3t), y=t\cdot\cos(t) \ (0\leq t\leq 3\pi/2)$ построить касательную для точки $t=1/2$ (см. рис. 12). Выведите в 4-ом квадранте около кривой надпись для функции, желательно в том виде, который показан в надписи к рис. 12.

¹ *LyX* – надстройка, добавляющая к *LaTeX* визуальную оболочку, благодаря которой верстка книг, статей, презентаций, писем и других документов с техническим уклоном стала почти столь же простым делом, как и набор не содержащих формул текстов в каком-либо редакторе.

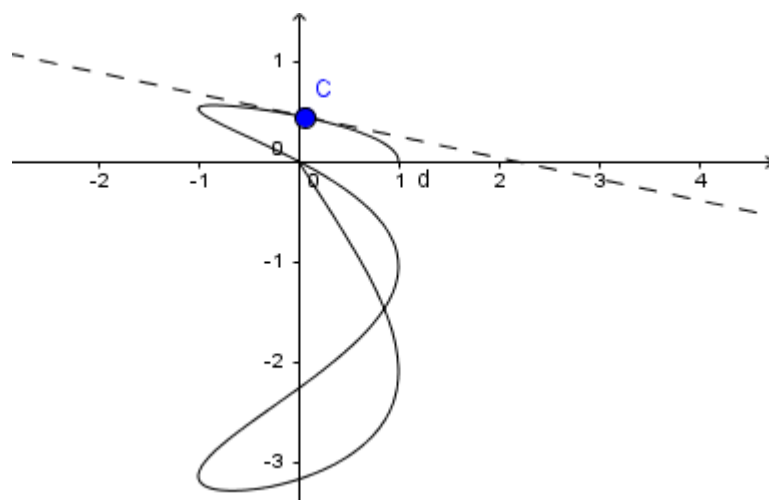


Рис. 12. Касательная к кривой $\begin{cases} \cos(3t) \\ t \cdot \cos(t) \end{cases} (0 \leq t \leq 3\pi / 2)$ при $t=1/2$

A. <i>Tangent</i> [<i>A</i> , <i>conic</i>]	C. <i>Tangent</i> [<i>A</i> , <i>fun</i>]	E. <i>Tangent</i> [<i>A</i> , <i>spline</i>]
B. <i>Tangent</i> [<i>obj</i> , <i>conic</i>]	D. <i>Tangent</i> [<i>circ1</i> , <i>circ2</i>]	

Построение касательных к кривым

Если кривые, к которым проводятся касательные, отдельно не выведены, то по командам *Tangent* они не появятся.

A) Командой *A* проводятся касательные из точки *A* к конике *conic*.

B) Командой *B* проводятся касательные к конике *conic* параллельно объекту *obj*, который может быть прямой, отрезком или лучом.

C) Командой *C* проводятся касательные:

- в точке $(x[A], fun(x[A]))$ кривой, заданной функцией *fun* одной переменной *x*;
- из некоторой точки *A* к кривой, определяемой неявно заданной функцией *fun*;
- в некоторой точке *A*, лежащей на кривой, определяемой параметрически заданной функцией *fun*.

D) Командой *D* проводятся касательные к двум окружностям *circ1* и *circ2*.


E) Командой *E* проводятся касательные к сплайну в точке *A*, заданной на сплайне.



Поляра. Пусть *A* – некоторая точка и *L* – невырожденная коническая кривая. Пусть *B* – такая точка, что прямая *AB* пересекает *L* в точках *C* и *D* и четверка точек *A*, *B*, *C* и *D* является гармонической, то есть $\frac{AC}{CB} : \frac{AD}{DB} = -1$, где *AC*,






CB, *AD* и *DB* – длины отрезков с учетом знака, если положительным на *AB* считать направление от *A* к *B*. Точка *A* называется полюсом поляры, точки *A* и *B* – базисной или основной парой, точки *C* и *D* – делящей парой. Полярной точки *A* относительно кривой *L* называют множество всех точек *B* таких, что

четверки точек A, B, C и D являются гармоническими. Поляра является прямой линией.

Инструментом “ Поляра” строится поляра. Выполняется операция последовательными щелчками мыши сначала в точке A (полюс поляры), а затем по кривой L .

Пример 27. Построим поляру для эллипса и проверим выполнение гармонического отношения для какой-либо ее точки (см. рис. 13):

Решение. Будем решать поставленную задачу так:

- инструментом “ Эллипс” сформируем на полотне эллипс по двум фокусам и точке на нем;
- через контекстное меню изменим имена фокусов и точки эллипса соответственно на E, F и G . Это не обязательно и сделано лишь для того, чтобы высвободить имена A, B и C ;
- инструментом “ Точка” сформируем на полотне точку A – полюс поляры;
- инструментом “ Поляра” создадим поляру, щелкнув мышью сначала по точке A , а затем по эллипсу;
- инструментом “ Точка на объекте” зафиксируем какую-либо точку B на поляре;
- проведем прямую через точки A и B и в позициях пересечения ее с эллипсом инструментом “ Пересечение” сформируем точки C и D ;
- подсчитаем величину $-(AC/CB)/(AD/DB)$, записав ее в строке ввода. На панели объектов получим вывод в виде $d=-1$, то есть четверка точек A, B, C и D действительно является гармонической.

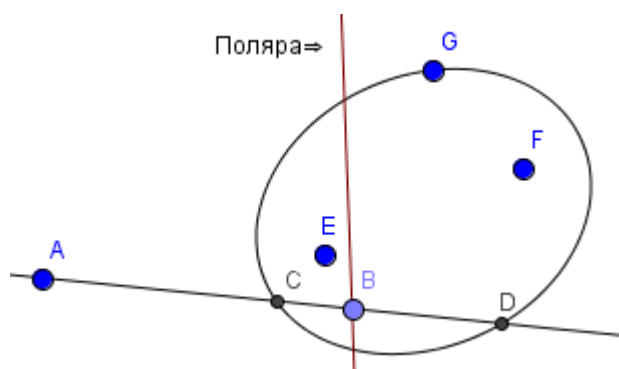


Рис. 13. Поляра для эллипса


Смещая мышью по поляре точку B , можно видеть, что при любых B внутри эллипса величина двойного отношения $d=-1$ не изменяется, а при B на границе эллипса или вне его d не определено.


Пример 28. Постройте поляру для параболы и проверьте выполнение гармонического отношения для нее.


Пример 29. Постройте полярю для гиперболы и проверьте выполнение гармонического отношения для нее.

$A. Polar[A, conic]$	<p><i>Вывод поляр</i></p> <p>А) Командой A выводится поляр с полюсом в точке A. Поляра строится по отношению к конике $conic$.</p>
----------------------	---



Аппроксимация. Инструмент “ Аппроксимация” используется для проведения линии линейной регрессии $y(x)=a \cdot x+b$ для некоторой совокупности точек M , то есть для построения прямой, наилучшим образом приближающей точки M в смысле метода наименьших квадратов. Выполнить операцию можно одним из следующих способов:

1) При активной кнопке “” протяжкой мыши по полотну выделить бледно-голубым цветом прямоугольную область. При отпускании кнопки мыши для точек M , попавших в выделенную область или на ее границу строится линия регрессии;

2) При активной кнопке “” щелкнуть левой кнопкой мыши по какому-либо списку на панели “Объекты”. Для точек этого списка будет построена линия регрессии. Создать список можно через строку ввода, например, так: $\{A, B, F\}$ или $\{(1,1), (2,3), (3,2), (4,5)\}$ и т. п.

3) Через строку ввода выполнить команду $FitLine[lis]$. Для списка точек lis будет построена линия регрессии.


Пример 30. Постройте линию регрессии для точек: $(-1.2, 1.6), (-0.6, 2), (1.3, 2), (1.1, 2.7), (0.2, 4), (2, 5), (3, 4), (4.6, 5), (-2, 1), (-1.6, 2.5), (-0.7, 3.3)$.

$A. FitLine[lis]$	<p><i>Линейная регрессия</i></p> <p>А) Командой A по списку точек lis формируется и выводится линия линейной регрессии.</p>
-------------------	---



Локус. Пусть имеется точка A , которая может перемещаться по некоторой кривой L (отрезку, прямой линии, окружности, границе многоугольника и т. д.). Пусть позиция точки B жестко связана с позицией точки A . Например: $B=(y[A], x[A]), B=(y[A], -x[A]), B=(\sin(x[A]), y[A]), B=(g'(x[A]), f(x[A])) \dots, \dots$, где f, g – встроенные или пользовательские функции; $x[A], y[A]$ – встроенные функции, возвращающие соответственно абсциссу и ординату точки A . Рассматриваемый инструмент используется для построения локуса точки B , то есть траектории, которая описывается точкой B при перемещении A по кривой L^2 . Для построения локуса точки B достаточно сначала активировать инстру-

² В более широком смысле локус – это множество точек, удовлетворяющих определенным условиям.

мент “ Локус”, затем последовательно щелкнуть левой кнопкой мыши сначала по точке A , а затем по зависимой от нее точке B . Точку B проще всего задавать через строку ввода. Заметим, что если взять $B=(y[A], x[A])$, то ее локусом будет обратная функция по отношению к функции, по которой движется A .

Пример 31. Пусть точка A движется по параболе $y(x)=x^2$, а точка B связана с A зависимостью $B=(y[A], \sin(4x[A]))$. Тогда локусом точки B является штриховая линия, показанная на рис. 14. Иными словами, при протаскивании мышью точки A по параболе можно наблюдать перемещение точки B по указанной штриховой локусной линии.

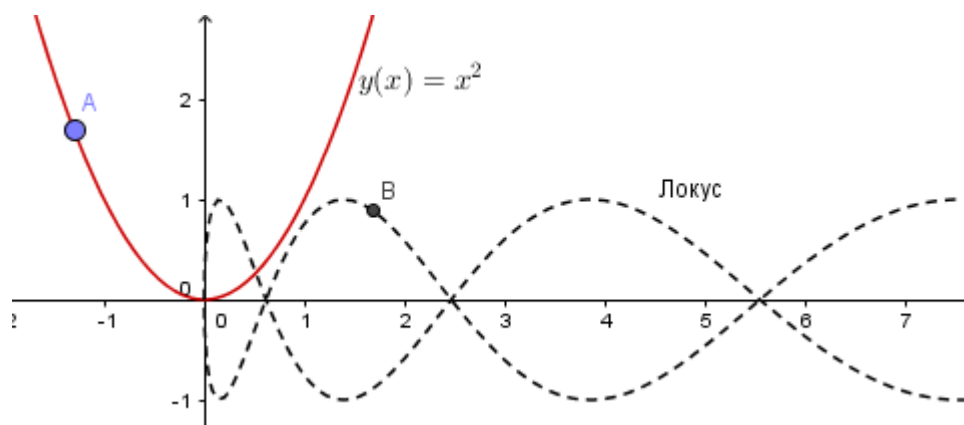




Рис. 14. Локус точки B (штриховая линия), связанной с точкой A , движущейся по кривой $y=x^2$

Построить локус, представленный на рис. 14, можно, например, так:

- в строке ввода введем выражение x^2 и инструментом “ Точка на объекте” укажем на появившейся параболе некоторую точку A ;
- в строке ввода введем точку B : $(y[A], \sin(4*x[A]))$;
- активируем инструмент “ Локус”;
- щелкнем мышью сначала по точке A , а затем по точке B ;
- при необходимости у появившегося локуса, показанного штриховой линией, через панель “Настройки” можно изменить те или иные свойства.

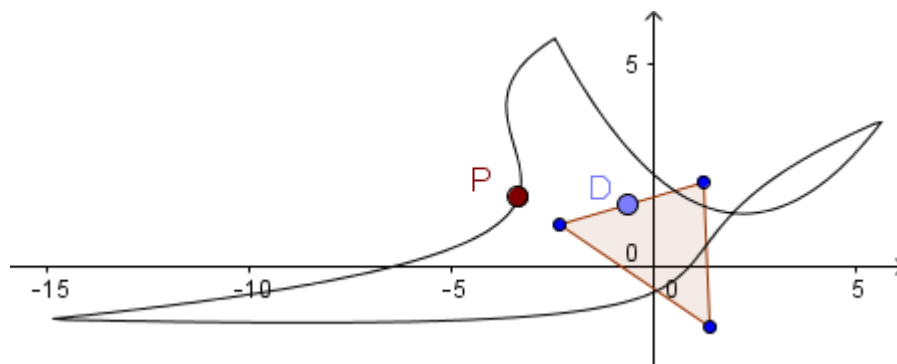





Рис. 15. Локус точки P , связанной с точкой D , движущейся по границе треугольника

Пример 32. Пусть точка D движется по границе треугольника, а точка P связана с ней зависимостью $P=(x[D]^3-2y[D], x[D]+y[D]^2)$. Тогда локусом точки P является линия, показанная на рис. 15. При протаскивании мышью точки D по границе треугольника можно наблюдать перемещение точки P по локусной линии.

Построить локус, представленный на рис. 15, можно, например, так:

- инструментом “ Многоугольник” создадим многоугольник и спрячем обозначения всех его элементов;
- инструментом “ Точка на объекте” прикрепим к стороне многоугольника точку D ;
- в строке ввода введем точку $P=(x(D)^3-2*y(D), x(D)+y(D)^2)$;
- активируем инструмент “ Локус”;
- щелчком мышью сначала по точке D , а затем по точке P . Появится локус точки D , показанный на рис. 15, то есть траектория P , когда D бежит по границе треугольника.

Пример 33. Используя инструмент “Локус” постройте обратные функции для функций x^2 , x^3 , $\sin(x)$, $\cos(x)$.

Пример 34. Сформируйте на полотне свободную точку A и связанную с ней точку $B=(y[A]/x[A], x[A]/y[A])$. Для обеих точек через контекстное меню включите режим “Оставлять след”. Понаблюдайте изменения траектории точки B при перемещении точки A мышью.

<p>A. $Locus[D, A]$ B. $Locus[D, t]$ C. $Locus[f(x,y), P]$</p>

Построение локуса командами

A) Здесь A – некоторая точка, связанная с определенным “линейным” объектом L (отрезком, прямой линией, окружностью, границей многоугольника и т. д.),

D – зависимая от A или, по-другому, локусная точка. По команде A выводится локус точки D , то есть траектория D , соответствующая перемещению A по L .

B) Пусть задан ползунок t с границами изменения от a до b ($a < b$) и $f(t)$, $g(t)$ – выражения от t . По команде B выводится траектория локусной точки $D=(f(t), g(t))$, при изменении t от a до b .

C) По команде C численно находится и выводится частное решение дифференциального уравнения первого порядка $\frac{dy}{dx} = f(x, y)$, проходящее через точку P .

Пример 35. Введем через строку ввода переменную $t=5$ и включим на панели “Объекты” соответствующую t радиокнопку. На панели “Полотно” будет выведен ползунок с возможными значениями t от -5 до 5 и текущим значением 5 . Через строку ввода сформируем локусную точку $D=(2\sin(3t), \cos(5t))$

и затем введем команду $Locus[D, t]$. В результате этих действий появится локусная линия, показанная на рис. 16. При изменении значения ползунка t можно наблюдать перемещение локусной точки D по сформированной локусной линии.

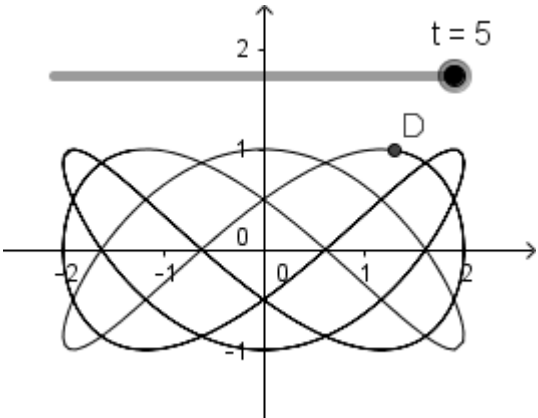


Рис. 16. Локус точки D по значениям ползунка t

Пример 36. Пусть для дифференциального уравнения $\frac{dy}{dx} = \frac{1-3x^2}{y}$ ищется решение, которое при $x=1$ должно равняться 3, то есть проходить через точку $(1, 3)$. Найти приближенное решение можно с помощью команды $Locus[(1-3*x^2), (1, 3)]$ (см. рис. 17). В данном случае исходное уравнение является уравнением с разделяющимися переменными и для него легко найти точное общее решение $y^2=2x-2x^3+C$, а значит и точное частное решение – $y^2=2x-2x^3+9$ (при $x=1, y=3$). Это решение можно вывести как неявную функцию командой $ImplicitCurve[y^2+2x^3-2x-9]$. Сделав это на том же самом графике, получим полное (визуальное) совпадение кривых. Что касается двух надписей на графике, то они получены с помощью инструмента “ABC Текст” при включенной кнопке “*LaTeX*-формула” вводом следующих выражений:

$$\begin{aligned} &\Leftarrow Locus \left[\frac{1-3x^2}{y}, (1,3) \right] \quad , \\ &2x^3-2x+y^2=9 \Rightarrow \quad . \end{aligned}$$

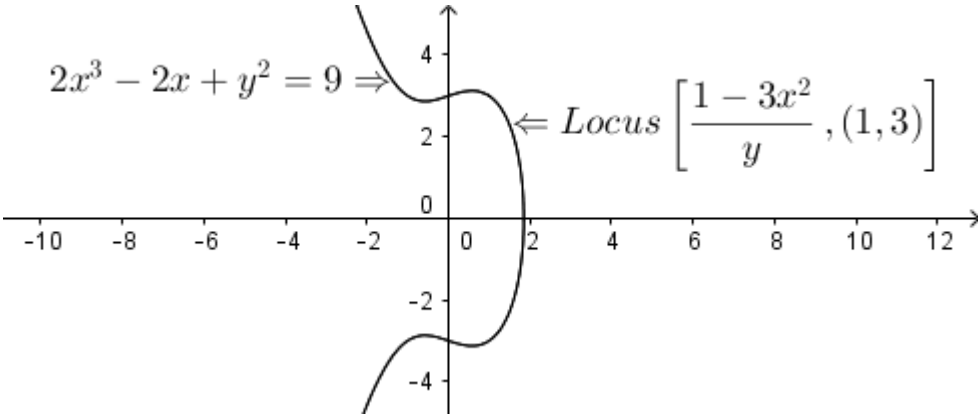


Рис. 17. Численное решение дифференциальных уравнений с помощью локуса

A. <i>Perimeter</i> [<i>locus</i>]
B. <i>Length</i> [<i>locus</i>]
C. <i>First</i> [<i>locus</i> , <i>n</i>]

Длина, количество точек и точки локуса

А) По команде *A* вычисляется длина кривой локуса *locus*, если она конечна.


В) По команде *B* подсчитывается количество точек, в которых вычислялся локус *locus*.

С) По команде *C* возвращается список первых *n* вычисленных точек локуса *locus*.

3.3.6. Многоугольники

Здесь рассмотрены инструменты для построения плоских многоугольников общего положения, а также правильных, жестких и векторных многоугольников. Изменение формы многоугольника общего положения возможно протягиванием по полотну любой из его вершин. У правильного многоугольника форму изменить можно протягиванием любой из двух его начальных вершин. Местоположение жесткого многоугольника на полотне может меняться, но его форма при этом сохраняется. Изменение формы векторного многоугольника возможно протягиванием по полотну любой из его вершин, кроме начальной.



Многоугольник. Инструмент “ **Многоугольник**” используется для построения многоугольников общего положения. Операция выполняется последовательными щелчками или по существующим точкам, или по позициям полотна. Если в позиции точки не было, то она там появляется. Завершается операция при щелчке на начальной точке. Все последовательно проходимые точки соединяются отрезками линий, причем даже там, где отрезки уже были. Фактически при создании *n*-угольника, кроме него самого формируется *n* точек, если их еще не было, и *n* отрезков. Если при построении многоугольника *M* держать нажатой клавишу *Alt*, то внутренние углы *M* будут кратны 15° . Связанные области внутри многоугольника по умолчанию раскрашиваются бледно-розовым цветом, хотя в дальнейшем и цвет, и иные его свойства через панель “*Настройки*” могут быть изменены. В частности, вместо раскраски области можно использовать различные заливки цветными декоративными шаблонами. Протягиванием мыши многоугольник можно перемещать по полотну или копировать через буфер памяти как единое целое. Изменение формы многоугольника достигается протягиванием любой из его вершин. Если сформированный многоугольник *M* без самопересечений, то имя *M* имеет значение, равное его текущей площади. Пересечение многоугольников показывается с наложением действующих раскрасок или заливок.

На рис. 18 приведены примеры многоугольников: треугольник, четырехугольник и шестиугольник. Последний из них имеет пересечения сторон и,

кроме того, его вершина I лежит на стороне GF четырехугольника. Это означает, что I является связанной точкой, то есть перемещать I можно только по стороне GF . Остальные вершины шестиугольника и других объектов являются свободными точками и могут перемещаться по полотну в любых направлениях и на какое угодно расстояние. При этом, при перемещении точек G и F , точка I также будет перемещаться. При необходимости точку I можно открепить, то есть сделать свободной (инструмент “Прикрепить/Снять точку”).

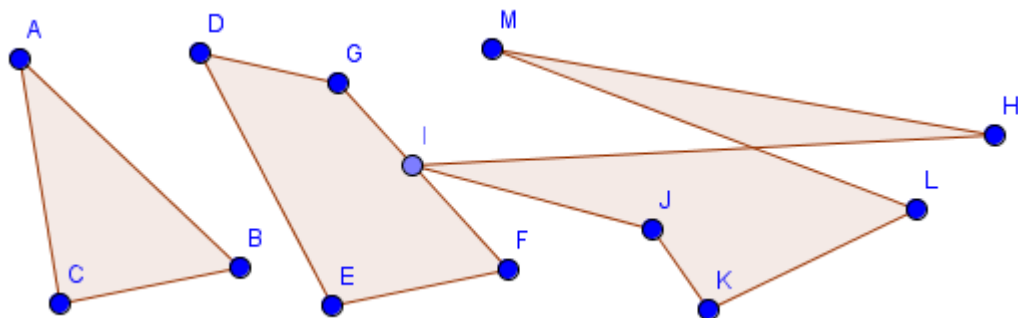



Рис. 18. Многоугольники общего положения

Пример 37. Создайте чертеж с многоугольниками, аналогичный показанному на рис. 18, со связанной вершиной I . Превратите точку I в свободную, и перемещая I , преобразуйте шестиугольник так, чтобы стороны его стали не пересекающимися.



Правильный многоугольник. Инструмент “ Правильный многоугольник” применяется для построения правильных многоугольников. Операция выполняется следующим образом. Двумя последовательными щелчками по существующим точкам или позициям формируются две смежные вершины многоугольника и выводится панель с запросом об общем количестве вершин. После указания в строке ввода целого числа n ($n \geq 3$) строится правильный n -угольник M . Начальные вершины M могут быть как свободными, так и связанными. В первом случае ограничений для их перемещения нет, а во втором – они могут перемещаться внутри объекта, с которым связаны. Перемещение любой из двух начальных вершин меняет длину стороны M . Остальные вершины M связаны с начальными вершинами, меняют позиции вместе с ними и их нельзя протаскивать мышью. Значение имени сформированного многоугольника равно его текущей площади. Изменять свойства правильных многоугольников можно так же, как и многоугольников общего положения, то есть через панель “Настройки”.

Пример 38. На рис. 19 слева демонстрируется раскраска, а справа – заливка пересечения шестиугольника общего положения и правильного семиугольника. Создайте аналог чертежа рис. 19 и поэкспериментируйте с различными раскрасками и заливками многоугольников.

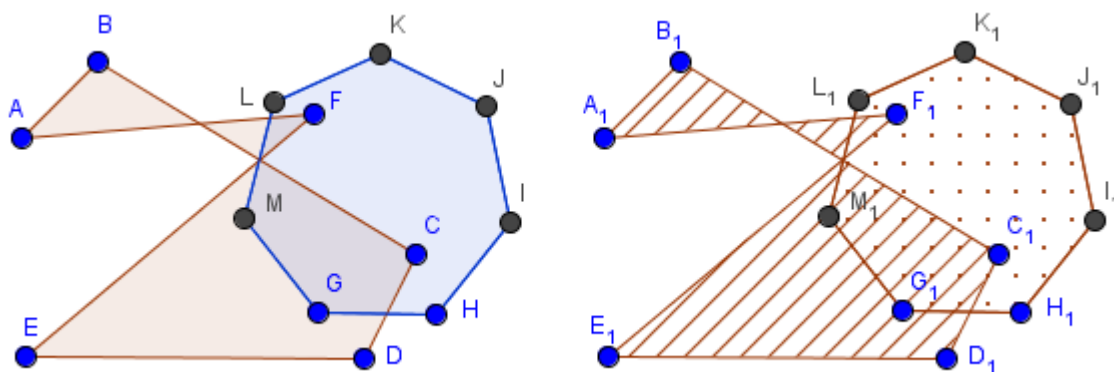


Рис. 19. Раскраска и заливка пересечений многоугольников


A. $Polygon[A1, A2, \dots, An]$
 B. $Polygon[\{A1, A2, \dots, An\}]$
 C. $Polygon[A, B, n]$

Вывод многоугольников


A-B) По командам A и B выводится многоугольник с вершинами в точках $A1, A2, \dots, An$. Сами точки по A не выводятся.

C) По команде C выводится правильный n -угольник со стороной, равной отрезку AB .



Жесткий многоугольник. Инструмент “ Жесткий многоугольник” используется для построения жесткого многоугольника. Операция выполняется точно так же, как и при построении обычного многоугольника, но в данном случае в результате получается многоугольник, у которого нельзя изменить форму, но можно изменять местоположение на полотне. При завершении операции все вершины полученного многоугольника M , кроме первых двух, становятся невидимыми. Протягиванием начальной вершины (или любой внутренней позиции M) осуществляется перемещение M по полотну. Протягиванием второй вершины реализуется вращение M вокруг начальной точки. Невидимые вершины M можно визуализировать, но ни перемещений, ни вращений с их помощью совершать нельзя. Если сформированный многоугольник M без самопересечений, то имя M имеет значение, равное его текущей площади. Если при построении M держать нажатой клавишу Alt , то внутренние углы M будут кратны 15° .




Векторный многоугольник. Инструмент “ Векторный многоугольник” используется для построения векторного многоугольника. Операция выполняется точно так же, как и при построении обычного многоугольника, но в данном случае в результате получается многоугольник M , который протягиванием начальной точки можно перемещать по полотну, не изменяя формы. Все остальные вершины M ведут себя обычным образом. Если сформированный многоугольник M без самопересечений, то имя M имеет значение, равное его текущей площади. Если при построении M держать нажатой клавишу Alt , то внутренние углы M будут кратны 15° .

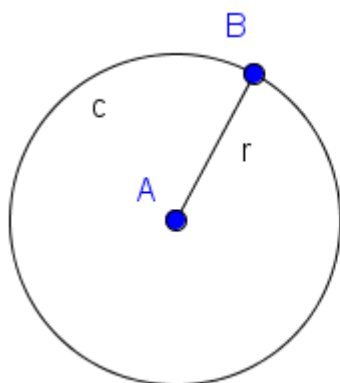
3.3.7. Окружности

В этом пункте описаны инструменты для решения ряда задач, связанных с различными способами построения окружностей, полуокружностей, дуг и секторов.



Окружность по центру и точке. Инструмент “ *Окружность по центру и точке*” используется для построения окружности по ее центру и точке на окружности. Выполняется операция последовательными щелчками по двум точкам или позициям полотна. В последнем случае на месте позиций возникают точки. Протаскиванием точек можно изменять местоположение окружности и ее радиус.




Пример 39. Решим следующую задачу. Построить окружность по центру A и точке B . Создать надписи для вывода текущих значений радиуса $r=AB$, площади круга $S=\pi r^2$ и длины окружности $L=2\pi r$, которые должны автоматически обновляться при перемещениях точек A и B по полотну (см. рис. 20).



Площадь круга $S = \pi r^2 \rightarrow 8.61$,
Длина окружности $L = 2\pi r \rightarrow 10.4$.

Рис. 20. Окружность и надписи к ней с постоянной и вычисляемой частями

Решение. Следующие действия решают поставленную задачу:





- инструментом “ *Окружность по центру и точке*” выведем окружность;
- инструментом “ *Отрезок*” соединим точки A и B отрезком и через контекстное меню изменим его обозначение на r ;
- инструментом “ *Текст*” создадим надпись из двух строк с постоянными и переменными частями. Для этого на открывшейся панели “*Текст*” в поле *Edit* в *LaTeX*-режиме введем выражение вида:

$$\begin{aligned} \text{Площадь круга } \$S=\pi r^2 \rightarrow \$ & \boxed{\pi * AB^2} , \\ \text{Длина окружности } \$L=2\pi r \rightarrow \$ & \boxed{2\pi * AB} . \end{aligned}$$

В окне “Предпросмотр” можно наблюдать за формируемой надписью. После закрытия панели “Текст” надпись автоматически переносится на полотно и ее дальнейшее редактирование возможно через панель “Настройки”, где можно, например, изменить шрифт, его начертание и размер, а также цвет текста.

Пример 40. Пусть c , d и e – три непересекающиеся окружности разных радиусов. Построить модель для экспериментальной проверки утверждения: “Точки пересечения общих внешних касательных к парам окружностей c и d , d и e , e и c лежат на одной прямой”.

Решение. Требуемая модель может быть построена так:

- инструментом “ Окружность” выведем три непересекающиеся окружности c , d и e ;
- через строку ввода функцией $Tangent[c, d]$ проведем внешние и внутренние касательные к окружностям c и d . Через контекстное меню внутренние касательные сделаем невидимыми. Инструментом “ Пересечение” отметим точку I – пересечения внешних касательных. Аналогичные действия проведем и для пар окружностей d , e и e и c ;
- инструментом “ Прямая” проведем прямую через точки I и G ;
- инструментом “ Текст” создадим надпись с динамической (вычисляемой) частью для экспериментальной проверки того, что точка H лежит на прямой IG . Сделать это можно с помощью выражения

Точки I, G, H лежат на одной прямой \rightarrow $AreCollinear[I, G, H]$.

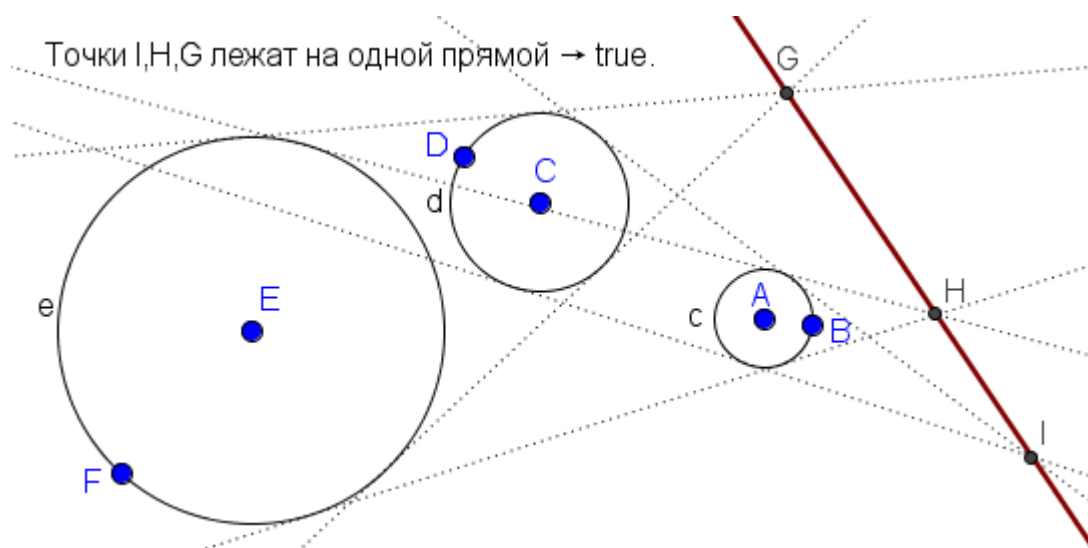



Рис. 21. I, G, H – точки пересечения внешних касательных к парам окружностей






Перемещая окружности и изменяя их радиусы в правой вычисляемой части надписи всегда будем иметь значение *true*. Этим экспериментально и подтверждается проверяемое утверждение.



Окружность по центру и радиусу. Инструмент “ *Окружность по центру и радиусу*” используется для построения окружности по центру и радиусу. Выполняется операции так. Производится щелчок по точке или позиции на полотне – будущему центру окружности. Если щелчок был по позиции, то в ней формируется точка. В любом случае появляется панель для ввода величины радиуса. В ее строке ввода требуется задать выражение с неотрицательным значением без единиц измерения. По умолчанию радиус измеряется в сантиметрах. При нажатии на кнопку *Ok* панель закрывается и выводится требуемая окружность.

Пример 41. Пусть требуется вписать в окружность правильный n -угольник ($n=3,4, \dots, 20$).

Решение. Вписать в окружность правильный многоугольник можно, например, так:

- инструментом “ *Окружность по центру и точке*” выведем исходную окружность (A – центр, B – точка на окружности);
- инструментом “ *Ползунок*” выведем одноименный управляющий элемент, включив на открывшейся панели радиокнопку “*Целое число*” и задав для переменной n диапазон значений от 3 до 20 с шагом 1;
- инструментом “ *Окружность по центру и радиусу*” выведем вспомогательную окружность с центром в точке B радиуса $2 \cdot AB \cdot \sin(\pi/n)$ и через панель “*Настройки*” изменим стиль ее вывода на штриховой;
- инструментом “ *Пересечение*” зафиксируем одну из точек C пересечения двух окружностей;
- инструментом “ *Правильный многоугольник*” построим правильный n -угольник со стороной BC , указав точки B , C и переменную n .

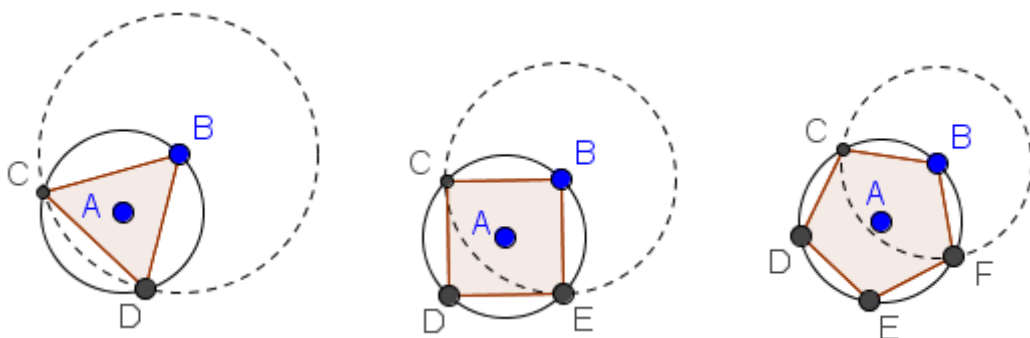








Рис. 22. Правильный n -угольник, вписанный в окружность

Перемещая бегунок слайдера, можно наблюдать изменения чертежа. На рис. 22 чертеж показан при $n=3, 4$ и 5 , причем в нем все имена, кроме точки A и вершин вписанного многоугольника, сделаны невидимыми. Изменить размеры чертежа можно перемещением точки B .

Пример 42. Пусть требуется описать около окружности правильный n -угольник ($n=3,4, \dots, 20$).

Решение. Описать около окружности правильный многоугольник можно, например, так:

- инструментом “ Окружность по центру и точке” выведем исходную окружность (A – центр, B – точка на окружности)
- инструментом “ Ползунок” выведем одноименный управляющий элемент, включив на открывшейся панели радиокнопку “Целое число” и задав для переменной n диапазон значений от 3 до 20 с шагом 1;
- инструментом “ Окружность по центру и радиусу” нарисуем вспомогательную окружность с центром в точке A радиуса $AB/\cos(\pi/n)$ и через панель “Настройки” изменим стиль ее вывода на штриховой;
- инструментом “ Касательная” проведем касательную в точке B к исходной окружности;
- инструментом “ Пересечение” зафиксируем точки пересечения C и D полученной прямой со вспомогательной окружностью;
- инструментом “ Правильный многоугольник” построим правильный n -угольник, указав точки D и C и введя n .

Перемещая бегунок, можно наблюдать изменения чертежа. На рис. 23 чертеж показан при $n=3, 4$ и 5 . Изменять размеры изображения можно перемещением точек A и B .

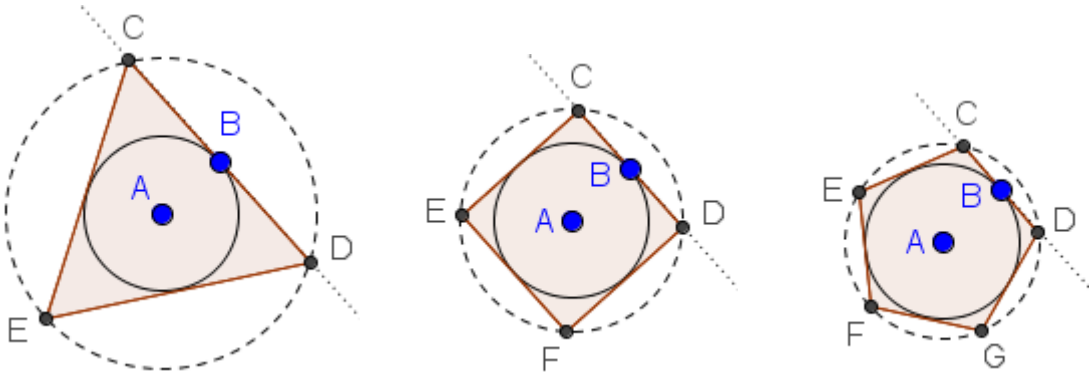





Рис. 23. Правильный n -угольник, описанный около окружности

 *Циркуль.* Если предыдущим инструментом окружность строилась по *центру и радиусу*, то можно сказать, что инструмент “ Циркуль” строит окружность по *радиусу и центру*. Делается это так. Последовательными щелчками

по двум точкам полотна задается радиус, как отрезок между ними. Далее реализуется щелчок по любой точке или позиции – центру окружности. Если последний щелчок был по позиции, то в ней формируется точка. Отметим, что радиус окружности является объектом, зависящим от тех точек, по которым он строился, и поэтому перемещения любой из этих точек будут приводить к изменению радиуса, а значит и самой окружности.



Окружность по трем точкам. Инструмент “ *Окружность по трем точкам*” используется для построения окружности щелчками мышью по трем точкам или позициям на полотне. При щелчках по позициям на их месте формируются точки. Если все точки оказываются на одной прямой, то выводится вырожденная окружность, то есть прямая, проходящая через эти точки.

Примеры 43.

- 1. Около треугольника описать окружность.
- 2. Около правильного многоугольника описать окружность.
- 3. В правильный многоугольник вписать окружность.
- 4. В треугольник вписать окружность.

A. <i>Circle</i> [<i>A</i> , <i>B</i>]	C. <i>Circle</i> [<i>A</i> , <i>seq</i>]
B. <i>Circle</i> [<i>A</i> , <i>r</i>]	D. <i>Circle</i> [<i>A</i> , <i>B</i> , <i>C</i>]


Вывод окружностей
А) По команде *A* выводится окружность с центром в точке *A*, проходящая через точку *B*.

В) По команде *B* выводится окружность с центром в точке *A* радиуса *r* (*r* ≥ 0).


С) По команде *C* выводится окружность с центром в точке *A* радиусом, равным длине отрезка *seq*.

Д) По команде *D* выводится окружность, проходящая через три заданные точки *A*, *B* и *C*.




Полуокружность по двум точкам. Инструмент “ *Полуокружность по двум точкам*” используется для проведения полуокружности щелчками мышью по двум точкам или позициям на полотне. Там, где щелчки были по позициям, формируются точки. Полуокружность проводится так, что исходные или созданные точки оказываются для нее концевыми точками диаметра. Направление выпуклости полуокружности зависит от очередности производимых щелчков.




Дуга по центру и двум точкам. Инструмент “ *Дуга по центру и двум точкам*” используется для построения дуги окружности последовательными щелчками мышью по трем точкам или позициям. Там, где щелчки были по

позициям, формируются точки. Первая точка соответствует центру окружности, вторая – определяет величину радиуса окружности как расстояние от нее до первой точки. Третья точка задает направление дуги, проводимой по часовой стрелке. Она может быть указана в любой позиции полотна.








Дуга по трем точкам. Инструмент “ *Дуга по трем точкам*” используется для построения дуги окружности последовательными щелчками мышью по трем точкам или позициям. Там, где щелчки были по позициям, формируются точки. Исходные или созданные точки задают соответственно начальную, промежуточную и конечную точки дуги.




Сектор по центру и двум точкам. Инструмент “ *Сектор по центру и двум точкам*” используется для построения сектора последовательными щелчками мышью по трем точкам или позициям. Там, где щелчки были по позициям, формируются точки. Первая точка соответствует центру окружности, вторая – определяет радиус окружности как отрезок от нее до первой точки. Третья точка задает направление дуги сектора, проводимой по часовой стрелке. Она может быть указана в любой позиции полотна.

Пример 44. Вписать окружность в сектор с углом, меньшим 180° .

Решение. Последовательность действий для решения предложенной задачи может быть, например, такой:

- инструментом “ *Сектор по центру и двум точкам*” (A – центр, B и C – точки) выведем сектор s ;
- инструментом “ *Биссектриса угла*” проведем биссектрису $\angle CAB$;
- инструментом “ *Пересечение*” найдем точку D пересечения биссектрисы и дуги окружности сектора;
- инструментом “ *Перпендикулярная прямая*” из точки B проведем линию, перпендикулярную к биссектрисе;
- инструментом “ *Пересечение*” найдем точку E пересечения полученных взаимно перпендикулярных линий;
- через строку ввода введем AB и изменим обозначение отрезка на R ;
- через строку ввода введем BE и изменим обозначение отрезка на a ;
- радиус вписанной в сектор окружности через величины a и R находится по формуле $R \cdot a / (R + a)$ (докажите!). Через строку ввода введем выражение $R \cdot a / (R + a)$ и изменим сформированное для него обозначение на r ;
- через строку ввода введем выражение $(R - r) / r$ и дадим этому числу новое имя λ (λ – отношение, в котором центр вписанной в сектор окружности делит R);

- через строку ввода введем выражение $(A+\lambda \cdot D)/(1+\lambda)$ (формула деления отрезка в данном отношении) и получим центр F вписанной в сектор окружности;

- инструментом “ Окружность по центру и точке” проведем вписанную окружность с центром в F через точку D .

Перемещениями точек A, B и C можно изменять местоположение и иные характеристики сектора. На рис. 24 приведены два статичных кадра, получаемых при смещениях точки C .

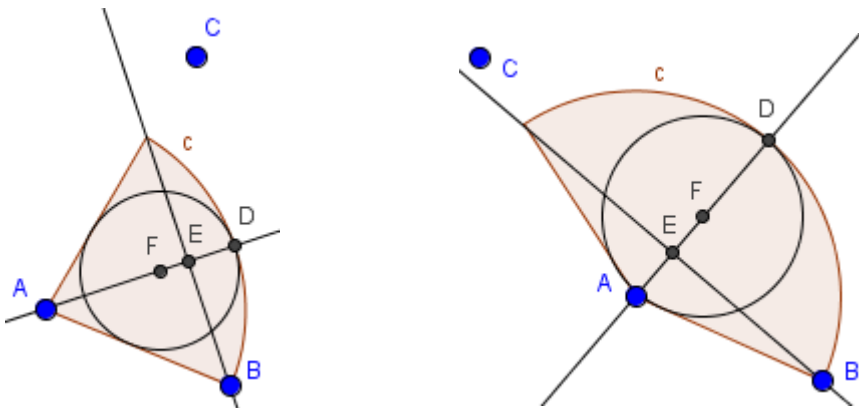



Рис. 24. Сектор и вписанная в него окружность



Сектор по трем точкам. Инструмент “ Сектор по трем точкам” используется для построения сектора последовательными щелчками мышью по трем точкам или позициям. Там, где щелчки были по позициям, формируются точки. Исходные или созданные точки задают соответственно начальную, промежуточную и конечную точки дуги сектора.

A. <i>Semicircle</i> [A, B] B. <i>CircularArc</i> [A, B, C] C. <i>CircumCircularArc</i> [A, B, C]	D. <i>CircularSector</i> [A, B, C] E. <i>CircumCircularSector</i> [A, B, C]
---	--

Вывод дуг окружностей и секторов

- A) По команде A выводится полуокружность с концевыми точками A и B .
- B) По команде B выводится дуга окружности с центром в точке A , находящаяся между прямыми AB и AC . Дуга всегда рисуется от точки B против часовой стрелки. Точка C не обязана находиться на дуге.
- C) По команде C выводится дуга окружности, проходящая через точки A, B, C , из которых A и C – концевые точки.
- D) По команде D выводится сектор окружности, находящийся между прямыми AB и AC с центром в точке A . Дуга сектора всегда рисуется от точки B против часовой стрелки. Точка C не обязана находиться на дуге.

Е) По команде *E* выводится сектор окружности, проходящая через точки *A*, *B*, *C*, из которых *A* и *C* – концевые точки.

3.3.8. Конические сечения

Конические сечения, или, по-другому, коники – это кривые, получаемые пересечением плоскости с круговым конусом. Основные типы невырожденных конических сечений – это эллипс, гипербола и парабола. К вырожденным коническим сечениям относят точку, прямую и пару прямых. В декартовых координатах конические сечения описываются уравнением вида $ax^2 + bxy + cy^2 + dx + ey + f = 0$. Ниже речь идет об инструментах, позволяющих строить конкретные невырожденные коники, а также некоторую конику по пяти точкам, никакие четыре из которых не лежат на одной прямой.

Все невырожденные конические сечения, кроме окружности, можно описать одним и тем же способом. Выберем на плоскости точку *F* (фокус), прямую *d* (директрису) и зададим вещественное число $e \geq 0$ (эксцентриситет). Тогда совокупность точек *M*, для которых отношение расстояний до фокуса и до директрисы постоянно и равно *e*, является коническим сечением, то есть $FM/MA = e$, где точка *A* лежит на директрисе и $MA \perp d$. При этом, в зависимости от величины эксцентриситета, коническое сечение является параболой – $e=1$, эллипсом – $e < 1$ или гиперболой – $e > 1$.



Эллипс. Инструмент “Эллипс” используется для построения эллипса последовательными щелчками мышью по трем точкам или позициям. Там, где щелчки были по позициям, формируются точки. Первая и вторая точки соответствуют фокусам эллипса, а третья точка лежит на эллипсе (см. рис. 25).

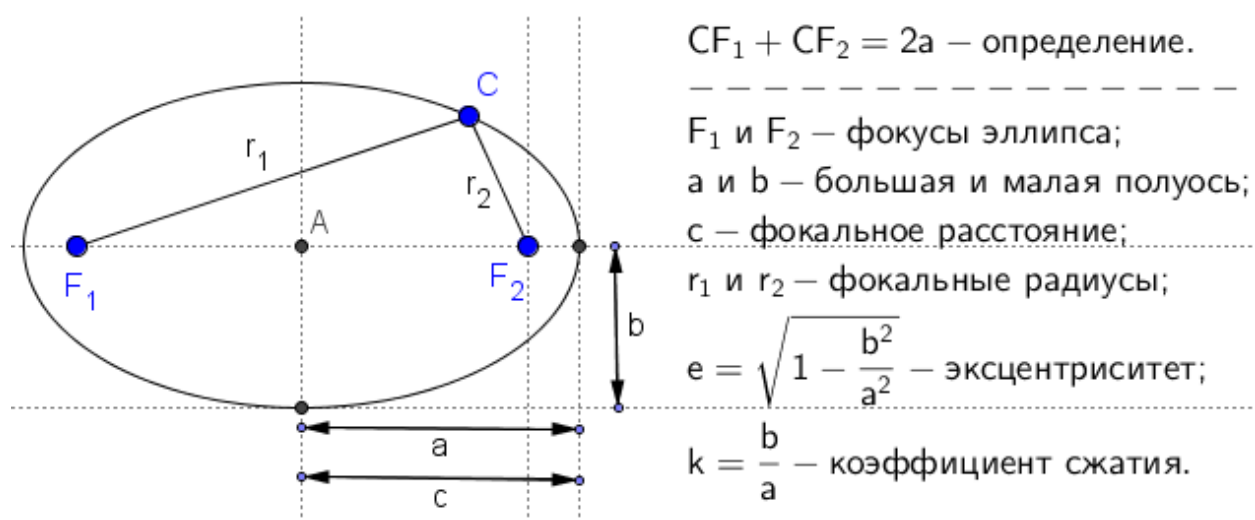


Рис. 25. Эллипс и некоторые его характеристики





Примеры 45.

1. Сформируйте изображение, показанное на рис. 25.

2. Проведите в эллипсе вписанную и описанную окружности.
3. Создайте надпись для вывода площади эллипса ($S=\pi \cdot a \cdot b$).




Пример 46. В данном примере демонстрируется определение эллипса как “совокупности точек плоскости, для которых сумма расстояний до двух данных точек F_1 и F_2 , называемых фокусами, постоянна и больше F_1F_2 ”.

Решение. Проведем следующие действия:

- инструментом “ Эллипс” выведем эллипс c по фокусам и точке C на нем. Через контекстное меню переименуем фокусы на F_1 и F_2 и спрячем точку C ;
- инструментом “ Точка на объекте” прикрепим к эллипсу точку D ;
- инструментом “ Отрезок” соединим отрезками точку D с фокусами F_1 и F_2 . Через панель “Настройки” переименуем построенные отрезки соответственно на r_1 и r_2 , увеличим их ширину и сделаем темно-красными;
- инструментом “ Текст” выведем на полотно текстовую надпись с постоянной и переменной частями. Делается это вводом в поле *Edit* панели *Text* выражения

$$r_1 + r_2 = \boxed{r_1 + r_2} .$$

В результате этих действий получим нечто похожее на рис. 26. Если теперь перемещать точку D по эллипсу, а по-иному и не получится, то правая вычисляемая часть надписи изменяться не будет. Если же сместить точку F_1 или точку F_2 , то, вообще говоря, изменится и $r_1 + r_2$, а вместе с ним и переменная часть надписи. Далее можно снова перемещать точку D и т. д.

Через контекстное меню точки D можно также запускать и останавливать анимацию, то есть автоматическое движение D по эллипсу с актуализацией надписи. Приостанавливать анимацию можно щелчком по кнопке “”, появившейся в нижнем левом углу полотна. При останове анимации кнопка “” принимает вид “” и щелчок по ней запускает анимацию заново.

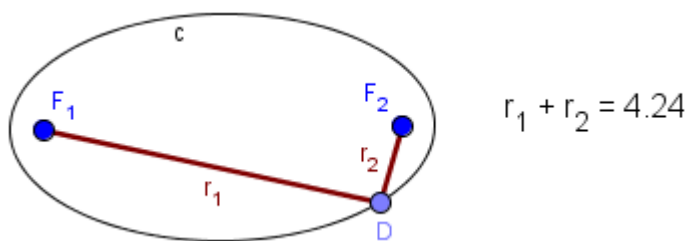






Рис. 26. Модель для демонстрации определения эллипса

Пример 47. Создадим динамическую модель для вычерчивания эллипса с полуосями a и b .

Решение. На рис. 27а показана готовая к работе модель, а на рис. 27b – результат некоторых действий с ней. Построить требуемую модель можно так:

- инструментом “ Ползунок” выведем два одноименных управляющих элемента a и b , задав для них значения в диапазоне от 0.1 до 4 с шагом 0.1. Установим начальные значения для ползунков соответственно 2 и 1;
- инструментом “ Окружность по центру и радиусу” проведем две концентрические окружности из некоторой точки-центра A , указав значение радиусов, равными соответственно a и b ;
- инструментом “ Точка на объекте” сформируем на внешней окружности точку B (см. рис. 27а). Инструментом “ Отрезок” соединим отрезком точки A и B . Инструментом “ Пересечение” найдем точку пересечения C отрезка AB и внутренней окружности;
- инструментом “ Прямая” проведем вспомогательную приблизительно горизонтальную линию AD ;
- инструментом “ Параллельная прямая” через точку C проведем прямую, параллельную AD . Инструментом “ Перпендикулярная прямая” через точку B проведем прямую, перпендикулярную AD . Инструментом “ Пересечение” найдем точку пересечения E проведенных двух прямых;
- инструментом “ Отрезок” соединим отрезком точки C и E , а также B и E . Изменим стиль вывода отрезков CE , BE и точки E ;

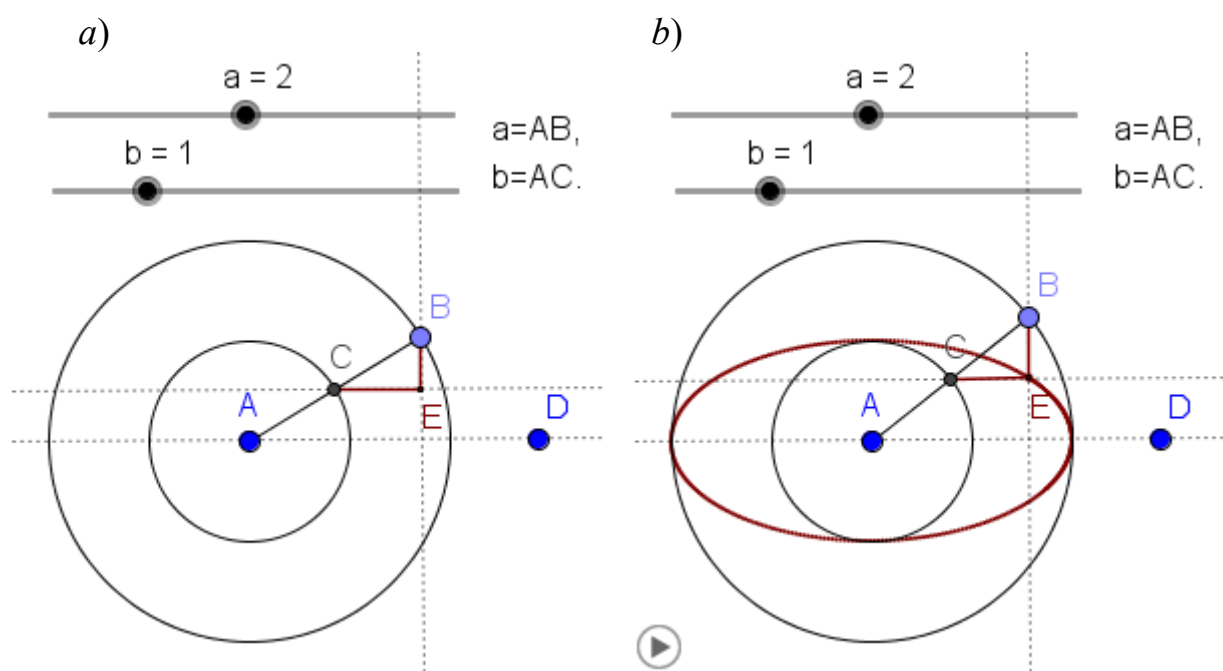


Рис. 27 (а, б). Модель для вычерчивания эллипса с полуосями a и b

Построение модели закончено. Работать с ней следует так. Через контекстное меню для точки E включим режим “Оставлять след” (*Trace On*), а для точки B – режим “Анимировать” (*Animation On*). При этом сразу же запускается анимация и точкой E вычерчивается эллипс.

Замечания. 1. Управление размерами полуосей эллипса осуществляется бегунками. После изменения значений бегунков перед запуском анимации можно убрать уже имеющуюся трассировку, то есть след точки E . Делается это ключом $Ctrl+F$, командой меню *View\Refresh View*, или командой *Zoom-In[1]*, выполняемой из строки ввода.

2. Если команду *ZoomIn[1]* задать в качестве скрипта для бегунков, выполняемого по событию *On Update*, то чистка следа будет происходить при любых изменениях значений слайдеров, а также при остановке анимации по кнопке “⏏”. При этом остановка анимации через контекстное меню след сохраняет.

Пример 48. Для приближенного вычисления периметра эллипса с полуосями a и b можно использовать одну из следующих формул [85]:

$$L1 = 4 \frac{\pi ab + (a-b)^2}{a+b}, \quad (2)$$

$$L2 = 4(a^p + b^p)^{1/p}, \quad \text{где } p = \ln(2) / \ln\left(\frac{\pi}{2}\right) - \text{(Р. Майертенс)}, \quad (3)$$

$$L3 = \pi \left[3(a+b) - \sqrt{(3a+b)(3b+a)} \right] - \text{(С. А. Рамануджан)}, \quad (4)$$

$$L4 = 4(a+b) - \frac{2(4-\pi)ab}{((a^q + b^q)/2)^{1/q}} - \text{(Д. Кантрел)}, \quad (5)$$

где $q = 0.825056176207$.






$$L5 = 4(a+b) \left(\frac{\pi}{4} \right)^{\left(\frac{4ab}{(a+b)^2} \right)} - \text{(Ш. Зафари)}. \quad (6)$$


Построить динамическую модель для выяснения вопроса “по какой из формул (2)-(6) предпочтительнее приближенно вычислять периметр эллипса при его различных полуосях и эксцентриситете”. При вычислениях можно использовать, тот факт, что точное значение периметра эллипса задается полным эллиптическим интегралом второго рода

$$L = 4a \int_0^{\pi/2} \sqrt{1 - \left(1 - \frac{b^2}{a^2}\right) \cdot \cos^2(x)} dx, \quad (7)$$

который в элементарных функциях не выражается, но в *GeoGebra* его можно вычислить с точностью до 15 знаков после десятичной точки.

Решение. Построить требуемую модель (см. рис. 28) можно так:

- инструментом “ *Эллипс*” выведем эллипс по фокусам A, B и точке C на нем;
- инструментом “ *Прямая*” проведем прямую через фокусы A и B ;
- инструментом “ *Срединный перпендикуляр*” проведем перпендикулярную к AB прямую посередине между фокусами A и B ;
- инструментом “ *Пересечение*” отметим центр эллипса D как точку пересечения двух проведенных прямых, а также точки E и F пересечения этих прямых с эллипсом;
- инструментом “ *Отрезок*” проведем полуоси эллипса DE и DF , переименуем их соответственно на a и b и изменим стиль вывода;
- через строку ввода сформируем вспомогательные константы $p = \ln(2)/\ln(\pi/2)$ и $q = 0.825056176207$ и приближенно вычислим интеграл (7):

- через меню командой “*Options/Rounding/5 Decimal Places*” назначим вывод числовых значений с 5 знаками после десятичной точки;
- инструментом “ Текст” выведем на полотно текстовую надпись с помощью выражения:

$$\begin{aligned} |L-L1| &\rightarrow \boxed{\text{abs}(L-4(\pi^*a^*b+(a-b)^2)/(a+b))} \quad , \\ |L-L2| &\rightarrow \boxed{\text{abs}(L-4(a^p+b^p)^{(1/p)})} \quad , \\ |L-L3| &\rightarrow \boxed{\text{abs}(L-\pi^*(3(a+b)-\text{sqrt}((3a+b)^*(3b+a))))} \quad , \\ |L-L4| &\rightarrow \boxed{\text{abs}(L-(4(a+b)-(2(4-\pi)^*a^*b)/((a^q+b^q)/2)^{(1/q)}))} \quad , \\ |L-L5| &\rightarrow \boxed{\text{abs}(L-4(a+b)(\pi/4)^{(4a^*b/(a+b)^2)})} \quad . \end{aligned}$$

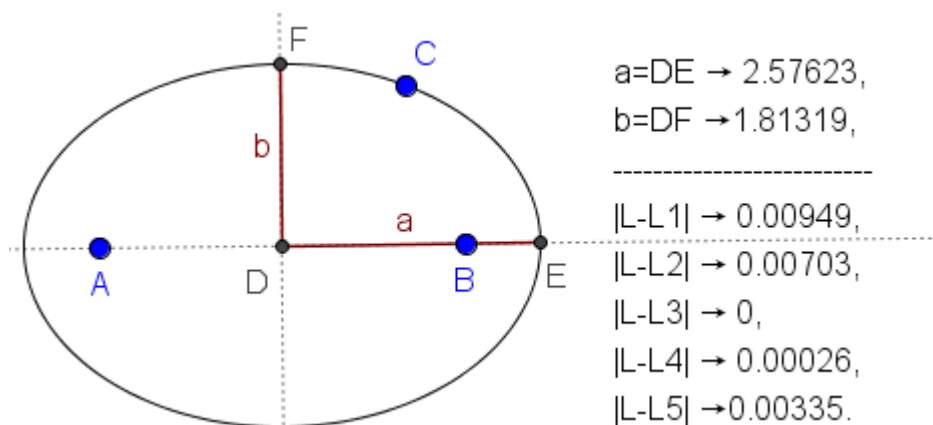















Рис. 28. Модель для вычислений периметра эллипса по разным формулам

Полученная в результате предпринятых действий модель представлена на рис. 28. Точки A , B и C в ней являются свободными. Провести эксперименты с моделью и сделать соответствующие выводы мы предоставляем читателю.

Пример 49. Теорема Фрежье. Пусть P – произвольная точка эллипса. Сформировать динамическую модель для экспериментальной проверки утверждения: “Все хорды, видные из P под прямым углом, пересекаются в одной точке, лежащей на нормали к эллипсу, проходящей через P .”

Решение. Будем считать, что эллипс задан фокусами и точкой на нем. Требуемая модель может быть построена так:

- инструментом “ Эллипс” выведем эллипс по двум фокусам и точке на нем. Через контекстное меню переименуем фокусы на F_1 и F_2 , а точку на эллипсе спрячем;
- инструментом “ Точка на объекте” прикрепим к эллипсу произвольную точку A ;
- проведем три хорды эллипса BD , EF и GH , которые видны из точки A под углом 90° . Первую хорду BD строим так. Инструментом “ Точка на объекте” прикрепим к эллипсу произвольную точку B . Инструментом “ Прямая” проведем прямую между точками B и A . Инструментом “ Перпендикулярная прямая” через точку A проведем прямую, перпендикулярную BA . Инструментом “ Пересечение” сформируем точку D – пересечения проведенного перпендикуляра с эллипсом. Инструментом “ Отрезок” проведем отрезок BD и изменим стиль его вывода. Вторая и третья хорды создаются по аналогии с BD ;
- инструментом “ Пересечение” сформируем точку I пересечения проведенных хорд (визуально кажется, что все три хорды пересекаются в одной точке, хотя пока это экспериментально не подтверждено);
- инструментом “ Касательная” проведем касательную к эллипсу в точке A . Инструментом “ Прямая” проведем прямую через точки I и A . Изменим стиль вывода проведенных прямых. Экспериментально с помощью инструмента “ Отношение” убедимся, что прямая IA перпендикулярна касательной, то есть, что точка I лежит на нормали. Инструментом “ Угол” создадим метку прямого угла в точке A (необязательно);
- инструментом “ Текст” создадим проверочную надпись с помощью выражения

BD , EF и GH пересекаются

в одной точке $I \rightarrow \boxed{AreConcurrent[h,k,n]}$.

Построенная модель показана на рис. 29а. Перемещая точку A по эллипсу с помощью мыши или включив для нее анимацию, в правой вычисляемой части надписи всегда будем иметь значение *true*. Этим и подтверждается проверяемое утверждение.

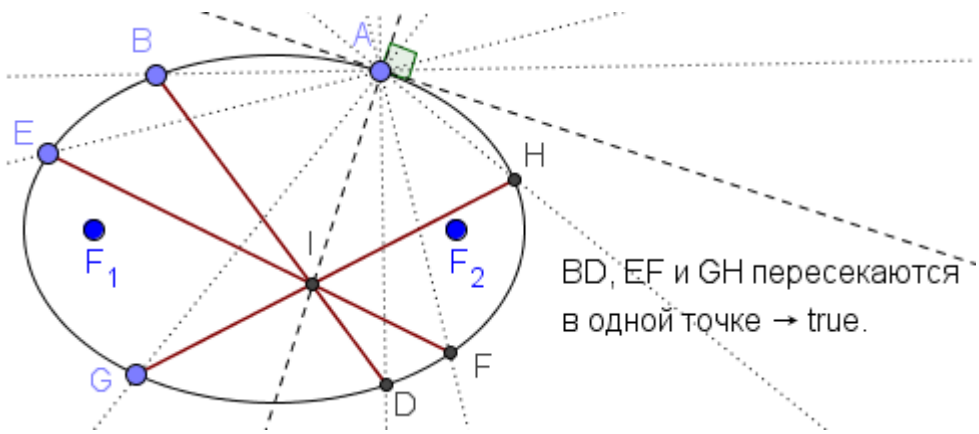


Рис. 29а. Динамическая модель для проверки утверждения примера 49

Замечание. Если в построенной модели через контекстное меню для точки A включить анимацию (*Animation*), а для точки I – трассирование (*Trace On*), то получим изображение, представленное на рис. 29 б. Размер точки I предварительно следует уменьшить. Из полученного изображения напрашивается вывод о наличии более содержательных утверждений из условий теоремы Фрежье по сравнению с теми, которые указаны в примере 49.

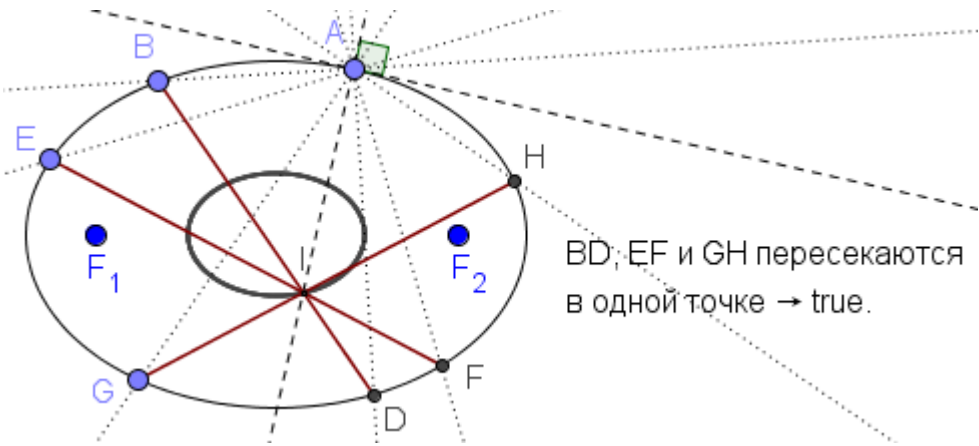



Рис. 29б. Траектория точки I при перемещении A по эллипсу

A. $Ellipse[F_1, F_2, a]$
B. $Ellipse[F_1, F_2, P]$

Вывод эллипса
 А) По команде A выводится эллипс с фокусами в точках F_1 и F_2 и большой полуосью, определяемой аргументом a . Если a – число и $a>0$, то a непосредственно задает длину большей полуоси эллипса. Если a – имя отрезка, то его длина задает длину большей полуоси эллипса. Заметим, что при выводе кривой условие $2a>F_1F_2$ не проверяется, и вместо эллипса может быть выведена гипербола.

В) По команде *B* выводится эллипс с фокусами в точках F_1 и F_2 , проходящий через точку P .



Гипербола. Инструмент “ Гипербола” позволяет строить гиперболу последовательными щелчками мышью по трем точкам или позициям. Там, где щелчки были по позициям, формируются точки. Первая и вторая точки соответствуют фокусам гиперболы, а третья точка – лежит на гиперболе.

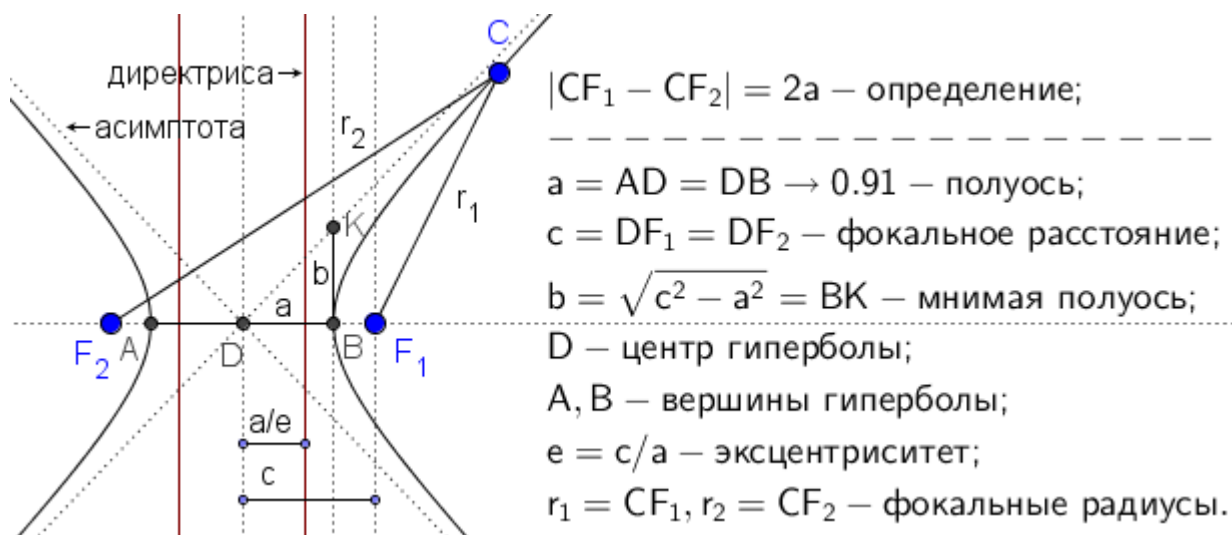






Рис. 30. Гипербола и некоторые его характеристики

Пример 50. Сформируйте изображение, показанное на рис. 30.

Пример 51. В данном примере демонстрируется определение гиперболы как “совокупности точек плоскости, для которых абсолютное значение разности расстояний до двух данных точек F_1 и F_2 , называемых фокусами, постоянна и меньше F_1F_2 ”.

Решение. Проведем следующие действия:

- инструментом “ Гипербола” выведем гиперболу по фокусам и точке C на ней. Через контекстное меню переименуем фокусы на F_1 и F_2 и спрячем точку C ;
- инструментом “ Точка на объекте” прикрепим к гиперболе точку D ;
- инструментом “ Отрезок” соединим отрезками точку D с фокусами F_1 и F_2 . Через панель “Настройки” переименуем построенные отрезки соответственно на r_1 и r_2 , увеличим их ширину и сделаем темно-красными;
- инструментом “ Текст” выведем на полотно текстовую надпись с постоянной и переменной частями. Делается это вводом в поле *Edit* панели *Text* выражения

$$|r_1 - r_2| = \boxed{\text{abs}(r_1 - r_2)} .$$

В результате этих действий получим нечто похожее на рис. 31. Если теперь точку D перемещать по той или другой ветви гиперболы, то надпись изменяться не будет. Если же переместить фокус F_1 или фокус F_2 , то, вообще говоря, изменится и $d=|r_1-r_2|$, а вместе с ним и переменная часть надписи. При измененном d можно снова перемещать D и т. д. Анимация, то есть автоматическое перемещение точки D , в данном случае реализуется тем же способом, что и для соответствующего тренажера для эллипса. При фиксированных F_1 и F_2 , когда D “убегает в бесконечность” по той или иной ветви гиперболы, выражение d может меняться из-за ошибок округления.

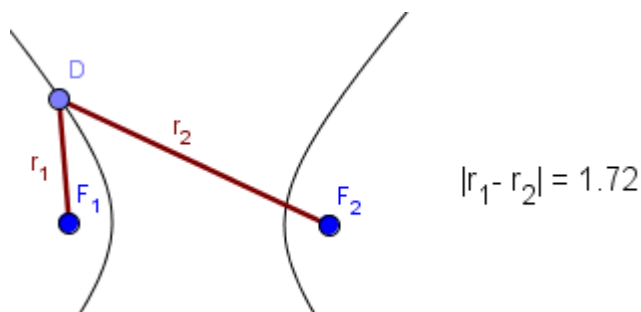










Рис. 31. Модель для демонстрации определения гиперболы

Пример 52. Создадим динамическую модель для вычерчивания гиперболы, заданной фокусами F_1 и F_2 и одной из своих вершин A .

Решение. Будем считать, что гипербола должна быть получена как след от движения по полотну одной или более точек. Возможный вариант решения предложенной задачи таков:

- инструментом “ Прямая” сформируем на полотне прямую, проходящую через две точки, и переименуем эти точки на F_1 и F_2 , считая их фокусами будущей гиперболы;
- инструментом “ Срединный перпендикуляр” проведем прямую, перпендикулярную F_1F_2 и проходящую посередине отрезка F_1F_2 , инструментом “ Пересечение” сформируем точку пересечения этого перпендикуляра и отрезка F_1F_2 , то есть центр гиперболы. Переименуем полученную точку на D ;
- инструментом “ Точка на объекте” между F_1 и D прикрепим точку A – вершину гиперболы. Инструментом “ Отражение относительно точки” по точкам A и D построим вторую вершину гиперболы и переименуем ее на B ;
- инструментом “ Точка на объекте” зафиксируем вне отрезка F_1F_2 на его продолжении любую точку C ;
- инструментом “ Окружность по центру и радиусу” проведем окружность с центром F_1 и радиусом CB , а также окружность с центром F_2 и радиусом CA ;

- инструментом “ Пересечение” сформируем точки пересечения F и E двух построенных окружностей. Поскольку $F_1E=CB$ и $F_2E=CA$, то в данном случае $F_1E-F_2E=CB-CA=CA+AB-CA=AB$ и точка E лежит на гиперболе. То же самое можно сказать и о точке F . Изменяя позицию точки C на прямой F_1F_2 (вне отрезка $[F_1, F_2]$), будем получать другие точки F и E на гиперболе;
- включим через контекстное меню оставление следа при перемещении точек F и E . Перемещая C по прямой F_1F_2 , будем формировать гиперболу, вычерчиваемую следами точек F и E . Можно также запустить анимацию для точки C .

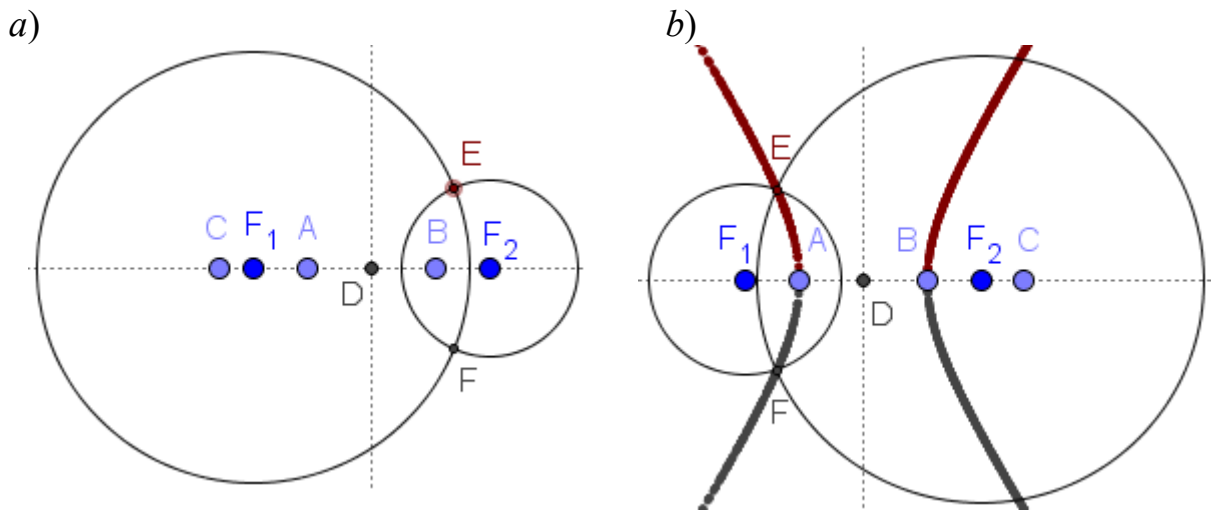


Рис. 32 (a, b). Модель для вычерчивания гиперболы, заданной фокусами F_1, F_2 и одной из своих вершин A

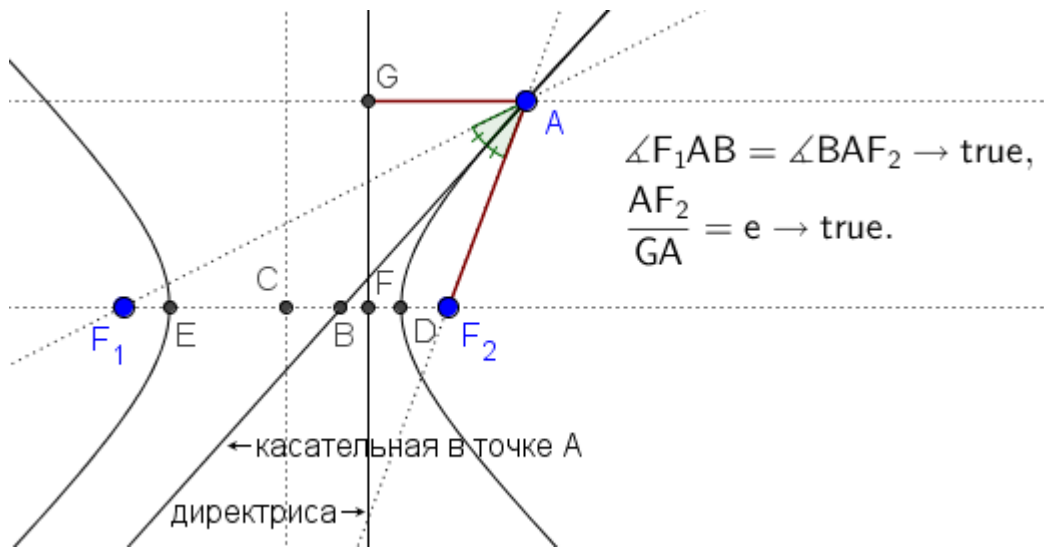


Рис. 33. Модель для проверки утверждений примера 53

Пример 53. Создайте динамическую модель (см. рис. 33) для экспериментальной проверки следующих утверждений:

1. *Оптическое свойство гиперболы.* Если F_1 и F_2 – фокусы гиперболы и A – произвольная точка на ней, то касательная к гиперболе в точке A является биссектрисой $\angle F_1AF_2$;

2. Отношение расстояний от любой точки гиперболы до ближайшего фокуса к расстоянию от этой точки до ближайшей директрисы постоянно и равно эксцентриситету e .


A. *Hyperbola*[F_1, F_2, a]
 B. *Hyperbola*[F_1, F_2, P]

Вывод гиперболы

A) По команде A выводится гипербола с фокусами в точках F_1 и F_2 и действительной полуосью, определяемой аргументом a . Если a – число и $a > 0$, то a непосредственно задает длину полуоси гиперболы. Если a – имя отрезка, то его длина задает длину большей полуоси гиперболы. Заметим, что при выводе кривой условие $F_1F_2 > 2a > 0$ не проверяется и вместо гиперболы может быть выведен эллипс.

B) По команде B выводится гипербола с фокусами в точках F_1 и F_2 , проходящая через точку P .



Парабола. Инструмент “ *Парабола*” позволяет строить параболу двумя щелчками мыши по точке и прямой, или по прямой и по точке или позиции. Если щелчок был по позиции, то в ней формируется точка. Прямая определяет директрису параболы, а точка – фокус параболы. Вместо прямой можно использовать луч или отрезок.

Пример 54. Сформируйте изображение, показанное на рис. 34.

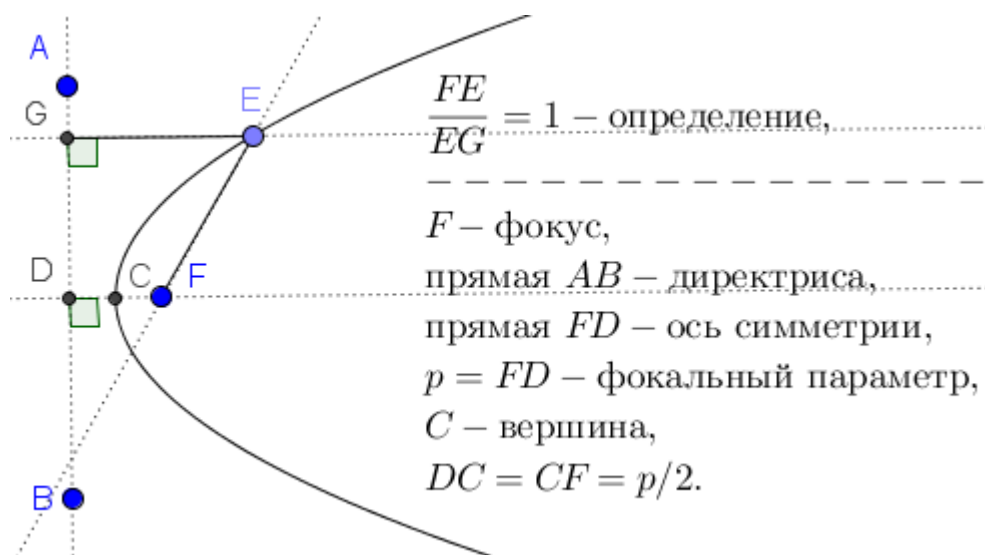










Рис. 34. Парабола и некоторые ее характеристики

Пример 55. В данном примере демонстрируется определение параболы как “совокупности точек плоскости, равноудаленных от некоторой точки F и прямой, называемых соответственно фокусом и директрисой параболы.

Решение. Проведем следующие действия:

- инструментом “ Прямая” проведем директрису будущей параболы. Через контекстное меню переименуем директрису на d и спрячем определяющие ее точки. Инструментом “ Точка” поместим точку вне d и переименуем ее на F ;
- инструментом “ Парабола” выведем параболу по фокусу F и директрисе d ;
- инструментом “ Точка на объекте” прикрепим к параболе точку C . Инструментом “ Перпендикулярная прямая” проведем через точку C прямую, перпендикулярную d . Инструментом “ Пересечение” сформируем точку пересечения D директрисы d и проведенного перпендикуляра;
- инструментом “ Отрезок” соединим отрезками точку C с фокусом F и точкой D ;
- инструментом “ Текст” выведем на полотно текстовую надпись с постоянной и переменной частями. Делается это вводом в поле *Edit* панели *Text* выражения

$$FC=CD \rightarrow \boxed{FC==CD}.$$

В результате этих действий получим нечто похожее на рис. 35. Далее можно проверить, что если перемещать точку C по параболе, изменять позицию директрисы d или смещать фокус F , надпись меняться не будет. Анимация, то есть автоматическое перемещение точки C , в данном случае реализуется тем же способом, что и для соответствующей модели для эллипса.

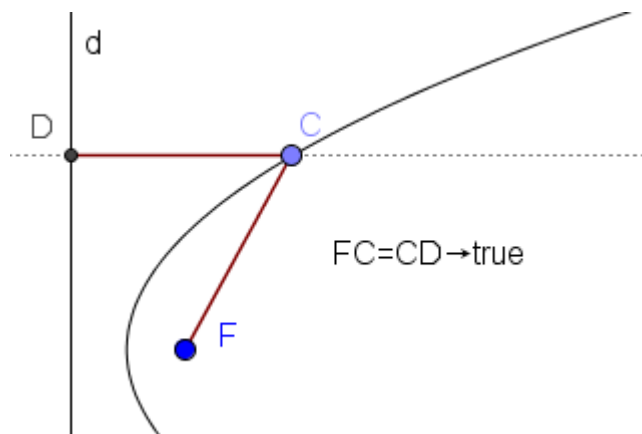











Рис. 35. Модель для демонстрации определения параболы

Пример 56. Создадим динамическую модель для вычерчивания параболы, заданной фокусом F и директрисой d .

Решение. Будем считать, что парабола должна быть получена как след от движения некоторой точки по полотну. Возможный вариант решения предложенной задачи таков:

- инструментом “ Прямая” проведем директрису будущей параболы. Через контекстное меню переименуем директрису на d и спрячем определяющие ее точки. Инструментом “ Точка” поместим на полотно точку вне d и переименуем ее на F ;
- инструментом “ Перпендикулярная прямая” проведем через точку F прямую, перпендикулярную d . Инструментом “ Пересечение” сформируем точку пересечения C директрисы d и проведенного перпендикуляра;
- инструментом “ Точка на объекте” прикрепим к директрисе некоторую точку D . Инструментом “ Перпендикулярная прямая” проведем прямую через точку D перпендикулярно d ;
- инструментом “ Прямая” проведем прямую через точки D и F . Инструментом “ Срединный перпендикуляр” проведем прямую, перпендикулярную DF и проходящую через середину отрезка DF . Инструментом “ Пересечение” сформируем точку пересечения E этой прямой и перпендикуляра к d . Из равнобедренности $\triangle DEF$ вытекает, что $DE=EF$, то есть точка E лежит на параболе;
- включим через контекстное меню оставление следа при перемещении точки E . Перемещая D по директрисе d будем формировать параболу, вычерчиваемую следами точки E . Можно также запустить анимацию для точки D .

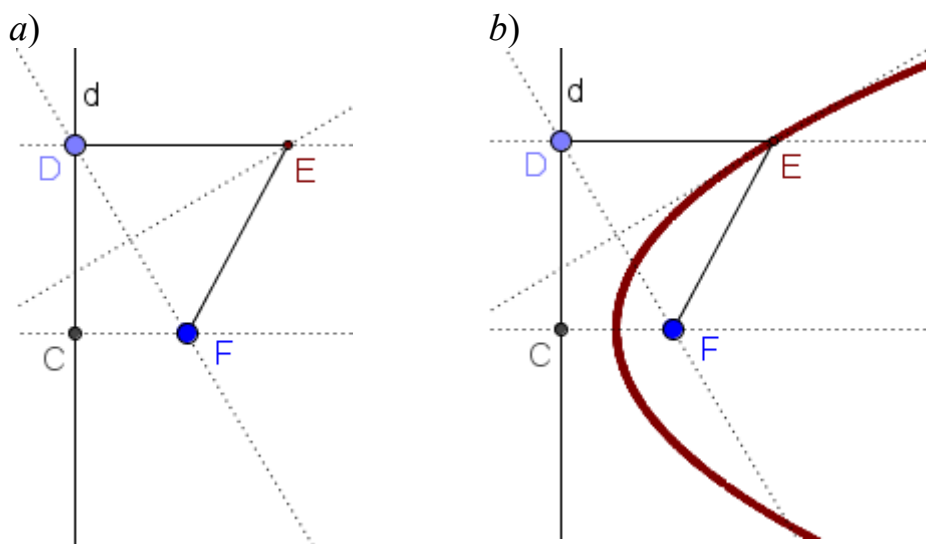


Рис. 36 (a, b). Модель для вычерчивания параболы, заданной фокусом F и директрисой d

Пример 57. Создадим динамическую модель для вычерчивания параболы, заданной фокусом и вершиной.

Пример 58. Оптическое свойство параболы. Создать динамическую модель для экспериментальной проверки того, что любой луч, параллельный оси

симметрии параболы, отражаясь от параболы, проходит через фокус и наоборот, луч исходящий из фокуса отражается параболой по прямой, параллельной ее оси симметрии.

Пример 59. Дополнение к оптическому свойству параболы. Пусть парабола построена по директрисе d и фокусу F . Пусть, далее, через некоторую точку C оси симметрии перпендикулярно к ней проведена прямая g , D – точка пересечения g с параболой, E – произвольная точка на g , h – прямая, проходящая через E параллельно оси симметрии, G – точка пересечения h с параболой (см. рис. 37а, 37b). Создать динамическую модель для экспериментальной проверки того, что для произвольной точки C , расположенной на оси симметрии параболы от ее вершины по направлению к фокусу и далее, справедливы соотношения:

$$FH = \begin{cases} FE + EG, & \text{если } CH \geq CG; \\ FE - EG, & \text{если } CH < CG. \end{cases} \tag{8}$$

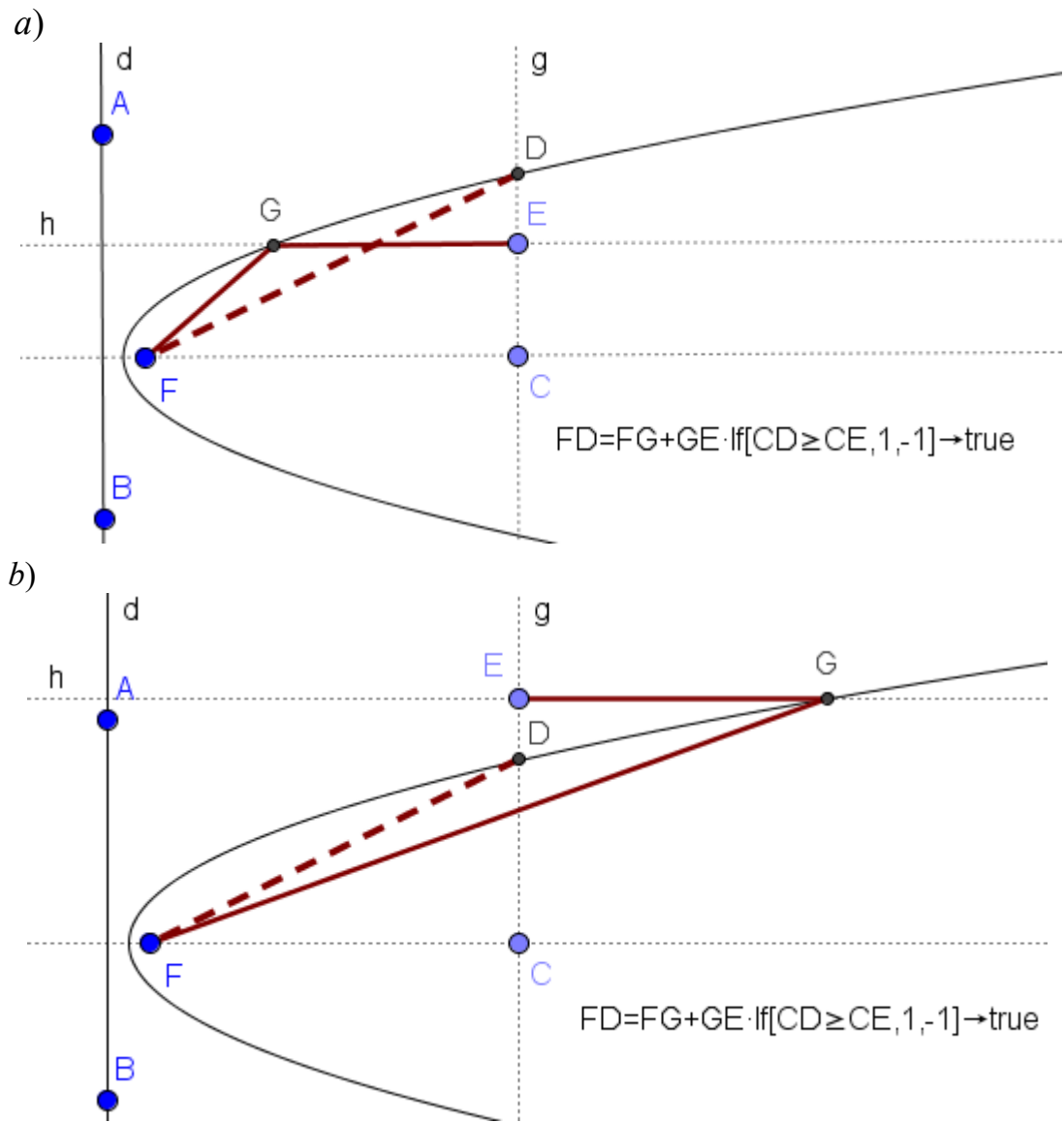


Рис. 37 (а, b). Модель для проверки дополнения к оптическому свойству параболы

На модели рис. 37a свободными объектами являются директриса d , фокус F , а также прикрепленные к прямым точки E и C . Изменение позиций d и F приводит к другим параболам, а изменение E и C – к перестраиванию проверяемой конфигурации. При этом проверка утверждения возможна как при фиксированном отрезке FD – перемещение E по g , так и при изменяемом отрезке FD – перемещение C по оси симметрии. Рис. 37b отличается от рис. 37a только позицией точки E на вертикальной прямой к оси параболы. Соотношения (8) для проверочной надписи преобразованы к виду $FD=FG+If[CD\geq CE,1,-1]\cdot GE$. Поясним, почему используется функция $If[CD>CE,1,-1]$, а не $Sign[CD-CE]$. Дело в том, что при неопределенном условии в If оно считается ложным, а неопределенный аргумент $Sign$ вычислить функцию не позволяет. В нашем случае точка E не определена при позициях C от вершины параболы и далее в направлении директрисы и после нее. В этом случае If возвращает -1, но это нам и нужно.


Пример 60. Построить модель, аналогичную модели примера 59, но с прикрепленной к параболе точкой G и несвободной точкой E . Другими словами, точку G разрешается перемещать по параболе, а точка E фиксируется как пересечение прямых g и h (см. рис. 37a и 37b).

A. Parabola[F, d]

Вывод параболы

А) По команде A выводится парабола с фокусом в точке F и директрисой d . Директриса d может быть задана уравнением прямой линии, именем уравнения прямой линии или функцией $Line[A, B]$, где A, B – точки.



Коника по 5 точкам. Конические сечения, или, по-другому, коники – это кривые, получаемые пересечением плоскости с круговым конусом. Мы уже рассмотрели построение невырожденных коник типа эллипс, гипербола или парабола по тем или иным их характеристикам. Инструмент “ Коника по 5 точкам” используется для построения конического сечения по пяти различным точкам. Если по крайней мере 4 точки лежат на одной линии, то коническое сечение не определено.

Замечание. На панели “Объекты” созданные конические сечения представляются в виде “ $h:a\cdot x^2+b\cdot x\cdot y+c\cdot y^2+d\cdot x+e\cdot y=f$ ”, где h – имя коники, $a\cdot x^2+b\cdot x\cdot y+c\cdot y^2+d\cdot x+e\cdot y=f$ – уравнение коники (a, b, c, d, e и f – действительные числа). Получить список коэффициентов коники по ее имени h можно с помощью команды $Coefficients[h]$, возвращающей список вида $\{a, c, -f, b, d, e\}$. Вместо имени h аргументом команды $Coefficients$ может быть уравнение коники с любым порядком следования ее членов.



Пример 61. Постройте гиперболу Киперта для произвольного $\triangle ABC$. Она проходит через вершины $\triangle ABC$, его центроид и ортоцентр. Последние две точки можно вывести на полотно через строку ввода функцией *TriangleCenter*[*A,B,C,k*] соответственно при $k=2$ и $k=4$.



Пример 62. Постройте гиперболу Фейербаха для произвольного $\triangle ABC$. Она проходит через вершины $\triangle ABC$, его инцентр (центр вписанной окружности) и ортоцентр. Последние две точки можно вывести на полотно через строку ввода функцией *TriangleCenter*[*A,B,C,k*] соответственно при $k=1$ и $k=4$.

<p>A. <i>Conic</i>[<i>A, B, C, D, E</i>] B. <i>Conic</i>[<i>a, b, c, d, e, f</i>]</p>	<p><i>Вывод коники по пяти точкам и коэффициентам уравнения</i> A) Командой <i>A</i> выводится коника по 5 заданным точкам <i>A, B, C, D, E</i>.</p>
--	---

B) Командой *B* выводится коника, имеющая уравнение $a \cdot x^2 + d \cdot x \cdot y + b \cdot y^2 + e \cdot x + f \cdot y = -c$.

3.3.9. Углы, длины, площади

Основная тема, обсуждаемая в данном пункте, – это инструменты и команды для работы с углами. Кроме того, рассматриваются вопросы вычисления длин, периметров и площадей для ряда объектов, а также действие специальных инструментов “ *Наклон прямой*” и “ *Создать список*”.

 *Угол.* Инструмент “ *Угол*” формирует угол, точнее – графическую метку угла, если на полотне указать: два пересекающихся объекта, каждый из которых может быть прямой, лучом или отрезком; два вектора; три точки или позиции; многоугольник. В последнем случае формируются графические метки сразу всех углов многоугольника, хотя возможно и их индивидуальное создание.

Если указывается позиция на полотне, то в этом месте появляется точка. При указании трех точек вторая из них становится вершиной угла. По умолчанию углы показываются в диапазоне $0..2\pi$, но в установках этот диапазон может быть изменен на $0..\pi$ или $\pi..2\pi$. Спрятать угол или его обозначение, как и любой другой объект, можно через контекстное меню. Порядок указания точек важен. Если при щелчках по точкам в порядке *A, B* и *C* формируется $\angle ABC$, то при щелчках по точкам *C, B* и *A* формируется $\angle CBA = 2\pi - \angle ABC$ и наоборот.

Векторы не обязательно должны исходить из одной точки. Угол в этом случае формируется в начальной точке первого указанного вектора.

При указании многоугольника общего положения, жесткого многоугольника или векторного многоугольника сразу формируются метки всех его внут-

ренных или внешних углов. Внутренние углы отмечаются тогда, когда многоугольник создавался указанием точек или позиций в порядке их следования против часовой стрелки. Внешние углы отмечаются тогда, когда многоугольник создавался указанием точек или позиций в порядке их следования по часовой стрелке. Для правильных многоугольников отмечаются внутренние углы. Внешние углы в этом случае могут индивидуально отмечаться указанием соответствующих сторон или вершин многоугольника.

Если некоторые углы объекта помечены, а затем объект претерпевает те или иные изменения, то ориентация всех или некоторых углов может измениться. В этом случае изменяются и метки соответствующих углов. На рис. 38a приведен $\triangle ABC$ с помеченными углами при вершинах A , B и C . На рис. 38b показан этот же треугольник после изменения позиции его вершины A . Как видим, в данном случае метки углов заменились метками дополнительных до 360° углов.

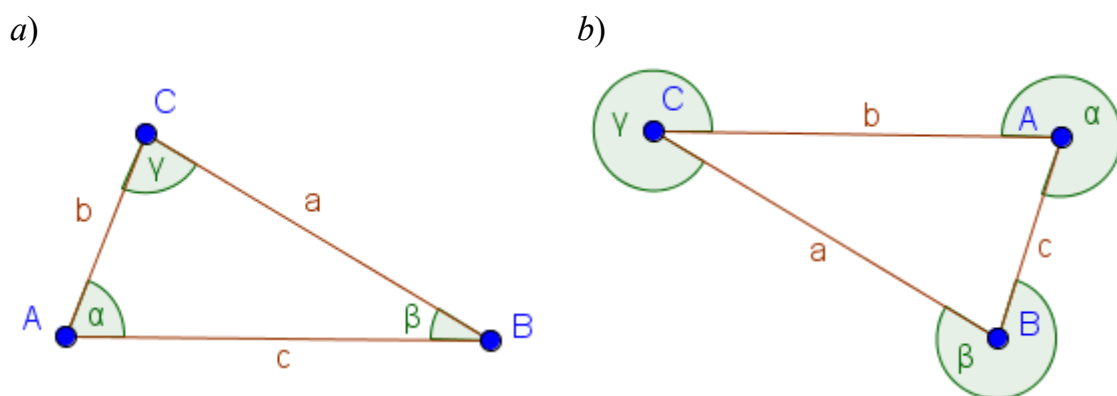


Рис. 38 (a,b). Изменение меток углов при перемещении вершины A $\triangle ABC$

Если метки углов создавались сразу все, то есть для объекта “треугольник”, то восстановить их на полученном изображении уже нельзя. Но если они создавались для каждого угла в отдельности, то через панель “Настройки” это сделать можно изменением определения углов, например, так $Angle[C,B,A] \rightarrow Angle[A,C,B]$, Более того, можно сделать и так, чтобы динамические изменения треугольника вообще не влияли на метки углов. Добиться этого можно, определяя метки через строку ввода или переопределяя их на панели “Настройки” с помощью функций $Angle$ и If , например, так:

$$\begin{aligned} &If[Angle[C,A,B] < \pi, Angle[C,A,B], Angle[B,A,C]], \\ &If[Angle[A,B,C] < \pi, Angle[A,B,C], Angle[C,B,A]], \\ &If[Angle[B,C,A] < \pi, Angle[B,C,A], Angle[A,C,B]]. \end{aligned}$$

На панели “Настройки” можно задать размер выводимых меток углов, их цвет или заполнение теми или иными шаблонами, различное декоративное оформление. Некоторые возможные варианты оформления меток углов показаны на рис. 39.

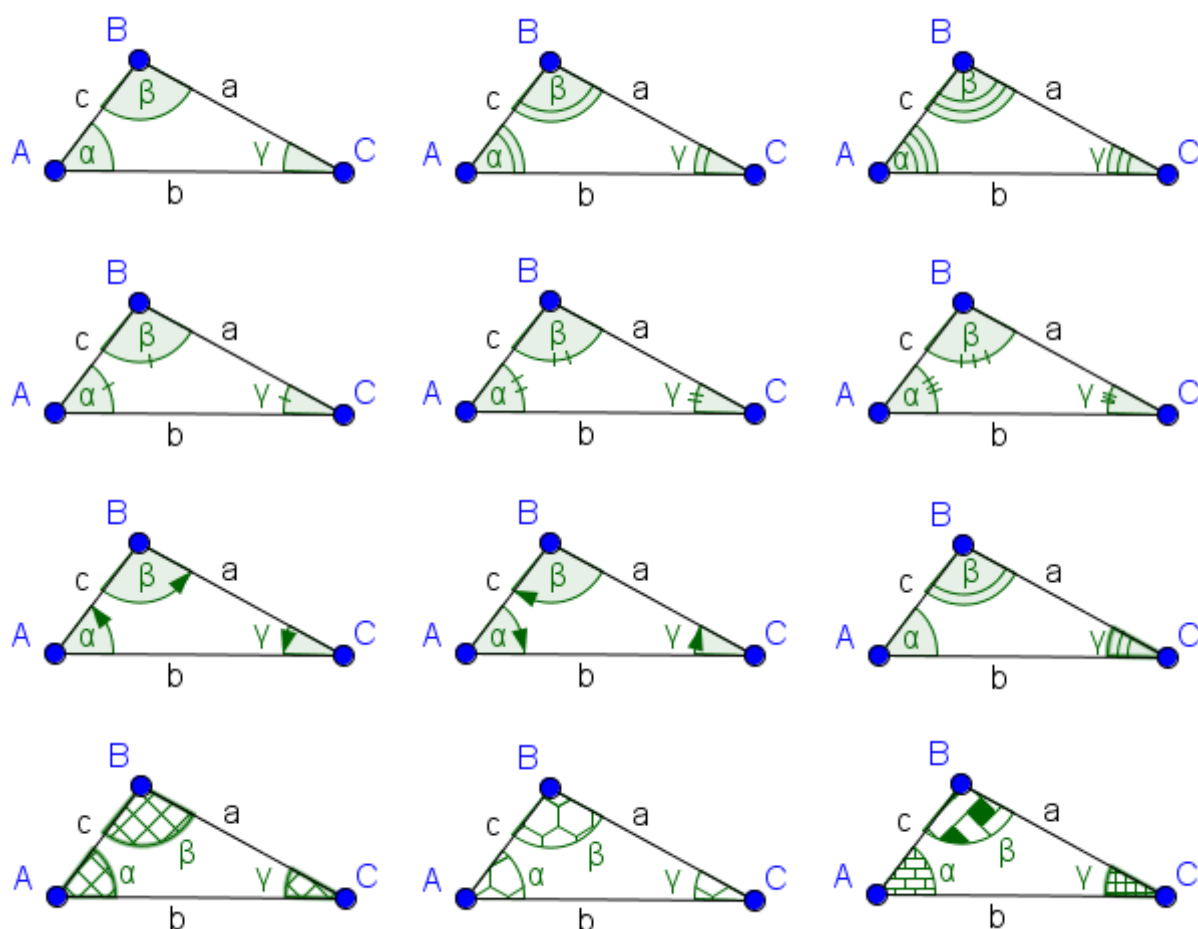


Рис. 39. Некоторые варианты оформления меток углов

Остановимся еще на нескольких моментах, связанных с углами:

1) *Одинаковые имена углов.* Иногда для разных углов желательно иметь одинаковые имена, например, в двух подобных треугольниках соответствующие равные углы требуется обозначить одинаковыми греческими буквами. Но это противоречит уникальности имен объектов. Выход из этой ситуации таков. Через панель “*Настройки*” сформировать для углов одного из треугольников кроме имен еще и заголовки (*captions*) и включить вывод заголовков, а не имен. Именно это и демонстрируется примером рис. 40.

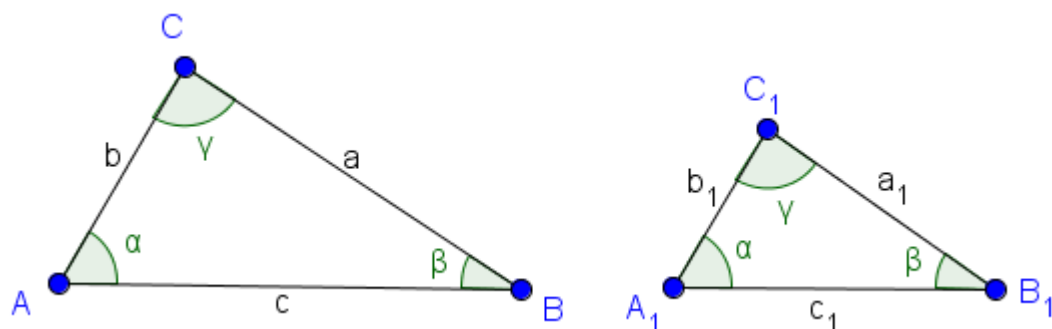


Рис. 40. Вывод одинаковых “имен” углов

2) *Метки углов, частично перекрывающихся друг друга.* Если требуется создать метки двух углов с общей вершиной, которые частично перекрывают друг друга, то сделать это можно за счет изменения размеров меток так, как это показано на рис. 41.

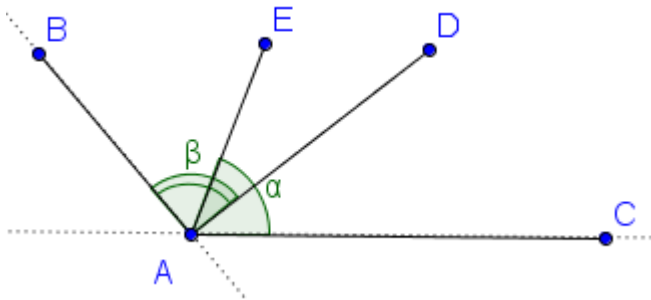


Рис. 41. Метки углов, частично перекрывающихся друг друга

3) *Метки прямых углов.* Установки вида меток прямых углов осуществляются радиокнопками на вкладке “Дополнительно” (*Advanced*) панели “Настройки”. Все возможные здесь случаи показаны на рис. 42. При этом обозначение “ $\alpha=90^\circ$ ” может быть спрятано или выведено иначе. Заметим, что включенная установка воздействует сразу на все углы текущего документа, в том числе и на ранее созданные.












Установка	<input type="radio"/> <input type="checkbox"/>	<input type="radio"/> <input checked="" type="checkbox"/>	<input type="radio"/> <input checked="" type="checkbox"/>	<input type="radio"/> Off
Результат				

Рис. 42. Установки для меток прямых углов и их действие

Пример 63. Сформировать динамическую модель для экспериментальной проверки утверждения: “Геометрическим местом точек, из которых данный эллипс виден под прямым углом, является окружность с центром в центре эллипса”.

Решение. Будем считать, что эллипс задан фокусами и точкой на нем. Пусть a и b – полуоси этого эллипса. Проверим, что из точек окружности с центром в центре эллипса и радиусом $\sqrt{a^2 + b^2}$ точки эллипса действительно видны под углом 90° , и что других таких точек больше нет. Требуемая модель может быть построена так:

- инструментом “ Эллипс” выведем эллипс по двум фокусам и точке на нем. Через контекстное меню фокусы переименуем на F_1 и F_2 , а точку на эллипсе спрячем;

- инструментом “ Прямая” проведем прямую между F_1 и F_2 . Инструментом “ Срединный перпендикуляр” проведем прямую, перпендикулярную F_1F_2 ;
- инструментом “ Пересечение” сформируем точки A и B – пересечения эллипса с проведенными прямыми (см. рис. 43), а также точку D – пересечения этих прямых, то есть центр эллипса;
- инструментом “ Параллельная прямая” через точку A проведем прямую, параллельную F_1F_2 , а через точку B – прямую, параллельную DA . Инструментом “ Пересечение” сформируем точку E – пересечения проведенных прямых. Ясно, что из точки E эллипс виден под углом 90° и, кроме того, $DE = \sqrt{a^2 + b^2}$;
- инструментом “ Окружность по центру и точке” проведем окружность с центром в D через точку E ;
- инструментом “ Точка на объекте” прикрепим к окружности произвольную точку F . Инструментом “ Касательная” проведем касательные из точки F к эллипсу и изменим стиль их вывода. Инструментом “ Пересечение” сформируем точки G и H – касания с эллипсом проведенных прямых;
- инструментом “ Угол” отметим $\angle GFH$ и $\angle AEB$ и через контекстное меню спрячем их обозначение. Углы окажутся помеченными как прямые;
- инструментом “ Текст” создадим проверочную надпись с помощью выражения

$$\angle GFH=90^\circ \rightarrow \boxed{\alpha==90^\circ} .$$

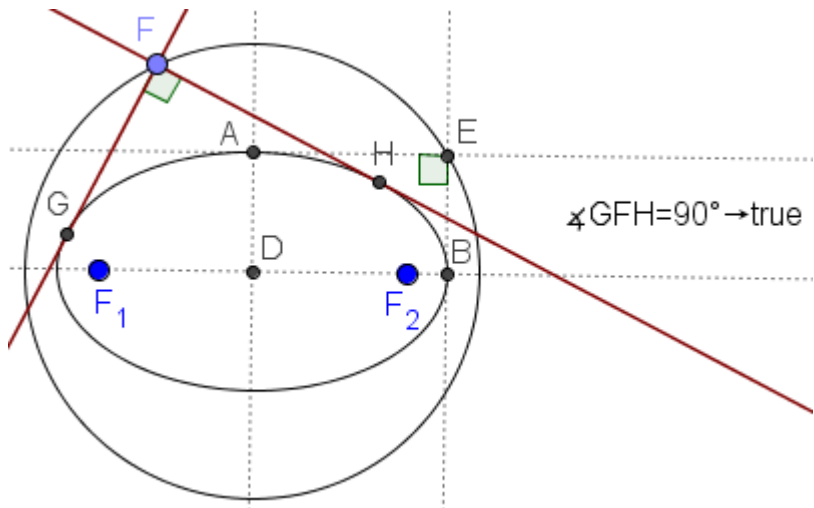


Рис. 43. Динамическая модель для проверки утверждения примера 61

Перемещая точку F по окружности с помощью мыши или включив для нее анимацию, в правой вычисляемой части надписи всегда будем иметь значение *true*. Этим и подтверждается, что из точек окружности эллипс виден под

прямым углом. То, что таких точек нет внутри эллипса или вне его – факт очевидный.

Примеры 64. Оптическое свойство эллипса. Создать динамическую модель для экспериментальной проверки утверждения: “Если в один из фокусов эллипса (эллиптического зеркала) поместить источник света, то лучи света отразившись от эллипса соберутся в другом фокусе”. Модель должна выглядеть приблизительно так, как это показано на рис. 44.

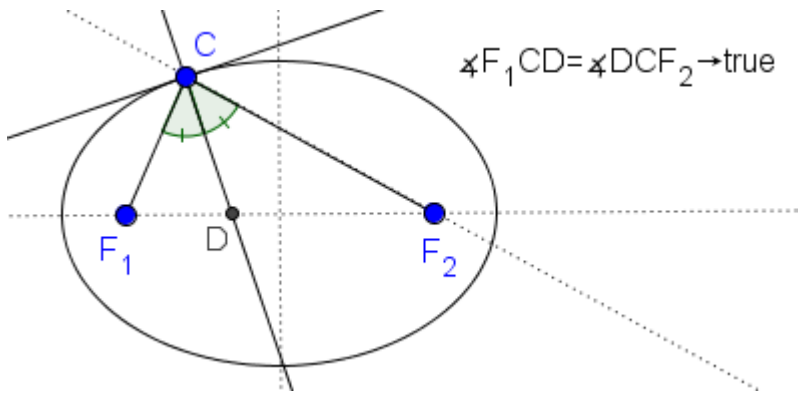


Рис. 44. Динамическая модель для проверки оптического свойства эллипса

Примеры 65. Оптическое свойство гиперболы. Создать динамическую модель для экспериментальной проверки утверждения: “Если в один из фокусов гиперболы (гиперболического зеркала) поместить источник света, то лучи света отразившись от гиперболы расходятся так, как если бы они исходили из другого фокуса”. Модель должна выглядеть приблизительно так, как это показано на рис. 45.

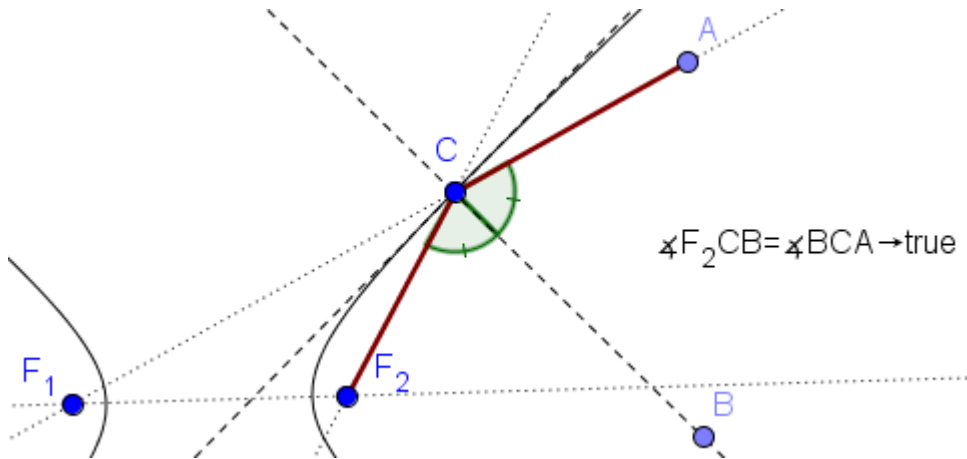


Рис. 45. Динамическая модель для проверки оптического свойства гиперболы

Примеры 64. Оптическое свойство параболы. Создать динамическую модель для экспериментальной проверки утверждения: “Если в фокус параболы

(параболического зеркала) поместить источник света, то лучи света отразившись от параболы, идут параллельно ее оси симметрии”. Модель должна выглядеть приблизительно так, как это показано на рис. 46.

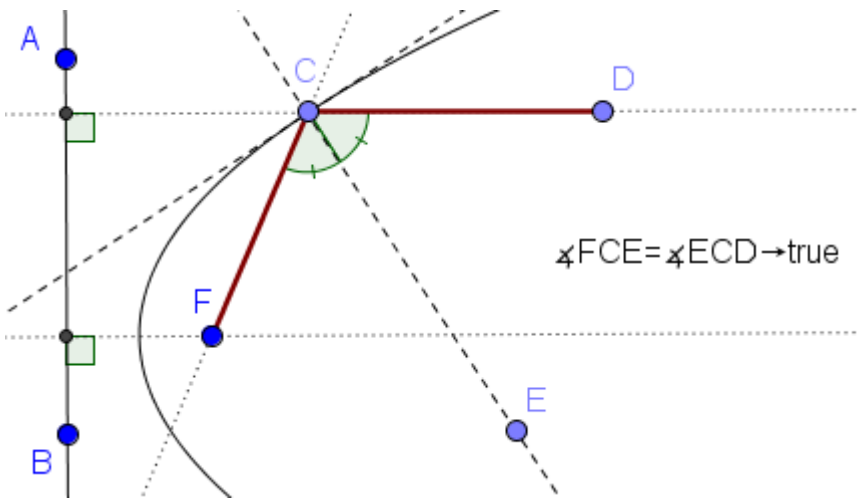


Рис. 46. Динамическая модель для проверки оптического свойства параболы



Угол заданной величины. Инструмент “*Угол заданной величины*” используется для формирования угла заданного размера. Действует инструмент так. Производятся щелчки мышью по двум точкам или позициям полотна, и в окно ввода появившейся панели вводится размер создаваемого угла. Если щелчки производились по позициям, то на их месте появляются точки. Вершиной угла становится вторая из точек. Стороны угла автоматически не формируются, а указываются лишь их направления. Углы в поле ввода можно задавать и в градусах, и в радианах. По умолчанию значение задается в градусах, что отражается добавлением к нему символа “°”. С клавиатуры символ “°” можно вводить ключом *Alt+O*. Если этот символ убрать, то значение будет восприниматься в радианах. По умолчанию угол отсчитывается в направлении “против часовой стрелки” (*counter clockwise*), но прямо на панели ввода его можно изменить на “по часовой стрелке” (*clockwise*).

A. <i>Angle[conic]</i> B. <i>Angle[vector]</i> C. <i>Angle[point]</i>	D. <i>Angle[number]</i> E. <i>Angle[polygon]</i>	F. <i>Angle[vec1, vec2]</i> G. <i>Angle[line1, line2]</i>
---	---	--

Вывод углов

А) По команде *A* выводится метка угла между положительным направлением оси абсцисс и главной осью конического сечения *conic* (эллипса, гиперболы, параболы). Аргумент может быть задан именем или уравнением коники. Метка появляется в центре (эллипс, гипербола) или в вершине (парабола).

В) По команде *B* выводится метка угла между положительным направлением оси абсцисс и вектором *vector*. Аргумент может быть задан именем или уравнением вектора. Метка появляется у основания вектора.

С) По команде *C* выводится метка угла между положительным направлением оси абсцисс и виртуальным вектором, исходящим из начала координат с концом в точке *point*. Точка может быть задана именем или в координатной форме. Метка появляется в начале координат.


Д) По команде *D* на панели “Объекты” формируется угол величиной *number*. Если установлен вывод в градусах, то *Angle[1]* возвращает 57.3° иначе – $1\ rad$. Метка угла не выводится.

Е) По команде *E* выводятся метки всех углов многоугольника *polygon*. Если многоугольник создавался с ориентацией против часовой стрелки, то метятся его внутренние углы, иначе – внешние углы. Исключением являются правильные многоугольники, у которых всегда метятся внутренние углы. Аргумент может быть задан именем многоугольника или функцией *Polygon[...]*. Метки углов появляются у соответствующих вершин многоугольника.

Ф) По команде *F* выводится метка угла, на который нужно повернуть в положительном направлении вектор *vec1*, чтобы его направление совпало с направлением вектора *vec2*. Метка появляется у основания вектора *vec1*.

Г) По команде *G* выводится метка угла между прямыми *line1* и *line2*. Если прямые пересекаются, то в точке их пересечения и появляется метка. Если прямые параллельны, то угол, равный 0 , создается без вывода метки. Формируемая метка зависит от ориентации прямых, то есть от того, как они создавались.



Расстояние или длина. Инструмент “ *Расстояние или длина*” позволяет определять кратчайшее расстояние в сантиметрах между двумя объектами, которые могут быть прямыми или лучами, а также между точкой и объектом, который может быть точкой, прямой или лучом. Кроме того, этот инструмент позволяет находить длину отрезка, периметр окружности, длину дуги окружности, периметр эллипса и периметр многоугольника. При этом во всех случаях формируется и выводится динамическая надпись с постоянной и переменной (вычисляемой) частями. В постоянной части записывается, о каком расстоянии, длине, периметре идет речь, а в переменной – каково текущее значение этого параметра. При изменении позиций объектов переменная часть надписи автоматически актуализируется.

A. <i>Distance[P, obj]</i> B. <i>Distance[line1, line2]</i>
--

Вычисление расстояний

А) По команде *A* на панели “Объекты” формируется кратчайшее расстояние между точкой *P* и объектом *obj*, который может быть точкой, отрезком, прямой, лучом, ломаной или функцией. При этом на полотно никакой надписи не выводится.

В) По команде *B* на панели “Объекты” формируется кратчайшее расстояние между двумя прямыми. На полотно никакой надписи не выводится.

A. <i>Length</i> [<i>vector</i>]	D. <i>Length</i> [<i>text</i>]	G. <i>Length</i> [<i>func</i> , <i>P</i> , <i>Q</i>]
B. <i>Length</i> [<i>P</i>]	E. <i>Length</i> [<i>arc</i>]	H. <i>Length</i> [<i>parcur</i> , <i>a</i> , <i>b</i>]
C. <i>Length</i> [<i>list</i>]	F. <i>Length</i> [<i>func</i> , <i>a</i> , <i>b</i>]	I. <i>Length</i> [<i>parcur</i> , <i>P</i> , <i>Q</i>]

Вычисление длин и периметров.

По командам *A-J* создаются объекты типа “Число” и никаких надписей на полотно не выводится.

A) По команде *A* формируется длина вектора *vector*.

B) По команде *B* формируется расстояние между началом координат и точкой *P*.

C) По команде *C* подсчитывается количество элементов списка *list* верхнего уровня вложенности.

D) По команде *D* подсчитывается количество символов текста *text*. Если аргумент *text* – это имя текстового объекта, то его надо записывать без кавычек. В противном случае кавычки обязательны.

E) По команде *E* формируется длина дуги окружности.


F) По команде *F* формируется длина кривой, заданной функцией *func* одной переменной *x* на отрезке: $a \leq x \leq b$ ($a < b$).

G) По команде *G* формируется длина кривой, заданной функцией *func* одной переменной *x* между точками *P* и *Q* ее графика. Фактически у точек *P* и *Q* используются лишь их абсциссы.

H) По команде *H* формируется длина параметрически заданной кривой *parcur* при изменении значения параметра *t* на отрезке: $a \leq t \leq b$ ($a < b$). Кривая может быть задана именем или функцией *Curve*.

I) По команде *I* формируется длина параметрически заданной кривой *parcur* между точками *P* и *Q* ее графика. Точки *P* и *Q* должны размещаться на графике. Кривая может быть задана именем или функцией *Curve*.



Площадь. Инструмент “ *Площадь*” позволяет определять площадь многоугольника, круга и эллипса. По нему формируется и выводится динамическая надпись с постоянной и переменной (вычисляемой) частями. В постоянной части сообщается, о площади какого объекта идет речь, а в переменной – каково текущее значение этой площади. При изменении объектов переменная часть автоматически актуализируется.

A. <i>Area</i> [<i>P</i> ₁ , <i>P</i> ₂ , ..., <i>P</i> _{<i>n</i>}]
B. <i>Area</i> [<i>poly</i>]
C. <i>Area</i> [<i>conic</i>]

Вычисление площади многоугольника, круга и эллипса.

По командам *A-C* создается объект типа “Число”, являющийся алгебраической площадью некоторого объекта. Никаких надписей на полотне не формируется.

A) По команде *A* подсчитывается площадь многоугольника. Аргументы *P*₁, *P*₂, ..., *P*_{*n*} являются точками, определяющими последовательные смежные

вершины многоугольника. Их можно задавать именами или в координатной форме.

В) По команде *B* подсчитывается площадь многоугольника *poly*, заданного именем или командой *Polygon*.

С) По команде *C* подсчитывается площадь круга или эллипса *conic*, заданного именем, командой *Circle[...]* или командой *Ellipse[...]*.

A. <i>IntegralBetween</i> [<i>f</i> , <i>g</i> , <i>a</i> , <i>b</i>]
B. <i>IntegralBetween</i> [<i>f</i> , <i>g</i> , <i>a</i> , <i>b</i> , <i>bool</i>]

Вычисление площади между графиками функций

А) По команде *A* на промежутке от *a* до *b* выводятся графики функций *f*(*x*) и *g*(*x*) одной переменной *x*, промежуток между ними на участке от *a* до *b* затеняется, вычисляется площадь между *f*(*x*) и *g*(*x*) и формируется и выводится динамическая надпись о площади типа “*имя переменной* = *значение площади*”. Фактически вычисляется алгебраическая площадь, то есть определенный интеграл $\int_a^b (f(x) - g(x)) dx$.

В) Команда *B* при *bool*=*true* выполняется точно так же, как *A*, а при *bool*=*false* затенение промежутка между *f* и *g* проводится, площадь не вычисляется, а надпись о ней приобретает форму “*имя переменной* = ?”.

Примеры 65. Вычислить площадь между кривыми, заданными функциями *cos*(*x*) и *sin*(*x*) на промежутке $[0, 2\pi]$, а также функциями $2+\cos(x)$ и $2+\sin(x)$ на промежутке $[0, \pi/4]$. Кроме того, вывести затененную фигуру между функциями $2+\cos(x)$ и $2+\sin(x)$ на промежутке $[\pi, 2\pi]$ без вычисления ее площади.

Решение. Все вычисления организуем на одном графике. Для этого через строку ввода введем три команды:

- 1) *IntegralBetween*[*cos*(*x*), *sin*(*x*), 0, $2*\pi$];
- 2) *IntegralBetween*[$2+\cos(x)$, $2+\sin(x)$, 0, $\pi/4$];
- 3) *IntegralBetween*[$2+\cos(x)$, $2+\sin(x)$, π , $2*\pi$, *false*].

Результат выполнения этих команд показан на рис. 47.

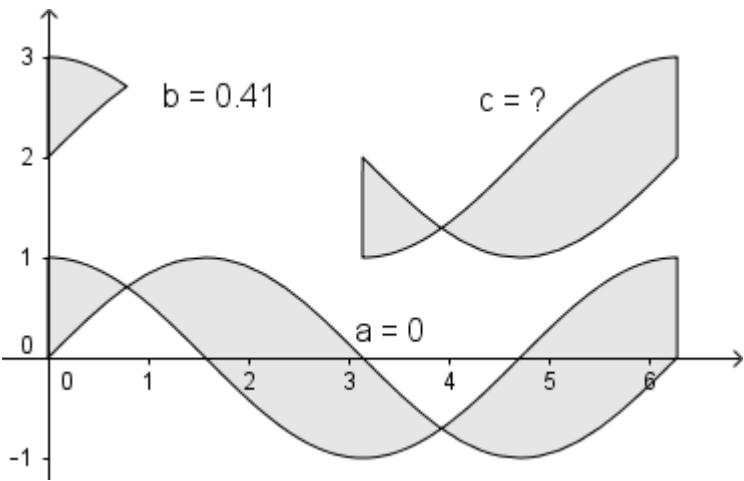



Рис. 47. Вычисление площадей между двумя кривыми



Наклон прямой. Инструмент “ **Наклон прямой**” возвращает переменную со значением тангенса угла наклона к оси абсцисс указанной прямой, отрезка или луча. Кроме того, на графике появляется затененный направляющий прямоугольный треугольник наклона с гипотенузой, исходящей из начальной точки определения исходной линии и идущей вдоль этой линии. Длины катетов этого треугольника равны 1 и модулю тангенса угла наклона. Около второго катета выводится динамическая надпись вида “*переменная=тангенс угла наклона*” (см. рис. 48). Через панель “*Настройки*” размер и стиль вывода направляющего треугольника могут быть изменены.

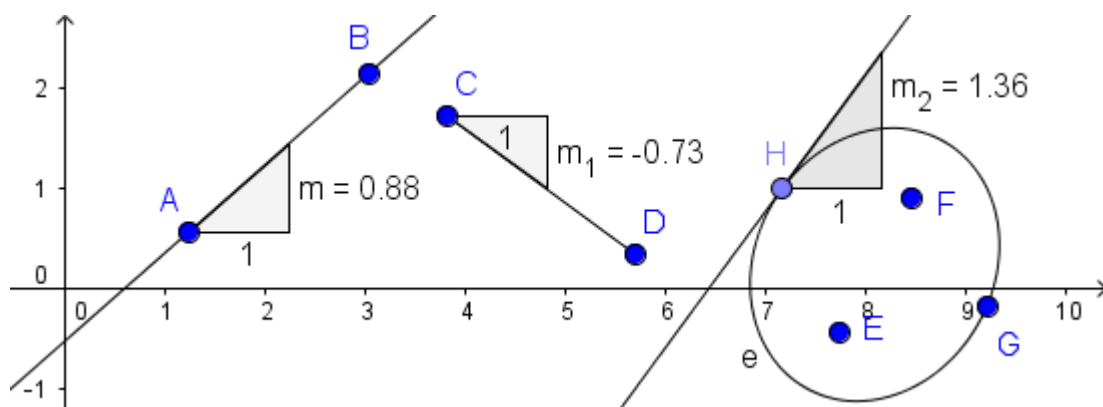




Рис. 48. Наклон прямой, отрезка и касательной к эллипсу

A. Slope[obj]

Вывод направляющего треугольника наклона

A) По команде A для прямой, отрезка или луча выводится прямоугольный треугольник наклона.




Создать список. Инструмент “ **Создать список**” используется для формирования списка точек вида $\{(a, b), (c, d), \dots\}$ из числа тех точек, которые уже имеются на панели “*Полотно*”. Выполняется данная операция так. При активном инструменте “” и нажатой левой кнопке мыши протяжкой курсора выделяется цветом некоторая прямоугольная область. При отпускании кнопки мыши все точки, оказавшиеся внутри выделенной области попадают в новый список. Не учитываются только точки, являющиеся вершинами объекта “*Многоугольник*”. Если в выделенном прямоугольнике точек нет, то пустой список не создается.

Заметим, что данный инструмент используется и в электронных таблицах для создания списков из точек, размещенных в ее ячейках.

3.3.10. Линейные и нелинейные преобразования объектов

Ниже рассмотрены средства для выполнения различных преобразований объектов встроенными инструментами и (или) командами. Преобразования, которые могут быть реализованы как инструментами, так и командами – это зеркальное отражение объекта относительно прямой (осевая симметрия), отражение объекта относительно точки (центральная симметрия), инверсия относительно окружности, поворот вокруг точки, параллельный перенос по вектору, гомотетия относительно точки. Кроме того, командами *Shear* и *Stretch*, не имеющими инструментальных аналогов, можно совершать разнообразные сдвиги объектов вдоль и поперек направлений.



Отражение относительно прямой. Инструмент “ Отражение относительно прямой” используется для построения объекта B , являющегося зеркальным отражением объекта A относительно некоторой прямой m . Запускается процесс создания B щелчком сначала по объекту A , а затем по прямой m . При отражении относительно прямой по каждой точке $P \in A$ формируется точка $Q \in B$ такая, что, во-первых, Q расположена на перпендикулярной прямой, опущенной из P на m , во-вторых, P и Q находятся на одинаковом расстоянии от m и, в-третьих, $Q=P$ тогда и только тогда, когда P лежит на m . Специфично работает рассматриваемый инструмент только с объектами-векторами, а именно, вектор сначала отражается относительно прямой, но затем результат параллельно переносится в начало исходного вектора. В качестве A могут выступать практически любые объекты, m может быть не только прямой, но и лучом или отрезком.

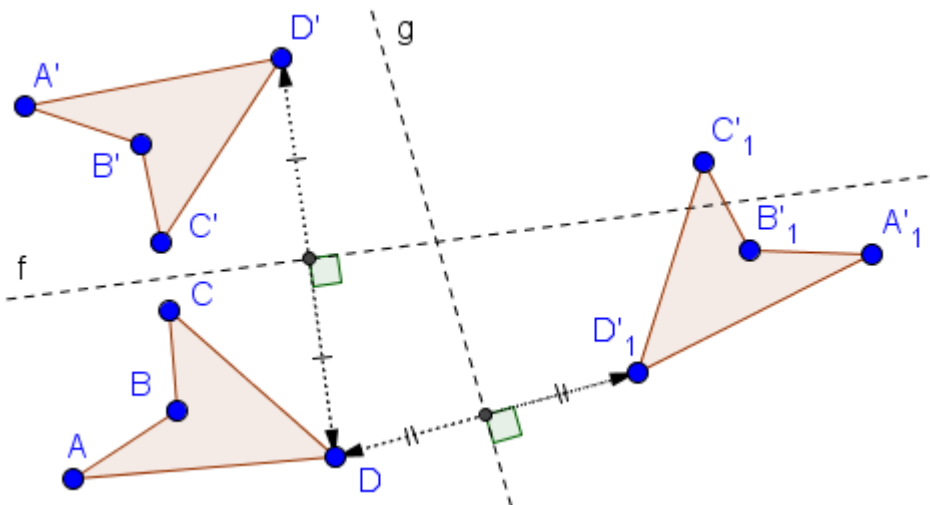


Рис. 49. Отражение четырехугольника $ABCD$ относительно прямых f и g

Рассмотренное преобразование A в B обладает симметрией, в том смысле, что оно переводит B в A . Поэтому его называют также осевой симметрией, а прямую m – осью симметрии.

На рис. 49 показано отражение четырехугольника $ABCD$ относительно прямых f и g . Отдельно выделено отражение вершины D от этих же прямых.


В общем случае отражение точки $P(x,y)$ в точку Q относительно прямой, заданной уравнением $a \cdot x + b \cdot y + c = 0$, аналитически описывается так:


$$P(x,y) \rightarrow Q\left(x - \frac{2a(ax + by + c)}{a^2 + b^2}, y - \frac{2b(ax + by + c)}{a^2 + b^2}\right).$$


$A. \text{Reflect}[obj, line]$

Отражение относительно прямой

A) По команде A реализуется отражение объекта obj относительно прямой $line$.



Отражение относительно точки. Инструмент “ Отражение относительно точки” используется для построения объекта B , являющегося отражением объекта A относительно некоторой точки O . Запускается процесс создания B щелчком сначала по объекту A , а затем по точке O . В качестве A могут выступать практически любые объекты. При отражении относительно точки O по каждой точке $P \in A$ формируется точка $Q \in B$ такая, что, во-первых Q расположена на прямой PO , во-вторых, P и Q находятся на одинаковом расстоянии от O и, в-третьих, $Q=P$ тогда и только тогда, когда $P=O$. Специфично работает рассматриваемый инструмент только с объектами-векторами, а именно, вектор сначала отражается относительно точки P , но затем результат параллельно переносится в начало исходного вектора. Таким образом A и B оказываются исходящими из одной точки, но противоположно направленными.

Рассмотренное преобразование A в B обладает симметрией, в том смысле, что оно же преобразует и B в A . Точку O называют центром симметрии. То есть фактически инструмент “ Отражение относительно точки” реализует преобразование, называемое в геометрии центральной симметрией. Заметим, что центральная симметрия является частным случаем преобразования поворота, то есть это поворот относительно центра O на 180° .

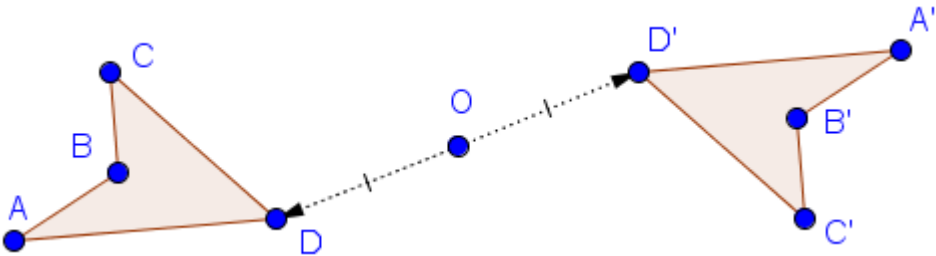


Рис. 50. Отражение четырехугольника $ABCD$ относительно прямых f и g


На рис. 50 показано отражение четырехугольника $ABCD$ относительно точки O . Отдельно выделено отражение вершины D относительно O .


В общем случае отражение точки $P(x, y)$ в точку Q относительно точки $O=(x_0, y_0)$ аналитически описывается так:

$$P(x, y) \rightarrow Q(2x_0 - x, 2y_0 - y).$$

A. Reflect[obj, point]


Отражение относительно точки
 А) По команде A реализуется отражение объекта obj относительно точки $point$.



Отражение относительно окружности. Инструмент “ *Отражение относительно окружности*” используется для построения по данному объекту A так называемого инверсного ему объекта B по отношению к некоторой фиксированной окружности. При этом говорят, что объект B получен из объекта A преобразованием инверсии относительно этой окружности. Реализуется преобразование при щелчке сначала по объекту A , а затем по окружности. Пусть C – окружность с центром в точке O радиуса R . При инверсии A в B относительно C каждая точка $P \in A$ ($P \neq O$) преобразуется в точку $Q \in B$ следующим образом. Во-первых, Q лежит на прямой PO и, во-вторых, выполняется соотношение $PO \cdot QO = R^2$. При $P=O$ считается, что Q – бесконечно удаленная точка. Точку O называют полюсом или центром инверсии. Если точка Q инверсная для P , то P инверсная для Q . Точки, инверсные друг другу, называют взаимно обратными точками.

Преобразование инверсии обладает такими свойствами:

- внутренние точки для инверсной окружности при инверсии переходят во внешние точки для этой окружности и наоборот;
- прямая, проходящая через полюс, при преобразовании инверсии переходит сама в себя;
- прямая, не проходящая через полюс, при преобразовании инверсии переходит в окружность, проходящую через полюс;
- окружность, проходящая через полюс, при преобразовании инверсии переходит в прямую, не проходящую через полюс;
- окружность, не проходящая через полюс, при преобразовании инверсии переходит в окружность, не проходящую через полюс;
- инверсия является конформным отображением, то есть сохраняет углы между кривыми.

Покажем, как по точке P без использования инструмента “” построить инверсную ей точку Q относительно заданной окружности с центром в точке O и радиуса R . Рассмотрим возможные случаи.

1) $PO > R$, то есть точка P находится вне инверсной окружности. Проведем отрезок PO и из P касательную к окружности (см. рис. 51). Из точки касания

D опустим перпендикуляр на OP и пусть Q – точка пересечения этого перпендикуляра и OP . Тогда Q и является инверсной для P точкой, что вытекает из следующей цепочки отношений:

$$\triangle OQD \sim \triangle ODP \Rightarrow \frac{OQ}{R} = \frac{R}{OP} \Rightarrow OP \cdot OQ = R^2.$$

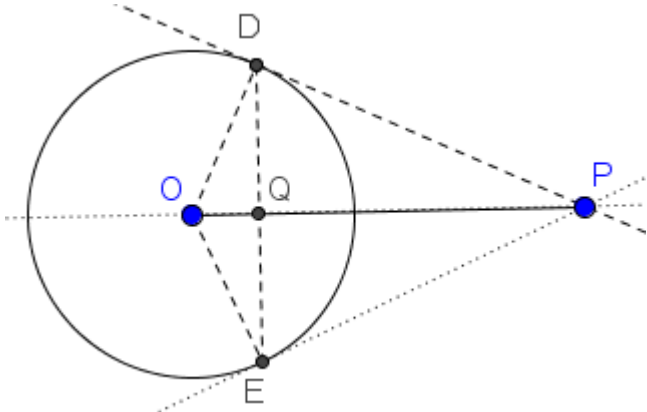



Рис. 51. Построение инверсной для P точки Q

2) $PO < R$, то есть P находится внутри инверсной окружности. Через точки O и P проведем прямую и из точки P восстановим к ней перпендикуляр. В точке D – пересечения этого перпендикуляра с инверсной окружностью проведем к ней касательную. Тогда точка Q – пересечения касательной с продолжением OP , и является инверсной для P точкой.

3) $PO = R$, то есть P находится на инверсной окружности. Тогда $Q = P$.

Примеры 66. Здесь демонстрируется формирование инверсий некоторых объектов относительно окружностей, показанных на соответствующих изображениях. Все инверсии построены инструментом “ *Отражение относительно окружности*”. На рис. 52 кардиоида сформирована как инверсия параболы, созданной по фокусу F и директрисе d . На рис. 53 приведены примеры инверсии квадрата. На рис. 54 представлена инверсия графика функции $\sin(x)$.

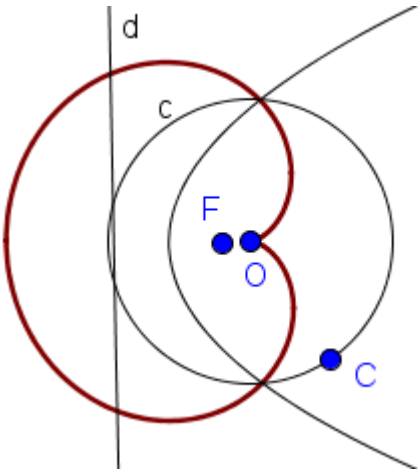


Рис. 52. Кардиоида – инверсия параболы относительно окружности

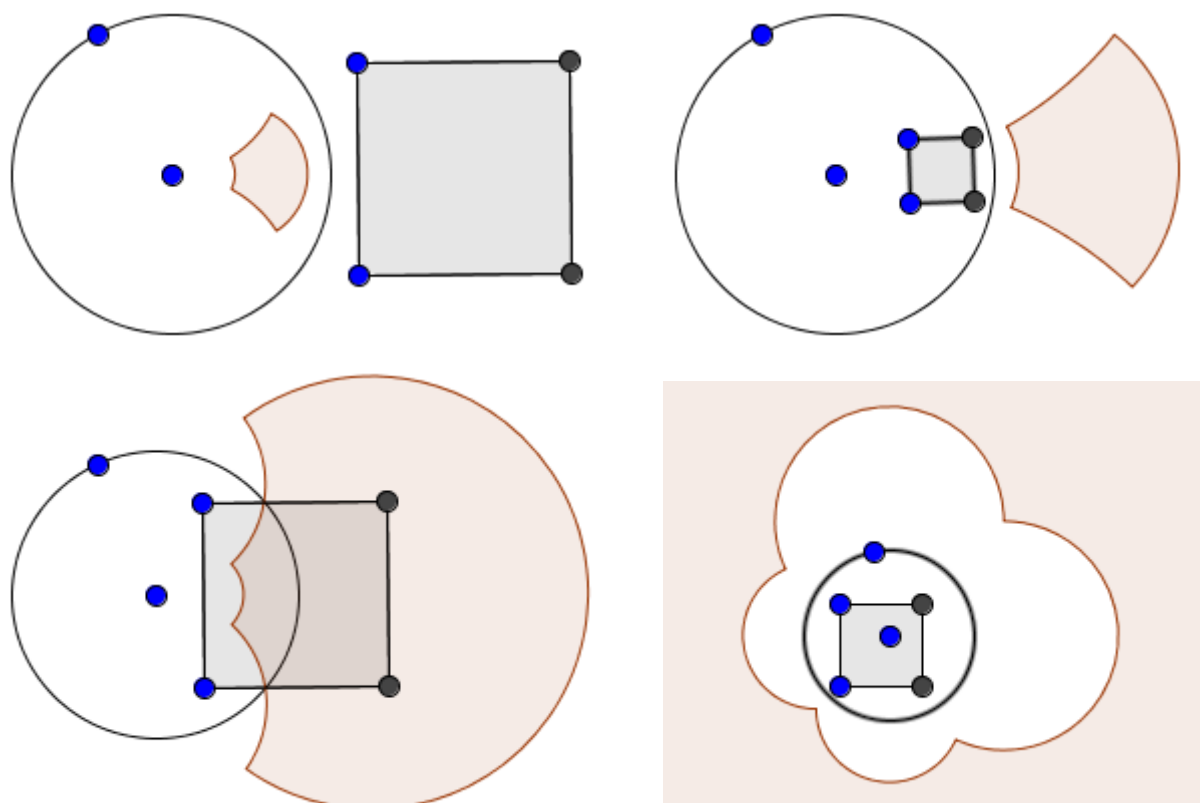


Рис. 53. Примеры инверсии квадрата относительно заданной окружности

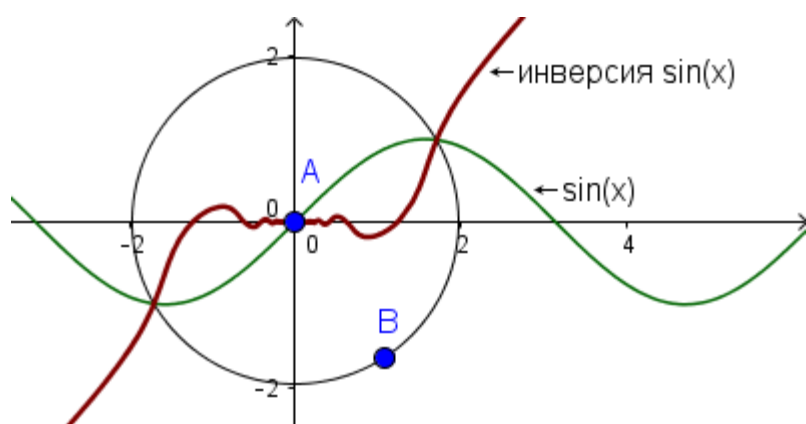


Рис. 54. Инверсия графика функции $\sin(x)$ относительно окружности

В общем случае инверсия относительно окружности с центром в точке $O=(x_0, y_0)$ и радиусом R аналитически описывается так:


$$P(x, y) \rightarrow Q \left(x_0 + \frac{R^2(x - x_0)}{(x - x_0)^2 + (y - y_0)^2}, y_0 + \frac{R^2(y - y_0)}{(x - x_0)^2 + (y - y_0)^2} \right).$$

A. Reflect[obj, circle]

Инверсия относительно окружности

А) По команде А реализуется инверсия объекта *obj* относительно окружности *circle*.



Поворот вокруг точки. Инструмент “ **Поворот вокруг точки**” используется для построения копии объекта, повернутого вокруг некоторой точки на заданный угол. При этом указанные точку и угол называют соответственно центром поворота и углом поворота. При выполнении операции сначала надо щелкнуть левой кнопкой мыши по объекту, затем по центру поворота, и, наконец, ввести в поле ввода появившейся панели требуемый угол поворота. По умолчанию поворот реализуется против часовой стрелки, хотя прямо при задании угла радиокнопкой можно переключиться на поворот по часовой стрелке. При угле поворота, равном 180° , преобразование поворота превращается в центральную симметрию.

На рис. 55 показан поворот четырехугольника $ABCD$ относительно точки O на угол α . Отдельно выделен поворот вершины D .

В общем случае поворот точки $P(x, y)$ вокруг точки $O=(x_0, y_0)$ на угол α аналитически описывается так:

$$P(x, y) \rightarrow Q((x - x_0) \cos \alpha - (y - y_0) \sin \alpha + x_0, (x - x_0) \sin \alpha + (y - y_0) \cos \alpha + y_0).$$

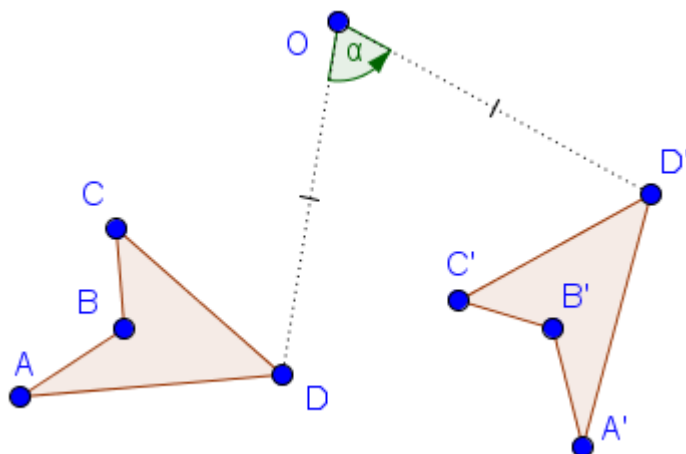


Рис. 55. Поворот четырехугольника $ABCD$ относительно точки O на угол α

A. $Rotate[obj, \alpha]$


B. $Rotate[obj, \alpha, O]$

Поворот (вращение) объекта

A) По команде A реализуется поворот объекта obj относительно начала координат на угол α .

B) По команде B реализуется поворот объекта obj относительно точки O на угол α .



Параллельный перенос по вектору. Инструмент “ **Параллельный перенос по вектору**” используется для построения копии объекта путем его параллельного переноса на данный вектор, то есть на заданную величину по заданному направлению. При выполнении операции сначала надо щелкнуть левой кнопкой мыши по объекту, а затем по вектору, на которой будет осуществлен параллельный перенос. Возможен и другой вариант выполнения операции –

как и ранее сначала щелкнуть левой кнопкой мыши по объекту, затем по точке или позиции и, наконец, снова по точке или позиции. При щелчках по позициям на их месте появляются точки. Между двумя точками появляется новый вектор с началом в первой точке и концом во второй точке. Этот вектор и используется для параллельного переноса исходного объекта.

На рис. 56 показан перенос четырехугольника $ABCD$ по векторам \overrightarrow{IJ} и \overrightarrow{KL} . Отдельно выделен перенос по этим векторам вершины D .

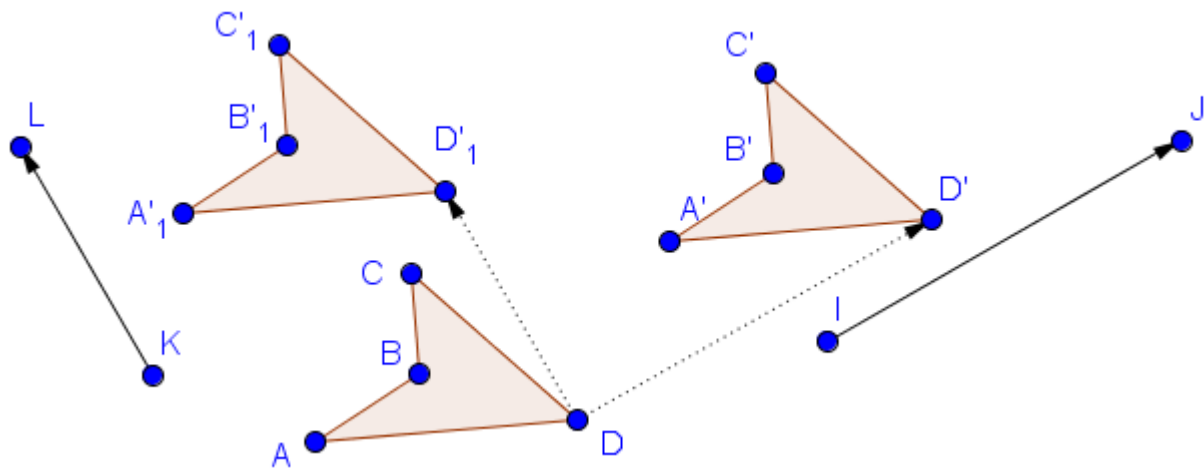


Рис. 56. Параллельный перенос четырехугольника $ABCD$ по векторам \overrightarrow{IJ} и \overrightarrow{KL}

В общем случае параллельный перенос точки $P(x, y)$ в точку Q по заданному вектору v аналитически описывается так: $\overrightarrow{PQ} = v$. Если у вектора v начальная и конечная точки – $I(x_1, y_1)$ и $J(x_2, y_2)$, то это означает, что:

$$P(x, y) \rightarrow Q(x + x_2 - x_1, y + y_2 - y_1).$$

A. <code>Translate[obj, vec]</code>
B. <code>Translate[vec, P]</code>

Параллельный перенос по вектору
 А) По команде *A* реализуется параллельный перенос объекта *obj* по вектору *vec*.

В) По команде *B* реализуется параллельный перенос вектора *vec* в точку *P*, то есть начало *vec* совмещается с *P*.



Гомотетия относительно точки. Инструмент “Гомотетия относительно точки” используется для построения по некоторому объекту A подобного ему объекта B с заданным коэффициентом подобия с размещением его в определенной позиции относительно указанной точки (центра) F . При выполнении операции сначала надо щелкнуть левой кнопкой мыши по объекту, а затем по точке или позиции, и, наконец, ввести в поле ввода появившейся панели коэффициент подобия. Если щелчок производился по позиции, то на этом месте появляется точка. Если O – начало координат, то гомотетия с центром F и коэффициентом подобия k преобразует точки $P \in A$ в точки $Q \in B$ так, что

$\overrightarrow{OQ} = k \cdot \overrightarrow{OP} - \overrightarrow{OF}$ ($k \neq 0$). Заметим, что $|k|$ влияет только на размер полученного объекта; знак k , позиции центра и исходного объекта – на позицию полученного объекта. Гомотетия с $k=-1$ является центральной симметрией. На рис. 57 приведен пример гомотетии с коэффициентом 2 и центром F .

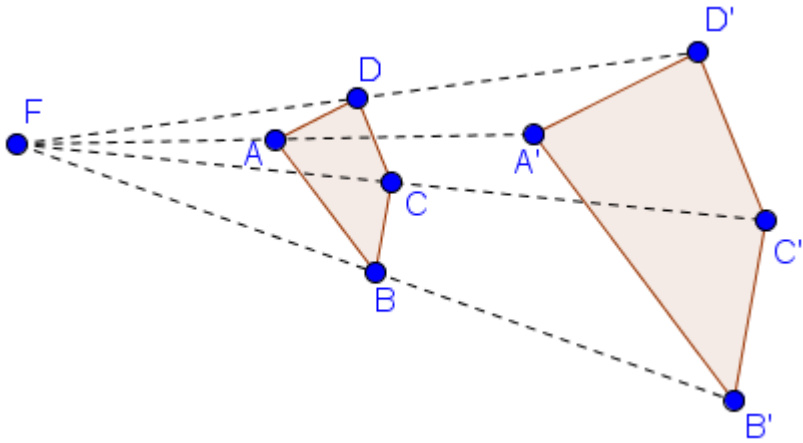


Рис. 57. Гомотетия четырехугольника с коэффициентом 2 и центром F

В общем случае гомотетия с центром в точке $F(x_0, y_0)$ и коэффициентом подобия k переводит точку $P(x, y)$ в точку Q так:

$$P(x, y) \rightarrow Q(k \cdot x - x_0, k \cdot y - y_0).$$

<div data-bbox="138 1167 425 1252" data-label="List-Group"> <ul style="list-style-type: none"> A. <i>Dilate</i>[<i>obj</i>, <i>k</i>] B. <i>Dilate</i>[<i>obj</i>, <i>k</i>, <i>P</i>] </div>	<div data-bbox="479 1149 1023 1187" data-label="Section-Header"> <p><i>Гомотетия относительно точки</i></p> </div> <div data-bbox="479 1193 1369 1274" data-label="Text"> <p>A) По команде <i>A</i> реализуется гомотетия объекта <i>obj</i> с коэффициентом подобия k относительно начала координат.</p> </div> <div data-bbox="114 1321 1369 1402" data-label="Text"> <p>B) По команде <i>B</i> реализуется гомотетия объекта <i>obj</i> с коэффициентом подобия k относительно точки <i>P</i>.</p> </div>
---	--

<div data-bbox="138 1467 470 1597" data-label="List-Group"> <ul style="list-style-type: none"> A. <i>Shear</i>[<i>obj</i>, <i>line</i>, <i>k</i>] B. <i>Stretch</i>[<i>obj</i>, <i>line</i>, <i>k</i>] C. <i>Stretch</i>[<i>obj</i>, <i>vec</i>] </div>	<div data-bbox="538 1449 1118 1487" data-label="Section-Header"> <p><i>Сдвиги вдоль и поперек направления</i></p> </div> <div data-bbox="538 1494 1369 1617" data-label="Text"> <p>Здесь рассматриваются команды сдвигов объектов вдоль линии, поперек линии и по заданному вектору. Аналоги соответствующих инструментов для этих команд отсутствуют.</p> </div>
--	---

A) Команда *A* реализует преобразование сдвига (*скоса*) объекта *obj* вдоль прямой *line* с коэффициентом k ($k \neq 0$). Знак k задает направление перемещения. Выполняется данная операция следующим образом. Все точки $P \in obj$, расположенные на прямой *line*, остаются без изменений. Каждая точка $P \in obj$, не лежащая на прямой *line*, сдвигается параллельно *line* на величину, пропорциональную расстоянию от *P* до *line* с коэффициентом пропорциональности k . Сдвиги в разных полуплоскостях от *line* происходят в противоположных направлениях. Фактически сдвигается копия объекта *obj*, а сам объект остается

без изменений. Отметим, что при сдвигах типа *shear* площади объектов не изменяются.

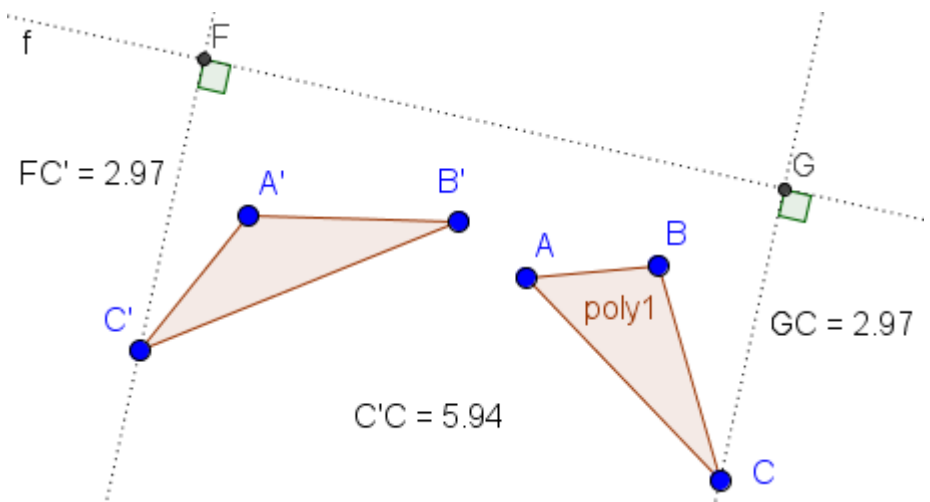


Рис. 58. Сдвиг $\triangle ABC$ вдоль линии f с коэффициентом 2

В общем случае понять, как вдоль прямой m с коэффициентом k выполняется сдвиг некоторой точки $P(x_0, y_0)$ в точку $Q(x_1, y_1)$, можно из рис. 59. На нем демонстрируется не только преобразование *shear* конкретной точки $P \rightarrow Q$, но и для двух возможных вариантов расположения прямой m показаны все случаи направлений перемещения точки при различных знаках k .

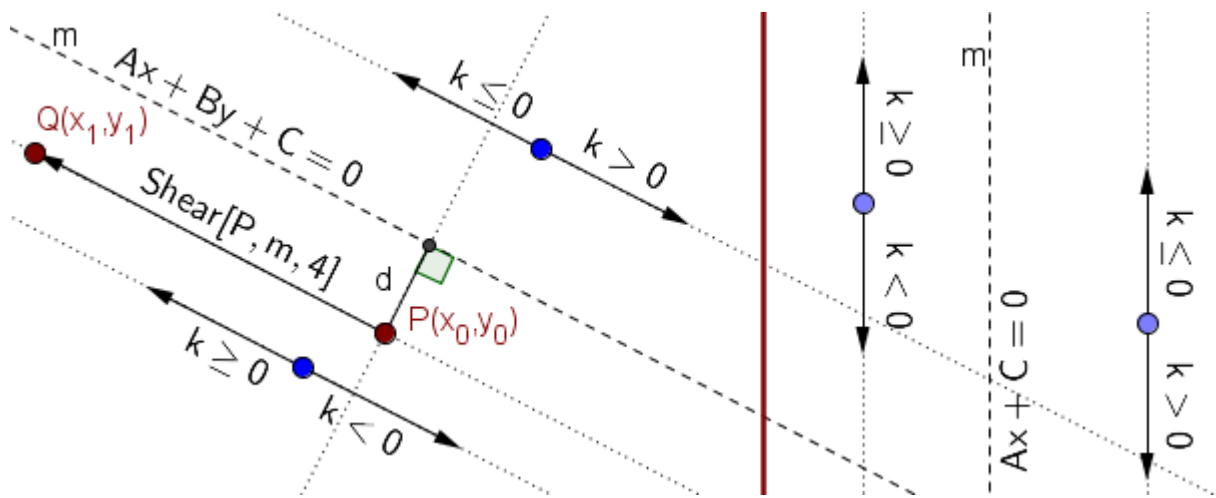


Рис. 59. Сдвиг конкретной точки $P(x_0, y_0) \rightarrow Q(x_1, y_1)$ вдоль прямой m и влияние знака k на направление сдвига

Покажем, что при сдвиге $P(x_0, y_0) \rightarrow Q(x_1, y_1)$ вдоль прямой m ($Ax + By + C = 0$) с коэффициентом k координаты x_1 и y_1 точки Q можно определять по формулам:

$$\begin{cases} x_1 = x_0 + t \cdot B \\ y_1 = y_0 - t \cdot A \end{cases}, \quad \text{где } t = \frac{k \cdot (Ax_0 + By_0 + C)}{A^2 + B^2}. \tag{9}$$

Рассмотрим общий случай, когда $B \neq 0$, то есть m не перпендикулярна оси абсцисс. Из того, что длина отрезка PQ равна $|k| \cdot d$ и точка Q лежит на прямой, параллельной m , получаем:

$$\begin{cases} (x_1 - x_0)^2 + (y_1 - y_0)^2 = k^2 d^2, \\ y_1 - y_0 = -\frac{A}{B}(x_1 - x_0). \end{cases}$$

Решая эту систему уравнений относительно x_1 и y_1 , получим

$$\begin{cases} x_1 = x_0 \pm t \cdot B \\ y_1 = y_0 \mp t \cdot A \end{cases}, \quad \text{где } t = \frac{k \cdot (Ax_0 + By_0 + C)}{A^2 + B^2}, \quad (10)$$

что соответствует заданию двух возможных направлений на прямой m и, следовательно, двум возможным вариантам реализации *Shear*. В *GeoGebra* осуществлена реализация (9).

Остается рассмотреть случай $B=0$. Поскольку при $B=0$ прямая m перпендикулярна оси абсцисс, то:

$$\begin{cases} x_1 = x_0 \\ y_1 = y_0 \pm kd \end{cases} \Rightarrow \begin{cases} x_1 = x_0 \\ y_1 = y_0 \pm k \left(x_0 - \left(-\frac{C}{A} \right) \right) \end{cases} \Rightarrow \begin{cases} x_1 = x_0 \\ y_1 = y_0 \pm k \left(x_0 + \frac{C}{A} \right) \end{cases}.$$

Таким образом снова получены соотношения (10), но при $B=0$. Как и при $B \neq 0$, в данном случае из двух возможностей вычисления y_1 в *Geogebra* реализован вариант со знаком “-” после y_0 и, тем самым, формулы (9) установлены.

B) Команда B реализует преобразование сдвига (*stretch*) объекта obj поперек прямой $line$ с коэффициентом k ($k \neq 0$). Знак k задает направление перемещения, при этом объекты с k и $-k$ являются зеркальным отражением друг друга относительно $line$. Выполняется данная операция следующим образом. Все точки $P \in obj$, расположенные на прямой $line$, остаются без изменений. Каждая точка $P \in obj$, не лежащая на прямой $line$, сдвигается перпендикулярно $line$ на величину, пропорциональную расстоянию от P до $line$ с коэффициентом пропорциональности k . Сдвиги в разных полуплоскостях от $line$ происходят в противоположных направлениях. Фактически сдвигается копия объекта obj , а сам объект остается без изменений.

На рис. 60 показаны сдвиги $\triangle ABC$ ($poly1$) и $\triangle FIH$ ($poly2$) поперек линии f с коэффициентом 2, осуществленные соответственно командами $Stretch[poly1, f, 2]$ и $Stretch[poly2, f, 2]$. На изображении можно видеть пояснения к тому, как осуществлен сдвиг конкретной точки C ($C \rightarrow C'$).

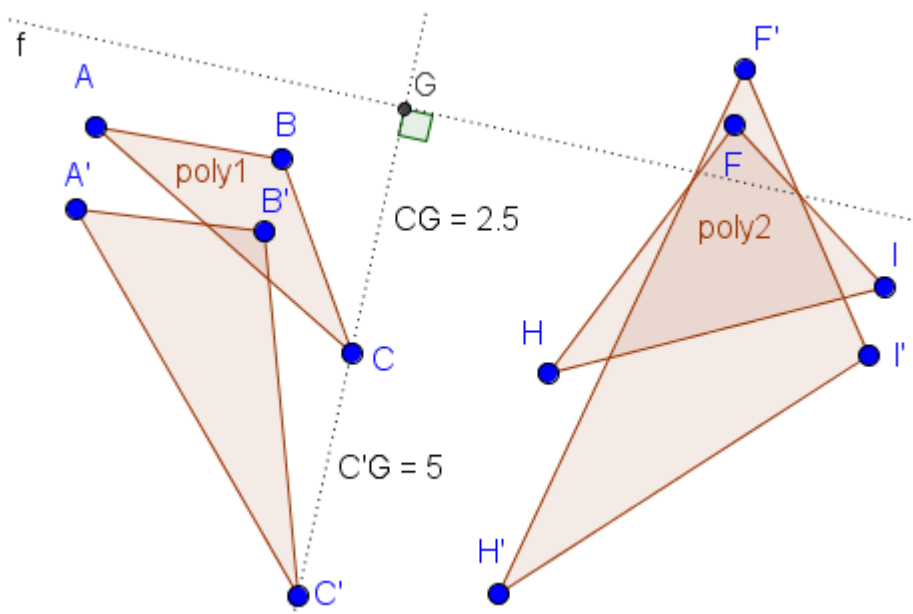


Рис. 60. Сдвиг $\triangle ABC$ и $\triangle FIH$ перпендикулярно линии f с коэффициентом 2

Покажем, что при сдвиге $P(x_0, y_0) \rightarrow Q(x_1, y_1)$ поперек прямой m ($Ax + By + C = 0$) с коэффициентом k координаты x_1 и y_1 точки Q можно определять по формулам:

$$\begin{cases} x_1 = x_0 + t \cdot A \\ y_1 = y_0 + t \cdot B \end{cases}, \quad \text{где } t = \frac{(k-1) \cdot (Ax_0 + By_0 + C)}{A^2 + B^2}. \quad (11)$$

Рассмотрим общий случай, когда $A \neq 0$, то есть m не параллельна оси абсцисс. Из того, что длина отрезка PQ равна $|k-1| \cdot d$ и точка Q лежит на прямой, перпендикулярной m , получаем:

$$\begin{cases} (x_1 - x_0)^2 + (y_1 - y_0)^2 = (k-1)^2 d^2, \\ y_1 - y_0 = \frac{B}{A}(x_1 - x_0). \end{cases}$$

Решая эту систему уравнений относительно x_1 и y_1 , получим

$$\begin{cases} x_1 = x_0 \pm t \cdot A \\ y_1 = y_0 \pm t \cdot B \end{cases}, \quad \text{где } t = \frac{(k-1) \cdot (Ax_0 + By_0 + C)}{A^2 + B^2}, \quad (12)$$





что соответствует заданию двух возможных направлений на прямой m и, следовательно, двум возможным вариантам реализации *Stretch*. В *GeoGebra* осуществлена реализация (11).

Остается рассмотреть случай $A=0$. Поскольку при $A=0$ прямая m параллельна оси абсцисс, то:

$$\begin{cases} x_1 = x_0 + (k-1)d \\ y_1 = y_0 \end{cases} \Rightarrow \begin{cases} x_1 = x_0 \pm (k-1) \left(x_0 + \frac{C}{B} \right) \\ y_1 = y_0 \end{cases}.$$

Таким образом снова получены соотношения (11), но при $A=0$. Как и при $A \neq 0$ в данном случае из двух возможностей вычисления x_1 в *GeoGebra* реализован вариант со знаком “+” после x_0 и, тем самым, формулы (11) установлены.


Для большей убедительности проделанных выкладок, создадим динамическую модель для экспериментальной проверки справедливости формул (11). Сделать это можно следующей последовательностью действий:

- инструментом “ Прямая” проведем прямую через две точки и через контекстное меню изменим ее имя на m , а имена точек – на R и S соответственно. Пока будем считать, что прямая m не перпендикулярна оси абсцисс;
- через строку ввода определим коэффициенты прямой m , введя три соотношения $A=x[m]$, $B=y[m]$, $C=z[m]$. Определим также переменную $k=0.4$ и, включив на панели “Объекты” соответствующую радиокнопку, превратим ее в бегунок;
- инструментом “ Точка” сформируем точку D вне m . Переименуем D на P и через панель “Настройки” зададим для нее заголовок $P(x_0, y_0)$ ($P(x_0, y_0)$), а также установим вывод заголовка вместо P ;
- инструментом “ Перпендикулярная прямая” проведем прямую, перпендикулярную m и проходящую через точку P ;
- инструментом “ Пересечение” сформируем точку D пересечения m и проведенного перпендикуляра;
- через строку ввода вычислим вспомогательную переменную t и по ней точку Q :

$$t=(k-1)*(A*x[P]+B*y[P]+C)/(A^2+B^2);$$

$$Q=(x[P]+t*A, y[P]+t*B).$$

Через панель “Настройки” для точки Q сформируем заголовок $Q(x_1, y_1)$ ($Q(x_1, y_1)$) и установим его вывод вместо имени Q ;

- инструментом “ Текст” в *LaTeX*-режиме создадим проверочную надпись, введя выражение:

$$t=\frac{(k-1)(Ax_0+By_0+C)}{A^2+B^2}= \frac{(k-1)(A*x(P)+B*y(P)+C)/(A^2+B^2)}{x[Q]==x[P]+t*A},$$

$$x_1=x_0+t*A \rightarrow \frac{y[Q]==x[P]+t*B}{y_1=y_0+t*B}.$$

Первая строка проверочной надписи являются информационной и демонстрирует не только значение t , но и по какой формуле проводятся вычисления. Правые части двух следующих строк вычисляются заново при изменении значений бегунка, перемещении свободных точек R и S по полотну, а также точки P по прямой, перпендикулярной m . Однако во всех ситуациях мы получаем одно и то же значения $true$, что и является экспериментальным подтверждением справедливости формул (11).

Построенная модель для проведения экспериментов представлена на рис. 61. Несмотря на то, что создавалась модель для случая, когда m не перпендикулярна оси абсцисс, ее можно использовать и для этого случая. На рис. 62 показан сдвиг по вектору u $\triangle CDE$ ($poly1$).

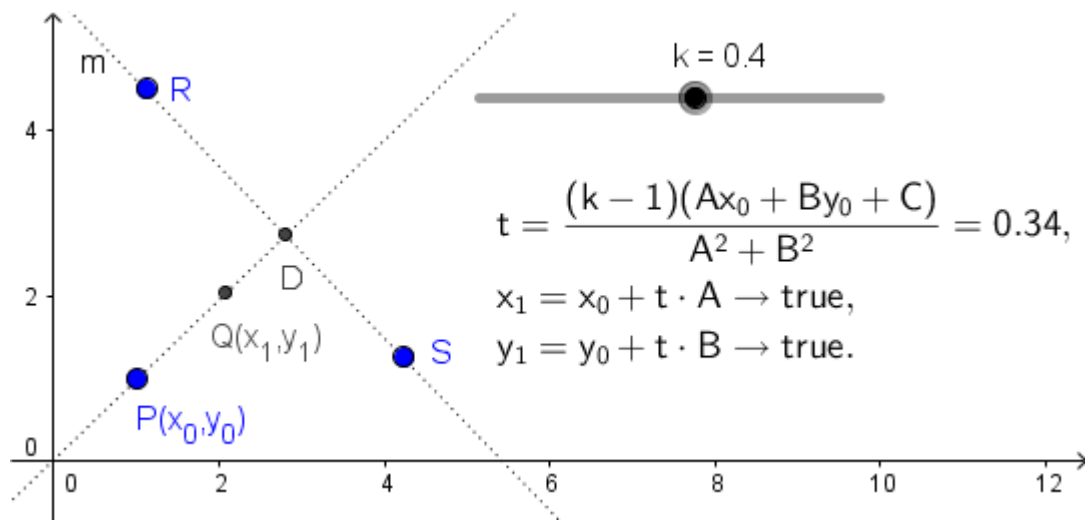


Рис. 61. Динамическая модель для экспериментальной проверки формул (11)

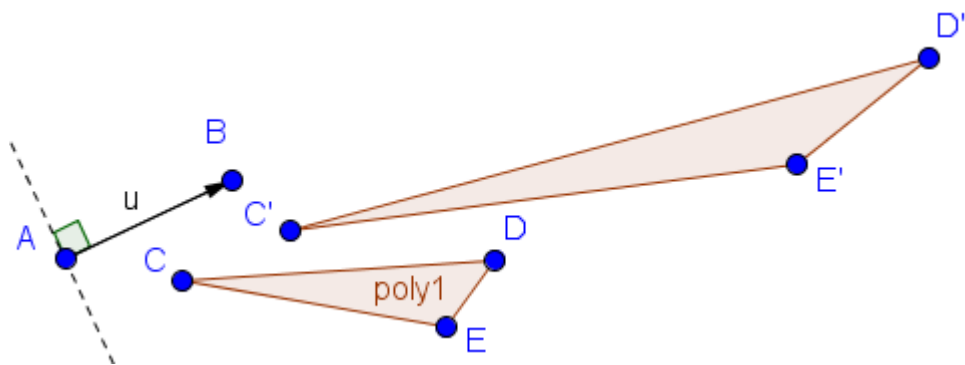


Рис. 62. Сдвиг $\triangle CDE$ параллельно вектору u

С) Команда C реализует преобразование сдвига ($stretch$) объекта obj параллельно ненулевому вектору vec . Это означает следующее. Сдвиг происходит с коэффициентом k , равным длине вектора vec , поперек прямой f , перпендикулярной вектору vec и проходящей через его начало. То есть команда C выполняется точно так же, как и команда $Stretch[obj, f, Length[vec]]$.

На рис. 63 показаны сдвиги изображения смайлика с именем $pic1$, осуществленные командами: $Shear[pic1, f, 1.5]$ – поперек линии f с коэффициентом

3.5, $Stretch[pic1, f, 3.5]$ – вдоль линии f с коэффициентом 3.5 и $Stretch[pic1, u]$ – параллельно вектору u . В последнем случае команды $Stretch[pic1, u]$ и $Stretch[pic1, f, Length[u]]$ равносильны, где f – прямая, перпендикулярная вектору u и проходящая через его начало. Предварительно изображение смайлика было загружено на полотно из заранее заготовленного файла *smiley.png* с помощью инструмента “Изображение”.

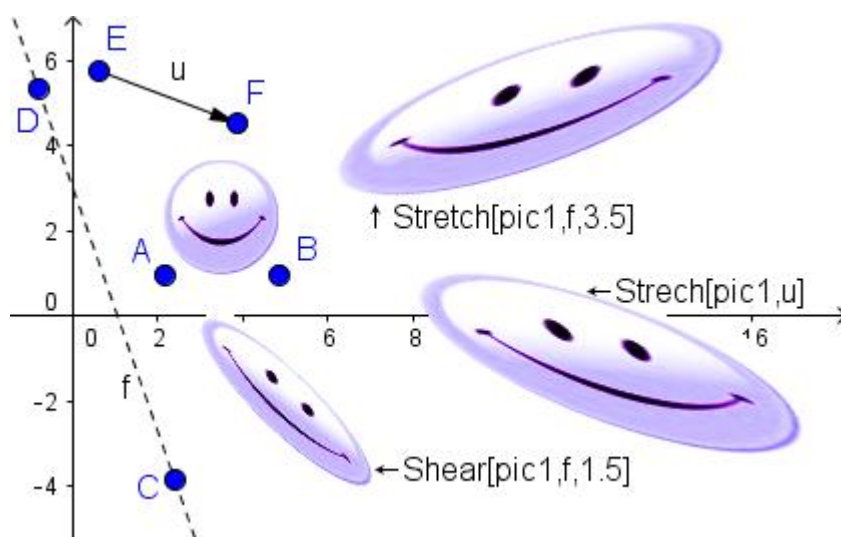


Рис. 63. Сдвиг изображения поперек линии f с коэффициентом 3.5 ($Stretch$), вдоль линии f с коэффициентом 1.5 ($Shear$) и по вектору u ($Stretch$)


3.3.11. Тексты и рисование

В этом пункте обсуждаются средства, которые позволяют выводить тексты, формулы и изображения; рисовать от руки, создавая временные презентационные объекты в виде поясняющих надписей и акцентирующих внимание фигур; вводить объекты *GeoGebra*, рисуя от руки их приблизительную форму; получать разнообразную информацию о функциях одной переменной, а также рассматривается вопрос об отношении пары объектов.

ABC Текст. Для работы инструментом “ABC Текст” необходимо щелкнуть по его пиктограмме, а затем по позиции в документе, с которой должен располагаться верхний левый угол бокса будущего статического или динамического текста. Эти действия приводят к открытию панели “Текст”, на которой и реализуется первичный ввод и редактирование нового текста. При щелчке по кнопке *Ok* панель “Текст” закрывается и текст переносится на полотно. Динамические тексты обычно состоят из неизменяемой части и постоянно актуализируемой (вычисляемой) части. В дальнейшем созданный текст может быть отредактирован через страницу “Текст” панели “Настройки”. Ввод текста может протекать обычным набором символов или в *LaTeX*-режиме. В последнем слу-

чае в текст могут вставляться любые формулы, причем знать язык научной издательской системы *LaTeX2e* совсем не обязательно. Как правило, ввод текстов и формул легко реализуется с помощью управляющих элементов и многочисленных таблиц символов панели “Текст” или страницы “Текст”. Об этом подробно рассказывается в разделе 4.

Для получения динамического текста, являющегося формальным описанием того или иного объекта можно просто перетащить мышью копию этого описания с панели “Объекты” на полотно. Этим приемом можно создавать текст сразу для нескольких выделенных объектов. На рис. 64 показан результат “перетаскивания” описания прямой a и описаний трех выделенных точек A , B и C . При этом, как видим, было создано два текстовых объекта.

Подробное описание работы с текстами с использованием инструмента “ Текст” и специальных команд вынесено в отдельный раздел 4 – “Вставка текстов и формул”.

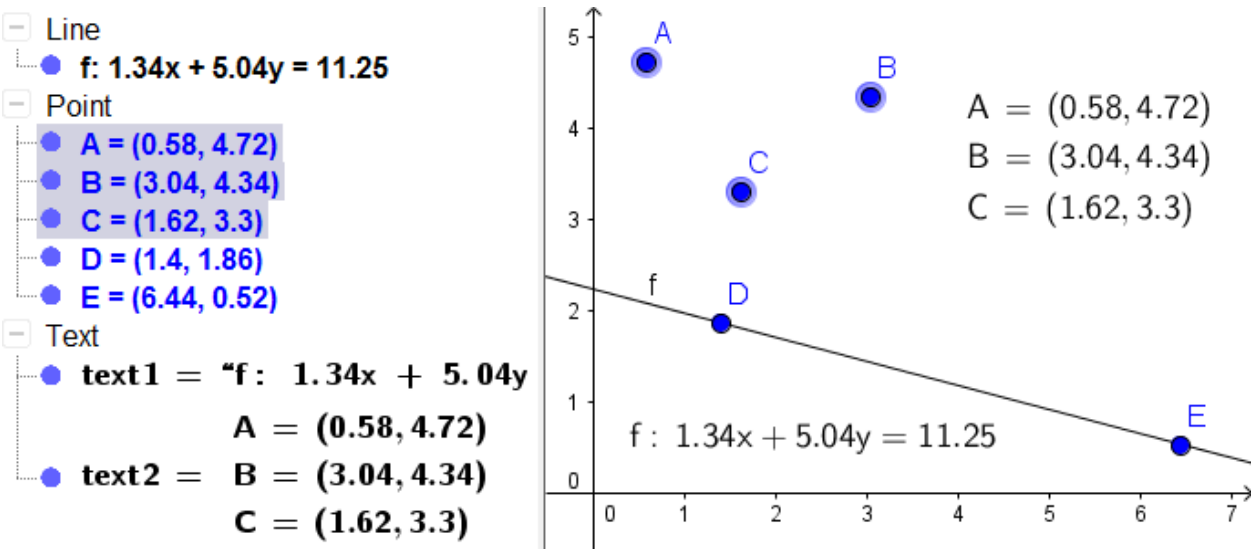






Рис. 64. Формирование описаний объектов перетаскиванием их мышью

 **Изображение.** Инструмент “ Изображение” используется для вставки в документ рисунков, схем, диаграмм и иных изображений из графических файлов с расширениями: *jpg, jpeg, png, gif, bmp* или *svg*. Вставленные изображения являются полноценными объектами *GeoGebra*. Вставка реализуется щелчком левой кнопки мыши сначала по пиктограмме инструмента, а затем по некоторой позиции на полотне, где должно размещаться изображение. Далее, на появившейся панели “Открыть” следует найти нужный файл, щелкнуть по его имени и затем щелкнуть по кнопке “Открыть”. Указанная выше позиция в документе становится левым нижним углом бокса изображения. Вставленное изображение при активированном инструменте “ Перемещать” можно мышью перемещать по полотну. Вместе с объектом “Изображение” в документ вставляются еще два дополнительных управляющих объ-

екта – “Точка”. Размещаются они в левом и правом нижних углах бокса изображения. При необходимости координаты этих точек можно изменять протяжкой мыши или каким-либо иным образом. Это будет приводить к изменению позиции изображения и его ширины. При этом длина изображения по умолчанию изменяется пропорционально ширине, хотя можно изменять ее и иным образом. Для этого требуется сформировать третью управляющую независимую или зависимую точку в левом верхнем углу бокса. Независимую точку можно создать так. Сначала инструментом “ Точка” в левый верхний угол бокса поместить некоторую точку. Затем на панели “Настройки” объявить эту точку четвертым углом бокса (*RClick(изображение)/Свойства/Координаты/Угол 4(имя точки)*). Зависимая точка создается аналогично, но при ее объявлении надо задавать не имя точки, а некоторое выражение, связывающее ее с другими объектами. Например, если *A* – имя первой точки, то третья точка может быть задана, скажем, так – $(x[A], 3)$. При необходимости все или некоторые управляющие точки можно сделать невидимыми или вообще удалить. В последнем случае изображение изменяется, приобретая свои реальные размеры. Отметим также, что если при выводе изображения щелчок мышью производился не по пустой позиции полотна, а по существующей точке, то она и оказывается первой управляющей точкой.

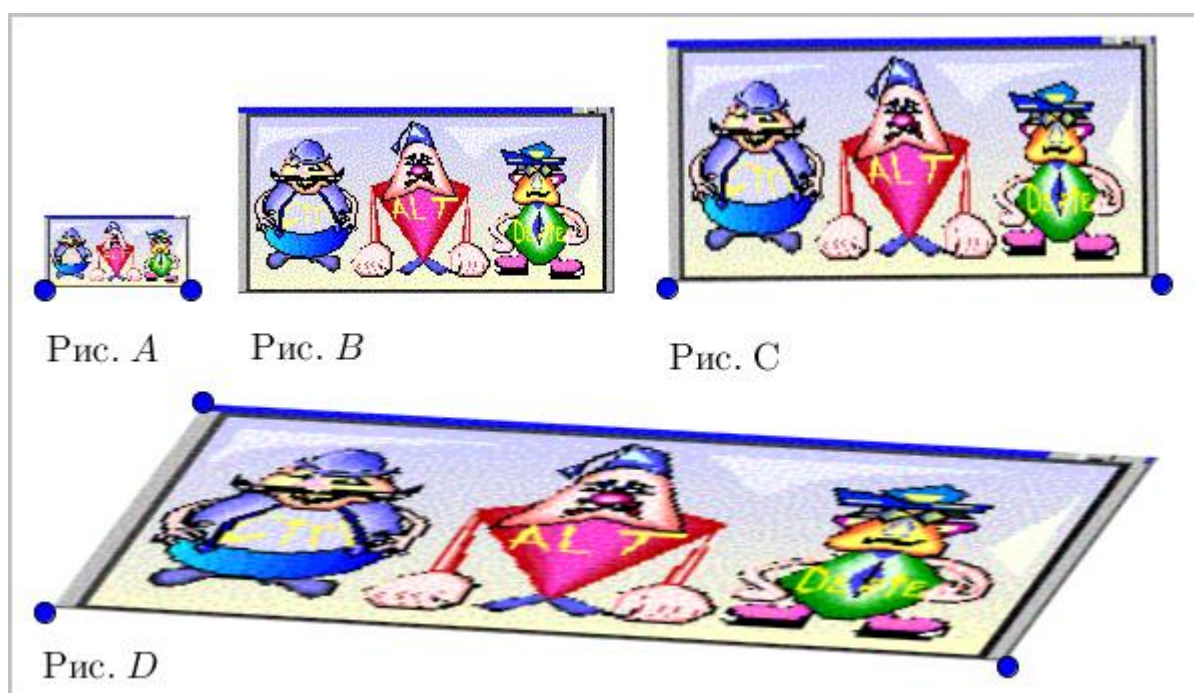










Рис. 65. Вставка в документ изображений

На рис. 65 демонстрируется вставка в документ изображений. Были проведены такие операции. Сначала из файла *Ctrl_Alt_Delete.gif* в документ вставлены 4 одинаковых изображения (рис. 65А-65D). Полученный рис. 65А оставлен без изменений. У рис. 65В были удалены управляющие точки, и он принял свои естественные размеры. С помощью протягивания управляющих точек у рис. 65С изменены размеры. Далее, сначала в левый верхний угол рис. 65D


добавлена третья управляющая точка, а затем протягиванием управляющих точек он и получил представленный вид.



Карандаш. Инструмент “ Карандаш” (“Перо”) позволяет добавлять на полотно объекты типа *PenStroke* в виде рукописных текстов и рисунков. Этот инструмент особенно полезен при презентациях, проводимых с помощью документов *GeoGebra*. В процессе представления документов с помощью данного инструмента объекты панели “Полотно” можно быстро выделять теми или иными пометками и столь же быстро освобождать их от пометок, ставших уже ненужными. Вставка объекта типа *PenStroke* начинается с щелчка левой кнопки мыши сначала по пиктограмме инструмента, а затем по некоторой позиции на полотне. После этого протаскиванием курсора мыши при нажатой ее левой клавише осуществляется формирование объекта. При рисовании левую кнопку мыши можно отпустить и изменить позицию курсора без рисования, а затем снова нажать и продолжить рисование. По умолчанию объект рисуется черным цветом, но через пиктографическое меню панели “Полотно” (“*Styling Bar*”) можно выбрать другой цвет, толщину и тип выводимых линий. При нажатии правой кнопки мыши осуществляется переход в режим *правки* объекта, при котором можно стереть те или иные его детали или весь объект целиком. Впрочем, полностью удалять объекты *PenStroke* быстрее общим способом – через контекстное меню. Созданные объекты типа *PenStroke* можно редактировать обычным образом через панель “Настройки”.


При переходе к любому инструменту, отличному от инструмента “ Карандаш”, формирование объекта *PenStroke* завершается созданием некоторого графического *svg*-файла. Количество таких объектов в документе может быть любым. Их, как и любые другие объекты *GeoGebra*, можно перемещать по полотну (инструмент “”), отражать относительно прямой и точки (инструменты “” и “”), проводить инверсию относительно окружности (инструмент “”), поворачивать вокруг некоторой точки на заданный угол (инструмент “”), формировать подобные им объекты с заданным коэффициентом подобия (“”) и т. д.



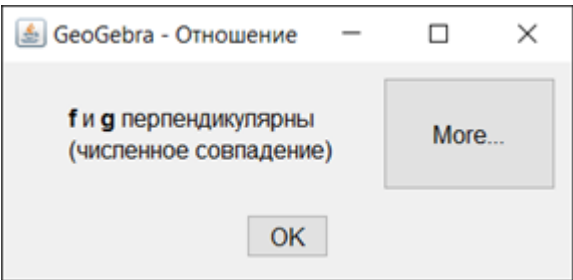
Фигура от руки. Инструмент “ Фигура от руки” позволяет протяжкой курсора мыши, то есть от руки, нарисовать эскиз круга, эллипса, отрезка, многоугольника, кривой и т. п. Система попытается распознать полученный предварительный набросок объекта и преобразовать его в подходящий реальный объект *GeoGebra*. Если это сделать удастся, то эскиз заменяется соответствующим объектом точной формы, если нет – эскиз просто исчезает. Если рисуется кривая и по ней формируется функция, то в дальнейшем можно будет вы-

числять ее значения, выполнять над ней ряд преобразований, таких как отражения, повороты и т. п., применять к ним различные функции подгонки, вычислять интегралы, но касательные и производные для таких функций не поддерживаются.



Отношение объектов. Инструмент “ *Отношение объектов*” позволяет получить информацию о взаимном расположении двух объектов документа. Например, выяснить, являются ли две прямые перпендикулярными или параллельными, лежит ли точка на прямой или конике, касается или пересекает прямая конику, являются или нет два объекта равными и т. п. Операция реализуется сначала щелчком по инструменту, затем щелчками по объектам, об отношении которых требуется получить информацию. При этом появляется отдельное окно с соответствующим сообщением. По умолчанию проверочные вычисления являются приближенными, хотя и проводятся с большой точностью – до 15 значащих цифр. Щелчком по кнопке “More” (дополнительно) можно запустить и символьные вычисления. Если их провести удастся, то выводится более точная информация. На рис 66a и 66b показаны сообщения, полученные об отношении двух прямых g и f . В данном случае символьные вычисления подтвердили их перпендикулярность.

a)



b)

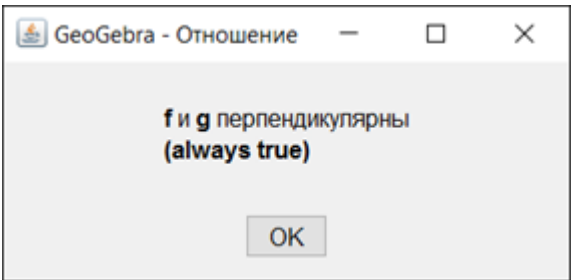



Рис. 66 (a, b). Вид информационных сообщений об отношении объектов



Инспектор функций. Инструмент “ *Инспектор функций*” используется для получения информации о функциях одной переменной x , выведенных на полотно через строку ввода некоторым выражением $g(x)$. Операция реализуется сначала щелчком по инструменту, затем щелчком по некоторой точке (x_0, y_0) кривой, сведения о которой нас интересуют. При этом на графике автоматически цветом выделяется часть кривой на интервале (обычно длины 2) с центром в x_0 , а также промежуток между кривой и осью абсцисс. Кроме того, выводится панель “*Исследователь функций*” с двумя страницами “*Интервал*” и “*Точки*”, на которых размещаются некоторые данные о кривой на выделенном интервале, и ряд управляющих элементов для работы с графиком и таблицей. Сразу же отметим, что все, что будет выведено на графике с помощью управляющих элементов панели “*Исследователь функций*” исчезнет при ее закрытии. На рис. 67 (a, b) показан пример – функция $x \cdot \sin(x)$, выделенная на

интервале (2,4), и панель “Исследователь функций” с разными открытыми страницами.

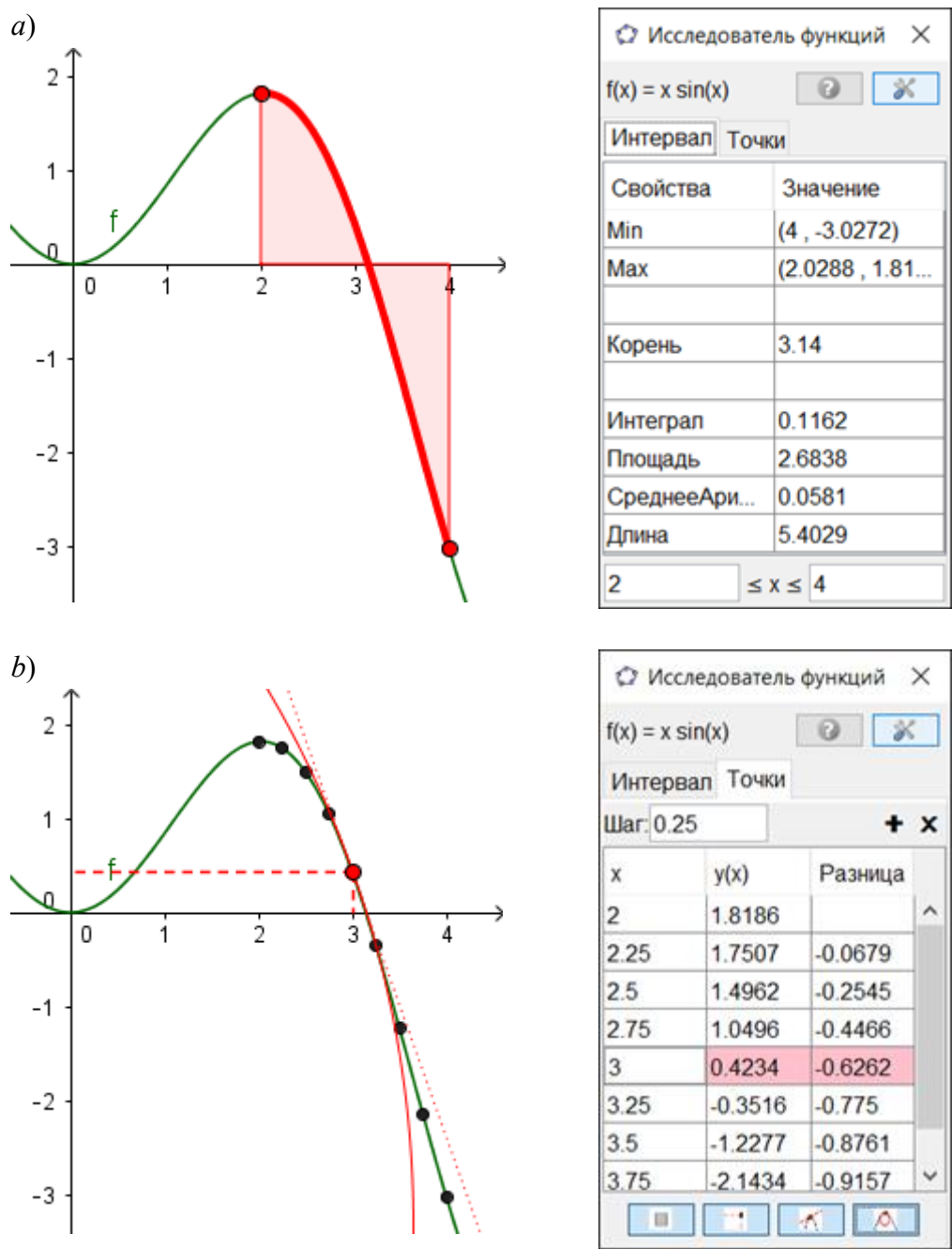






Рис. 67 (a, b). Демонстрация работы с инспектором функций

Внизу страницы “Интервал” мы видим два поля ввода с границами выбранного интервала, которые, при необходимости, могут быть изменены с актуализацией выделения на графике в реальном времени. Кроме того, страница “Интервал” содержит определение исследуемой функции, а также кнопки “Справки” и “Настройки”. По первой из них можно выйти на краткую *online* помощь по описываемому инструменту, а по второй – изменить количество разрядов после десятичной точки у выводимых в таблице чисел.

При переходе ко второй странице “Точки” в ее стартовом состоянии мы видим таблицу, в которой по умолчанию показана только одна точка (x_0, y_0) , а

прежнее выделение графика заменено показом этой точки. Однако набор из 4 управляющих кнопок внизу страницы, а также кнопки “+” и “X”, расположенные над таблицей, позволяют провести ряд операций по изменению и графика, и самой таблицы.

Начнем разговор с управляющих кнопок внизу страницы, работающих как двухпозиционные переключатели. Щелчки по ним выполняют некоторую операцию или отменяют ее. Вид, название и действие этих кнопок таковы:


- “ Показать набор точек”. Щелчок по кнопке приводит к появлению (удалению) таблицы точек на выбранном интервале. По умолчанию точки строятся с шагом 0.25, хотя это значение можно изменить. Эти же точки выводятся и на графике;
- “ Показать проекции”. Щелчок по кнопке приводит к появлению на графике проекций на оси координат начальной точки (x_0, y_0) . Щелчок по любой строке таблицы отменяет вывод прежних проекций и выводит их для текущей точки;
- “ Показать касательную”. Щелчок по кнопке приводит к появлению на графике касательной в начальной точке (x_0, y_0) . Щелчок по любой строке таблицы отменяет вывод прежней касательной и выводит ее для текущей точки;
- “ Показать соприкасающуюся окружность”. Щелчок по кнопке приводит к появлению на графике соприкасающейся окружности в начальной точке (x_0, y_0) . Щелчок по любой строке таблицы отменяет вывод прежней соприкасающейся окружности и выводит ее для текущей точки;

Кнопка “+” является заголовком раскрывающегося меню, выборы которого позволяют добавлять следующие столбцы к текущей таблице:

- “Первая производная” – добавление столбца первых производных функции в соответствующих точках;
- “Вторая производная” – добавление столбца вторых производных функции в соответствующих точках;
- “Разница” – добавление столбца разностей значений второй производной функции в текущей и предыдущей точках (начиная со второй строки);
- “Кривизна” – добавление столбца значений кривизны функции в соответствующих точках.

Наконец, кнопка “X” позволяет удалить последний из столбцов, выведенный выбором кнопки-меню “+”. Столбцы “ x ” и “ $y(x)$ ” этой кнопкой удалить нельзя.

3.3.12. Логические выражения


Альтернативой и дополнением к рассмотренному в предыдущем пункте инструменту “ Отношение объектов” является функция *Relation*, а также

серия логических функций с префиксами *Are* и *Is* (есть). Все они кратко описаны ниже. Но прежде чем вести о них речь, кратко остановимся на общих вопросах построения логических выражений в *GeoGebra*.

Таблица 1

Операторы отношений и логические операторы

Оператор	Название	Ввод с клавиатуры	Логическое выражение	Типы операндов
$\stackrel{?}{=}$	Равно	$==$	$p \stackrel{?}{=} q$	p, q – числа, точки, линии, коники
\neq	не равно	\neq	$p \neq q$	p, q – числа, точки, линии, коники
$<$	меньше	$<$	$p < q$	p, q – числа
$>$	больше	$>$	$p > q$	p, q – числа
\leq	меньше или равно	\leq	$p \leq q$	p, q – числа
\geq	больше или равно	\geq	$p \geq q$	p, q – числа
\in	принадлежит		$p \in list$	p – число, $list$ – список чисел
\parallel	параллельны		$p \parallel q$	p, q – прямые
\perp	перпендикулярны		$p \perp q$	p, q – прямые
\neg	унарный оператор “отрицание”	$!$	$\neg p$	p – булево выражение
\wedge	оператор “конъюнкция” (логическое “И”)	$\&\&$	$p \wedge q$	p, q – булевы выражения
\vee	оператор “дизъюнкция” (логическое “ИЛИ”)	\parallel	$p \vee q$	p, q – булевы выражения

Простым логическим выражением называют выражения вида $p \otimes q$, где p и q – некоторые объекты и \otimes – оператор отношения ($=$, \neq , $<$, $>$, \leq , \geq , \in , \parallel , \perp). Значениями таких выражений являются логические константы *true* (истина) и *false* (ложь) (см. табл. 1). Из простых логических выражений с помощью логических операторов “ \neg ” (не), “ \vee ” (или) и “ \wedge ” (и) формируются составные логические выражения. Их значениями также являются логические константы *true* и *false*. Значения выражений с логическими операторами в зависимости от значений истинности операндов вычисляются так, как это показано в табл. 2. Ввод логических выражений осуществляется по-разному: при записи команд – через строку ввода, при формировании текстов и формул – с помощью инструмента “ Текст” – через поле ввода панели “Текст”, при написании скриптов – через панель “Настройки”. В первом случае ввод операторов отношений


и логических операторов возможен с клавиатуры или с панели символов, которая раскрывается щелчком по кнопке “”, расположенной в конце строки ввода. Во втором случае ввод операторов осуществляется аналогично первому случаю, но соответствующая панель символов раскрывается с панели “Текст” (Символы/Основные). В третьем случае ввод операторов осуществляется только с клавиатуры. Простые и сложные логические выражения называют также булевыми выражениями. Кроме того, в GeoGebra есть функции, которые в качестве аргументов имеют булевы выражения (*Prove*, *ProveDetails*, ...), а также функции, которые возвращают логические значения *true* и *false* (*AreCollinear*, *AreConcurrent*, ...).

Таблица 2

Вычисление выражений с логическими операторами

Таблица истинности (<i>p</i> , <i>q</i> – произвольные логические выражения)					
Операнды/ Оператор	<i>p</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>
	<i>q</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
\neg	$\neg p$	<i>false</i>		<i>true</i>	
\wedge	$p \wedge q$	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>
\vee	$p \vee q$	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>

Операторы \neg , \wedge и \vee удовлетворяют следующим соотношениям:

- $p \wedge q \Leftrightarrow q \wedge p$ – коммутативность \wedge ;
- $p \vee q \Leftrightarrow q \vee p$ – коммутативность \vee ;
- $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$ – ассоциативность \wedge ;
- $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$ – ассоциативность \vee ;
- $(p \wedge q) \vee (p \wedge r) \Leftrightarrow p \wedge (q \vee r)$ – дистрибутивность;
- $(p \vee q) \wedge (p \vee r) \Leftrightarrow p \vee (q \wedge r)$ – дистрибутивность;
- $p \vee (\neg p) \Rightarrow true$;
- $p \wedge (\neg p) \Rightarrow false$;
- $\neg(\neg p) \Rightarrow p$;
- $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ – закон Моргана;
- $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$ – закон Моргана;

Здесь символ “ \Leftrightarrow ” следует читать как “тогда и только тогда”, а символ “ \Rightarrow ” – как “следует”.

Отметим, что “составные” отношения вида

$$a < x \wedge x < b, a < x \wedge x = b, a = x \wedge x < b, a = x \wedge x = b.$$

в Geogebra можно записывать напрямую:

$$a < x < b, \ a < x \leq b, \ a \leq x < b, \ a \leq x \leq b.$$

Что касается операторов исключающего “ИЛИ” (\vee), импликации (\rightarrow) и эквиваленции (\leftrightarrow), то в *GeoGebra* их нет, да и используются они редко. Впрочем, их действие легко описывается (выражается) через типовые логические операторы \neg , \wedge и \vee так:

- $p \rightarrow q \Leftrightarrow \neg p \vee q$;
- $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$;
- $p \vee q \Leftrightarrow \neg(p \leftrightarrow q)$.

Теперь можно перейти к описанию обещанных логических функций.

A. *Relation*[*obj1*, *obj2*]

Получение информации об отношении объектов

A) По команде *A* выясняется, в каком отношении находятся объекты *obj1* и *obj2*. В аргументах объекты могут задаваться непосредственно или своими именами.

A. *AreCollinear*[*po1*, *po2*, *po3*]

B. *AreConcyclic*[*po1*, *po2*, *po3*, *po4*]

Точки на объекте

A) По команде *A* выясняется, лежат ли точки *po1*, *po2* и *po3* на одной прямой, и, если да, то возвращается *true*, иначе – *false*.

B) По команде *B* выясняется, лежат ли точки *po1*, *po2*, *po3* и *po4* на одной окружности, и, если да, то возвращается *true*, иначе – *false*.

A. *AreParallel*[*li1*, *li2*]

B. *ArePerpendicular*[*li1*, *li2*]

Перпендикулярные и параллельные прямые

Здесь аргументы *li1* и *li2* могут задаваться именами линий или непосредственно.

A) По команде *A* выясняется, являются ли линии *li1* и *li2* параллельными, и, если да, то возвращается *true*, иначе – *false*. Команда *A* равносильна вычислению выражения $li1 \parallel li2$. Например, *AreParallel*[*f*, *g*] и *f*||*g* (*f*, *g* – имена прямых) или *AreParallel*[*y=x*, *y=x+5*] и $(y=x) \parallel (y=x+5)$.

B) По команде *B* выясняется, являются ли линии *li1* и *li2* перпендикулярными, и, если да, то возвращается *true*, иначе – *false*. Команда *B* равносильна вычислению выражения $li1 \perp li2$. Например, *ArePerpendicular*[*f*, *g*] и *f*⊥*g* (*f*, *g* – имена прямых) или *ArePerpendicular*[*y=-x*, *y=x+5*] и $(y=-x) \perp (y=x+5)$.

A. *AreConcurrent*[*li1*, *li2*, *li3*]

Пересечение трех прямых

A) По команде *A* выясняется, пересекаются ли линии *li1*, *li2* и *li3* в одной точке, и, если да, то возвращается *true*, иначе – *false*.

A. *AreCongruent*[*obj1*, *obj2*]

B. *AreEqual*[*obj1*, *obj2*]

Конгруэнтность и равенство объектов

Напомним, что в евклидовой геометрии два объекта (фигуры) называются конгруэнтными, если одна из них может быть получена из другой сдвигом, вращением,

зеркальным отображением или композицией этих преобразований. Конгруэнтность – задает отношение эквивалентности на множестве геометрических фигур. Из этого определения вытекает, что точки всегда конгруэнтны друг другу. Две фигуры считаются равными если они конгруэнтны и совпадают все их элементы, соответствующие друг другу.

Объекты в аргументах команд *A* и *B* могут задаваться непосредственно или своими именами.

А) По команде *A* выясняется, являются ли объекты *obj1* и *obj2* конгруэнтными, и, если да, то возвращается *true*, иначе – *false*.

В) По команде *B* выясняется, являются ли объекты *obj1* и *obj2* равными, и, если да, то возвращается *true*, иначе – *false*. Команда *B* равносильна вычислению выражения *obj1==obj2*.

<p>A. <i>IsInRegion</i>[<i>po</i>, <i>region</i>]</p>	<p><i>Принадлежность точки данной области</i></p> <p>А) По команде <i>A</i> проверяется, принадлежит ли точка <i>po</i> области <i>region</i>, и, если да, то возвращается <i>true</i>, иначе – <i>false</i>. Аргументом <i>region</i> может быть многоугольник, окружность (круг), полуокружность (полукруг), сектор, коника (эллипс, гипербола, парабола) и область решений неравенства с двумя переменными. Команда <i>A</i> равносильна вычислению выражения <i>po∈region</i>.</p>
---	--

<p>A. <i>IsDefined</i>[<i>obj</i>]</p>	<p><i>Определен ли объект</i></p> <p>А) По команде <i>A</i> возвращается значение <i>true</i> или <i>false</i> в зависимости от того, определен или нет объект <i>obj</i>. Если объект <i>obj</i> не определен и в <i>obj</i> есть неопределенные переменные, то по <i>A</i> получаем сообщение об ошибке. Примеры: <i>A</i>=(2, 3), <i>IsDefined</i>[<i>A</i>] → <i>true</i>; если <i>C</i> и <i>D</i> определены, то <i>IsDefined</i>[<i>Line</i>[<i>C</i>, <i>D</i>]] → <i>true</i>; <i>IsDefined</i>[<i>y=x+5</i>] → <i>true</i>.</p>
--	---




<p>A. <i>IsInteger</i>[<i>num</i>] B. <i>IsPrime</i>[<i>num</i>]</p>	<p><i>Является ли значение числом или простым числом</i></p> <p>А) По команде <i>A</i> возвращается значение <i>true</i> или <i>false</i> в зависимости от того, является ли числом значение выражения <i>num</i> или нет. Например, <i>IsInteger</i>[2+6/2] → <i>true</i>; <i>IsInteger</i>[3+<i>a</i>] → <i>false</i>; если <i>a</i>=7 сформировано вводом числа 7, то <i>IsInteger</i>[3+<i>a</i>] → <i>true</i>.</p> <p>В) По команде <i>B</i> возвращается значение <i>true</i> или <i>false</i> в зависимости от того, является ли простым числом значение выражения <i>num</i> или нет. Например, <i>IsPrime</i>[2+6/2] → <i>true</i>; если <i>a</i>=8 сформировано вводом числа 8, то <i>IsInteger</i>[3+<i>a</i>] → <i>true</i>; <i>IsPrime</i>[4+<i>a</i>] → <i>false</i>. Значение <i>num</i> не должно превышать величину 9007199254740992. При <i>num</i>>9007199254740992 следует использовать символьные вычисления.</p>
--	--

<p>A. <i>Prove</i>[<i>bool</i>] B. <i>ProveDetails</i>[<i>bool</i>]</p>	<p><i>Форсирование символьных вычислений</i></p> <p>А) По команде <i>A</i> символьно вычисляется логическое выражение <i>bool</i> и возвращается полученное значения <i>true</i> или <i>false</i>.</p>
---	--

Заметим, что по командам, возвращающим логические значения, вычисления проводятся приближенно, хотя и с большой точностью – до 15 значащих цифр. Если мы желаем, чтобы по таким командам проводились символьные вычисления, то их следует использовать в качестве аргументов в командах A или B . При этом следует учитывать следующее обстоятельство. По A и B возвращается значение $true$ только в тех случаях, когда теми или иными соображениями удастся установить, что $bool=true$ при всех значениях переменных, которые входят в $bool$ и связаны со свободными или частично свободными (прикрепленными) объектами. Иными словами, при динамических изменениях свободных объектов, влияющих на значения переменных $bool$, значение $bool$ должно оставаться равным $true$. Если символьные вычисления провести не удастся, выводится символ “?”.

B) По команде B делается попытка символьно вычислить логическое выражение $bool$. В результате вычислений возвращается некоторый список, который может быть таким:

- $\{\}$ – GeoGebra не в состоянии вычислить выражение $bool$;
- $\{false\}$ – $bool \neq true$ при всех возможных случаях;
- $\{true\}$ – $bool=true$ при всех возможных случаях;
- $\{true, \{\text{последовательность дополнительных условий}\}\}$ – $bool=true$ при некоторых дополнительных условиях;
- $\{true, \{\dots\}\}$ – $bool=true$ при некоторых дополнительных условиях, записать которые не удалось.

Пример 67. Пусть на полотно через строку ввода выведены три точки $A=(1, 1)$, $B=(3, 1.5)$ и $C=(5, 2)$, через точки A и C проведена прямая f (инструмент “”), на f закреплена некоторая точка D (инструмент “”) и через строку ввода для точек A и D построена средняя точка $E=MidPoint[A,D]$. Инструментом “ Текст” создадим проверочные надписи с помощью следующих двух двухстрочных выражений:

$$\begin{aligned} AreCollinear[A,B,C] &\rightarrow \boxed{AreCollinear[A,B,C]} , \\ Prove[AreCollinear[A,B,C]] &\rightarrow \boxed{Prove[AreCollinear[A,B,C]]} . \\ \\ AreCollinear[A,D,E] &\rightarrow \boxed{AreCollinear[A,D,E]} , \\ Prove[AreCollinear[A,D,E]] &\rightarrow \boxed{Prove[AreCollinear[A,D,E]]} . \end{aligned}$$

Выведенные по ним надписи показаны на рис. 68. Поясните полученные значения ($true, false, true$ и $true$) в правых частях этих надписей.

Решение. В первой строке первой надписи в правой вычисляемой части реализуются приближенные вычисления для текущих координат точек A , B и C . Но точки $A=(1, 1)$, $B=(3, 1.5)$ и $C=(5, 2)$ действительно расположены на одной прямой. Отсюда и получаем значение $true$. Во второй строке первой

надписи благодаря использованию команды *Prove* в вычисляемой части реализуются символьные вычисления. Но не любые три точки лежат на одной прямой. Отсюда и получаем значение *false*.

В первой строке второй надписи в правой вычисляемой части реализуются приближенные вычисления для текущих координат точек *A*, *D* и *E*. Но при данных координатах эти точки действительно расположены на одной прямой. Отсюда и получаем значение *true*. Во второй строке второй надписи благодаря использованию команды *Prove* в вычисляемой части реализуются символьные вычисления. Но точки *D* и *E* строились так, что при любой позиции свободной точки *A* все три точки *A*, *D* и *E* всегда оказываются на одной прямой. Отсюда и получаем значение *true*.

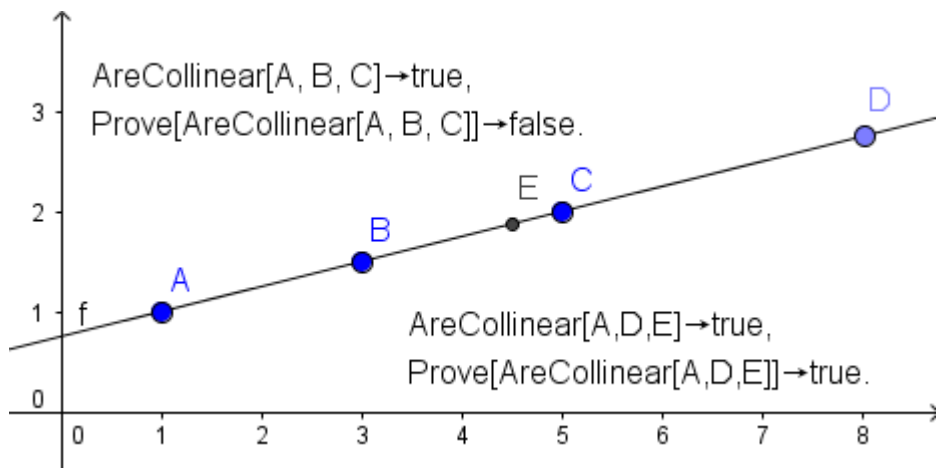


Рис. 68. Приближенные и символьные вычисления в примере 67

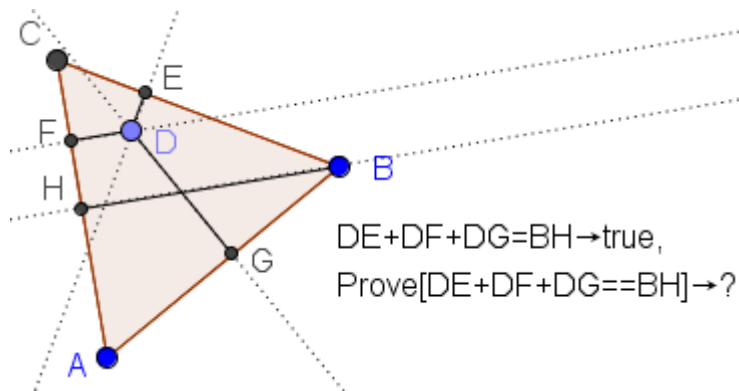


Рис. 69. Команда *Prove* и теорема Вивиани


Замечание. Пока еще нельзя утверждать, что средства типа *Prove* и *ProveDetails* достаточно эффективны при доказательстве теорем. Например, система не в состоянии установить справедливость даже такого простого утверждения, как теорема Вивиани: “В равностороннем треугольнике сумма расстояний от любой точки треугольника до его сторон постоянна и равна высоте” (см. рис. 69).

3.3.13. Управляющие элементы

GeoGebra предлагает пользователю набор стандартных управляющих интерфейсных элементов “Ползунок” (слайдер), “Флажок” (независимый переключатель), “Кнопка” (кнопка общего назначения) и “Окно ввода” (текстовое поле). В одном документе можно разместить любое количество управляющих элементов любого типа. Формируются они одноименными инструментами и часто используются при экспериментальной проверке справедливости тех или иных утверждений, то есть, их верификации. Пусть *cont* – название управляющего элемента. Вставка *cont* в документ реализуется щелчком левой кнопки мыши сначала по пиктограмме *cont*, а затем по некоторой позиции на полотне, где первоначально должен размещаться левый верхний угол бокса *cont*. Эта позиция может быть фиксированной или свободной. В первом случае элемент нельзя перемещать протаскиванием мышью по полотну, а во втором – можно. По умолчанию позиция бокса *cont* фиксирована, а изменяется ее состояние через контекстное меню командой *RClick(control)/Закрепить cont*. Удалить элемент из документа можно через контекстное меню командой *RClick(cont)/Удалить*. При этом для элементов “Ползунок” и “Флажок” удаление можно реализовать всегда, а для элементов “Кнопка” и “Окно ввода” – только при свободной (не фиксированной) позиции бокса.

Отметим также, что по умолчанию позиция на экране любого управляющего элемента является абсолютной, то есть неизменяемой при перемещении системы координат или масштабировании изображения. Но для ползунка ее можно сделать и относительной. Для этого через контекстное меню или на вкладке “Координаты” панели “Настройки” следует выключить независимый переключатель “Абсолютная позиция на экране”. После этого ползунок будет перемещаться вместе с системой координат и увеличиваться или уменьшаться в соответствии с проводимым масштабированием изображения.



Ползунок. Инструмент “ Ползунок” используется для вставки в документ управляющего элемента, называемого ползунком или слайдером. Применяются ползунки для установки начальных значений конкретных параметров (переменных величин) решаемой задачи и их автоматического или ручного изменения в некотором диапазоне. При вставке ползунка в документ открывается диалоговая панель “Ползунок” (рис. 70), на которой уже заданы установки, определяющие его свойства по умолчанию. Управляющими элементами панели “Ползунок” эти установки можно изменять по своему усмотрению и делается это так:

- в поле ввода “Имя” формируется имя создаваемого ползунка. Это имя выводится рядом с движком ползунка и перемещается вместе с ним. Если через панель “Настройки” для ползунка сформировать заголовок, то он будет выводиться вместо имени;

- одной из трех радиокнопок в левом верхнем углу панели определяется, чем является текущее значение ползунка: *числом*, *углом в градусах* или *целым числом*. Символ “°” градуса с клавиатуры можно вводить ключом *Alt+o*;
- на закладке “*Интервал*” в трех полях ввода задаются начальное *mi* и конечное *ma* значения диапазона изменения движка слайдера, а также шаг *h* его изменения;
- независимым переключателем “*Случайное число*” при анимации задается тип изменения позиции движка слайдера. При выключенном переключателе происходят *последовательные* изменения позиции движка в диапазоне $[mi, ma]$ с шагом *h*, а при включенном переключателе – *случайные* изменения позиции движка слайдера в этом же диапазоне $[mi, ma]$;

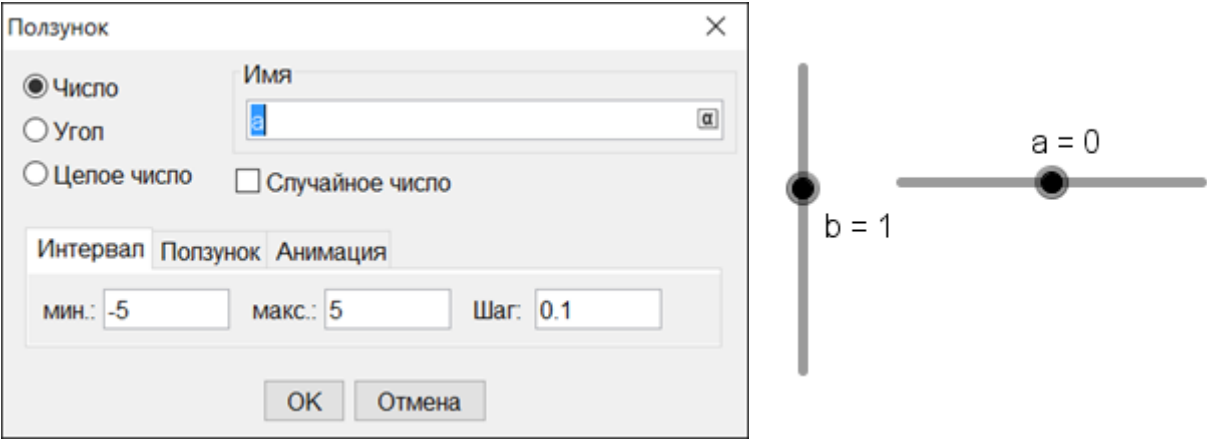


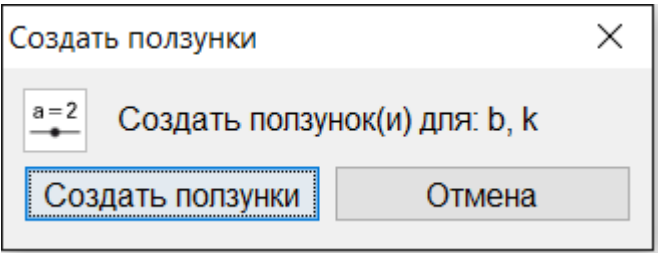
Рис. 70. Панель для установки и изменения свойств ползунков и их виды

- на закладке “*Ползунок*” независимым переключателем можно закрепить или снять закрепление позиции левого верхнего угла бокса ползунка. Закрепленный ползунок перемещать по документу нельзя, незакрепленный – можно и делается это протаскиванием его мышью. На этой же закладке имеется список – для выбора горизонтального или вертикального расположения ползунка и соответствующего направления перемещения его движка, а также поле ввода – для задания длины направляющего отрезка, по которому перемещается движок ползунка;
- на закладке “*Анимация*” имеются такие управляющие элементы: поле ввода – для задания скорости перемещения движка при анимации, и список, выборы которого определяют тип перемещения движка. Эти выборы и соответствующие им типы перемещения могут быть такими: “*колебание*” (осцилляция) – движок непрерывно перемещается между границами диапазона его изменения, начиная движение направо (или вниз) от текущей позиции и т. д; “*увеличение*” – движок непрерывно перемещается направо (или вниз) от текущей позиции до конечного значения диапазона, затем направо (или вниз) от начального значения диапазона и т. д; “*уменьшение*” – движок непрерывно перемещается налево (или вверх) от текущей позиции до начального значения диапазона, затем налево (или вверх) от конечного значения диапазона и т. д; “*увеличение (один раз)*” – движок перемещается направо (или вниз) от текущей

позиции до конечного значения диапазона и анимация прекращается. Запуск анимации и ее прекращение можно осуществлять через контекстное меню командой *Click(направляющая движка)/Animation On*. При запущенной анимации в левом нижнем углу панели “Полотно” появляется кнопка “⏸”. Щелчок левой кнопкой мыши по кнопке “⏸” превращает ее в кнопку “▶” и приостанавливает анимацию. Щелчок левой кнопкой мыши по кнопке “▶” возвращает ей первоначальный вид “⏸” и продолжает анимацию с прерванного места. При отключении анимации кнопка “⏸” исчезает.

При щелчке по кнопке *Ok* или по ключу *Enter* панель “Ползунок” закрывается и соответствующий управляющий элемент оказывается на полотне. Через панель “Настройки” для него можно реализовать некоторые дополнительные установки, в том числе и изменить ширину направляющего отрезка его движка.

В тех случаях, когда система обнаруживает неопределенные параметры, она предлагает пользователю автоматически создать один или несколько ползунков. Например, если в строку ввода ввести выражение $y=k \cdot x+b$, где переменные k и b неопределены, то появится следующая панель



с запросом о формировании сразу двух ползунков для параметров b и k . При отказе от их создания появится сообщение о некорректности ввода, при согласии – на панелях “Объекты” и “Полотно” появится изображение, показанное на рис. 71.

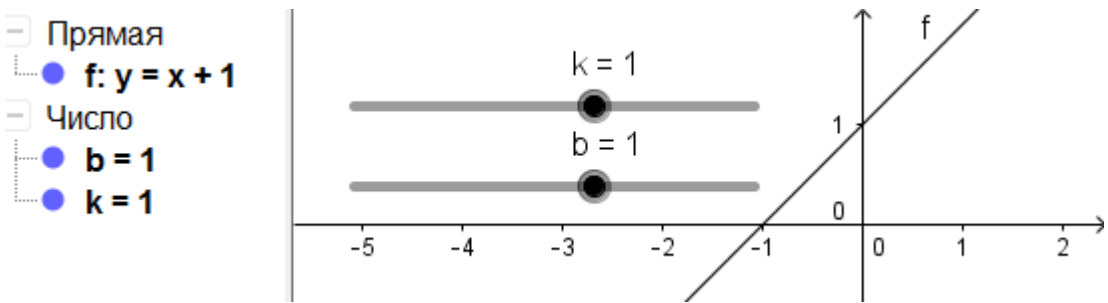





Рис. 71. Автоматическое создание двух ползунков

Иными словами, создаются два ползунка b и k с начальными значениями 1, и для этих значений выведен график функции $y=k \cdot x+b$, то есть прямая f . Если необходимо, то свойства ползунков и прямой f могут быть изменены так, как

это необходимо пользователю. Но можно и сразу начинать работу с полученной динамической моделью, то есть протаскивать мышью движки ползунков и наблюдать за автоматической актуализацией положения прямой f или запустить анимацию по одному из параметров b или k , или сразу по двум этим параметрам.

Пример 68. Создать ползунок с целочисленными значениями $n=3, 4, \dots, 20$ и связать его с правильным n -угольником со стороной, проходящей через две заданные точки A и B . Иными словами, при перемещениях движка ползунка при его текущем значении n ($3 \leq n \leq 20$) должен выводиться правильный n -угольник со стороной AB .

Решение. Выполним следующую последовательность действий:

- инструментом “ Ползунок” выведем управляющий элемент – ползунок с именем n , диапазоном $[3, 20]$ и шагом 1. По умолчанию текущее значение n равно 3;
- инструментом “ Точка” выведем две точки A и B ;
- инструментом “ Правильный многоугольник” проведем n -угольник по точкам A, B и параметру n (в данный момент $n=3$).

Теперь с помощью ползунка простой протяжкой его движка мы можем строить правильные n -угольники при любых целых n ($3 \leq n \leq 20$). На рис. 72 показан многоугольник при $n=7$. Можно также запустить анимацию с последовательными или случайными значениями n .

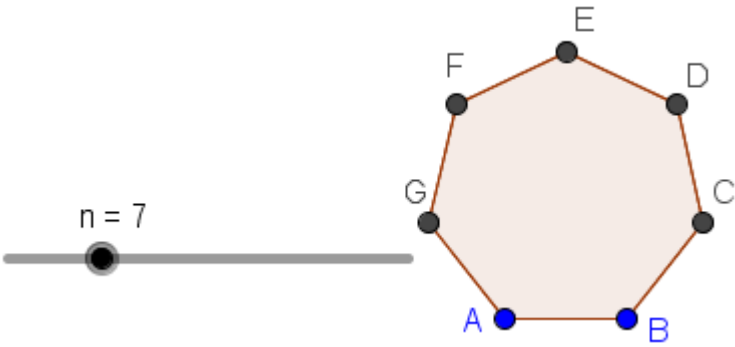


Рис. 72. Построение правильных n -угольников с помощью ползунка





```
A. Slider[mi, ma, h, speed, len, bool-angle, bool-hv, bool-anima, bool-random]
```

Создание ползунка

A) По команде A создается ползунок. Обязательными являются только первые два аргумента A , задающие диапазон $[mi, ma]$ ($mi < ma$) изменения движка ползунка. Однако, если требуется задать аргумент с номером k ($k > 2$), то все предыдущие аргументы также должны быть заданы. Аргументы, начиная с третьего и далее, имеют следующий смысл:

- h – шаг для движка. По умолчанию $h=0.1$;

- *speed* – скорость перемещения движка при анимации объекта, связанного с данным ползунком. По умолчанию *speed=1*;
- *len* – длина направляющего отрезка движка. По умолчанию *len=200px*;
- *bool-angle* – логическое значение, определяющее для чего строится ползунок: для угла (*true*) или для числа (*false*). По умолчанию *bool-angle=false*;
- *bool-hv* – логическое значение определяющее, как должен быть расположен ползунок: горизонтально (*true*) или вертикально (*false*). По умолчанию *bool-hv=true*;
- *bool-anima* – логическое значение включающее автоматическую анимацию (*true*) или выключающее ее (*false*). По умолчанию *bool-anima=false*;
- *bool-random* – логическое значение, задающее последовательные изменения позиции движка слайдера в диапазоне $[mi, ma]$ с шагом *h* (*false*) или случайные изменения позиции движка слайдера в этом же диапазоне $[mi, ma]$ (*true*). По умолчанию *bool-random=false*.

 **Флажок.** Инструмент “ Флажок”, называемый также независимым переключателем, имеет два состояния: “” – включен и “” – выключен. При создании флажок связывается с одним или с несколькими объектами документа, на которые он и должен воздействовать – включение флажка делает связанные с ним объекты видимыми, а выключение – невидимыми. При вставке флажка в документ открывается диалоговая панель “Флажок...” (см. рис. 73), на которой можно сформировать заголовок флажка, выводимый справа от него, и выбрать из списка те объекты, которые требуется связать с данным флажком. Если заголовок не задавать, то вместо него будет выводиться имя флажка.

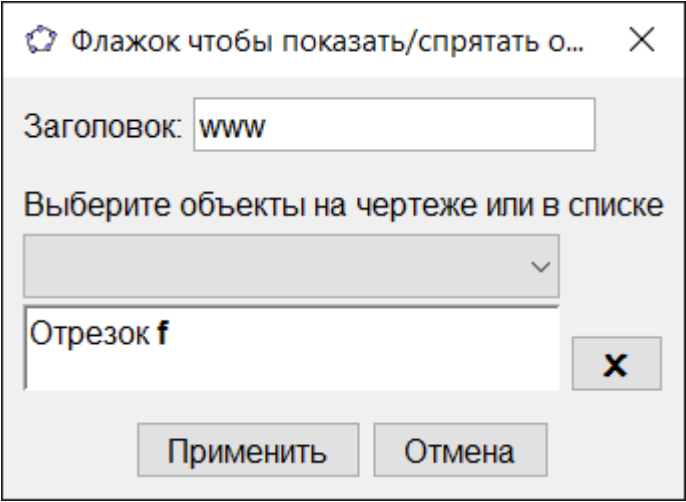


Рис. 73. Панель для формирования флажков

Объекты можно выбирать не только из списка панели “Флажок”, но также и из списка панели “Объекты”, или указывать непосредственно на панели “Полотно”. Выбранные объекты размещаются построчно в поле панели

“Флажок”, справа от которого расположена кнопка “☐ x””. Щелчок по ней удаляет текущий объект из списка уже выбранных объектов. При щелчке по кнопке “Применить” панель “Флажок” закрывается и соответствующий управляющий элемент оказывается на полотне.

На рис. 74а показаны примеры включенных флажков и связанных с ними объектов: первый флажок связан с прямой f , но не с точками D и E , через которые f проходит, второй флажок связан с окружностью g , но не с центром F и точкой G , через которую g проходит, третий флажок связан с точкой H и $\triangle ABC$ ($poly1$), но не с точками и отрезками, являющимися вершинами и сторонами этого треугольника. На рис. 74b показаны те же объекты при выключенных флажках.

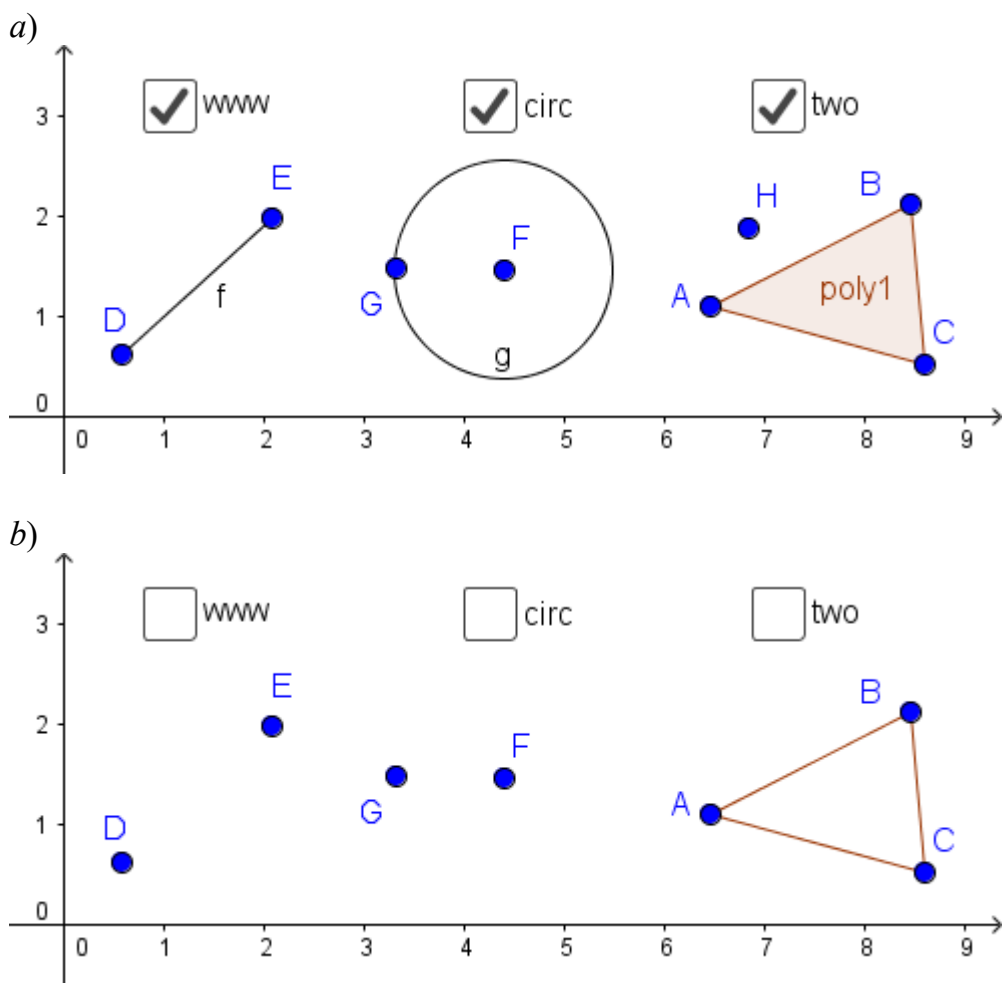






Рис. 74 (a, b). Примеры флажков и связанных с ними объектов

Замечание. Визуализировать и прятать объекты можно не только с помощью флажков, но и включением и выключением радиокнопок около соответствующих объектов на панели “Объекты”.

Пример 69. Построить многоугольник P и вывести три флажка, по которым прячутся и выводятся соответственно вершины P , стороны P и сам P .

Решение. Выполним следующую последовательность действий:

- инструментом “ Многоугольник” выведем некоторый многоугольник, скажем, пятиугольник $ABCDE$;
- инструментом “ Флажок” выведем флажок с именем “*вершины*” и свяжем его с вершинами A, B, C, D и E пятиугольника $ABCDE$;
- инструментом “ Флажок” выведем флажок с именем “*стороны*” и свяжем его со сторонами a, b, c, d и e пятиугольника $ABCDE$;
- инструментом “ Флажок” выведем флажок с именем “*многоугольник*” и свяжем его с пятиугольником $ABCDE$ (объект “*многоугольник1*”);
- разместим флажки друг под другом, а пятиугольник $ABCDE$ – справа от флажков.

Модель для демонстрации действия флажков построена. Всего имеется 8 вариантов их включений, 4 из них приведены на рис. 75.

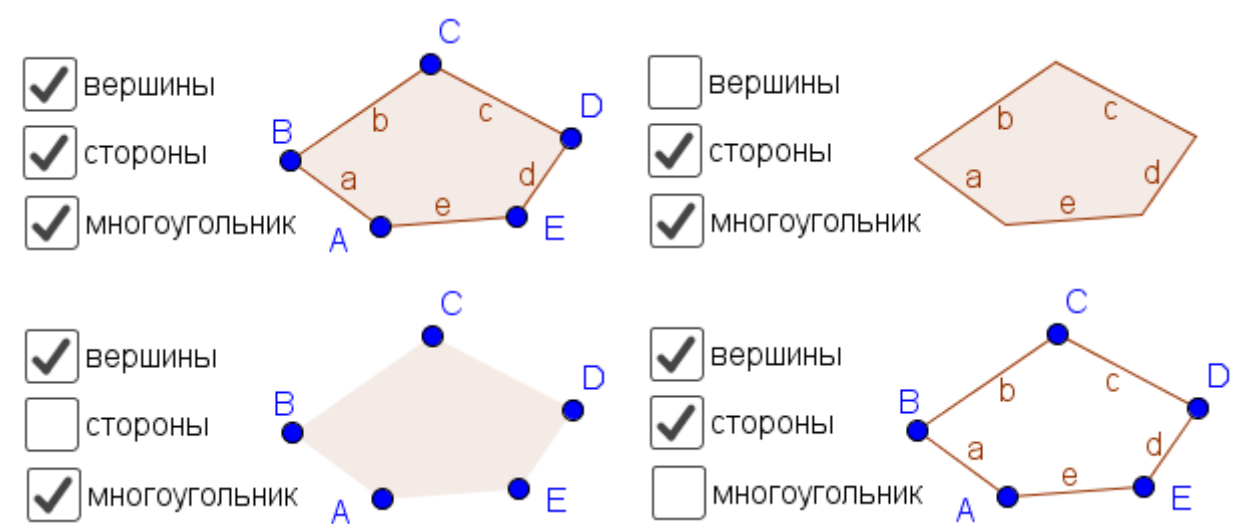



Рис. 75. Действия флажков в примере 69

<p>A. <code>Checkbox[]</code> B. <code>Checkbox[caption]</code></p>	<p>C. <code>Checkbox[list]</code> D. <code>Checkbox[caption, list]</code></p>	<p>Создание флажка</p> <p>А) По команде <i>A</i> выводится включенный флажок, не связанный с какими-либо объектами и имеющий надпись в виде имени флажка.</p> <p>В) По команде <i>B</i> выводится включенный флажок, не связанный с какими-либо объектами и имеющий надпись в виде заголовка <i>caption</i>. Значение аргумента <i>caption</i> должно быть строковым.</p> <p>С) По команде <i>C</i> выводится флажок, связанный с объектами, перечисленными в списке <i>list</i> ($\{l_1, l_2, \dots\}$). Надпись у флажка совпадает с его именем. Первоначально флажок включен, то есть объекты списка <i>list</i> видимы.</p> <p>Д) По команде <i>D</i> выводится флажок, связанный с объектами, перечисленными в списке <i>list</i>. Надпись у флажка совпадает с заголовком <i>caption</i>. Значение аргумента <i>caption</i> должно быть строковым. Первоначально флажок включен, то есть объекты списка <i>list</i> видимы.</p>
--	--	--



Кнопка. Инструмент “ Кнопка” предназначен для формирования кнопки общего назначения, щелчки по которой запускают на выполнение некоторый *GeoGebra*-скрипт, то есть последовательность *GeoGebra*-команд. При вставке кнопки в документ открывается диалоговая панель “Кнопка” (см. рис. 76), на которой можно сформировать выводимый на кнопке заголовок и записать скрипт, по которому будут осуществляться требуемые действия. Если заголовок не задавать, то вместо него на кнопку будет выводиться ее автоматически формируемое имя: *кнопка1*, *кнопка2*, ... (*button1*, *button2*, ...). Дополнительно через закладку “Стиль” панели “Настройки” на кнопку можно поместить некоторый предопределенный рисунок, или загрузить требуемое изображение из файла с расширением *jpg*, *jpeg*, *png*, *gif*, *bmp* или *svg*. Размер формируемой кнопки по умолчанию не фиксирован и связан с выводимой на ней надписью и изображением. Впрочем, на той же самой закладке “Стиль” панели “Настройки” размер кнопки можно изменить, введя и зафиксировав для нее требуемые ширину и высоту. При щелчке по кнопке “Ok” панель “Кнопка” закрывается и соответствующий управляющий элемент оказывается на полотно.

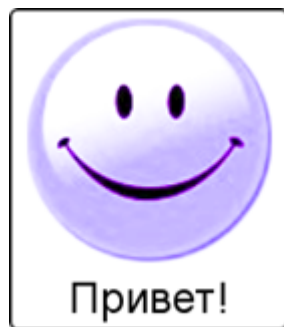
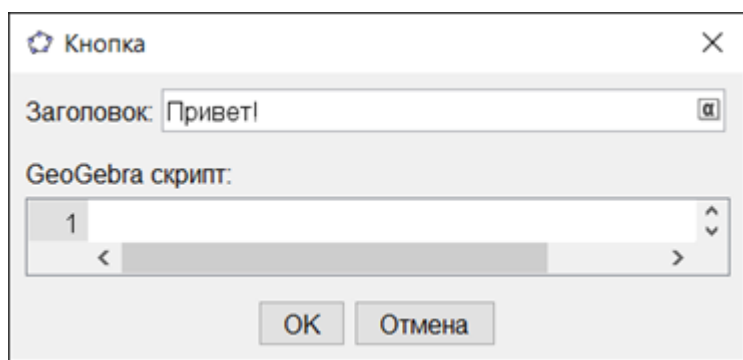


Рис. 76. Панель для формирования кнопок общего назначения и кнопка

- | |
|---|
| <p>A. <i>Button</i>[]</p> <p>B. <i>Button</i>[<i>caption</i>]</p> |
|---|

Создание кнопки общего назначения

A) По команде A выводится кнопка общего назначения с надписью в виде имени кнопки. Написание скрипта для кнопки и изменение ее свойств реализуется через панель “Настройки”.

B) По команде B выводится кнопка общего назначения с надписью *caption*, являющейся заголовком кнопки. Значение аргумента *caption* должно быть строковым. Написание скрипта для кнопки и изменение ее свойств реализуется через панель “Настройки”.

В следующем примере при создании скрипта для раскраски многоугольника нам придется работать с различными цветами. Имена основных предопределенных цветов, используемых в *GeoGebra*, приведены в табл. 3. Имена

ряда дополнительных цветов, которые также можно использовать при написании скриптов, перечислены в <https://wiki.geogebra.org/en/Reference:Colors>. Там можно видеть более сотни имен различных цветов, например, цвет *SeaGreen*, отсутствующий в табл. 3. Заметим, что в скриптах цвет должен указываться в англоязычной нотации.

Пример 70. Построить многоугольник *P* и для экспериментов с цветом *P* вывести два управляющих элемента:

Решение. Выполним следующую последовательность действий:


- ползунок, задающий величину непрозрачности g ($0 \leq g \leq 1$) для *P*;

Таблица 3

Предопределенные имена цветов (для указания в скриптах)

Имя цвета	Перевод	Имя цвета	Перевод
<i>Black</i>	Черный	<i>Lime</i>	Зеленовато-желтый
<i>Dark Gray</i>	Темно-серый	<i>Cyan</i>	Голубой
<i>Gray</i>	Серый	<i>Turquoise</i>	Бирюзовый
<i>Dark Blue</i>	Темно-голубой	<i>Light Blue</i>	Светло-голубой
<i>Blue</i>	Синий	<i>Aqua</i>	Цвет морской волны
<i>Dark Green</i>	Темно-зеленый	<i>Silver</i>	Серебрянный
<i>Green</i>	Зеленый	<i>Light Gray</i>	Светло-серый
<i>Maroon</i>	Темно-бордовый	<i>Pink</i>	Розовый
<i>Crimson</i>	Малиновый	<i>Violet</i>	Фиолетовый
<i>Red</i>	Красный	<i>Yellow</i>	Желтый
<i>Magenta</i>	Пурпурный	<i>Light Yellow</i>	Светло-желтый
<i>Indigo</i>	Сине-фиолетовый	<i>Light Orange</i>	Светло-оранжевый
<i>Purple</i>	Багровый	<i>Light Violet</i>	Светло-фиолетовый
<i>Brown</i>	Коричневый	<i>Light Purple</i>	Светло-фиолетовый
<i>Orange</i>	Оранжевый	<i>Light Green</i>	Светло-зеленый
<i>Gold</i>	Золотой	<i>White</i>	Белый


- кнопку, щелчки по которой при заданной непрозрачности g будут последовательно и циклически менять цвета *P* так: *Black*, *Dark Gray*, ..., *Gold*, *Lime*, ..., *White*, *Black*, ... (см. табл. 3);



- инструментом  *Многоугольник* выведем некоторый многоугольник, скажем, шестиугольник *ABCDEF* (имя *poly1*);

- через строку ввода определим переменную $t=32$ и список имен основных предопределенных цветов:

$lis=\{ "Black", "Dark Gray", "Gray", "Dark Blue", "Blue", "Dark Green", "Green", "Maroon", "Crimson", "Red", "Magenta", "Indigo", "Purple", "Brown", "Orange", "Gold", "Lime", "Cyan", "Turquoise", "Light Blue", "Aqua", "Silver", "Light Gray",$

"Pink", "Violet", "Yellow", "Light Yellow", "Light Orange", "Light Violet", "Light Purple", "Light Green", "White"};


- инструментом “ Текст” выведем текст, являющийся именем текущего цвета. Это имя может быть получено как результат выполнения команды $Element[lis, t]$. При формировании текста данную команду необходимо поместить в пустой бокс. Увеличим размер шрифта текста и перетащим его на полотно в позицию ниже прямоугольника P ;

- инструментом “ Кнопка” выведем кнопку общего назначения с заголовком “Цвет” и предопределенным изображением “”. Свяжем с кнопкой следующий скрипт, выполняющийся при каждом щелчке по этой кнопке:

$$t = \text{Mod}[t, 32] + 1$$

$$\text{SetColor}[poly1, Element[lis, t]]$$

Поясним работу написанного скрипта. В нем переменная t последовательно и циклически по модулю 32 будет увеличиваться на 1. Поскольку начальное значение $t=32$, то изменяться t будет так: 1, 2, ..., 32, 1, 2, Команда второй строки скрипта устанавливает для многоугольника $poly1$ цвет, имеющий имя $Element[lis, t]$. Но это есть элемент из позиции t ($t=1, 2, \dots$) в созданном ранее списке цветов lis . Таким образом каждый щелчок по кнопке будет раскрашивать многоугольник в “следующий” цвет;

- инструментом “ Ползунок” выведем ползунок g , задав для него диапазон $[0, 1]$, шаг 0.1, вертикальное направление, вывод метки в виде “имя=значение” и, наконец, скрипт по событию *on-update* в виде команды $SetFilling[poly1, g]$. Этот скрипт в реальном времени будет устанавливать непрозрачность многоугольника $poly1$ в значение g при всяком изменении g ;

- через строку ввода сформируем вертикально выводимый текст “Непрозрачность”, что можно сделать командой $RotateText["Непрозрачность", 90^\circ]$. Изменим размер шрифта текста и перетащим его в позицию слева от ползунка.

Модель для экспериментов с цветом многоугольника P построена (см. рис. 77). Щелкая мышью по кнопке “Цвет”, мы можем установить для P любой из 32 основных цветов табл. 3. Протаскивая движок ползунка, можно зафиксировать требуемую непрозрачность g ($g=0, 0.1, 0.2, \dots, 1$) для цвета. Можно также запустить анимацию по изменению непрозрачности для любого фиксированного цвета ($RClick(\text{ползунок})/Анимация$).

Замечания. 1. Как мы уже отмечали, в табл. 3 перечислены имена лишь основных предопределенных цветов *GeoGebra*. Но ничто не мешает нам включить в список lis один или более дополнительных цветов. Например, если добавить в список имя *SeaGreen*, и первую строку скрипта, связанного с кнопкой “Цвет”, изменить на $t = \text{Mod}[t, 33] + 1$, то новый цвет будет работать так же, как и все другие цвета.

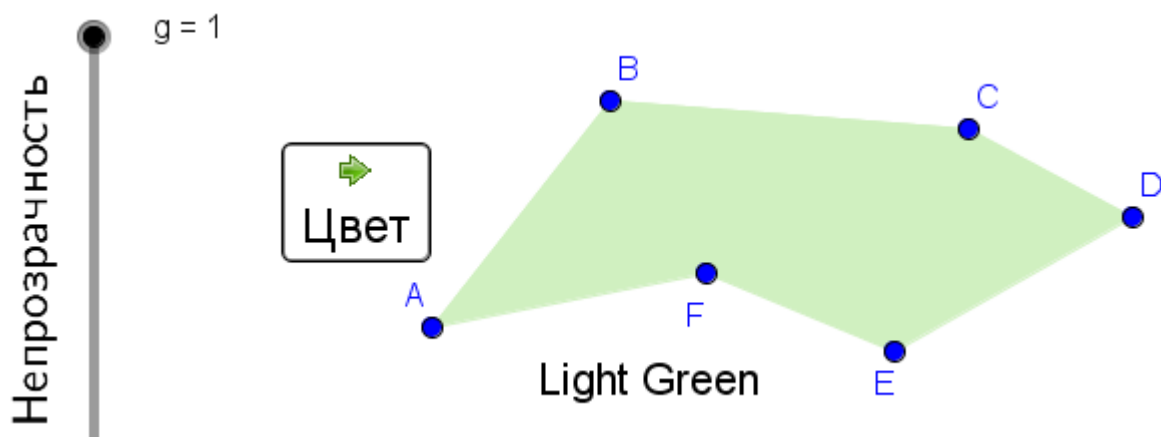


Рис. 77. Раскраска многоугольника основными цветами при различной непрозрачности g ($0 \leq g \leq 1$)

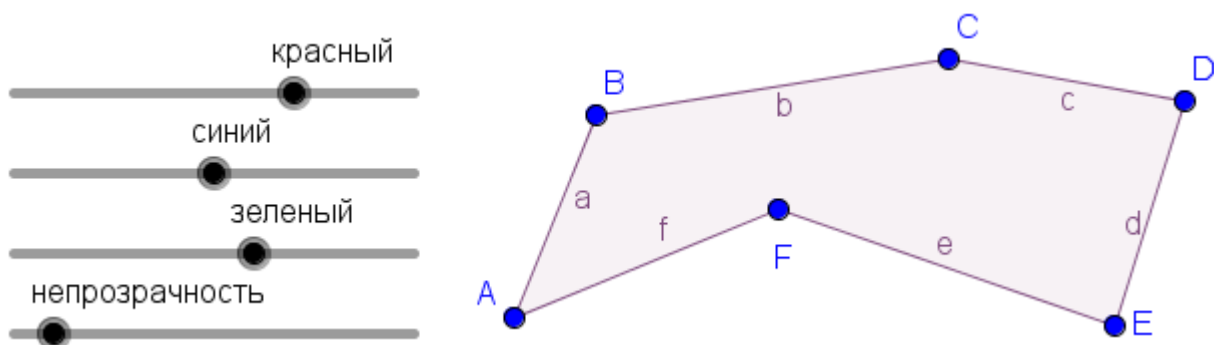



Рис. 78. Управление цветом и непрозрачностью объекта 4 ползунками

2. Если требуется выводить в документе не английские имена цветов, а русские, то наряду со списком *lis* следовало бы создать соответствующий список русских имен цветов – *lisru* = { Черный, ... }, ввести его через строку ввода, а сам текст формировать с помощью команды *Element[*lisru*, *t*]*.

3. Если требуется изменять цвет сторон многоугольника независимо от цвета его внутренней части, то сделать это можно так. Через строку ввода ввести новую переменную $s=32$. Инструментом “ Кнопка” вывести вторую кнопку общего назначения с заголовком “Цвет сторон” и связать с ней следующий скрипт, выполняющийся при каждом щелчке по этой кнопке:


```
s=Mod[s, 32]+1
SetColor[a, Element[lis, s]]
SetColor[b, Element[lis, s]]
SetColor[c, Element[lis, s]]
SetColor[d, Element[lis, s]]
SetColor[e, Element[lis, s]]
SetColor[f, Element[lis, s]]
```

После этих действий новая кнопка готова выполнять свою функцию.

4. В команде *SetColor[obj, col]* аргументы *obj* и *col* – это соответственно объект раскраски и имя некоторого predefined цвета. Допускается и другой синтаксис для этой команды – *SetColor[obj, r, g, b]*, где *r, g, b* ($0 \leq r, g, b \leq 1$) – доли красного, зеленого и синего цветов при раскраске объекта *obj* по *rgb*-схеме.

5. Команда *SetDynamicColor[obj, r, g, b, opacity]* аналогична команде *SetColor[obj, r, g, b]*, но в ней дополнительно последним аргументом указывается непрозрачность *opacity* ($0 \leq opacity \leq 1$). Это дает возможность четырьмя ползунками полностью управлять как цветом объекта *obj*, так и его непрозрачностью. Причем делается это при одном и том же скрипте для каждого из ползунков *r, s, t* и *u* – *SetDynamicColor[obj, r, s, t, u]* (см. рис. 78).



Окно ввода. Инструмент “ Окно ввода” используется для создания управляющих элементов, называемых окнами ввода или текстовыми полями, которые позволяют интерактивно изменять различные объекты, связанные с данными окнами. С одним окном можно связать один управляемый объект. При вставке окна ввода в документ открывается диалоговая панель “Окно ввода”, на которой можно сформировать выводимый справа от окна заголовок и задать управляемый объект (см. рис. 79 а). Если заголовок не задавать, то справа от выводимого окна ввода будет выводиться автоматически формируемое для нее имя: *окноВвода1*, *окноВвода2*, ... (*InputBox1*, *InputBox2*, ...). Управляемый объект должен быть выбран из раскрываемого одноименного списка панели “Окно ввода”. При щелчке по кнопке *Ok* или по ключу *Enter* панель “Окно ввода” закрывается и соответствующий управляющий элемент оказывается на полотне. Длину выводимого поля можно изменить через панель “Настройки”.

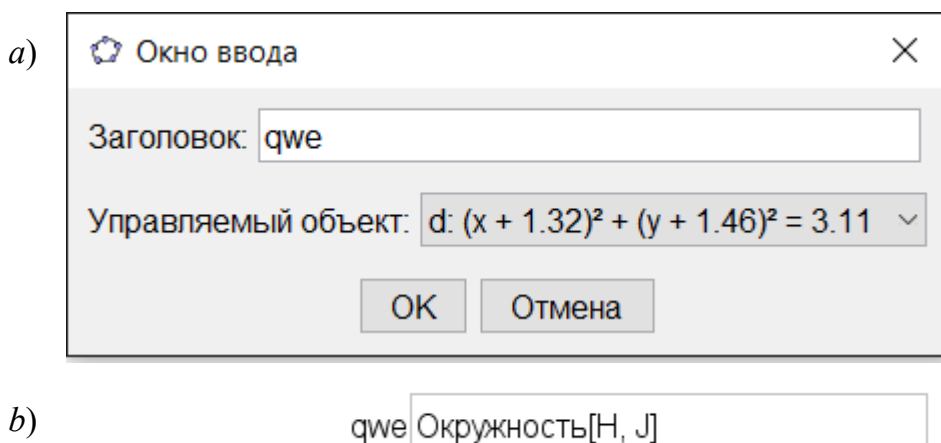






Рис. 79 (a, b). Панель для формирования окна ввода и полученное окно ввода

Дальнейшая правка содержания поля ввода (см. 73b) будет соответствующим образом воздействовать на связанный с ним объект, а непосредствен-

ные изменения связанного объекта – приводить к адекватной правке содержания поля ввода. При правке экземпляры объектов одного типа могут замещаться не только экземплярами объектов этого же типа, но и объектами иного типа. Например, запись функции $x \cdot \sin(x)$ можно превратить в запись функции $x^2 + x$, в команду вывода треугольника – $Polygon[A, B, C]$, в команду вывода окружности – $Circle[B, R]$ и т. д. Но в любом случае перед появлением нового объекта предыдущий объект удаляется.

Пример 71. Пусть на полотне выведены точки A, B, C, D, E . Сформировать два поля ввода для проведения экспериментов с ними. В первое из них должно вводиться неотрицательное число, а во второе – окружность с заданным центром и радиусом.

Решение. Выполним такие действия:

- инструментом “ Точка” выведем точки A, B, C, D, E ;
- через строку ввода сформируем переменную $r=1$;
- инструментом “ Окружность по центру и радиусу” выведем окружность с центром в точке C радиуса r ;
- инструментом “ Окно ввода” выведем поле ввода с заголовком r и свяжем его с объектом r ;
- инструментом “ Окно ввода” выведем поле ввода с заголовком $circ$ и свяжем его с выведенной окружностью (объектом $Circle[C, r]$);

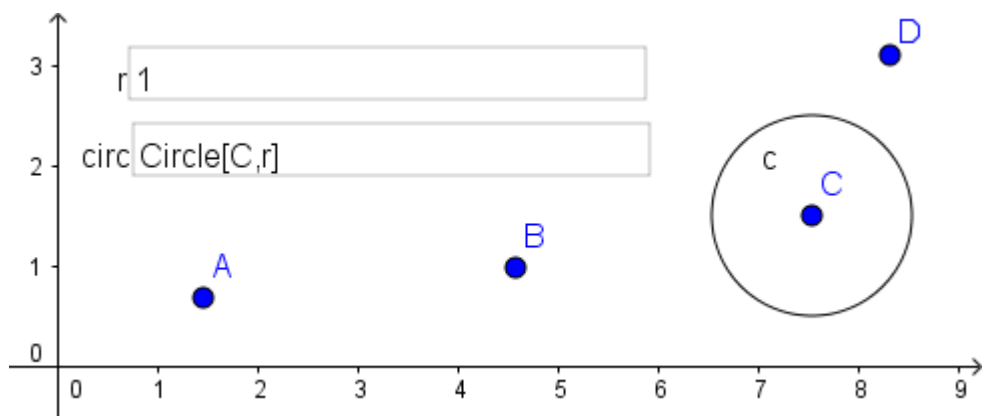


Рис. 80. Модель для экспериментов с полями ввода

Модель для экспериментов с полями ввода готова. Изменяя в первом поле величину r или во втором поле точку B или r (B на A, D, C, E ; r – любым положительным числом), будем видеть актуализацию изображения на панели “*Полотно*” и соответствующие изменения на панели “*Объекты*”. Заметим, что ничто не мешает нам ввести во второе поле, например, функцию $x + \cos(x)$, команду $Segment[A, B]$ и т. п.

A. <code>InputBox[]</code>
B. <code>InputBox[linked-obj]</code>



Создание поля ввода
А) По команде A формируется пустое поле ввода с надписью, которая является автоматически

формируемым именем поля: *окноВвода1*, *окноВвода2*, ... (*InputBox1*, *InputBox2*, ...) и располагается слева от него. В поле может быть введен любой объект. Изменение свойств поля ввода, в том числе и его длины, реализуется через панель “Настройки”.

В) Команда *B* выполняется как *A*, но в сформированном поле сразу указывается связанный с ним объект *linked-obj*.

3.4. Пример использования панели “Полотно 2”


Первая панель для вывода графики – “Полотно”, вторая – “Полотно 2”. Есть задачи, для решения которых целесообразно иметь дело сразу с обеими графическими панелями. Об одной такой задаче и пойдет речь ниже [47,48].


Для просмотра фрагментов документа можно использовать инструменты “ Увеличить” и “ Уменьшить”. Но делать это можно и иным путем, например, проецируя отдельные фрагменты документа на всю вторую графическую панель “Полотно 2”. Обсудим, как это может быть организовано.

Пример 72. Создать на панели “Полотно” объект *V*, называемый просмотрным окном. Окно *V* должно быть замкнутой прозрачной прямоугольной областью со сторонами, параллельными осям координат. Протаскиванием мышью *V* должно перемещаться по панели “Полотно”, а протаскиванием мышью вершин *V* – изменять размеры *V*. Далее отдельные фрагменты панели “Полотно”, видимые через *V*, то есть попавшие внутрь *V*, должны оперативно отображаться на всей второй графической панели “Полотно 2”.

Решение. В [47, 48] предложено одно из возможных решений поставленной задачи. Ниже демонстрируются три решения этой задачи, в том числе и решение из [47, 48]. Правда, все они отличаются друг от друга лишь началом – построением просмотрного окна *V*. Итак, предложенная задача может быть решена следующей последовательностью действий:

1. В новом *ggb*-документе через меню командой “Настройки\Обозначения\Не вводить” сформируем установку, в соответствии с которой обозначения создаваемых в дальнейшем объектов выводиться не будут;

2. Инструментом “ Точка” создадим две свободные точки. Их имена *A* и *B* на чертеже не выводятся. Через строку ввода сформируем по ним две зависимые точки ($x[B]$, $y[A]$) и ($x[A]$, $y[B]$);

3. Инструментом “ Многоугольник” соединим выведенные точки, формируя прямоугольник с вершинами в этих точках;

4. Через контекстное меню командами “*RClick(точка)\Показывать объект*” зависимые точки сделаем невидимыми;

5. Через контекстное меню командой “*Rclick(прямоугольник)\Свойства*” откроем панель установок “*Настройки*”, и на странице “*Стиль*” включим независимый переключатель “*Инвертировать заливку*”. При необходимости на странице “*Цвет*” изменим цвет и прозрачность заливки. Фактически, просмотрное окно *V* полностью готово для использования. Изменять размеры *V* можно протаскиванием вершин, являющихся свободными точками, а перемещать *V* надо так: щелкнуть левой кнопкой мыши рядом с *V*, но вне *V*, и, не отпуская кнопку, протаскивать *V* по полотну. Заметим, что если инвертирование заливки не производилось, то щелкать надо не вне, а внутри *V*;

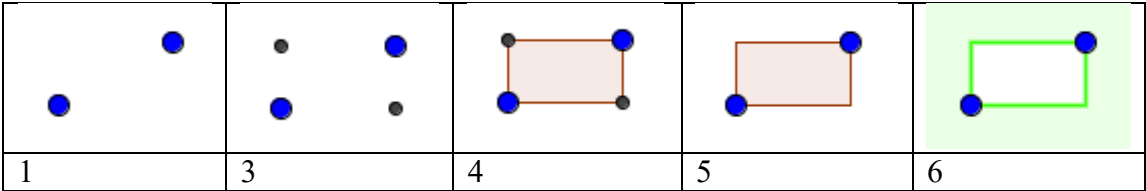


Рис. 81. Этапы построения смотрового окна *V*

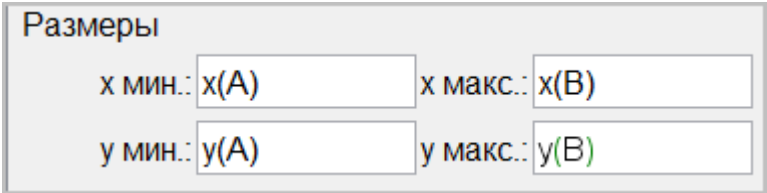


Рис. 82. Связывание размеров смотрового окна *V* и окна панели “*Полотно 2*”

6. Через меню командой “*Вид\Полотно 2*” откроем вторую графическую панель “*Полотно 2*”. Передвигая линию, разделяющую панели “*Полотно*” и “*Полотно 2*”, установим между ними подходящую границу;

7. Через контекстное меню командой “*RClick(Полотно 2)\Полотно*” откроем панель “*Настройки*” и реализуем следующие установки для панели “*Полотно 2*”:

- на страницах “*Ось абсцисс*” и “*Ось ординат*” включим независимые переключатели “*Прикрепить к границам*”. Это обеспечит рисование осей координат у границ окна;
- на странице “*Основные*” зададим максимальные и минимальные значения по осям координат. Если эти значения связать с параметрами просмотрного окна *V* так, как это показано на рис. 82, то фактически будет установлено взаимно однозначное соответствие между точками *V* и точками *W* – окна панели “*Полотно 2*”, причем изменение размеров *V* будет автоматически приводить к соответствующему изменению масштаба в *W*, а перемещение *V* – к равномерному смещению меток на осях координат в *W*;


8. Через меню командой “*Настройки\Обозначения\Не вводить*” сформируем установку, в соответствии с которой в дальнейшем обозначения создаваемых объектов выводиться будут;

9. Сформируем на панели “*Полотно*” какое-либо изображение. Например, выведем:

- график функции одной переменной $f(x) = 1 + \sin[(x-2)^{4/5}]$ на отрезке $[0,9]$ вводом выражения $f(x)=Function[1+\sin[(x-2)^(4/5)], x, 0, 9];$
- график параметрически заданной функции

$$\begin{cases} x = 3 + 2 \cos(5t) \\ y = 2 + 3 \sin(3t) \end{cases} \quad (0 \leq t \leq 2\pi)$$

вводом выражения $g(t)=Curve[3+2*\cos(5*t), 2+3*\sin(3*t), t, 0, 2*\pi];$

- график неявно заданной функции $(x-7)^4 + (y-3)^3 - 2 = 0$ вводом выражения $ImplicitCurve[(x-7)^4+(y-3)^3-2];$
- окружность инструментом “ Окружность по центру и точке”;

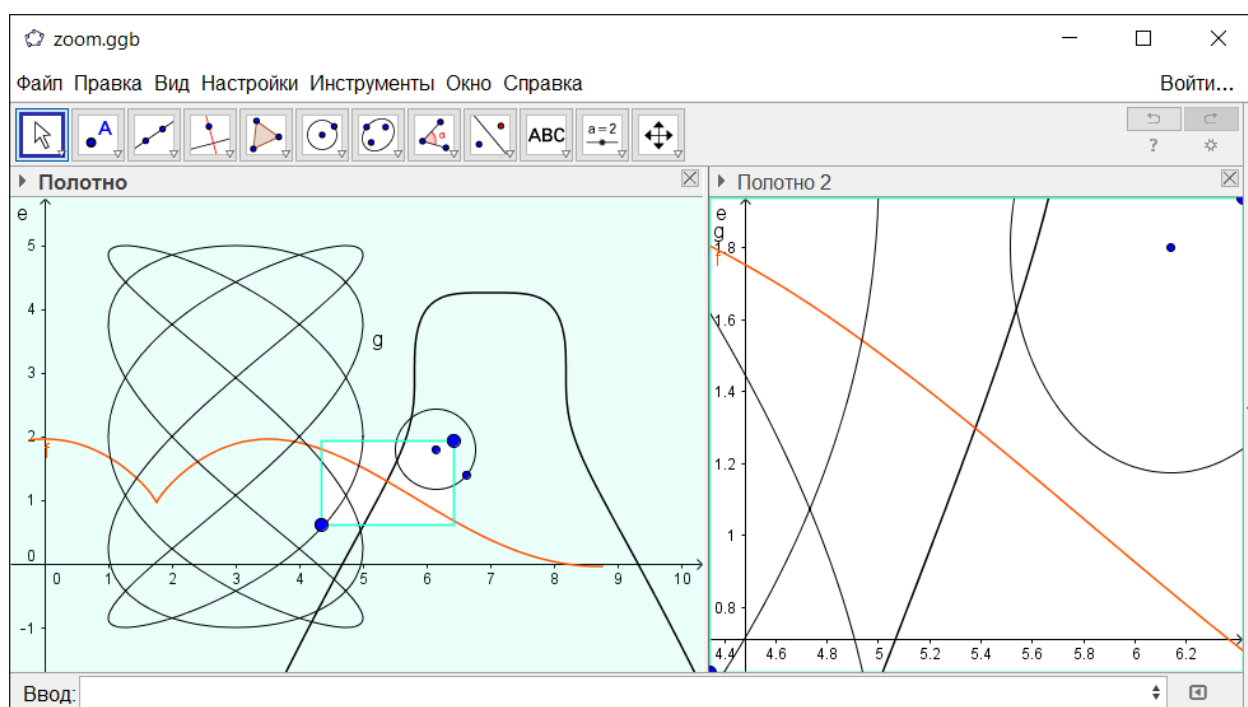




Рис. 83. Работа с созданным смотровым окном


10. Установим видимость всех или некоторых объектов панели “Полотно” и на второй графической панели “Полотно 2”. Для этого через контекстное меню последовательно выполним серию команд “*RClick(объект)\Свойства\Дополнительно*☒ *Полотно 2*”, то есть включим независимые переключатели для вывода соответствующих объектов на панели “Полотно 2”;

11. Теперь можно экспериментировать. Для перемещения смотрового окна V курсор следует подвести к границе V с внешней стороны V и нажать левую кнопку мыши. Когда курсор примет вид “”, окно V можно протаскивать по полотну и наблюдать изображение из V в окне панели “Полотно 2”. Протаскиванием угловых точек окна V можно изменять его размеры.

4. Вставка текстов и формул

Для создания текстов и формул в документах *GeoGebra* имеется инструмент “ Текст” и ряд специальных команд. Можно также формировать тексты простым перетаскиванием описаний объектов из панели “Объекты” на панель “Полотно”. В дальнейшем под текстами мы будем подразумевать и традиционные тексты, и произвольные математические выражения (формулы). Обычно тексты в *GeoGebra* имеют небольшой размер и используются для вставки надписей и комментариев к геометрическим или иным объектам документа. Сами они также являются объектами. Создаются тексты на специальной панели в обычном режиме или в режиме “*LaTeX*-формула”. В первом случае текст вводится непосредственным набором составляющих его символов и затем как есть переносится в документ, во втором случае текст формируется на некотором подмножестве языка научной издательской системы *LaTeX2_ε* [4, 12], интерпретируется в реальном времени в окне предпросмотра, а затем переносится в документ. Еще раз подчеркнем, что доступными являются далеко не все средства *LaTeX2_ε*, например, нельзя пользоваться многими окружениями и командами даже ядра системы и тем более открывать пакеты расширений. Однако то, что доступно, фактически можно использовать без знания синтаксиса и семантики языка, работая с входящей в систему надстройкой, которая позволяет простыми нажатиями левой кнопки мыши не только вводить основные конструкции языка или их шаблоны, но также способствует ненавязчивому фоновому освоению основ языка. Указанная надстройка является упрощенным аналогом графической оболочки *LyX* для *LaTeX2_ε* [16]. Заметим, что тексты *GeoGebra* могут содержать *html*-символы Юникода.

4.1. Формирование текстов инструментом “ Текст”

Инструмент “ Текст” – это основное средство для создания текстов и для работы с ним необходимо щелкнуть по пиктограмме инструмента, а затем по позиции в документе, с которой сформированный текст будет располагаться. Эти действия приводят к открытию панели “Текст”, показанной на рис. 1. Протаскиванием мышью сторон или углов этой панели можно менять ее размеры.

4.1.1. Управляющие элементы панели “Текст”

На панели “Текст” (см. рис. 1) мы видим следующие управляющие элементы:

- 1) поле “Правка” для ввода текста с клавиатуры или с помощью кнопок раскрываемых вложенных списков “Символы” и “LaTeX-формула”;
- 2) поле “Предпросмотр” для просмотра в реальном времени вводимого текста или результата интерпретации формируемого *LaTeX*-кода;
- 3) независимый переключатель “LaTeX-формула”. Если он включен, то вводимый в поле “Правка” текст будет считаться *LaTeX*-кодом, и в поле “Предпросмотр” мы будем видеть его интерпретацию, то есть вид, который он будет иметь в документе;
- 4) кнопка “Help” для перехода к *online* руководству по системе *GeoGebra* для получения справок;
- 5) список “Символы”, предоставляющий большой набор символов Юникода, разбитых на 13 категорий, которые представлены кнопочными подсписками (см. табл. 1). В поле “Правка” любой из этих символов вводится щелчком левой кнопки мыши по его пиктограмме;

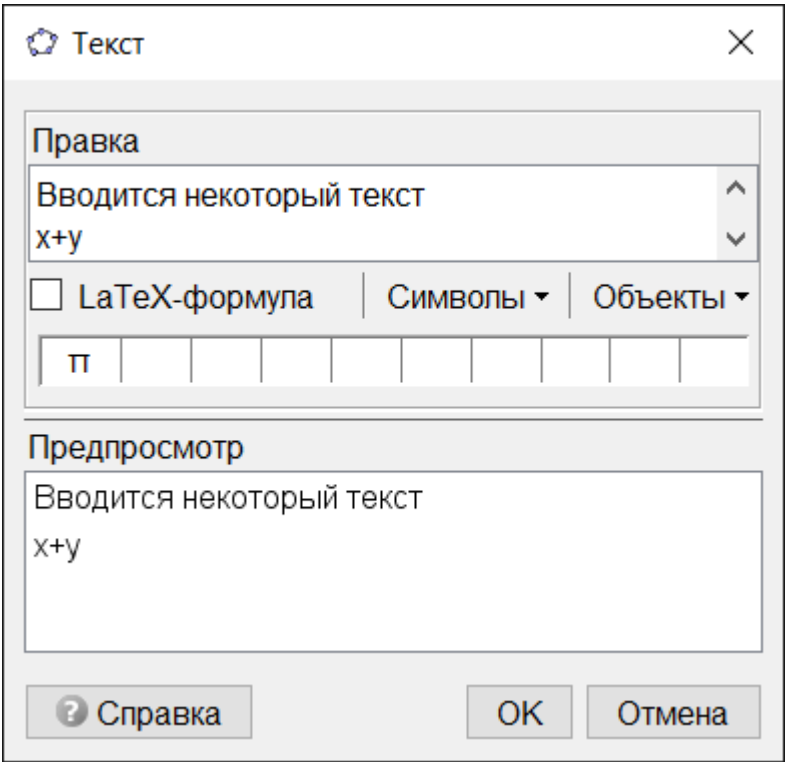



Рис. 1. Панель “Текст”

- 6) список “LaTeX-формула”, доступный лишь при включенном одноименном независимом переключателе. Используя любой из его 11 кнопочных подсписков, можно вводить выражения или их части. При этом в поле “Правка”

будут создаваться реальные команды *LaTeX*, запись которых начинается с обратного слэша “\”, а в поле “*Предпросмотр*” можно видеть соответствующие им выражения (см. табл. 2).

7) окно “” со списком десяти последних уникальных символов, которые уже вводились в текущей сессии. Символ π находится в этом окне до начала ввода. Каждый вводимый и отсутствующий в списке символ помещается в начало списка со сдвигом имеющихся в нем элементов вправо;

8) список “*Объекты*” для ввода возвращаемых объектами значений. В этот список помещены имена всех объектов, которые уже созданы в текущем документе. При щелчке по имени объекта оно помещается в поле “*Правка*” в специальный бокс (рамку). В формируемый текст попадает значение этого имени, что можно видеть в окне “*Предпросмотр*”. Если требуется ввести некоторое выражение, в котором имеются имена объектов, то делать это надо следующим образом: сначала в списке “*Объекты*” нажать на выбор “*Пустая рамка*”, а затем в появившемся в поле “*Правка*” пустом боксе набрать выражение;

9) кнопки “*Отмена*” или “ \times ” для закрытия панели “*Текст*” с отменой всех проведенных действий и возврата в документ;

10) кнопка *Ok* для закрытия панели “*Текст*” с переносом созданного текста в документ. Этот текст переносится на панель “*Полотно*”, а на панели “*Объекты*” он получает имя в виде “*надпись1*”, “*надпись2*”, ... (*text1*, *text2*, ...). В дальнейшем созданные текстовые объекты могут быть отформатированы на панели “*Настройки*”. Там же можно изменить и их имена.

4.1.2. Символы Юникода списка “*Символы*”

В списке “*Символы*” панели “*Текст*” представлены 13 именованных кнопочных подсписков с символами Юникода (см. табл. 1). В поле “*Правка*” любой из этих символов вводится простым щелчком левой кнопки мыши по его пиктограмме. Набранные символы переносятся в документ щелчком по кнопке *Ok* как обычный текст или как интерпретация *LaTeX*-текста в зависимости от того, выключен или включен независимый переключатель *LaTeX*-текст. В последнем случае пробелы между обычными символами, вообще говоря, игнорируются. А около бинарных операторов (+, -, ...) слева и справа сохраняется по одному широкому пробелу, а если их не было, то они вставляются. Для принудительной вставки пробелов между символами можно использовать одну или более команд “*\пробел*” (без кавычек), каждая из которых форсирует добавление одного широкого пробела. Сказанное о пробелах можно понять на приведенных ниже примерах, где обычный текст слева выведен так, как был набран, а справа он же выведен при включенном переключателе *LaTeX*-текст:

Обычный текст

ябл око, помидор, ★ ∇↓, ...

сумма $x+y$

сумма\ $x+y$

LaTeX – текст

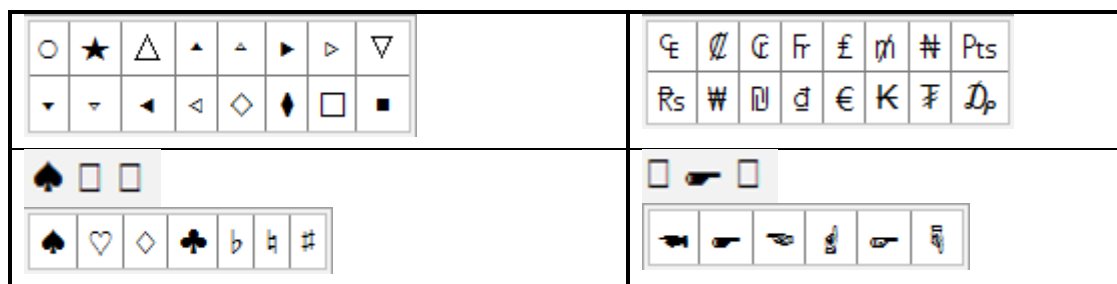
яблоко, помидор, ★∇↓, ...

сумма $x + y$

сумма $x + y$

Таблица 1

Группы символов Юникода, вводимых с панели “Текст”																																																																																																	
<div>Основные</div> <table><tr><td>α</td><td>β</td><td>γ</td><td>δ</td><td>ε</td><td>ζ</td><td>η</td><td>θ</td><td>κ</td><td>λ</td><td>μ</td></tr><tr><td>ξ</td><td>ρ</td><td>σ</td><td>τ</td><td>φ</td><td>ϕ</td><td>χ</td><td>ψ</td><td>ω</td><td>Γ</td><td>Δ</td></tr><tr><td>Θ</td><td>Π</td><td>Σ</td><td>Φ</td><td>Ω</td><td>∞</td><td>⊗</td><td>≈</td><td>≠</td><td>≤</td><td>≥</td></tr><tr><td>¬</td><td>∧</td><td>∨</td><td>→</td><td>∥</td><td>⊥</td><td>∈</td><td>⊆</td><td>⊂</td><td>⊄</td><td>²</td></tr><tr><td>³</td><td>°</td><td>í</td><td>π</td><td>e</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	α	β	γ	δ	ε	ζ	η	θ	κ	λ	μ	ξ	ρ	σ	τ	φ	ϕ	χ	ψ	ω	Γ	Δ	Θ	Π	Σ	Φ	Ω	∞	⊗	≈	≠	≤	≥	¬	∧	∨	→	∥	⊥	∈	⊆	⊂	⊄	²	³	°	í	π	e							<div>x ÷ −</div> <table><tr><td>×</td><td>÷</td><td>−</td><td>·</td><td>◦</td><td>•</td><td>±</td><td>∓</td></tr><tr><td>√</td><td>≠</td><td>≤</td><td>≥</td><td>≈</td><td>~</td><td>≈</td><td>≅</td></tr><tr><td>≠</td><td>∞</td><td>∠</td><td>⊄</td><td>⊥</td><td>∥</td><td>⊥</td><td></td></tr><tr><td>⊕</td><td>⊖</td><td>⊗</td><td>⊘</td><td>⊙</td><td></td><td></td><td></td></tr></table>	×	÷	−	·	◦	•	±	∓	√	≠	≤	≥	≈	~	≈	≅	≠	∞	∠	⊄	⊥	∥	⊥		⊕	⊖	⊗	⊘	⊙												
α	β	γ	δ	ε	ζ	η	θ	κ	λ	μ																																																																																							
ξ	ρ	σ	τ	φ	ϕ	χ	ψ	ω	Γ	Δ																																																																																							
Θ	Π	Σ	Φ	Ω	∞	⊗	≈	≠	≤	≥																																																																																							
¬	∧	∨	→	∥	⊥	∈	⊆	⊂	⊄	²																																																																																							
³	°	í	π	e																																																																																													
×	÷	−	·	◦	•	±	∓																																																																																										
√	≠	≤	≥	≈	~	≈	≅																																																																																										
≠	∞	∠	⊄	⊥	∥	⊥																																																																																											
⊕	⊖	⊗	⊘	⊙																																																																																													
<div>Α Β Γ</div> <table><tr><td>Α</td><td>Β</td><td>Γ</td><td>Δ</td><td>Ε</td><td>Ζ</td><td>Η</td><td>Θ</td></tr><tr><td>Ι</td><td>Κ</td><td>Λ</td><td>Μ</td><td>Ν</td><td>Ξ</td><td>Ο</td><td>Π</td></tr><tr><td>Ρ</td><td>Σ</td><td>Τ</td><td>Υ</td><td>Φ</td><td>Χ</td><td>Ψ</td><td>Ω</td></tr><tr><td>α</td><td>β</td><td>γ</td><td>δ</td><td>ε</td><td>ζ</td><td>η</td><td>θ</td></tr><tr><td>ι</td><td>κ</td><td>λ</td><td>μ</td><td>ν</td><td>ξ</td><td>ο</td><td>π</td></tr><tr><td>ρ</td><td>σ</td><td>τ</td><td>υ</td><td>φ</td><td>χ</td><td>ψ</td><td>ω</td></tr><tr><td>φ</td><td>ε</td><td>θ</td><td>ς</td><td></td><td></td><td></td><td></td></tr></table>	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	φ	ε	θ	ς					<div>Σ ∂ ∇</div> <table><tr><td>Σ</td><td>∂</td><td>∇</td><td>Δ</td><td>Π</td><td>∏</td><td>∫</td><td>∫</td></tr><tr><td>∫</td><td>∫</td><td>∞</td><td></td><td></td><td></td><td></td><td></td></tr></table> <div>∅ ∩ ∪</div> <table><tr><td>∅</td><td>∩</td><td>∪</td><td>∈</td><td>∉</td><td>⊂</td><td>⊄</td><td>⊆</td></tr><tr><td>⊄</td><td>⊃</td><td>⊄</td><td>≅</td><td>≠</td><td>⊂</td><td>⊄</td><td>⊆</td></tr><tr><td>ℝ</td><td>ℤ</td><td>ℕ</td><td>ℕ</td><td>ℕ</td><td>ℕ</td><td>ℕ</td><td>ℕ</td></tr></table>	Σ	∂	∇	Δ	Π	∏	∫	∫	∫	∫	∞						∅	∩	∪	∈	∉	⊂	⊄	⊆	⊄	⊃	⊄	≅	≠	⊂	⊄	⊆	ℝ	ℤ	ℕ	ℕ	ℕ	ℕ	ℕ	ℕ
Α	Β	Γ	Δ	Ε	Ζ	Η	Θ																																																																																										
Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π																																																																																										
Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω																																																																																										
α	β	γ	δ	ε	ζ	η	θ																																																																																										
ι	κ	λ	μ	ν	ξ	ο	π																																																																																										
ρ	σ	τ	υ	φ	χ	ψ	ω																																																																																										
φ	ε	θ	ς																																																																																														
Σ	∂	∇	Δ	Π	∏	∫	∫																																																																																										
∫	∫	∞																																																																																															
∅	∩	∪	∈	∉	⊂	⊄	⊆																																																																																										
⊄	⊃	⊄	≅	≠	⊂	⊄	⊆																																																																																										
ℝ	ℤ	ℕ	ℕ	ℕ	ℕ	ℕ	ℕ																																																																																										
<div>∇ ∃ ∄</div> <table><tr><td>∇</td><td>∃</td><td>∄</td><td>≈</td><td>≅</td><td>≠</td><td>∧</td><td>∨</td></tr><tr><td>⊕</td><td>≈</td><td>≈</td><td>⊥</td><td>⊥</td><td>⊂</td><td>∴</td><td>∴</td></tr></table>	∇	∃	∄	≈	≅	≠	∧	∨	⊕	≈	≈	⊥	⊥	⊂	∴	∴	<div>0 1 2</div> <table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>+</td><td>-</td><td>=</td><td>(</td><td>)</td><td>π</td><td>°</td><td></td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>+</td><td>-</td><td>=</td><td>(</td><td>)</td><td></td><td></td><td></td><td></td><td></td></tr></table>	0	1	2	3	4	5	6	7	8	9	+	-	=	()	π	°				0	1	2	3	4	5	6	7	8	9	+	-	=	()																																													
∇	∃	∄	≈	≅	≠	∧	∨																																																																																										
⊕	≈	≈	⊥	⊥	⊂	∴	∴																																																																																										
0	1	2	3	4	5	6	7	8	9																																																																																								
+	-	=	()	π	°																																																																																											
0	1	2	3	4	5	6	7	8	9																																																																																								
+	-	=	()																																																																																													
<div>← ↑ →</div> <table><tr><td>←</td><td>↑</td><td>→</td><td>↓</td><td>↔</td><td>↕</td><td>↖</td><td>↗</td></tr><tr><td>↘</td><td>↙</td><td>⇐</td><td>⇑</td><td>⇒</td><td>⇓</td><td>⇔</td><td>⇕</td></tr></table>	←	↑	→	↓	↔	↕	↖	↗	↘	↙	⇐	⇑	⇒	⇓	⇔	⇕	<div>↔ ↗ ↘</div> <table><tr><td>↔</td><td>↗</td><td>↘</td><td>↙</td><td>↖</td><td>↗</td><td>↘</td><td>↙</td></tr><tr><td>↗</td><td>↘</td><td>↙</td><td>↖</td><td>↗</td><td>↘</td><td>↙</td><td>↖</td></tr><tr><td>↖</td><td>↗</td><td>↘</td><td>↙</td><td>↖</td><td>↗</td><td>↘</td><td>↙</td></tr><tr><td>↙</td><td>↖</td><td>↗</td><td>↘</td><td>↙</td><td>↖</td><td>↗</td><td>↘</td></tr><tr><td>↘</td><td>↙</td><td>↖</td><td>↗</td><td>↘</td><td>↙</td><td>↖</td><td>↗</td></tr></table>	↔	↗	↘	↙	↖	↗	↘	↙	↗	↘	↙	↖	↗	↘	↙	↖	↖	↗	↘	↙	↖	↗	↘	↙	↙	↖	↗	↘	↙	↖	↗	↘	↘	↙	↖	↗	↘	↙	↖	↗																																								
←	↑	→	↓	↔	↕	↖	↗																																																																																										
↘	↙	⇐	⇑	⇒	⇓	⇔	⇕																																																																																										
↔	↗	↘	↙	↖	↗	↘	↙																																																																																										
↗	↘	↙	↖	↗	↘	↙	↖																																																																																										
↖	↗	↘	↙	↖	↗	↘	↙																																																																																										
↙	↖	↗	↘	↙	↖	↗	↘																																																																																										
↘	↙	↖	↗	↘	↙	↖	↗																																																																																										
<div>□ ★ △</div>	<div>€ ∅ Ⓒ</div>																																																																																																



Если переключатель “*LaTeX*-формула” включен, то есть работа ведется в *LaTeX*-режиме, то при вводе текста полезными могут быть приведенные ниже команды вставки разнообразных пробелов и ряд иных команд:

- `\пробел` – вставка широкого пробела (см. выше);
- `\;` – вставка толстого пробела (чуть меньше широкого пробела);
- `\:` – вставка среднего пробела;
- `\,` – вставка тонкого пробела;
- `\quad` – вставка пробела шириной, как у буквы *M* (1 *em*);
- `\qquad` – вставка пробела шириной, как у 2 букв *M* (2 *em*);
- `\hspace{wi}` – вставка горизонтального пробела шириной в *wi* (с обязательным указанием единиц измерения – *cm*, *mm*, ...). Значение *wi* может быть как положительным, так и отрицательным;
- `\!` – тонкий отрицательный пробел (сдвиг содержания после “`\!`” влево);
- `\` – переход на другую строку;
- `\vspace{he}` – переход на другую строку со вставкой вертикального пробела высотой в *he* (с указанием единиц измерения – *cm*, *mm*, ...);
- `\символ` – ввод служебных знаков `$`, `&`, `%`, `_`, `~`, `{` и `}`, то есть, эти знаки вводятся как команды добавлением перед ними знака обратного слэша “`\`” (без кавычек);
- `\dots` или `\ldots` – ввод нижнего многоточия (...);
- `\cdots` – ввод среднего многоточия (⋯);
- `\vdots` – ввод вертикального многоточия (⋮);
- `\ddots` – ввод диагонального многоточия (⋱);
- `\backslash` – ввод обратного слэша (`\`);
- `\slash` – ввод слэша (`/`);
- `\|` – ввод двойной вертикальной линии (`||`). Одинарная вертикальная линия (`|`) вводится с клавиатуры.

Остановимся еще на таком вопросе. Если требуется ввести символ, отсутствующий в списках “Символы” панели “Текст” и не вводимый командами *LaTeX*, то сделать это можно командой `UnicodeToLetter[cod]` (*cod* – десятичный код символа). Например, командой `UnicodeToLetter[38642]` выводится

японский иероглиф 雲 (облако), а по команде `UnicodeToLetter[33590]` – китайский иероглиф 茶 (чай).

4.1.3. Список “*LaTeX*-формула” и его подписки

Пусть на панели “Текст” независимым переключателем включен *LaTeX*-режим. Тогда доступными становятся кнопочные подписки списка “*LaTeX*-формула”. Щелчками левой кнопки мыши по соответствующим пиктограммам этих подписков в окно “Правка” помещаются реальные *LaTeX*-команды, интерпретация которых сразу же видна в окне “Предпросмотр”. Соответствующие команды можно также набирать и непосредственно с клавиатуры. Более того, с клавиатуры можно вводить и ряд таких *LaTeX*-команд, которые через подписки списка “*LaTeX*-формула” создать невозможно. Эти и некоторые другие вопросы и обсуждаются в данном пункте.

В табл. 2 представлен вид каждого из подписков списка “*LaTeX*-формула”, *LaTeX*-команды, вводимые щелчками по соответствующим кнопкам этих подписков, и формируемые по командам выражения. В последующих подпунктах подробно обсуждаются возможности, предоставляемые некоторыми из указанных подписков.

Таблица 2

<i>LaTeX</i> -команды, создаваемые с помощью списка “ <i>LaTeX</i> -формула”		
Подсписок	Выражение	<i>LaTeX</i> -код
<div>Корни и дроби</div> <div><div><div><div>$\frac{a}{b}$</div><div>x^a</div><div>x_a</div><div>\sqrt{x}</div><div>$\sqrt[n]{x}$</div><div>$\binom{a}{b}$</div></div></div></div>	$\frac{a}{b}$	<code>\frac{a}{b}</code>
	x^a	<code>x^{a}</code>
	x_a	<code>x_{a}</code>
	\sqrt{x}	<code>\sqrt{x}</code>
	$\sqrt[n]{x}$	<code>\sqrt[n]{x}</code>
	$\binom{a}{b}$	<code>\binom{a}{b}</code>
<div>Суммы и интегралы</div> <div><div><div><div>\sum</div><div>\sum_a^b</div><div>\int</div><div>\int_a^b</div><div>\oint</div><div>\oint_a^b</div><div>$\lim_{x \rightarrow \infty}$</div></div></div></div>	\sum	<code>\sum{ }</code>
	\sum_a^b	<code>\sum_{a}^{b}{...}</code>
	\int	<code>\int{...}</code>
	\int_a^b	<code>\int_{a}^{b}{...}</code>

	\oint	<code>\oint{ }</code>
	\oint_a^b	<code>\oint_{a}^{b}{ }</code>
	$\lim_{x \rightarrow \infty}$	<code>\lim_{x \to \infty}</code>
Парные скобки, соразмерные высоте содержания <div><div>()</div><div>[]</div><div>{ }</div><div> </div></div>	<code>()</code>	<code>\left(... \right)</code>
	<code>[]</code>	<code>\left[... \right]</code>
	<code>{ }</code>	<code>\left\{ ... \right\}</code>
	<code> </code>	<code>\left ... \right </code>
Диакритические знаки <div><div><div>$\acute{x}$$\grave{x}$$\tilde{x}$$\bar{x}$$\breve{x}$$\check{x}$</div><div>$\hat{x}$$\vec{x}$$\dot{x}$$\ddot{x}$$\mathring{x}$</div></div></div>	\acute{x}	<code>\acute{x}</code>
	\grave{x}	<code>\grave{x}</code>
	\tilde{x}	<code>\tilde{x}</code>
	\bar{x}	<code>\bar{x}</code>
	\breve{x}	<code>\breve{x}</code>
	\check{x}	<code>\check{x}</code>
	\hat{x}	<code>\hat{x}</code>
	\vec{x}	<code>\vec{x}</code>
	\dot{x}	<code>\dot{x}</code>
	\ddot{x}	<code>\ddot{x}</code>
	\mathring{x}	<code>\mathring{x}</code>
	\overline{xx}	<code>\overline{xx}</code>
Диакритические знаки (доп.) <div><div><div>$\overline{xx}$$\underline{xx}$$\overbrace{xx}$$\underbrace{xx}$$\overleftarrow{xx}$$\overleftrightarrow{xx}$</div><div>$\overrightarrow{xx}$$\overleftarrow{xx}$$\overleftrightarrow{xx}$$\overleftarrow{\overrightarrow{xx}}$$\overrightarrow{\overleftarrow{xx}}$$\overleftrightarrow{\overleftrightarrow{xx}}$</div></div></div>	\underline{xx}	<code>\underline{xx}</code>
	\overbrace{xx}	<code>\overbrace{xx}</code>
	\underbrace{xx}	<code>\underbrace{xx}</code>
	\overleftarrow{xx}	<code>\overleftarrow{xx}</code>
	\overleftrightarrow{xx}	<code>\overleftrightarrow{xx}</code>
	\overrightarrow{xx}	<code>\overrightarrow{xx}</code>
	$\overleftarrow{\overrightarrow{xx}}$	<code>\overleftarrow{\overrightarrow{xx}}</code>
	$\overrightarrow{\overleftarrow{xx}}$	<code>\overrightarrow{\overleftarrow{xx}}</code>
	$\overleftrightarrow{\overleftrightarrow{xx}}$	<code>\overleftrightarrow{\overleftrightarrow{xx}}</code>
	\widetilde{xx}	<code>\widetilde{xx}</code>
	\widehat{xx}	<code>\widehat{xx}</code>
	$\begin{array}{ccc} a & b & c \end{array}$	<code>\begin{array}{ccc} a & b & c \end{array}</code>
	$a \ \& \ b \ \& \ c$	<code>a \& b \& c</code>

<table><tr><td>a</td><td>b</td><td>c</td></tr><tr><td>a</td><td>b</td><td>c</td></tr><tr><td>a</td><td>b</td><td>c</td></tr><tr><td>d</td><td>e</td><td>f</td></tr><tr><td>g</td><td>h</td><td>i</td></tr></table>	a	b	c	a	b	c	a	b	c	d	e	f	g	h	i		<code>\end{array}</code>																																						
a	b	c																																																					
a	b	c																																																					
a	b	c																																																					
d	e	f																																																					
g	h	i																																																					
	a b c	<code>\begin{array}{l}</code> <code>$a \\\mathbf{b} \\\mathbf{c}$</code> <code>\end{array}</code>																																																					
	a b c d	<code>\begin{array}{ll}</code> <code>$a \& b \\\mathbf{c \& d}$</code> <code>\end{array}</code>																																																					
	a b c d e f g h i	<code>\begin{array}{lll}</code> <code>$a \& b \& c \\\mathbf{d \& e \& f}$</code> <code>$g \& h \& i$</code> <code>\end{array}</code>																																																					
Готические латинские буквы	<table><tr><td>\mathfrak{A}</td><td>\mathfrak{B}</td><td>\mathfrak{C}</td><td>\mathfrak{D}</td><td>\mathfrak{E}</td><td>\mathfrak{F}</td><td>\mathfrak{G}</td><td>\mathfrak{H}</td><td>\mathfrak{I}</td><td>\mathfrak{J}</td><td>\mathfrak{K}</td><td>\mathfrak{L}</td><td>\mathfrak{M}</td></tr><tr><td>\mathfrak{N}</td><td>\mathfrak{O}</td><td>\mathfrak{P}</td><td>\mathfrak{Q}</td><td>\mathfrak{R}</td><td>\mathfrak{S}</td><td>\mathfrak{T}</td><td>\mathfrak{U}</td><td>\mathfrak{V}</td><td>\mathfrak{W}</td><td>\mathfrak{X}</td><td>\mathfrak{Y}</td><td>\mathfrak{Z}</td></tr><tr><td>\mathfrak{a}</td><td>\mathfrak{b}</td><td>\mathfrak{c}</td><td>\mathfrak{d}</td><td>\mathfrak{e}</td><td>\mathfrak{f}</td><td>\mathfrak{g}</td><td>\mathfrak{h}</td><td>\mathfrak{i}</td><td>\mathfrak{j}</td><td>\mathfrak{k}</td><td>\mathfrak{l}</td><td>\mathfrak{m}</td></tr><tr><td>\mathfrak{n}</td><td>\mathfrak{o}</td><td>\mathfrak{p}</td><td>\mathfrak{q}</td><td>\mathfrak{r}</td><td>\mathfrak{s}</td><td>\mathfrak{t}</td><td>\mathfrak{u}</td><td>\mathfrak{v}</td><td>\mathfrak{w}</td><td>\mathfrak{x}</td><td>\mathfrak{y}</td><td>\mathfrak{z}</td></tr></table>	\mathfrak{A}	\mathfrak{B}	\mathfrak{C}	\mathfrak{D}	\mathfrak{E}	\mathfrak{F}	\mathfrak{G}	\mathfrak{H}	\mathfrak{I}	\mathfrak{J}	\mathfrak{K}	\mathfrak{L}	\mathfrak{M}	\mathfrak{N}	\mathfrak{O}	\mathfrak{P}	\mathfrak{Q}	\mathfrak{R}	\mathfrak{S}	\mathfrak{T}	\mathfrak{U}	\mathfrak{V}	\mathfrak{W}	\mathfrak{X}	\mathfrak{Y}	\mathfrak{Z}	\mathfrak{a}	\mathfrak{b}	\mathfrak{c}	\mathfrak{d}	\mathfrak{e}	\mathfrak{f}	\mathfrak{g}	\mathfrak{h}	\mathfrak{i}	\mathfrak{j}	\mathfrak{k}	\mathfrak{l}	\mathfrak{m}	\mathfrak{n}	\mathfrak{o}	\mathfrak{p}	\mathfrak{q}	\mathfrak{r}	\mathfrak{s}	\mathfrak{t}	\mathfrak{u}	\mathfrak{v}	\mathfrak{w}	\mathfrak{x}	\mathfrak{y}	\mathfrak{z}	\mathfrak{A} \mathfrak{B} ... \mathfrak{Z} \mathfrak{a} ... \mathfrak{z}	<code>\mathfrak{A}</code> <code>\mathfrak{B}</code> ... <code>\mathfrak{Z}</code> <code>\mathfrak{a}</code> ... <code>\mathfrak{z}</code>
\mathfrak{A}	\mathfrak{B}	\mathfrak{C}	\mathfrak{D}	\mathfrak{E}	\mathfrak{F}	\mathfrak{G}	\mathfrak{H}	\mathfrak{I}	\mathfrak{J}	\mathfrak{K}	\mathfrak{L}	\mathfrak{M}																																											
\mathfrak{N}	\mathfrak{O}	\mathfrak{P}	\mathfrak{Q}	\mathfrak{R}	\mathfrak{S}	\mathfrak{T}	\mathfrak{U}	\mathfrak{V}	\mathfrak{W}	\mathfrak{X}	\mathfrak{Y}	\mathfrak{Z}																																											
\mathfrak{a}	\mathfrak{b}	\mathfrak{c}	\mathfrak{d}	\mathfrak{e}	\mathfrak{f}	\mathfrak{g}	\mathfrak{h}	\mathfrak{i}	\mathfrak{j}	\mathfrak{k}	\mathfrak{l}	\mathfrak{m}																																											
\mathfrak{n}	\mathfrak{o}	\mathfrak{p}	\mathfrak{q}	\mathfrak{r}	\mathfrak{s}	\mathfrak{t}	\mathfrak{u}	\mathfrak{v}	\mathfrak{w}	\mathfrak{x}	\mathfrak{y}	\mathfrak{z}																																											
Рукописные латинские буквы	<table><tr><td>\mathcal{A}</td><td>\mathcal{B}</td><td>\mathcal{C}</td><td>\mathcal{D}</td><td>\mathcal{E}</td><td>\mathcal{F}</td><td>\mathcal{G}</td><td>\mathcal{H}</td><td>\mathcal{I}</td><td>\mathcal{J}</td><td>\mathcal{K}</td><td>\mathcal{L}</td><td>\mathcal{M}</td></tr><tr><td>\mathcal{N}</td><td>\mathcal{O}</td><td>\mathcal{P}</td><td>\mathcal{Q}</td><td>\mathcal{R}</td><td>\mathcal{S}</td><td>\mathcal{T}</td><td>\mathcal{U}</td><td>\mathcal{V}</td><td>\mathcal{W}</td><td>\mathcal{X}</td><td>\mathcal{Y}</td><td>\mathcal{Z}</td></tr></table>	\mathcal{A}	\mathcal{B}	\mathcal{C}	\mathcal{D}	\mathcal{E}	\mathcal{F}	\mathcal{G}	\mathcal{H}	\mathcal{I}	\mathcal{J}	\mathcal{K}	\mathcal{L}	\mathcal{M}	\mathcal{N}	\mathcal{O}	\mathcal{P}	\mathcal{Q}	\mathcal{R}	\mathcal{S}	\mathcal{T}	\mathcal{U}	\mathcal{V}	\mathcal{W}	\mathcal{X}	\mathcal{Y}	\mathcal{Z}	\mathcal{A} \mathcal{B} ... \mathcal{Z}	<code>\mathcal{A}</code> <code>\mathcal{B}</code> ... <code>\mathcal{Z}</code>																										
\mathcal{A}	\mathcal{B}	\mathcal{C}	\mathcal{D}	\mathcal{E}	\mathcal{F}	\mathcal{G}	\mathcal{H}	\mathcal{I}	\mathcal{J}	\mathcal{K}	\mathcal{L}	\mathcal{M}																																											
\mathcal{N}	\mathcal{O}	\mathcal{P}	\mathcal{Q}	\mathcal{R}	\mathcal{S}	\mathcal{T}	\mathcal{U}	\mathcal{V}	\mathcal{W}	\mathcal{X}	\mathcal{Y}	\mathcal{Z}																																											
Ажурные латинские буквы	<table><tr><td>\mathbb{A}</td><td>\mathbb{B}</td><td>\mathbb{C}</td><td>\mathbb{D}</td><td>\mathbb{E}</td><td>\mathbb{F}</td><td>\mathbb{G}</td><td>\mathbb{H}</td><td>\mathbb{I}</td><td>\mathbb{J}</td><td>\mathbb{K}</td><td>\mathbb{L}</td><td>\mathbb{M}</td></tr><tr><td>\mathbb{N}</td><td>\mathbb{O}</td><td>\mathbb{P}</td><td>\mathbb{Q}</td><td>\mathbb{R}</td><td>\mathbb{S}</td><td>\mathbb{T}</td><td>\mathbb{U}</td><td>\mathbb{V}</td><td>\mathbb{W}</td><td>\mathbb{X}</td><td>\mathbb{Y}</td><td>\mathbb{Z}</td></tr></table>	\mathbb{A}	\mathbb{B}	\mathbb{C}	\mathbb{D}	\mathbb{E}	\mathbb{F}	\mathbb{G}	\mathbb{H}	\mathbb{I}	\mathbb{J}	\mathbb{K}	\mathbb{L}	\mathbb{M}	\mathbb{N}	\mathbb{O}	\mathbb{P}	\mathbb{Q}	\mathbb{R}	\mathbb{S}	\mathbb{T}	\mathbb{U}	\mathbb{V}	\mathbb{W}	\mathbb{X}	\mathbb{Y}	\mathbb{Z}	\mathbb{A} \mathbb{B} ... \mathbb{Z}	<code>\mathbb{A}</code> <code>\mathbb{B}</code> ... <code>\mathbb{Z}</code>																										
\mathbb{A}	\mathbb{B}	\mathbb{C}	\mathbb{D}	\mathbb{E}	\mathbb{F}	\mathbb{G}	\mathbb{H}	\mathbb{I}	\mathbb{J}	\mathbb{K}	\mathbb{L}	\mathbb{M}																																											
\mathbb{N}	\mathbb{O}	\mathbb{P}	\mathbb{Q}	\mathbb{R}	\mathbb{S}	\mathbb{T}	\mathbb{U}	\mathbb{V}	\mathbb{W}	\mathbb{X}	\mathbb{Y}	\mathbb{Z}																																											
Латинский курсив	<table><tr><td>\mathscr{A}</td><td>\mathscr{B}</td><td>\mathscr{C}</td><td>\mathscr{D}</td><td>\mathscr{E}</td><td>\mathscr{F}</td><td>\mathscr{G}</td><td>\mathscr{H}</td><td>\mathscr{I}</td><td>\mathscr{J}</td><td>\mathscr{K}</td><td>\mathscr{L}</td><td>\mathscr{M}</td></tr><tr><td>\mathscr{N}</td><td>\mathscr{O}</td><td>\mathscr{P}</td><td>\mathscr{Q}</td><td>\mathscr{R}</td><td>\mathscr{S}</td><td>\mathscr{T}</td><td>\mathscr{U}</td><td>\mathscr{V}</td><td>\mathscr{W}</td><td>\mathscr{X}</td><td>\mathscr{Y}</td><td>\mathscr{Z}</td></tr></table>	\mathscr{A}	\mathscr{B}	\mathscr{C}	\mathscr{D}	\mathscr{E}	\mathscr{F}	\mathscr{G}	\mathscr{H}	\mathscr{I}	\mathscr{J}	\mathscr{K}	\mathscr{L}	\mathscr{M}	\mathscr{N}	\mathscr{O}	\mathscr{P}	\mathscr{Q}	\mathscr{R}	\mathscr{S}	\mathscr{T}	\mathscr{U}	\mathscr{V}	\mathscr{W}	\mathscr{X}	\mathscr{Y}	\mathscr{Z}	\mathscr{A} \mathscr{B} ... \mathscr{Z}	<code>\mathscr{A}</code> <code>\mathscr{B}</code> ... <code>\mathscr{Z}</code>																										
\mathscr{A}	\mathscr{B}	\mathscr{C}	\mathscr{D}	\mathscr{E}	\mathscr{F}	\mathscr{G}	\mathscr{H}	\mathscr{I}	\mathscr{J}	\mathscr{K}	\mathscr{L}	\mathscr{M}																																											
\mathscr{N}	\mathscr{O}	\mathscr{P}	\mathscr{Q}	\mathscr{R}	\mathscr{S}	\mathscr{T}	\mathscr{U}	\mathscr{V}	\mathscr{W}	\mathscr{X}	\mathscr{Y}	\mathscr{Z}																																											
Пробел (толстый)		<code>\;</code>																																																					

4.1.4. Дробь, степени, корни, индексы

При создании формул наиболее часто приходится пользоваться подписанием “Корни и дроби” списка “*LaTeX-формула*”. Щелчками левой кнопки мыши

по пиктограммам этого подсписка в поле “Правка” панели “Текст” формируются следующие *LaTeX*-команды ввода:

- обыкновенных дробей – $\frac{x}{y} \rightarrow \text{\texttt{\textbackslashfrac\{x\}\{y\}}}$;
- выражений со степенями – $x^y \rightarrow \text{\texttt{\textbackslashx^y}}$;
- выражений с индексами – $x_k, y_{i,j} \rightarrow \text{\texttt{\textbackslashx_k}}, \text{\texttt{\textbackslashy_{i,j}}}$;
- корней квадратных – $\sqrt{x} \rightarrow \text{\texttt{\textbackslashsqrt\{x\}}}$;
- корней n -ой степени – $\sqrt[n]{x} \rightarrow \text{\texttt{\textbackslashsqrt[n]\{x\}}}$;
- биномиальных коэффициентов $\binom{a}{b} \rightarrow \text{\texttt{\textbackslashbinom\{a\}\{b\}}}$.

В код вместо величин x, y, k, i, j и n могут быть вставлены другие выражения. Это и дает возможность формировать коды для вывода выражения любой сложности. Например, создать код, по которому выводится выражение

$$\frac{\sqrt{x} + \sqrt[3]{x}}{x_2 + x_3 + 1},$$

можно следующей последовательностью действий:

- 1) $\text{Click}\left(\frac{a}{b}\right) \rightarrow \text{\texttt{\textbackslashfrac\{a\}\{b\}}}$;
- 2) Удаляем a и оставляем курсор на месте $\rightarrow \text{\texttt{\textbackslashfrac\{\}\{b\}}}$;
- 3) $\text{Click}\left(\sqrt{x}\right) \rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}\}\{b\}}}$;
- 4) Набираем на клавиатуре знак “+” $\rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}+\}\{b\}}}$;
- 5) $\text{Click}\left(\sqrt[n]{x}\right) \rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[n]{x}\}\{b\}}}$;
- 6) Символ n исправляем на 3 $\rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[3]{x}\}\{b\}}}$;
- 7) Удаляем b и оставляем курсор на месте $\rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[3]{x}\}\{\}\{\}}}$;
- 8) $\text{Click}(x_a) \rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[3]{x}\}\{x_{\text{\texttt{\textbackslasha}}}\}}}$;
- 9) Вместо a записываем 2 и сдвигаем курсор на 1 позицию вправо $\text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[3]{x}\}\{x_{\text{\texttt{\textbackslash2}}}\}}}$;
- 10) Набираем на клавиатуре знак “+” $\rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[3]{x}\}\{x_{\text{\texttt{\textbackslash2}}}\{+\}}}}$;
- 11) $\text{Click}(x_a) \rightarrow \text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[3]{x}\}\{x_{\text{\texttt{\textbackslash2}}}\{+x_{\text{\texttt{\textbackslasha}}}\}}}}$;
- 12) Заменяем a на 3, сдвигаем курсор на 1 позицию вправо, добавляем 1 и в поле “Правка” получаем окончательный код:

$$\text{\texttt{\textbackslashfrac\{\sqrt{x}+\sqrt[3]{x}\}\{x_{\text{\texttt{\textbackslash2}}}\{+x_{\text{\texttt{\textbackslash3}}}\{+1\}}\}}}, \quad (1)$$

а в поле “Предпросмотр” требуемое выражение (1). Разумеется, код (1) можно было бы ввести в поле “Правка” напрямую с клавиатуры, вообще не используя подсписок “Корни и дроби”.

4.1.5. Пределы операторов

Запись пределов операторов интегрирования, суммирования, произведения, предела (*lim*) и т. п. возможна в двух вариантах – выше и ниже оператора или справа от оператора

$$\int_a^b \text{ и } \int^b_a ; \sum_a^b \text{ и } \sum^b_a ; \prod_a^b \text{ и } \prod^b_a ; \lim_{x \rightarrow \infty} \text{ и } \lim_{x \rightarrow \infty}$$

В зависимости от требуемого варианта записи полученный код следует подправить так. Если желательно иметь пределы выше и ниже оператора, то после команды вывода оператора следует добавить дополнительную команду “\limits”. Если желательно иметь пределы правее оператора, то после команды вывода оператора следует добавить дополнительную команду “\nolimits”. Например, приведенная выше строка с операторами вводилась по такому коду:

```
\int\nolimits_{a}^{b} и \int\limits_{a}^{b}\quad;\;  
\sum\nolimits_{a}^{b} и \sum\limits_{a}^{b}\quad;\;  
\prod\nolimits_{a}^{b} и \prod\limits_{a}^{b}\quad;\;  
\lim\nolimits_{x\to\infty} и \lim\limits_{x\to\infty};
```

И еще один момент, связанный с записью пределов операторов. Если под запись предела требуется отвести несколько последовательных строк, то сделать это можно, заключив эти строки в команду \substack (см. пример 1с).

Примеры 1. Запись пределов операторов.

- a) $f(x)=\int\nolimits_{-\infty}^x e^{-t^2}dt \rightarrow f(x) = \int_{-\infty}^x e^{-t^2}dt$
- b) $f(x)=\int\limits_{-\infty}^x \{e^{-t^2}\}dt \rightarrow f(x) = \int_{-\infty}^x e^{-t^2}dt$
- c) $\prod_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} Q_{i,j} \rightarrow \prod_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} Q_{i,j}$
- d) $f'(x)=\lim\nolimits_{\Delta x \rightarrow 0} \frac{f(x+\Delta x)-f(x)}{\Delta x} \rightarrow f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x)-f(x)}{\Delta x}$
- e) $f'(x)=\lim\limits_{\Delta x \rightarrow 0} \frac{f(x+\Delta x)-f(x)}{\Delta x} \rightarrow f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x)-f(x)}{\Delta x}$

4.1.6. Скобки переменных размеров

В табл. 3 приведены 4 типа парных скобках, соразмерных высоте содержания. Отметим что такие скобки могут быть иной формы, разного вида слева и справа, а также одиночными, то есть только левыми или только правыми. В общем случае для формирования левой и правой скобок, соразмерных высоте содержания, используется код

$$\backslash left \lambda \ldots \backslash right \mu , \tag{2}$$

где λ, μ – символы табл. 3, вводимые с клавиатуры или из списка “Символы” панели “Текст”, или команды из той же табл. 3. Заметим, что любые ограничители, формируемые по символам или командам, могут быть как левыми, так и правыми скобками, а отсутствие скобок задается точкой для λ или μ .

Таблица 3

Символы и команды для задания левых и правых скобок, соразмерных содержанию

Символ λ, μ	()	[]			↑	↓	↕	↗	↘	↕	.
Скобка	()	[]			↑	↓	↕	↗	↘	↕	нет скобки
Команда λ, μ	\{	\}	\	\angle		\rangle		\lfloor		\rfloor		\lceil	\rceil
Скобка	{	}		<		>		⌊		⌋		⌈	⌉
Команда λ, μ	\slash			\backslash									
Скобка	/			\									

Примеры 2. Различные скобки переменной длины (соразмерные по высоте содержанию).

a) \left| \begin{array}{l} a & b & c \\ d & e & f \\ g & h & i \end{array} \right|

b) \left| \begin{array}{l} a & b \\ c & d \end{array} \right|

→

a

b

c

d

e

f

g

h

i

- c) $\left\{ \begin{array}{l} x + y + z = 7 \\ x - y + 2z = 7 \\ 2x + 3y - z = 4 \end{array} \right.$
- d) $\int_a^b \frac{1}{2} (1+x)^{-3/2} dx = -\frac{1}{\sqrt{1+x}} \Big|_a^b$
- e) $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
- f)
$$g(x) = \begin{cases} f_1(x), & \text{если } x \geq 0; \\ f_2(x), & \text{если } -1 \leq x < 0; \\ f_3(x), & \text{если } x < -1. \end{cases}$$

4.1.7. Парные скобки на разных строках

Принято считать, что при записи многострочных формул парные скобки, или, по-другому, соответствующие друг другу открывающая и закрывающая скобки должны быть одинакового размера. Если такие скобки расположены в одной строке, а выводятся они командами `\left` и `\right`, то так и будет. Но в случае, когда скобки находятся в разных строках, их размер может не совпадать. Например, по коду:

$$x = \left(\left(a + b + c + d + e \right) \cdot \left(\frac{x}{y} + \frac{y}{z} + \frac{z}{x} \right) \right) + w.$$

выводится “некрасивая” формула

$$x = ((a + b + c + d + e) \cdot \left(\frac{x}{y} + \frac{y}{z} + \frac{z}{x} \right)) + w.$$

Здесь первая и последняя скобки соответствуют друг другу, но имеют неодинаковый размер. Можно, конечно, вместо скобок переменных размеров `\left` и `\right` использовать скобки предопределенных размеров, но правильно подобрать их возможно далеко не всегда. Более простой и универсальный прием таков. Вставить в требуемые места кода вертикальные пробельные фантомы, увеличивающие “высоту” формулы в разных местах до требуемой величины. Это приведет и к увеличению вертикальных размеров соответствующих скобок. В нашем случае достаточно в первой строке кода после первой круглой скобки командой `\vphantom{\frac{x}{y}}` вставить фантом `\frac{x}{y}`. На выводе в формуле по вертикали он займет такое же место, как и сама дробь `\frac{x}{y}`. А именно этой дробью и определяется высота соответствующей скобки второй строки формулы. Тем самым, размеры по высоте и открывающей, и закрывающей скобок будут определяться одинаковым выражением, а значит – окажутся равными. Таким образом, по коду

```
x = \left(\vphantom{\frac{x}{y}}\left(a+b+c+d+e\right)\cdot \right.\backslash
\left.\left(\frac{x}{y}+\frac{y}{z}+\frac{z}{x}\right)\right)+w.
```

получим

$$x = \left((a + b + c + d + e) \cdot \left(\frac{x}{y} + \frac{y}{z} + \frac{z}{x} \right) \right) + w.$$

4.1.8. Скобки предопределенных размеров

В любом месте формируемого выражения можно поставить ограничитель фиксированного размера. Сделать это можно командой “`\R \lambda`”, где $\mathcal{R} \in \{big, Big, bigg, Bigg\}$, λ – символ (кроме “.”) или команда табл. 3. Например, по коду

```
\Bigg(\bigg(\Big(\big( (скобки\ конкретных\ размеров) \big)\Big)\bigg)\Bigg)
```

выводится выражение

$$\left(\left(\left(\left(\left(\text{скобки конкретных размеров}\right)\right)\right)\right)\right).$$

К словам *big*, *Big*, *bigg*, *Bigg* можно приписывать справа буквы *l* (*left*) или *r* (*right*). На результат эти изменения влияния не окажут, но в некоторых случаях позволяют упростить понимание кода.

4.1.9. Дополнительные диакритические знаки

В подписке дополнительных диакритических знаков списка “*LaTeX-формула*” собраны знаки надчеркивания и подчеркивания, причем не для отдельных символов, а для слов и предложений. Щелчок левой кнопкой мыши по пиктограмме знака приводит к появлению в поле “*Правка*” панели “*Текст*” соответствующей *LaTeX*-команды. Например, щелчком по кнопке “ \overleftrightarrow{xx} ” в это поле вставляется команда “`\overbraceleftrightarrow{xx}`”. Если вместо *xx* подставить любой другой текст, то в поле “*Предпросмотр*”, а затем и в документе мы получим этот текст с двунаправленной стрелкой сверху. Так же формируются и другие знаки надчеркивания и подчеркивания. Заметим, что знаки “ — ”, “ \rightarrow ”, “ \leftarrow ”, “ \leftrightarrow ”, “ \Rightarrow ”, “ \Leftarrow ”, “ \Leftrightarrow ” размещаются снизу или сверху по всей длине текста, а знаки “ \wedge ” и “ \sim ” размещаются сверху не более чем над 5 средними символами текста.

И еще один вопрос, связанный с подчеркиваниями и надчеркиваниями. В некоторых случаях желательно над знаком надчеркивания или под знаком подчеркивания разместить какой-либо поясняющий текст. Например,

$$\overbrace{1+1+\dots+1}^{n \text{ единиц}}, \quad \underbrace{1+1+\dots+1}_{n \text{ единиц}}.$$

Делать это можно вставкой поясняющего текста как степени или как индекса предыдущего выражения. Скажем показанное выше выражение создано по такому коду:

$$\overbrace{1+1+\dots+1}^{\{n \text{ единиц}\}}, \quad \underbrace{1+1+\dots+1}_{\{n \text{ единиц}\}}.$$

Если поясняющий текст должен занимать несколько строк, то в этом случае его следует заключить в бокс командой – `\substack{поясняющий текст}`. Например,

$$\underbrace{1+1+\dots+1}_{\substack{n \text{ единиц}, \\ (n \leq 77)}} \rightarrow \underbrace{1+1+\dots+1}_{\substack{n \text{ единиц}, \\ (n \leq 77)}}$$

Разумеется, возможны и иные варианты размещения поясняющих текстов при наличии знаков подчеркивания и надчеркивания. Вот некоторые примеры с соответствующим кодом:

$$\overbrace{1+1+\dots+1}^{\limits_{\{n \text{ единиц}\}}} \rightarrow \overbrace{1+1+\dots+1}^{n \text{ единиц}}$$

$$\begin{aligned}
\text{\texttt{\textbackslash underbrace}\{1+1+...+1\}\text{\texttt{\textbackslash limits}^{\{n\text{ единиц}\}}} &\rightarrow \underbrace{1+1+...+1}_{n \text{ единиц}} \\
\text{\texttt{\textbackslash underbrace}\{1+1+...+1\}} &\rightarrow \underbrace{1+1+...+1}_{n \text{ единиц, } n>5} \\
\text{\texttt{\textbackslash underline}\{\text{\texttt{\textbackslash overbrace}\{1+1+...+1\}}\}} &\rightarrow \underline{\overbrace{1+1+...+1}}
\end{aligned}$$

4.1.10. Матрицы

Подсписок “*Матрицы*” списка “*LaTeX-формула*” используется для вставки в поле “*Правка*” команд вывода матриц (*массивов, таблиц*) размером 1×3 , 3×1 , 2×2 и 3×3 . Эти команды создаются щелчками левой кнопки мыши по кнопкам подсписка с соответствующими пиктограммами. Для матрицы 2×2 (кнопка с пиктограммой “ $\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}$ ”) код выглядит так:

$$\text{\texttt{\textbackslash begin}\{array\}\{\}\text{\texttt{\textbackslash&}}\text{\texttt{\textbackslash&}}\text{\texttt{\textbackslash end}\{array\}}, \quad (3)$$

а формируется по нему “*матрица*” – “ $\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}$ ” (без кавычек). Несколько замечаний по формированию кода для вывода матриц.

А. Чтобы (3) считать кодом для вывода матрицы, необходимо добавить к нему команды формирования ограничителей вида “(“ и “)” или “||” и “||”. Например, так:

$$\text{\texttt{\textbackslash left}\{\text{\texttt{\textbackslash begin}\{array\}\{\}\text{\texttt{\textbackslash&}}\text{\texttt{\textbackslash&}}\text{\texttt{\textbackslash end}\{array\}}\text{\texttt{\textbackslash right}\}}, \quad (4a)$$

$$\text{\texttt{\textbackslash left}\{\text{\texttt{\textbackslash&}}\text{\texttt{\textbackslash begin}\{array\}\{\}\text{\texttt{\textbackslash&}}\text{\texttt{\textbackslash end}\{array\}}\text{\texttt{\textbackslash right}\}\}, \quad (4b)$$

Тогда по (4a) и (4b) будут выводиться соответственно матрицы

$$\left(\begin{array}{cc} a & b \\ c & d \end{array} \right) \text{ и } \left\| \begin{array}{cc} a & b \\ c & d \end{array} \right\|.$$

В. Элементы матриц, то есть тело матрицы, заданы величинами a , b , c и d . В реальности это могут быть любые тексты и выражения. Так что в (4a) и (4b) их необходимо подправить, после чего результат в окне “*Предпросмотр*” может выглядеть, например, так:

$$\left(\begin{array}{cc} \text{сахар} & \text{мед} \\ \text{варенье} & \text{конфеты} \end{array} \right) \text{ и } \left\| \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right\|.$$

С. По первой матрице нетрудно понять, что элементы ее столбцов по умолчанию выравнены по центру относительно друг друга. Но можно выровнять их и по левому краю, и по правому краю, причем индивидуально для каждого из столбцов. А делается это так. В (4a) между пустыми фигурными скобками {} формируется преамбула – слово из m букв l (left), c (center) и r (right), где m – количество столбцов матрицы. Этими буквами указывается тип выравнивания в столбцах, а именно, буквы l , c и r в позиции k ($1 \leq k \leq m$) задают выравнивание элементов столбца k соответственно по левому краю, по центру и по правому краю. Если в преамбуле k букв l , c , r и $k < m$, то последние $m - k$ столбцов выравниваются по центру. Если $k > m$, то последние $k - m$ букв для выравнивания не учитываются. Например, по коду

```

\left(
\begin{array}{lr}
отвертки & дрели & плоскогубцы \\
молотки & пилы & топоры \\
гвозди & шурупы & саморезы
\end{array}
\right)

```

выводится матрица

$$\left(\begin{array}{ccc} \text{отвертки} & \text{дрели} & \text{плоскогубцы} \\ \text{молотки} & \text{пилы} & \text{топоры} \\ \text{гвозди} & \text{шурупы} & \text{саморезы} \end{array} \right).$$

Д. Продолжим начатый в пункте С разговор о преамбуле матрицы. В нее, между букв l , c и r , а также до них и после можно вставлять любое количество символов “|”. Каждый из них на соответствующей позиции в матрице выводит вертикальную черту. Что касается горизонтальных черт, то их можно вывести командами \hline, размещаемыми сразу после преамбулы и (или) команд “\”. Например, по коду

```

\begin{array}{||lr|c||} \hline \hline
отвертки & дрели & плоскогубцы \\ \hline
молотки & пилы & топоры \\ \hline
гвозди & шурупы & саморезы \\ \hline \hline
\end{array}.

```

выводится матрица

отвертки	дрели	плоскогубцы
молотки	пилы	топоры
гвозди	шурупы	саморезы

Обратите внимание, что в данном случае скобки “\left| ... \right|” нам не понадобились – они выведены преамбулой “{||lr|c||}”.

Е. Количество строк и (или) столбцов в матрице может быть любым. Например, если речь идет о матрице размером 2×5, то формировать ее можно, например, по такому шаблону:

```
\begin{array}{||cccc|}
a11 & a12 & a13 & a14 & a15\\
a21 & a22 & a23 & a24 & a25\\
\end{array}
```

где a_{ij} ($i=1, 2; j=1, 2, \dots, 5$) – элементы матрицы.

Аналогично матрицам формируются и определители матриц, только в данном случае для крайних ограничителей следует использовать одиночные прямые линии – “|”, а количество строк в соответствующей матрице должно быть равно количеству ее столбцов.

Г. Объединение n последовательных ячеек в строке таблицы реализуется командой `\multicolumn{n}{выравнивание}{содержимое}`, вставляемой в код на место их определения. Вот конкретный пример такого кода


```
\begin{array}{|c|l|c|r|c|}
a & \multicolumn{3}{c}{Объединение} & a \\
a & a & a & a & a \\
a & a & a & a & a
\end{array}
```

и вывод по нему:

a	Объединение			a
a	a	a	a	a
a	a	a	a	a

В заключение этого пункта отметим, что нельзя задавать конкретную ширину ячеек столбца, создавать многострочные ячейки или объединять несколько последовательных ячеек столбца в одну ячейку.

4.2. Команды для работы с текстами

Для работы с текстами имеется ряд специальных команд (функций), выполняемых при их наборе в строке ввода. Поэтому они мало связаны с ранее рассмотренными средствами ввода текстов, предоставляемыми инструментом “ Текст” и вводимыми через панель “Текст”. Кратко опишем эти команды.

B) Если значения выражений $nu1$ и $nu2$ – действительные числа, то каждое из них преобразуется по A в отдельности и затем формируется текст в виде точки $\left(\frac{a}{b}, \frac{c}{d}\right)$.

C) Значение x аргумента nu должно быть действительным числом. Оно и возвращается в виде текста в научном формате $m.n \times 10^k$ ($1 \leq m \leq 9$, m, k – целые).

D) Значение x аргумента nu должно быть действительным числом, значение pr – целым числом. По D возвращается число x в научном формате с округлением до pr значащих цифр.

Примеры 4.

1) $FractionText[7.4] \rightarrow \frac{37}{5}$ (7.4 – число, $\frac{37}{5}$ – текст (формула).

$FractionText[Slope[-0.5*x+1]] \rightarrow \frac{-1}{2}$, $FractionText[1.4+7.3i] \rightarrow \left(\frac{7}{5}, \frac{73}{10}\right)$.

2) $ScientificText[567] \rightarrow 5.67 \times 10^2$,
 $ScientificText[0.000000567] \rightarrow 5.670000000000001 \times 10^{-7}$.

3) $ScientificText[3.1456, 8] \rightarrow 3.1456000 \times 10^0$,
 $ScientificText[3.1456, 3] \rightarrow 3.15 \times 10^0$.

A. $SurdText[nu]$ B. $SurdText[(nu1, nu2)]$	C. $SurdText[nu, \{t1, t2, \dots, tk\}]$
--	--

A) По A делается попытка представить числовое действительное значение nu в виде текста вида $\frac{a+b\sqrt{c}}{d}$, где a, b, c, d – целые числа, $c \geq 0$. Если такое представление найти не удалось, то возвращается значение nu .

B) По B обе компоненты точки ($nu1, nu2$) преобразуются по тем же правилам, что и аргумент nu в A .

C) По C делается попытка представить числовое значение nu в виде текста в форме $n0+n1 \cdot t1+n2 \cdot t2+\dots+nk \cdot tk$, где $n0, n1, \dots, nk$ – целые числа, а элементы $t1, t2, \dots, tk$ могут быть такими: $\exp(1), \exp(2), \dots, \sqrt{2}, \sqrt{3}, \dots, \ln(2), \dots$.

Примеры 5.

1) $SurdText[(2+3*\sqrt{5})/7] \rightarrow \frac{2+3\sqrt{5}}{7}$.

2) $SurdText[1.70710678119] \rightarrow \frac{2+\sqrt{2}}{2}$.

$SurdText[7.777] \rightarrow 7.777000000000001$.

3) $SurdText[((3+7*\sqrt{2})/5, 3.31662479036] \rightarrow \left(\frac{3+7\sqrt{2}}{5}, \sqrt{11}\right)$.

- 4) $\text{SurdText}[7.70674230226, \{\text{sqrt}(2), \text{sqrt}(3)\}] \rightarrow 3\sqrt{2} + 2\sqrt{3}$,
 $\text{SurdText}[16.5111739319, \{\ln(3), \ln(5), \ln(11)\}] \rightarrow$
 $\ln(3)+2\cdot\ln(5)+3\cdot\ln(11)+5$.

A. $\text{ContinuedFraction}[nu]$
B. $\text{ContinuedFraction}[nu, lev]$
C. $\text{ContinuedFraction}[nu, lev, boolsh]$

Вывод непрерывной дроби

А) По *A* число *nu* разлагается в непрерывную дробь, представляющую *nu* с точностью до 10^{-8} . Неполные частные разложения – это целые числа, находящиеся слева от знака “+”.

В) По *B* число *nu* разлагается в непрерывную дробь с количеством неполных частных, не превосходящих значения целой части *lev* (*lev*>0). В любом случае это количество меньше или равно количеству неполных частных, возвращаемых по *A*.

С) Первые два аргумента *C* имеют тот же самый смысл, что и в *B*. Аргумент *lev* не обязателен. Третий булевский аргумент *boolsh* задает форму вывода непрерывной дроби. При *boolsh*=*true* дробь выводится в так называемой короткой форме, то есть в виде последовательности неполных частных, заключенных в квадратные скобки [*q*₀; *q*₁, ..., *q*_{*n*}]. При *boolsh*=*false* дробь выводится в обычном развернутом виде.

Примеры 6.

- 1) $\text{ContinuedFraction}[3.77] \rightarrow$

$$3 + \frac{1}{1 + \frac{1}{3 + \frac{1}{2 + \frac{1}{1 + \frac{1}{6 + \frac{1}{1}}}}}}$$
- 2) $\text{ContinuedFraction}[3.7, 3] \rightarrow$

$$3 + \frac{1}{1 + \frac{1}{3 + \dots}}$$
- 3) $\text{ContinuedFraction}[3.77, \text{true}] \rightarrow$

$$[3; 1, 3, 2, 1, 6, 1].$$

A. $\text{Text}[expr]$	C. $\text{Text}[expr, (a, b)]$
B. $\text{Text}[expr, boolsub]$	D. $\text{Text}[expr, (a, b), boolsub]$
	E. $\text{Text}[expr, (a, b), boolsub, boollatex]$

Вывод текста

А) По *A* выражение *expr* выводится как обычный текст. Предварительно в *expr* подставляются текущие значения всех переменных. Текст выводится где-то рядом с началом координат и затем мышью его можно перетащить на требуемое место.

В) Команда *B* при булевой переменной *boolsub*=*true* выполняется как *A*. При *boolsub*=*false* выражение *expr* выводится по *B* как обычный текст, причем без подстановок текущих значений переменных.

С) Команда *C* выполняется как *A*, но текст прикрепляется к позиции (a, b) , то есть нижний левый угол бокса текста помещается в точку (a, b) .

D) Команда *D* выполняется как *B*, но текст прикрепляется к позиции (a, b) .

E) Команда *E* при булевой переменной *boollatex=false* выполняется как *D*. При *boollatex=true* выражение выводится как *LaTeX*-текст.

Примеры 7.

- 1) $\text{Text}[x^2 + \sin(3 \cdot x)] \rightarrow x^2 + \sin(3x)$;
- 2) Вводим $x=2$, вводим $\text{Text}[x^2 + \sin(3 \cdot x)] \rightarrow 4 + \sin(6)$.
- 3) Вводим $x=2$, вводим $\text{Text}[x^2 + \sin(3 \cdot x), \text{false}] \rightarrow x^2 + \sin(3x)$.
- 4) Вводим $x=2$, вводим $\text{Text}[x^2 + \sin(3 \cdot x), \text{false}, \text{true}] \rightarrow x^2 + \sin(3x)$.

A. $\text{FormulaText}[\text{expr}]$ B. $\text{FormulaText}[\text{expr}, \text{boolsub}]$	C. $\text{FormulaText}[\text{expr}, \text{boolsub}, \text{boolshow}]$
--	---

Вывод формул

A) По *A* выражение *expr* выводится как *LaTeX*-текст. Предварительно в *expr* подставляются текущие значения всех переменных кроме x, y, z .

B) Команда *B* при булевой переменной *boolsub=true* выполняется как *A*. При *boolsub=false* выражение *expr* выводится по *B* как *LaTeX*-текст, причем без подстановок текущих значений каких-либо переменных.

C) Команда *C* выполняется аналогично *B* с таким дополнением. Если вместо выражения *expr* используется его имя *g* ($g(\dots) = \text{expr}$), то возможны такие случаи. Если булева переменная *boolshow=true*, то выводится *LaTeX*-текст для $g(\dots) = \text{expr}$, иначе – для *expr*.

Примеры 8.

- 1) $\text{FormulaText}[z \cdot x^3 + 5 \cdot x/y^2] \rightarrow z \cdot x^3 + 5 \cdot \frac{x}{y^2}$;
- 2) Вводим $a=5$, вводим $b=1$, вводим $x=2$,
вводим $\text{FormulaText}[a \cdot x^3 + 5 \cdot x/b^2] \rightarrow 5 \cdot x^3 + 5 \cdot \frac{x}{1^2}$.
- 3) Вводим $a=2$, вводим $y=1$, вводим $x=3$,
вводим $\text{FormulaText}[a \cdot x^3 + 5 \cdot x/y^2, \text{true}] \rightarrow 2x^3 + 5 \cdot \frac{x}{y^2}$;
- 4) $\text{FormulaText}[a \cdot x^3 + 5 \cdot x/y^2, \text{false}] \rightarrow a \cdot x^3 + 5 \cdot \frac{x}{y^2}$.
- 5) Вводим $f(x, y) = x^2 + \sin(1 + y^2)$,
вводим $\text{FormulaText}[f, \text{false}, \text{true}] \rightarrow f(x, y) = x^2 + \sin(1 + y^2)$;
- 6) $\text{FormulaText}[f, \text{false}, \text{false}] \rightarrow x^2 + \sin(1 + y^2)$

A. <i>RotateText</i> [" <i>expr</i> ", <i>angle</i>]	B. <i>VerticalText</i> [" <i>expr</i> "] C. <i>VerticalText</i> [" <i>expr</i> ", (<i>a</i> , <i>b</i>)]
---	---

Вращения текста

A) По *A* выражение *expr* рассматривается как *LaTeX*-текст. По этому тексту формируется выражение и затем его бокс выводится повернутым вокруг левого верхнего угла на угол *angle* против часовой стрелки. По умолчанию угол измеряется в радианах. Но прямо на месте можно указывать и градусы (45° , 30° , ...).

B) Команда *B* выполняется аналогично *B* при *angle*= 90° .

C) Команда *C* выполняется аналогично *B*, но результат появляется не в начале координат, а в позиции (*a*, *b*).

Пример 9.

$$\text{RotateText}["\frac{a \cdot x + b}{a \cdot x - b} + 2", 20^\circ] \rightarrow \frac{a \cdot x + b}{a \cdot x - b} + 2$$

A. <i>TableText</i> [<i>lis1</i> , <i>lis2</i> , ..., <i>lisk</i>]
B. <i>TableText</i> [<i>lis1</i> , <i>lis2</i> , ..., <i>lisk</i> , <i>align</i>]

Вывод таблиц

Здесь *lis1*, *lis2*, ..., *lisk* – списки, элементы которых могут быть выражениями типа *expr* из команды *FormulaText*; *align* – необязательный строковый аргумент для задания типа выравнивания элементов в выводимой таблице и для указания ориентации этой таблицы. Пусть *m* – максимальное из количеств элементов в списках *lis1*, *lis2*, ..., *lisk*.

A) По *A* выводится таблица размером $k \times m$ (*k* строк, *m* столбцов). В позицию (*i*, *j*) ($i=1, 2, \dots, k; j=1, 2, \dots, m$) этой таблицы выводится элемент, находящийся в позиции *j* *i*-го списка. Если в списке такого элемента нет, то позиция (*i*, *j*) таблицы остается пустой. Элементы помещаются в ячейки таблицы как *LaTeX*-тексты, получаемые командой *FormulaText*. При этом выравнивание элементов реализуется по левому краю.

B) По *B* таблица выводится, как по *A*, но со следующими уточнениями, организуемыми аргументом *align*. Возможные значения *align* – “*vl*”, “*vc*”, “*vr*”, “*v*”, “*h*”, “*hl*”, “*hc*”, “*hr*”. По умолчанию *align*=“*hl*”. Смысл отдельных букв, составляющих значение *align* таков:

- *h* – ориентация таблицы горизонтальная, то есть элементы каждого списка размещаются в отдельной строке, как в случае *A*;
- *v* – ориентация таблицы вертикальная, то есть элементы каждого списка размещаются в отдельном столбце (таблица строится как в *A* и затем транспонируется)
- *l* – элементы выравниваются в таблице по левому краю ячейки;
- *c* – элементы выравниваются в таблице по центру ячейки;
- *r* – элементы выравниваются в таблице по правому краю ячейки.

Примеры 10.

1) *TableText*[{1, x, x², x³}, {sin(x), cos(x)}, {x, y, 2, 3}] →

1	x	x ²	x ³
sin (x)	cos (x)		
x	y	2	3

.

2) *TableText*[{1, x, x², x³}, {sin(x), cos(x)}, {x, y, 2,

1	sin (x)	x
x	cos (x)	y
x ²		2
x ³		③

.

5. Центры треугольника и их свойства

Треугольник – простейшая плоская фигура, но его изучение, начатое в древнейшие времена, продолжается и по сей день. Без всякого преувеличения можно сказать, что треугольник является важнейшей и неисчерпаемой фигурой геометрии. В этом разделе речь пойдет о некоторых центрах треугольников, в том или ином смысле являющихся для него особыми (специфическими) точками. После древних греков, которые уже оперировали такими понятиями, как инцентр, центроид, центр описанной окружности и ортоцентр, было изучено много интересных особых точек треугольника. В 1994 году Кларк Кимберлинг (*Clark Kimberling*) провел классификацию имеющихся центров и постоянно продолжает пополнять свою энциклопедию (базу) новыми интересными центрами [56,57,77]. Среди прочих данных в энциклопедию включаются трилинейные и барицентрические координаты центров треугольников.

5.1. Некоторые центры треугольников

A. TriangleCenter[A, B, C, n]

Построение треугольных центров

А) Здесь A, B, C – точки, n ($1 \leq n \leq 3053$) – натуральное число, такое, что $X(n)$ – имя центра треугольника в энциклопедии центров Кларка Кимберлинга [56,57,73]. По команде *TriangleCenter* выводится точка, которая для $\triangle ABC$ является треугольным центром под именем $X(n)$.

В табл. 1 приведены названия первых 18 центров треугольника и даны пояснения к их непосредственному построению без использования команды *TriangleCenter*. В колонке “Имя” указывается имя центра по классификации Кимберлинга, а также его имя, наиболее часто применяемое в геометрии в англоязычных источниках. При построении динамических моделей использовать для центров именно эти имена совсем не обязательно. Если на одном изображении одновременно должны присутствовать несколько центров, то приемлемыми для них могут быть как обозначения Кимберлинга $X(n)$, так и их стилизованные версии в виде Xn (без круглых скобок) или X_n (с индексом).

В последующих разделах решается большое количество задач по построению динамических моделей, связанных с теми или иными центрами треугольников. Следует учесть, что при создании моделей:

- прямые линии (но не отрезки (сегменты)!) практически всегда будут выводиться тонкими и в точечном стиле: *толщина* – 1.5, *стиль* – “-----”. Чтобы в текущей сессии при работе с документами избежать непрерывной корректировки стиля линий, эти установки надо сделать действующими по умолчанию, то есть зафиксировать на странице “Установки по умолчанию” па-

нели “*Настройки*”. При выполнении дополнительной команды меню “*Настройки\Сохранить настройки*” установленные значения по умолчанию будут действовать и в новых сессиях. Возвратить действие всех системных установок по умолчанию можно командой меню “*Настройки\Сбросить настройки*”;

- обозначения объектов, кроме имен точек, как правило, выводиться не будут. Для этого достаточно через меню выполнить команду “*Настройки\Обозначения\Только для точек*”. Впрочем, если для конкретного объекта обозначение все же вывести требуется, то это можно сделать через контекстное меню (*RClick(объект)\Показывать обозначение*);

- стиль вывода того или иного объекта иногда будет корректироваться без соответствующего на то указания.

Таблица 1

Некоторые центры треугольников и их определение

ИМЯ	НАЗВАНИЕ	ДОПОЛНИТЕЛЬНОЕ НАЗВАНИЕ. ПОЯСНЕНИЯ
X(1) I	Инцентр (<i>Incenter</i>)	<i>Центр вписанной окружности</i> . Точка пересечения биссектрис треугольника. Точка, равноудаленная от сторон треугольника. С инцентром тесно связаны еще три точки – центры вневписанных окружностей (<i>excenters</i>). Они также равноудалены от сторон треугольника или их продолжений.
X(2) G	Центроид (<i>Centroid</i>)	<i>Точка пересечения медиан треугольника</i> , центр тяжести, центр масс.
X(3) O	Центр описанной окружности (<i>Circumcenter</i>)	<i>Точка пересечения медиатрис</i> – перпендикуляров к серединам сторон треугольника (“се-рединных” перпендикуляров).
X(4) H	Ортоцентр (<i>Orthocenter</i>)	Центр пересечения высот треугольника или их продолжений.
X(5) N	Центр окружности 9 точек	<i>Центр окружности Эйлера</i> . Основания трёх высот треугольника, середины трёх его сто-рон и середины трёх отрезков, соединяющих вершины треугольника с его ортоцентром, ле-жат на окружности 9 точек.
X(6) K	Точка Лемуана	Точка пересечения симедиан треугольника, то есть отрезков прямых, симметричных его ме-дианам относительно соответствующих бис-сектрис.
X(7) Ge	Точка Жергонна	Точка пересечения отрезков, соединяющих вершины треугольника с точками касания противоположных сторон вписанной окруж-ностью.

X(8) Na	Точка Нагеля	Точка пересечения отрезков, соединяющих вершины треугольника с точками касания противоположных сторон соответствующими вневписанными окружностями.
X(9) M	Средняя точка (<i>Mittenpunkt</i>)	Средняя точка или центр эллипса Мандара – это точка пересечения симедиан треугольника, у которого вершинами являются центры вневписанных окружностей исходного треугольника.
X(10) Sp	Центр Шпикера	<i>Инцентр срединного треугольника</i> , то есть центр окружности, вписанной в треугольник с вершинами в серединах сторон исходного треугольника. Центр Шпикера является центром масс периметра исходного треугольника.
X(11) F	Точка Фейербаха	Точка касания окружности 9 точек и вписанной окружности. Иногда точками Фейербаха называют также и точки касания окружности 9 точек с 3 вневписанными окружностями.
X(12)	Точка, гармонически сопряженная с X(11) по отношению к точкам X(1) и X(5). (<i>Harmonic conjugate of X(11)</i>)	Две точки <i>C</i> и <i>D</i> на отрезке <i>AB</i> называются гармонически сопряженными, если они делят <i>AB</i> в одном и том же отношении, причем одна внутренним образом, а другая – внешним образом. Любая точка отрезка, кроме его середины, имеет гармонически сопряженную точку. Впрочем, для середины отрезка гармонически сопряженной можно считать бесконечно удаленную точку
X(13) X X(14) X'	Первая точка Ферма Вторая точка Ферма	1-я точка Ферма (<i>точка Ферма-Торричелли</i>). Пусть на сторонах треугольника вовне построены три правильных треугольника. Тогда 1-ая точка Ферма – это точка пересечения шести кривых – трех окружностей, описанных около этих треугольников, и трех прямых, соединяющих вершины исходного треугольника с противоположными вершинами построенных треугольников. 2-я точка Ферма. Вторая точка Ферма определяется как 1-я точка Ферма, но правильные треугольники строятся не во вне, а внутрь исходного треугольника.
X(15) S X(16)	Первая точка Аполлония	Точки Аполлония это – точки, от которых расстояния до вершин треугольника обратно пропорциональны длинам противоположных

S'	Вторая точка Аполлония	сторон. Любой треугольник, отличный от равностороннего, имеет две точки Аполлония.
X(17) N	Первая точка Наполеона	1-я точка Наполеона – это точка пересечения трех линий, соединяющих центры внешних правильных треугольников, построенных на сторонах исходного треугольника, с противоположными вершинами этого треугольника. 2-я точка Наполеона определяется аналогично 1-ой точке Наполеона, но правильные треугольники на сторонах исходного треугольника строятся не во вне, а внутри его.
X(18) N'	Вторая точка Наполеона	

5.2. Инцентр

В энциклопедии Кимберлинга инцентр, то есть центр вписанной окружности, именуется как $X(1)$ и через строку ввода его можно строить командой $TriangleCenter[A,B,C,1]$. Мы это будем делать с помощью встроенных инструментов, учитывая, что инцентр – это точка пересечения биссектрис треугольника. В *GeoGebra* имеется команда, по которой выводится не только инцентр, но сама вписанная в $\triangle ABC$ окружность – это $Incircle[A,B,C]$. По умолчанию инцентру по $Incircle$ присваивается имя *inc* возможно с индексами. Через контекстное меню имя *inc* можно изменить.

Определение 1. Вневписанной окружностью называется окружность, касающаяся одной из сторон треугольника и продолжений двух других его сторон. Центром любой из трех вневписанных окружностей является точка пересечения биссектрисы внутреннего угла и двух биссектрис внешних углов треугольника, не смежных с этим внутренним углом.

Центры вневписанных окружностей (*excenters*), как и инцентр, равноудалены от сторон треугольника или их продолжений.

Пусть задан $\triangle ABC$ со сторонами a, b и c , противолежащими вершинам A, B и $C, p=(a+b+c)/2$ – полупериметр, точка O – инцентр, точки E, F и G – центры вневписанных окружностей, d_a, d_b и d_c – расстояния от инцентра до вершин треугольника и D_a, D_b и D_c – расстояния между центрами вневписанных окружностей (см. рис. 1а).

Справедливы следующие утверждения:

1. *Деление биссектрис.* Инцентр делит биссектрисы внутренних углов A, B и C в отношении соответственно $(b+c)/a, (a+c)/b$ и $(a+b)/c$.

2. *Длина биссектрисы.* Для биссектрисы BD $\triangle ABC$ справедлива формула $BD^2=BA\cdot BC-AD\cdot DC$.

3. Расстояние между инцентром и центрами вневписанных окружностей равны:

$$d_a = a \cdot \sqrt{\frac{bc}{p(p-a)}}, \quad d_b = b \cdot \sqrt{\frac{ac}{p(p-b)}}, \quad d_c = c \cdot \sqrt{\frac{ab}{p(p-c)}}. \quad (1)$$

4. Расстояние между центрами вневписанных окружностей равны:

$$D_a = a \cdot \sqrt{\frac{bc}{(p-b)(p-c)}}, \quad D_b = b \cdot \sqrt{\frac{ac}{(p-a)(p-c)}}, \quad D_c = c \cdot \sqrt{\frac{ab}{(p-a)(p-b)}}. \quad (2)$$

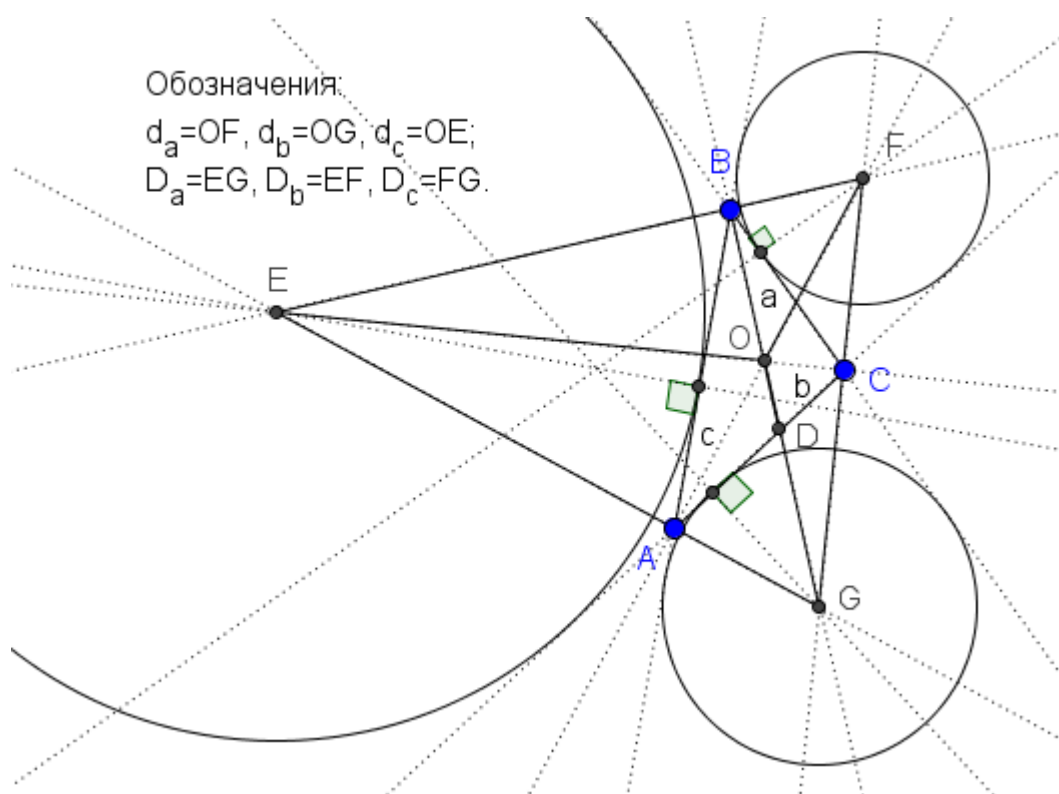


Рис. 1а. К задачам 1-4

Построение инцентра. Изображение, представленное на рис. 1б, является результатом построения инцентра как точки пересечения биссектрис внутренних углов треугольника. Создавалось оно следующим образом:

- через меню блокируем вывод обозначений объектов, кроме вывода имен точек (*Настройки/Обозначения/Только для точек*). Этот пункт будем выполнять при построении практически всех треугольных центров;
- инструментом “Точка” сформируем три точки *A*, *B* и *C*;
- инструментом “Прямая” проведем прямые линии через пары точек *A* и *B*, *B* и *C*, *C* и *A*;




- инструментом “ Отрезок” соединим пары точек A и B , B и C , C и A , то есть создадим $\triangle ABC$;
- инструментом “ Биссектриса угла” проведем биссектрисы каждого угла $\triangle ABC$;
- инструментом “ Точка на объекте” сформируем инцентр и через контекстное меню поменяем его имя на I .

Рис. 1с получен редактированием рис. 1b, к которому добавлены точки D , E , F и на панели “Настройки” изменен стиль вывода биссектрис, а также размер и цвет инцентра I .

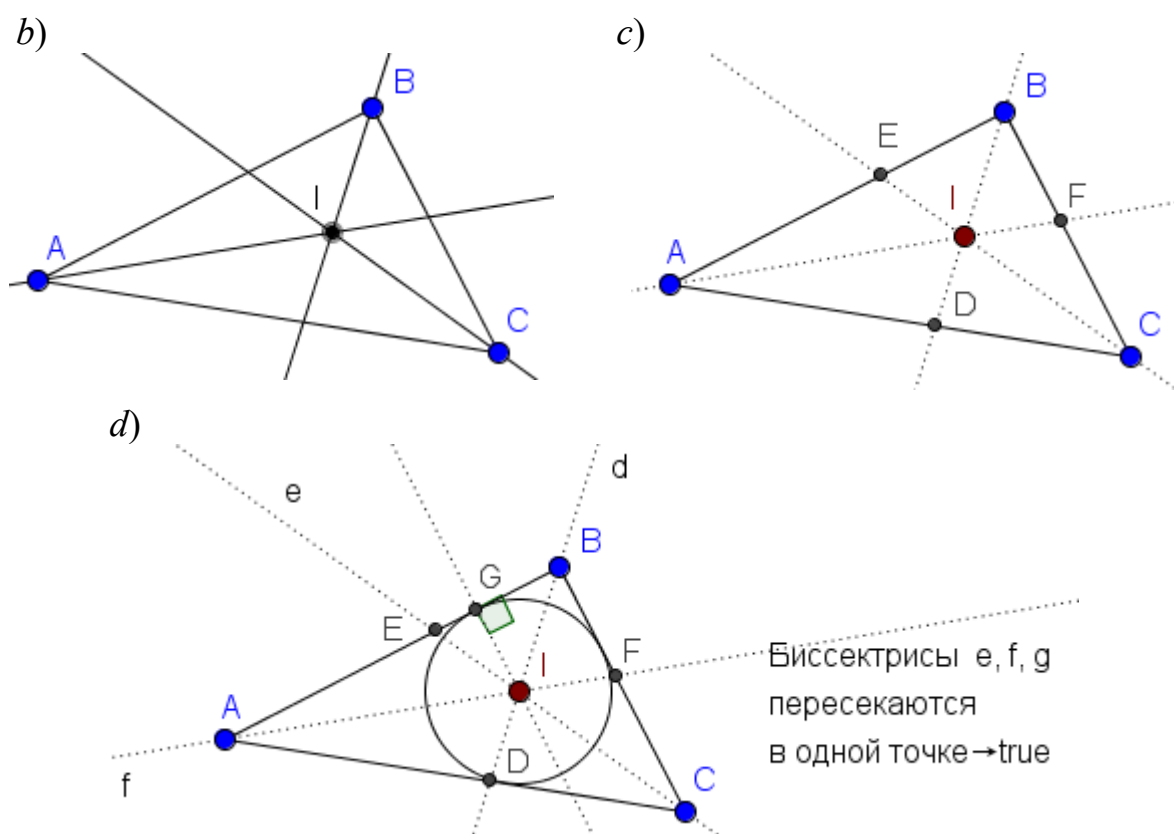







Рис. 1 (b, c, d). Построение инцентра встроенными инструментами

Добавить к рис. 1с вписанную окружность можно, например, так:


- инструментом “ Перпендикулярная прямая” из инцентра проведем перпендикуляр к одной из сторон, например, к AB ;
- инструментом “ Точка на объекте” зафиксируем точку G пересечения проведенного перпендикуляра и соответствующей стороны треугольника;
- инструментом “ Окружность по центру и точке” проведем требуемую вписанную окружность;
- рис. 1d получен редактированием созданного изображения, в котором инструментом “ Угол” проставлена метка прямого угла и затем инструмен-

том “ Текст” сформирована надпись из трех строк. Для того, чтобы вычисляемая часть надписи выводила логическое значение *true* (или *false*), ее следует оформлять в боксе в виде:

Биссектрисы d, e, f
пересекаются
в одной точке $AreConcurrent[d, e, f]$.

Поскольку точки A, B, C – независимы, то перемещая их по плоскости, будем получать динамическую картинку, то есть другие треугольники с актуализацией для них всех проделанных построений.

Пример 1. Создать динамическую модель для экспериментальной проверки утверждения 1: “Инцентр $\triangle ABC$ делит биссектрисы углов A, B и C соответственно в отношении $(b+c)/a, (a+c)/b$ и $(a+b)/c$ ”.

Решение. Реализуем требуемую проверку для одного из углов, например, для угла A . Нам надо проверить значение истинности логического выражения $AI/IF == (AB+AC)/BC$ (см. рис. 1e). Если это выражение ввести в строку ввода, то на панели “Объекты” получим логическое значение *true*. При перемещении независимых точек A, B или C значение этого выражения изменяться не будет. Удобнее организовать указанную проверку прямо на панели “Полотно”. Для этого инструментом “ Текст” где-нибудь рядом с треугольником следует сформировать надпись с переменной (вычисляемой) частью. Для этого на раскрывшейся панели “Текст” в поле “Правка” при включенной кнопке “LaTeX-формула” следует записать выражение

$$\frac{AI}{IF} = \frac{AB+AC}{BC} \rightarrow \boxed{\frac{AI}{IF} == (AB+AC)/BC} .$$

Для того, чтобы выражение справа от символа “ \rightarrow ” перевычислялось вместе с изменениями параметров модели, его необходимо сформировать в пустой рамке (в рамке объектов).

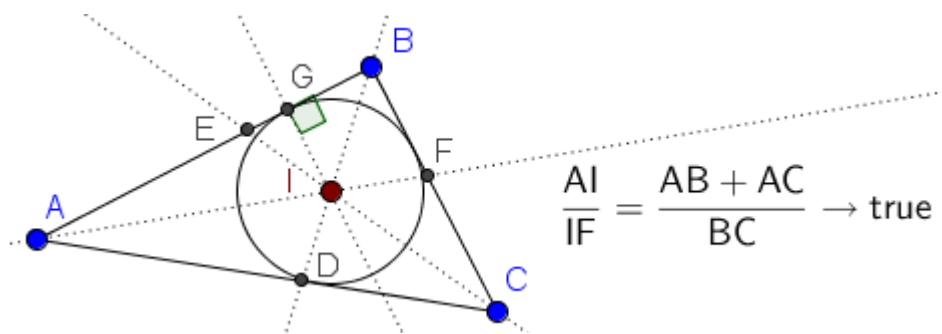




Рис. 1e. Модель для экспериментальной проверки утверждения примера 1

Пример 2. Создать динамическую модель для экспериментальной проверки утверждения 2: “Для биссектрисы $BD \triangle ABC$ справедлива формула $BD^2 = BA \cdot BC - AD \cdot DC$ ”.

Решение. Продолжим построение модели, начиная с изображения, представленного на рис. 1d:

- удалим надпись и инструментом “ Отрезок” сформируем отрезок BD (необязательно);
- инструментом “ Текст” при включенной кнопке “*LaTeX-формула*” создадим проверочную надпись, подтверждающую или опровергающую утверждение примера 2:

$$BD^2 = BA \cdot BC - AD \cdot DC \rightarrow \boxed{BD^2 == BA \cdot BC - AD \cdot DC}.$$

Перемещая независимые точки A , B и C по плоскости, мы всегда будем получать в правой части надписи значение *true*, что и требовалось проверить.

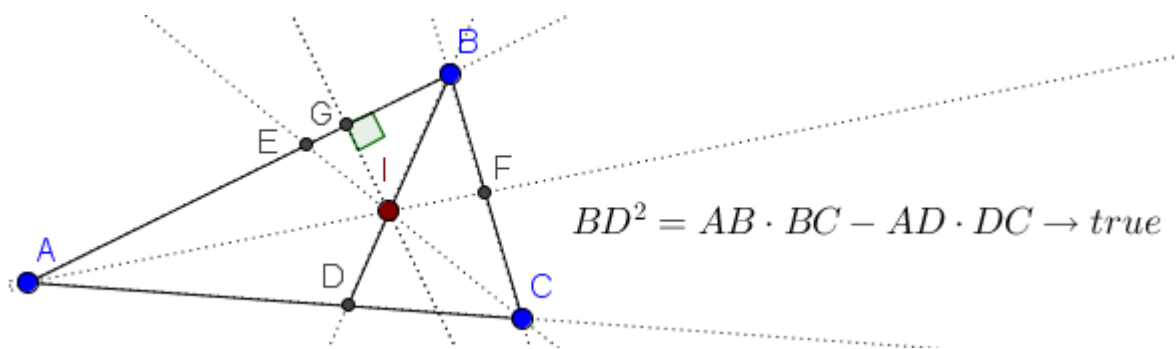













Рис. 1f. Модель для экспериментальной проверки утверждения примера 2

Пример 3. Создать динамическую модель для экспериментальной проверки утверждений 3 и 4, то есть проверки справедливости формул (1) и (2).

Решение. На рис 1a показано статичное изображение для пояснений к утверждениям 1-4. Сейчас нам предстоит создать аналогичную динамическую модель (см. рис. 1g) и сделать это можно так:

- блокируем вывод обозначений объектов, кроме вывода имен точек;
- инструментом “ Точка” сформируем три точки A , B и C ;
- инструментом “ Отрезок” проведем отрезки через пары точек B и C , C и A , A и B , то есть создадим $\triangle ABC$. Через контекстное меню обозначения a , b и c сторон $\triangle ABC$ сделаем видимыми;
- инструментом “ Прямая” проведем прямые линии через пары точек A и B , B и C , C и A ;
- инструментом “ Биссектриса угла” проведем биссектрисы внутренних и внешних углов $\triangle ABC$;

- инструментом “ Точка на объекте” зафиксируем точку D пересечения внутренних биссектрис $\triangle ABC$. Через контекстное меню переименуем D на O . Тем самым построен центр вписанной окружности;
- инструментом “ Точка на объекте” зафиксируем точки D, E и F пересечения биссектрис внешних углов. Тем самым построены центры внеписанных окружностей;
- инструментом “ Перпендикулярная прямая” из точек D, E и F проведем перпендикулярные линии к соответствующим сторонам $\triangle ABC$. Инструментом “ Точка на объекте” создадим точки G, H и I пересечения этих линий со сторонами треугольника. Инструментом “ Угол” в полученных точках G, H и I проставим метки прямых углов. Через контекстное меню спрячем имена точек G, H и I и обозначения меток прямых углов. Инструментом “ Окружность по центру и точке” проведем внеписанные окружности (необязательно);
- инструментом “ Отрезок” соединим центр O вписанной окружности с центрами D, E и F внеписанных окружностей, а также центры D, E и F друг с другом. Изменим цвет проведенных отрезков (необязательно);
- через контекстное меню изменим имя прямой p на p_1 . Тем самым имя p стало неиспользуемым и через строку ввода можно сформировать вспомогательную переменную $p=(a+b+c)/2$. Сделаем это:

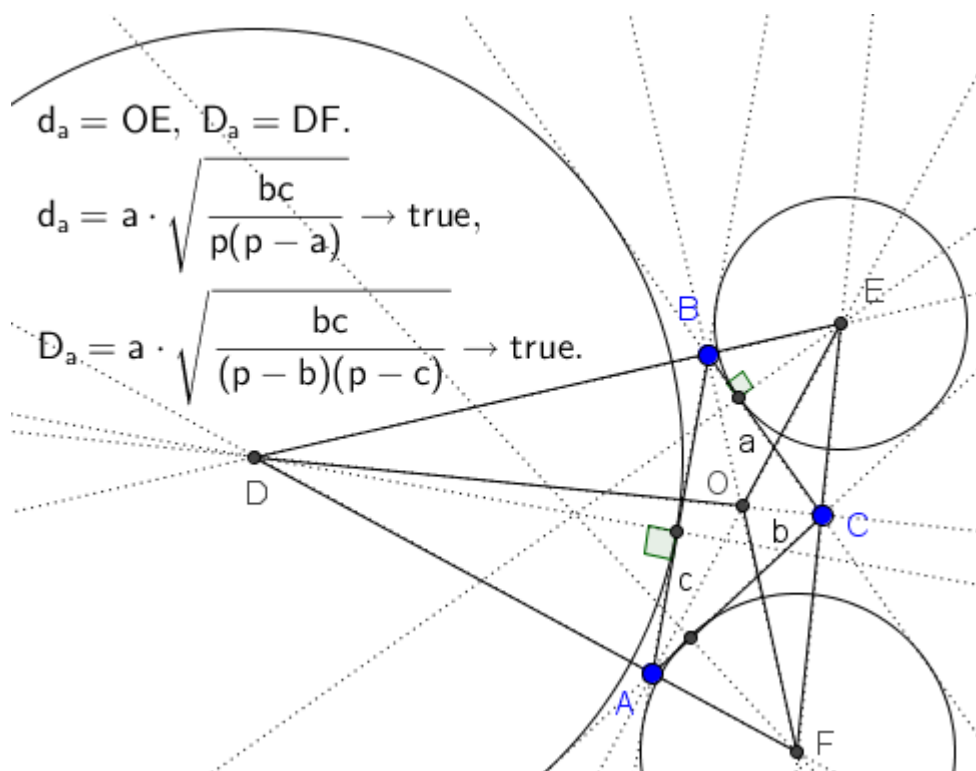



Рис. 1g. Модель для экспериментальной проверки утверждений примера 3

• инструментом “ Текст” при включенной кнопке “LaTeX-формула” создадим надпись заново так, чтобы она подтверждала или опровергала утверждения 3 и 4. При этом ограничимся проверкой формул лишь для d_a и D_a :

$$d_{\{a\}}=OE, \backslash, \backslash, D_{\{a\}}=DF \quad .\backslash$$

$$d_{\{a\}}=a\cdot\sqrt{\frac{bc}{p(p-a)}}\rightarrow \boxed{OE==a*\sqrt{(b*c)/(p*(p-a))}} \quad ,\backslash$$

$$D_{\{a\}}=a\cdot\sqrt{\frac{bc}{(p-b)(p-c)}}\rightarrow \boxed{DF==a*\sqrt{(b*c/((p-b)*(p-c)))}} \quad .$$

Перемещая независимые точки A , B и C по плоскости, мы всегда будем получать в правой части второй и третьей строки надписи значение *true*, что и требовалось проверить.

5.3. Центроид

В энциклопедии Кимберлинга центроид (точка пересечения медиан треугольника, центр тяжести, центр масс, барицентр) именуется как $X(2)$, и для $\triangle ABC$ через строку ввода его можно строить командой *TriangleCenter*[$A,B,C,2$]. Мы это будем делать с помощью встроенных инструментов, учитывая, что центроид является точкой пересечения медиан треугольника.






Справедливы утверждения:

1) Центроид $\triangle ABC$ по трем вершинам A , B и C можно находить как точку $((x[A]+x[B]+x[C])/3, (y[A]+y[B]+y[C])/3)$. Каждую медиану в направлении от вершины к стороне центроид делит в отношении 2:1. Если d – длина медианы, то $d^2=AB^2/2+AC^2/2-BC^2/4$. Центроид, соединенный с вершинами, разбивает исходный треугольник на три равновеликих треугольника, то есть на три треугольника, имеющих равные площади. Центроид, соединенный с серединами сторон, разбивает исходный треугольник на три равновеликих четырехугольника.

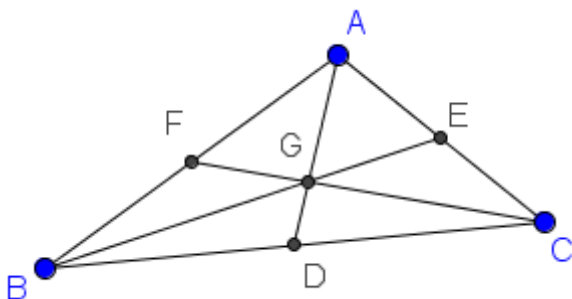
2) *Теорема Лейбница*. Сумма квадратов расстояний от центроида до вершин меньше, чем эта сумма от любой другой точки.

3) Начнем с введения понятия. Чевией называют отрезок, соединяющий вершину $\triangle ABC$ с какой-либо точкой на его противоположной стороне. Будем рассматривать тройки чевий, исходящих из разных вершин и пересекающихся в одной точке внутри треугольника, например, 3 биссектрисы внутренних углов, 3 медианы и т. д. В этом случае стороны треугольника делятся на 6 отрезков. А теперь утверждение. Произведение трех из шести отрезков без общих концов, на которые стороны треугольника делятся чевиями, проходящими через центроид, больше такого же произведения для трех чевий, проходящих через любую другую точку внутри треугольника.

Построение центроида. Изображение, представленное на рис. 2а, является результатом построения центроида как точки пересечения медиан треугольника. Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек;
- инструментом “ Точка” сформируем три точки A , B и C ;
- инструментом “ Отрезок” создадим $\triangle ABC$;
- инструментом “ Середина или центр” сформируем точки на серединах сторон треугольника;
- инструментом “ Отрезок” проведем три медианы, соединив полученные точки с вершинами противоположных вершин треугольника ABC ;
- инструментом “ Точка на объекте” сформируем центроид G ;

а)



б)

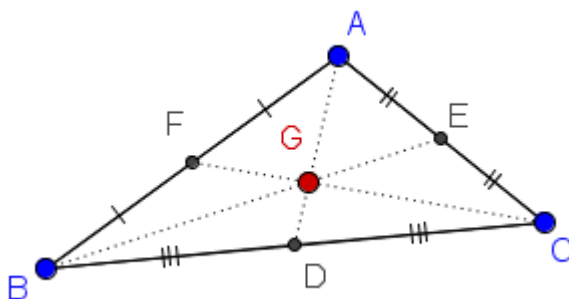




Рис. 2 (а, б). Построение центроида с помощью встроенных инструментов

Рис. 2б получен редактированием рис 2а, в котором изменен стиль вывода медиан, размер и цвет центроида G , а также добавлены одинаковые риски на равных по длине отрезках, на которые медианы делят соответствующие стороны треугольника. Для реализации последнего действия потребовалось сначала провести отрезки BF , FA , AE , EC , CD и DB , а затем уже через панель “Настройки” ставить на них указанные риски (декорации).

Пример 4. Пусть G – центроид $\triangle ABC$. Создать динамическую модель для экспериментальной проверки утверждений:

- Центроид G и точка $((x[A]+x[B]+x[C])/3, (y[A]+y[B]+y[C])/3)$ совпадают;
- Если d – длина медианы, то $d^2 = AB^2/2 + AC^2/2 - BC^2/4$.

Решение:

- продолжим работу с изображением рис. 2б и инструментом “ Отрезок” проведем медиану AD . Через контекстное меню сделаем ее имя d видимым;
- инструментом “ Текст” создадим проверочную надпись с вычисляемыми правыми частями, которые и будут сигнализировать нам о выполнении или невыполнении сформулированных утверждений:

$$G=((x[A]+x[B]+x[C])/3, (y[A]+y[B]+y[C])/3) \rightarrow$$

$$\boxed{G==((x[A]+x[B]+x[C])/3, (y[A]+y[B]+y[C])/3)} ,$$

$$d^2=AB^2/2+AC^2/2-BC^2/4 \rightarrow \boxed{d^2==AB^2/2+AC^2/2-BC^2/4} .$$

Результат этих действий можно видеть на рис 2с. При перемещении независимых точек A , B или C по плоскости изменяемые части надписей автоматически будут перевычисляться, но их значение будет оставаться равным *true*, что и требовалось проверить.

$$G = ((x[A] + x[B] + x[C])/3, (y[A] + y[B] + y[C])/3) \rightarrow true,$$

$$d^2 = AB^2/2 + AC^2/2 - BC^2/4 \rightarrow true.$$

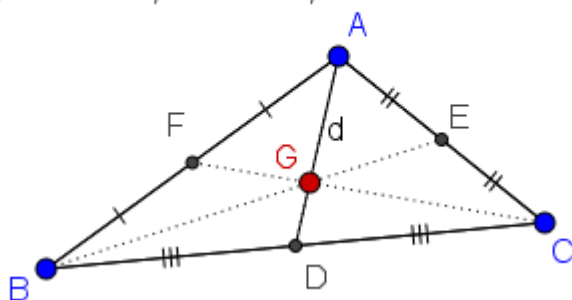



Рис. 2с. Модель для экспериментальной проверки утверждения примера 4


Пример 5. Пусть G – центроид $\triangle ABC$. Создать динамическую модель для экспериментальной проверки следующих утверждений:

1) центроид G , соединенный с вершинами $\triangle ABC$, разбивает его на три равновеликих треугольника, то есть площади треугольников GAB , GBC и GCA равны.

2) центроид, соединенный с серединами сторон $\triangle ABC$, разбивает его на три равновеликих четырехугольника,

Решение:

- продолжим работу с изображением рис. 2b. Инструментом “ Многоугольник” сформируем три треугольника GAB , GBC и GAC , а также три четырехугольника $AFGE$, $CEGD$ и $BDGF$. На панели “Полотно” можно видеть, что треугольники получили имена *poly1*, *poly2* и *poly3*, а четырехугольники – *poly4*, *poly5* и *poly6*;

- инструментом “ Текст” выведем две надписи с вычисляемыми частями, которые и будут сигнализировать нам о выполнении или невыполнении сформулированных выше утверждений. Формировать надписи будем в *LaTeX*-режиме, то есть при включенной кнопке *LaTeX*-формула.

Для проверки утверждения 1 надпись создадим вводом текста:

$$S_{\{\triangle AGB\}}=S_{\{\triangle BGC\}} \rightarrow \boxed{poly1==poly2} \setminus\setminus ,$$

$$\backslash vspace{6mm}$$

$$S_{\{\Delta AGB\}}=S_{\{\Delta CGA\}}\rightarrow \boxed{poly1==poly3} \quad .$$

Для проверки утверждения 2 надпись создадим вводом текста:

$$(S_{\{\AFGE\}}=S_{\{\CEGD\}})\rightarrow \boxed{poly4==poly5} \quad \backslash\backslash ,$$

$$(S_{\{\AFGE\}}=S_{\{\BFGD\}})\rightarrow \boxed{poly4==poly6} \quad .$$

Результат этих действий можно видеть на рис 2d. При перемещении независимых точек A , B или C по плоскости вместе с ΔABC будет меняться центрoид G , треугольники AGB , BGC и CGA , четырехугольники $AFGE$, $CEGD$ и $BFGD$, а также их площади. При этом изменяемые части надписей автоматически будут перевычисляться, но их значение будет оставаться равным *true*, что и требовалось проверить.

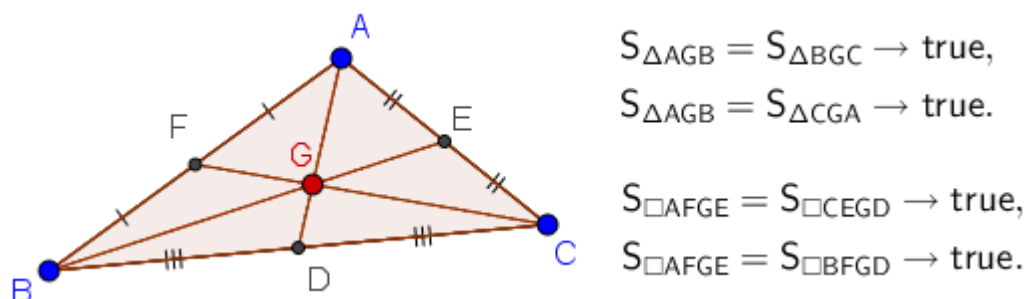









Рис. 2d. Модель для экспериментальной проверки утверждений примера 5

Пример 6. Создать динамическую модель для экспериментальной проверки утверждения: “Сумма квадратов расстояний от центрoида G до вершин ΔABC меньше чем, эта сумма от любой другой точки D ”.

Решение. Требуемая модель представлена на рис. 2e. Создавалась она следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек;
- инструментом “ Точка” сформируем три точки A , B и C ;
- инструментом “ Отрезок” создадим ΔABC ;
- командой $G=TriangleCenter[A,B,C,2]$ через строку ввода сформируем центрoид G ΔABC ;
- инструментом “ Отрезок” соединим точку G с вершинами ΔABC ;
- инструментом “ Точка” сформируем на плоскости произвольную точку D ;
- инструментом “ Отрезок” соединим точку D с вершинами ΔABC ;
- инструментом “ Прямая” вдоль отрезков AG , BG , CG , AD , BD и CD проведем линии и изменим их стиль на точечный;
- инструментом “ Текст” в *LaTeX*-режиме с помощью выражений

$$GA^2 + GB^2 + GC^2 = \boxed{GA^2 + GB^2 + GC^2}, \\ DA^2 + DB^2 + DC^2 = \boxed{DA^2 + DB^2 + DC^2}.$$

создадим проверочную надпись.

Точки A , B , C и D являются независимыми. Перемещая их можно наблюдать изменение значений в правых частях обеих строк надписи. Если D не совпадает с G , то во всех случаях значение первой строки оказывается меньше значения второй строки, что и требовалось проверить.

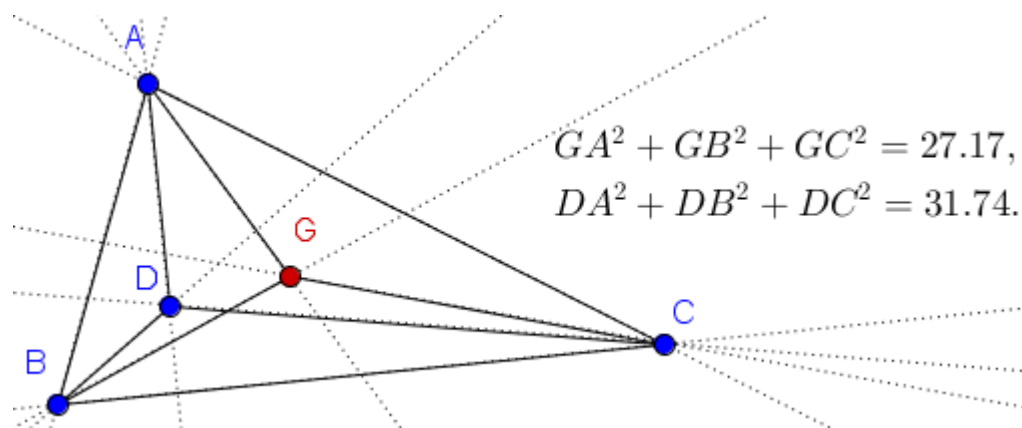



Рис. 2е. Модель для экспериментальной проверки утверждения примера 6


Пример 7. Создать динамическую модель для экспериментальной проверки утверждения: “Произведение трех из шести отрезков без общих концов, на которые стороны треугольника делятся чевианами, проходящими через центроид, больше такого же произведения для трех чевиан, проходящих через любую другую точку внутри треугольника”.

Решение. Произведение трех из шести отрезков без общих концов, на которые стороны треугольника делятся чевианами, проходящими через центроид, легко считается – это $a \cdot b \cdot c / 8$, где a , b и c – длины сторон треугольника. Поэтому надо лишь показать, что для любой другой точки D такое произведение отрезков не превосходит $a \cdot b \cdot c / 8$. При этом какую брать для произведения тройку отрезков без общих концов роли не играет. Согласно теоремы Чевы такие произведения равны. Требуемая модель представлена на рис. 2ф. Создавалась она следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментом “ Точка” сформируем три точки A , B и C . Инструментом “ Отрезок” создадим $\triangle ABC$;

- инструментом “ Точка на объекте” сформируем произвольную точку D внутри $\triangle ABC$. Инструментом “ Прямая” проведем прямые из каждой вершины треугольника через точку D . Инструментом “ Точка на объекте” сформируем точки E , F и H пересечения этих прямых со сторонами

$\triangle ABC$ (см. рис 2f). Инструментом “ Отрезок” проведем отрезки из вершин треугольника в созданные точки (необязательно);

- инструментом “ Текст” в *LaTeX*-режиме с помощью выражений

$$8AE \cdot HB \cdot FC \leq a \cdot b \cdot c \rightarrow \boxed{8AE*HB*FC \leq a*b*c}, \backslash \backslash$$

$$AE \cdot HB \cdot FC = CE \cdot HA \cdot FB \rightarrow \boxed{AE*HB*FC == CE*HA*FB} .$$

создадим проверочную надпись. Вторая ее строка необязательна. По ней проверяется утверждение теоремы Чебы.

Точки A, B, C и D являются независимыми. Перемещая их, можно наблюдать, что если D находится внутри $\triangle ABC$, то значения в правых вычисляемых частях надписи всегда равны *true*, что и требовалось проверить.

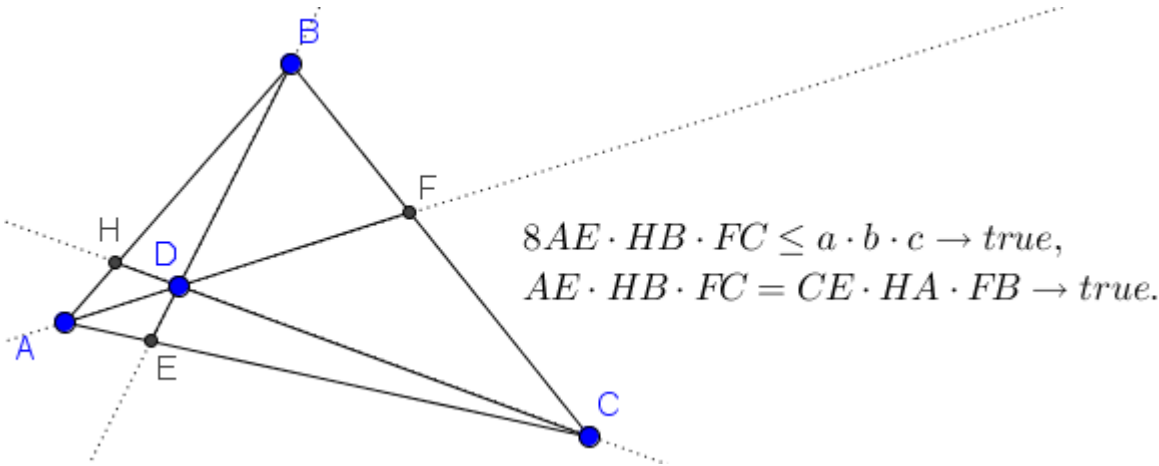


Рис. 2f. Модель для экспериментальной проверки утверждения примера 7

5.4. Центр описанной окружности

В энциклопедии Кимберлинга центр описанной окружности, то есть точка пересечения “*серединных*” перпендикуляров (перпендикуляров к серединам сторон треугольника) именуется как $X(3)$, и для $\triangle ABC$ через строку ввода его можно строить командой *TriangleCenter*[$A,B,C,3$].

Определение. Средние линии $\triangle ABC$ разбивают его на четыре равных треугольника, подобных $\triangle ABC$, с коэффициентом подобия $1/2$. Треугольник из средних линий называется *срединным* треугольником, другие три треугольника – *отсеченными* треугольниками.

Справедливы такие утверждения:

- 1) Окружности, описанные вокруг отсеченных треугольников, пересекаются в одной точке, которая является центром описанной окружности исходного треугольника.

2) *Формула Эйлера*. Если в треугольнике R , r и d – соответственно радиус описанной окружности, радиус вписанной окружности и расстояние между центрами этих окружностей, то имеет место формула Эйлера: $d^2 = R^2 - 2 \cdot R \cdot r$.

3) *Теорема Симсона*. Если в $\triangle ABC$ некоторая точка D расположена на его описанной окружности, то основания перпендикуляров, опущенных из D на стороны треугольника или их продолжения, лежат на одной прямой. Эта прямая называется прямой Симсона.

4) *Теорема Мансиона*. Отрезок, соединяющий центры **вписанной** и **вне-вписанной** окружностей треугольника, делится описанной окружностью пополам.

5) *Теорема о трезубце* (теорема трилистника, теорема Клайнера). Если H – точка пересечения биссектрисы угла A с описанной окружностью треугольника ABC , D и G – соответственно центры вписанной и невписанной окружностей, касающихся стороны BC , то $HD = HG = HB = HC$.

6) *Соотношение между радиусами*. Если в треугольнике R , r , r_a , r_b , r_c – соответственно радиус описанной окружности, радиус вписанной окружности и радиусы невписанных окружностей, то имеет место формула




$$r_a + r_b + r_c = 4R + r.$$




7) *Формула Карно*. Пусть в $\triangle ABC$ точка D есть центр описанной окружности, d_1 , d_2 и d_3 – расстояния от D до сторон треугольника, R – радиус описанной окружности и r – радиус вписанной окружности. Тогда




$$\pm d_1 \pm d_2 \pm d_3 = R + r, \quad (3)$$

где слагаемые d_1 , d_2 и d_3 в левой части формулы берутся со знаком минус, когда высота из D целиком лежит вне $\triangle ABC$, и со знаком плюс – в противном случае.

Построение центра описанной окружности. Изображение, представленное на рис. 3а, является результатом построения центра описанной окружности как точки пересечения перпендикуляров к серединам сторон треугольника. Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментом “ Точка” сформируем три точки A , B и C . Инструментом “ Прямая” проведем прямые через пары точек A и B , B и C , C и A . Инструментом “ Отрезок” проведем стороны $\triangle ABC$;

- инструментом “ Середина или центр” сформируем средние точки D , E и F сторон треугольника. Инструментом “ Перпендикулярная прямая” в средних точках проведем перпендикулярные линии к сторонам треугольника. Инструментом “ Точка на объекте” сформируем центр описанной окружности как точку пересечения построенных перпендикуляров и изменим ее имя на O ;

- инструментом “ Отрезок” соединим точку O с серединами сторон треугольника (необязательно). Инструментом “ Окружность по центру и точке” проведем описанную окружность (центр – O , точка – A);
- инструментом “ Угол” проставим метки прямых углов у оснований перпендикуляров. Изменим стиль вывода всех прямых (но не отрезков) на точечный;
- добавим одинаковые риски на равные по длине отрезки сторон треугольника. Для реализации этого действия предварительно следует сформировать указанные отрезки, а затем уже через панель “Настройки” ставить на них риски.

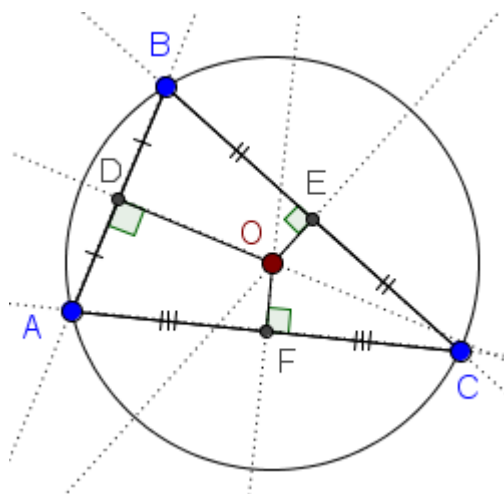








Рис. 3а. Построение центра описанной окружности


Пример 8. Построить модель для экспериментальной проверки утверждения: “Окружности, описанные вокруг отсеченных треугольников, пересекаются в одной точке, которая является центром описанной окружности исходного $\triangle ABC$ ”.

Решение. Требуемая модель представлена на рис. 3б. Создавалась она следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментом “ Точка” сформируем три точки A , B и C . Инструментом “ Прямая” проведем прямые через пары точек A и B , B и C , C и A . Инструментом “ Отрезок” проведем стороны $\triangle ABC$;
- инструментом “ Середина или центр” сформируем средние точки сторон треугольника и через панель “Настройки” добавим одинаковые риски на равные по длине отрезки сторон треугольника;
- инструментом “ Окружность по трем точкам” проведем описанные окружности вокруг отсеченных треугольников и исходного треугольника ABC ;

- проведенные окружности оказались пересекающимися в одной точке.

Инструментом “ Точка на объекте” сформируем их точку пересечения и переименуем ее на O ;

- инструментом “ Окружность по центру и точке ” проведем описанную окружность (центр – O , точка – A);
- командой

$$\text{Intersect}[\text{PerpendicularBisector}[\text{Segment}[A,C]], \\ \text{PerpendicularBisector}[\text{Segment}[B,C]]]$$

через строку ввода определим центр G описанной вокруг $\triangle ABC$ окружности. На рис. 3b точек O и G оказываются совпадающими;

- инструментом “ Текст” с помощью выражений

$$O=G \rightarrow \boxed{O==G}, \backslash$$

O на пересечении окружностей \rightarrow

$$\boxed{\text{AreConcyclic}[A,F,D,O] \wedge \text{AreConcyclic}[F,B,E,O] \wedge \text{AreConcyclic}[D,E,C,O]} .$$

создадим проверочную надпись.

При перемещениях независимых точек A, B, C можно видеть, что во всех ситуациях “малые” окружности остаются пересекающимися в одной точке G , а точки O и G остаются совпадающими. Этот факт подтверждается надписью, что и требовалось проверить.

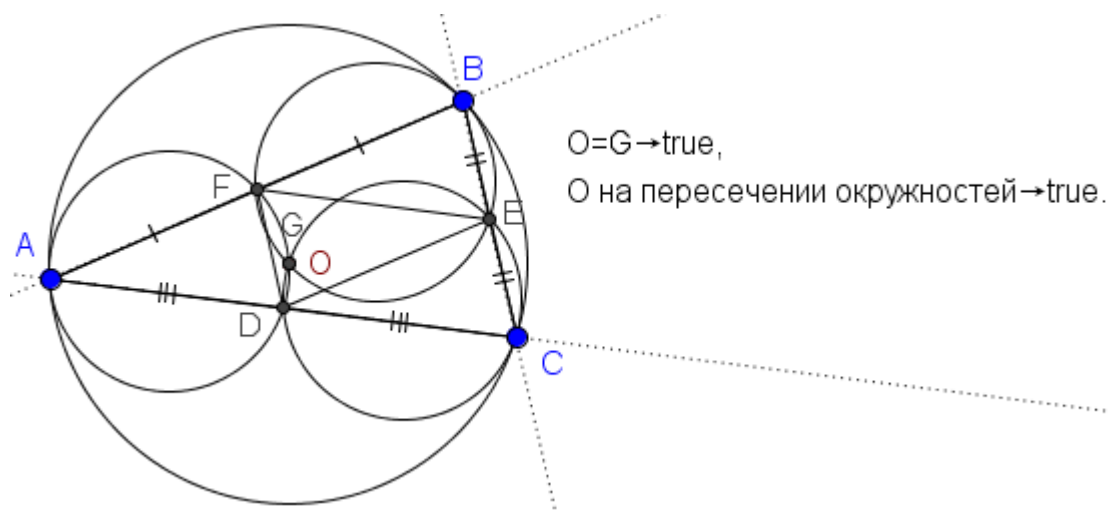









Рис. 3b. Модель для экспериментальной проверки утверждения примера 8

Пример 9. Пусть имеется треугольник и R , r и d – радиус его описанной окружности, радиус вписанной окружности и расстояние между центрами

этих окружностей. Построить модель для экспериментальной проверки формулы Эйлера: $d^2=R^2-2\cdot R\cdot r$.

Решение. Искомая модель, представленная на рис. 3с, создавалась следующим образом:

- построим, как мы это уже неоднократно делали, $\triangle ABC$;
- командами $TriangleCenter[A,B,C,3]$ и $TriangleCenter[A,B,C,1]$ сформируем точку D – центр описанной окружности и точку E – центр вписанной окружности;
- инструментом “ Перпендикулярная прямая” из точки E проведем перпендикулярную линию к стороне BC . Инструментом “ Точка на объекте” создадим точку F пересечения этой линии с BC . Инструментом “ Угол” в точке F поставим метку прямого угла;
- инструментом “ Окружность по трем точкам” проведем описанную окружность $\triangle ABC$ (необязательно). Инструментом “ Окружность по центру и точке” проведем вписанную окружность $\triangle ABC$ (необязательно);
- инструментом “ Отрезок” проведем отрезки: AD – радиус описанной окружности, EF – радиус вписанной окружности, DE – отрезок между центрами D и E . Через контекстное меню изменим обозначения этих отрезков соответственно на R , r и d ;
- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 9:

$$d^2= \boxed{DE^2} ,\\ R^2-2Rr= \boxed{AD^2-2\cdot AD\cdot EF} .$$

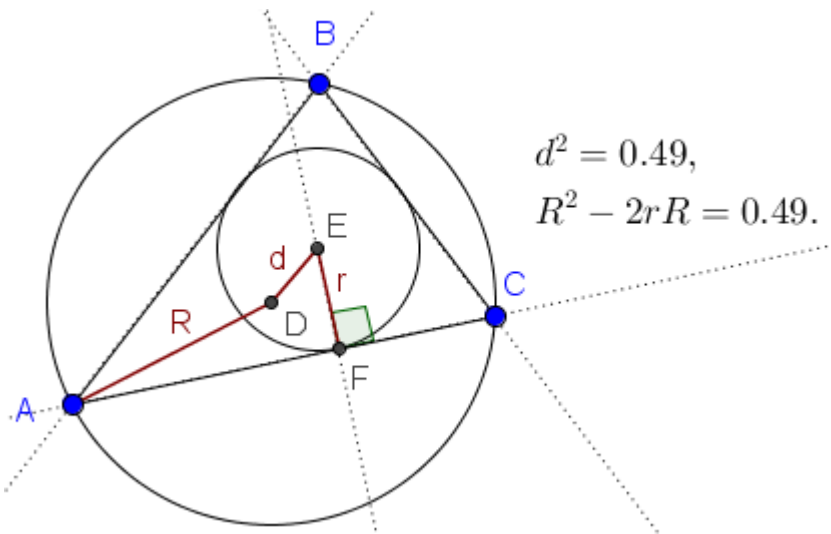








Рис. 3с. Модель для экспериментальной проверки формулы Эйлера ($d^2=R^2-2rR$)

Экспериментируя с полученной моделью, мы можем наблюдать совпадение изменяющихся численных результатов для d^2 и R^2-2Rr , что и требовалось

проверить. Кроме того, изменяющееся изображение продемонстрирует все возможные варианты, которые необходимо рассмотреть при действительном доказательстве данного утверждения Эйлера.

Пример 10. Пусть имеется $\triangle ABC$ и D – некоторая точка на его описанной окружности. Построить модель для экспериментальной проверки теоремы Симсона: “Основания перпендикуляров, опущенных из D на стороны треугольника или их продолжения, лежат на одной прямой”.

Решение. Искомая модель, представленная на рис. 3d, создавалась следующим образом:

- построим $\triangle ABC$ и инструментом “ Окружность по трем точкам” проведем описанную окружность;
- инструментом “ Точка на объекте” сформируем на окружности некоторую точку D ;
- инструментом “ Перпендикулярная прямая” проведем перпендикулярные линии к сторонам треугольника или к их продолжениям;
- инструментом “ Точка на объекте” сформируем точки пересечения E , G и F проведенных линий со сторонами $\triangle ABC$ или их продолжениями. Через панель “Настройки” изменим стиль вывода точек E , G и F и создадим метки прямых углов (см. рис. 3d);
- инструментом “ Прямая” проведем прямую через точки E и F . Можно видеть, что точка G также окажется на этой прямой;
- инструментом “ Текст” сформируем надпись, подтверждающую или опровергающую принадлежность точки G проведенной прямой EF :

$$\text{точка } G \text{ лежит на прямой } EF \rightarrow \boxed{\text{AreCollinear}[E,G,F]} .$$

Перемещая независимые точки A , B и C по плоскости или точку D по окружности мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

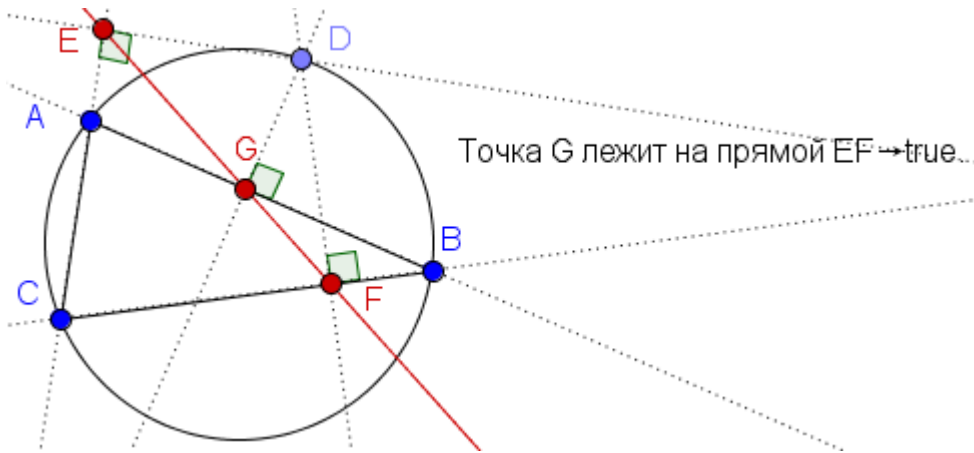












Рис. 3d. Модель для экспериментальной проверки теоремы Симсона

Пример 11. Построить модель для экспериментальной проверки теоремы Мансиона: “Отрезок, соединяющий центры **вписанной** и **вневписанной** окружностей треугольника, делится описанной окружностью пополам”.

Решение. Искомая модель, представленная на рис. 3е, создавалась следующим образом:

- построим $\triangle ABC$ и инструментом “ Окружность по трем точкам” проведем описанную окружность;
- инструментом “ Биссектриса угла” проведем биссектрисы внутренних и внешних углов $\triangle ABC$;
- инструментом “ Точка на объекте” сформируем точку D пересечения биссектрис внутренних углов, то есть центр вписанной окружности, и точку пересечения биссектрис внешних углов при вершинах B и C , то есть центр одной из вневписанных окружностей;
- (необязательно) инструментом “ Перпендикулярная прямая” из точки E проведем перпендикулярную линию к BC . Инструментом “ Точка на объекте” сформируем точку F пересечения этой линии с BC и сделаем ее невидимой. Инструментом “ Окружность по центру и точке” проведем вне-вписанную окружность (центр – E , точка – F);
- (необязательно) инструментом “ Перпендикулярная прямая” из точки D проведем перпендикулярную линию к BC . Инструментом “ Точка на объекте” сформируем точку G пересечения этой линии с BC и сделаем ее невидимой. Инструментом “ Окружность по центру и точке” проведем вписанную окружность (центр – D , точка – G);
- инструментом “ Отрезок” проведем отрезок DE ;
- командой $Intersect[g,s]$ сформируем точку H – пересечения отрезка DE (s) с описанной окружностью (g);

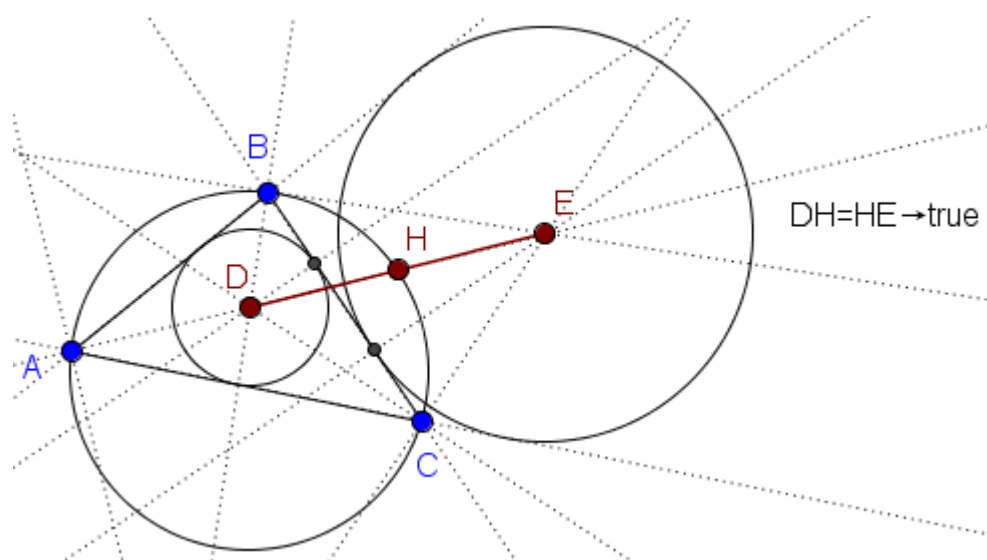



Рис. 3е. Модель для экспериментальной проверки теоремы Мансиона



- инструментом “ Текст” создадим проверочную надпись, подтверждающую или опровергающую утверждение примера 11:

$$DH=HE \rightarrow \boxed{DH==HE} .$$

Перемещая независимые точки A , B и C по плоскости мы всегда будем получать в правой вычисляемой части проверочной надписи значение *true*, что и требовалось проверить.

Пример 12. Построить модель для экспериментальной проверки теоремы о трезубце: “Если H – точка пересечения биссектрисы угла A с описанной окружностью $\triangle ABC$, D и G – соответственно центры вписанной и внеписанной окружностей, касающихся стороны BC , то $HD=HG=HB=HC$.”

Решение. Искомая модель, представленная на рис. 3f, создавалась следующим образом:

- продолжим работу, начав с изображения рис. 3е. Инструментом “ Отрезок” проведем отрезки BH и CH ;
- инструментом “ Текст” модифицируем надпись, дополнив ее двумя строками:

$$\begin{aligned} DH=HE &\rightarrow \boxed{DH==HE} , \\ DH=HB &\rightarrow \boxed{DH==HB} , \\ DH=HC &\rightarrow \boxed{DH==HC} . \end{aligned}$$

Перемещая независимые точки A , B и C по плоскости мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

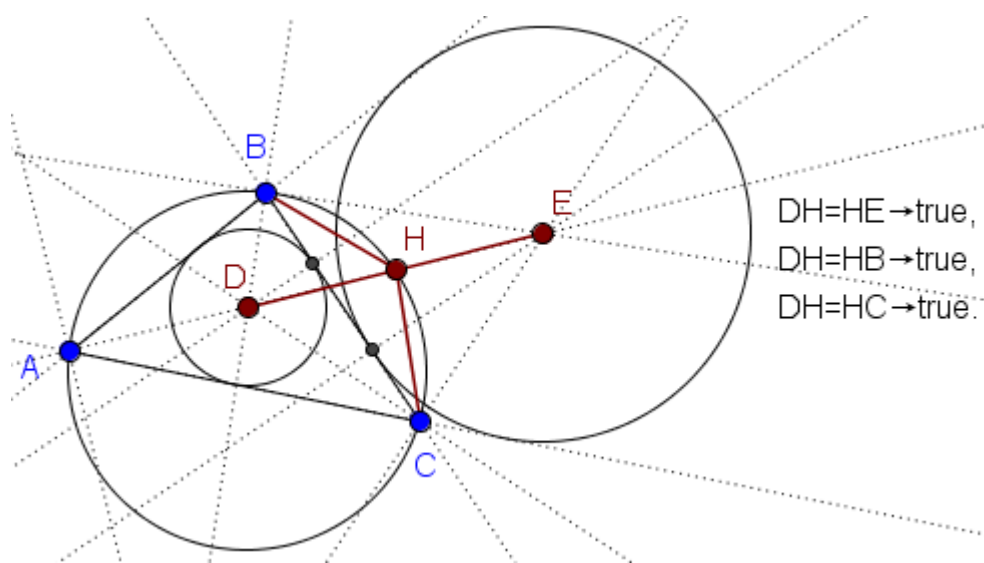











Рис. 3f. Модель для экспериментальной проверки теоремы о трезубце



Пример 13. Построить модель для экспериментальной проверки утверждения “ $r_a + r_b + r_c = 4R + r$ ”, где R – радиус описанной окружности, r – радиус вписанной окружности и r_a, r_b, r_c – радиусы внеписанных окружностей.



Решение. Искомая модель, представленная на рис. 3g, создавалась следующим образом:


- инструментами “ Точка”, “ Прямая” и “ Отрезок” построим $\triangle ABC$. Инструментом “ Окружность по трем точкам” проведем описанную окружность $\triangle ABC$ (необязательно);

- инструментом “ Биссектриса угла” проведем внутренние и внешние биссектрисы углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точку D пересечения биссектрис внутренних углов (D – центр вписанной окружности) и точки E, F и G пересечения биссектрис внешних углов (E, F и G – центры внеписанных окружностей);


- инструментом “ Перпендикулярная прямая” проведем через точку D линию, перпендикулярную к стороне AC . Инструментом “ Точка на объекте” сформируем точку H пересечения этой линии с AC . Инструментом “ Окружность по центру и точке” проведем вписанную окружность (центр – D , точка – H) (необязательно);

- инструментом “ Срединный перпендикуляр” проведем перпендикуляры к серединам сторон BC и AC . Инструментом “ Точка на объекте” сформируем точку I пересечения этих перпендикуляров (I – центр описанной окружности);

- инструментом “ Перпендикулярная прямая” из центров внеписанных окружностей E, F и G проведем перпендикулярные линии к стороне AC или ее продолжению. Инструментом “ Точка на объекте” сформируем точки пересечения J, K и L этих перпендикуляров со стороной AC или ее продолжением;

- инструментом “ Отрезок” сформируем отрезки EJ, GK, FL, IC и DH , равные радиусам внеписанных, описанной и вписанной окружностей;

- переименуем через контекстное меню отрезки EJ, GK, FL, IC и DH соответственно на r_c, r_b, r_a, R и r ;

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 13:

$$r_{\{a\}} + r_{\{b\}} + r_{\{c\}} = r + 4R \rightarrow \boxed{FL + GK + EJ = r + 4 * R} .$$

Перемещая независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части проверочной надписи значение *true*, что и требовалось проверить.

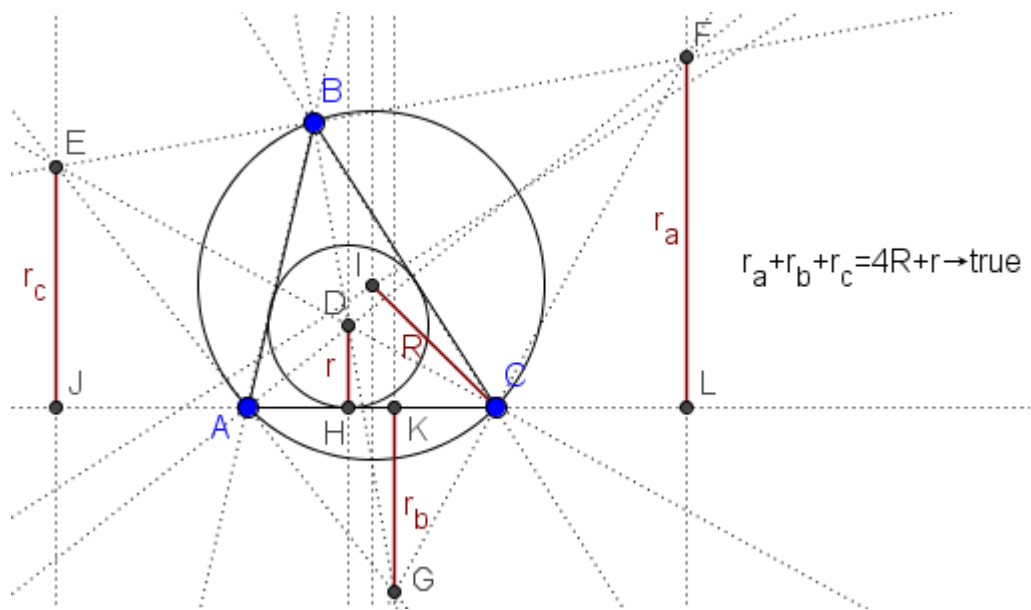












Рис. 3г. Модель для экспериментальной проверки утверждения примера 13


Пример 14. Построить модель для экспериментальной проверки формулы Карно (3).

Решение. Искомая модель, представленная на рис. 3h, создавалась следующим образом:

- инструментами “ Точка”, “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Срединный перпендикуляр” проведем перпендикуляры к серединам сторон $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точку D пересечения этих перпендикуляров (центр описанной окружности) и точки пересечения E , F и I их пересечения со сторонами треугольника. Инструментом “ Окружность по центру и точке” проведем описанную окружность (центр – D , точка – B);
- инструментом “ Биссектриса угла” проведем внутренние биссектрисы $\angle ABC$ и $\angle BCF$. Инструментом “ Точка на объекте” сформируем точку пересечения H одной из биссектрис со стороной AC и точку пересечения G самих биссектрис. Инструментом “ Окружность по центру и точке” проведем вписанную окружность (центр – G , точка – H);
- инструментом “ Отрезок” проведем отрезки DF , DI , DE , DB и GH . Через контекстное меню заменим имена отрезков DB и GH соответственно на R и r ;

- через строку ввода сформируем вспомогательные переменные s_1, s_2 и s_3 со значениями ± 1 :

$$\begin{aligned} s_{\{1\}} &= \text{If}[\text{If}[\text{Angle}[A,B,C]>180^\circ, \text{Angle}[C,B,A], \text{Angle}[A,B,C]]>90^\circ, -1, 1], \\ s_{\{2\}} &= \text{If}[\text{If}[\text{Angle}[B,A,C]>180^\circ, \text{Angle}[C,A,B], \text{Angle}[B,A,C]]>90^\circ, -1, 1], \\ s_{\{3\}} &= \text{If}[\text{If}[\text{Angle}[B,C,A]>180^\circ, \text{Angle}[A,C,B], \text{Angle}[B,C,A]]>90^\circ, -1, 1]; \end{aligned}$$

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую формулу Карно:

$$s_{\{1\}} \cdot DE + s_{\{2\}} \cdot DI + s_{\{3\}} \cdot DF = R + r \rightarrow$$

$s_{\{1\}} \cdot DE + s_{\{2\}} \cdot DI + s_{\{3\}} \cdot DF = R + r$

Перемещая независимые точки A, B и C по плоскости, мы всегда будем получать в правой вычисляемой части проверочной надписи значение *true*, что и требовалось проверить.

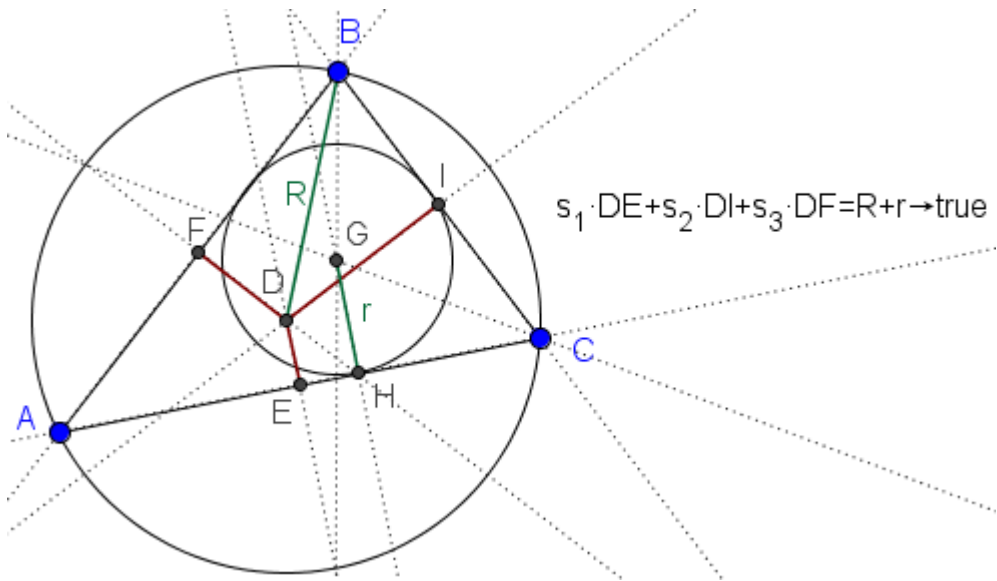


Рис. 3h. Модель для экспериментальной проверки формулы Карно

5.5. Ортоцентр

В энциклопедии Кимберлинга ортоцентр, то есть центр пересечения высот треугольника или их продолжений, именуется как X(4), и для $\triangle ABC$ через строку ввода его можно строить командой *TriangleCenter*[A,B,C,4].

- Определения.* 1. Треугольник с вершинами в основаниях высот треугольника исходного треугольника называют ортотреугольником.
2. Четверку точек из трех вершин треугольника и его ортоцентра называют ортоцентрической системой.

Справедливы такие утверждения:

1) Вершины остроугольного треугольника являются центрами вневписанных окружностей его ортотреугольника.

2) Расстояние от вершины треугольника до ортоцентра вдвое больше, чем расстояние от центра описанной окружности до противоположной стороны для этой вершины.

3) Ортоцентр остроугольного треугольника ABC делит его высоту из вершины A на отрезки, отношение которых, считая от A , равно $\frac{\cos(A)}{\cos(B) \cdot \cos(C)}$.

Аналогичные соотношения имеют место и для вершин B и C .

4) Ортотреугольник остроугольного $\triangle ABC$ имеет наименьший периметр из всех треугольников, вписанных в $\triangle ABC$.

5) Точки, симметричные ортоцентру треугольника относительно его сторон, лежат на описанной окружности. Точки, симметричные ортоцентру треугольника относительно середин его сторон, также лежат на описанной окружности и совпадают с точками, диаметрально противоположными соответствующим вершинам.




6) *Пять окружностей.* У всех четырех треугольников с вершинами в ортоцентрической системе радиусы описанных окружностей равны. На окружности такого же радиуса лежат и отличные от D центры трех из этих окружностей.


7) Пусть a, b и c – стороны (и их длины) $\triangle ABC$, R – радиус описанной окружности, a_1 и a_2, b_1 и b_2, c_1 и c_2 – отрезки (и их длины), на которые ортоцентром делятся высоты, считая соответственно от вершин A, B, C . Тогда имеют место соотношения:


$$a_1 \cdot a_2 = b_1 \cdot b_2 = c_1 \cdot c_2, \quad (4)$$

$$a_1^2 + b_1^2 + c_1^2 + a^2 + b^2 + c^2 = 12R^2. \quad (5)$$

Построение ортоцентра. Изображение, представленное на рис. 4а, является результатом построения ортоцентра как центра пересечения высот треугольника или их продолжений. Создавалось оно следующим образом:

- инструментами “ Точка”, “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- инструментом “ Перпендикулярная прямая” через вершины к сторонам треугольника проведем перпендикулярные линии;

- инструментом “ Точка на объекте” сформируем точку D пересечения проведенных линий (ортоцентр) и точки E, F и G пересечения этих линий со сторонами треугольника или их продолжениями;

- создадим метки прямых углов, но не инструментами, а командами *Angle* через строку ввода. Выглядеть они могут, например, так:

$$IF[Angle[A,G,C]==90^\circ, Angle[A,G,C], Angle[C,G,A]],$$

$$IF[Angle[B,F,A]==90^\circ, Angle[B,F,A], Angle[A,F,B]],$$

$$IF[Angle[C,E,B]==90^\circ, Angle[C,E,B], Angle[B,E,C]].$$

Подобное задание меток обеспечивает правильность их вывода при динамических изменениях объектов. Например, при перемещении вершины A угол BAC стал тупым и ортоцентр D вышел за пределы $\triangle ABC$ (см. рис. 4b). И несмотря на изменение ориентации углов их метки остались правильными.

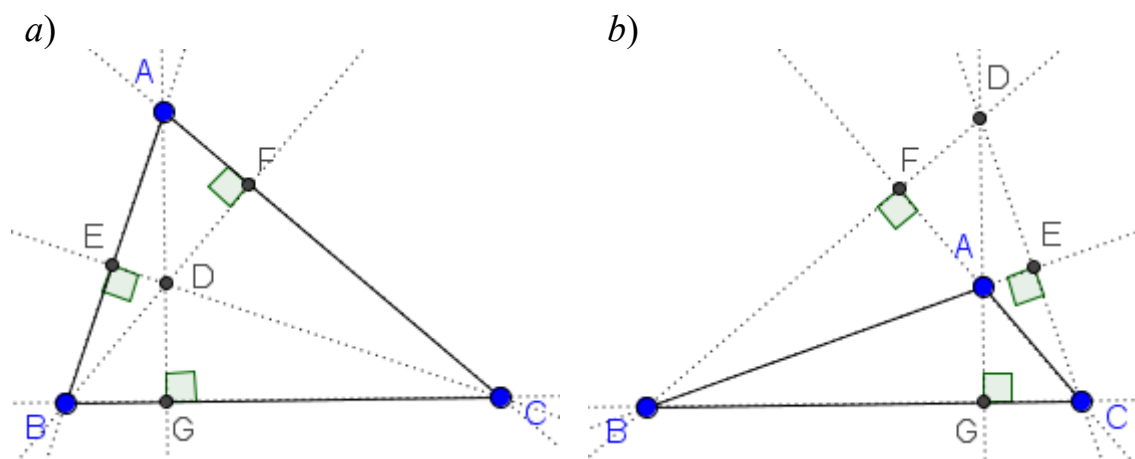









Рис. 4 (a, b). Построение ортоцентра с помощью встроенных инструментов





Пример 15. Построить модель для экспериментальной проверки утверждения: “Вершины остроугольного треугольника являются центрами вневписанных окружностей его ортотреугольника”.

Решение. Будем проверять приведенное утверждение для одной из вершин треугольника. Искомая модель, представленная на рис. 4с, формировалась следующим образом:

- инструментами “ Точка”, “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Перпендикулярная прямая” через вершины к сторонам треугольника проведем перпендикулярные линии. Инструментом “ Точка на объекте” сформируем ортоцентр D $\triangle ABC$ и вершины E, F, G его ортотреугольника;
- инструментами “ Отрезок” создадим ортотреугольник EFG . Инструментом “ Прямая” проведем линии вдоль сторон EFG ;
- через строку ввода командой

$$Intersect[AngleBisector[E,G,Reflect[F,G]], AngleBisector[G,E,Reflect[F,E]]]$$

сформируем центр H внеписанной окружности для стороны EG ортотреугольника. В данном случае (см. рис. 4с) вершина B исходного треугольника и центр внеписанной окружности H ортотреугольника совпадают;

- (необязательно) инструментом “ Перпендикулярная прямая” из H проведем перпендикулярную линию к EG . Инструментом “ Точка на объекте” сформируем точку пересечения J проведенной линии и EG . Инструментом “ Окружность по центру и точке” проведем внеписанную окружность (центр – H , точка – J);
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 15:

$$B=H \rightarrow \boxed{B==H} .$$

Перемещая независимые точки A, B и C по плоскости так, чтобы $\triangle ABC$ оставался остроугольным (точка D находится внутри $\triangle ABC$), мы всегда будем получать в правой вычисляемой части надписи значение *true*, то есть $B=H$, что и требовалось проверить.

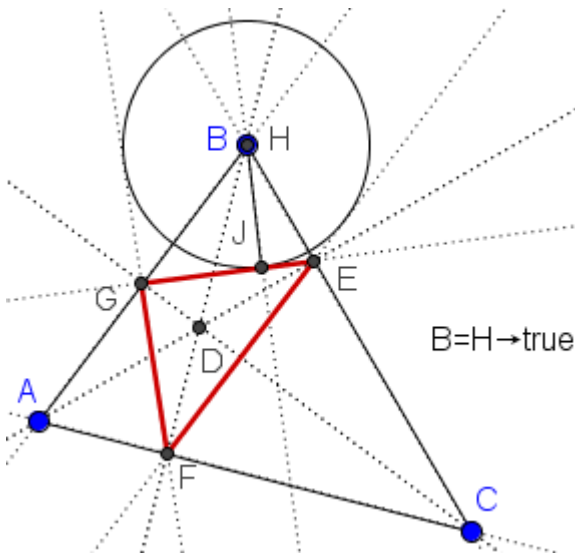











Рис. 4с. Модель для экспериментальной проверки утверждения примера 15

Пример 16. Построить модель для экспериментальной проверки утверждения: “Расстояние от вершины треугольника до ортоцентра вдвое больше, чем расстояние от центра **описанной окружности** до противоположной стороны для этой вершины”.

Решение. Искомая модель, представленная на рис. 4d, создавалась следующим образом:

- инструментами “ Точка”, “ Прямая” и “ Отрезок” построим $\triangle ABC$. Инструментом “ Окружность по трем точкам” проведем описанную окружность для этого треугольника (не обязательно);


- инструментом “ *Перпендикулярная прямая*” проведем перпендикуляры из вершин ко всем сторонам $\triangle ABC$. Инструментом “ *Точка на объекте*” сформируем ортоцентр D как точку пересечения этих перпендикуляров;
- инструментом “ *Срединный перпендикуляр*” проведем перпендикуляры к серединам сторон AC и BC $\triangle ABC$. Инструментом “ *Точка на объекте*” сформируем точки E и F пересечения созданных перпендикуляров со сторонами AC и BC , а также центр G описанной окружности как точку пересечения этих перпендикуляров;
- инструментом “ *Отрезок*” построим отрезки BD – от вершины до ортоцентра и GF – от центра описанной окружности до стороны AC ;
- (необязательно) командами

$$IF[Angle[Line[B,D], Line[C,A]]==90^\circ, \\ Angle[Line[B,D], Line[C,A]], Angle[Line[D,B], Line[C,A]]]$$

и

$$IF[Angle[Line[G,F], Line[C,A]]==90^\circ, \\ Angle[Line[G,F], Line[C,A]], Angle[Line[F,G], Line[C,A]]],$$

выполняемыми через строку ввода, проставим метки прямых углов и радиокнопками панели “*Объекты*” визуализируем их;

- инструментом “ *Текст*” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 16:

$$BD=2GF \rightarrow \boxed{BD==2*GF}.$$

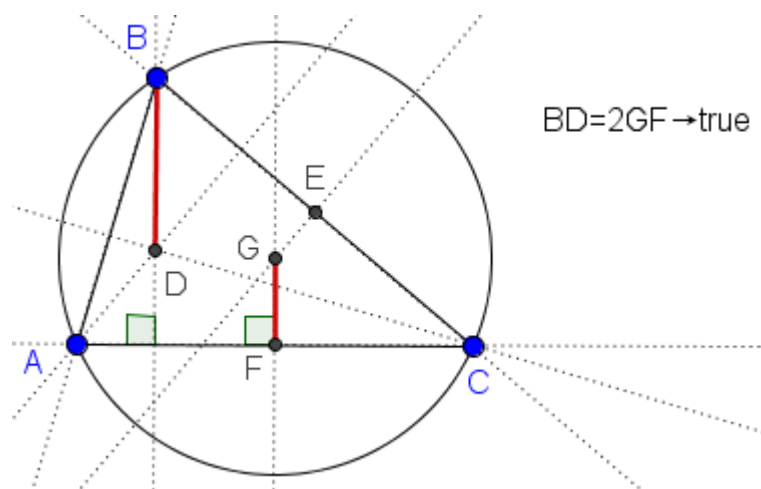









Рис. 4d. Модель для экспериментальной проверки утверждения примера 16

Перемещая независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

Пример 17. Построить модель для экспериментальной проверки утверждения: “Ортоцентр остроугольного $\triangle ABC$ делит его высоту из вершины A на отрезки с отношением, считая от A , равным $\frac{\cos(A)}{\cos(B) \cdot \cos(C)}$ ”.

Решение. Искомая модель, представленная на рис. 4е, создавалась следующим образом:

- инструментами “ Точка”, “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Перпендикулярная прямая” проведем перпендикуляры из вершин A и B к противоположным сторонам $\triangle ABC$ или к их продолжениям. Инструментом “ Точка на объекте” сформируем ортоцентр D как точку пересечения этих перпендикуляров, а также точки E и F – пересечения этих перпендикуляров со сторонами треугольника;
- инструментом “ Угол” в точках E и F создадим метки прямых углов и метки углов A, B, C (необязательно);
- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 17:

$$\frac{AD}{DE} = \frac{\cos(A)}{\cos(B)\cos(C)} \Rightarrow \boxed{AD/DE == \cos(\text{Angle}[B,A,C]) / (\cos(\text{Angle}[A,B,C]) * \cos(\text{Angle}[A,C,B]))} .$$

Отметим, что треугольник является остроугольным тогда и только тогда, когда его ортоцентр находится внутри треугольника. Перемещая независимые точки A, B и C по плоскости так, чтобы ортоцентр D оставался внутри $\triangle ABC$, мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

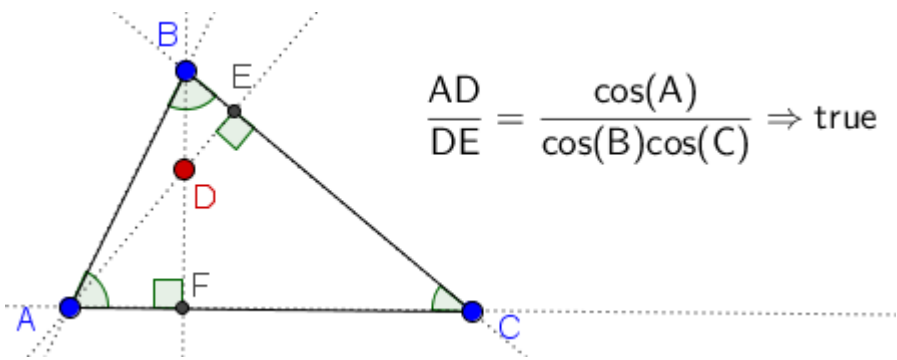


Рис. 4е. Модель для экспериментальной проверки утверждения примера 17

Пример 18. Построить модель для экспериментальной проверки утверждения: “Ортотреугольник остроугольного $\triangle ABC$ имеет наименьший периметр из всех треугольников, вписанных в ABC ”.

Решение. Искомая модель, представленная на рис. 4f, создавалась следующим образом:

- инструментами “Точка”, “Прямая” и “Отрезок” построим $\triangle ABC$;
- инструментами “Перпендикулярная прямая” проведем перпендикулярные линии из вершин к сторонам треугольника. Инструментом “Точка на объекте” сформируем ортоцентр D и точки E , F и G пересечения этих линий со сторонами треугольника;
- инструментом “Многоугольник” построим ортотреугольник EFG ;
- инструментом “Точка на объекте” прикрепим к сторонам $\triangle ABC$ произвольные точки H , I и J . Инструментом “Многоугольник” построим $\triangle HIJ$ и изменим стиль его вывода;
- инструментом “Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 18:

$$FG+GE+EF \leq HJ+JI+IH \rightarrow \boxed{FG+GE+EF \leq HJ+JI+IH}.$$

Перемещая независимые точки A , B и C по плоскости так, чтобы ортоцентр D оставался внутри $\triangle ABC$, или перемещая прикрепленные к сторонам треугольника точки H , I и J , мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

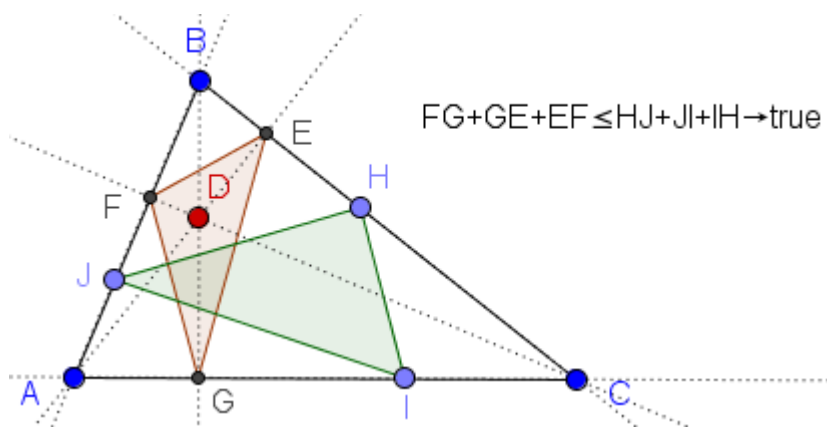
















Рис. 4f. Модель для экспериментальной проверки утверждения примера 18

Пример 19. Построить модель для экспериментальной проверки утверждения: “Точки, симметричные ортоцентру треугольника относительно его сторон или их продолжений, лежат на описанной окружности. Точки, симметричные ортоцентру треугольника относительно середин его сторон, также лежат на описанной окружности и совпадают с точками, диаметрально противоположными соответствующим вершинам”.

Решение. Искомая модель, представленная на рис. 4g, создавалась следующим образом:

- инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Перпендикулярная прямая” через вершины к сторонам треугольника проведем перпендикулярные линии. Инструментом “ Точка на объекте” сформируем ортоцентр D и точки E, F и G пересечения этих линий со сторонами треугольника. Инструментом “ Отрезок” соединим точки E, F и G с вершинами треугольника. Инструментом “ Угол” поставим метки прямых углов (необязательно);
- инструментом “ Середина или центр” создадим точки H, I и J , являющиеся серединами сторон $\triangle ABC$. Инструментом “ Перпендикулярная прямая” проведем перпендикулярные линии к серединам сторон AC и BC . Инструментом “ Точка на объекте” сформируем центр K описанного круга как точку пересечения этих перпендикуляров. Инструментом “ Окружность по трем точкам” проведем описанную окружность для $\triangle ABC$;
- инструментом “ Отражение относительно прямой” построим точки, симметричные ортоцентру D относительно сторон треугольника ABC . Через контекстное меню переименуем созданные точки на D_1, D_2 и D_3 . Можно видеть, что сформированные точки лежат на описанной окружности;
- инструментом “ Отражение относительно точки” построим точки H, I и J , симметричные ортоцентру относительно середин сторон $\triangle ABC$. Через контекстное меню переименуем созданные точки на D_3, D_4 и D_6 . Можно видеть, что сформированные точки лежат на описанной окружности;
- командами $A_1=Reflect[A,K], B_1=Reflect[B,K], C_1=Reflect[C,K]$ создадим точки A_1, B_1 и C_1 , симметричные вершинам A, B и C относительно центра K описанной окружности. Можно видеть, что точки A_1, B_1 и C_1 совпадают соответственно с точками D_4, D_5 и D_6 . Инструментом “ Отрезок” проведем отрезки AA_1, BB_1 и CC_1 и изменим стиль их вывода на точечный;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую все утверждения примера 19:

Точки D_1, D_2, D_3 и D_4 на одной окружности \rightarrow

$\boxed{AreConcyclic[D_1,D_2,D_3,D_4]}$,

Точки D_2, D_3, D_4 и D_5 на одной окружности \rightarrow

$\boxed{AreConcyclic[D_2,D_3,D_4,D_5]}$,

Точки D_3, D_4, D_5 и D_6 на одной окружности \rightarrow

$\boxed{AreConcyclic[D_3,D_4,D_5,D_6]}$,

$A_1=D_4 \rightarrow \boxed{A_1==D_4}$,

$$\begin{aligned}
 B_1=D_5 &\rightarrow \boxed{B_1==D_5}, \\
 C_1=D_6 &\rightarrow \boxed{C_1==D_6}, \\
 K \text{ середина отрезка } AA_1 &\rightarrow \boxed{K==\text{Midpoint}[A,A_1]}, \\
 K \text{ середина отрезка } BB_1 &\rightarrow \boxed{K==\text{Midpoint}[B,B_1]}, \\
 K \text{ середина отрезка } CC_1 &\rightarrow \boxed{K==\text{Midpoint}[C,C_1]}.
 \end{aligned}$$

Перемещая независимые точки A , B и C по плоскости мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

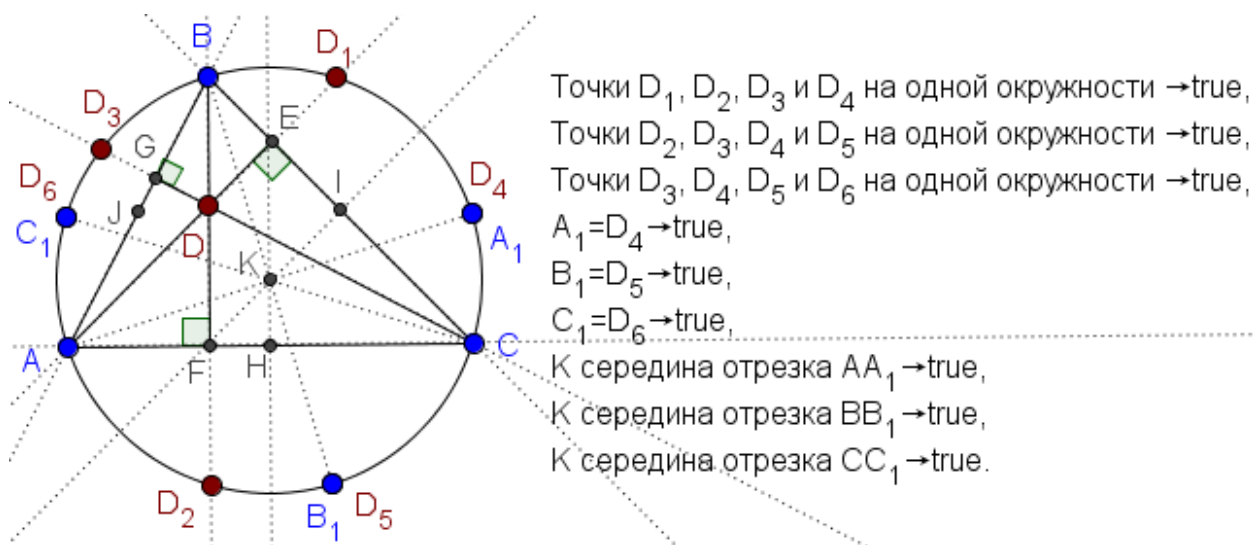







Рис. 4g. Модель для экспериментальной проверки утверждения примера 19

Пример 20. Построить модель для экспериментальной проверки утверждения: “У каждого из четырех треугольников с вершинами в ортоцентрической системе радиусы описанных окружностей равны. Кроме того, центры трех из этих окружностей лежат на окружности такого же радиуса с центром в ортоцентре исходного треугольника.”

Решение. Искомая модель, представленная на рис. 4h, создавалась следующим образом:

- инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментами “ Перпендикулярная прямая” проведем перпендикулярные линии из вершин треугольника к противоположным сторонам или их продолжениям. Инструментом “ Точка на объекте” сформируем ортоцентр D и точки пересечения этих линий со сторонами треугольника;
- инструментом “ Окружность по трем точкам” для каждого из четырех треугольников с вершинами в ортоцентрической системе проведем описанную окружность;

- инструментом “ Отрезок” соединим ортоцентр D с вершинами треугольника. Инструментом “ Срединный перпендикуляр” проведем перпендикулярные линии к полученным отрезкам DA , DB и DC , а также к сторонам AC и BC исходного треугольника. Инструментом “ Точка на объекте” сформируем центры H, I, J и K всех четырех окружностей;
- инструментом “ Окружность по трем точкам” через точки H, I и J проведем окружность и изменим стиль ее вывода на точечный;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 20:

$$\begin{aligned}
 HD=DI &\rightarrow \boxed{HD==DI} , \\
 HD=DJ &\rightarrow \boxed{HD==DJ} , \\
 HD=KC &\rightarrow \boxed{HD==KC} .
 \end{aligned}$$

Перемещая независимые точки A, B и C по плоскости мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

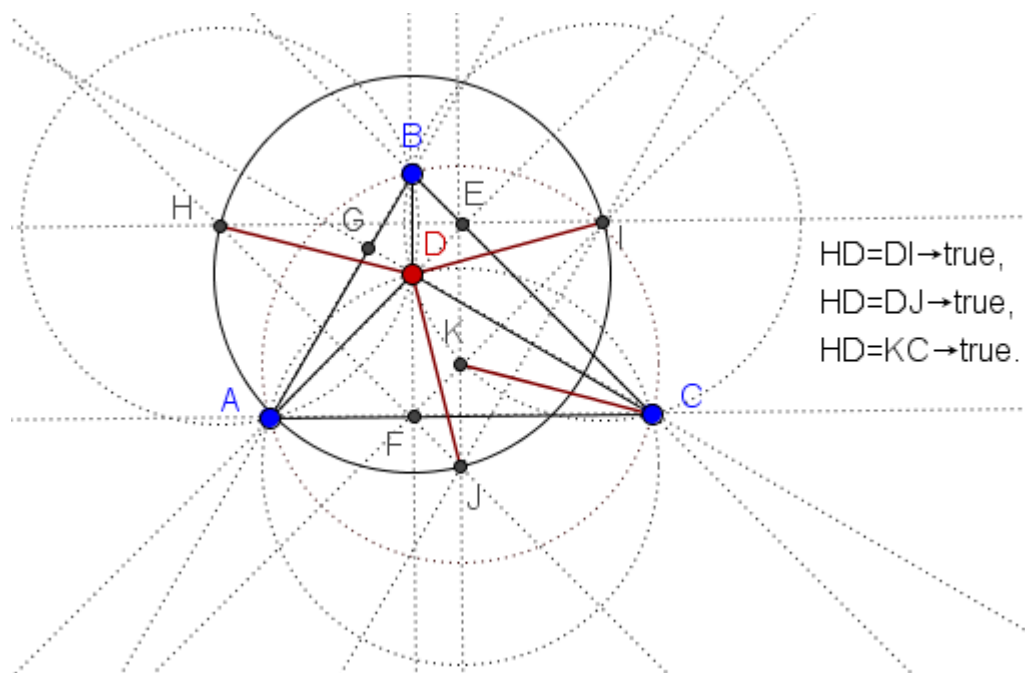












Рис. 4h. Модель для экспериментальной проверки утверждения примера 20

Пример 21. Построить модель для экспериментальной проверки утверждения (7), то есть формул (4) и (5) (см. выше)

Решение. Искомая модель, представленная на рис. 4i, создавалась следующим образом:

- инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$. Стороны треугольника против вершин A , B и C переименуем соответственно на a , b и c ;
- инструментами “ Перпендикулярная прямая” проведем перпендикулярные линии из вершин к противоположным сторонам $\triangle ABC$. Инструментом “ Точка на объекте” сформируем ортоцентр D и точки E , F и G пересечения этих линий со сторонами треугольника;
- инструментом “ Отрезок” проведем отрезки AD и DF , BD и DG , CD и DE . Через контекстное меню поменяем их имена соответственно на a_1 и a_2 , b_1 и b_2 , c_1 и c_2 ;
- инструментом “ Срединный перпендикуляр” проведем перпендикуляры к серединам сторон AC и BC . Инструментом “ Точка на объекте” сформируем точку H пересечения этих перпендикуляров, то есть центр описанной окружности. Инструментом “ Отрезок” проведем отрезок CH и изменим его имя на R . Инструментом “ Окружность по центру и точке” проведем описанную окружность (центр – H , точка – C) (необязательно);
- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 21:

$$a_1 \cdot a_2 = b_1 \cdot b_2 = c_1 \cdot c_2 \rightarrow$$

$$a_1 \cdot a_2 == b_1 \cdot b_2 == c_1 \cdot c_2 \quad ,$$

$$a_1^2 + b_1^2 + c_1^2 + a^2 + b^2 + c^2 = 12 \cdot R^2 \rightarrow$$

$$a_1^2 + b_1^2 + c_1^2 + a^2 + b^2 + c^2 == 12 \cdot R^2 \quad .$$

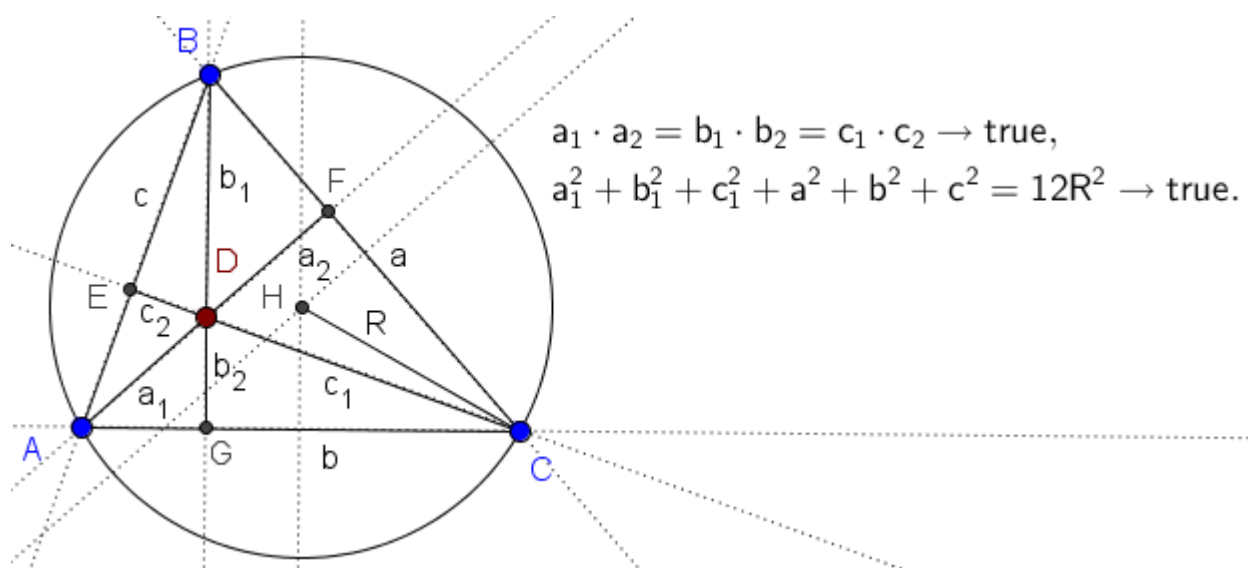


Рис. 4i. Модель для экспериментальной проверки утверждения примера 21

Перемещая независимые точки A, B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

Замечание. Соотношение (5) означает, что высоты треугольника ортоцентром делятся на части, произведения которых равны. То есть, высоты ведут себя как хорды некоторого круга, пересекающиеся в точке D . А это значит, что концевые точки любых двух высот лежат на одной окружности. Данный факт подтверждается моделью, представленной на рис. 4j. Проверочная надпись для нее формировалась так:

Точки A, B, G и F на одной окружности \rightarrow

$AreConcyclic[A, B, G, F]$

,

Точки A, C, F и E на одной окружности \rightarrow

$AreConcyclic[A, C, F, E]$

,

Точки B, C, G и E на одной окружности \rightarrow

$AreConcyclic[B, C, G, E]$

.

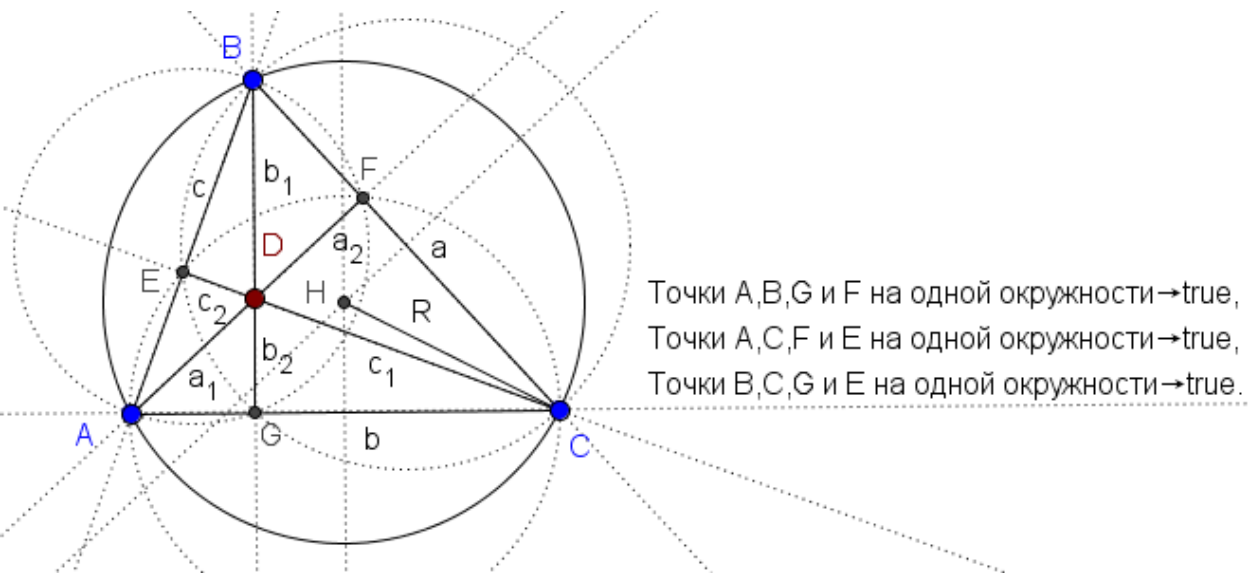


Рис. 4j. Дополнение к модели рис. 4i

5.6. Центр окружности 9 точек

Имеет место следующее утверждение: “Основания трёх высот треугольника, середины трёх его сторон и середины трёх отрезков, соединяющих вершины треугольника с его ортоцентром, лежат на одной окружности”. Эта окружность и называется окружностью девяти точек или окружностью Эйлера. В энциклопедии Кимберлинга центр окружности девяти точек именуется как $X(5)$, и для $\triangle ABC$ через строку ввода его можно строить командой `Triangle-Center[A,B,C,5]`.

Справедливы такие утверждения:




1) Центр окружности девяти точек лежит на одной прямой с ортоцентром и центром описанной окружности точно посередине между ними. Эта прямая называется прямой Эйлера.



2) Радиус окружности девяти точек равен половине радиуса описанной окружности.




3) Окружность девяти точек делит пополам любой отрезок, который соединяет ортоцентр с произвольной точкой, лежащей на описанной окружности.


4) *Теорема Фейербаха.* Окружность девяти точек произвольного треугольника касается вписанной и всех трёх внеписанных окружностей этого треугольника.


Построение окружности девяти точек. Изображение, представленное на рис. 5а, является результатом построения окружности, проходящей через основания высот треугольника, то есть окружности, описанной около ортотреугольника. Это изображение представляет собой динамическую модель для экспериментальной проверки того факта, что построена именно окружность девяти точек. Иными словами, на ней располагаются и середины сторон исходного треугольника, и середины отрезков, соединяющих вершины этого треугольника с ортоцентром. Создавалась модель следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Точка”, “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- инструментом “ Перпендикулярная прямая” через вершины к сторонам треугольника проведем перпендикулярные линии. Инструментом “ Точка на объекте” сформируем ортоцентр D и точки E , F и G пересечения проведенных линий со сторонами треугольника или их продолжениями. Изменим стиль вывода точек E , F и G ;

- инструментом “ Окружность по трем точкам” проведем окружность через точки E , F и G . Инструментом “ Срединный перпендикуляр” проведем перпендикулярные линии к воображаемым отрезкам GF и EF . Инструментом “ Точка на объекте” сформируем точку H пересечения этих линий, то есть центр окружности девяти точек, а также точки I , J и K пересечения этой окружности со сторонами $\triangle ABC$. Изменим стиль вывода точек I , J и K ;

- инструментом “ Точка на объекте” сформируем точки L , M и N – пересечения окружности с линиями, перпендикулярными к сторонам $\triangle ABC$. Изменим стиль вывода этих точек;

- инструментом “ Текст” сформируем единую проверочную надпись вводом нескольких фрагментов текста с вычисляемыми правыми частями в строках:

$$\begin{aligned} AJ=LC &\rightarrow \boxed{AJ==JC} \ , \\ AI=IB &\rightarrow \boxed{AI==IB} \ , \\ BK=KC &\rightarrow \boxed{BK==KC} \ , \end{aligned}$$

$$\begin{aligned}
 AM=MD &\rightarrow \boxed{AM==MD} \text{ ,} \\
 BL=LD &\rightarrow \boxed{BL==LD} \text{ ,} \\
 CN=ND &\rightarrow \boxed{CN==ND} \text{ ,} \\
 HF=HI=HJ=HK &\rightarrow \boxed{HF==HI==HJ==HK} \text{ ,} \\
 HF=HL=HM=HN &\rightarrow \boxed{HF==HL==HM==HN} \text{ .}
 \end{aligned}$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки проверочной надписи значение *true*, что и требовалось проверить. На рис. 5b показано состояние модели при конкретном смещении точки B .

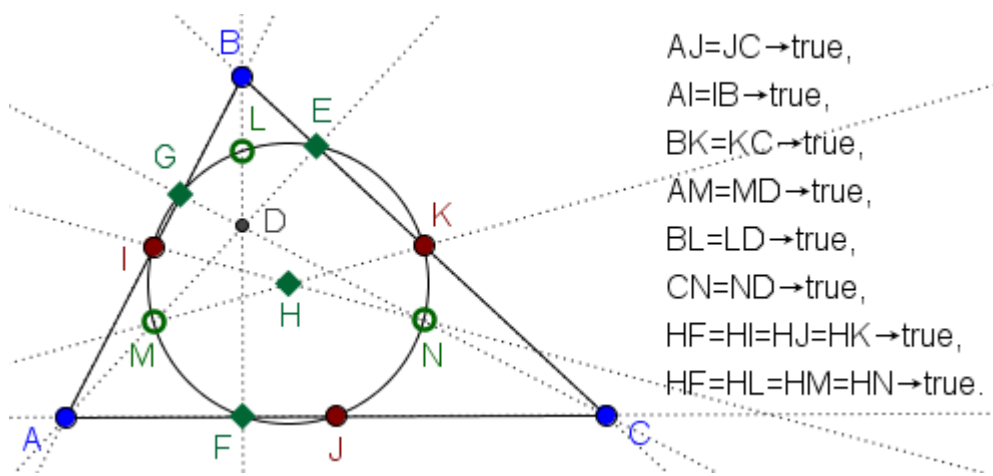


Рис. 5a. Модель для экспериментов с окружностью девяти точек

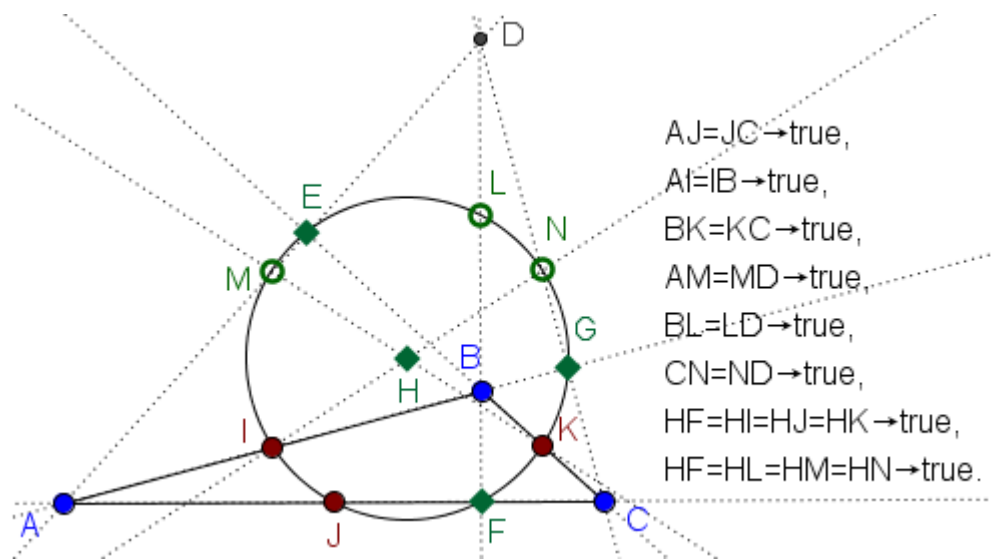







Рис. 5b. Модель для экспериментов с окружностью девяти точек

Пример 22. Построить модель для экспериментальной проверки утверждения: “Центр окружности девяти точек лежит на одной прямой с ортоцентром и центром описанной окружности точно посередине между ними”.

Решение. Искомая модель, представленная на рис. 5с, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 5а. Инструментом “ Окружность по трем точкам” проведем описанную окружность к $\triangle ABC$. Удалим проверочную надпись. Ее нам придется сформировать заново;
- инструментом “ Срединный перпендикуляр” проведем перпендикулярные линии к сторонам AC и BC . Инструментом “ Точка на объекте” сформируем точку пересечения O этих линий, то есть центр описанной окружности;
- инструментом “ Прямая” проведем прямую через точки D и O и изменим стиль ее вывода;
- инструментом “ Текст” сформируем проверочную надпись, демонстрирующую, что H лежит на построенной прямой и делит отрезок DO пополам:

$$H \text{ лежит на отрезке } DO \rightarrow \boxed{IsInRegion[H, Polygon[D,O]]} \text{ ,}$$

$$DH=HO \rightarrow \boxed{DH==HO} \text{ .}$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

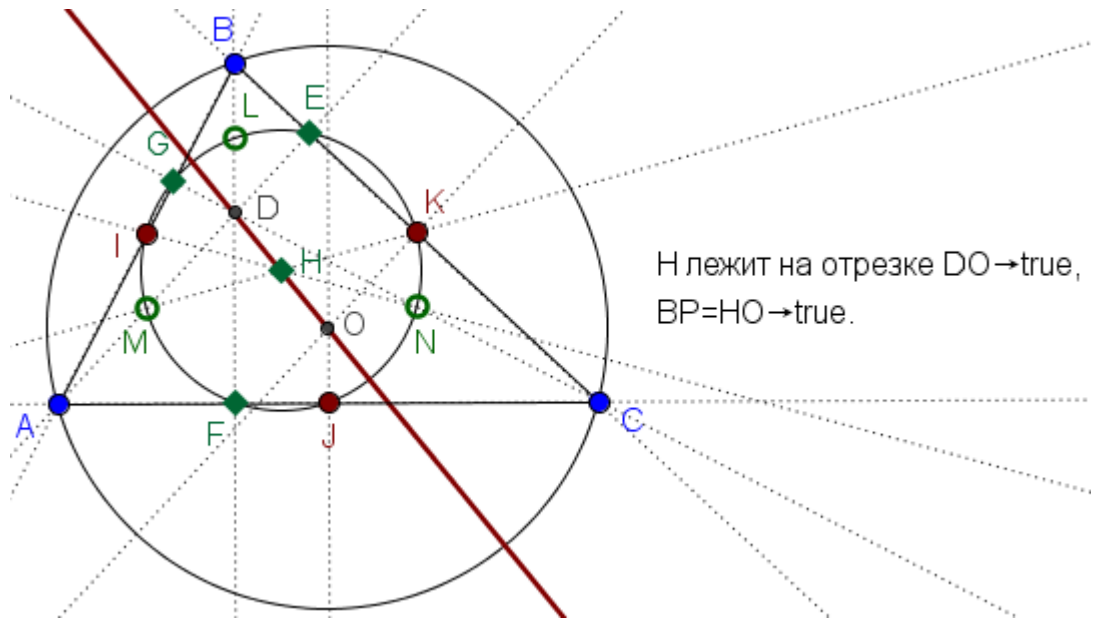


Рис. 5с. Модель для экспериментальной проверки утверждения примера 22


Пример 23. Построить модель для экспериментальной проверки утверждений:


1) Радиус окружности девяти точек равен половине радиуса описанной окружности;


2) Окружность девяти точек делит пополам любой отрезок, который соединяет ортоцентр с произвольной точкой, лежащей на описанной окружности.


Решение. Искомая модель, представленная на рис. 5d, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 5с. Через контекстное меню удалим ненужные нам объекты: проверочную надпись и прямую Эйлера;

- инструментом “ Точка на объекте” прикрепим в любом месте описанной окружности к $\triangle ABC$ точку P ;

- инструментом “ Отрезок” проведем отрезки: DP – между ортоцентром D и точкой P , HK – радиус окружности девяти точек и OC – радиус описанной окружности;

- инструментом “ Точка на объекте” сформируем точку Q – пересечения отрезка DP и окружности девяти точек;

- инструментом “ Текст” сформируем проверочную надпись для проверки сформулированных в примере утверждений:

$$OC=2HK \rightarrow \boxed{OC==2*HK} ,$$

$$DQ=QP \rightarrow \boxed{DQ=QP} .$$

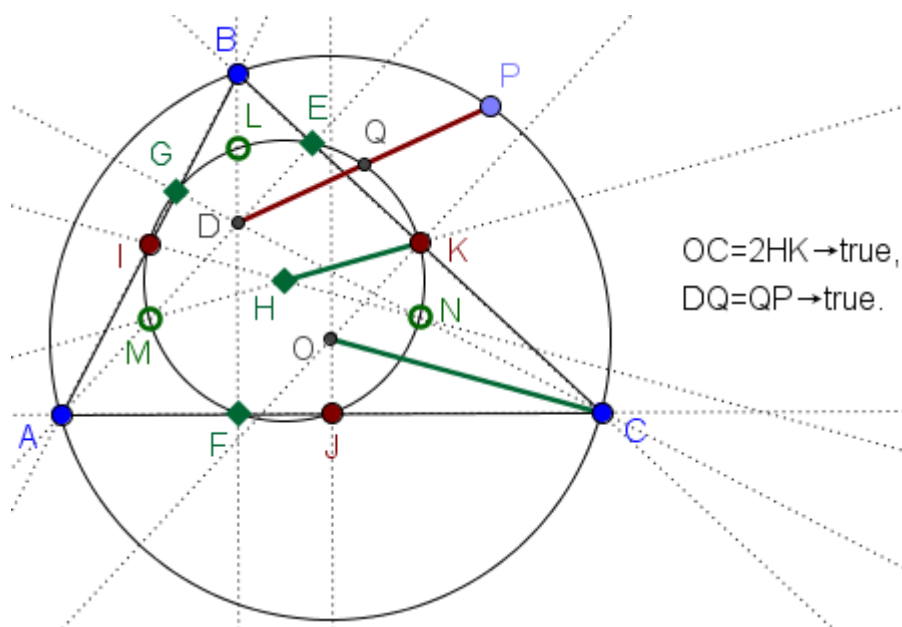


Рис. 5d. Модель для экспериментальной проверки утверждений примера 23


Перемещая на построенной модели независимые точки A , B и C по плоскости и прикрепленную точку P по описанной окружности, мы всегда будем



получать в правых частях каждой строки проверочной надписи значение *true*, что и требовалось проверить.



Пример 24. Построить модель для экспериментальной проверки теоремы Фейербаха “Окружность девяти точек произвольного треугольника касается вписанной и всех трёх вневписанных окружностей этого треугольника”.



Решение. Искомая модель, представленная на рис. 5е и 5f, создавалась следующим образом:


- продолжим построение модели, начиная с изображения, показанного на рис. 5с. Через контекстное меню удалим ненужные нам объекты – проверочную надпись, описанную окружность, прямую Эйлера, точки *O* и *D*, перпендикулярные линии к сторонам треугольника в точках *J* и *K*;

- через строку ввода введем команду `Checkbox[{E,F,G,I,J,K,L,M,N}]`, создающую независимый переключатель  (флажок), по которому можно показывать и прятать перечисленные в списке точки. На панели “Настройки” зададим заголовок для переключателя “*Ф/Все*”. Выключенный флажок будет демонстрировать на окружности девяти точек, только четыре точки, которые упоминаются в теореме Фейербаха, включенный – добавлять к ним еще и “свои” девять точек. Для того, чтобы перетащить переключатель в нужную позицию, его через контекстное меню следует открепить (`RClick(переключатель)/Fix Checkbox`);

- инструментом “ Биссектриса угла” проведем биссектрисы внутренних и внешних углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем центр *D* вписанной окружности и центры *O*, *P* и *Q* вневписанных окружностей как точки пересечения соответствующих биссектрис;

- инструментом “ Перпендикулярная прямая” из точек *D*, *P*, *Q* и *O* проведем перпендикулярные линии соответственно к сторонам *BC*, *BC*, *AC* и *AB* треугольника или их продолжениям. Инструментом “ Точка на объекте” сформируем точки *R*, *S*, *T* и *U* – пересечения проведенных линий со сторонами треугольника или их продолжениями;

- инструментом “ Окружность по центру и точке” проведем вписанную и три вневписанных окружности; инструментом “ Точка на объекте” сформируем точки *V*, *W*, *Z* и *A*₁ – пересечения окружности девяти точек с вписанной окружностью и тремя вневписанными окружностями;

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 24 (*k* – окружность девяти точек, *e*₁, *f*₁, *g*₁ и *h*₁ – вписанная и вневписанные окружности):

$$\begin{aligned} \text{Пересечение } k \text{ с } e_1 &\rightarrow \boxed{V==Intersect[k, e_1]} , \\ \text{Пересечение } k \text{ с } f_1 &\rightarrow \boxed{Z==Intersect[k, f_1]} , \\ \text{Пересечение } k \text{ с } g_1 &\rightarrow \boxed{W==Intersect[k, g_1]} , \end{aligned}$$

Пересечение k с $h_1 \rightarrow \overline{\overline{A_1 == Intersect[k, h_1]}}$.

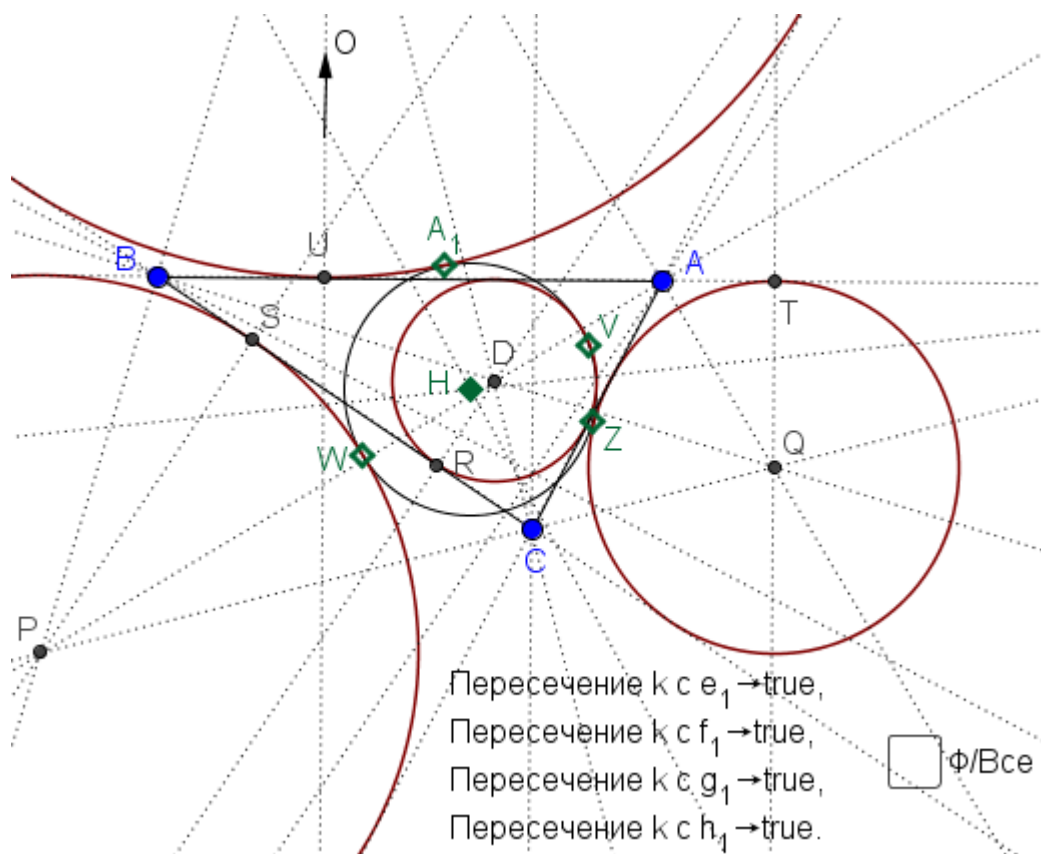


Рис. 5e. Модель для экспериментальной проверки утверждения примера 24

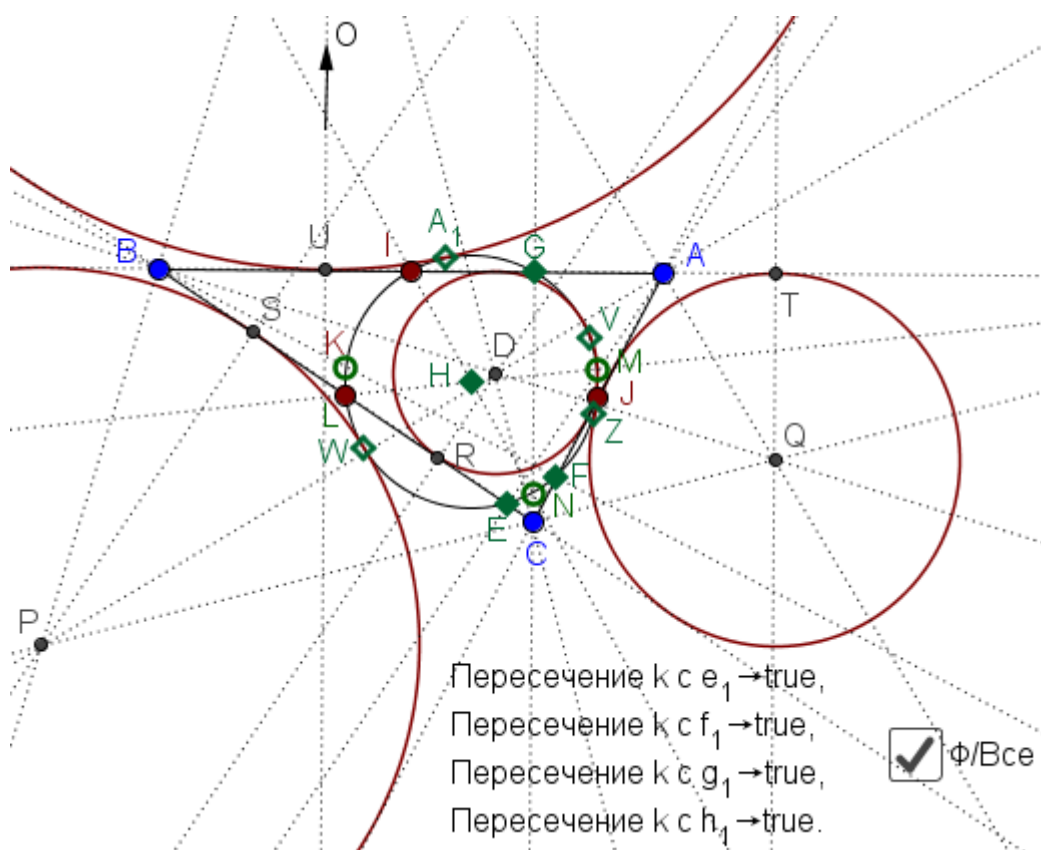









Рис. 5f. Модель для экспериментальной проверки утверждения примера 24

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки проверочной надписи значение *true*, что и требовалось проверить. Рис. 5*f* получен из рис. 5*e* при включенном флажке.

Пример 25. Построить модель для экспериментальной проверки теоремы Мавло: “Окружность девяти точек $\triangle ABC$ отсекает от него внешним образом три дуги так, что длина наибольшей из них равна сумме длин двух других дуг”.

Решение. Если e, f и g – дуги (и их длины), то фактически надо показать, что $h+k+p=2\max(h,k,p)$. Искомая модель, представленная на рис. 5*g*, создавалась следующим образом:

- инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$. Командой `TriangleCenter[A,B,C,5]` сформируем центр D окружности девяти точек $\triangle ABC$;
- инструментом “ Середина или центр” создадим точку E , являющуюся серединой стороны BC . Инструментом “ Окружность по центру и точке” проведем окружность 9 точек (центр – D , точка – E);
- инструментом “ Точка на объекте” сформируем точки F, G, H, I и J – пересечения окружности со сторонами треугольника. Инструментом “ Дуга по центру и двум точкам” проведем дуги с центром в D между парами точек E и F, G и H, I и J . Они получают имена f, e и g соответственно;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 25:

$$e+f+g=2\text{Max}[\{e,f,g\}] \rightarrow \boxed{e+f+g==2*\text{Max}[\{e,f,g\}]} .$$

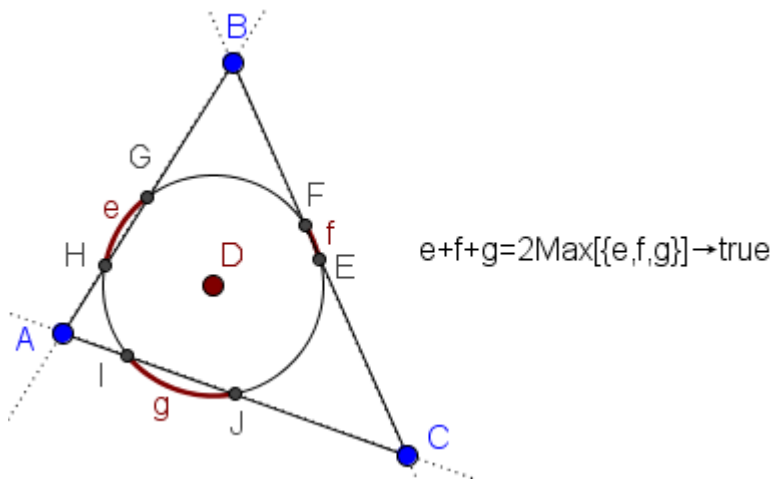








Рис. 5*g*. Модель для экспериментальной проверки утверждения примера 25

Перемещая независимые точки A , B и C по плоскости мы всегда будем получать в правой части надписи значение *true*, что и требовалось проверить.

Пример 26. Построить модель для экспериментальной проверки теоремы Гамильтона: “Отрезки, соединяющие ортоцентр с вершинами остроугольного $\triangle ABC$, разбивают его на три треугольника, имеющих ту же самую окружность девяти точек, что и исходный $\triangle ABC$. Эти треугольники называют треугольниками Гамильтона”.

Решение. Искомая модель, представленная на рис. 5h, создавалась следующим образом:

- инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командой $TriangleCenter[A,B,C,4]$ сформируем ортоцентр D $\triangle ABC$. Инструментом “ Отрезок” соединим ортоцентр с вершинами треугольника;
- командами $TriangleCenter[X,Y,Z,5]$ выведем центры E, F, G и H окружностей девяти точек для треугольников ABC, ABD, BCD и CAD (наша цель – проверить, что они совпадают);
- инструментом “ Середина или центр” создадим точки I, J, K, L, M и N , являющиеся серединами сторон треугольников ABC, ABD, BCD и CAD ;
- инструментом “ Окружность по центру и точке” для $\triangle ABC$ проведем окружность девяти точек (центр – E , точка – J). Эта окружность пройдет и через точки I и K (наша цель – проверить, что точки L, M и N также лежат на этой окружности);
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 26:

$$E=F=G=H \rightarrow \boxed{E==F==G==H} ,$$

$$L \text{ на окружности } (IJK) \rightarrow \boxed{AreConcyclic[I, J, K, L]} ,$$

$$N \text{ на окружности } (IJK) \rightarrow \boxed{AreConcyclic[I, J, K, N]} ,$$

$$M \text{ на окружности } (IJK) \rightarrow \boxed{AreConcyclic[I, J, K, M]} .$$

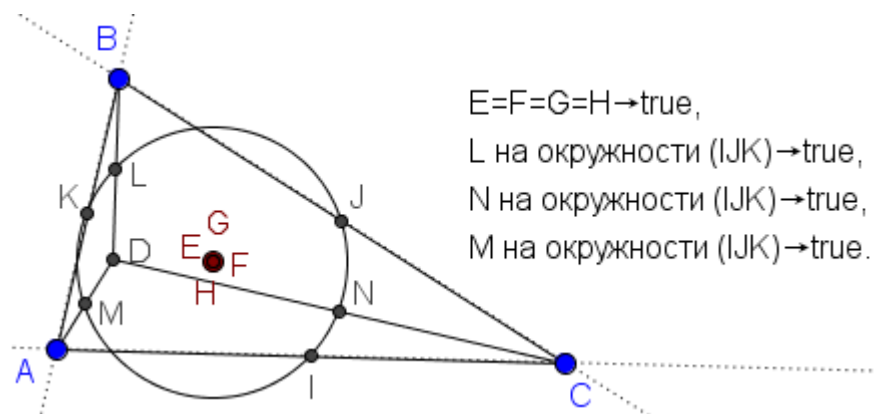


Рис. 5h. Модель для экспериментальной проверки утверждения примера 26

Перемещая независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

Замечания. 1. Если $\triangle ABC$ – прямоугольный, то ортоцентр совпадает с вершиной прямого угла, два треугольника Гамильтона вырождаются в катеты, а третий треугольник Гамильтона совпадает с исходным $\triangle ABC$. Формально теорема Гамильтона справедлива и в этом предельном случае.

2) *Аналог теоремы Гамильтона для тупоугольного $\triangle ABC$.* Пусть в тупоугольном $\triangle ABC$: A , B – вершины острых углов, C – вершина тупого угла и O – ортоцентр. Тогда $\triangle ABO$ является остроугольным и C – его ортоцентр (точки C и O поменялись местами). Таким образом $\triangle ABO$ разбит на треугольники Гамильтона ABC , BCO и ACO , и к $\triangle ABO$ можно применять теорему Гамильтона. Отметим, что модель, представленную на рис. 5h, можно использовать и для экспериментов с тупоугольными треугольниками.

5.7. Точка Лемуана

Начнем с определений.

Чевианой треугольника называют луч из его вершины, проходящий через точку на противоположной стороне или на её продолжении.

Симедианой треугольника называют его чевиану, исходящую из некоторой вершины, симметрично лучу медианы относительно луча биссектрисы, проведенных из этой же вершины (биссектриса разделяет медиану и симедиану).

Тангенциальным треугольником по отношению к непрямоугольному $\triangle ABC$ называют треугольник, стороны которого в точках A , B и C касаются описанной окружности исходного треугольника.

Педальным или *подёрным* треугольником произвольной точки E относительно исходного $\triangle ABC$ – называется треугольник, вершинами которого являются основания перпендикуляров, опущенных из точки E на стороны $\triangle ABC$ или их продолжения.

Известно, что: “Симедианы треугольника пересекаются в одной точке”. Эта точка называется точкой Лемуана. В энциклопедии Кимберлинга она именуется как $X(6)$ и через строку ввода для $\triangle ABC$ ее можно строить командой `TriangleCenter[A,B,C,6]`.

Справедливы такие утверждения:

1) Для непрямоугольного треугольника симедианы проходят через вершины тангенциального треугольника.



2) Отрезки, на которые симедиана делит противоположную сторону, **пропорциональны квадратам** прилежащих сторон;


3) Расстояния от точки Лемуана до **сторон** треугольника пропорциональны длинам сторон.



4) Сумма квадратов расстояний от точки Лемуана до сторон треугольника меньше, чем эта сумма от любой другой точки.


5) Точка Лемуана исходного треугольника является **центроидом** своего **педального треугольника** (и это единственная такая точка в треугольнике).



Построение точки Лемуана. Изображение, представленное на рис. 6а, является результатом построения точки Лемуана, исходя из ее определения как точки пересечения семидиан треугольника. Это изображение является динамической моделью со свободными точками A , B и C . Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- инструментом “ Биссектриса угла” проведем биссектрисы внутренних углов $\triangle ABC$;

- инструментом “ Середина или центр” создадим точки D , E и F , являющиеся соответственно серединами сторон AB , BC и AC $\triangle ABC$. Инструментом “ Прямая” проведем прямые через пары точек A и E , B и F , C и D , являющиеся концами медиан треугольника (необязательно);

- инструментом “ Отражение относительно прямой” построим точки D' , E' и F' , симметричные относительно соответствующих биссектрис точкам D , E и F . Изменим стиль вывода построенных точек, лежащих на семидианах;

- инструментом “ Луч” из вершин треугольника через точки D' , E' и F' проведем три семидианы и изменим стиль их вывода. Инструментом “ Точка на объекте” сформируем точку Лемуана G как точку пересечения семидиан.

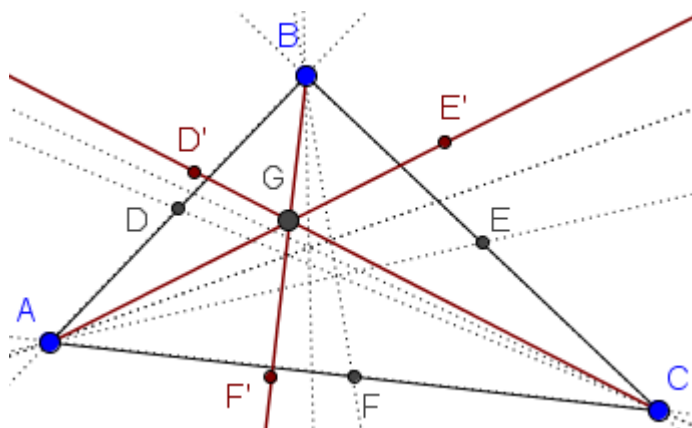











Рис. 6а. Модель для экспериментов с точкой Лемуана

Пример 27. Построить модель для экспериментальной проверки утверждения: “Для непрямоугольного треугольника седмианы проходят через вершины тангенциального треугольника”.

Решение. Искомая модель, представленная на рис. 5b, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Окружность по трем точкам” проведем описанную окружность к $\triangle ABC$. Инструментом “ Касательная” сформируем касательные к окружности в вершинах A, B и C исходного треугольника. Инструментом “ Точка на объекте” создадим вершины D, E , и F тангенциального треугольника как точки пересечения проведенных касательных. Для непрямоугольного $\triangle ABC$ таких точек действительно будет три;
- инструментом “ Отрезок” проведем стороны тангенциального треугольника (необязательно). Инструментом “ Прямая” сформируем прямые, проходящие через противоположные вершины исходного и тангенциального треугольников и изменим стиль их вывода. Инструментом “ Точка на объекте” создадим точку G пересечения проведенных прямых;
- командой `TriangleCenter[A,B,C,6]` создадим точку Лемуана H исходного $\triangle ABC$;
- инструментом “ Текст” сформируем проверочную надпись в виде:

$$H=G \rightarrow \boxed{H==G} .$$

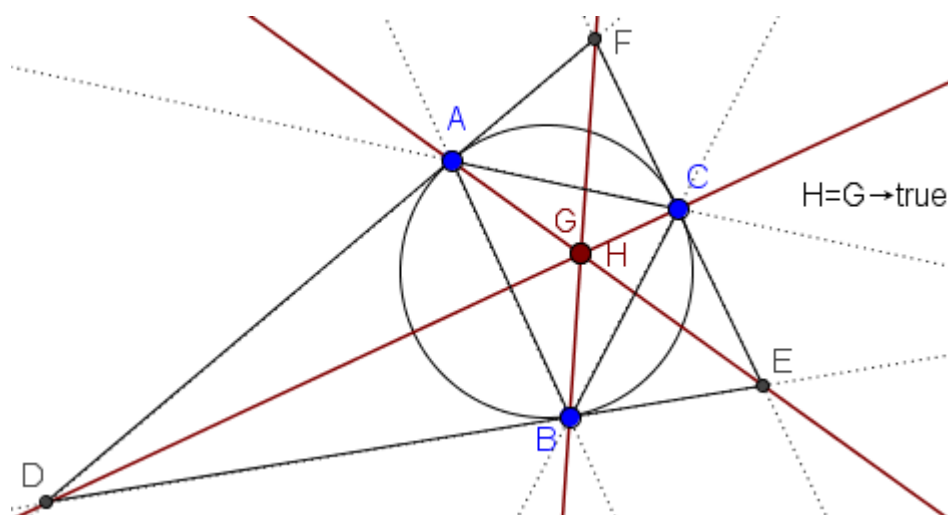


Рис. 6b. Модель для экспериментальной проверки утверждения примера 27

Перемещая на построенной модели независимые точки A, B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значе-

ние *true*, то есть экспериментальное подтверждение того, что *G* является точкой Лемуана, а вместе с тем и экспериментальное подтверждение прохождения семидиан через вершины тангенциального треугольника.

Пример 28. Построить модель для экспериментальной проверки утверждений:

- 1) Отрезки, на которые симедиана делит противоположную сторону, пропорциональны квадратам прилежащих сторон;
- 2) Расстояния от точки Лемуана до сторон треугольника пропорциональны длинам сторон.

Решение. Искомая модель, представленная на рис. 5с, создавалась следующим образом:

- инструментами “Прямая” и “Отрезок” построим $\triangle ABC$ и командой `TriangleCenter[A,B,C,6]` создадим точку Лемуана *D*;
- инструментом “Прямая” из вершин треугольника через точку *D* проведем три прямые, содержащие семидианы и изменим стиль их вывода;
- инструментом “Точка на объекте” создадим точки *E*, *F* и *G* – пересечения семидиан со сторонами $\triangle ABC$;
- инструментом “Перпендикулярная прямая” из точки *D* проведем перпендикулярные линии к сторонам треугольника. Инструментом “Точка на объекте” сформируем точки *I*, *J*, *H* пересечения проведенных линий со сторонами треугольника. Инструментом “Отрезок” проведем перпендикуляры из точки *D* к сторонам треугольника (необязательно);

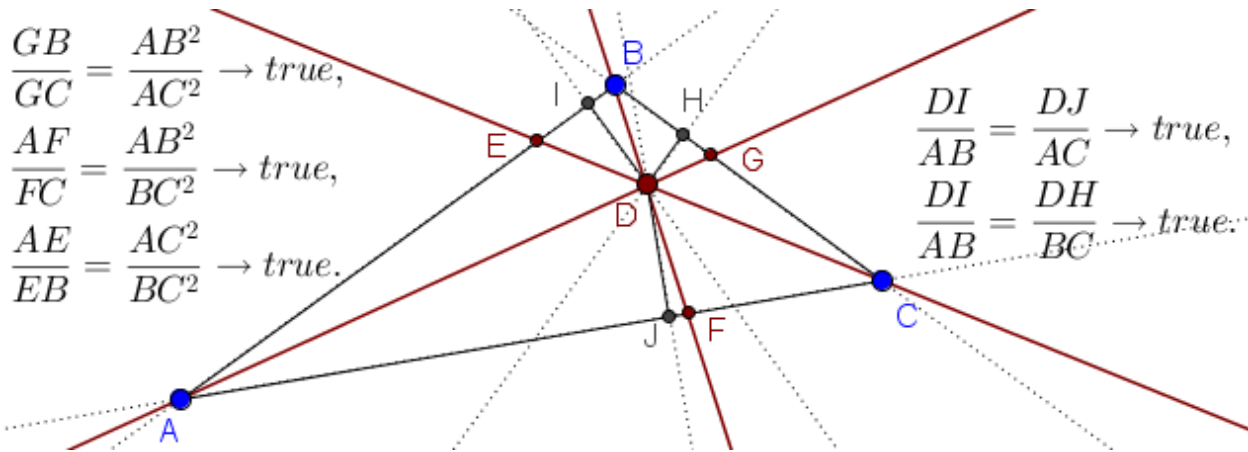


Рис. 6с. Модель для экспериментальной проверки утверждений примера 28

- инструментом “Текст” в *LaTeX*-режиме сформируем две проверочные надписи для проверки каждого из утверждений примера 28:

$$\frac{GB}{GC}=\frac{AB^2}{AC^2}\rightarrow \boxed{\frac{GB}{GC}==AB^2/AC^2} \, ,$$

$$\frac{AF}{FC}=\frac{AB^2}{BC^2}\rightarrow \boxed{\frac{AF}{FC}==AB^2/BC^2} \, ,$$

$$\frac{AE}{EB} = \frac{AC^2}{BC^2} \rightarrow \boxed{AE/EB == AC^2/BC^2} . ;$$

и

$$\frac{DI}{AB} = \frac{DJ}{AC} \rightarrow \boxed{|DI|/|AB| == |DJ|/|AC|} ,$$




$$\frac{DI}{AB} = \frac{DH}{BC} \rightarrow \boxed{|DI|/|AB| == |DH|/|BC|} .$$


Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки обеих надписей значение *true*, что и требовалось проверить.

Пример 29. Построить модель для экспериментальной проверки утверждения: “Сумма квадратов расстояний от точки Лемуана до сторон треугольника меньше, чем эта сумма от любой другой точки”.

Решение. Искомая модель, представленная на рис. 6d, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 6с. Через контекстное меню удалим ненужные нам объекты – проверочные надписи и прямые CE , BF и AG , через которые проходят симедианы. Точка их пересечения D , то есть точка Лемуана, останется на модели;

- инструментом “ Точка на объекте” сформируем где-либо на плоскости точку E . Инструментом “ Перпендикулярная прямая” из точки E проведем перпендикулярные линии к сторонам треугольника. Инструментом “ Точка на объекте” отметим точки F , G и K – пересечения построенных линий со сторонами или продолжениями сторон $\triangle ABC$. Изменим стиль вывода точек F , G и K ;

- инструментом “ Отрезок” построим перпендикуляры EF , EG и EK из E на стороны или продолжения сторон треугольника ABC (необязательно);

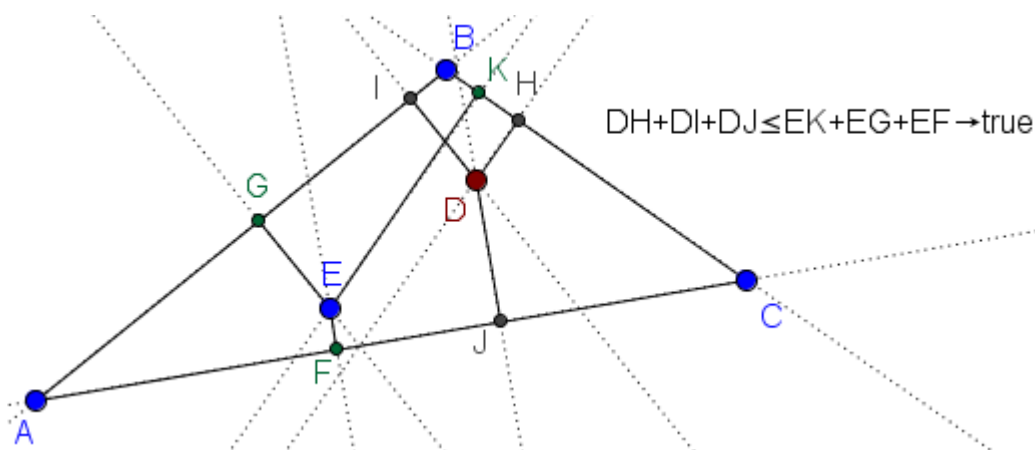



Рис. 6d. Модель для экспериментальной проверки утверждений примера 29

- инструментом “ Текст” сформируем проверочную надпись для проверки утверждения примера 29:




$$DH+DI+DJ \leq EF+EK+EG \rightarrow \boxed{DH+DI+DJ \leq EF+EK+EG} .$$

Перемещая на построенной модели независимые точки A , B , C и E по плоскости, мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

Пример 30. Построить модель для экспериментальной проверки утверждения: “Точка Лемуана исходного треугольника является **центроидом** своего **педального треугольника**”.

Решение. Искомая модель, представленная на рис. 6е, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 6д. Через контекстное меню удалим ненужные нам объекты – проверочную надпись и точку E . Связанные с ней точки K , F , G и отрезки EK , EF , EG также будут удалены;

- инструментом “ Перпендикулярная прямая” из точки Лемуана D проведем перпендикулярные линии к сторонам $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точки I , J и H пересечения этих линий со сторонами $\triangle ABC$. Инструментом “ Отрезок” построим отрезки IJ , IH и HJ , то есть стороны педального треугольника для точки Лемуана D исходного $\triangle ABC$ и изменим стиль их вывода;

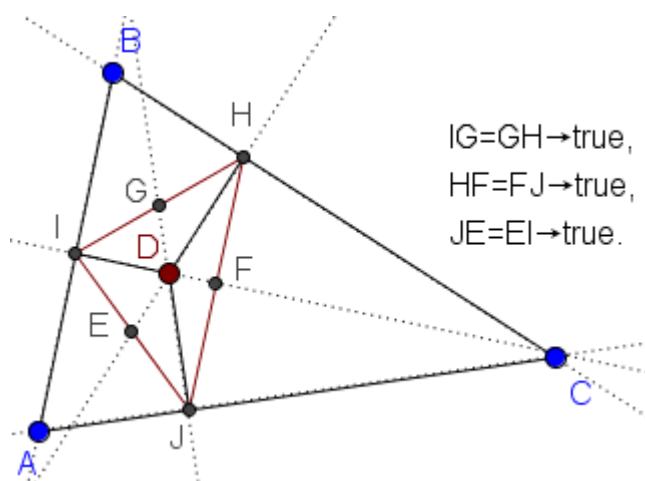



Рис. 6е. Модель для экспериментальной проверки утверждений примера 30

- если точки E , F и G – являются серединами сторон педального треугольника, то D – ортоцентр этого треугольника. Поэтому инструментом “ Текст” проверочную надпись можно сформировать так:








$$\begin{aligned} IG=GH &\rightarrow \boxed{IG==GH} , \\ HF=FJ &\rightarrow \boxed{HF==FJ} , \end{aligned}$$

$$JE=EI \rightarrow \boxed{JE==EI}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

Пример 31. Построить модель для экспериментальной проверки утверждения: “Точка Лемуана совпадает с точкой пересечения прямых, соединяющих середины сторон треугольника с серединами соответствующих им высот”.

Решение. Данное утверждение дает достаточно простой способ построения точки пересечения семидиан, то есть точки Лемуана. Искомая модель, представленная на рис. 5f, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Перпендикулярная прямая” из вершин $\triangle ABC$ проведем перпендикулярные линии к его сторонам. Инструментом “ Точка на объекте” сформируем точки D , E и F пересечения этих линий со сторонами треугольника;
- инструментом “ Середина или центр” создадим точки I , H и G , являющиеся соответственно серединами сторон AB , BC и AC $\triangle ABC$, а также точки J , K и L , являющиеся серединами высот этого треугольника;
- инструментом “ Прямая” проведем прямые, соединяющие середины сторон $\triangle ABC$ с серединами соответствующих им высот, то есть прямые, проходящие через пары точек G и K , I и J , H и L ;
- инструментом “ Точка на объекте” сформируем точку M пересечения проведенных линий. Командой `TriangleCenter[A,B,C,6]` выведем точку Лемуана N . Остается проверить, что $N=M$;

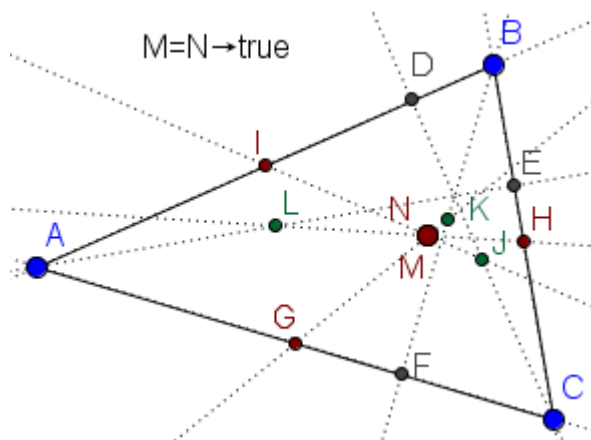


Рис. 6f. Модель для экспериментальной проверки утверждений примера 31

- инструментом “ABC Текст” сформируем проверочную надпись для проверки утверждения примера 31:

$$M=N \rightarrow \boxed{M==N}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

5.8. Точка Жергонна

Точка Жергонна (Gergonne point) – это точка пересечения отрезков, соединяющих вершины треугольника с точками касания противоположных сторон вписанной окружностью. В энциклопедии Кимберлинга она именуется как $X(7)$, и для $\triangle ABC$ через строку ввода ее можно строить командой *TriangleCenter*[$A,B,C,7$].

Определение. Треугольник с вершинами в точках касания противоположных сторон исходного $\triangle ABC$ вписанной окружностью называется треугольником Жергонна.

Справедливы такие утверждения:

1) Точка Жергонна является **точкой Лемуана** треугольника Жергонна, то есть треугольника с вершинами в точках касания сторон треугольника со вписанной окружностью.

2) Пусть в $\triangle ABC$: R – радиус описанной окружности, r – радиус вписанной окружности, p – полупериметр. Пусть S_1 и S_2 – расстояния от точки Жергонна соответственно до инцентра и центра описанной окружности. Имеют место следующие формулы квадратов расстояний:

$$S_1^2 = r^2 - \frac{3p^2r^2}{(4R+r)^2} \quad \text{– до инцентра,} \quad (6)$$

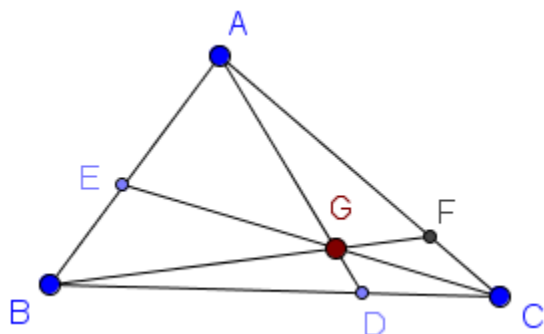
$$S_2^2 = R^2 - \frac{4p^2r(R+r)}{(4R+r)^2} \quad \text{– до центра описанной окружности.} \quad (7)$$

3) *Обобщение утверждения 2.* Пусть в $\triangle ABC$: a , b и c – длины сторон треугольника; $p=(a+b+c)/2$ – полупериметр; r и R – радиусы вписанной и описанной окружностей; J – точка Жергонна $\triangle ABC$; D – произвольная точка плоскости; d_1 , d_2 и d_3 – расстояния от вершин A , B и C до точки D ; d – расстояние между точкой Жергонна J и точкой D . Тогда

$$d^2 = \frac{d_1^2(p-b)(p-c) + d_2^2(p-a)(p-c) + d_3^2(p-b)(p-a)}{r(4R+r)} + \frac{4p^2r(R+r)}{(4R+r)^2}. \quad (8)$$

Поскольку D – произвольная точка плоскости, то по (8) можно находить расстояние между точкой Жергонна и любым другим центром треугольника.

4) *Теорема Жергонна*. Пусть три чевианы, проведенные к различным сторонам треугольника пересекаются в одной точке G внутри треугольника ABC . Тогда имеют место соотношения a и b , представленные на рис. 7а справа. В частности, эти соотношения справедливы и в том случае, когда G – это точка Жергонна.



$$a) \frac{GD}{AD} + \frac{GF}{BF} + \frac{GE}{CE} = 1,$$

$$b) \frac{GA}{AD} + \frac{GB}{CE} + \frac{GC}{BF} = 2.$$

Рис. 7а. Иллюстрация к теореме Жергонна

Построение точки Жергонна. Изображение, представленное на рис. 7b, является результатом построения точки Жергонна, исходя из ее определения. Это изображение является динамической моделью со свободными точками A , B и C . Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Биссектриса угла” проведем биссектрисы внутренних углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точку D – пересечения биссектрис;
- инструментом “ Перпендикулярная прямая” из точки D проведем перпендикулярные линии к сторонам треугольника. Инструментом “ Точка на объекте” сформируем точки E, F и G – пересечения перпендикулярных линий со сторонами треугольника;

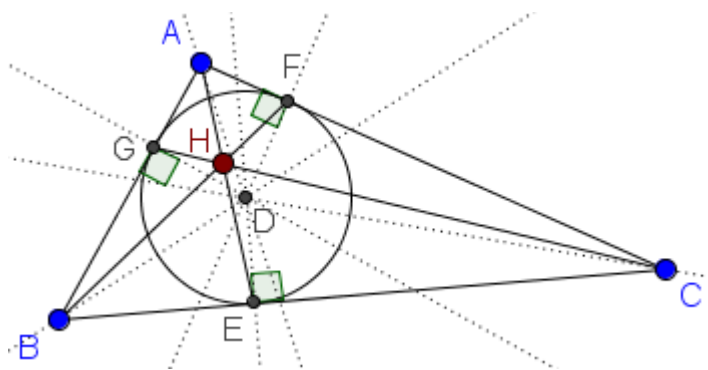







Рис. 7b. Модель для экспериментов с точкой Жергонна

- (необязательно) инструментом “ Угол” в точках E , F и G создадим метки прямых углов. Инструментом “ Окружность по центру и точке” проведем вписанную окружность для $\triangle ABC$;
- инструментом “ Отрезок” проведем отрезки AE , BF и CG . Их пересечение и дает нам точку Жергонна H .

Пример 32. Построить модель для экспериментальной проверки утверждения: “Точка Жергонна $\triangle ABC$ является **точкой Лемуана** треугольника Жергонна EFG ”.

Решение. Искомая модель, представленная на рис. 7с, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, показанного на рис. 7b. Через контекстное меню удалим ненужные нам метки прямых углов и спрячем обозначение D центра вписанной окружности. Инструментом “ Отрезок” построим треугольник Жергонна EFG (необязательно);
- командой $L=TriangleCenter[E,F,G,6]$ сформируем точку Лемуана L для $\triangle EFG$;
- инструментом “ Текст” сформируем проверочную надпись для проверки утверждения примера 32 (H – точка Жергонна $\triangle ABC$, L – точка Лемуана $\triangle EFG$):

$$H=L \rightarrow \boxed{H=L}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

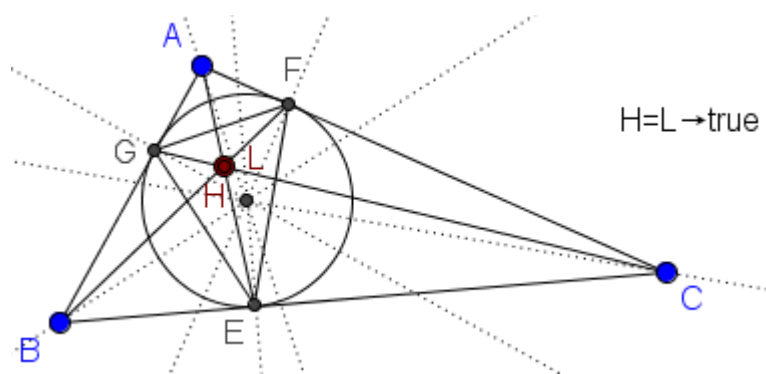






Рис. 7с. Модель для экспериментальной проверки утверждений примера 32

Пример 33. Построить модель для экспериментальной проверки формул (6) и (7) (см. выше утверждение 2).

Решение. Искомая модель, представленная на рис. 7d, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, показанного на рис. 7b. Инструментом “ Срединный перпендикуляр” проведем прямые, перпендикулярные сторонам AC и BC $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точку I – пересечения этих перпендикуляров. Инструментом “ Окружность по центру и точке” проведем описанную окружность для $\triangle ABC$ (необязательно);
- через строку ввода вычислим вспомогательную величину $t=((AB^2+BC^2+CA^2)/2)^{1/2}$;
- инструментом “ Текст” в режиме *LaTeX* сформируем надпись для проверки утверждений примера 33:

$$S_1^2 = r^2 - \frac{3p^2 r^2}{(4R+r)^2} \rightarrow$$

$$\boxed{DH^2 = DF^2 - 3 \cdot t \cdot DF^2 / (4 \cdot IC + DF)^2},$$

$$S_2^2 = R^2 - \frac{4p^2 r(R+r)}{(4R+r)^2} \rightarrow$$

$$\boxed{IH^2 = IC^2 - 4 \cdot t \cdot DF \cdot (IC + DF) / (4 \cdot IC + DF)^2}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

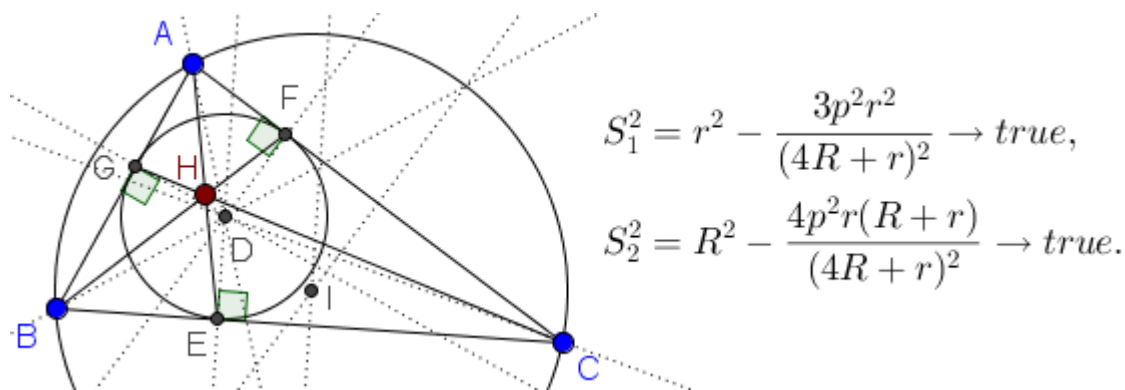












Рис. 7d. Модель для экспериментальной проверки утверждений примера 33

Пример 34. Построить модель для экспериментальной проверки формулы (8) (см. выше утверждение 3; <http://otkrytie.edu.yar.ru/discover/99/s6/1b.html>).

Решение. Искомая модель, представленная на рис. 7e, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$. Через контекстное меню изменим имена сторон треугольника на a , b и c ;
- командами $J=TriangleCenter[A,B,C,7]$, $F=TriangleCenter[A,B,C,1]$ и $G=TriangleCenter[A,B,C,3]$ создадим соответственно точку Жергонна J , инцентр F и центр описанной окружности для $\triangle ABC$;

- инструментом “ Точка” сформируем произвольную точку D на плоскости. Инструментом “ Отрезок” соединим D отрезками с вершинами A , B и C треугольника, а также с точкой Жергонна J . Изменим имена полученных отрезков соответственно на d_1 , d_2 , d_3 и d , а также стиль вывода d ;
- инструментом “ Перпендикулярная прямая” из точки F проведем линию, перпендикулярную к стороне AB . Инструментом “ Точка на объекте” сформируем точку E пересечения этой линии с AB . Инструментом “ Угол” поставим метку прямого угла в точке E (необязательно);
- инструментом “ Окружность по центру и точке” проведем вписанную окружность (центр – F , точка – E) и описанную окружность (центр – G , точка – A) для $\triangle ABC$. Изменим стиль вывода окружностей;
- инструментом “ Отрезок” проведем радиусы FH вписанной и GA описанной окружностей. Изменим стиль вывода радиусов и их имена соответственно на r и R ;
- через строку ввода сформируем вспомогательную переменную p , равную полупериметру $\triangle ABC$: $p=(a+b+c)/2$;
- инструментом “ Текст” в режиме *LaTeX* сформируем надпись для проверки утверждений примера 34:

$$d^2=\frac{d_1^2(p-b)(p-c)+d_2^2(p-a)(p-c)+d_3^2(p-b)(p-a)}{r(4R+r)}-\frac{4p^2r(R+r)}{(4R+r)^2}\rightarrow$$

$$d^2==(d_1^2*(p-b)*(p-c)+d_2^2*(p-a)*(p-c)+d_3^2*(p-b)*(p-a))/(r*(4R+r))-4*p^2*r*(R+r)/(4*R+r)^2$$

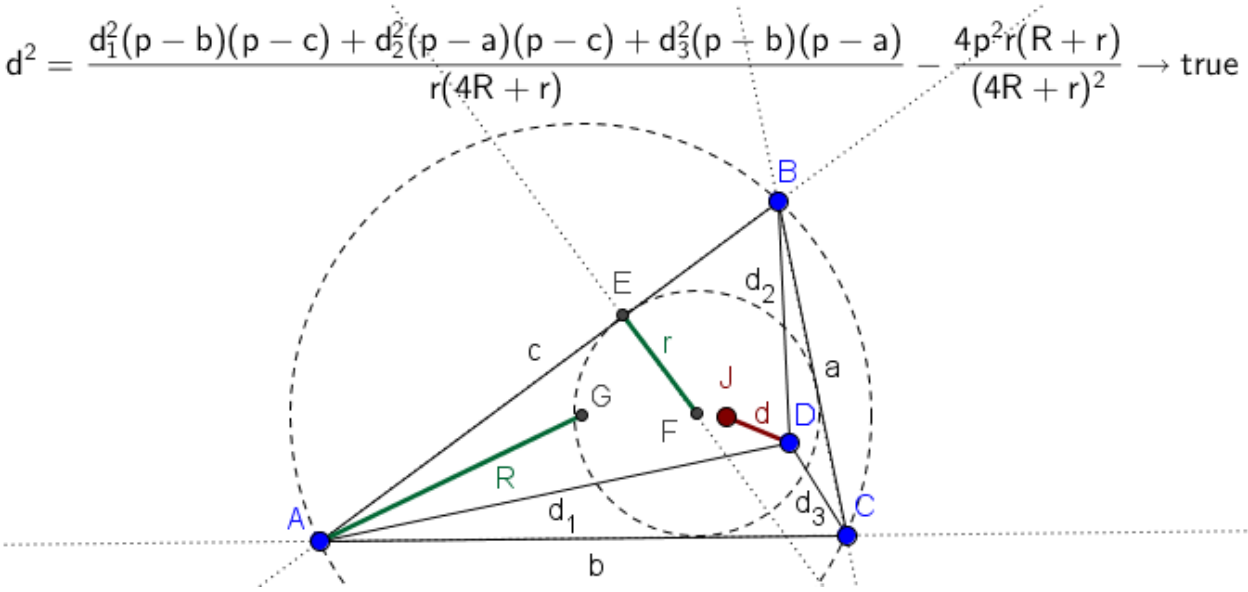



Рис. 7е. Модель для экспериментальной проверки утверждений примера 34

Перемещая на построенной модели независимые точки A , B , C и D по плоскости, мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

Пример 35. Проверим справедливость теоремы Жергонна (см. выше) в случае, если чевианы проходят через точку Жергонна.

Решение. Искомая модель, представленная на рис. 7f, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 7b. Через контекстное меню спрячем обозначение D – центра вписанной окружности;
- инструментом “ Текст” сформируем в *LaTeX*-режиме надписи для проверки теоремы Жергонна в виде:

$$\frac{HE}{AE}+\frac{HF}{BF}+\frac{HG}{CG}=\frac{HE/AE+HF/BF+HG/CG}{1},\\ \frac{AH}{AE}+\frac{BH}{BF}+\frac{CH}{CG}=\frac{AH/AE+BH/BF+CH/CG}{1}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях первой и второй строк проверочной надписи соответственно значения 1 и 2, что и требовалось проверить.

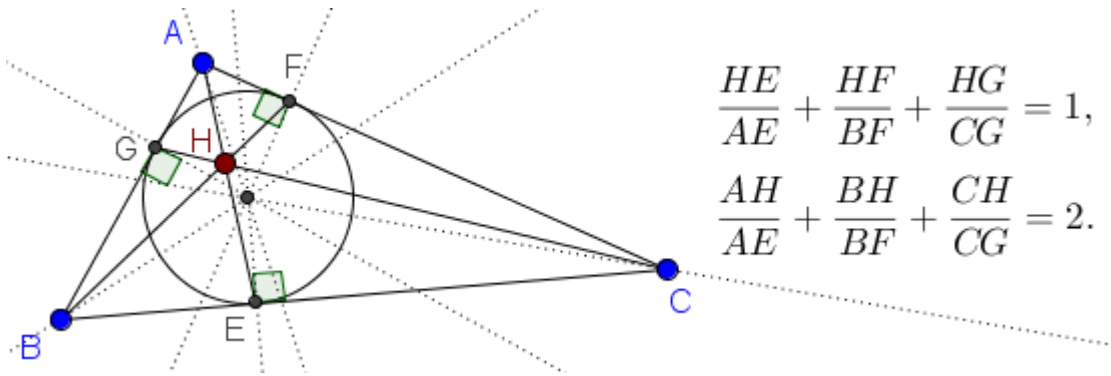


Рис. 7f. Модель для экспериментальной проверки утверждений примера 35

5.9. Точка Нагеля

Точка Нагеля (*Nagel point*) – это точка пересечения отрезков, соединяющих вершины $\triangle ABC$ с точками касания его противоположных сторон с вне-вписанными окружностями. В энциклопедии Кимберлинга она именуется как $X(8)$, и для $\triangle ABC$ через строку ввода ее можно построить командой *TriangleCenter*[$A,B,C,8$].

Начнем с определения. Пусть P_A , P_B и P_C – точки пересечения отрезков, определяющих точку Нагеля, со сторонами исходного $\triangle ABC$. Треугольник с вершинами в точках P_A , P_B и P_C называют треугольником Нагеля.

Справедливы такие утверждения:

1) Точка Нагеля, центроид и инцентр лежат на одной прямой, причем центроид делит отрезок между точкой Нагеля и инцентром в отношении 2:1. Эта прямая называется второй прямой Эйлера (или прямой Эйлера-Нагеля).

2) Каждая вершина треугольника Нагеля делит периметр исходного треугольника пополам.

3) Если через вершины исходного $\triangle ABC$ провести прямые, параллельные противоположным сторонам, то точка Нагеля $\triangle ABC$ будет точкой пересечения биссектрис полученного треугольника.

4) Перпендикуляры к сторонам треугольника, проведенные в вершинах его треугольника Нагеля, пересекаются в одной точке, которая симметрична центру вписанной окружности относительно центра описанной окружности.

5) *Расстояния от точки Нагеля до других центров.* Пусть в $\triangle ABC$: a , b и c – стороны (и их длины), противолежащие вершинам A , B и C ; r – радиус вписанной окружности; R – радиус описанной окружности; $p = (a+b+c)/2$ – полупериметр, N – точка Нагеля. Тогда квадрат расстояния или расстояние от точки X до точки Нагеля N вычисляются так:

$$a) X - \text{вершина } A \triangle ABC: AN^2 = a(4p - 3a) - \frac{4abc}{p};$$

$$b) X - \text{центр } O \text{ описанной окружности: } ON = R - 2r;$$

$$c) X - \text{центроид } M: MN^2 = \frac{4}{9}(p^2 - 16Rr + 5r^2);$$



$$d) X - \text{инцентр } I: IN^2 = p^2 - 16Rr + 5r^2;$$








$$e) X - \text{ортоцентр } H: HN^2 = 4R(R - 2r);$$

$$f) X - \text{центр окружности 9 точек } E: EN^2 = \frac{1}{4}(R^2 + 2(p^2 - 16Rr + 3r^2)).$$

Замечание. Если d – расстояние между инцентром и центром описанной окружности, то согласно формуле Эйлера $d^2 = R(R - 2r)$. Поэтому из (5e) следует, что расстояние от ортоцентра до точки Нагеля равно удвоенному расстоянию между инцентром и центром описанной окружности.

Построение точки Нагеля. Изображение, представленное на рис. 8а, является результатом построения точки Нагеля, исходя из ее определения, то есть как точки пересечения отрезков, соединяющих вершины $\triangle ABC$ с точками касания его противоположных сторон вневписанными окружностями. Это изображение является динамической моделью со свободными точками A , B и C . Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- инструментом “ Биссектриса угла” проведем биссектрисы внешних углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем центры D , E и F внеписанных окружностей;
- инструментом “ Перпендикулярная прямая” из точек D , E и F проведем перпендикулярные линии к сторонам треугольника. Инструментом “ Точка на объекте” сформируем точки G , H и I – пересечения перпендикулярных линий со сторонами треугольника. Инструментом “ Окружность по центру и точке” проведем внеписанные окружности к $\triangle ABC$ (необязательно);
- инструментом “ Отрезок” соединим отрезками вершины A , B и C $\triangle ABC$ с полученными точками G , H и I . Инструментом “ Точка на объекте” сформируем точку Нагеля пересечения J этих отрезков и через контекстное меню переименуем ее на N .

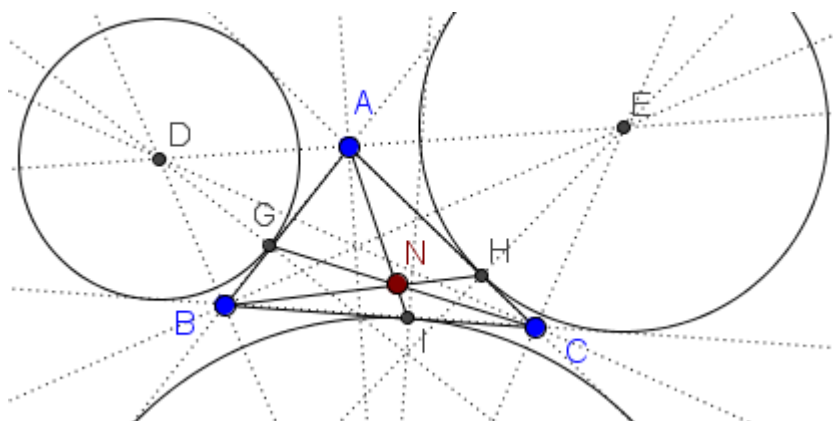




Рис. 8a. Модель для экспериментов с точкой Нагеля

Пример 36. Построить модель для экспериментальной проверки утверждения: “Точка Нагеля, центроид и инцентр лежат на одной прямой (второй прямой Эйлера), причем центроид делит отрезок между точкой Нагеля и инцентром в отношении 2:1”.

Решение. Искомая модель, представленная на рис. 8b, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 8a. Инцентр J и центроид K выведем через строку ввода соответственно командами $TriangleCenter[A,B,C,k]$ ($k=1,2$);
- инструментом “ Отрезок” соединим отрезком точку Нагеля N и инцентр J (необязательно);
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 36:

$$K \text{ лежит на } [NG] \rightarrow \boxed{AreCollinear[K,N,J]} ,$$

$$NK=2JK \rightarrow \boxed{NK==2JK}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях первой и второй строк надписи значения *true*, что и требовалось проверить.

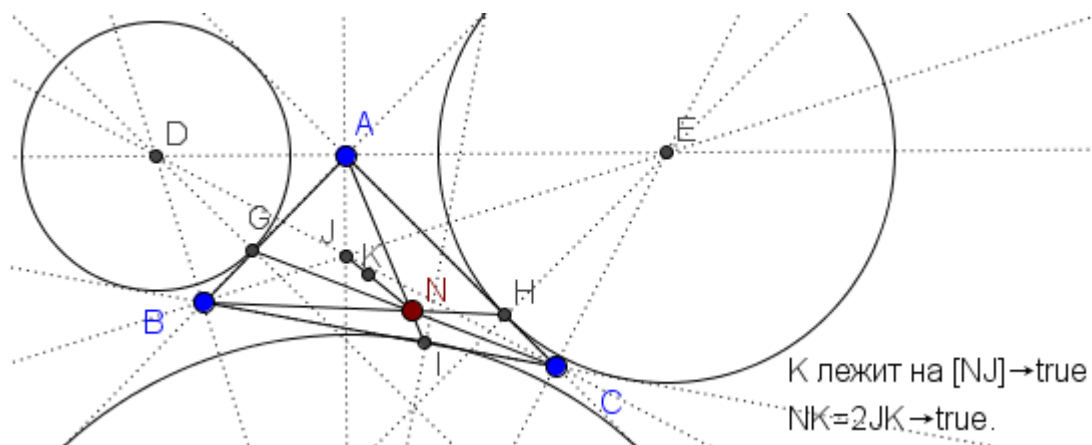



Рис. 8b. Модель для экспериментальной проверки утверждения примера 35

Пример 37. Построить модель для экспериментальной проверки утверждения: “Каждая вершина треугольника Нагеля делит периметр исходного треугольника пополам”.

Решение. Искомая модель, представленная на рис. 8с, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 8а. Построим треугольник Нагеля GHI , соединив инструментом “ Отрезок” точки G и H , H и I , I и G (не обязательно);

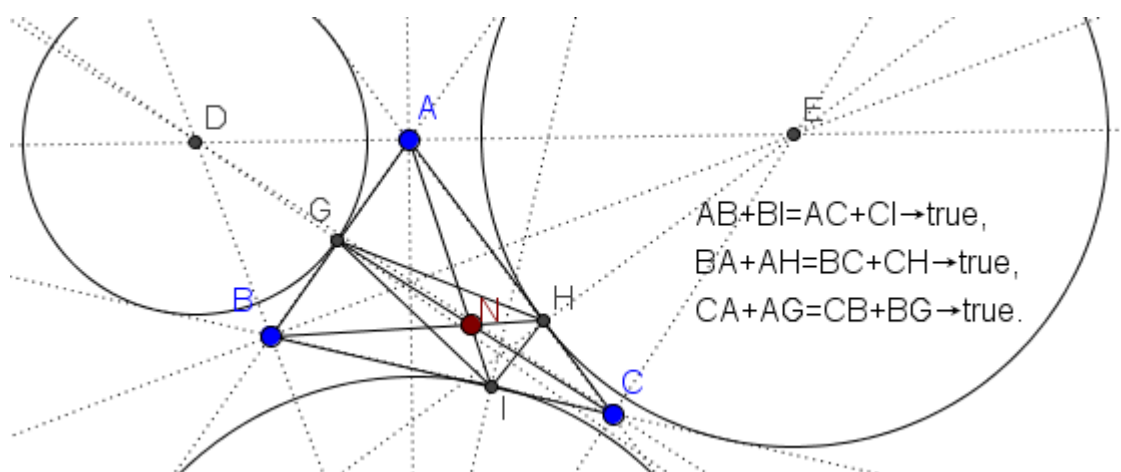



Рис. 8с. Модель для экспериментальной проверки утверждения примера 36





- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 37:

$$\begin{aligned}
 AB+BI=AC+CI &\rightarrow \boxed{AB+BI==AC+CI} , \\
 BA+AH=BC+CH &\rightarrow \boxed{BA+AH==BC+CH} , \\
 CA+AG=CB+BG &\rightarrow \boxed{CA+AG==CB+BG} .
 \end{aligned}$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значение *true*, что и требовалось проверить.

Пример 38. Построить модель для экспериментальной проверки утверждения: “Если через вершины исходного $\triangle ABC$ провести прямые, параллельные противоположным сторонам, то точка Нагеля $\triangle ABC$ будет точкой пересечения биссектрис полученного треугольника”.

Решение. Искомая модель, представленная на рис. 8d, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 8a. Инструментом “ Параллельная прямая” через вершины $\triangle ABC$ проведем прямые, параллельные противоположным сторонам. Инструментом “ Точка на объекте” сформируем точки L , K и J – пересечения проведенных прямых;
- инструментом “ Отрезок” построим $\triangle LKJ$, соединив отрезками пары точек L и K , K и J , J и L (не обязательно). Инструментом “ Биссектриса угла” проведем биссектрисы внутренних углов $\triangle LKJ$ (не обязательно);
- командой $V=Intersect[AngleBisector[L,J,K], AngleBisector[J,K,L]]$ сформируем точку V – пересечения биссектрис внутренних углов $\triangle LKJ$;

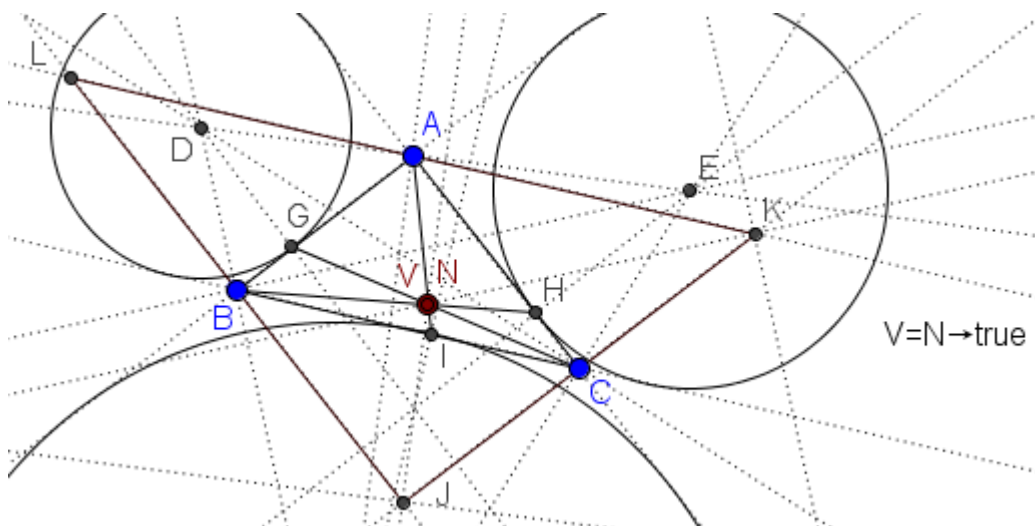



Рис. 8d. Модель для экспериментальной проверки утверждения примера 38




- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 38:

$$V=N \rightarrow \boxed{V==N} .$$

Перемещая на построенной модели независимые точки A, B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значение *true*, что и требовалось проверить.

Пример 39. Построить модель для экспериментальной проверки утверждения: “Перпендикуляры к сторонам треугольника, проведенные в вершинах его треугольника *Нагеля*, пересекаются в одной точке, которая симметрична центру вписанной окружности (инцентру) относительно центра описанной окружности”.

Решение. Искомая модель, представленная на рис. 8е, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 8с, удалив с него проверочную надпись. Инструментом “ Точка на объекте” сформируем точку J – пересечения перпендикуляров к сторонам треугольника, проведенных в вершинах треугольника Нагеля (эти перпендикуляры на рисунке уже имеются);
- командами $TriangleCenter[A,B,C,k]$ ($k=1,3$) сформируем соответственно K – центр вписанной окружности и L – центр описанной окружности. Инструментом “ Отрезок” построим отрезок KJ (не обязательно);
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 39:

$$L \text{ лежит на } [KJ] \rightarrow \boxed{AreCollinear[L,K,J]} ,$$

$$KL=LJ \rightarrow \boxed{KL==LJ} .$$

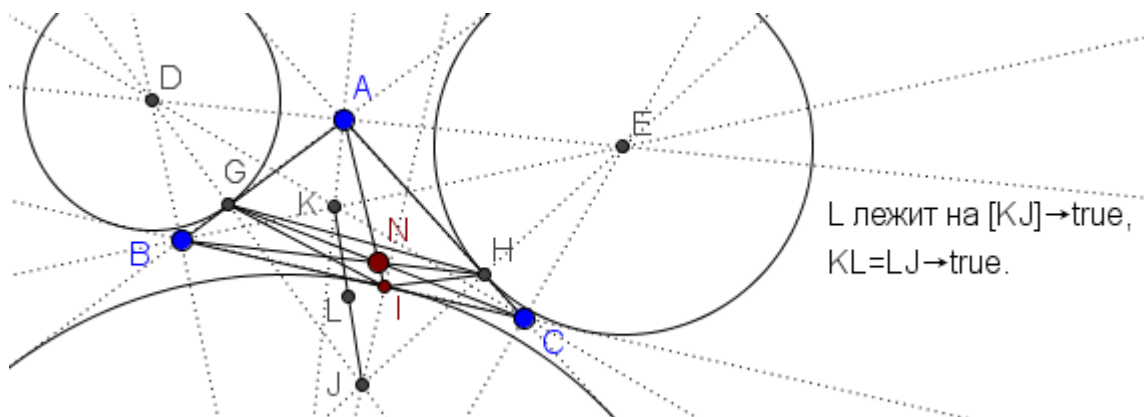




Рис. 8е. Модель для экспериментальной проверки утверждения примера 39



Перемещая на построенной модели независимые точки A, B и C по плоскости, мы всегда будем получать в правых вычисляемых частях первой и второй строк надписи значения *true*, что и требовалось проверить.

Пример 40. Построить модель для экспериментальной проверки утверждений 5a-5f (см. выше) о расстояниях от точки Нагеля до других центров.


Решение:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами *TriangleCenter* выведем все центры треугольника, участвующие в вычислениях по формулам из 5a-5f. Чтобы сразу определиться с именами центров, вводить команды будем так:

$$\begin{aligned} N &= \text{TriangleCenter}[A, B, C, 8], & O &= \text{TriangleCenter}[A, B, C, 3], \\ I &= \text{TriangleCenter}[A, B, C, 1], & H &= \text{TriangleCenter}[A, B, C, 4], \\ M &= \text{TriangleCenter}[A, B, C, 2], & E &= \text{TriangleCenter}[A, B, C, 5]; \end{aligned}$$

- инструментом “ Перпендикулярная прямая” из точки I проведем перпендикулярную линию к какой-либо стороне $\triangle ABC$, например, к BC . Инструментом “ Точка на объекте” сформируем точку D – пересечения проведенного перпендикуляра со стороной треугольника или с ее продолжением (ID – радиус вписанной окружности);

- переименуем через контекстное меню стороны $\triangle ABC$ на a, b и c и через строку ввода вычислим некоторые вспомогательные величины: $p = (a+b+c)/2$, $r = \text{Distance}[I, D]$, $R = \text{Distance}[O, B]$, $t = p^2 - 16Rr + 5r^2$;

- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 40:

$$\begin{aligned} AN^2 &= a(4p-3a) - \frac{4abc}{p} \rightarrow \boxed{AN^2 == a*(4*p-3*a)-4*a*b*c/p}, \\ ON &= R-2r \rightarrow \boxed{ON == R-2*r}, \\ MN^2 &= \frac{4}{9}(p^2-16Rr+5r^2) \rightarrow \boxed{MN^2 == 4*t/9}, \\ IN^2 &= p^2-16Rr+5r^2 \rightarrow \boxed{IN^2 == t}, \\ HN^2 &= 4R(R-2r) \rightarrow \boxed{HN^2 == 4*R*(R-2*r)}, \\ EN^2 &= \frac{1}{4}(R^2+2(p^2-16Rr+3r^2)) \rightarrow \boxed{EN^2 == (R^2+2*(t-2*r^2))/4}. \end{aligned}$$

Перемещая на построенной модели независимые точки A, B и C по плоскости, мы всегда будем получать в правых вычисляемых частях всех строк надписи значения *true*, что и требовалось проверить.

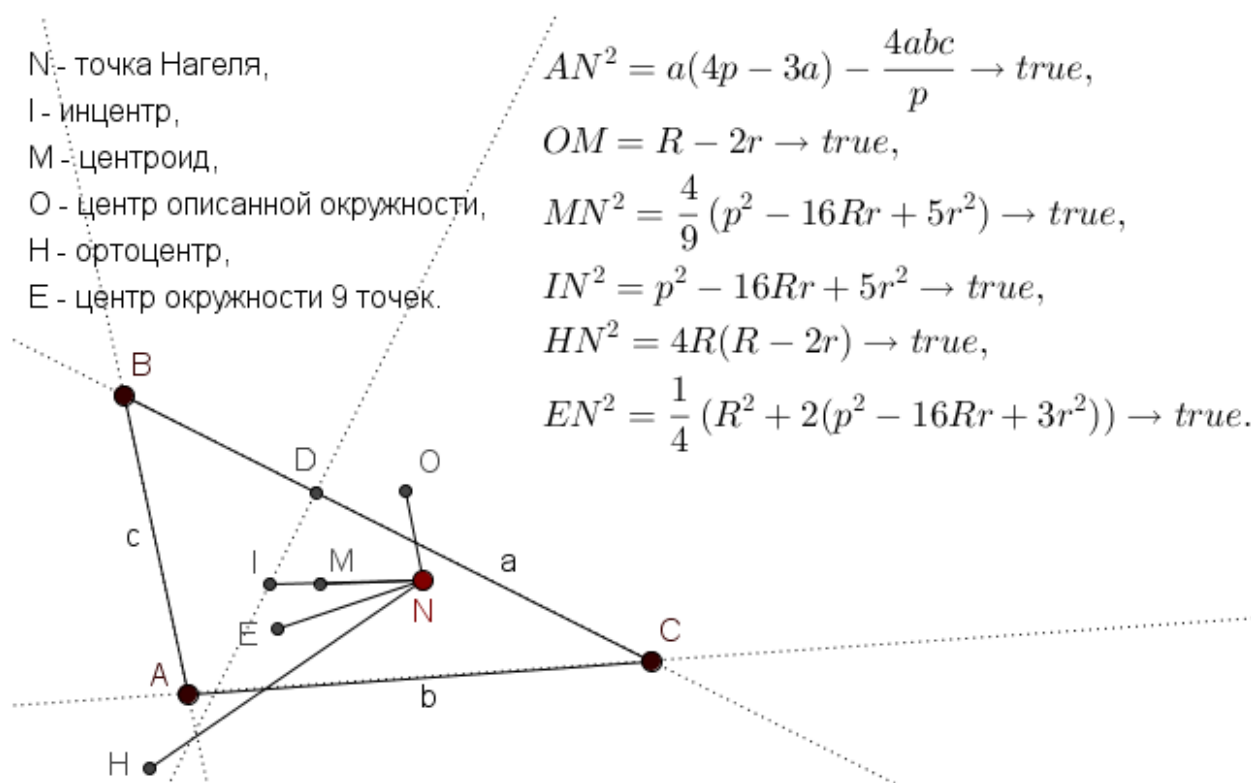


Рис. 8f. Модель для экспериментальной проверки утверждений примера 40

5.10. Средняя точка

Средняя точка (*Mittenpunkt*) или центр эллипса Мандара – это точка пересечения симедиан треугольника, у которого вершинами являются центры внеписанных окружностей исходного треугольника. В энциклопедии Кимберлинга она именуется как $X(9)$, и для $\triangle ABC$ через строку ввода ее можно строить командой `TriangleCenter[A,B,C,9]`. Средняя точка треугольника называется также и центром эллипса Мандара потому, что она является центром вписанного в треугольник эллипса, касающегося сторон треугольника в тех же точках, что и внеписанные окружности (см. рис 9d). Для построения эллипса Мандара трех точек недостаточно, но если центр эллипса Мандара тем или иным образом уже сформирован, то дополнительные точки эллипса можно получить как отражения точек касания от этого центра.

Справедливы такие утверждения:

1) Прямые, соединяющие центры внеписанных окружностей с серединами соответствующих сторон $\triangle ABC$, пересекаются в точке, являющейся средней точкой этого треугольника.

2) Пусть M , G и S – это соответственно средняя точка, точка Жергонна и центроид $\triangle ABC$. Указанные точки лежат на одной прямой, причем $GC=2CM$.

3) Пусть M , I и S – это соответственно средняя точка, инцентр и точка пересечения симедиан $\triangle ABC$. Справедливы следующие факты:

а) Точки M , I и S лежат на одной прямой, причем

$$\frac{IM}{MS} = \frac{2(a^2 + b^2 + c^2)}{(a + b + c)^2}. \quad (9)$$












б) Точки M , I и S удовлетворяют соотношениям:

$$\frac{IM}{IS} = \frac{2(a^2 + b^2 + c^2)}{2(ab + ac + bc) - (a^2 + b^2 + c^2)}, \quad (10)$$

$$\frac{MS}{IS} = \frac{(a + b + c)^2}{2(ab + ac + bc) - (a^2 + b^2 + c^2)}. \quad (11)$$

Заметим, что для любого треугольника с длинами сторон a , b и c выполняется неравенство $a^2 + b^2 + c^2 < 2(ab + ac + bc)$.

Построение средней точки (центра эллипса Мандара). Изображение, представленное на рис. 9а, является результатом построения средней точки треугольника, исходя из ее определения. Это изображение является динамической моделью со свободными точками A , B и C . Создавалась модель следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Биссектриса угла” проведем биссектрисы внешних и внутренних углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем центры D , E и F вневписанных окружностей;
- инструментом “ Прямая” проведем прямые линии через пары точек D и E , E и F , F и D . Инструментом “ Отрезок” построим $\triangle DEF$;
- инструментами “ Середина или центр” сформируем точки G , H и I , являющиеся серединами сторон $\triangle DEF$;
- инструментом “ Отражение относительно прямой” построим точки H' , G' и I' , симметричные соответственно точкам H , G и I относительно биссектрис внутренних углов $\triangle DEF$. Инструментом “ Прямая” проведем симедианы $\triangle DEF$. Инструментом “ Точка на объекте” сформируем точку J – пересечения симедиан, то есть требуемую среднюю точку;
- выведем среднюю точку еще и с помощью команды $K=TriangleCenter[A,B,C,9]$;
- инструментом “ Текст” сформируем проверочную надпись в виде:

$$K=J \rightarrow \boxed{K==J}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

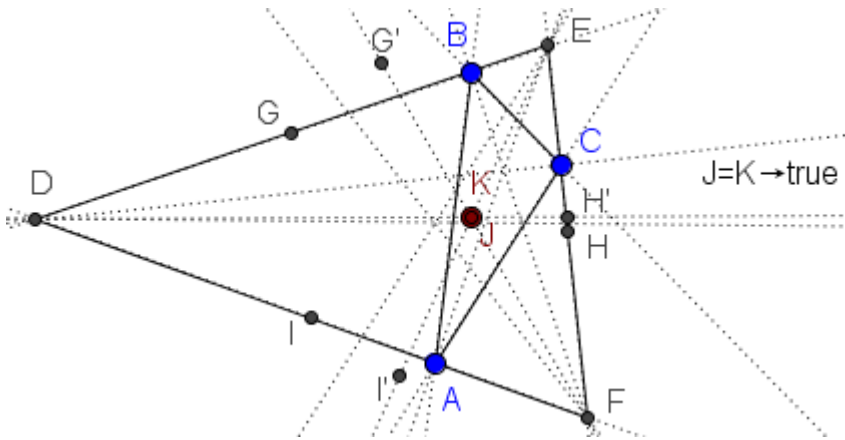


Рис. 9а. Модель для экспериментов со средней точкой треугольника

Пример 41. Построить модель для экспериментальной проверки утверждения: “Прямые, соединяющие центры вневписанных окружностей с серединами соответствующих сторон исходного треугольника, пересекаются в средней точке”.

Решение. Искомая модель, представленная на рис. 9b, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Биссектриса угла” проведем биссектрисы внешних углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем центры D , E и F вневписанных окружностей;
- инструментами “ Середина или центр” сформируем точки G , H и I , являющиеся серединами сторон $\triangle ABC$;
- инструментом “ Прямая” соединим центры вневписанных окружностей с серединами соответствующих сторон исходного треугольника. Инструментом “ Точка на объекте” сформируем точку J – пересечения проведенных прямых линий;
- командой `TriangleCenter[A,B,C,9]` сформируем среднюю точку K ;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 41:

$$J=K \rightarrow \boxed{J==K} \text{ .}$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

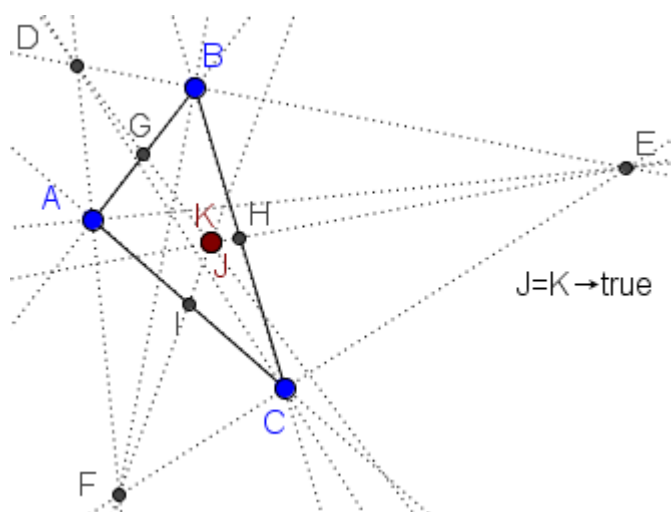






Рис. 9b. Модель для экспериментальной проверки утверждения примера 41

Пример 42. Построить модель для экспериментальной проверки утверждения: “Средняя точка M , точка Жергонна G и центроид $T \triangle ABC$ лежат на одной прямой, причем $GT=2TM$ ”.

Решение. Искомая модель, представленная на рис. 9с, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $M=TriangleCenter[A,B,C,9]$, $G=TriangleCenter[A,B,C,7]$ и $T=TriangleCenter[A,B,C,2]$ построим среднюю точку M , точку Жергонна G и центроид T ;
- инструментами “ Прямая” соединим точки G и M (необязательно);
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 42:

$$T \text{ лежит на } [GM] \rightarrow AreCollinear[T,G,M] \quad ,$$

$$GT=2TM \rightarrow GT==2*TM \quad .$$

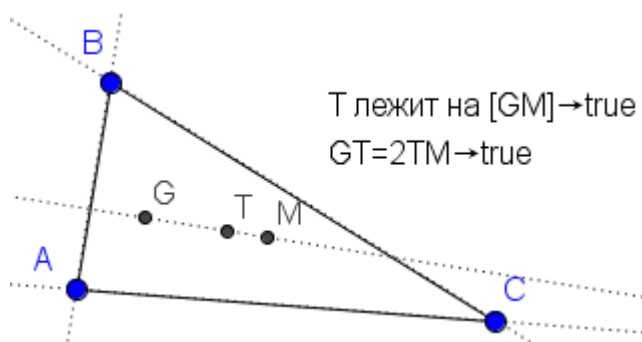











Рис. 9с. Модель для экспериментальной проверки утверждений примера 42

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях первой и второй строк надписи значения *true*, что и требовалось проверить.

Пример 43. Построить модель для экспериментирования с эллипсом Мандара.

Решение. Требуемая модель представлена на рис. 9d. Остановимся подробнее на том, как она создавалась:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Биссектриса угла” проведем биссектрисы внешних углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем центры D , E и F вневписанных окружностей;
- инструментом “ Перпендикулярная прямая” из центров вневписанных окружностей D , E и F проведем перпендикулярные линии к сторонам $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точки G , H и I – пересечения проведенных линий со сторонами $\triangle ABC$. Инструментом “ Окружность по центру и точке” проведем вневписанные окружности к $\triangle ABC$ (необязательно);
- командой `TriangleCenter[A,B,C,9]` создадим точку J – центр эллипса Мандара;
- инструментом “ Отражение относительно точки” создадим точки H' и I' , являющиеся отражением относительно центра Мандара J точек H и I соответственно;
- инструментом “ Коника по 5 точкам” проведем эллипс через точки G , H , I , H' и I' .

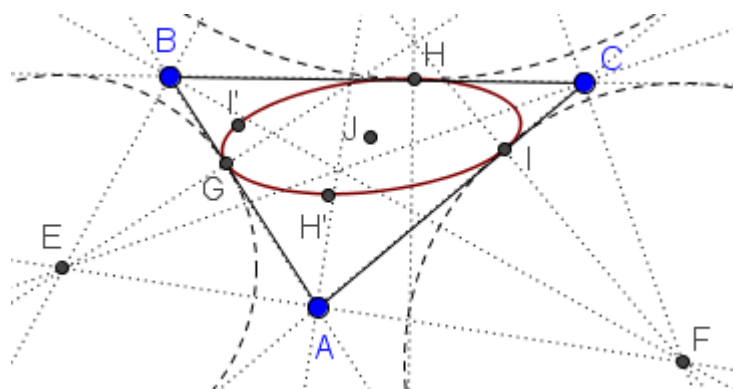






Рис. 9d. Модель для экспериментов с эллипсом Мандара

Пример 44. Построить модель для экспериментальной проверки утверждения: “Средняя точка M , инцентр I и точка пересечения симедиан S $\triangle ABC$ лежат на одной прямой, причем выполняется условие (9)”.

Решение. Искомая модель, представленная на рис. 9е, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $M=TriangleCenter[A,B,C,9]$, $I=TriangleCenter[A,B,C,1]$ и $S=TriangleCenter[A,B,C,6]$ построим среднюю точку M , инцентр I и точку пересечения симедиан S ;
- инструментами “ Прямая” соединим точки S и M ;
- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 44:

$$\begin{aligned} & \Lambda, \text{ лежит на } [SM] \rightarrow \boxed{AreCollinear[I,S,M]}, \\ & \frac{IM}{MS} = \frac{2(a^2+b^2+c^2)}{(a+b+c)^2} \rightarrow \\ & \boxed{IM/MS == 2*(AB^2+BC^2+CA^2)/(AB+BC+CA)^2}. \end{aligned}$$

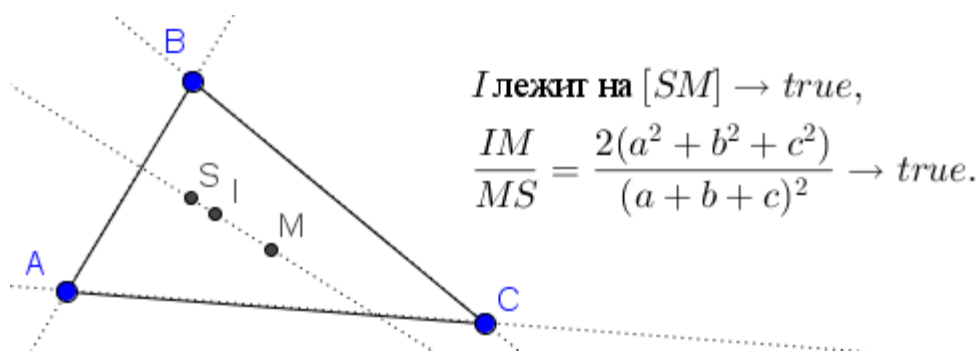


Рис. 9е. Модель для экспериментальной проверки утверждений примера 44


Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях первой и второй строк надписи значения *true*, что и требовалось проверить.

Пример 45. Построить модель для экспериментальной проверки утверждения: “Средняя точка M , инцентр I и точка пересечения симедиан S $\triangle ABC$ удовлетворяют условиям (10) и (11)”.

Решение. Искомая модель, представленная на рис. 9f, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 9е, удалив с него через контекстное меню проверочную надпись и изменив имена сторон треугольника на a , b и c ;
- через строку ввода сформируем вспомогательные переменные

$$w1=a*b+b*c+c*a, \quad w2=a^2+b^2+c^2, \quad w3=(a+b+c)^2;$$

• инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 45:

$$\frac{IM}{IS}=\frac{2(a^2+b^2+c^2)}{2(ab+bc+ca)-(a^2+b^2+c^2)}\rightarrow$$

$$\boxed{IM/IS==2*w2/(2*w1-w2)}\;,$$

$$\frac{MS}{IS}=\frac{(a+b+c)^2}{2(ab+bc+ca)-(a^2+b^2+c^2)}\rightarrow$$

$$\boxed{MS/IS==w3/(2*w1-w2)}\;.$$

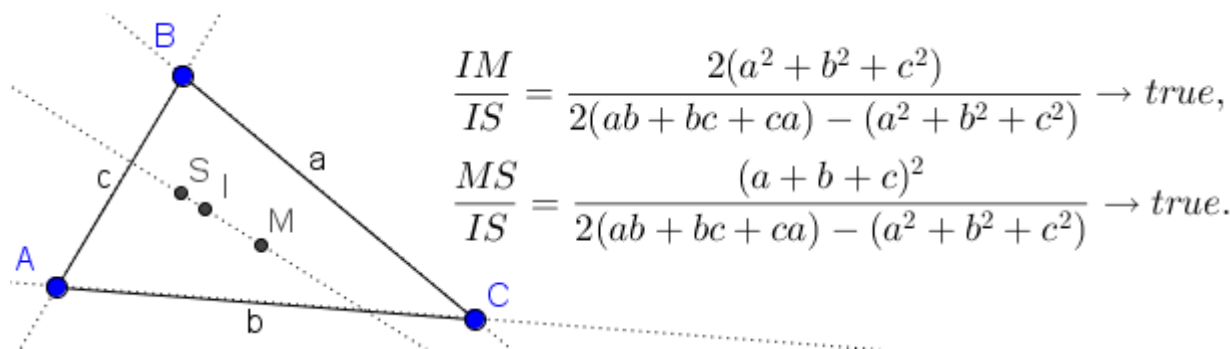


Рис. 9f. Модель для экспериментальной проверки утверждений примера 45

Перемещая на построенной модели независимые точки *A*, *B* и *C* по плоскости, мы всегда будем получать в правых вычисляемых частях первой и второй строк надписи значения *true*, что и требовалось проверить.

5.11. Центр Шпикера

Центр Шпикера (Spieker center) – это инцентр срединного треугольника, то есть центр окружности, вписанной в треугольник с вершинами в серединах сторон исходного треугольника. В энциклопедии Кимберлинга он именуется как *X(10)*, и для $\triangle ABC$ через строку ввода его можно строить командой *TriangleCenter[A,B,C,10]*.

Определения. а) Пусть имеется окружность с центром в точке *O* радиуса *r* и точка *M*, расположенная вне окружности. Степенью точки *M* относительно данной окружности называют величину OM^2-r^2 . Таким образом, корень квадратный из степени точки *M* равен расстоянию от *M* до точки касания окружности с касательной, проведенной к ней из точки *M*.

б) Радикальной окружностью называется окружность с центром в точке Шпикера *S* радиуса

$$r=\frac{1}{2}\sqrt{ab+ac+bc-\frac{2abc}{a+b+c}}. \tag{12}$$

Справедливы такие утверждения:



1) Центр Шпикера X_{10} лежит на второй прямой Эйлера, причем $X_1X_{10}=X_{10}X_8$, $X_1X_2=2X_2X_{10}$. Ранее мы видели, что инцентр X_1 , центроид X_2 и точка Нагеля X_8 также лежат на второй прямой Эйлера и $X_8X_2=2X_2X_1$.


2) Центр Шпикера X_{10} лежит на одной прямой со средней точкой X_9 и ортоцентром X_4 .


3) Центр Шпикера лежит на гиперболе Киперта, проходящей через вершины исходного треугольника, его ортоцентр X_4 и центроид X_2 .



4) Центр Шпикера S по отношению к внеписанным окружностям имеет одинаковые степени. Радикальная окружность пересекает внеписанные окружности под прямым углом.

Построение центра Шпикера. Изображение, представленное на рис. 10а, является результатом построения центра Шпикера, исходя из его определения, то есть как инцентра срединного треугольника для исходного $\triangle ABC$. Это изображение является динамической моделью со свободными точками A , B и C . Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- инструментами “ Середина или центр” сформируем точки D , E и F , являющиеся серединами сторон $\triangle ABC$, и добавим одинаковые риски на равные отрезки сторон $\triangle ABC$. Для реализации последнего действия сначала необходимо провести отрезки BE , EC , CF , FA , AD и DB , а затем уже через панель “Настройки” ставить на них указанные риски (декорации).

- инструментом “ Отрезок” построим срединный $\triangle DEF$;

- инструментом “ Биссектриса угла” проведем биссектрисы внутренних углов срединного $\triangle DEF$. Инструментом “ Точка на объекте” сформируем точку пересечения построенных биссектрис и через контекстное меню поменяем ее имя на X_{10} . Центр Шпикера X_{10} построен;

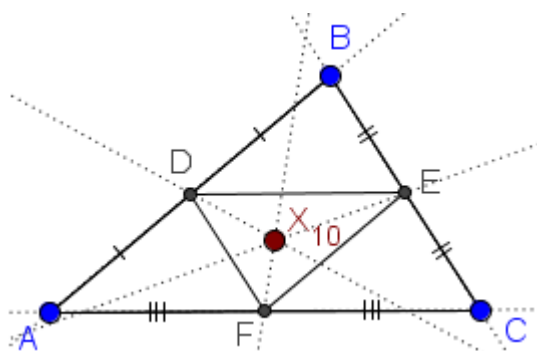






Рис. 10а. Модель для экспериментов с центром Шпикера

Пример 46. Построить модель для экспериментальной проверки утверждения: “Центр Шпикера X_{10} вместе с инцентром X_1 , центроидом X_2 и точкой

Нагеля X_8 лежит на второй прямой Эйлера, причем $X_1X_{10}=X_{10}X_8$, $X_1X_2=2X_2X_{10}$ и $X_8X_2=2X_2X_1$.

Решение. Искомая модель, представленная на рис. 10b, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $X_{10}=TriangleCenter[A,B,C,10]$, $X_1=TriangleCenter[A,B,C,1]$, $X_2=TriangleCenter[A,B,C,2]$, $X_8=TriangleCenter[A,B,C,8]$ выведем требуемые центры X_{10} , X_1 , X_2 и X_8 $\triangle ABC$;
- инструментом “ Прямая” проведем прямую через точки X_1 и X_8 . Это и есть вторая прямая Эйлера. Ранее мы видели, что точка X_2 также лежит на этой прямой, причем выполняется соотношение $X_8X_2=2X_2X_1$;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 46:

$$\begin{aligned}
 X_{10} \text{ лежит на 2 прямой Эйлера} &\rightarrow \boxed{AreCollinear[X_{10},X_1,X_8]} , \\
 X_1X_{10}=X_{10}X_8 &\rightarrow \boxed{Distance[X_1,X_{10}]==Distance[X_{10},X_8]} , \\
 X_1X_2=2X_2X_{10} &\rightarrow \boxed{Distance[X_1,X_2]==2*Distance[X_2,X_{10}]} , \\
 X_8X_2=2X_2X_1 &\rightarrow \boxed{Distance[X_8,X_2]==2*Distance[X_2,X_1]} .
 \end{aligned}$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях всех строк надписи значения *true*, что и требовалось проверить.

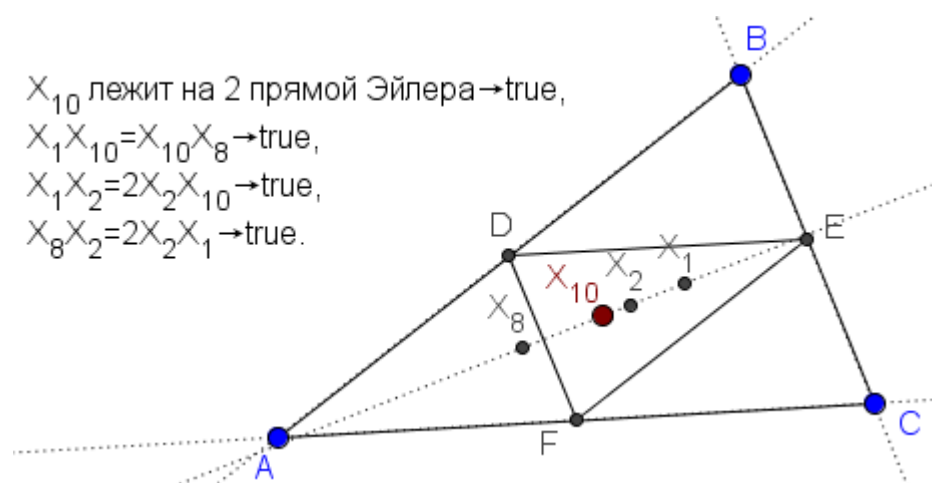




Рис. 10b. Модель для экспериментальной проверки утверждений примера 46

Пример 47. Построить модель для экспериментальной проверки утверждения: “Центр Шпикера X_{10} лежит на одной прямой со средней точкой X_9 и ортоцентром X_4 .”

Решение. Искомая модель, представленная на рис. 10с, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 10b. Прежде всего удалим с модели проверочную надпись;
- командами $X_9=TriangleCenter[A,B,C,9]$ и $X_4=TriangleCenter[A,B,C,4]$ выведем требуемые центры X_9 и X_4 $\triangle ABC$;
- инструментом “ Прямая” проведем прямую через точки X_9 и X_4 ;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 47:

$$X_{10} \text{ лежит на } [X_9X_4] \rightarrow \boxed{AreCollinear[X_{10},X_9,X_4]} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

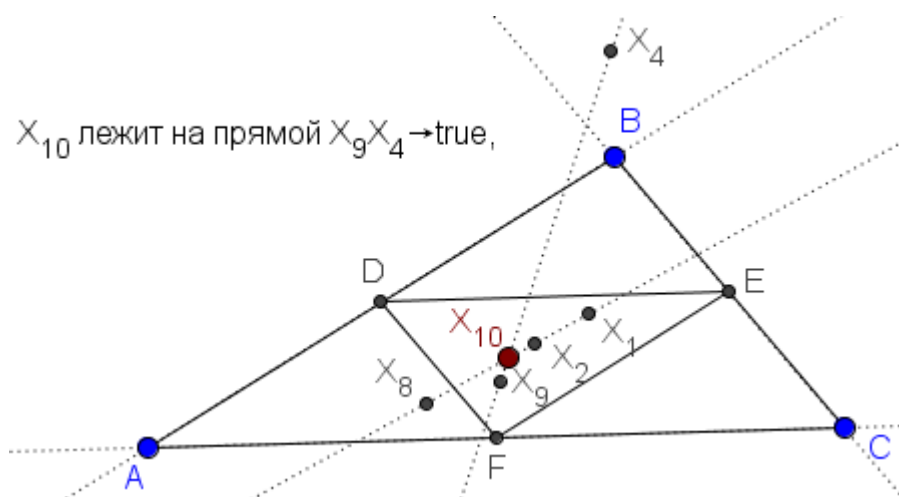






Рис. 10с. Модель для экспериментальной проверки утверждений примера 47

Пример 48. Построить модель для экспериментальной проверки утверждения: “Центр Шпикера X_{10} лежит на гиперболе Киперта, проходящей через вершины исходного треугольника, его ортоцентр X_4 и центроид X_2 ”.

Решение. Искомая модель, представленная с левой стороны рис. 10d, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $X_4=TriangleCenter[A,B,C,4]$ и $X_2=TriangleCenter[A,B,C,2]$ построим ортоцентр и центроид $\triangle ABC$;
- инструментом “ Коника по 5 точкам” проведем гиперболу Киперта через точки A , B , C , X_4 и X_2 . Она получит имя p . Сделаем его видимым;
- инструментом “ Кнопка” выведем на панель “Полотно” командную кнопку *Ok*, поменяв ее заголовок на “ p и X ” и задав *GeoGebra*-скрипт в виде

$Relation[p, X_{10}]$. Тогда при изменениях независимых точек A, B и C в любой статичной ситуации щелчок по кнопке “ p и X ” будет приводить к выполнению команды $Relation[p, X_{10}]$, а по ней в отдельном окне будет выводиться сообщение вида “ X_{10} лежит на p ” или “ X_{10} не лежит на p ”. Впрочем, проверять лежит ли центр Шпикера X_{10} на конике p можно и инструментом “ Отношение объектов”, действие которого приводит к выводу тех же самых сообщений. В нашем случае мы всегда будем иметь сообщение, показанное с правой стороны рис. 10d, что и требовалось проверить.

Однако проще и удобней работать с проверочными надписями, как мы поступали в предыдущих примерах. В нашем случае надпись можно сформировать так:

$$X_{10} \text{ на гиперболе Киперта} \rightarrow \boxed{IsInRegion[X_{10}, p]} .$$

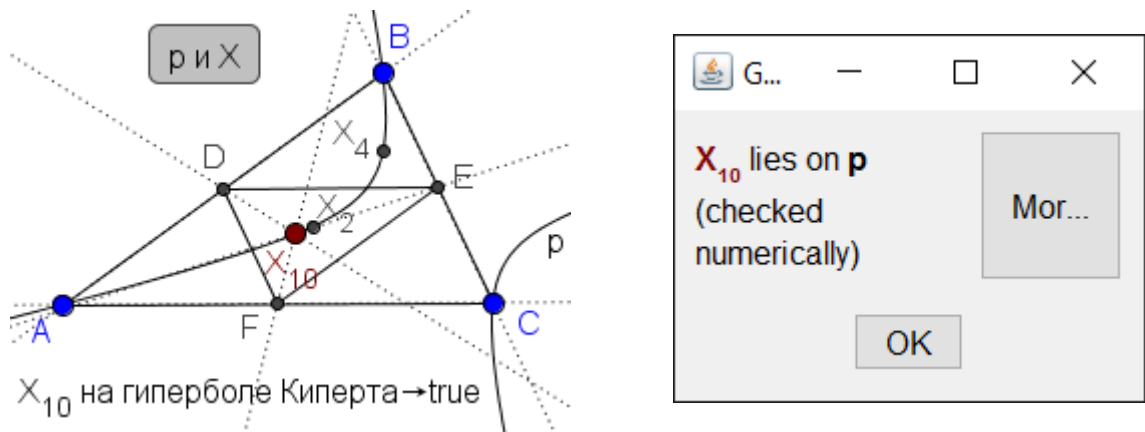









Рис. 10d. Модель для экспериментальной проверки утверждений примера 48

Пример 49. Построить модель для экспериментальной проверки утверждений:

- а) Центр Шпикера S по отношению к вневписанным окружностям имеет одинаковые степени;
- б) Радикальная окружность пересекает вневписанные окружности под прямым углом.

Решение. Искомая модель, представленная на рис. 10e, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$. Переименуем стороны треугольника на a, b и c ;
- командой $S=TriangleCenter[A,B,C,10]$ выведем центр Шпикера $\triangle ABC$;
- инструментом “ Биссектриса угла” проведем биссектрисы внешних углов $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точки D, E и F пересечения построенных биссектрис, то есть центры вневписанных окружностей;

- инструментом “ Перпендикулярная прямая” из центров внеписанных окружностей D , E и F проведем перпендикулярные линии к сторонам $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точки пересечения проведенных линий со сторонами $\triangle ABC$. Инструментом “ Окружность по центру и точке” проведем внеписанные окружности к $\triangle ABC$;
- через строку ввода сформируем вспомогательную переменную r , вычислив ее по формуле (12): $r = \sqrt{(a*b + b*c + c*a - 2a*b*c/(a+b+c))/2}$;
- инструментом “ Окружность по центру и радиусу” проведем радикальную окружность, то есть окружность с центром в точке S радиуса r . Инструментом “ Пересечение” сформируем точки J , K , M , L , N и O пересечения радикальной окружности с внеписанными окружностями;
- инструментом “ Отрезок” сформируем $\triangle SDJ$. Чтобы проверить, что в точке J радикальная окружность пересекается с внеписанной окружностью под прямым углом, достаточно установить, что $\triangle SDJ$ – прямоугольный (SD – гипотенуза). Для точек K , M , L , N и O проверка реализуется таким же способом и ее проводить мы не будем;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 49:

$$SD^2 - DG^2 = SE^2 - HE^2 = SF - FI^2 \rightarrow$$

$$\boxed{SD^2 - DG^2 == SE^2 - HE^2 == SF - FI^2},$$

$$SD^2 = DJ^2 + SJ^2 \rightarrow \boxed{SD^2 == DJ^2 + SJ^2}.$$

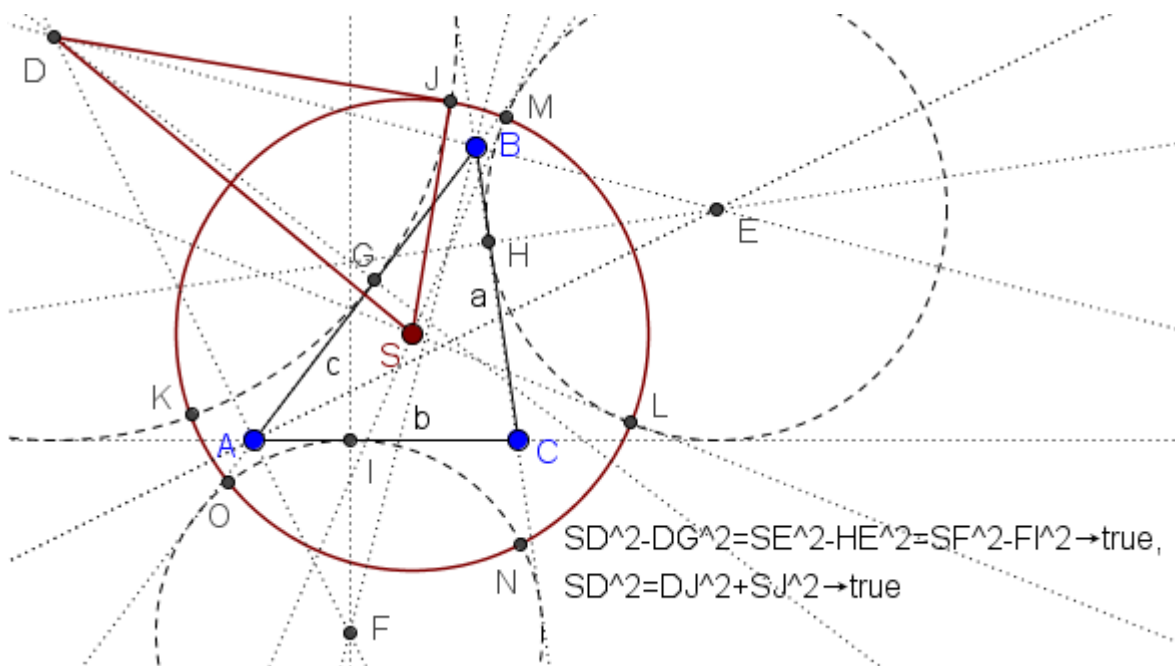


Рис. 10е. Модель для экспериментальной проверки утверждений примера 49

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях надписи значения *true*, что и требовалось проверить.

5.12. Точка Фейербаха

Точка Фейербаха (Feuerbach point) – это точка F касания вписанной окружности и окружности 9 точек, то есть окружности Фейербаха. Кроме того, точками Фейербаха Fa , Fb и Fc часто называют также и точки касания окружности девяти точек с тремя внеписанными окружностями. В энциклопедии Кимберлинга точка Фейербаха именуется как $X(11)$, и для $\triangle ABC$ через строку ввода ее можно строить командой `TriangleCenter[A,B,C,11]`.

Справедливы такие утверждения:

1) Точка Фейербаха F лежит на линии, соединяющей центр окружности девяти точек и инцентр.

2) Если t_1, t_2, t_3 – расстояния от точки Фейербаха F до вершин срединного треугольника, то $t_1+t_2+t_3=2\max(t_1,t_2,t_3)$, то есть наибольшее из трех расстояний равно сумме двух других расстояний.

3) Если в $\triangle ABC$ точка I – инцентр и F, Fa, Fb и Fc – точки Фейербаха, то прямые IF, AFa, BFb, CFc пересекаются в одной точке.



4) Гиперболой Фейербаха называют гиперболу, проходящую через вершины $\triangle ABC$, его инцентр и ортоцентр. Точка Фейербаха F является центром гиперболы Фейербаха и лежит на окружности, проведенной через основания биссектрис внутренних углов треугольника.









5) На гиперболе Фейербаха кроме инцентра X_1 и ортоцентра X_4 лежат также: X_9 – средняя точка, X_8 – точка Нагеля, X_7 – точка Жергонна и X_{21} – точка Шиффлера. Вне диапазона X_1 - X_{21} имеются и другие центры, лежащие на гиперболе Фейербаха, например, X_{79} .

6) Пусть в $\triangle ABC$: r – радиус вписанной окружности; R – радиус описанной окружности; $p=(a+b+c)/2$ – полупериметр, L – расстояние от точки Фейербаха до центроида G . Тогда

$$L^2 = \frac{p^2(R+r) - r(4R+r)^2}{9(R-2r)}. \tag{13}$$

Построение точки Фейербаха F . Изображение, представленное на рис. 11а, является результатом построения точки Фейербаха F , исходя из ее определения как точки касания вписанной окружности и окружности 9 точек. Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- инструментами “ *Середина или центр*” сформируем точки D , E и F , являющиеся серединами сторон $\triangle ABC$;
- инструментом “ *Биссектриса угла*” проведем биссектрисы внутренних углов $\triangle ABC$. Инструментом “ *Точка на объекте*” сформируем центр G вписанной окружности;
- инструментом “ *Перпендикулярная прямая*” из точки G проведем прямую, перпендикулярную какой-либо стороне $\triangle ABC$, например, BC . Инструментом “ *Точка на объекте*” сформируем точку H – пересечения этой прямой с BC ;
- инструментом “ *Окружность по центру и точке*” через G и H проведем вписанную окружность для $\triangle ABC$. Инструментом “ *Окружность по трем точкам*” через E , F и G проведем окружность девяти точек. Инструментом “ *Пересечение*” сформируем точку I пересечения этих окружностей и изменим имена точек I на F и F на I . Модель для экспериментов с точкой Фейербаха F создана.

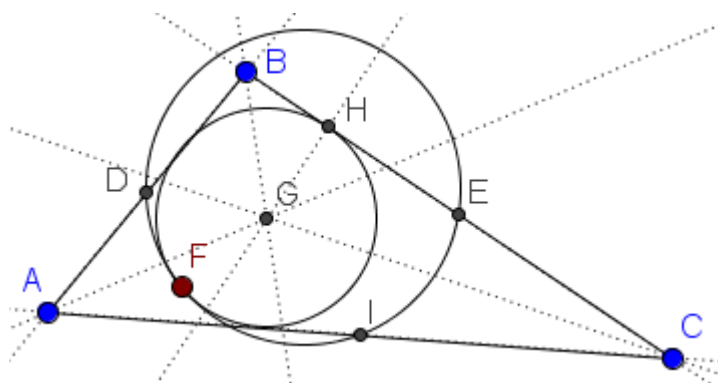










Рис. 11а. Модель для экспериментов с точкой Фейербаха F


Построение точек Фейербаха F , F_a , F_b и F_c . Изображение, представленное на рис. 11b, является результатом построения точек Фейербаха F , F_a , F_b и F_c , исходя из их определения, то есть как точек касания вписанной окружности и трех внеписанных окружностей с окружностью девяти точек. Создавалось оно следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ *Прямая*” и “ *Отрезок*” построим $\triangle ABC$;
- инструментом “ *Биссектриса угла*” проведем биссектрисы внутренних и внешних углов $\triangle ABC$. Инструментом “ *Точка на объекте*” сформируем центры D , E , F и G вписанной и внеписанных окружностей как точки пересечения соответствующих биссектрис;
- инструментом “ *Перпендикулярная прямая*” из точек D , E , F и G проведем прямые, перпендикулярные линиям соответственно к сторонам AB , AB ,

BC и CA $\triangle ABC$. Инструментом “ Точка на объекте” сформируем точки H , I , J и K пересечения этих линий со сторонами треугольника;

- инструментом “ Окружность по центру и точке” построим вписанную и внеписанные окружности $\triangle ABC$.

- инструментом “ Середина или центр” сформируем точки L , M и N , являющиеся серединами сторон $\triangle ABC$. Инструментом “ Окружность по трем точкам” через L , M и N проведем окружность девяти точек;

- инструментом “ Пересечение” создадим точки пересечения окружности девяти точек с вписанной окружностью и тремя внеписанными окружностями. Переименуем эти точки на F , F_a , F_b и F_c (см. рис. 11b). Модель для экспериментов с точками Фейербаха F , F_a , F_b и F_c создана.

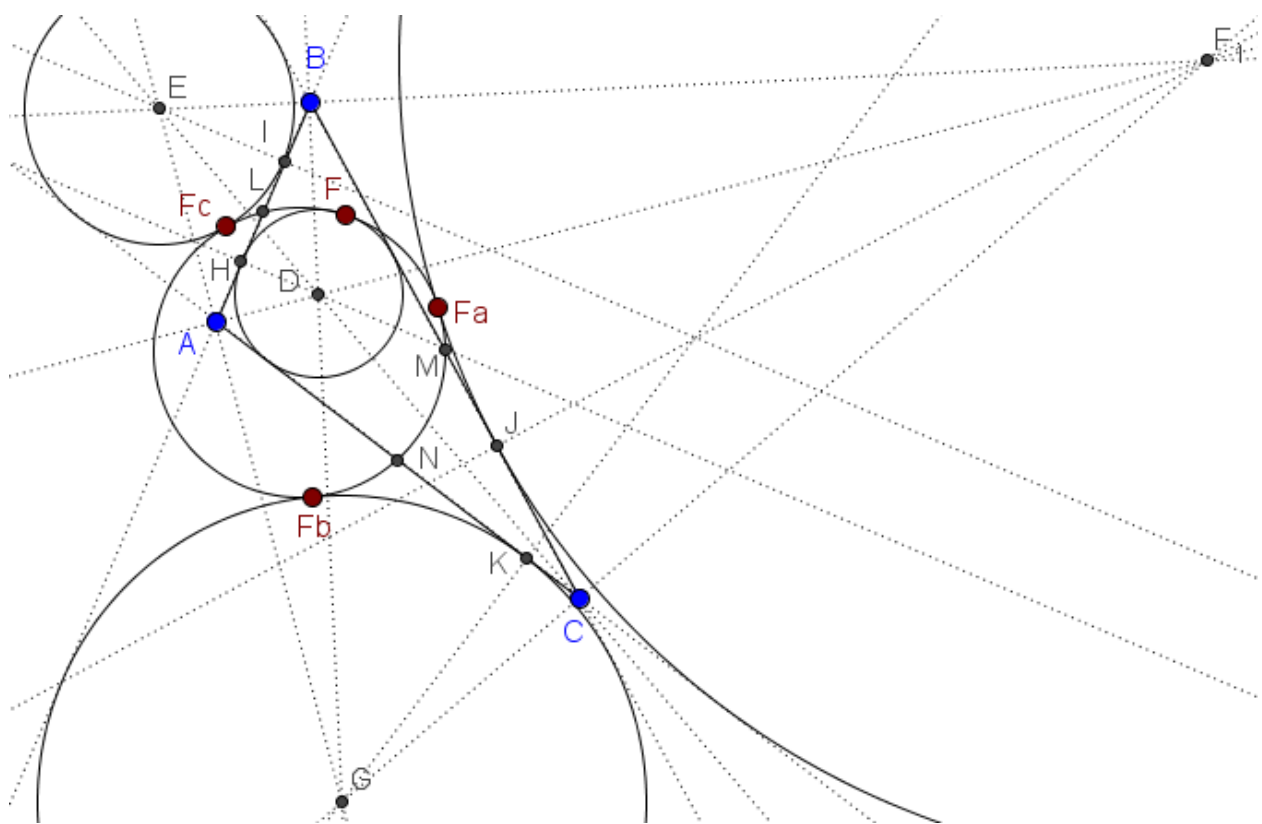





Рис. 11b. Модель для экспериментов с точками Фейербаха F , F_a , F_b и F_c


Пример 50. Построить модель для экспериментальной проверки утверждения: “Точка Фейербаха F лежит на линии, соединяющей центр окружности девяти точек и инцентр”.

Решение. Искомая модель, представленная на рис. 11b, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 11a. Инструментом “ Срединный перпендикуляр” проведем

прямые, перпендикулярные хордам DE и EI окружности девяти точек. Инструментом “ Пересечение” сформируем центр окружности девяти точек как точку J пересечения проведенных перпендикуляров;

- инструментом “ Прямая” проведем прямую через точки J и G и изменим стиль ее вывода. Визуально можно наблюдать, что центр окружности девяти точек лежит на этой прямой.

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 50:

$$F \text{ лежит на прямой } JG \rightarrow \boxed{AreCollinear[F, J, G]} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

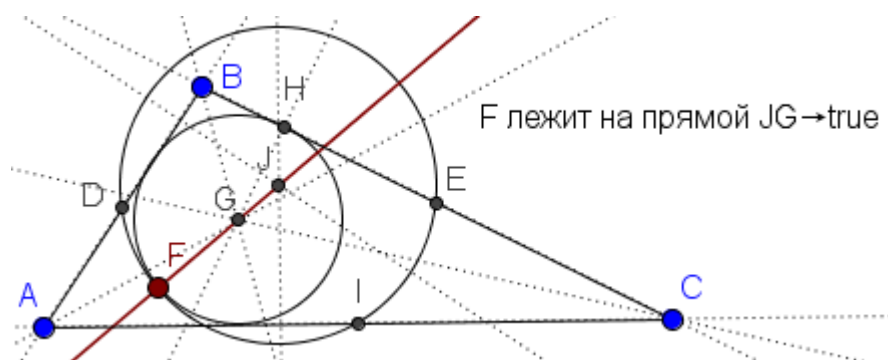




Рис. 11с. Модель для экспериментальной проверки утверждения примера 50

Пример 51. Пусть t_1, t_2, t_3 – расстояния от точки Фейербаха F до вершин срединного треугольника. Построить модель для экспериментальной проверки утверждения: “Наибольшее из трех расстояний t_1, t_2, t_3 равно сумме двух других расстояний, то есть $t_1+t_2+t_3=2\max(t_1,t_2,t_3)$ ”.

Решение. Искомая модель, представленная на рис. 11д, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 11а. Инструментом “ Отрезок” соединим вершины срединного треугольника, то есть точки D, E и I с центром F окружности девяти точек (не обязательно);

- через строку ввода сформируем три вспомогательные переменные: $t_1=Distance[F,D], t_2=Distance[F,E], t_3=Distance[F,I]$;

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 51:

$$t_1+t_2+t_3=2\max(t_1,t_2,t_3) \rightarrow \boxed{t_1+t_2+t_3==2*Max(\{t_1,t_2,t_3\})} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения $true$, что и требовалось проверить.

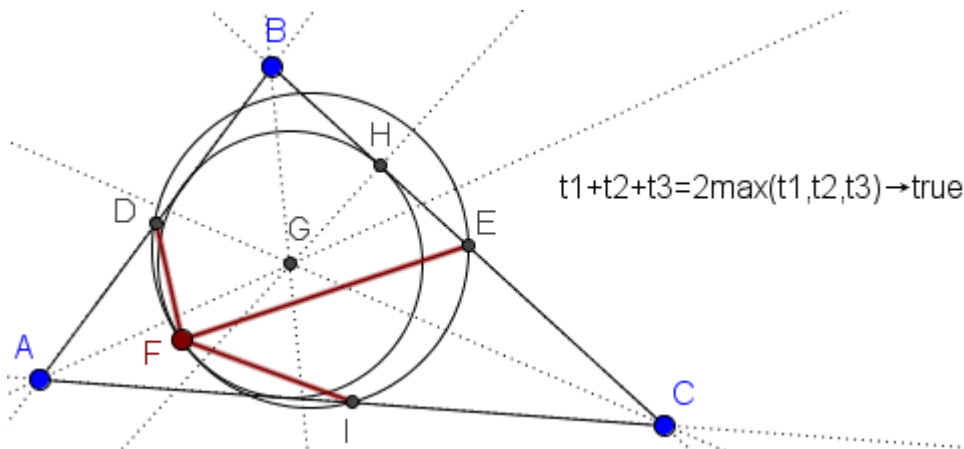


Рис. 11d. Модель для экспериментальной проверки утверждения примера 50

Пример 52. Построить модель для экспериментальной проверки утверждения: “Если в $\triangle ABC$ точка D – инцентр и F , Fa , Fb и Fc – точки Фейербаха, то прямые DF , AFa , BFb , CFc – пересекаются в одной точке”.

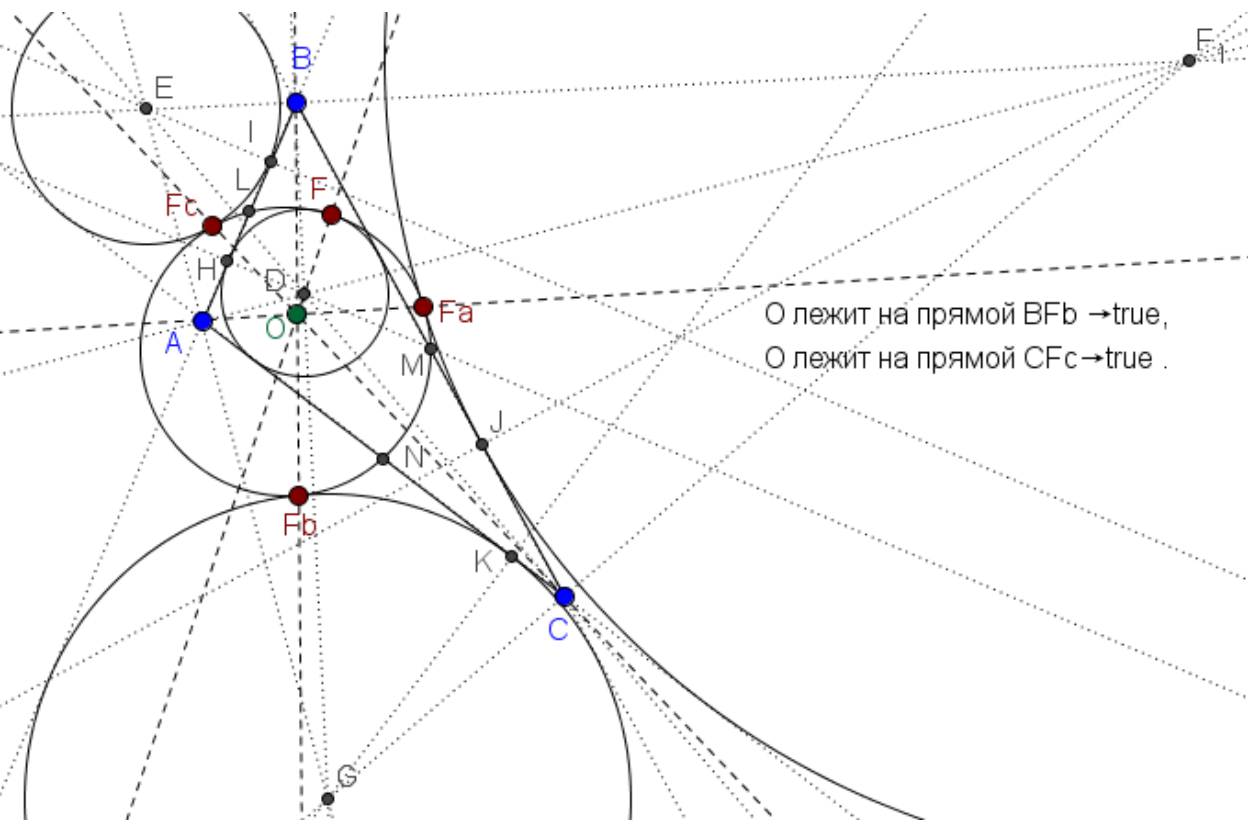





Рис. 11e. Модель для экспериментальной проверки утверждения примера 52

Решение. Искомая модель, представленная на рис. 11e, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 11b. Инструментом “ Прямая” проведем прямые через пары точек F, D и A, Fa . Инструментом “ Пересечение” сформируем точку O пересечения проведенных прямых;



- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую, что точка O лежит также и на прямых, проходящих через пары точек B, Fb и C, Fc :

$$\begin{aligned} O \text{ лежит на прямой } BFb &\rightarrow \boxed{AreCollinear[O, B, Fb]} , \\ O \text{ лежит на прямой } CFc &\rightarrow \boxed{AreCollinear[O, C, Fc]} . \end{aligned}$$


Перемещая на построенной модели независимые точки A, B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.




Пример 53. Построить модель для экспериментальной проверки утверждения: “Точка Фейербаха F является центром гиперболы Фейербаха и лежит на окружности, проведенной через основания биссектрис внутренних углов треугольника”.

Решение. Искомая модель, представленная на рис. 11f, создавалась следующим образом:


- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- командами $X_1=TriangleCenter[A,B,C,1]$, $X_4=TriangleCenter[A,B,C,4]$ и $X_{11}=TriangleCenter[A,B,C,11]$ выводим соответственно инцентр X_1 , ортоцентр X_4 и точку Фейербаха X_{11} ;

- инструментом “ Коника по пяти точкам” проведем гиперболу Фейербаха через вершины A, B, C треугольника и его центры X_1 и X_4 . Она получит имя g , которое сделаем видимым;

- инструментом “ Биссектриса угла” проведем биссектрисы внутренних углов $\triangle ABC$. Инструментом “ Пересечение” сформируем точки пересечения D, E и F построенных биссектрис со сторонами треугольника. Инструментом “ Окружность по трем точкам” через D, E и F проведем окружность. Она получит имя k , которое сделаем видимым;

- командой $Center[g]$ выведем центр G гиперболы g ;

- остается убедиться, что $X_{11}=G$ и X_{11} лежит на окружности k . Для этого инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 53:

$$X_{11}=G \rightarrow \boxed{X_{11}=G} ,$$

X_{11} лежит на окружности $\rightarrow \boxed{X_{11} == \text{Intersect}[k, X_{11}]}$.

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

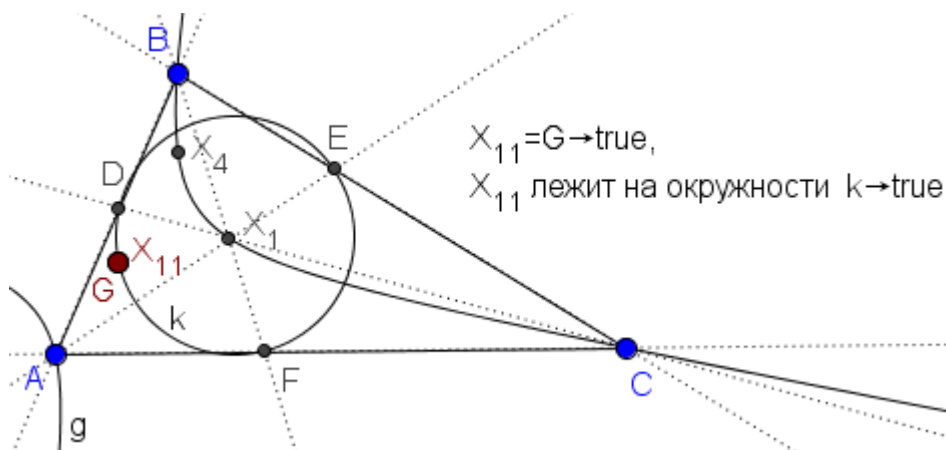



Рис. 11f. Модель для экспериментальной проверки утверждения примера 53

Пример 54. Построить модель для экспериментальной проверки утверждения: “На гиперболе Фейербаха кроме инцента X_1 , ортоцентра X_4 лежат также центры: X_9 – средняя точка, X_8 – точка Нагеля, X_7 – точка Жергонна и X_{21} – точка Шиффлера.

Решение. Искомая модель, представленная на рис. 11g, создавалась следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 11f. Удалим проверочную надпись и центр G гиперболы g ;
- командами $X_9 = \text{TriangleCenter}[A, B, C, 9]$, $X_8 = \text{TriangleCenter}[A, B, C, 8]$, $X_7 = \text{TriangleCenter}[A, B, C, 7]$ и $X_{21} = \text{TriangleCenter}[A, B, C, 21]$, выведем соответственно среднюю точку X_9 , точку Нагеля X_8 , точку Жергонна X_7 и точку Шиффлера X_{21} ;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждения примера 54:

X_9 лежит на гиперболе $g \rightarrow \boxed{X_9 == \text{Intersect}[g, X_9]}$,

X_8 лежит на гиперболе $g \rightarrow \boxed{X_8 == \text{Intersect}[g, X_8]}$,

X_7 лежит на гиперболе $g \rightarrow \boxed{X_7 == \text{Intersect}[g, X_7]}$,

X_{21} лежит на гиперболе $g \rightarrow \boxed{X_{21} == \text{Intersect}[g, X_{21}]}$.

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

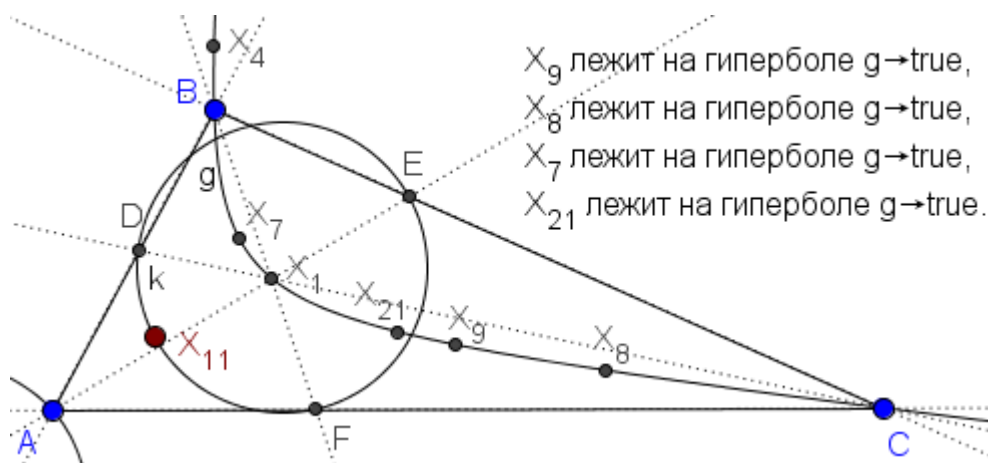


Рис. 11g. Модель для экспериментальной проверки утверждения примера 54

Замечание. Вне диапазона точек X_1 - X_{21} имеются и другие центры, лежащие на гиперболе Фейербаха, например, X_{79} .

Пример 55. Построить модель для экспериментальной проверки соотношения (13) утверждения (6) (см. выше).

Решение. Искомая модель, представленная на рис. 11h, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $F=TriangleCenter[A,B,C,11]$ и $G=TriangleCenter[A,B,C,2]$ выводим соответственно точку Фейербаха F и центр оид $\triangle ABC$;
- командами $r=Radius[Incircle[A,B,C]]$ и $R=Radius[Circle[A,B,C]]$ вычислим радиусы соответственно вписанной и описанной окружностей. Сформируем вспомогательную переменную, равную полупериметру треугольника – $p=(AB+BC+CA)/2$;
- инструментом “ Отрезок” соединим отрезком точки F и G . Через контекстное меню изменим имя полученного отрезка на L ;
- инструментом “ Текст” сформируем в *LaTeX*-режиме проверочную надпись, подтверждающую или опровергающую утверждение примера 55:

$$L^2 = \frac{p^2(R+r) - r(4R+r)^2}{9(R-2r)}$$

$$\boxed{L^2 == (p^2 * (R+r) - r * (4R+r)^2) / (9(R-2r))} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

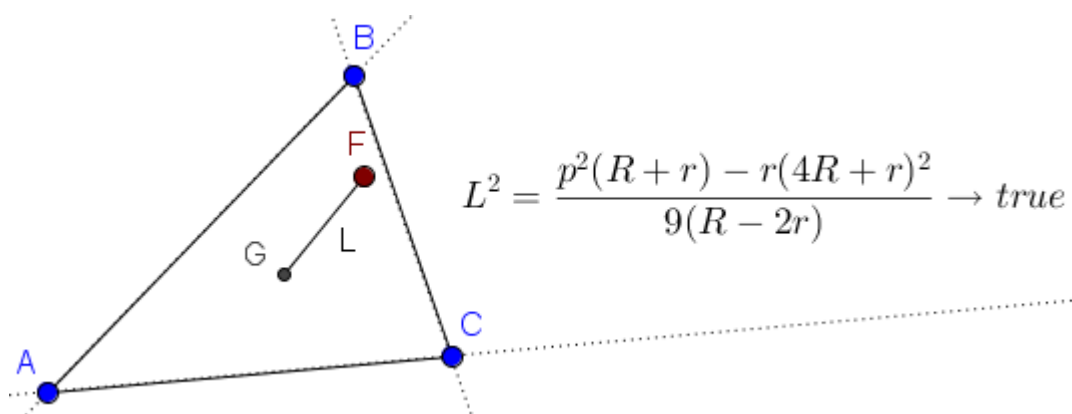


Рис. 11h. Модель для экспериментальной проверки утверждения примера 55

5.13. Точка, гармонически сопряженная точке Фейербаха




Напомним, что, две точки F и S называются гармонически сопряженными, если они делят отрезок IN в одном и том же отношении, причем одна внутренним образом, а другая – внешним образом. Любая точка F отрезка IN , кроме его середины, имеет гармонически сопряженную точку S , впрочем, для середины отрезка гармонически сопряженной можно считать бесконечно удаленную точку. Если F гармонически сопряжена с S (S гармонически сопряжена с F), то



$$\frac{FI}{FN} = \frac{SI}{SN}. \quad (14)$$


Поменяв в (14) местами средние члены, получаем новое отношение, которое фактически означает, что I гармонически сопряжена с N (N гармонически сопряжена с I). Исходя из этого обстоятельства, четверку точек I, N, F, S с условием (14) называют просто гармонической четверкой точек.



Для $\triangle ABC$ точки, именованные в энциклопедии Кимберлинга как $X(1)$ (инцентр I), $X(5)$ (окружность девяти точек N), $X(11)$ (точка Фейербаха F) и $X(12)$ (S), являются гармонической четверкой. Иными словами, справедливо следующее утверждение: “Центры I, N, F и S $\triangle ABC$ лежат на одной прямой и являются гармонически сопряженной четверкой точек”.


Прежде, чем говорить о гармонической четверке центров $\triangle ABC$, покажем как в общем случае к любым трем точкам N, I и F на одной прямой добавить еще одну точку S этой прямой так, чтобы вместе они составляли гармонически сопряженную четверку. Как это можно сделать, описано ниже в пунктах 1-6. Результат построения представлен изображением рис. 12a, которое одновременно является и моделью для экспериментов с четверкой гармонически сопряженных точек. На модели точки N, I и F , а также A, B и C являются независимыми. Итак, точку S можно строить так:

1) блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментом “ Точка” создадим две точки и переименуем их на N и F . Инструментом “ Прямая” через точки N и F проведем прямую a и изменим стиль ее вывода. Инструментом “ Точка на объекте” зафиксируем на a еще одну точку и переименуем ее на I ;

2) инструментом “ Прямая” через точку F проведем произвольную прямую FA , а через точку N – две произвольные прямые NB и NC . Инструментом “ Точка на объекте” зафиксируем точки D и E пересечения прямой FA соответственно с прямыми NB и NC ;

4) Инструментом “ Точка на объекте” зафиксируем точку G , как пересечение прямых ID и NE , и точку H как пересечение прямых ND и IE ;

5) инструментом “ Прямая” проведем прямую через точки G и H . Инструментом “ Точка на объекте” зафиксируем и переименуем на S точку пересечения прямой GH и исходной прямой a . Требуемая 4-я точка построена;

6) инструментом “ Текст” сформируем в *LaTeX*-режиме проверочную надпись, подтверждающую правильность построенной точки:

$$\frac{SI}{SN} = \frac{FI}{FN} \quad \boxed{SI/SN == FI/FN} .$$

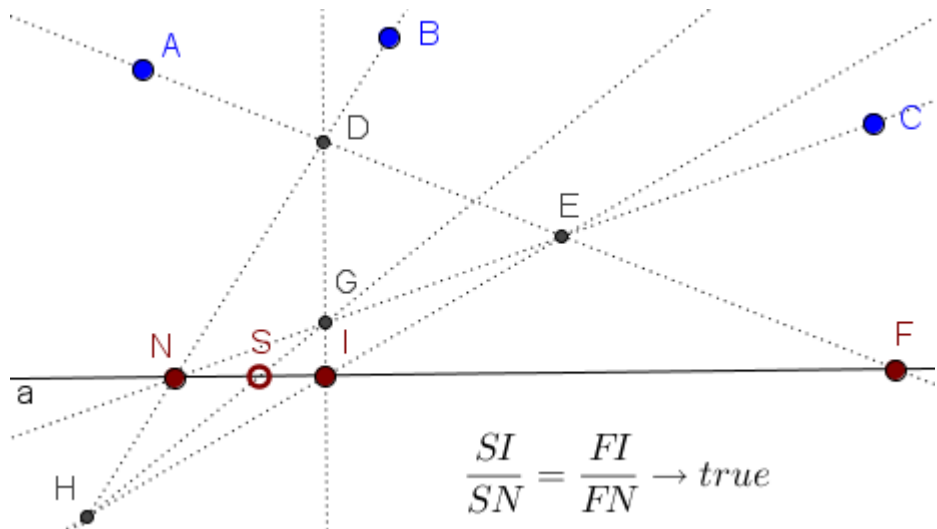





Рис. 12a. Построение 4-й гармонической точки S по 3 заданным точкам N, I, F


Пример 56. Построить модель для экспериментальной проверки утверждения: “Инцентр I , центр окружности девяти точек N , точка Фейербаха F и точка $S=X_{12} \triangle ABC$ лежат на одной прямой и являются гармонически сопряженной четверкой точек.

Решение. Искомая модель, представленная на рис. 12b, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- командами $I=TriangleCenter[A,B,C,1]$, $N=TriangleCenter[A,B,C,5]$, $F=TriangleCenter[A,B,C,11]$ и $S=TriangleCenter[A,B,C,12]$ выведем точки I , N , F и S ;

- инструментом “ Прямая” проведем прямую через точки N и I . Она получит имя g ;

- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 56:

$$F \text{ на прямой } NI \rightarrow \boxed{AreCollinera[F,N,I]} ,$$

$$S \text{ на прямой } NI \rightarrow \boxed{AreCollinera[S,N,I]} ,$$

$$\frac{FI}{FN} = \frac{SI}{SN} \rightarrow \boxed{FI/FN==SI/SN} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать во всех строках правых вычисляемых частей надписи значения *true*, что и требовалось проверить.

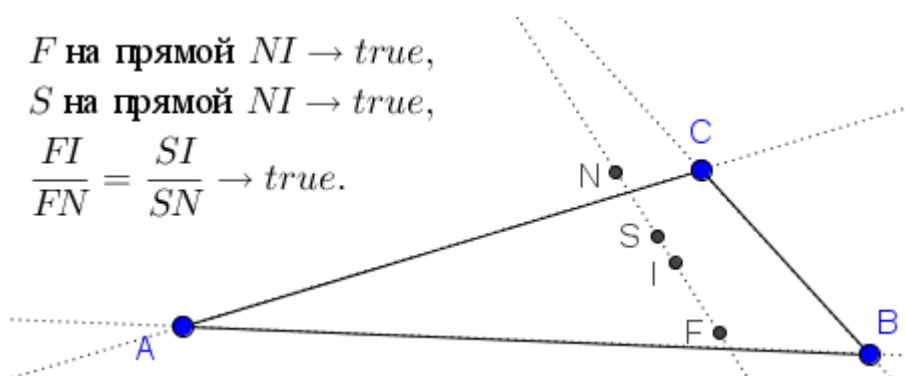


Рис. 12b. Модель для экспериментов с гармонической четверкой центров I , N , S , F

5.14. Точки Ферма

Точка Ферма или, по-другому, *первая точка Ферма* (*Fermat point*, точка Торричелли, 1-я изогональная точка) – это точка T , сумма расстояний от которой до вершин $\triangle ABC$ является минимальной. *Вторая точка Ферма* определяется по-иному (см. ниже). В энциклопедии Кимберлинга первая и вторая точки Ферма именуются соответственно как $X(13)$ и $X(14)$, и для $\triangle ABC$ через строку ввода их можно строить командами $TriangleCenter[A,B,C,n]$ ($n=13, 14$).

Для первой точки Ферма справедливы такие утверждения:

1) *Теорема* (Э. Торричелли, Б. Кавальери, Т. Симпсон, Ф. Хейнен, Ж. Бертран). Построим вне произвольного $\triangle ABC$ на его сторонах равносторонние треугольники ABC' , BCA' , CAB' . Тогда:

а) окружности, описанные вокруг правильных треугольников, и прямые AA' , BB' , CC' пересекаются в одной точке D (см. рис. 13a);

б) центры трех равносторонних треугольников, построенных на сторонах $\triangle ABC$, являются вершинами нового равностороннего треугольника (*теорема Наполеона*). Длины отрезков AA' , BB' и CC' , называемых сегментами Симпсона, равны между собой (см. рис. 13б);

с) если все углы исходного $\triangle ABC$ меньше 120° , то D лежит внутри треугольника и является его точкой Ферма T , стороны треугольника из точки T видны под углами 120° , а длины линий Симпсона равны величине $AT+BT+CT$ (см. рис. 13с).

Замечания. 1. Команда $TriangleCenter[A,B,C,13]$ правильно строит точку Ферма T только для случая, когда углы $\triangle ABC$ не превосходят 120° . В общем случае по ней фактически выводится не точка T , а точка D (см. 1а).

2. Если все углы $\triangle ABC$ меньше 120° , то точку Ферма T в соответствии с теоремой Торричелли можно находить, как пересечение любых двух из шести кривых – трех отрезков и трех окружностей. Если один из углов $\triangle ABC$ больше или равен 120° , то точка Ферма T совпадает с вершиной **тупого угла**.

Для второй точки Ферма справедливы такие утверждения:

2) Построим внутрь произвольного $\triangle ABC$ на его сторонах равносторонние треугольники ABC' , BCA' , CAB' . Тогда:

а) окружности, описанные вокруг правильных треугольников, и прямые AA' , BB' , CC' пересекаются в одной точке D , которая и называется второй точкой Ферма (см. рис. 13д);

б) текст как в 1б (см. рис. 13е);

с) точка D лежит вне $\triangle ABC$ или совпадает с одной из его вершин. Стороны треугольника из точки D видны под углами 120° или 60° . Если D не лежит на отрезках AA' , BB' и CC' , то

$$AA' = \text{abs}(2\max(\{DA, DB, DC\}) - DA - DB - DC). \quad (15)$$

Кроме того, справедливы соотношения:

$$\begin{aligned} DA' + DB + DC &= 2\max(\{DA', DB, DC\}), \\ DB' + DA + DC &= 2\max(\{DB', DA, DC\}), \\ DC' + DA + DB &= 2\max(\{DC', DA, DB\}). \end{aligned} \quad (16)$$

Замечания. Каждое из соотношений (16) означает, что длина максимального из отрезков равна сумме длин двух других отрезков (аналог суммы расстояний от точки Фейербаха F до вершин срединного треугольника). Далее, соотношения (15) и (16) установлены экспериментально с помощью *GeoGebra* и потому нуждаются в строгом доказательстве.



д) *Теорема Лестера*. Пусть в $\triangle ABC$ X_{13} – первая точка Ферма, X_{14} – вторая точка Ферма, X_5 – центр окружности девяти точек и X_3 – центр описанной


окружности. Точки X_{14} , X_{13} , X_5 и X_3 лежат на одной окружности, называемой окружностью Лестера. Заметим, что в случае равностороннего $\triangle ABC$ окружность Лестера вырождается в прямую, то есть имеет радиус, равный ∞ .



е) точки Ферма X_{13} и X_{14} находятся на гиперболе Киперта (см. примеры 48, 54).


Пример 57. Построить модель для экспериментальной проверки утверждения 1а.


Решение. Изображение, представленное на рис. 13а, создавалось следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;

- инструментом “ Правильный многоугольник” на сторонах $\triangle ABC$ сформируем правильные треугольники во внешние от $\triangle ABC$ направления. Через контекстное меню заменим имена вершин D , E и F на C' , A' и B' , (см. рис. 13а);

- инструментом “ Прямая” проведем прямые через пары точек A и A' , B и B' , C и C' и изменим стиль их вывода. Инструментом “ Окружность по трем точкам” проведем окружности ABC' , BCA' и CAB' . Они получают имена e , d_1 и c_1 ;

- инструментом “ Точка на объекте” сформируем точку D как пересечение трех построенных прямых и трех окружностей. Сделать это удастся, то есть в данном конкретном случае перечисленные объекты действительно пересекаются в одной точке;

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую, что и в иных ситуациях упомянутые три прямых и три окружности действительно пересекаются в одной точке:

$$\begin{aligned}
 D \text{ лежит на } AA' &\rightarrow \boxed{AreCollinear[D,A,A']} , \\
 D \text{ лежит на } BB' &\rightarrow \boxed{AreCollinear[D,B,B']} , \\
 D \text{ лежит на } CC' &\rightarrow \boxed{AreCollinear[D,C,C']} , \\
 D \text{ на окружности } ABC' &\rightarrow \boxed{D==Intersect[e,D]} , \\
 D \text{ на окружности } ACB' &\rightarrow \boxed{D==Intersect[d_1,D]} , \\
 D \text{ на окружности } BCA' &\rightarrow \boxed{D==Intersect[c_1,D]} .
 \end{aligned}$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

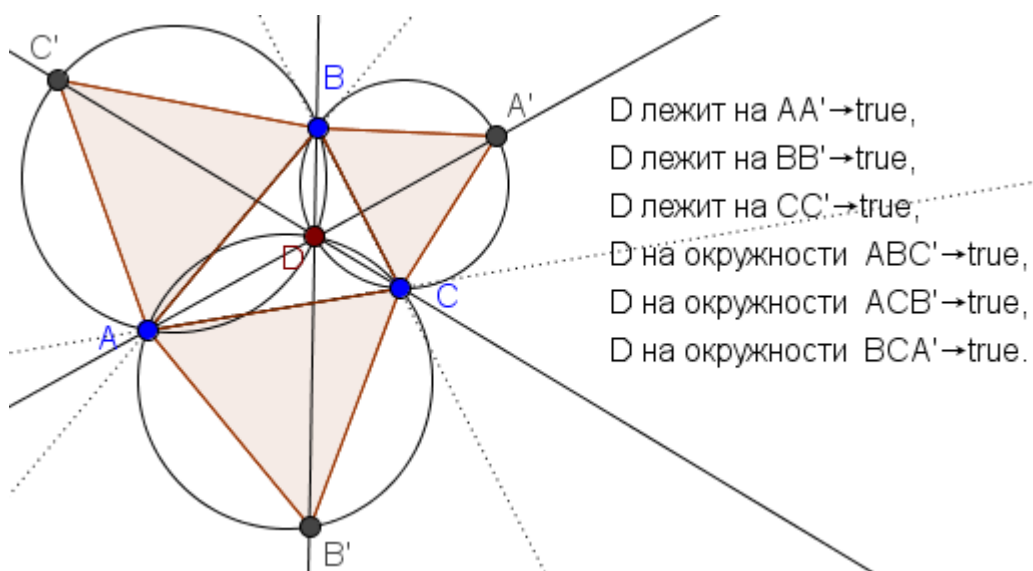



Рис. 13a. Модель для экспериментальной проверки утверждения примера 57

Пример 58. Построить модель для экспериментальной проверки утверждения 1b.

Решение. Изображение, представленное на рис. 13b, строилось следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 13a. Удалим проверочную надпись и линии AA' , BB' , CC' ;
- командами $P = \text{Center}[e]$, $Q = \text{Center}[c_1]$, $R = \text{Center}[d_1]$ выведем центры P , Q и R окружностей. Инструментом “ Отрезок” проведем стороны $\triangle PQR$ и поставим на них одинаковые декоративные риски (не обязательно);

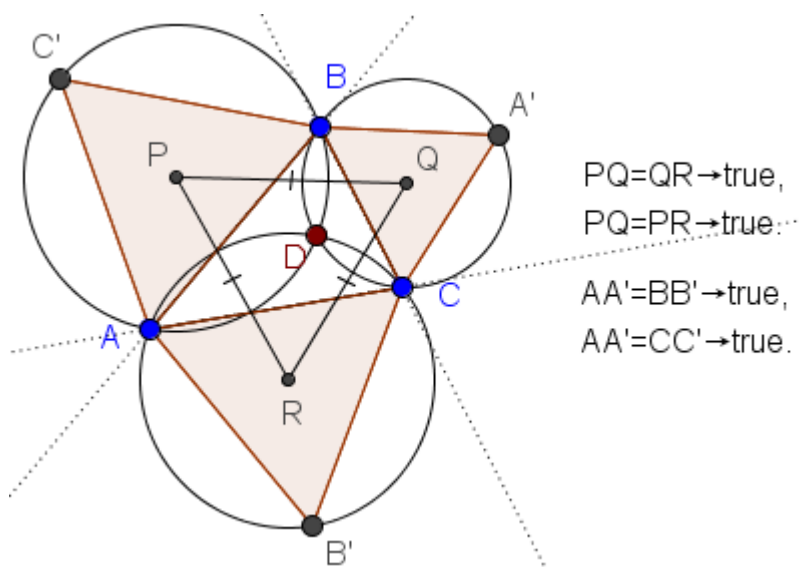



Рис. 13b. Модель для экспериментальной проверки утверждений примера 58



- инструментом “ Текст” сформируем две проверочные надписи, подтверждающие или опровергающие соответственно утверждение теоремы Наполеона и равенство длин сегментов Симпсона:

$$\begin{aligned}
PQ=QR &\rightarrow \boxed{PQ==QR} , \\
PQ=PR &\rightarrow \boxed{PQ==PR} , \\
AA'=BB' &\rightarrow \boxed{AA'=BB'} , \\
AA'=CC' &\rightarrow \boxed{AA'=CC'} .
\end{aligned}$$


Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки обеих надписей значения *true*, что и требовалось проверить.

Пример 59. Построить модель для экспериментальной проверки утверждения 1с.

Решение. Изображение, представленное на рис. 13с, строилось следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 13а. Удалим проверочную надпись;
- командой $T=TriangleCenter[A,B,C,13]$ сформируем точку Ферма T . Инструментом “ Точка” сформируем в любом месте точку E ;
- инструментом “ Текст” создадим информационную надпись, по которой всегда можно будет видеть текущие значения $\angle ABC$, $\angle BCA$ и $\angle CAB$:

$$\angle ABC = \boxed{Angle[A,B,C]} , \angle BCA = \boxed{Angle[B,C,A]} , \angle CAB = \boxed{Angle[C,A,B]} .$$

- инструментом “ Текст” сформируем проверочную надпись подтверждающую или опровергающую все утверждения примера 59:

Если все углы $< 120^\circ$,

то все утверждения $\rightarrow true$:

D находится внутри $\triangle ABC \rightarrow$

$$\boxed{IsInRegion[D, Polygon[A,B,C]]} ,$$

$$D=T \rightarrow \boxed{D==T} ,$$

$$TA+TB+TC < EA+EB+EC \rightarrow \boxed{TA+TB+TC < EA+EB+EC} ,$$

$$AA'=TA+TB+TC \rightarrow \boxed{AA'==TA+TB+TC} ,$$

$$\angle BDA = 120^\circ \rightarrow \boxed{Angle[B,D,A]} ,$$

$$\angle CDB = 120^\circ \rightarrow \boxed{Angle[C,D,B]} ,$$

$$\angle ADC = 120^\circ \rightarrow \boxed{Angle[A,D,C]} .$$

Перемещая на построенной модели независимые точки A, B, C и E по плоскости так, чтобы все углы исходного $\triangle ABC$ оставались меньше 120° , мы всегда будем получать в правых вычисляемых частях последних 6 строк второй надписи значения *true*, что и требовалось проверить.

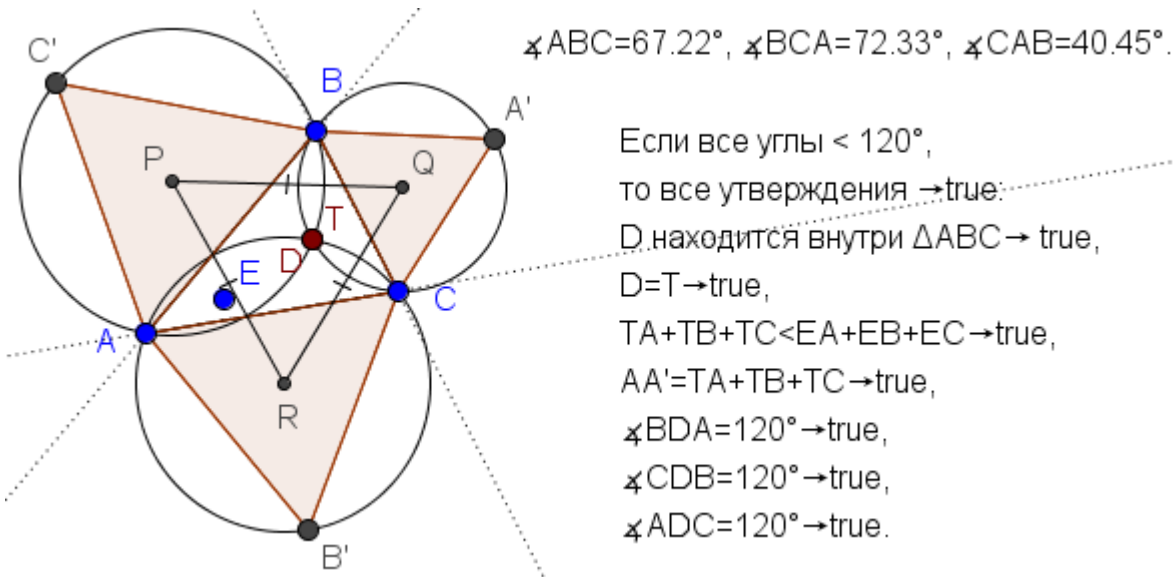


Рис. 13с. Модель для экспериментальной проверки утверждений примера 59

Пример 60. Построить модель для экспериментальной проверки утверждения 2а.

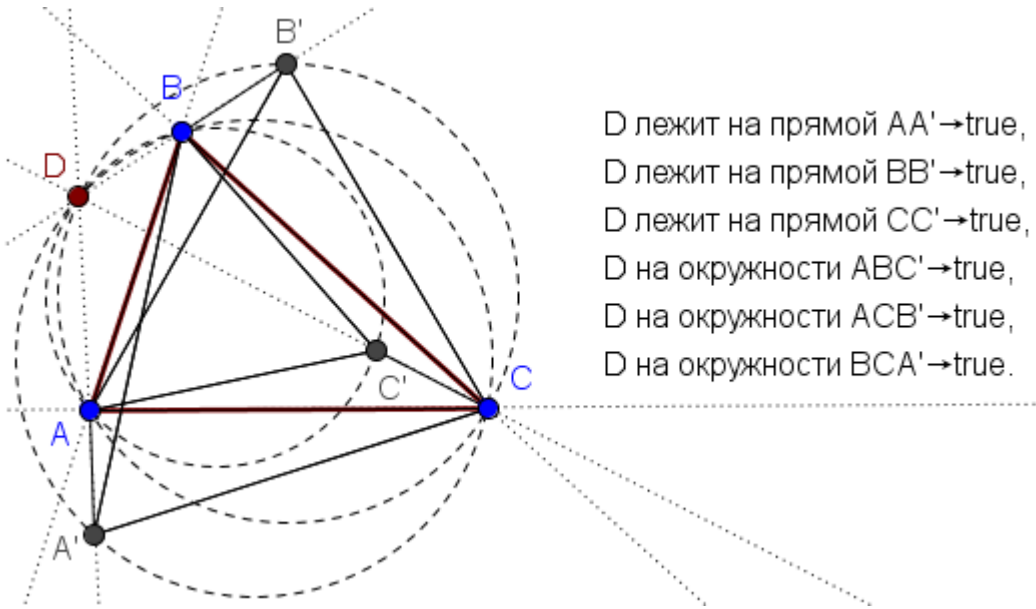


Рис. 13d. Модель для экспериментальной проверки утверждений примера 60

Решение. Изображение, представленное на рис. 13d, строилось аналогично изображению рис. 13a примера 57, но правильные треугольники формировались не по внешнему направлению от $\triangle ABC$, а по внутреннему. Перемещая на построенной модели независимые точки A, B и C по плоскости, мы все-

гда будем получать в правых вычисляемых частях каждой строки надписи значения *true*. Это означает, что три прямые и три окружности во всех случаях пересекаются в одной точке, что и требовалось проверить.

Пример 61. Построить модель для экспериментальной проверки утверждения 2*b*.

Решение. Изображение, представленное на рис. 13*e*, строилось аналогично изображению рис. 13*b* примера 58, но правильные треугольники формировались не по внешнему направлению от ΔABC , а по внутреннему. Перемещая на построенной модели независимые точки *A*, *B* и *C* по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки обеих надписей значения *true*, что и требовалось проверить.

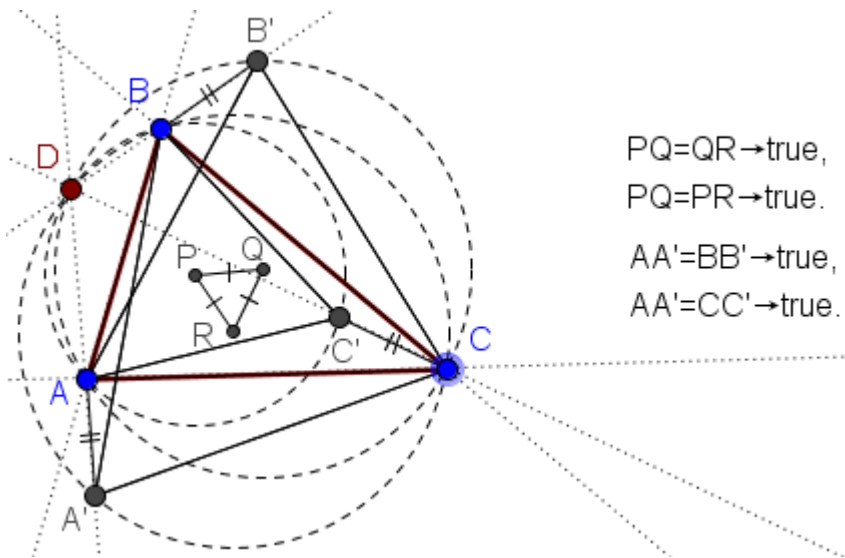


Рис. 13*e*. Модель для экспериментальной проверки утверждений примера 61

Пример 62. Построить модель для экспериментальной проверки утверждений 2*c*.

Решение. Изображение, представленное на рис. 13*f*1, строилось следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 13*e*, удалив с него проверочную надпись и ΔPQR ;
- инструментом “ABC Текст” сформируем новую проверочную надпись, подтверждающую или опровергающую все утверждения примера 62:

$$\begin{aligned}
 & \text{2-я точка Ферма } D \notin \Delta ABC \rightarrow \boxed{\neg IsInRegion[D, Polygon[A,B,C]]} \;, \\
 & \angle BDA = \boxed{If[Angle[B,D,A]<180^\circ, Angle[B,D,A], Angle[A,D,B]]} \;, \\
 & \angle CDB = \boxed{If[Angle[C,D,B]<180^\circ, Angle[C,D,B], Angle[B,D,C]]} \;, \\
 & \angle ADC = \boxed{If[Angle[A,D,C]<180^\circ, Angle[A,D,C], Angle[C,D,A]]} \;, \\
 & AA' = abs(2max(\{DA,DB,DC\})-DA-DB-DC) \rightarrow
 \end{aligned}$$

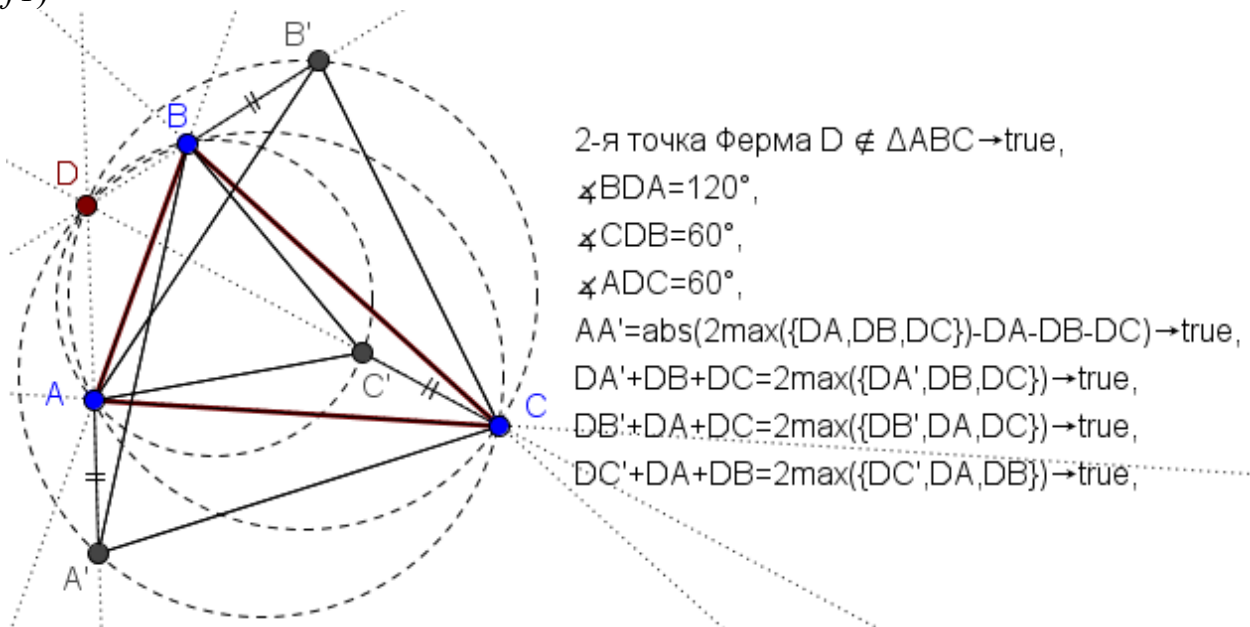
$$AA' == \text{abs}(2\text{Max}(\{DA, DB, DC\}) - DA - DB - DC) \quad ,$$

$$DA' + DB + DC = 2\text{max}(\{DA', DB, DC\}) \rightarrow \boxed{DA' + DB + DC == 2\text{Max}(\{DA', DB, DC\})} \quad ,$$

$$DB' + DA + DC = 2\text{max}(\{DB', DA, DC\}) \rightarrow \boxed{DB' + DA + DC == 2\text{Max}(\{DB', DA, DC\})} \quad ,$$

$$DC' + DA + DB = 2\text{max}(\{DC', DA, DB\}) \rightarrow \boxed{DC' + DA + DB == 2\text{Max}(\{DC', DA, DB\})} \quad .$$

f1)



f2)

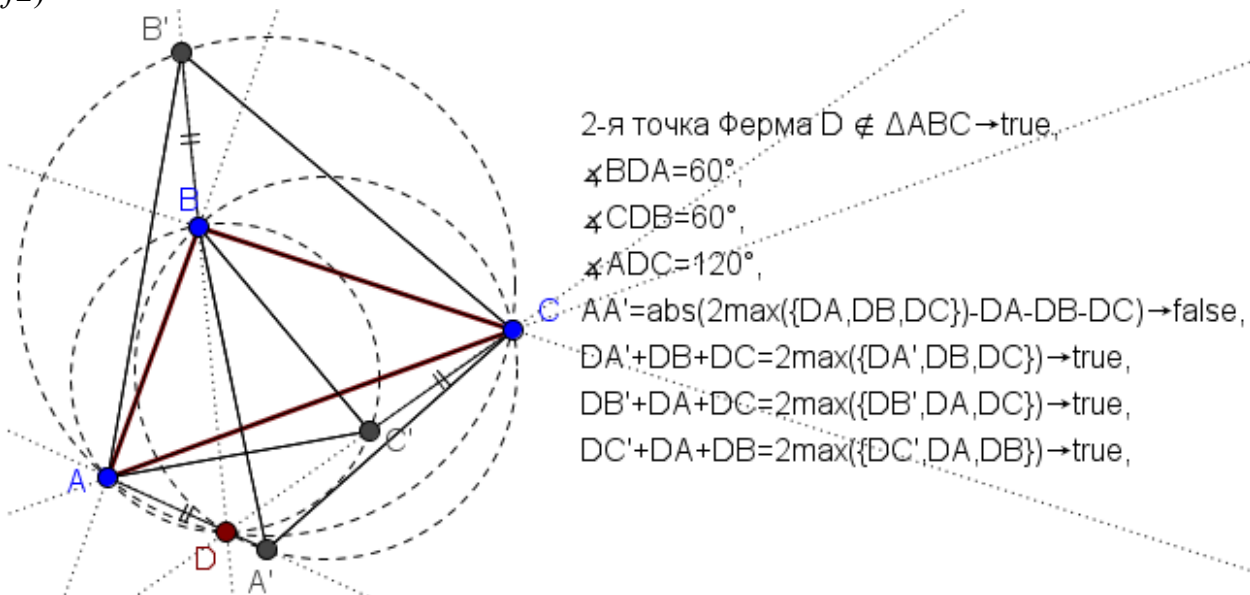


Рис. 13 (f1, f2). Модель для экспериментальной проверки утверждений примера 62


Перемещая на построенной модели независимые точки A , B и C по плоскости, мы будем получать в правой вычисляемой части первой строки надписи значение *true* во всех случаях, в которых точка D не совпадает с одной из вершин ΔABC . В трех следующих строках правые части всегда будут иметь значения 60° или 120° . В пятой строке правая часть будет равна *true* во всех тех

случаях, когда точка D не оказывается на отрезках AA' , BB' и CC' . И, наконец, в правых частях последних трех строк надписи всегда будем иметь значение *true*. Но ведь именно все это нам и надо было проверить.

Изображение на рис. 13f2 – это просто один их динамических кадров построенной модели, демонстрирующий особенности нахождения второй точки Ферма на любом из равных по длине отрезков AA' , BB' , или CC' . В данном случае длину этих отрезков по формуле $AA' = \text{abs}(2\max(\{DA, DB, DC\}) - DA - DB - DC)$ вычислять нельзя.

Пример 63. Построить модель для экспериментальной проверки утверждения 2d, то есть теоремы Лестера.

Решение. Изображение, представленное на рис. 13g, строилось следующим образом:

- продолжим построение модели, начиная с изображения, представленного на рис. 13d. Удалим проверочную надпись;
- командами $X_{14} = \text{TriangleCenter}[A, B, C, 14]$, $X_{13} = \text{TriangleCenter}[A, B, C, 13]$, $X_5 = \text{TriangleCenter}[A, B, C, 5]$ и $X_3 = \text{TriangleCenter}[A, B, C, 3]$ сформируем соответственно вторую точку Ферма X_{14} , первую точку Ферма X_{13} , центр окружности девяти точек X_5 и центр описанной окружности X_3 ;
- инструментом “ Окружность по трем точкам” проведем окружность через точки X_{13} , X_5 и X_3 . Она получит имя c_1 ;

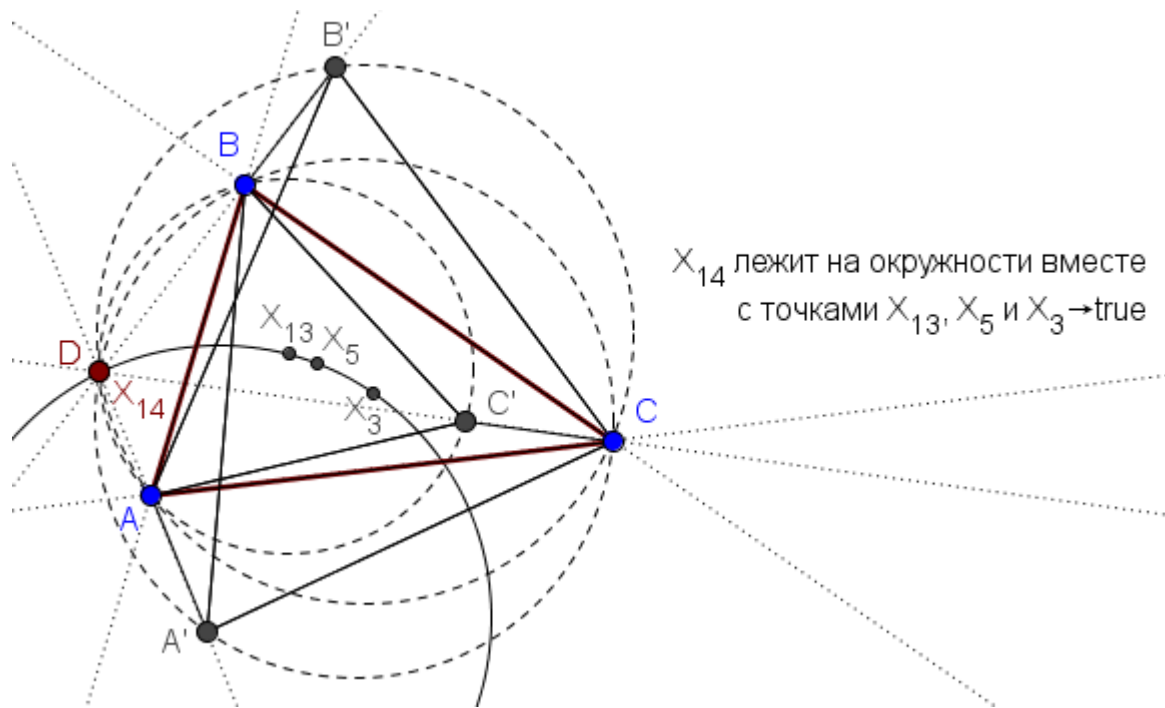



Рис. 13g. Модель для экспериментальной проверки утверждений примера 63

- инструментом “ Текст” сформируем проверочную надпись подтверждающую или опровергающую утверждения примера 63:




X_{14} лежит на окружности вместе

с точками X_{13} , X_5 и $X_3 \rightarrow \boxed{X_{14} == \text{Intersect}[c_1, X_{14}]}$.

Перемещая на построенной модели независимые точки A , B , C и E по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

Пример 64. Построить модель для экспериментальной проверки утверждения $2e$, то есть того, что “Точки Ферма X_{13} и X_{14} находятся на гиперболе Киперта”.

Решение. Ранее в примере 48 мы уже строили гиперболу Киперта по трем вершинам $\triangle ABC$, его центроиду X_2 и ортоцентру X_4 . Там же проверили, что центр Шпикера также находится на гиперболе Киперта. Данный пример решается аналогично. Искомая модель, представленная с левой стороны рис. 13h, создавалась следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $X_4 = \text{TriangleCenter}[A, B, C, 4]$, $X_2 = \text{TriangleCenter}[A, B, C, 2]$, $X_{10} = \text{TriangleCenter}[A, B, C, 10]$, $X_{13} = \text{TriangleCenter}[A, B, C, 13]$ и $X_{14} = \text{TriangleCenter}[A, B, C, 14]$ для $\triangle ABC$ построим центроид X_2 , ортоцентр X_4 , центр Шпикера X_{10} , первую точку Ферма X_{13} и вторую точку Ферма X_{14} ;
- инструментом “ Коника по пяти точкам” проведем гиперболу Киперта через точки A , B , C , X_2 и X_4 . Она получит имя p . Сделаем его видимым;

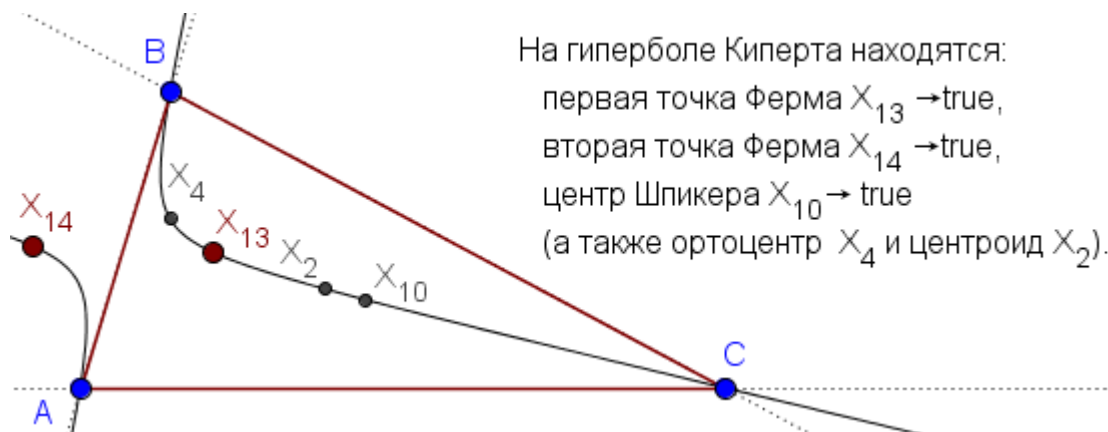



Рис. 13h. Модель для экспериментальной проверки утверждений примера 64

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 64:

На гиперболе Киперта находятся:

первая точка Ферма $X_{13} \rightarrow \boxed{X_{13} == \text{Intersect}[p, X_{13}]}$,

вторая точка Ферма $X_{14} \rightarrow \boxed{X_{14} == \text{Intersect}[p, X_{14}]}$,

центр Шпикера $X_{\{10\}} \rightarrow \boxed{X_{\{10\}} == \text{Intersect}[p, X_{\{10\}}]}$,
(а также ортоцентр $X_{\{4\}}$ и центроид $X_{\{2\}}$) .

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

5.15. Точки Аполлония

Точки Аполлония – это точки, расстояния от которых до вершин треугольника обратно пропорциональны длинам противоположных сторон. Любой треугольник, отличный от равностороннего, имеет две точки Аполлония. В энциклопедии Кимберлинга первая и вторая точки Аполлония именуются соответственно как $X(15)$ и $X(16)$, и для $\triangle ABC$ через строку ввода их можно строить командами $\text{TriangleCenter}[A,B,C,n]$ ($n=15, 16$).

Справедливы следующие утверждения:

1) Пусть в $\triangle ABC$ соединены отрезком основания биссектрис внутреннего и внешнего углов для одной вершины. Построим на этом отрезке окружность, как на диаметре. Точки пересечения трех таких окружностей, то есть концевые точки их общей хорды, и являются точками Аполлония. Это утверждение можно использовать для построения точек Аполлония.

2) Прямая, соединяющая точки Аполлония X_{15} и X_{16} , называется осью Брокара. На ней находятся и центр описанной окружности X_3 , и точка Лемуана X_6 .

3) *Соотношения между точками X_{15} , X_{16} , X_3 и X_6 оси Брокара.* Пусть R – радиус описанной окружности для $\triangle ABC$ и k – вспомогательный параметр:

$k = \sqrt{1 - \frac{(X_3 X_6)^2}{R^2}}$. Тогда имеют место следующие соотношения:

$$X_{15} = \frac{1}{1+k}(X_6 + k \cdot X_3), \quad X_{16} = \frac{1}{1-k}(X_6 - k \cdot X_3), \quad (17)$$

$$\frac{(X_6 X_n)^2}{(X_3 X_n)^2} + \frac{(X_3 X_6)^2}{R^2} = 1 \quad (n=15, 16). \quad (18)$$

4) Точки Аполлония изогонально сопряжены с точками Ферма: X_{15} с X_{13} и X_{16} с X_{14} .

Напомним конструктивное определение изогонального сопряжения точек P и Q по отношению к $\triangle ABC$. Говорят, что точка Q изогонально сопряжена P , если Q по P можно построить так:

- проведем прямые a , b и c через P и вершины треугольника, то есть через пары точек A и P , B и P , C и P ;
- проведем через P биссектрисы внутренних углов треугольника;










- построим прямые e, f и g , симметричные прямым a, b и c относительно биссектрис;
- можно показать, что прямые e, f и g пересекаются в одной точке. Обозначим ее через Q и считаем изогонально сопряженной P .

Ясно, что если Q изогонально сопряжена P , то P изогонально сопряжена Q , то есть можно говорить о паре изогонально сопряженных точек P и Q .

Замечание. Кроме X_{15}, X_{13} и X_{16}, X_{14} известны и другие пары изогонально сопряженных точек – центр описанной окружности и ортоцентр, центроид и точка Лемуана и т. д.

Пример 65. Построить модель для экспериментальной проверки утверждения 1.

Решение. Изображение, представленное на рис. 14а, строилось следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Биссектриса угла” проведем биссектрисы внутренних и внешних углов $\triangle ABC$;
- инструментом “ Точка на объекте” отметим по две точки пересечения каждой стороны треугольника или ее продолжения с внутренней и внешней биссектрисой, исходящих из одного угла: на стороне AC – D и E , на стороне BC – F и G , на стороне AB – H и I (см. рис.);
- инструментами “ Середина или центр” сформируем средние точки точки между D и E, F и G, H и I . Инструментом “ Окружность по центру и точке” на отрезках DE, FG и HI , как на диаметрах, построим окружности;
- инструментом “ Точка на объекте” отметим две точки N и M пересечения трех окружностей. Инструментом “ Отрезок” проведем общую хорду для трех окружностей (не обязательно);
- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 65. Нам требуется проверить, что точки N и M удовлетворяют определению точек Аполлония. Дополнительно командами $X_{15}=TriangleCenter[A,B,C,15]$ и $X_{16}=TriangleCenter[A,B,C,16]$ выведем точки Аполлония и проверим, что $X_{15}=M$ и $X_{16}=N$. Учитывая сказанное, надпись можно сформировать так:

$$\begin{array}{ll}
 NC \cdot AB = NB \cdot AC \rightarrow & \boxed{NC * AB == NB * AC} \text{ ,} \\
 NC \cdot AB = NA \cdot BC \rightarrow & \boxed{NC * AB == NA * BC} \text{ ,} \\
 MC \cdot AB = MB \cdot AC \rightarrow & \boxed{MC * AB == MB * AC} \text{ ,} \\
 MC \cdot AB = MA \cdot BC \rightarrow & \boxed{MC * AB == MA * BC} \text{ ,}
 \end{array}$$

$$M=X_{15} \vee N=X_{15} \rightarrow \boxed{M==X_{15} \vee N==X_{15}} ,$$

$$M=X_{16} \vee N=X_{16} \rightarrow \boxed{M==X_{16} \vee N==X_{16}} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

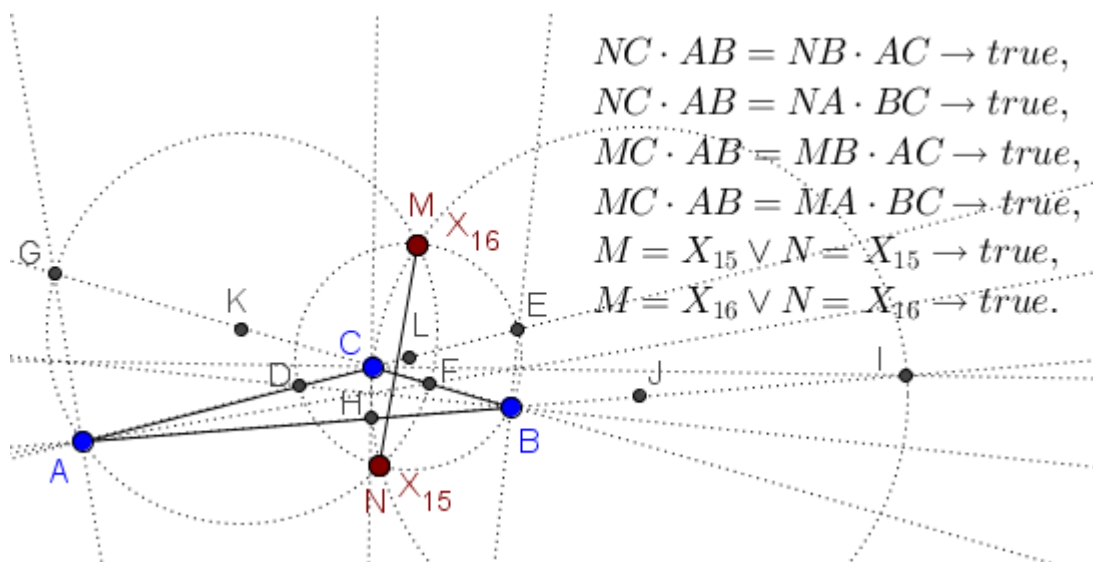


Рис. 14а. Модель для экспериментальной проверки утверждений примера 65

Пример 66. Построить модель для экспериментальной проверки утверждения 2: “Точки Аполлония X_{15} и X_{16} , центр описанной окружности X_3 и точка Лемуана X_6 находятся на одной прямой”, называемой осью Брокара.

Решение. Изображение, представленное на рис. 14b, строилось следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $X_{15}=TriangleCenter[A,B,C,15]$, $X_{16}=TriangleCenter[A,B,C,16]$, $X_3=TriangleCenter[A,B,C,3]$, $X_6=TriangleCenter[A,B,C,6]$, выведем точки X_{15} , X_{16} , X_3 и X_6 ;
- инструментами “ Прямая” проведем прямую через точки X_{15} и X_{16} ;
- инструментом “ Текст” сформируем проверочную надпись подтверждающую или опровергающую утверждение примера 66:

$$X_3 \text{ лежит на прямой Брокара} \rightarrow \boxed{AreCollinear[X_3, X_{16}, X_{15}]} ,$$

$$X_6 \text{ лежит на прямой Брокара} \rightarrow \boxed{AreCollinear[X_6, X_{16}, X_{15}]} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

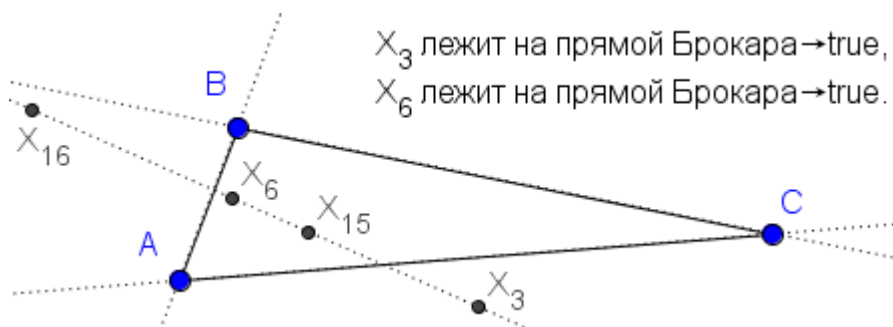


Рис. 14b. Модель для экспериментальной проверки утверждений примера 66

Пример 67. Построить модель для экспериментальной проверки соотношений (17)-(18) для точек X_{15} , X_{16} , X_3 и X_6 оси Брокара.

Решение. Изображение, представленное на рис. 14b, строилось следующим образом:

- продолжим построение модели, начиная с изображения рис. 14b. Удалим проверочную надпись;
- через строку ввода сформируем вспомогательную переменную:

$$k = \sqrt{1 - \frac{\text{Distance}[X_3, X_6]^2}{\text{Distance}[X_3, A]^2}};$$

- инструментом “ Текст” в *LaTeX*-режиме сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 67:

$$k = \sqrt{1 - \frac{(X_3 X_6)^2}{R^2}} =$$

$$\sqrt{1 - \frac{\text{Distance}[X_3, X_6]^2}{\text{Distance}[X_3, A]^2}}, \backslash$$

$$X_{15} = \frac{X_6 + k \cdot X_3}{1 + k} \rightarrow$$

$$X_{15} == (X_6 + k * X_3) / (1 + k), \backslash$$

$$X_{16} = \frac{X_6 - k \cdot X_3}{1 - k} \rightarrow$$

$$X_{16} == (X_6 - k * X_3) / (1 - k), \backslash$$

$$\frac{(X_6 X_{15})^2}{(X_3 X_{15})^2} + \frac{(X_3 X_6)^2}{R^2} = 1 \rightarrow$$

$$\frac{\text{Distance}[X_6, X_{15}]^2 / \text{Distance}[X_3, X_{15}]^2 + \text{Distance}[X_3, X_6]^2 / \text{Distance}[X_3, A]^2 == 1}, \backslash$$

$$\frac{(X_6 X_{16})^2}{(X_3 X_{16})^2} + \frac{(X_3 X_6)^2}{R^2} = 1 \rightarrow$$

$$\frac{\text{Distance}[X_6, X_{16}]^2 / \text{Distance}[X_3, X_{16}]^2 + \text{Distance}[X_3, X_6]^2 / \text{Distance}[X_3, A]^2 == 1}.$$

Перемещая на построенной модели независимые точки A, B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

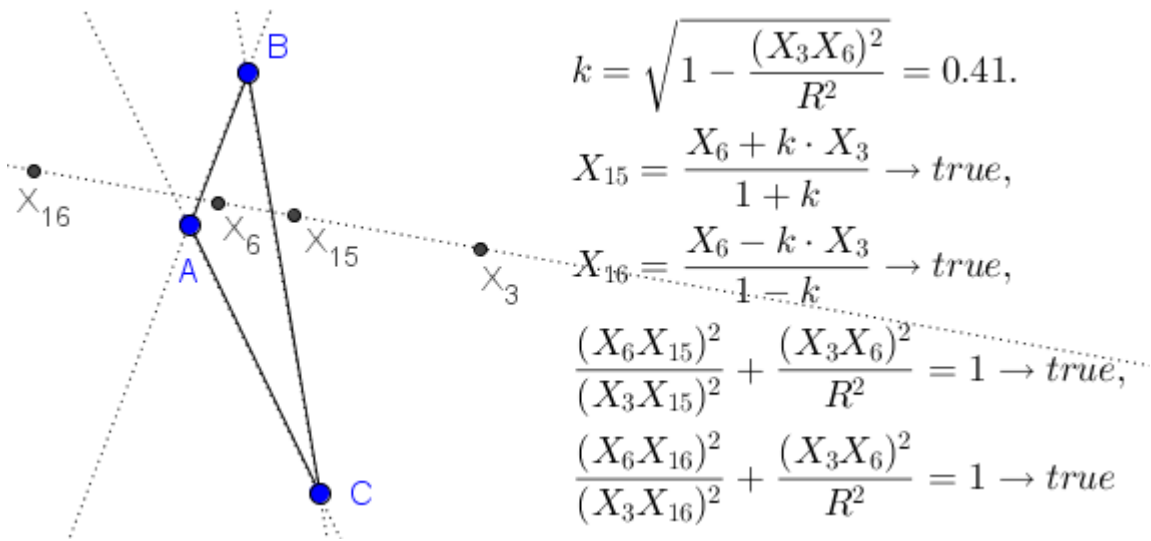


Рис. 14с. Модель для экспериментальной проверки утверждений примера 67

Пример 68. Построить модель для экспериментальной проверки утверждения: “Точки Аполлония изогонально сопряжены с точками Ферма: X_{15} с X_{13} и X_{16} с X_{14} ”.

Решение. Ограничимся построением модели для проверки того, что X_{15} с X_{13} изогонально сопряжены. Изображение, представленное на рис. 14d1 (или 14d2), строилось следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командой $X_{\{15\}} = TriangleCenter[A,B,C,15]$ выведем точку Аполлония X_{15} и начнем строить для нее изогонально сопряженную точку I ;
- инструментом “ Прямая” проведем прямые через X_{15} и вершины треугольника, то есть через пары точек A и X_{15} , B и X_{15} , C и X_{15} . Инструментом “ Биссектриса угла” проведем биссектрисы внутренних углов треугольника. Инструментом “ Отражение относительно прямой” построим точки E, F, D , симметричные X_{15} относительно биссектрис;
- инструментом “ Прямая” построим прямые m, n и p , симметричные относительно биссектрис прямым, проходящим через X_{15} (см. рис. 13d1, 13d2). Инструментом “ Точка на объекте” сформируем точку пересечения проведенных прямых – I ;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 68:

$$m, n \text{ и } p \rightarrow \text{пересекаются в одной точке} \rightarrow \boxed{AreConcurrent[m,n,p]} ,$$

В энциклопедии Кимберлинга первая и вторая точки Наполеона именуются соответственно как $X(17)$ и $X(18)$, и для $\triangle ABC$ через строку ввода их можно построить командами $TriangleCenter[A,B,C,n]$ ($n=17, 18$).

Справедливы следующие утверждения:

1) *Теорема Наполеона*. Треугольники Наполеона являются правильными.
 2) Прямая, проходящая через точки Наполеона, и прямая, проходящая через точки Ферма, пересекаются в точке Лемуана (точка пересечения симедиан).

3) Точки Наполеона лежат на гиперболе Киперта.







4) Абсолютная величина разности площадей внешнего и внутреннего треугольников Наполеона равна площади S исходного $\triangle ABC$.

5) Если n, m – длины сторон внешнего и внутреннего треугольников Наполеона, a, b и c – длины сторон $\triangle ABC$, то имеют место соотношения:

$$n^2 + m^2 = \frac{a^2 + b^2 + c^2}{3}, \tag{19}$$

$$n^2 = \frac{a^2 + b^2 + c^2}{6} + \frac{2S}{\sqrt{3}}, \quad m^2 = \frac{a^2 + b^2 + c^2}{6} - \frac{2S}{\sqrt{3}}. \tag{20}$$

Построение первой точки Наполеона. Модель для экспериментов с первой точкой Наполеона представлена на рис. 15а. Создавалась она в соответствии с определением данной точки следующим образом:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- инструментом “ Правильный многоугольник” на сторонах $\triangle ABC$ сформируем правильные треугольники $\triangle ABD$, $\triangle BCE$ и $\triangle CAF$ во внешние от $\triangle ABC$ направления. Поставим одинаковые метки на равные стороны треугольников. Командами $TriangleCenter[X,Y,Z,2]$ выведем центроиды G, H и I правильных треугольников;
- инструментом “ Прямая” проведем прямые через пары точек A и H , B и I , C и G . Они получают имена q, r и s . Инструментом “ Точка на объекте” сформируем точку пересечения J проведенных прямых. Командой $X_{17} = TriangleCenter[A,B,C,17]$ выведем точку X_{17} ;
- инструментом “ Текст” сформируем проверочную надпись, подтверждающую, что прямые, проведенные через пары точек A и H , B и I , C и G во всех случаях действительно пересекаются в одной точке, и что она совпадает с точкой X_{17} :

Прямые AH, BI и CG
 пересекаются в одной точке \rightarrow $AreConcurrent[q,r,s]$,
 $J=X_{17} \rightarrow$ $J==X_{17}$.

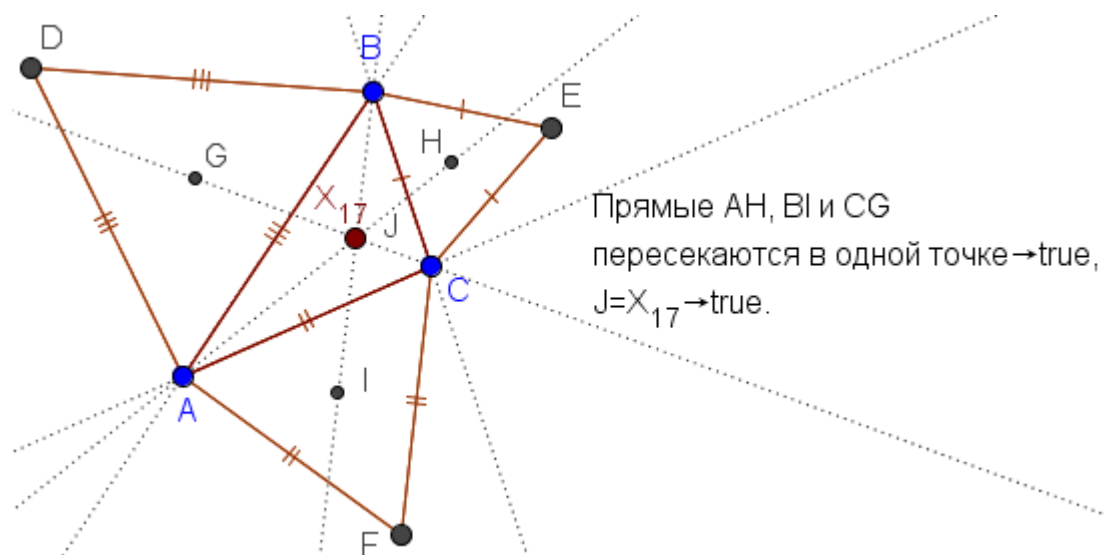


Рис. 15a. Модель для экспериментов с первой точкой Наполеона

Модель для экспериментов со второй точкой Наполеона представлена на рис. 15b. Создавалась она аналогично модели рис. 15a.

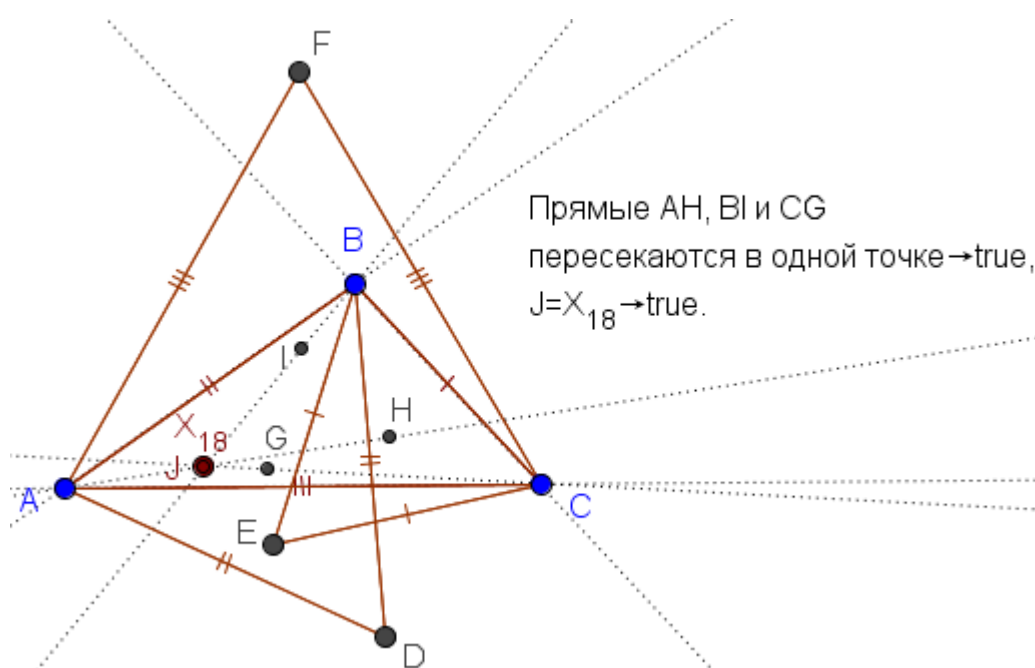







Рис. 15b. Модель для экспериментов со второй точкой Наполеона

Пример 69. Построить модель для экспериментальной проверки утверждения: “Прямые, проходящие соответственно через точки Наполеона X_{17} и X_{18} и точки Ферма X_{13} и X_{14} , пересекаются в точке Лемуана X_6 ”.

Решение. Сформируем изображение, представленное на рис. 15с:

- блокируем вывод обозначений объектов, кроме вывода имен точек. Инструментами “ Прямая” и “ Отрезок” построим $\triangle ABC$;
- командами $X_{\{n\}} = \text{TriangleCenter}[A, B, C, n]$ ($n=17, 18, 13, 14$) выведем требуемые нам центры $\triangle ABC$;

- инструментом “ Прямая” проведем прямые через пары точек X_{17}, X_{18} и X_{13}, X_{14} . Инструментом “ Точка на объекте” сформируем точку пересечения D проведенных прямых. Командой $X_6=TriangleCenter[A,B,C,6]$ выведем точку Лемуана $\triangle ABC$;

- инструментом “ Текст” сформируем проверочную надпись, подтверждающую или опровергающую утверждение примера 69:

$$D=X_6 \rightarrow \boxed{D==X_{\{6\}}} .$$

Перемещая на построенной модели независимые точки A, B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

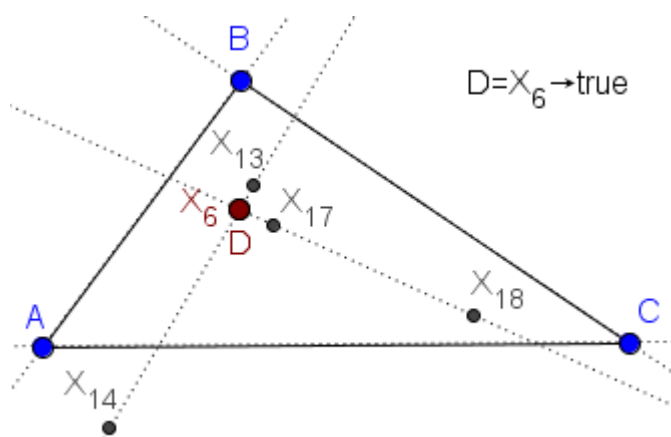



Рис. 15с. Модель для экспериментальной проверки утверждения примера 69

Пример 70. Построить модель для экспериментальной проверки утверждения: “Точки Наполеона лежат на гиперболе Киперта”.

Решение. Ранее в примере 64 мы уже проверяли, что гипербола Киперта проходит через центроид X_2 , ортоцентр X_4 , центр Шпикера X_{10} и точки Ферма X_{13} и X_{14} . Изображение, представленное на рис. 15d, получено так:

- продолжим построение модели, начиная с изображения рис. 13h из примера 64. Командами $X_{\{n\}}=TriangleCenter[A,B,C,n]$ ($n=17,18$) добавим к модели требуемые нам точки Наполеона X_{17} и X_{18} ;

- инструментом “ Текст” отредактируем код проверочной надписи, вставим в него перед последней строкой фрагмент:

первая точка Наполеона $\rightarrow \boxed{X_{\{17\}}==Intersect[g_1,X_{\{17\}}]} ,$

вторая точка Наполеона $\rightarrow \boxed{X_{\{18\}}==Intersect[g_1,X_{\{18\}}]} .$

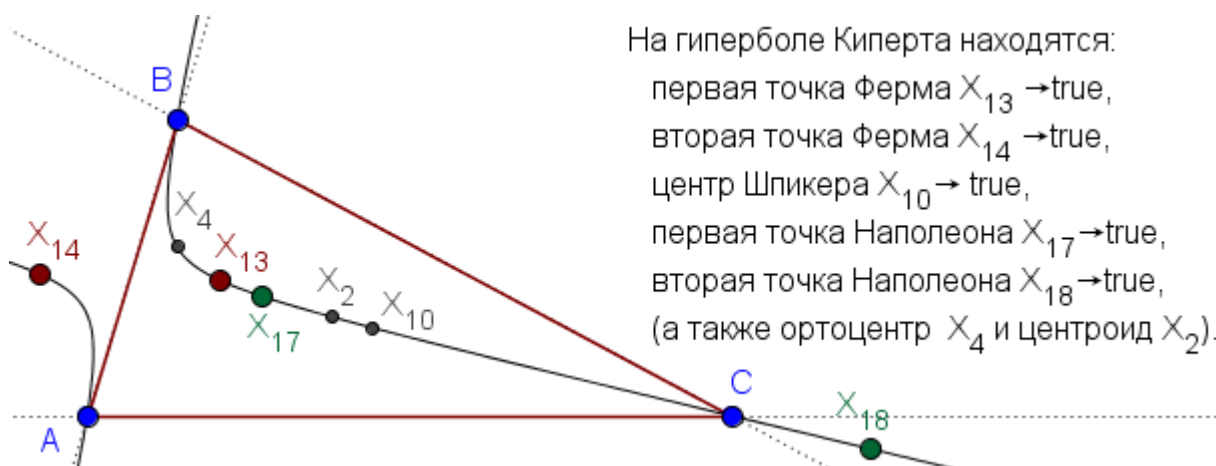





Рис. 15d. Модель для экспериментальной проверки утверждения примера 70

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

Пример 71. Построить модель для экспериментальной проверки утверждения: “Абсолютная величина разности площадей внешнего и внутреннего треугольников Наполеона равна площади S исходного $\triangle ABC$ ”.

Решение. Заметим, что площадь внутреннего треугольника может оказаться больше площади внешнего треугольника, и без слов “абсолютная величина” утверждение о площадях может оказаться неверным. Изображение, представленное на рис. 15e получено так:

- мы уже умеем строить треугольники Наполеона для $\triangle ABC$. Реализовав соответствующие действия на одном изображении (см. рис. 15e), получим внешний правильный $\triangle JNO$ и внутренний правильный $\triangle IGH$;
- инструментом “ Перпендикулярная прямая” из точки A проведем прямую, перпендикулярную стороне BC . Инструментом “ Точка на объекте” сформируем точку пересечения P проведенной прямой с отрезком BC или его продолжением;
- инструментом “ Текст” создадим проверочную надпись, подтверждающую или опровергающую утверждение примера 71:

$$S_{\{\triangle ABC\}} = |S_{\{\triangle JNO\}} - S_{\{\triangle IGH\}}| \rightarrow$$

$$\boxed{BC * AP / 2 == \text{abs}((OJ^2 - HG^2) * \sqrt{3}) / 4}.$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правой вычисляемой части надписи значения *true*, что и требовалось проверить.

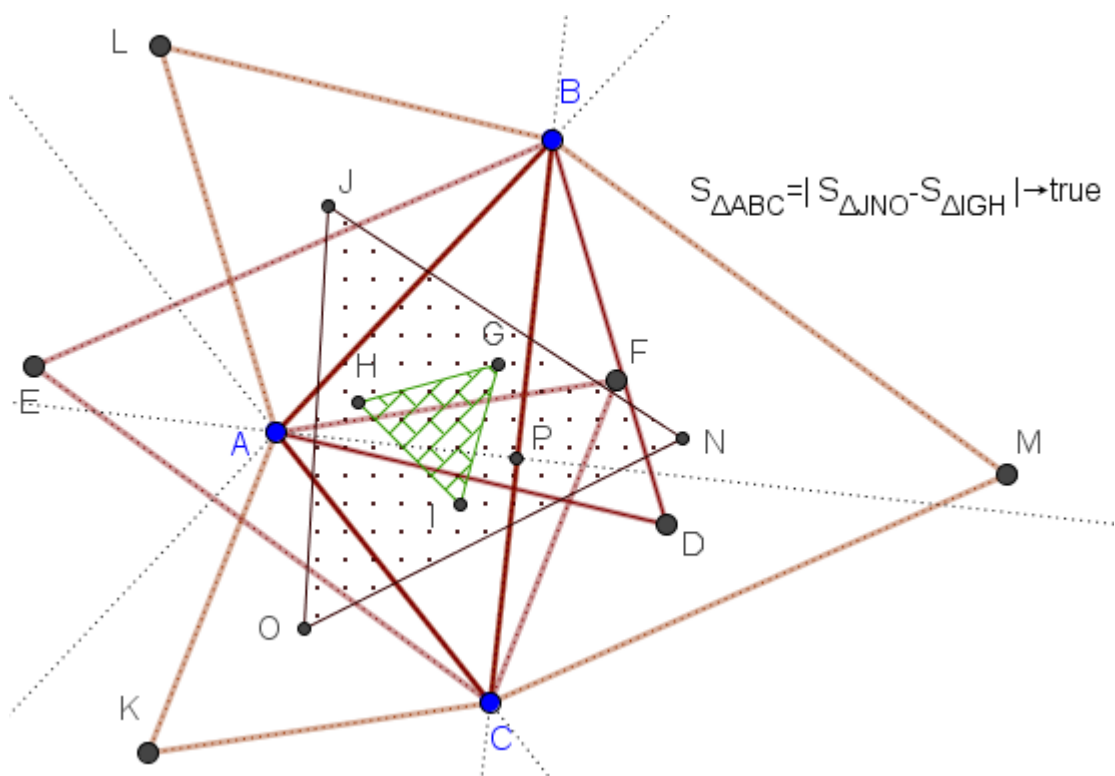



Рис. 15e. Модель для экспериментальной проверки утверждения примера 71

Пример 72. Построить модель для экспериментальной проверки соотношений (19) и (20) (см. выше).

Решение. Изображение, представленное на рис. 15f получено так:

- продолжим построение модели, начиная с изображения рис. 15e. Удалим проверочную надпись. Через контекстное меню проведем переименование отрезков BC , AC , AB , JO и HG соответственно на a , b , c , n и m ;
- инструментом “ Текст” создадим новую проверочную надпись, подтверждающую или опровергающую утверждение примера 72. Сделать это можно следующим кодом:

$$n^2 + m^2 = \frac{a^2 + b^2 + c^2}{3} \rightarrow$$

$$\boxed{n^2 + m^2 == (a^2 + b^2 + c^2)/3} , \backslash$$

$$n^2 = \frac{a^2 + b^2 + c^2}{6} + \frac{2 \cdot S}{\sqrt{3}} \rightarrow$$

$$\boxed{n^2 == (a^2 + b^2 + c^2)/6 + AP * CB / \sqrt{3}} , \backslash$$

$$m^2 = \frac{a^2 + b^2 + c^2}{6} - \frac{2 \cdot S}{\sqrt{3}} \rightarrow$$

$$\boxed{m^2 == (a^2 + b^2 + c^2)/6 - AP * CB / \sqrt{3}} . \backslash$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы всегда будем получать в правых вычисляемых частях каждой строки надписи значения *true*, что и требовалось проверить.

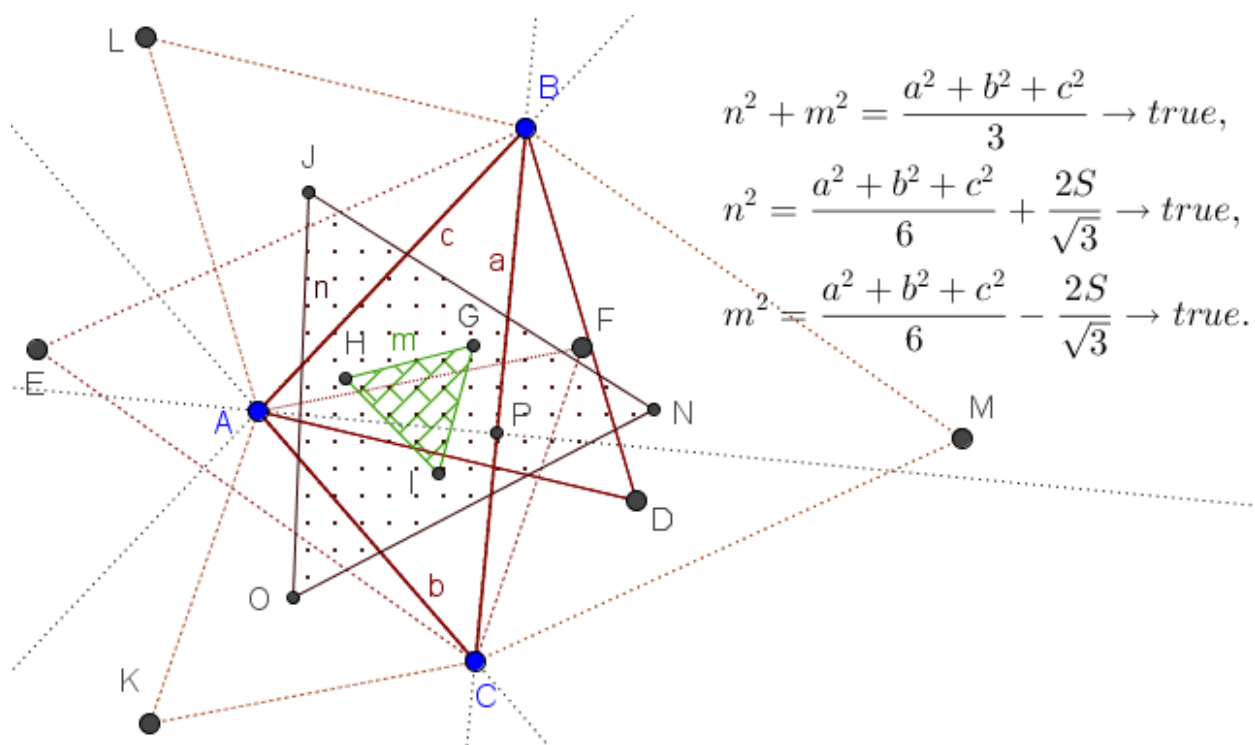











Рис. 15f. Модель для экспериментальной проверки утверждения примера 72

5.17. Обобщенные точки Наполеона

Определения первой и второй точек Наполеона, приведенные в предыдущем пункте, можно было бы переписать так. Пусть на сторонах $\triangle ABC$ во внешнем направлении как на основаниях построены равнобедренные треугольники с углом при вершине в 120° (или с углом при основании в 30°). Тогда точка пересечения трех прямых, проходящих через вершины внешних треугольников и противоположные вершины $\triangle ABC$, – это и есть первая точка Наполеона. Вторая точка Наполеона определяется как первая, но треугольники на сторонах $\triangle ABC$ строятся в направлении, противоположном внешнему. Из такого определения точек Наполеона легче переходить к их различным обобщениям. Например, можно считать угол при вершинах строящихся треугольников равным не 120° , а любому α ($0 < \alpha \leq 180^\circ$), или вообще избавиться от равнобедренности этих треугольников. Именно эти обобщения точек Наполеона и рассматриваются в данном разделе.

Пример 73. Пусть задан $\triangle ABC$ и на его сторонах, как на основаниях, во вне построены равнобедренные подобные треугольники. Построение должно формировать динамическую модель, в которой A, B, C являются свободными точками, а вершина G прикреплена к высоте из G на сторону AB . При этом любые изменения точек A, B, C и изменение вершины G , не переводящие ее в полуплоскость с $\triangle ABC$, должны приводить к изображению, в котором треугольники на сторонах $\triangle ABC$ остаются равнобедренными и подобными.

Решение. Приступим к созданию требуемой модели:

- инструментом “ Многоугольник” создадим $\triangle ABC$;
- инструментом “ Срединный перпендикуляр” проведем перпендикулярные линии к серединам сторон $\triangle ABC$;
- инструментом “ Точка на объекте” отметим точки D, E и F пересечения проведенных перпендикуляров со сторонами треугольника и прикрепим на перпендикуляре к стороне AB некоторую точку G ;
- инструментом “ Окружность по центру и радиусу” проведем окружность из точки E радиусом $BE \cdot GD / AD$. Инструментом “ Пересечение” сформируем требуемую нам точку пересечения H окружности и перпендикуляра. Через контекстное меню спрячем окружность, то есть сделаем ее невидимой;
- инструментом “ Окружность по центру и радиусу” проведем окружность из точки F радиусом $AF \cdot GD / AD$. Инструментом “ Пересечение” сформируем требуемую нам точку пересечения I окружности и перпендикуляра. Через контекстное меню спрячем окружность, то есть сделаем ее невидимой;
- инструментом “ Отрезок” проведем боковые стороны равнобедренных треугольников. По тому, как мы их строили, они подобны друг другу;
- инструментом “ Текст” в *LaTeX*-режиме создадим проверочную надпись, введя выражение:

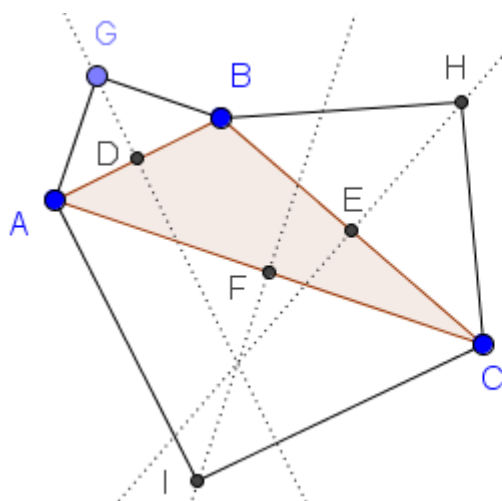
$$AG=GB \rightarrow \boxed{AG==GB}$$

$$BH=HC \rightarrow \boxed{BH==HC}$$

$$AI=IC \rightarrow \boxed{AI==IC}$$

$$\frac{AG}{AB} = \frac{BH}{BC} = \frac{CI}{AC} \rightarrow$$

$$\boxed{AG/AB==BH/BC==CI/AC} .$$







$$\begin{aligned} AG &= GB \rightarrow true, \\ BH &= HC \rightarrow true, \\ AI &= IC \rightarrow true, \\ \frac{AG}{AB} &= \frac{BH}{BC} = \frac{CI}{AC} \rightarrow true. \end{aligned}$$

Рис. 16а. Модель для экспериментальной проверки утверждений примера 73

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы будем изменять исходный $\triangle ABC$. Перемещая прикрепленную точку G по срединному перпендикуляру к AB будем изменять равнобедренные треугольники на сторонах $\triangle ABC$. При этом в правых вычисляемых частях всех строк проверочной надписи всегда получается значение *true*, что и является экспериментальным подтверждением правильности построенной модели.

Пример 74. Пусть задан $\triangle ABC$ и на его сторонах построены равнобедренные подобные треугольники так, как это описано в условиях примера 73. Построить динамическую модель для экспериментальной проверки того, что прямые, соединяющие точки G , H и I и противоположные вершины $\triangle ABC$, пересекаются в одной точке L , и эта точка лежит на гиперболе Киперта.

Решение. Если углы α при основаниях равнобедренных треугольников равны по 30° , то L – это первая точка Наполеона. В случае $0 \leq \alpha < 90^\circ$ такие точки будем называть обобщенными точками Наполеона 1 типа. Приступим к созданию требуемой модели, начиная с изображения, показанного на рис. 16а:

- через контекстное меню удалим проверочную надпись и инструментом “ Прямая” проведем прямые через пары точек A и H , B и I , C и G . Они получат имена p , q и r . Похоже, что они пересекаются в одной точке;
- инструментом “ Пересечение” сформируем точку J пересечения двух из этих прямых, скажем p и q . Нам еще предстоит экспериментально убедиться в том, что J остается точкой пересечения p и q при любых перемещениях по полотну вершин A , B , C и точки G по вертикальной прямой к стороне AB и что через J проходит и третья прямая r ;
- командами $TriangleCenter[A,B,C,k]$ ($k=2,4$) через строку ввода выведем центроид и ортоцентр $\triangle ABC$. Через контекстное меню спрячем обозначение выведенных точек. Инструментом “ Коника по 5 точкам” сформируем гиперболу Киперта. Она получит имя s ;
- инструментом “ Текст” создадим проверочную надпись, введя выражение:

1. A , B , C – свободные точки,
 G прикреплена к срединному перпендикуляру отрезка AB .
2. Прямые p , q , r пересекаются
 в точке $J \rightarrow \boxed{AreConcurrent[p,q,r]}$,
3. Расстояние от точки J до
 гиперболы Киперта $\rightarrow \boxed{Distance[J, s]}$.

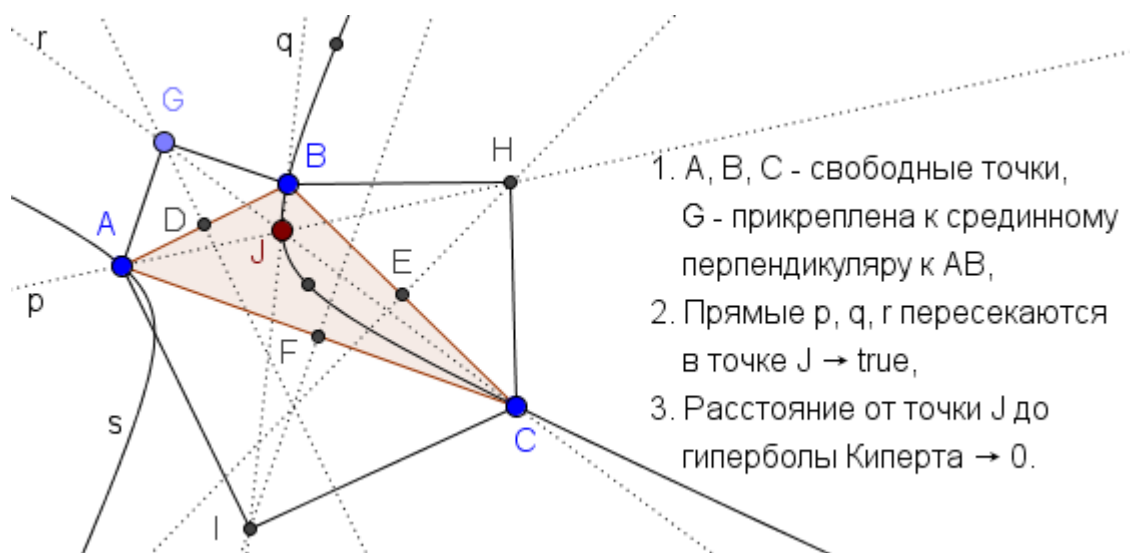


Рис. 16b. Модель для экспериментальной проверки утверждений примера 74

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы будем получать разные исходные $\triangle ABC$. Перемещая точку G по срединному перпендикуляру к AB , будем изменять равнобедренные треугольники на сторонах $\triangle ABC$, оставляя их подобными. При этом во втором пункте проверочной надписи в правой вычисляемой части всегда будем получать значение *true*, то есть экспериментальное подтверждение пересечения прямых p , q и r в одной точке. Пункт третий проверочной надписи демонстрирует нам, что точка J лежит на гиперболе Киперта. Заметим, что нуль в правой вычисляемой части получается потому, что по умолчанию вывод реализуется с двумя десятичными знаками после десятичной точки. Чтобы видеть более точное значение приближенно вычисленного расстояния от точки J до гиперболы, следует позаботиться о выводе в результате большего количества десятичных знаков. Если, например, через меню выполнить установку *Options/Rounding/15 Decimal Places*, то при перемещениях вершин A , B , C и (или) точки G пункт три надписи может оказаться и иным, скажем, таким:

3. Расстояние от точки J до
гиперболы Киперта → 0.0000000000000002.

Однако при экспериментировании мы всегда получали в результате первую отличную от нуля значащую цифру лишь на 15 позиции после запятой.

Пример 75. Пусть в условиях примера 73 равнобедренные подобные треугольники строятся на сторонах $\triangle ABC$ не вовне, а в противоположном направлении, и должна быть создана динамическая модель с такими же свойствами, как и в примере 73. Иными словами, любые изменения точек A , B , C и изменение вершины G , не переводящие ее в полуплоскость без $\triangle ABC$, должны приводить к изображению, в котором треугольники на сторонах $\triangle ABC$ остаются равнобедренными и подобными.

Решение. В данном случае модель строится по полной аналогии с моделью примера 73.

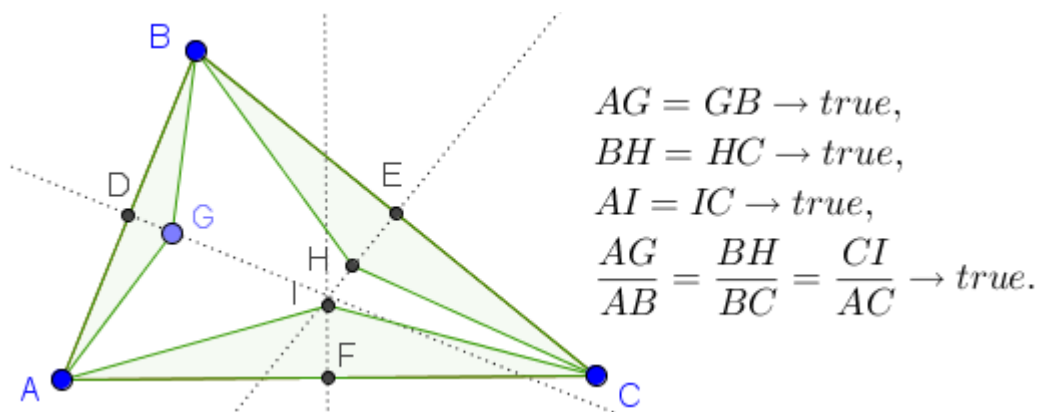


Рис. 16с. Модель для экспериментальной проверки утверждений примера 75

Пример 76. Пусть задан $\triangle ABC$ и на его сторонах построены равнобедренные подобные треугольники так, как это описано в условиях примера 75. Построить динамическую модель для экспериментальной проверки того, что прямые, соединяющие точки G, H и I и противоположные вершины $\triangle ABC$, пересекаются в одной точке J , и эта точка лежит на гиперболе Киперта (см. рис. 16d).

Решение. Если углы α при основаниях равнобедренных треугольников равны по 30° , то L – это вторая точка Наполеона. В случае $0 \leq \alpha < 90^\circ$ такие точки будем называть обобщенными точками Наполеона второго типа. В данном случае модель строится по полной аналогии с моделью примера 74.

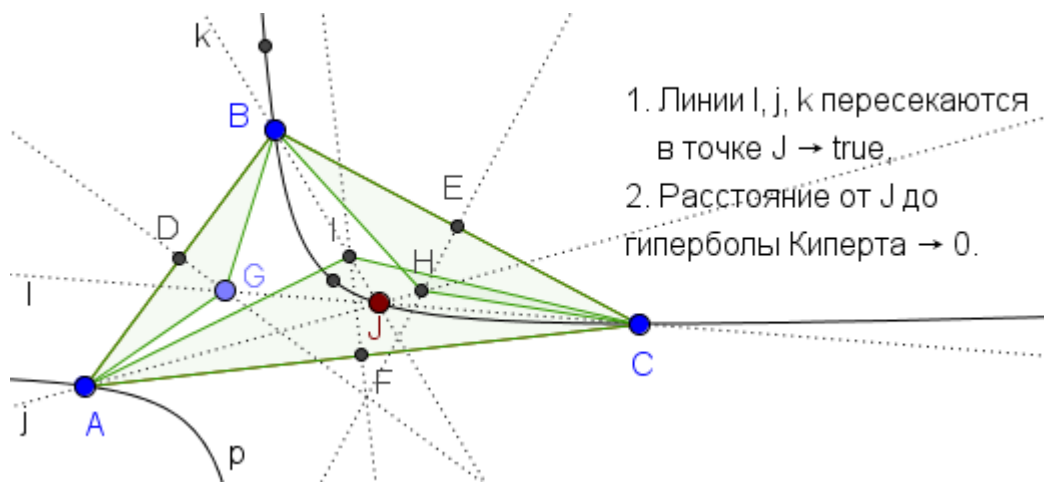








Рис. 16d. Модель для экспериментальной проверки утверждений примера 76



Пример 77. Пусть задан $\triangle ABC$, на его сторонах вовне построены подобные треугольники причем именно так, как это показано на рис. 16е, то есть: $\angle DAB = \angle KAC$, $\angle DBA = \angle HBC$ и $\angle HCB = \angle KCA$. Построение должно формировать динамическую модель, в которой A, B, C и D являются свободными точками. При этом любые изменения точек A, B, C и изменение вершины D , не



переводящие ее в полуплоскость с $\triangle ABC$, должны приводить к изображению, в котором треугольники на сторонах $\triangle ABC$ остаются подобными с той же самой ориентацией.



Решение. Требуемая модель может быть построена так:


- инструментом “ Многоугольник” создадим $\triangle ABC$;
- инструментом “ Прямая” проведем прямую через точки A и B ;
- инструментом “ Многоугольник” на стороне AB вовне от $\triangle ABC$ построим $\triangle DAB$ и изменим стиль его вывода. Инструментом “ Угол” в созданном треугольнике отметим декорациями $\angle BAD$ и $\angle DBA$ так, как это показано на рис. 16е, и приступим к построению треугольников на сторонах BC и AC $\triangle ABC$;

- инструментом “ Окружность по центру и радиусу” построим две окружности: первую – из точки B радиусом $BA \cdot BC / BD$ и вторую – из точки C радиусом $AD \cdot BC / BD$. Инструментом “ Пересечение” сформируем требуемую нам точку пересечения E этих окружностей. Через контекстное меню спрячем, то есть сделаем невидимыми окружности;

- инструментом “ Многоугольник” построим $\triangle BCE$ и изменим стиль его вывода. Инструментом “ Угол” отметим декорациями $\angle CBE$ и $\angle ECB$ так, как это показано на рис. 16е;

- инструментом “ Окружность по центру и радиусу” построим две окружности: первую – из точки A радиусом $AB \cdot AC / AD$ и вторую – из точки C радиусом $BD \cdot AC / AD$. Инструментом “ Пересечение” сформируем требуемую нам точку пересечения F этих окружностей. Через контекстное меню спрячем, то есть сделаем невидимыми окружности;

- инструментом “ Многоугольник” построим $\triangle ACF$ и изменим стиль его вывода. Инструментом “ Угол” отметим декорациями $\angle FAC$ и $\angle ACF$ так, как это показано на рис. 16е,

- инструментом “ Текст” в *LaTeX*-режиме создадим проверочную надпись, введя выражение:

$$\frac{AD}{CE} = \frac{DB}{BC} = \frac{BA}{BE} \rightarrow \boxed{AD/CE == DB/BC == BA/BE} ,$$

$$\frac{AD}{AC} = \frac{DB}{CF} = \frac{BA}{AF} \rightarrow \boxed{AD/AC == DB/CF == BA/AF} .$$

Перемещая на построенной модели независимые точки A , B и C по плоскости, мы будем изменять исходный $\triangle ABC$. Перемещая точку D , не пересекая

прямой AB , будем изменять треугольники на сторонах $\triangle ABC$. При этом в правых вычисляемых частях обеих строк проверочной надписи всегда будем видеть значение *true*, то есть экспериментальное подтверждение правильности построенной модели.

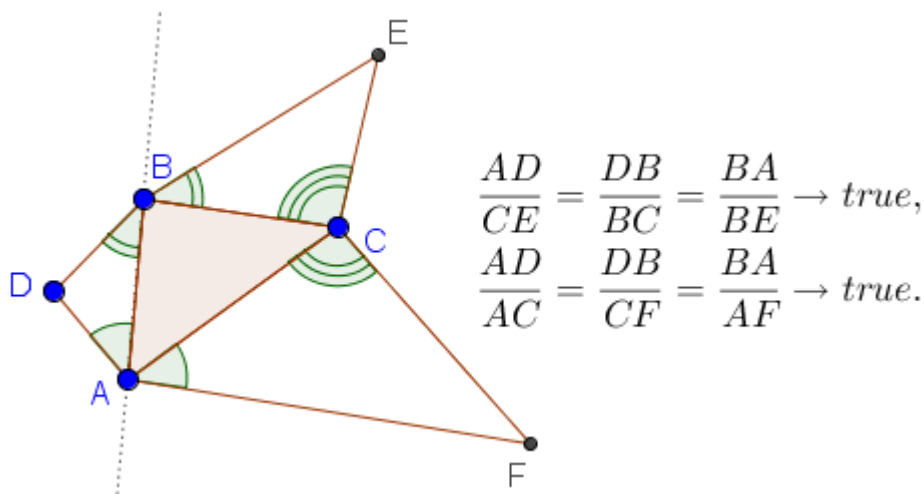






Рис. 16е. Модель для экспериментальной проверки утверждений примера 77

Пример 78. Пусть задан $\triangle ABC$ и на его сторонах построены подобные треугольники так, как это описано в условиях примера 77. Построить динамическую модель для экспериментальной проверки того, что прямые, соединяющие точки D, E и F и противоположные вершины $\triangle ABC$, пересекаются в одной точке.

Решение. Приступим к созданию требуемой модели, начиная с изображения, показанного на рис. 16е:

- через контекстное меню удалим проверочную надпись и инструментом “ Прямая” проведем прямые через пары точек A и E , B и F , C и D . Они получат имена i, j и l ;
- инструментом “ Пересечение” сформируем точку G пересечения этих прямых и изменим стиль ее вывода;
- командами $TriangleCenter[A,B,C,k]$ ($k=2,4$) через строку ввода выведем центроид и ортоцентр $\triangle ABC$. Через контекстное меню спрячем обозначение выведенных точек. Инструментом “ Коника по пяти точкам” по трем вершинам $\triangle ABC$, центроиду и ортоцентру сформируем гиперболу Киперта. Она получит имя p . Пункт необязателен и вставлен лишь для демонстрации того, что точка G , вообще говоря, не находится на p ;
- инструментом “ Текст” создадим проверочную надпись, введя выражение:

- Прямые i, j, l пересекаются
в точке $G \rightarrow \boxed{AreConcurrent[i,j,l]}$,
- Расстояние от точки G до

гиперболы Киперта \rightarrow $\boxed{Distance[G, p]}$.

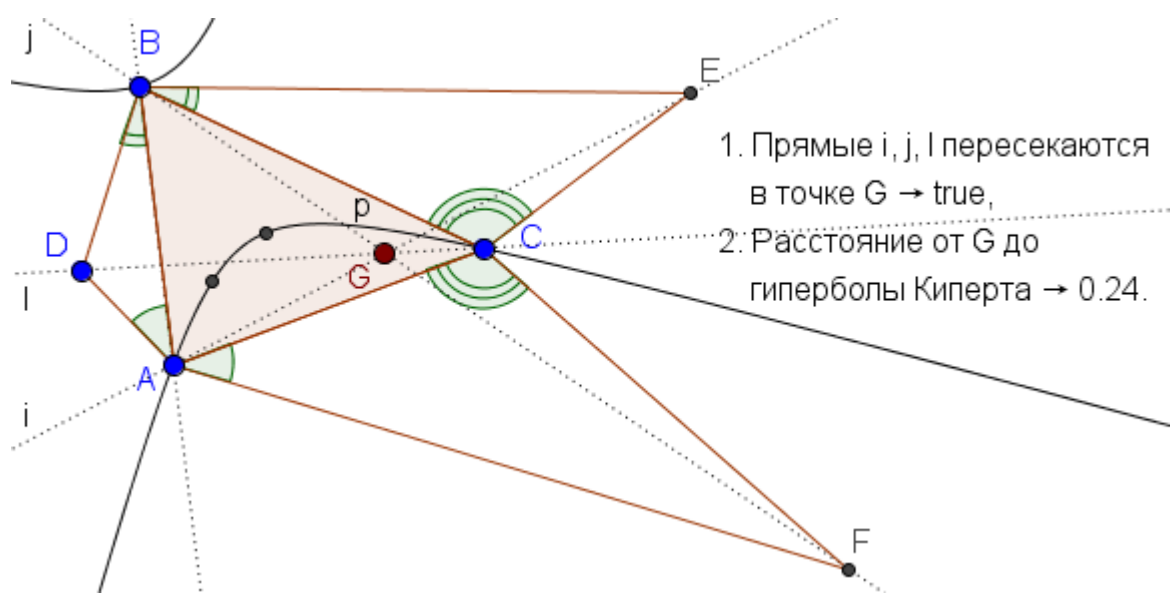


Рис. 16f. Модель для экспериментальной проверки утверждений примера 78

Пример 79. Пусть в условиях примера 73 такие же подобные треугольники строятся на сторонах $\triangle ABC$ не вовне, а в противоположном направлении, и должна быть создана динамическая модель с такими же свойствами, как и в примере 73. Иными словами, любые изменения точек A, B, C и изменение вершины G , не переводящие ее в полуплоскость без $\triangle ABC$, должны приводить к изображению, в котором треугольники на сторонах $\triangle ABC$ остаются подобными с той же самой ориентацией.

Решение. В данном случае модель строится по полной аналогии с моделью примера 73.

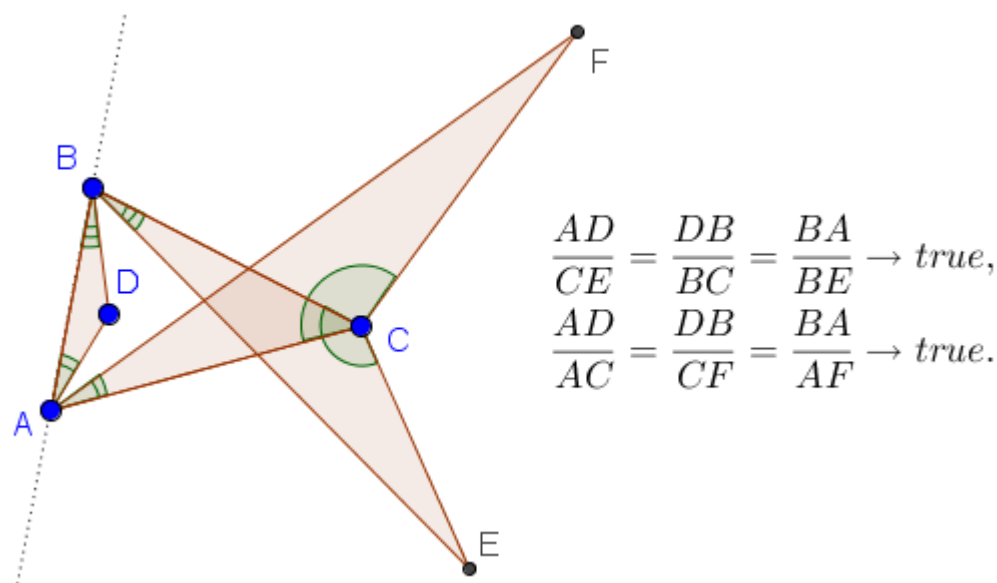


Рис. 16g. Модель для экспериментальной проверки утверждений примера 79

Пример 80. Пусть задан $\triangle ABC$ и на его сторонах построены подобные треугольники так, как это описано в условиях примера 79. Построить динамическую модель для экспериментальной проверки того, что прямые, соединяющие точки D , H и K и противоположные вершины $\triangle ABC$, пересекаются в одной точке L .

Решение. В данном случае модель строится по полной аналогии с моделью примера 78.

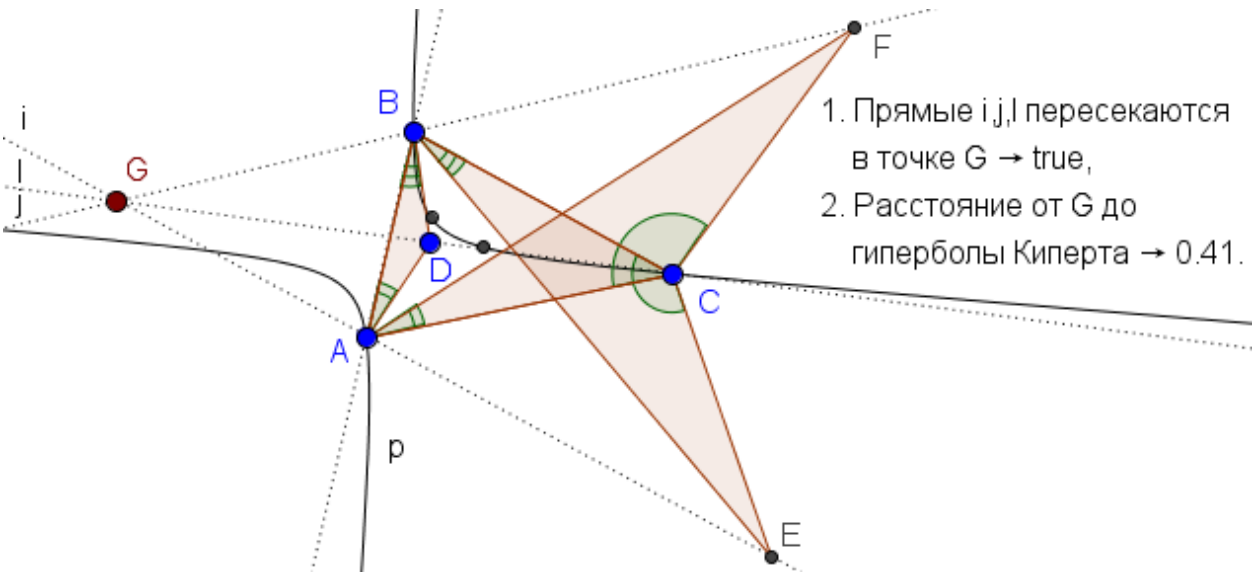







Рис. 16h. Модель для экспериментальной проверки утверждений примера 80

6. 2D-графики

Плоскую или двухмерную графику называют также 2D-графикой, а пространственную или трехмерную графику – 3D-графикой. Вывод плоских графиков может производиться как на панель (окно) “Полотно” (*Graphics*), так и на панель “Полотно 2” (*Graphics 2*). Вывод пространственных графиков реализуется на панель “Полотно 3D” (*3D Graphics*). В одной системе координат может быть выведено несколько графиков и не обязательно одного типа. В данном параграфе речь идет о работе с 2D-графиками. В *GeoGebra* статичная и динамическая визуализация всевозможных наборов точек и кривых организована при минимальных усилиях со стороны пользователя в основном с помощью набора команд (функций), вводимых через строку ввода. При этом панель “Настройки” предоставляет богатый и хорошо структурированный набор средств форматирования графиков. Ниже подробно рассказывается о средствах вывода *точечных графиков* (инструмент “ Точка”, список, *Sequence*), *графиков функций одной переменной* (*Function*), *графиков неявных функций* (*ImplicitCurve*), *графиков параметрически заданных функций* (*Curve*), *графиков функций в полярных координатах* (*Curve, Function, ImplicitCurve*, инструмент “ Локус”). Кроме того, обсуждается ряд специальных графиков, выводимых командами: *BarChart, Histogram, HistogramRight, BoxPlot, FrequencyPolygon, StepGraph, StickGraph* и *ResidualPlot*.

6.1. Точечные графики

I. *Работа инструментом “ Точка”*. Точечный график можно вывести, формируя его точки инструментом “ Точка” непосредственно на панели “Полотно”. Если те или иные выведенные точки требуется соединить отрезками прямых линий, то для этих целей следует использовать инструмент “ Отрезок”.

II. *Задание списка точек*. Точечный график можно вывести, задав в строке ввода список всех его последовательных точек в виде


$$lis = \{(a_1, b_1), (a_2, b_2), \dots (a_k, b_k)\}. \quad (1)$$

Если при вводе (1) точки не появились, то следует включить соответствующую радиокнопку на панели “Объекты”. Если левая часть (1) присутствует, то это имя списка, если левая часть (1) отсутствует, то выводимый список именуется системой. В нашем случае имя списка – это *lis*.

III. *Формирование последовательности точек.* Точечный график через строку ввода можно вывести с помощью команды

$$lis=Sequence[(ex1, ex2), var, start, end, inc], \quad (2)$$

где: $ex1, ex2$ – выражения; var – имя переменной ($var \neq x, var \neq y$); $start, end$ – начальное и конечное значения для var ; inc – шаг для var . Аргумент inc не обязателен и по умолчанию равен 1. По (2) последовательно генерируются точки $(ex1, ex2)$ сначала при $var=start$, затем при $var=start+inc$ и т. д. пока var не станет больше end ($start < end$) или var не станет меньше end ($start > end$). Полученные точки возвращаются в виде именованного системой или пользователем списка. В нашем случае имя списка – это lis .

Пары выведенных смежных точек можно соединить отрезками прямых линий с помощью инструмента  *Отрезок*” или через строку ввода, выполнив команду

$$Sequence[Segment[Element[lis, i], Element[lis, i+1]], i, 1, Dimension[lis]-1], \quad (3)$$

где $Element[lis, i]$ – элемент списка lis с индексом i (нумерация от 1 и далее), $Dimension[lis]$ – количество элементов в списке lis . $Segment[A, B]$ – отрезок прямой между точками A и B , $Sequence[...]$ – см. (2).

IV. *Сплайны.* Пусть пары выведенных смежных точек требуется соединить сплайновой кривой порядка n ($n \geq 3$). Сделать это можно с помощью команды

$$Spline[lis. n]. \quad (4)$$

В (4) аргумент n не обязателен и по умолчанию он равен 3.

Пример 1. Вывести график функции, заданной списком точек $\{(0, 2), (2, 0), (4, 0.5), (6, 2.5), (8, 1)\}$.

Решение. В соответствии с (1) решить предложенную задачу можно простым вводом списка $\{(0, 2), (2, 0), (4, 0.5), (6, 2.5), (8, 1)\}$. Результат представлен на рис. 1, где надпись сформирована как *LaTeX*-текст командой *FormulaText*[" $\{(0, 2), (2, 0), (4, 1), (6, 4), (8, 1)\}$ ”].

Пример 2. Вывести график функции, заданной последовательностью точек $(n, 2+\sin(n^2))$ ($n=1, 2, \dots, 8$).

Решение. В соответствии с (2) решить предложенную задачу можно командой $lis=Sequence[(n, 2+\sin(n^2)), n, 1, 8]$ (см. рис. 2). При этом надпись, приведенная на рисунке, сформирована как *LaTeX*-текст командой *FormulaText*[" $lis=Sequence[(n, 2+\sin(n^2)), n, 1, 8]$ ”].

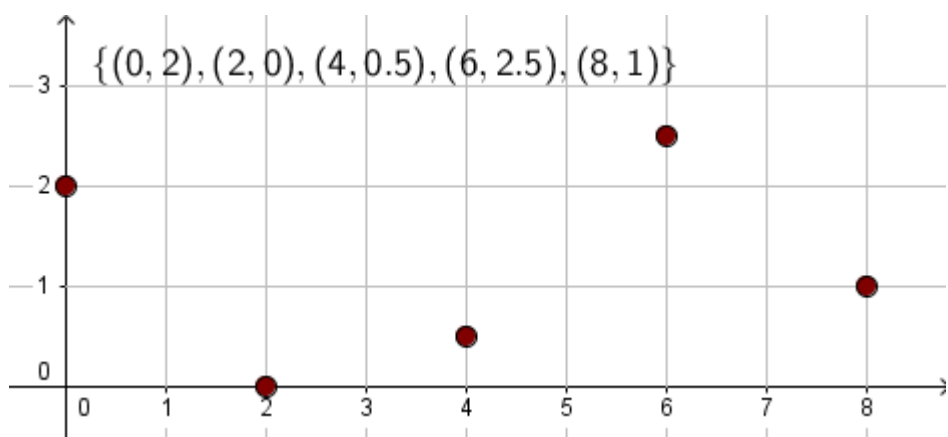


Рис. 1. Точечный график по заданному списку точек

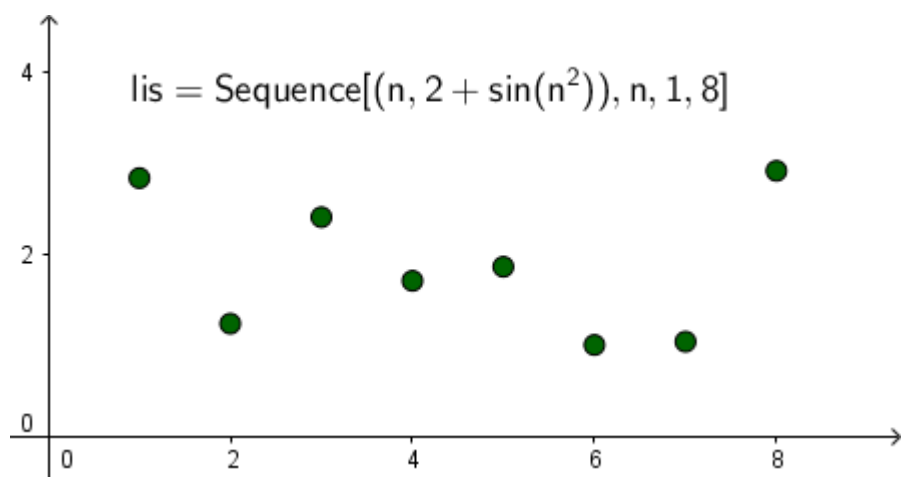


Рис. 2. Точечный график по сформированному списку точек $(n, 2 + \sin(n^2))$

Пример 3. Вывести график функции, заданной последовательностью точек $(n, 2 + \sin(n^2))$ ($n=1, 2, \dots, 8$), и соединить пары смежных точек отрезками прямых линий.

Решение. Выводим точки и первую строку надписи, как и в примере 2. Смежные точки соединяем отрезками линий в соответствии с (3) (см. рис. 3). Вторую двухстрочную строку надписи можно сформировать с помощью команды `TableText[{"первая строка"}, {"вторая строка"}]`.

Пример 4. Вывести график функции, заданной последовательностью точек $(n, 2 + \sin(n^2))$ ($n=1, 2, \dots, 8$) и построить для них сплайны 3 и 4 порядков.

Решение. Выводим точки и первую строку надписи, как и в примере 2. Сплайны строим в соответствии с (4). Вторую строку надписи формируем с помощью команды `TableText[{"Spline[lis, 3],"}, {"Spline[lis, 4]", "v"}]`.

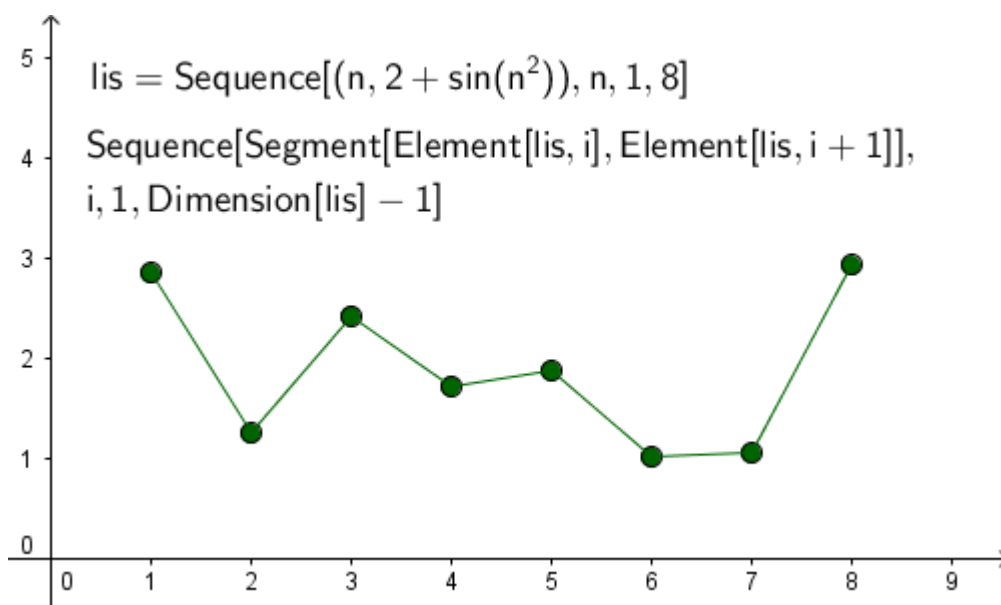


Рис. 3. Точки графика рис. 2, соединенные отрезками линий

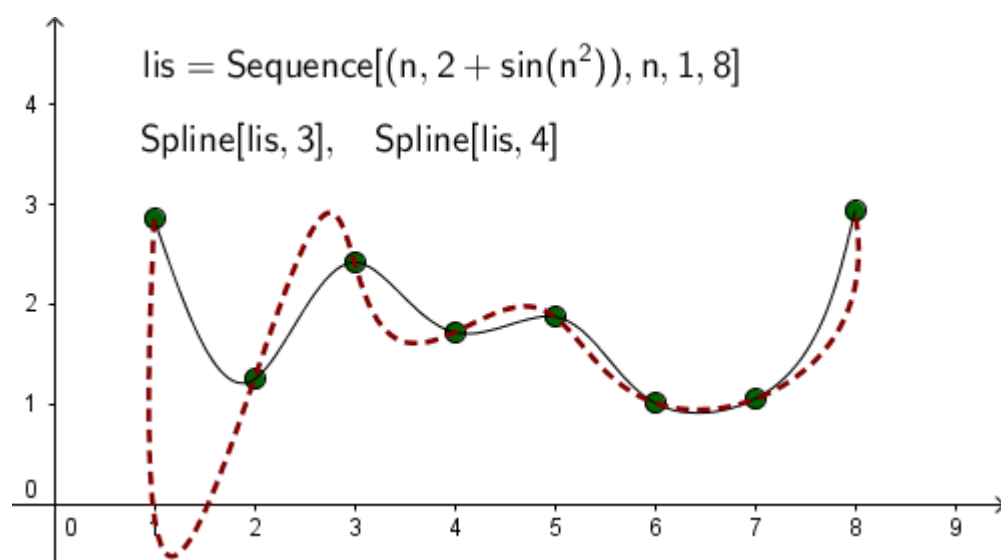


Рис. 4. Сплайны третьего и четвертого порядка для точек графика рис. 2

6.2. Графики функций одной переменной

I. *График по заданным ординатам.* Пусть на отрезке $[a, b]$ ($a < b$) требуется вывести график некоторой функции, для которой известны действительные ординаты y_0, y_1, \dots, y_n в последовательных точках, расположенных от a до b включительно на равных расстояниях друг от друга. Сделать это можно командой

$$\text{Function}[\{a, b, y_0, y_1, \dots, y_n\}]. \quad (5)$$

II. *График на заданном промежутке.* Пусть требуется вывести график функции $y = \text{expr}(x)$ одной переменной x на отрезке $[a, b]$ ($a < b$), где $\text{expr}(x)$ – некоторое выражение от x . Сделать это можно командой

$$\text{Function}[\text{expr}(x), x, a, b]. \quad (6)$$

Если имя переменной – x , то в (6) второй аргумент можно вообще не указывать. Если имя переменной отлично от x , то в (6) оно должно быть указано.

III. График $\text{expr}(x)$ на промежутке $(-\infty, \infty)$. Если выражение $\text{expr}(x)$ ($x+\sin(x)$, $2+\text{floor}(x)+\cos(x)$, ...) непосредственно ввести в строку ввода, то по умолчанию система будет пытаться строить график функции $y=\text{expr}(x)$ на промежутке $(-\infty, \infty)$. Имя переменной может быть только x . Для ограничения области вывода нужно использовать функцию (6).

Пример 5. Вывести график функции с заданными ординатами $\{2, 3, 0, 3, 1, 2, 4, 4, 1\}$ в равноотстоящих точках отрезка $[0, 8]$, включая его концы. Построить точки с указанными ординатами.

Решение. В соответствии с (5) это можно сделать командой $\text{Function}[\{0, 8, 2, 3, 0, 3, 1, 2, 4, 4, 1\}]$ (см. рис. 5). При этом точки можно вывести списком $\{(0, 2), (1, 3), (2, 0), (3, 3), (4, 1), (5, 2), (6, 4), (7, 4), (8, 1)\}$, а надпись – командой $\text{FormulaText}["\text{Function}[\{0, 8, 2, 3, 0, 3, 1, 2, 4, 4, 1\}]"]$.

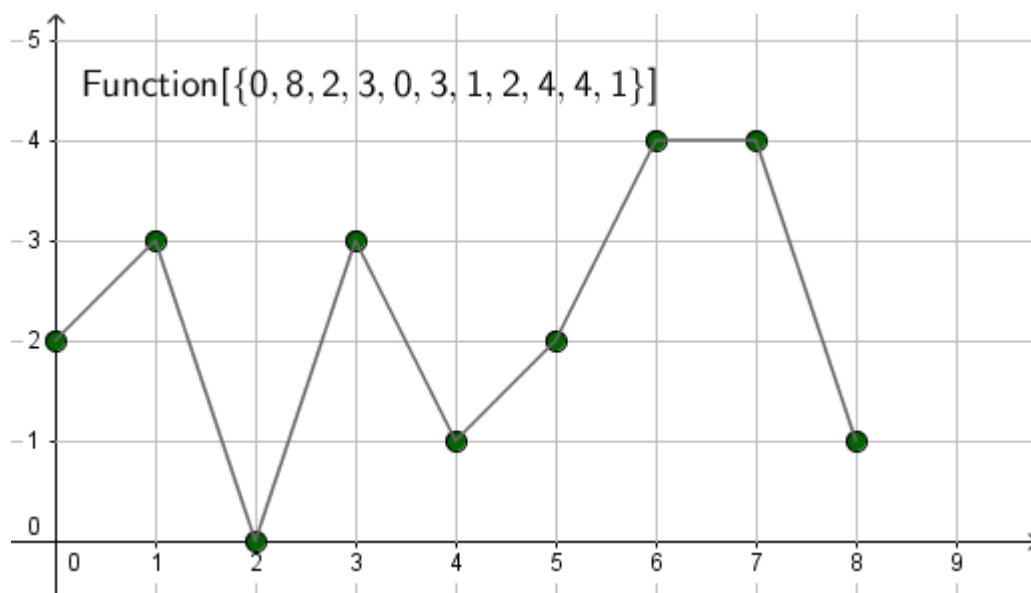


Рис. 5. Вывод графика функции по заданным ординатам

Пример 6. Вывести график функции $(\sin(x)+\text{ceil}(x))/3$ на отрезке $[0, 8]$ и точечный график этой функции в целочисленных абсциссах данного отрезка.

Решение. В соответствии с (6) требуемый график можно построить вводом команды $\text{Function}[(\sin(x)+\text{ceil}(x))/3, x, 0, 8]$ (см. рис. 6). При этом точечный график выведен по $\text{Sequence}[(n, (\sin(n)+\text{ceil}(n))/3), n, 0, 8]$, а надпись – командами $\text{FormulaText}["..."]$.

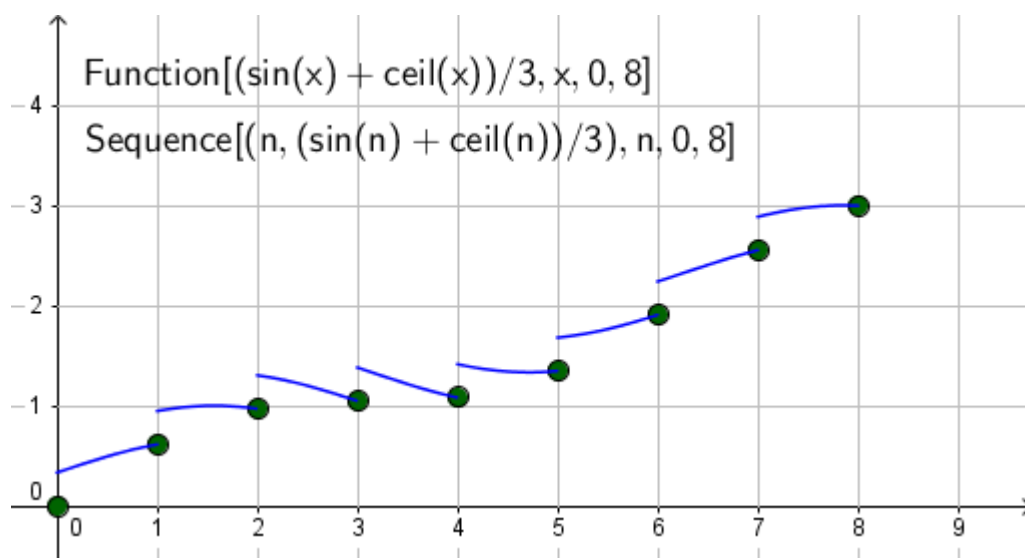


Рис. 6. Вывод графика функции на заданном отрезке

6.3. Графики неявных функций

I. Строить неявную полиномиальную кривую степени n можно с помощью команды

$$\text{ImplicitCurve}[\{P_1, P_2, \dots, P_k\}], \quad (7)$$

где $\{P_1, P_2, \dots, P_k\}$ – список точек, а $k=n(n+3)/2$. Иными словами, по $1(1+3)/2=2$ точкам можно строить прямую линию, по $k=2(2+3)/2=5$ точкам – коническую кривую, по $3(3+3)/2=9$ точкам – кривую третьего порядка и т. д.

II. Для вывода неявной кривой, аналитически заданной в виде $g(x,y)=0$, можно воспользоваться командой

$$\text{ImplicitCurve}[g(x, y)]. \quad (8)$$

Пример 7. Вывести список точек $li=\{(0, 2), (2, 0), (4, 1), (6, 4), (8, 1)\}$ и кривую, проходящую через указанные точки.

Решение. В соответствии с (7) это можно сделать последовательностью команд, показанных на рис. 7 в виде надписей. По ним и построена коническая кривая – гипербола. Сами надписи выводились командами $\text{FormulaText}["li=\{\text{точки}\} \setminus \}"]$ и $\text{FormulaText}["\text{ImplicitCurve}[li]"]$.

Пример 8. Вывести список точек $lis=\{(0, 2), (1, 3), (2, 0), (3, 2), (4, 1), (5, 5), (6, 4), (7, 3), (8, 1)\}$ и кривую, проходящую через указанные точки.

Решение. В соответствии с (7) это можно сделать последовательностью команд, показанных на рис. 8 в виде надписей. По ним и построена кривая третьего порядка. Сами надписи выводились командами $\text{FormulaText}["lis=\{\text{точки}\} \setminus \}"]$ и $\text{FormulaText}["\text{ImplicitCurve}[lis]"]$.

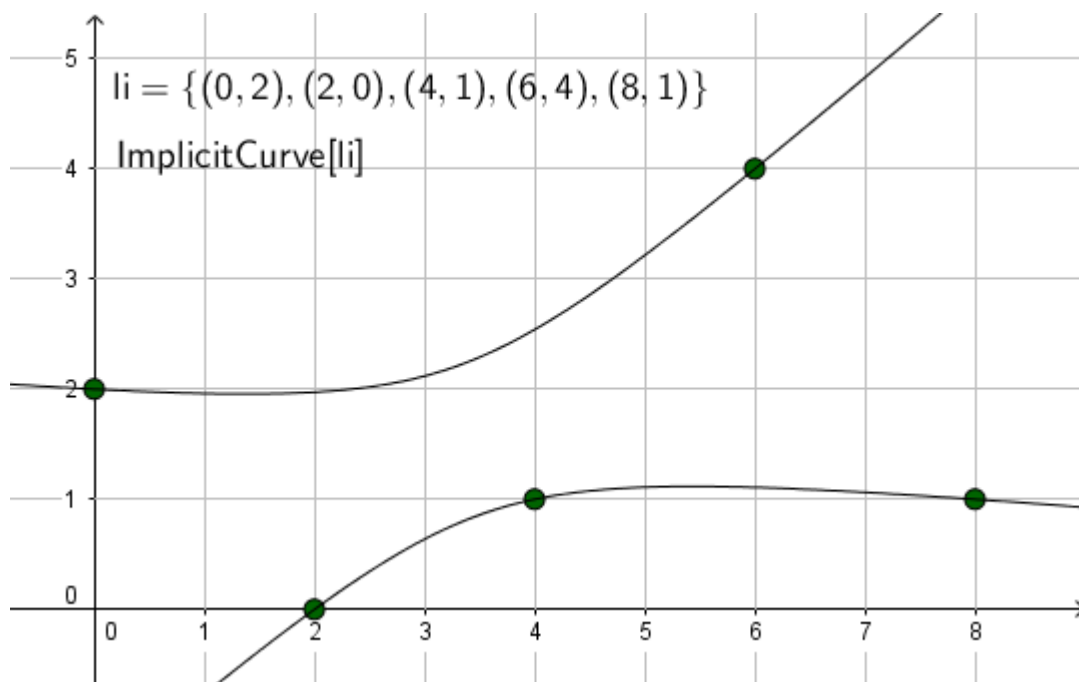


Рис. 7. Вывод графика неявной функции по пяти точкам

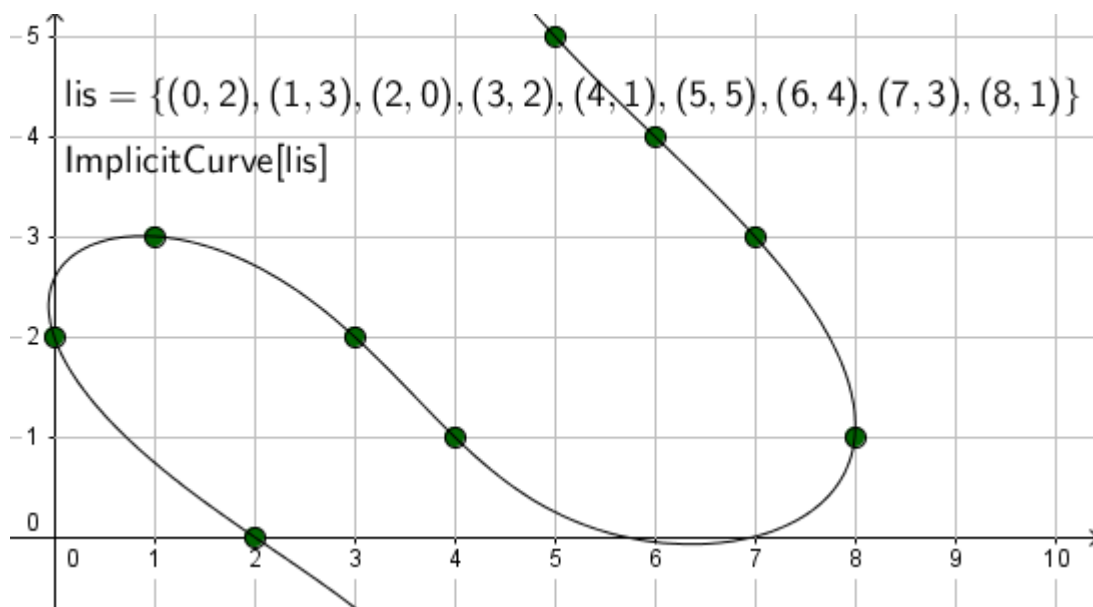


Рис. 8. Вывод графика неявной функции по девяти точкам

Пример 9. Вывести график неявно заданной функции

$$x^3 \cdot y^3 - x^4 + y^4 + x^3 + y^3 - x \cdot y - x + y + 1 = 0.$$

Решение. В соответствии с (8) это можно сделать командой `ImplicitCurve[$x^3 \cdot y^3 - x^4 + y^4 + x^3 + y^3 - x \cdot y - x + y + 1$]` (см. рис. 9). Надпись на этом рисунке выведена командой `FormulaText["ImplicitCurve[$x^3 y^3 - x^4 + y^4 + x^3 + y^3 - x y - x + y + 1$ "]]`.

Пример 10. Вывести график неявно заданной функции

$$x^5 + y^5 - 10xy^2 + y/2 = 0.$$

Решение. В соответствии с (8) это можно сделать командой `ImplicitCurve[$x^5+y^5-10\cdot x\cdot y^2+y/2$]` (см. рис. 9). Надпись на этом рисунке выведена командой `FormulaText[“ImplicitCurve[$x^5+y^5-10x\cdot y^2+y/2$]”]`.

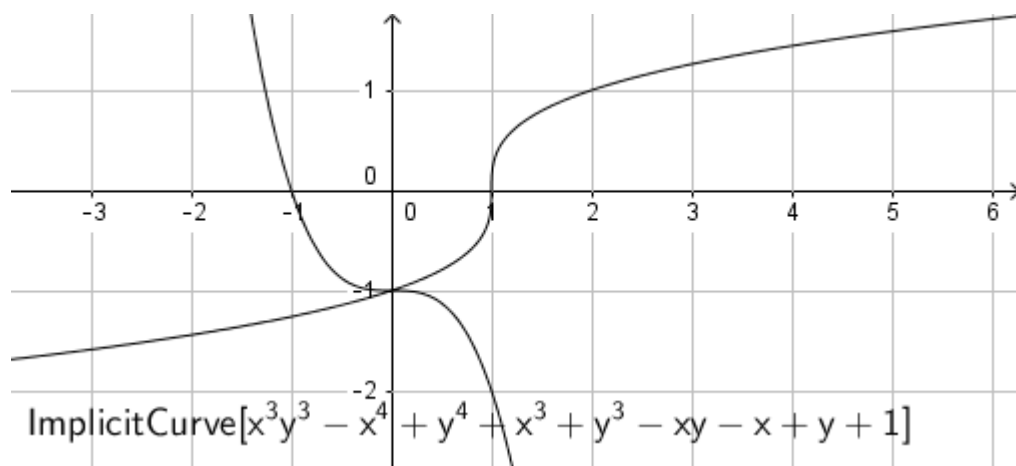


Рис. 9. Вывод графика неявной функции, заданной аналитически (уравнением)

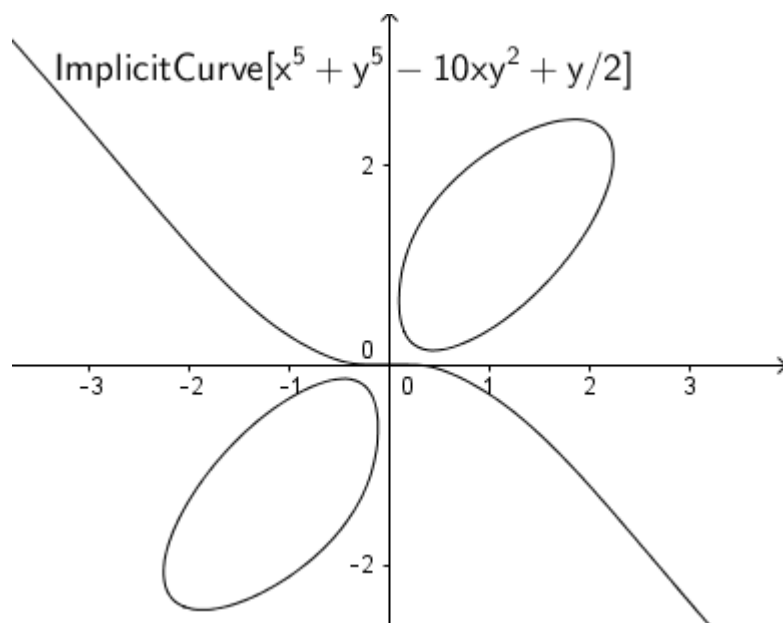


Рис. 10. Вывод графика неявной функции, заданной аналитически (уравнением)

6.4. Графики параметрических функций

Функции одной переменной в параметрическом виде задаются так:

$$\begin{cases} x = f(t) \\ y = g(t) \end{cases} \quad (a \leq t \leq b).$$

График такой функции можно сформировать командой

$$\text{Curve}[f(t), g(t), t, a, b]. \tag{9}$$

Пример 11. Вывести график функции, заданной параметрически в виде

$$\begin{cases} x = 3\left(\cos(t) + \frac{\cos(7.25t)}{7.25}\right) \\ y = 2\left(\sin(t) + \frac{\sin(7.25t)}{7.25}\right) \end{cases} \quad (0 \leq t \leq 8\pi). \tag{10}$$

Решение. В соответствии с (9) это можно сделать командой $\text{Curve}[3 \cdot (\cos(t) + \cos(7.25 \cdot t)/7.25), 2 \cdot (\sin(t) + \sin(7.25 \cdot t)/7.25), t, 0, 8 \cdot \pi]$.

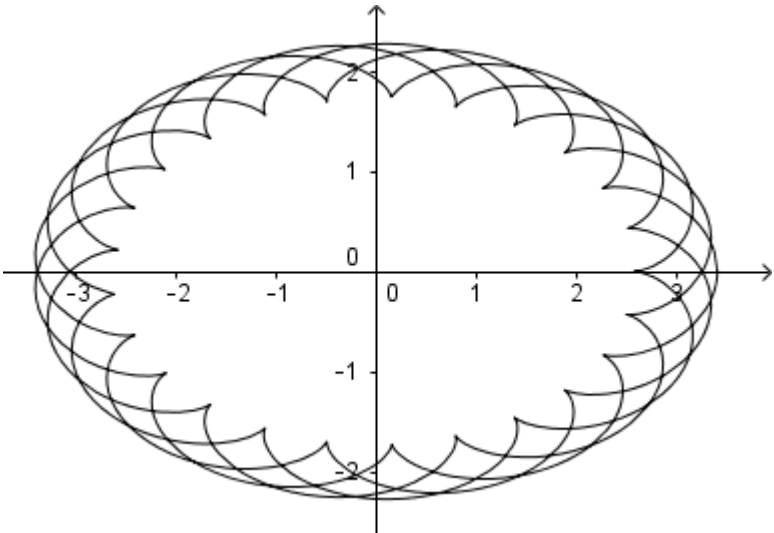


Рис. 11. Вывод графика функции, параметрически заданной в виде (10)

Пример 12. Вывести график функции, параметрически заданной в виде

$$\begin{cases} x = 2(\cos(t) + 0.5 \cos(2.1t)) \\ y = \sin(t) - 0.5 \cos(2.1t) \end{cases} \quad (0 \leq t \leq 20\pi). \tag{11}$$

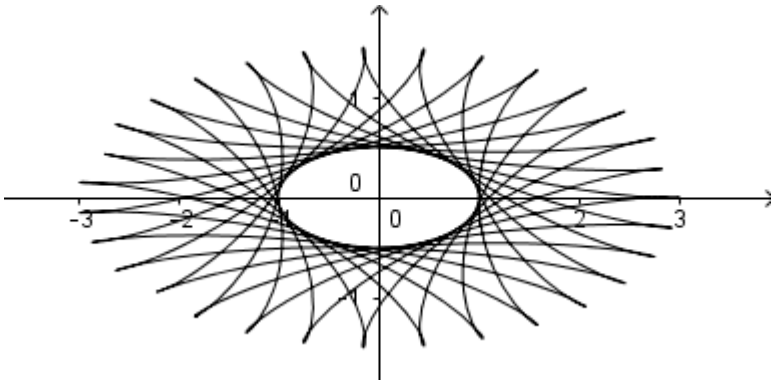


Рис. 12. Вывод графика функции, заданной параметрически в виде (11)

Решение. В соответствии с (9) это можно сделать командой `Curve[2·(cos(t)+0.5·cos(2.1·t)), sin(t)-0.5·sin(2.1·t), t, 0, 20·π]`.

Пример 13. Вывести график астроида

$$\begin{cases} x = 2 \sin^3(t) \\ y = 2 \cos^3(t) \end{cases} \quad (0 \leq t \leq 2\pi). \quad (12)$$

Решение. В соответствии с (9) это можно сделать командой `Curve[2·sin(t)^3, 2·cos(t)^3, t, 0, 2·π]`.

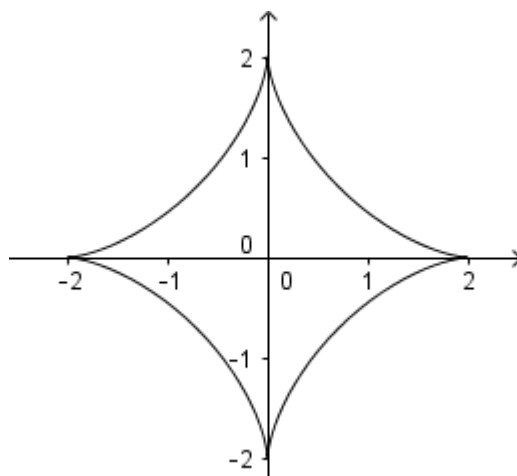


Рис. 13. Вывод астроида примера 13

6.5. Графики функций в полярных координатах

Полярная система координат на плоскости задаётся лучом и фиксированной на нем точкой, которые называются соответственно полярной осью и полюсом, или началом координат. Любая точка плоскости определяется двумя координатами: длиной полярного радиуса ($0 \leq r < \infty$) и значением полярного угла, который по модулю 2π приводится к диапазону: $0 \leq \theta < 2\pi$. Полярный радиус – это отрезок от полюса до точки плоскости. Полярный угол – это угол, на который нужно повернуть полярную ось вокруг полюса против часовой стрелки для того, чтобы попасть в эту точку. По умолчанию углы измеряются в радианах. При поворотах луча по часовой стрелке значение угла считается отрицательным.

На панели “*Настройки*” (вкладка “*Полотно*”, страница “*Сетка*”) можно задать вывод координатной сетки полярной системы координат. Там же задается и стиль ее вывода, то есть шаг сетки для радиуса, шаг сетки для угла, а также вид, толщина и цвет линий сетки. Однако вывод графиков функций в

полярной системе координат напрямую по уравнению невозможен. Рассмотрим два возможных подхода к выводу таких графиков.

6.5.1. Алгебраический подход

Пусть в полярной системе координат требуется вывести график функции $r=f(\theta)$ ($\theta_1 \leq \theta \leq \theta_2$). Выведем координатную сетку полярной системы (не обязательно), но точки (r, θ) графика будем выводить не в полярной, а в обычной декартовой прямоугольной системе, для которой $x=r \cdot \cos(\theta)$ и $y=r \cdot \sin(\theta)$, то есть $(x, y)=(r \cdot \cos(\theta), r \cdot \sin(\theta))$. Иными словами, вывести график $r=f(\theta)$ ($\theta_1 \leq \theta \leq \theta_2$) можно точками $(f(\theta) \cdot \cos(\theta), f(\theta) \cdot \sin(\theta))$. Фактически мы имеем уравнение кривой в параметрическом виде:

$$\begin{cases} x = f(\theta) \cdot \cos(\theta) \\ y = f(\theta) \cdot \sin(\theta) \end{cases} \quad (\theta_1 \leq \theta \leq \theta_2). \tag{13}$$

Но тогда вывод кривой (13) (см. (9)) можно реализовать функцией

$$\text{Curve}[f(\theta) \cdot \cos(\theta), f(\theta) \cdot \sin(\theta), \theta, \theta_1, \theta_2]. \tag{14}$$

Пример 14. Построить график кривой, для которой уравнение в полярных координатах имеет вид $r=2 \cdot \theta$ ($0 \leq \theta \leq 4\pi$). Эту кривую называют спиралью Архимеда.

Решение. В соответствии с (14) нам требуется выполнить команду $\text{Curve}[2 \cdot \theta \cdot \cos(\theta), 2 \cdot \theta \cdot \sin(\theta), \theta, 0, 4 \cdot \pi]$. В результате получим график, представленный на рис. 14.

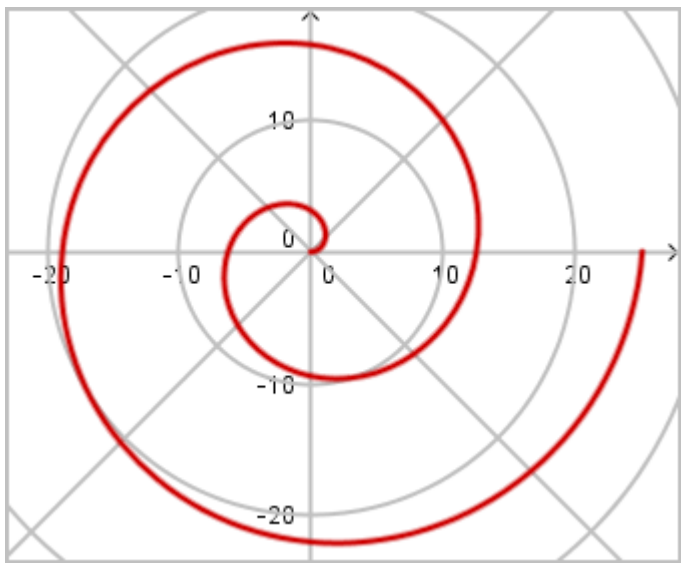


Рис. 14. Спираль Архимеда ($r=2 \cdot \theta, 0 \leq \theta \leq 4\pi$)

Пример 15. Построить график кривой, уравнение в полярных координатах которой имеет вид $r=5\sin(7t/4)$ ($0 \leq t \leq 8\pi$). Эту кривую называют полярной розой.

Решение. В соответствии с (14) нам требуется выполнить команду `Curve[3·sin(7·t/4)·cos(t), 3·sin(7·t/4)·sin(t), t, 0, 8·π]`. В результате получим график, представленный на рис. 15.

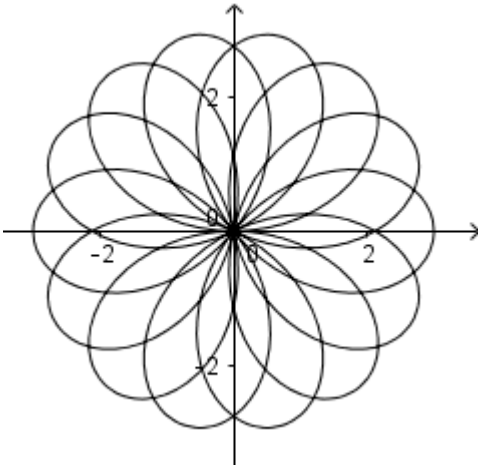


Рис. 15. Полярная роза ($r=5\sin(7t/4)$, $0 \leq t \leq 8\pi$)

Пример 16. Построить график кривой, для которой уравнение в полярных координатах имеет вид $r=e^{\sin(t)} - 2\cos(4t)$ ($0 \leq t \leq 2\pi$).

Решение. В соответствии с (14) нам требуется выполнить команду `Curve[(e^sin(t)-2·cos(4·t))·cos(t), (e^sin(t)-2·cos(4·t))·sin(t), t, 0, 2·π]`. В результате получим график, представленный на рис 16.

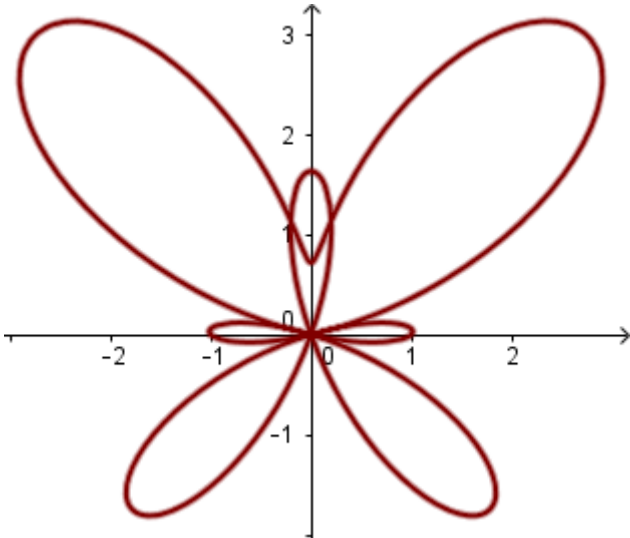


Рис. 16. Бабочка ($r = e^{\sin(t)} - 2\cos(4t)$, $0 \leq t \leq 2\pi$)

Пример 17. Построить график кривой, для которой уравнение в полярных координатах имеет вид $r=\theta/(\theta-1)$ ($-16 \leq \theta \leq 16$).

Решение. В соответствии с (14) нам требуется выполнить команду *Curve*[$((\theta/(\theta-1))\cdot\cos(\theta), (\theta/(\theta-1))\cdot\sin(\theta), \theta, -16, 16]$. В результате получим график, представленный на рис 17. Поскольку при $\theta=1$ мы имеем разрыв, то для того чтобы понять поведение функции в окрестности этой точки, создан ползунок a со значениями от 0 до 3 с шагом 0.01 и начальным значением 2. Кроме того, на кривую помещена точка $A=((a/(a-1))\cdot\cos(a), (a/(a-1))\cdot\sin(a))$. Изменяя значения ползунка a , можно видеть поведение кривой в окрестности точки $\theta=1$ (см. рис. 17).

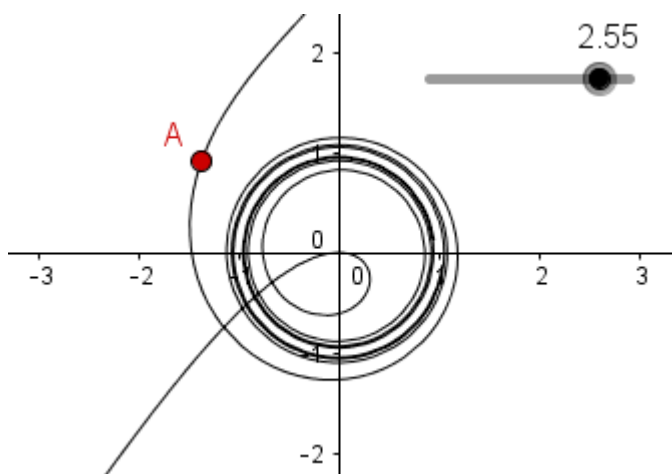


Рис. 17. Полярный график функции $r = \theta / (\theta - 1)$ ($-16 \leq \theta \leq 16$)


6.5.2. Геометрический подход

I. Рассмотрим подход к построению кривой в полярных координатах, основанный на использовании локуса. Пусть кривая, как и раньше, задана в виде $r=f(\theta)$ ($\theta_1 \leq \theta \leq \theta_2$) и с помощью встроенной команды

$$\text{Function}[f(\theta), \theta, \theta_1, \theta_2] \quad (15)$$


мы вывели ее в декартовой прямоугольной системе координат как график функции r одной переменной θ . Зафиксируем на этой кривой произвольную точку A . На таком же графике, но в полярной системе координат, точке A соответствует некоторая точка P . Ясно, что в нашем случае $P=(y[A]\cdot\cos(x[A]), y[A]\cdot\sin(x[A]))$, то есть точкой P определяется некоторый локус, соответствующий перемещению точки A по кривой $r=f(\theta)$ ($\theta_1 \leq \theta \leq \theta_2$), построенной в декартовых координатах. Из сказанного вытекает следующая возможность построения графика функции $r=f(\theta)$ ($\theta_1 \leq \theta \leq \theta_2$) в полярных координатах:

- вывести в декартовой системе координат вспомогательный график $r=f(\theta)$ как функцию одной переменной (*Function*[$f(\theta), \theta, \theta_1, \theta_2$]);
- закрепить на полученной кривой произвольную точку A ;
- через строку ввода ввести точку $P=(y[A]\cdot\cos(x[A]), y[A]\cdot\sin(x[A]))$;

- инструментом “ Локус” вывести график нашей функции в полярных координатах. Для этого требуется произвести щелчок левой кнопкой мыши сначала по точке A , а затем по точке P .

Пример 18. Построить график кривой $r=1-\sin(t)$ ($0 \leq t \leq 2\pi$) в полярных координатах (t, r) .

Решение. В соответствии с приведенным выше предписанием реализуем следующие действия:

- через строку ввода командой `Function[1-sin(t), t, 0, 2π]` строим вспомогательную кривую. На графике она показана штриховой линией;
- закрепляем на полученной кривой произвольную точку A ;
- через строку ввода формируем точку $P=(y[A]*\cos(x[A]), y[A]*\sin(x[A]))$;
- при активном инструменте “ Локус” щелкаем мышью сначала по точке A , а затем по точке P .

Результат этих действий можно видеть на рис. 18. Заметим, что вспомогательный график, как и точку A , можно не показывать. Спрятать их можно, выключив соответствующие радиокнопки на панели “Объекты”.

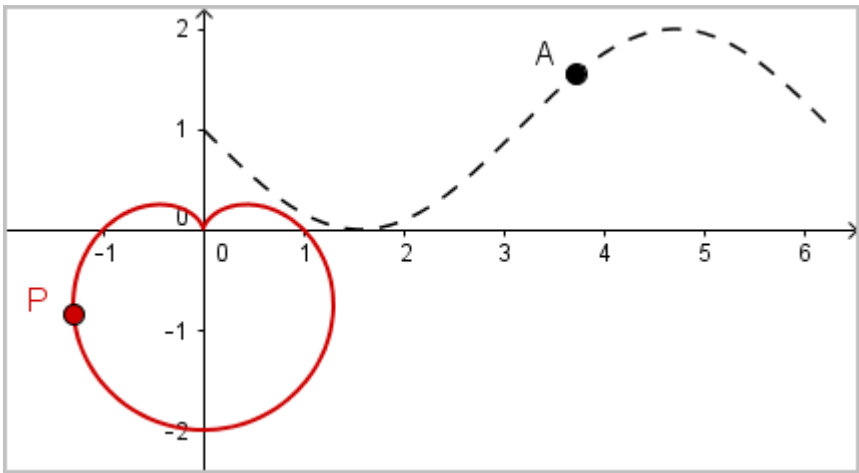



Рис. 18. Кардиоиды ($r=1-\sin(t)$, $0 \leq t \leq 2\pi$)

Замечание. Для вывода кардиоиды можно также было бы использовать и функцию `Curve[(1-sin(t))*cos(t), (1-sin(t))*sin(t), t, 0, π]`.

II. Описанный прием построения графиков в полярной системе координат можно применять и к некоторым функциям, заданным неявно: $G(r, \theta) = 0$. При этом все рассуждения остаются прежними, но вместо (15) необходимо использовать вспомогательную команду `ImplicitCurve[G(r, θ)]`. Заметим, что в `ImplicitCurve` функция G может быть только полиномом от r и θ , причем от r и θ необходимо перейти соответственно к переменным x и y . Областью определения $G(x, y)$ считается вся плоскость, хотя мы имеем дело всегда лишь с ее некоторой ограниченной прямоугольной областью, видимой в окне просмотра.

Пример 19. Построить график кривой $r^2 \cdot \theta = -(2 - r - \theta)^3$ в полярных координатах (θ, r) .

Решение. В соответствии с приведенным выше предписанием реализуем следующие действия:

- с помощью команды `ImplicitCurve[y^2.x+(2-y-x)^3]` строим вспомогательный график. На рис. 19 он представлен штрихпунктирной линией;
- закрепляем на полученной кривой произвольную точку A ;
- через строку ввода формируем точку $P=(y[A]*\cos(x[A]), y[A]*\sin(x[A]))$;
- при активном инструменте “ Локус” щелкаем мышью сначала по точке A , а затем по точке P .

Обратите внимание на 3 последних пункта: они точно такие же, как и в предыдущем примере. Результат указанных действий можно видеть на рис. 19.

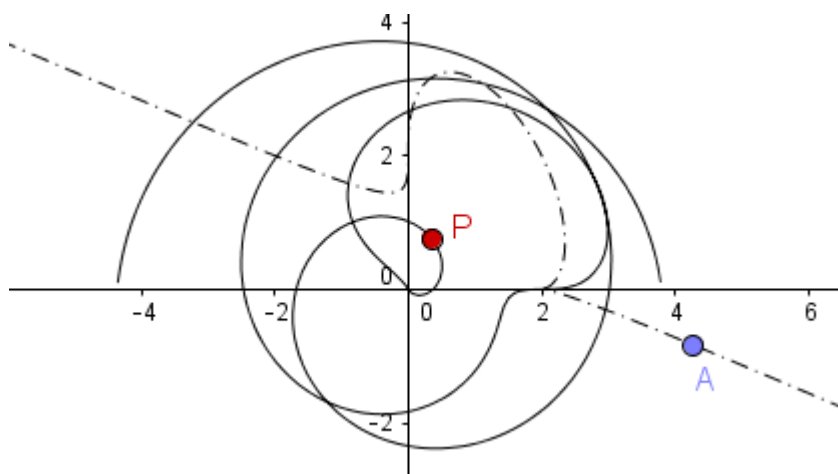



Рис. 19. Полярный график функции $r^2 \cdot \theta = -(2 - r - \theta)^3$ ($-6 \leq \theta \leq 12$)

Пример 20. Построить графики в полярных координатах (θ, r) функций:

$$\begin{aligned}\theta^2 + \theta \cdot r + r^2 &= 15, \\ \theta^2 \cdot r^3 &= (\theta^2 + r^2 - 3)^3.\end{aligned}$$

Решение. Для вывода первого графика выполним следующие действия:

- с помощью команды `ImplicitCurve[x^2+x*y+y^2-15]` строим вспомогательный график. На рис. 20a он представлен штрихпунктирной линией;
- закрепляем на полученной кривой произвольную точку A ;
- через строку ввода формируем точку $P=(y[A]*\cos(x[A]), y[A]*\sin(x[A]))$;
- при активном инструменте “ Локус” щелкаем мышью сначала по точке A и затем по точке P .

Результат указанных действий можно видеть на рис. 20a.

Вывод второго графика, показанного на рис 20b, проводится по полной аналогии.

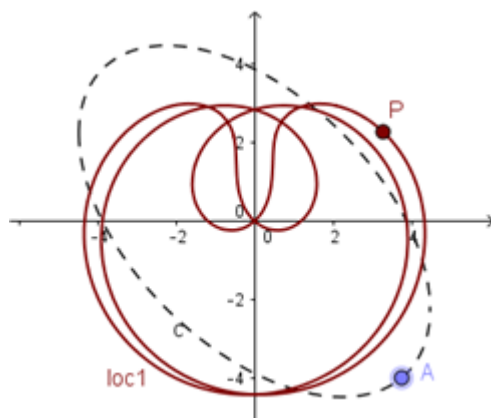


Рис. 20a. Полярный график функции $\theta^2 + \theta \cdot r + r^2 = 15$

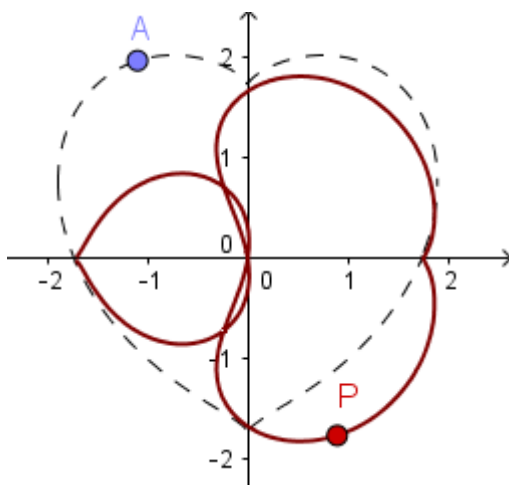


Рис. 20b. Полярный график функции $\theta^2 \cdot r^3 = (\theta^2 + r^2 - 3)^3$

Пример 21. Построить график функции $\theta^2 \cdot r^2 = (2 + \theta + r)^3$ в полярных координатах (θ, r) .

Решение. Выполняя действия, аналогичные тем, которые реализованы в примере 19, получим следующий график.

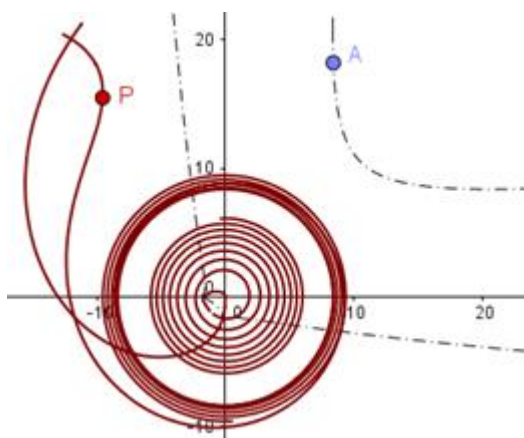


Рис. 21a. Полярный график функции $\theta^2 \cdot r^2 = (2 + \theta + r)^3$

6.6. Специальные графики

6.6.1. Гистограммы

Для вывода графиков типа “гистограмма” или, по-другому, столбчатая диаграмма имеются команды *BarChart*, *Histogram* и *HistogrammRight*. Команда *BarChart* позволяет проводить индивидуальное форматирование баров, команды *Histogram* и *HistogrammRight* дают возможность выводить бары различной ширины.

6.6.1.1. Вывод гистограмм командой *BarChart*

Команда *BarChart* имеет несколько разновидностей синтаксиса:

$$\text{BarChart}[lis1, lis2], \quad (16)$$

$$\text{BarChart}[lis, wi, \mu], \quad (17)$$

$$\text{BarChart}[lis1, lis2, wi], \quad (18)$$


$$\text{BarChart}[a, b, \text{expr}(v), v, v1, v2], \quad (19)$$

$$\text{BarChart}[a, b, \text{expr}(v), v, v1, v2, \text{step}]. \quad (20)$$

Опишем аргументы и выполнение каждой из команд (16)-(20) в отдельности. По этим командам выводятся вертикальные полосы (бары) с основаниями на оси абсцисс. Каждый бар может иметь индивидуальный стиль вывода (цвет, заполнение рисунком и т. п.). Точкой привязки отдельных баров к оси абсцисс считаются середины их сторон, лежащих на этой оси. Если ширина бара равна нулю, то бар выводится в виде вертикального отрезка.

I. *BarChart* в форме (16). Здесь $lis1=\{x1, x2, \dots, xk\}$ – список абсцисс, $lis2=\{y1, y2, \dots, yk\}$ – список ординат для соответствующих абсцисс из $lis1$. Упорядоченности значений $lis1$ не требуется. Значения в $lis2$ могут быть отрицательными. Каждой паре (xs, ys) ($s=1, 2, \dots, k$) соответствует один бар. Если $x2 > x1$, то по (16) бары выводятся в точках $(0, xs)$, имеют одну и ту же ширину $wi=x2-x1$, а их высоты равны ys ($s=1, 2, \dots, k$). Если $x2 \leq x1$, то по (16) бары выводятся в точках $(0, xs+|x2-x1|/2)$, имеют нулевую ширину, а их высоты равны ys ($s=1, 2, \dots, k$).

В дополнение к (16) можно использовать команду *DataFunction*[$lis1, lis2$], по которой выводится ломанная со звеньями между соседними точками: $(x1, y1), (x2, y2), \dots, (xk, yk)$. Сами точки не выводятся и не формируются.

Примеры гистограмм, формируемых командами (16), приведены на рис. 22a, 22b и 22c. Надписи на графиках формировались инструментом “ Текст” как *LaTeX*-формулы, что потребовало вводить фигурные скобки в виде

“\{” и “\}” (без кавычек). Это же замечание касается и надписей на остальных графиках данного пункта.

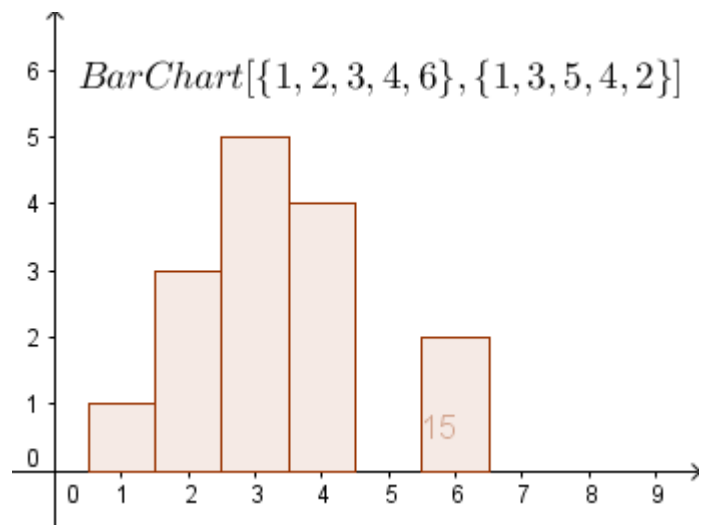


Рис. 22а. Вывод гистограмм (*BarChart*, синтаксис 16)

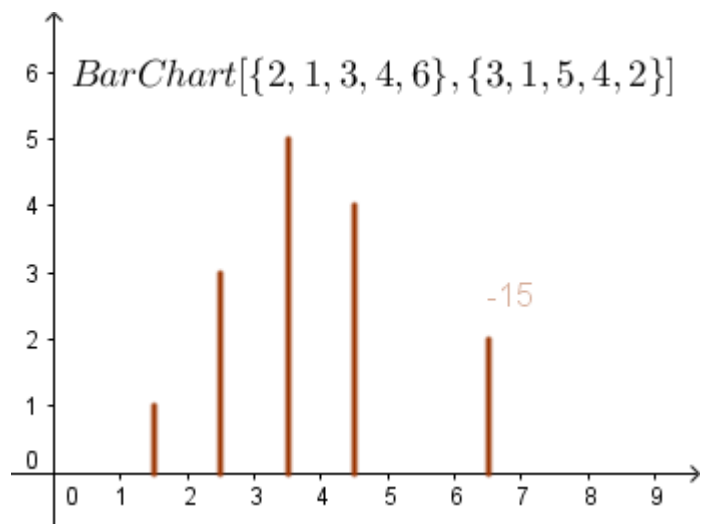


Рис. 22b. Вывод гистограмм (*Barchart*, синтаксис 16)

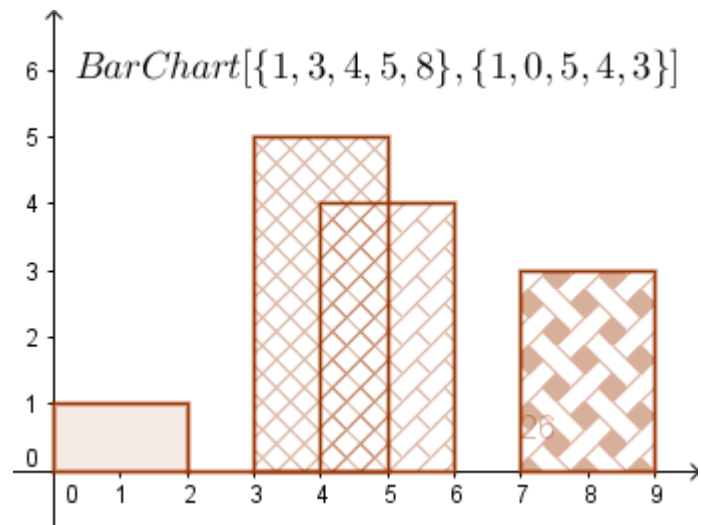


Рис. 22с. Вывод гистограмм (*BarChart*, синтаксис 16)

II. *BarChart* в форме (17). Здесь $lis=\{x_1, x_2, \dots, x_k\}$ – список абсцисс, w_i – ширина баров, μ – необязательный масштабный множитель для высот баров. Упорядоченности значений lis не требуется. Величина μ может быть отрицательной. Выполняется (17) так. Из списка lis формируются: $lis1$ – список различных значений из lis и список $lis2$ – соответствующих частот этих значений. Бары выводятся аналогично (16), но заданной ширины w_i , а если присутствует μ , то частоты еще и умножаются на μ .

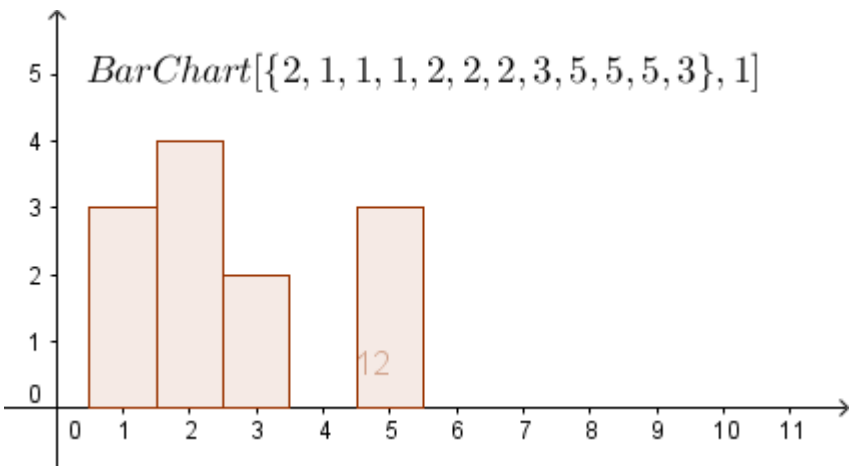


Рис. 23а. Вывод гистограмм (*BarChart*, синтаксис 17)

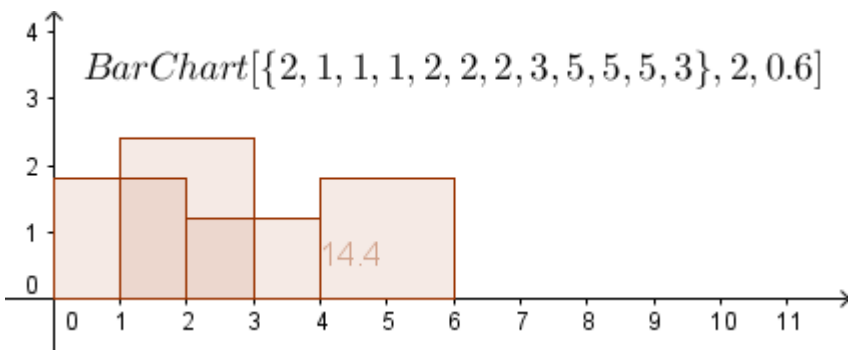


Рис. 23б. Вывод гистограмм (*BarChart*, синтаксис 17)

III. *BarChart* в форме (18). Здесь $lis1=\{x_1, x_2, \dots, x_k\}$ – список абсцисс, $lis2=\{y_1, y_2, \dots, y_k\}$ – список ординат для соответствующих абсцисс из $lis1$, w_i – ширина баров. В данном случае бары выводятся в точках $(0, x_s)$ ($s=1, 2, \dots, k$), имеют одну и ту же ширину w_i , а их высоты равны y_s ($s=1, 2, \dots, k$) (см. рис. 24). Отметим, что если бары имеют нулевую ширину, то в отличие от случая I они все равно выводятся в точках $(0, x_s)$ ($s=1, 2, \dots, k$).

IV. *BarChart* в форме (19). В данном случае аргументы *BarChart* таковы: a, b – граничные точки отрезка, на котором выводится гистограмма; v – переменная; $expr(v)$ – выражение от переменной v ; v_1, v_2 ($v_1 < v_2$) – начальное и конечное значения v . Выражение $expr(v)$ используется для вычисления высоты баров при изменении v от v_1 до v_2 с шагом 1. При $a < b$ ширина баров равна $(b-a)/(v_2-v_1+1)$, при $a \geq b$ ширина баров равна нулю.

Пример гистограммы, выводимой по *BarChart* в форме 19, приведен на рис. 25. Взяв $a=-0.5+v1$, $b=v2+0.5$, мы обеспечили единичную ширину баров и их вывод в целочисленных точках 0, 1, ..., 10.

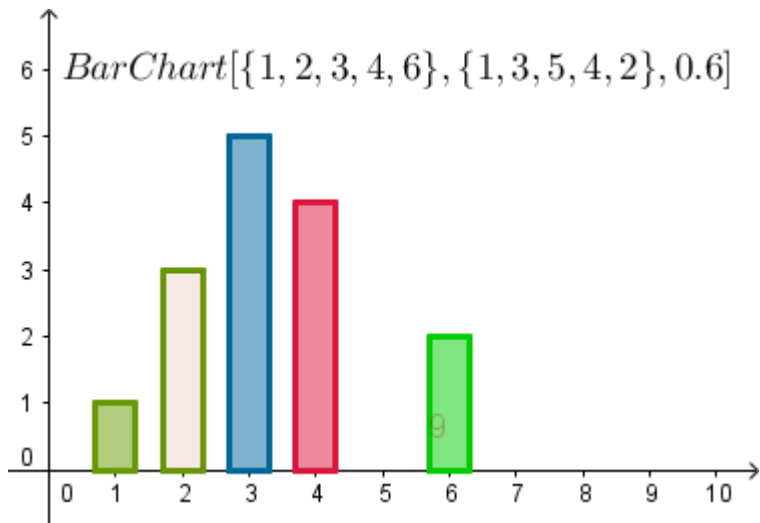


Рис. 24. Вывод гистограмм (*BarChart*, синтаксис 18)

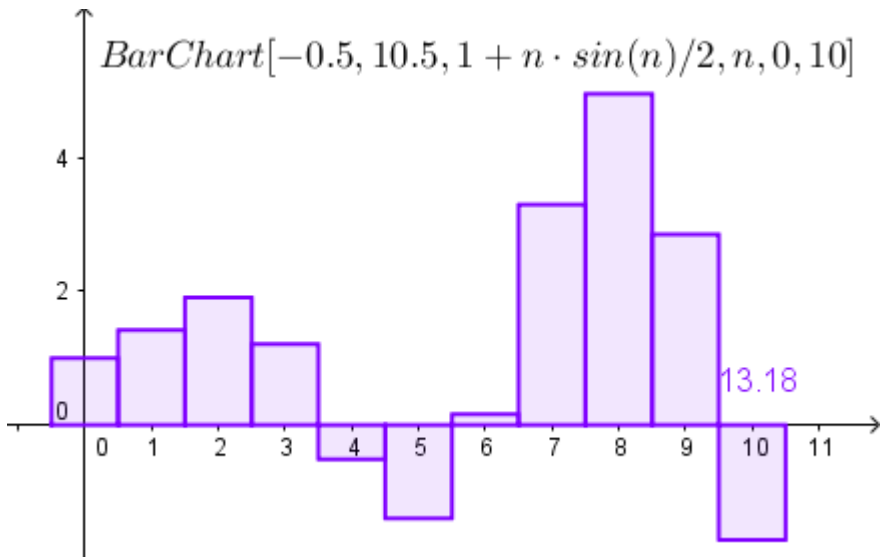



Рис. 25. Вывод гистограмм (*BarChart*, синтаксис 19)

V. *BarChart* в форме (20). В данном случае первые 6 аргументов *BarChart* такие же, как в IV. Последний аргумент *step* задает шаг для изменения переменной v . Выражение $expr(v)$ используется для вычисления высоты баров при изменении v от $v1$ до $v2$ с шагом *step*. При $a < b$ ширина баров равна $(b-a)/(\text{floor}((v2-v1)/\text{step})+1)$, при $a \geq b$ ширина баров равна нулю.

Пример гистограммы, выводимой по *BarChart* в форме 20, приведен на рис. 26. Для уяснения роли аргументов a , b и *step* были созданы 3 ползунка:

- a – начальное значение – -0.5, конечное значение – 16, шаг – 0.5;
- b – начальное значение – -0.5, конечное значение – 16, шаг – 0.5;
- *step* – начальное значение – 0.5, конечное значение – 2, шаг – 0.1.

Несколько слов про надпись графика. Как обычно, она сформирована с помощью инструмента “ Текст” как *LaTeX*-формула (знак умножения введен как команда `\cdot`). Далее, чтобы при перемещении движков у ползунков происходила актуализация в надписи значений a , b и $step$, эти аргументы вводились из списка объекты. Манипулируя движками ползунков, мы можем лучше понять смысл аргументов в (20).

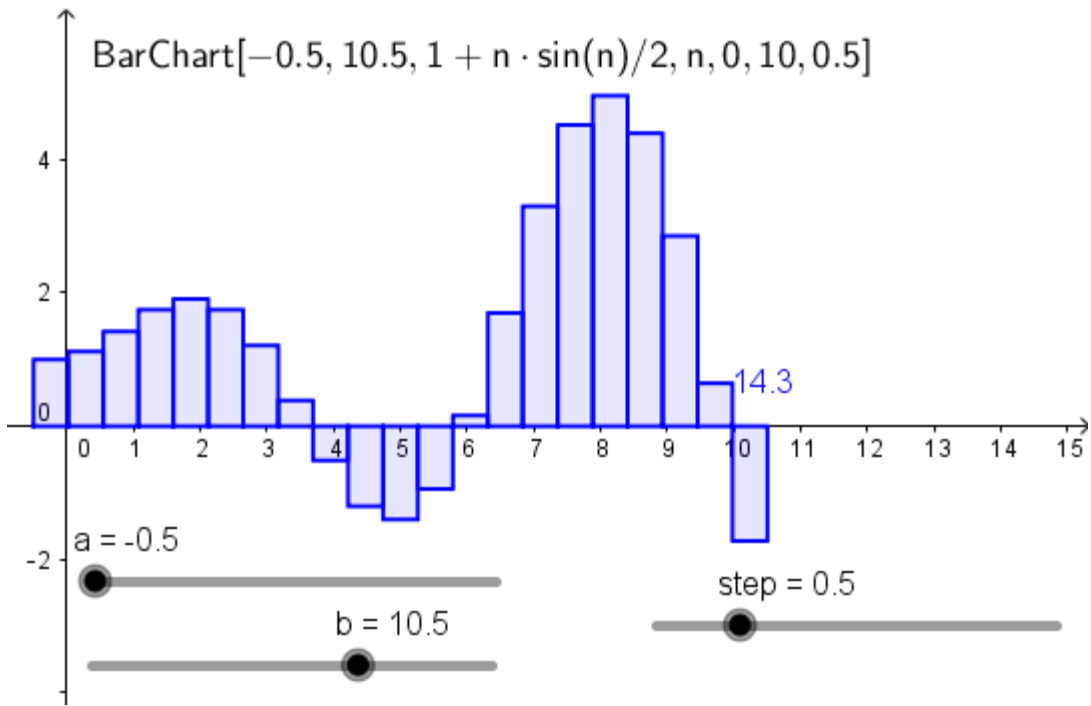


Рис. 26. Вывод гистограмм (*BarChart*, синтаксис 20)

6.6.1.2. Вывод гистограмм командой *Histogram*

Команда *Histogram* имеет несколько разновидностей синтаксиса:

$$Histogram[lis1, lis2], \tag{21}$$

$$Histogram[lis1, data, bool, \mu], \tag{22}$$

$$Histogram[bool1, lis1, data, bool, \mu]. \tag{23}$$

Опишем аргументы и выполнение каждой из команд (21)-(23) в отдельности.

I. *Histogram* в форме (21). Здесь $lis1=\{b_1, b_2, \dots, b_k, b_{k+1}\}$ – список абсцисс, задающий ширину $b_{s+1}-b_s$ ($s=1, 2, \dots, k$) последовательных баров, $lis2=\{h_1, h_2, \dots, h_k\}$ – список высот соответствующих баров. Таким образом по (21) выводится гистограмма, содержащая k баров, построенных на отрезках $[b_{s+1}, b_s]$, и имеющих высоты h_s ($s=1, 2, \dots, k$) (см. рис. 27).

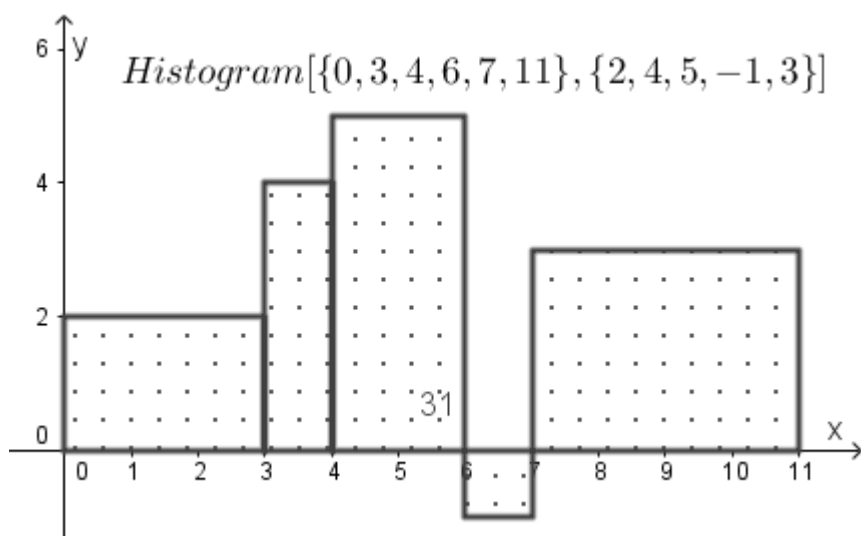


Рис. 27. Вывод гистограмм (*Histogram*, синтаксис 21)

II. *Histogram* в форме (22). Здесь $lis1 = \{b_1, b_2, \dots, b_k, b_{k+1}\}$ – список абсцисс, задающий ширину $b_{s+1} - b_s$ ($s=1, 2, \dots, k$) последовательных баров; *data* – список данных (абсцисс); необязательное булево значение *true* или *false*; μ – необязательный множитель. По умолчанию $bool=true$ и $\mu=1$. Аргументы *bool* и μ используются при определении высот баров. Обычно элементы *lis1* упорядочены по возрастанию, и описание проводится именно для этого случая. Упорядоченности значений *data* не требуется. Выполняется (22) так:

- подсчитываются количества n_1, n_2, \dots, n_k элементов из списка *data*, входящих соответственно в промежутки:

$$[b_1, b_2), [b_2, b_2), \dots, [b_{k-1}, b_k), [b_k, b_{k+1}] \quad (24)$$

Заметим, что левые концы промежутков в (24) принадлежат этим промежуткам, а правые – только последнему промежутку.

- далее на промежутках (24) строятся бары с высотами

$$h_s = \begin{cases} n_s & , \text{ если } bool = false \\ n_s \cdot \frac{\mu}{b_{s+1} - b_s} & , \text{ если } bool = true \end{cases} \quad (s = 1, 2, \dots, k). \quad (25)$$

На рис. 28a, 28b и 28c приведены гистограммы, выведенные соответственно командами *Histogram*[*lis1*, *data*, *false*], *Histogram*[*lis1*, *data*, *true*] и *Histogram*[*lis1*, *data*, *true*, 1/2], где $lis1 = \{1, 4, 5, 7, 12\}$, $data = \{1, 2, 3, 4, 4.1, 4.5, 4.6, 6, 7, 8, 9, 9.5, 10\}$.

III. *Histogram* в форме (23). Здесь к команде *Histogram* в синтаксисе (22) добавлен еще один булевский аргумент *bool1*. При $bool1=false$ (22) и (23) выполняются одинаково. При $bool1=true$ выполнение (23) проводится аналогично (22), но вычисленные значения h_s ($s=1, 2, \dots, k$) не являются высотами

баров, а используются для их кумулятивного вычисления, то есть высота бара s ($s=1, 2, \dots, k$) равна h_s плюс высота предыдущего бара, если он имеется.

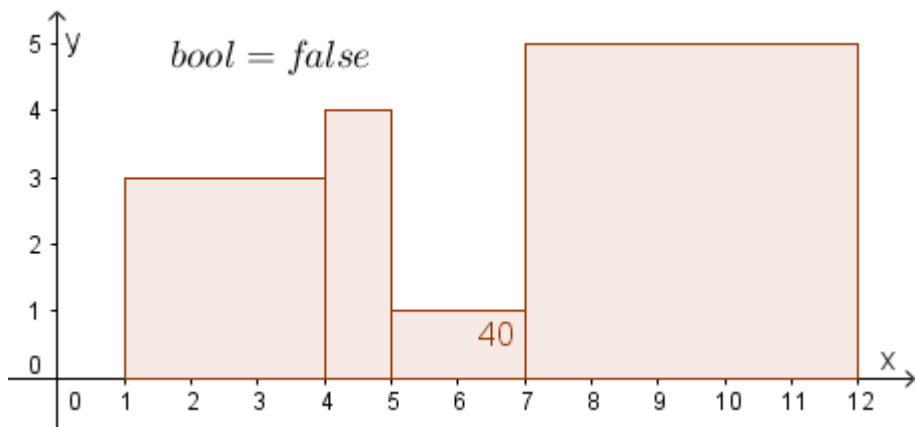


Рис. 28а. Вывод гистограмм (*Histogram*, синтаксис 22)

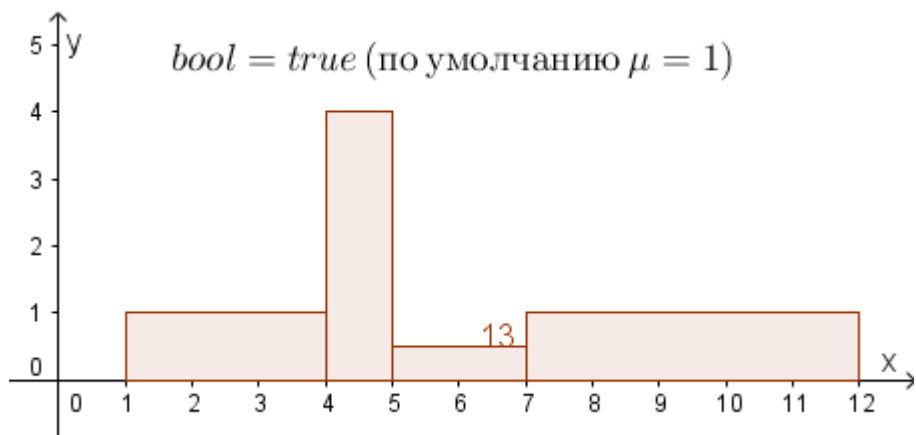


Рис. 28b. Вывод гистограмм (*Histogram*, синтаксис 22)

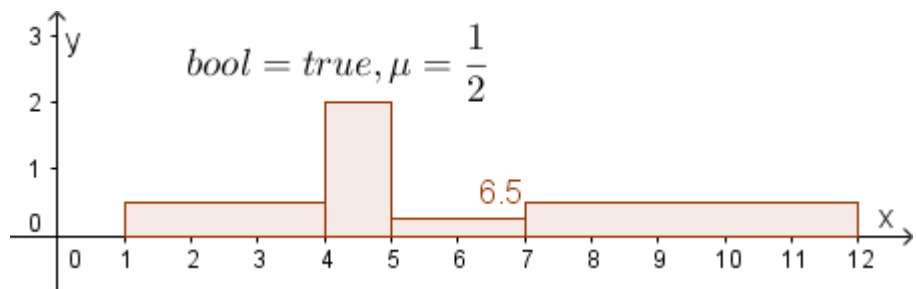


Рис. 28с. Вывод гистограмм (*Histogram*, синтаксис 22)

6.6.1.3. Вывод гистограмм командой *HistodramRight*

Команда *HistogramRight* имеет несколько разновидностей синтаксиса:

$$HistogramRight[*lis1*, *lis2*], \tag{26}$$

$$HistogramRight[*lis1*, *data*, *bool*, μ], \tag{27}$$

$$HistogramRight[*bool1*, *lis1*, *data*, *bool*, μ]. \tag{28}$$

Команды (26) и (21) выполняются одинаково. Команды (27)-(28) выполняются аналогично соответствующим командам (22)-(23) с единственным отличием. Подсчитываются количества n_1, n_2, \dots, n_k элементов из списка *data*, входящих не в промежутки (24), а в промежутки

$$[b_1, b_2], (b_2, b_2], \dots, (b_{k-1}, b_k], (b_k, b_{k+1}].$$

Иными словами, здесь правые концы промежутков принадлежат этим промежуткам, а левые – только первому промежутку.

6.6.2. Диаграммы размаха

Диаграммы размаха, называемые также боксовыми диаграммами, или рамочными графиками квартилей, служат для визуализации итоговых результатов обработки одномерных числовых статистических данных, обеспечивая простой и быстрый их обзор. Сделав данные наглядными, можно обнаружить скрытые в них закономерности или провести сравнительный анализ полученных характеристик нескольких выборок. Диаграммы размаха были предложены в 1977 г. американским математиком Дж. В. Тьюки (*John Wilder Tukey*)¹. Рассматриваемые ниже разновидности команды *BoxPlot* для вывода диаграмм размаха используют или выборки статистических данных в виде числовых списков, или характеристики таких данных. Поэтому для дальнейшего изложения нам необходимо остановиться на некоторых понятиях.

1. *Медиана* – одна из характеристик среднего значения выборки. Медианой называют значение M , разбивающее выборку на две части так, что половина наблюдений оказывается меньше или равна M , а половина – больше или равна M . Вычисляется медиана так:

- изучаемая выборка из n элементов сортируется в порядке неубывания ее членов. Получаемая при этом последовательность называется вариационным рядом, или порядковыми статистиками;

- если n нечетно, то медиана равна центральному элементу вариационного ряда – $x_{(n+1)/2}$. Если n четно, то медиана равна среднему арифметическому двух центральных элементов вариационного ряда – $(x_{n/2} + x_{n/2+1})/2$.

2. *p-квантиль*, или, по-другому, квантиль порядка p ($0 \leq p \leq 1$) – это такое число K_p , меньше или равно которому в выборке имеется приблизительно $Q1=p \cdot 100\%$ элементов. Например, 0.25-квантиль – это число, меньше или равно которому имеется примерно 25% выборки; 0.75-квантиль – число, меньше или равно которому имеется примерно 75% выборки. Вместо *p-квантилей* иногда используют *100·p-процентили*;

¹ Среди программистов Дж. В. Тьюки больше известен как автор двух компьютерных терминов: *bit* (1946) и *software* (1958).

3. *нижний квартиль* $Q1$, или нижняя квартильная точка – это 0.25-квантиль, или 25-процентиль;

4. *средний квартиль* $Q2$, или средняя квартильная точка – это 0.50-квантиль, или 50-процентиль. Фактически $Q2$ – это медиана;

5. *верхний квартиль* $Q3$ или верхняя квартильная точка – это 0.75-квантиль, или 75-процентиль. Среднее значение нижнего и верхнего квартилей приблизительно равно дисперсии выборки;

6. *квартильный размах*, или *интерквартильный размах* – интервал значений выборки, содержащий 50% ее значений между нижним и верхним квартилями;

7. *выброс* – это резко отклоняющееся в ту или в другую сторону значение в выборке. При этом выбросами считаются все значения выборки, большие $Q3+1.5\cdot(Q3-Q1)$, и меньшие $Q1-1.5\cdot(Q3-Q1)$, то есть выбросы лежат на краях вариационного ряда;

8. наименьшее значение выборки, или, по-другому, *нижний ус* (*begin*). Может рассчитываться как с учетом выбросов, так и без них;

9. наибольшее значение выборки, или, по-другому, *верхний ус* (*end*). Может рассчитываться как с учетом выбросов, так и без них.

Диаграмма размаха – это некоторая схема, которая наглядно демонстрирует следующие характеристики выборки: *begin*, $Q1$, M , $Q3$, *end* и, возможно, выбросы. На рис. 29 эти характеристики, вычисленные для выборки $\{1, 2, 2, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 8, 9.6\}$, показаны на горизонтальной диаграмме размаха. В *GeoGebra* выводятся именно такие диаграммы и делается это командой *BoxPlot*, возвращающей медиану. Построение диаграммы возможно с учетом или без учета выбросов. По умолчанию по ординате диаграммы рисуются в пределах от -1 до 1. Однако вдоль этой оси их можно масштабировать и сдвигать.

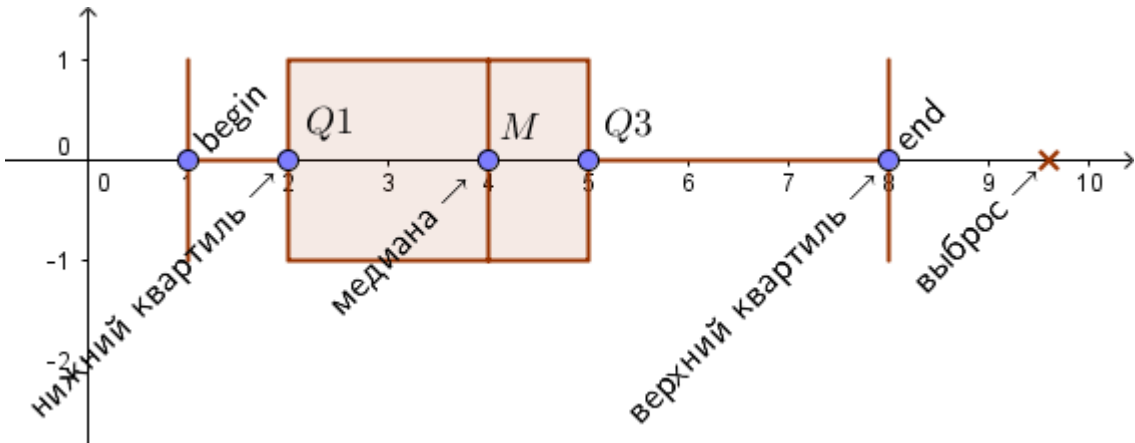


Рис. 29. Характеристики выборки, демонстрируемые диаграммой размаха (с учетом выбросов)

Для построения диаграмм размаха используются следующие разновидности синтаксиса команды *BoxPlot*:

BoxPlot[*yshift*, *уμ*, *data*], (29)

BoxPlot[*yshift*, *уμ*, *begin*, *Q1*, *M*, *Q3*, *end*], (30)

BoxPlot[*yshift*, *уμ*, *data*, *boolout*], (31)

BoxPlot[*yshift*, *уμ*, *da*, *freq*, *boolout*]. (32)

В (29)-(32) аргументы *yshift* и *уμ* – это соответственно величина смещения и масштабирующий множитель для графика вдоль оси ординат. Благодаря аргументу *yshift* в одной системе координат можно вывести несколько диаграмм размаха. Опишем остальные аргументы, а также выполнение каждой из разновидностей (29)-(32) команды *BoxPlot* в отдельности.

I. *BoxPlot* в форме (29). Здесь *data* – список чисел, представляющих собой выборку статистических данных, по которым и строится график. Упорядоченности *data* не требуется. Выполняется (29) так. По *data* строятся следующие характеристики выборки:

- *begin* – нижний ус (без учета выбросов);
- *end* – верхний ус (без учета выбросов)
- *Q1*, *Q3* – нижний и верхний квартили;
- *M* – медиана.

Далее по полученным характеристикам выводится диаграмма размаха без учета выбросов. Например, если *data*={1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 8, 9.6} и выполняется *BoxPlot*[0, 1, *data*], то выводится график, представленный на рис. 30 (сравните с графиком рис. 29, построенного для тех же данных).

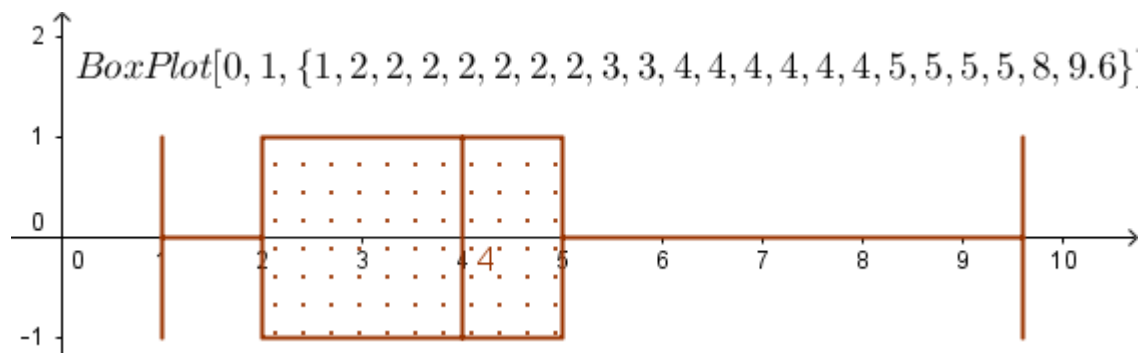


Рис. 30. Диаграмма размаха, выведенная по *BoxPlot* в синтаксисе (29)

II. *BoxPlot* в форме (30). Здесь вместо данных *data* аргументами непосредственно задаются характеристики выборки *begin*, *Q1*, *M*, *Q3* и *end*. По ним и формируются диаграмма размаха без учета выбросов и без каких-либо предварительных вычислений (см. рис. 31).

III. *BoxPlot* в форме (31). Здесь первые три аргумента в *BoxPlot* те же самые, что и в (29). Четвертый булевский аргумент *boolout* определяет размещение усов диаграммы. А именно, при *boolout*=*false* по (31) выводится диаграмма

размаха без учета выбросов, то есть такая же, как и по (29). При *boolout=true* по (31) выводится диаграмма размаха с учетом выбросов.

Диаграмма рис. 32 выведена для тех же данных, что и диаграмма рис. 30, но в данном случае выбросы учитывались, и потому значение 9.6, помеченное на графике крестиком, оказалось правее верхнего уса. Иными словами, верхний ус сдвинулся со значения 9.6 на предыдущее значение 8, которое к выбросам уже не относится: $Q3-1.5\cdot(Q3-Q1)=5-1.5\cdot(5-2)=0.5$, $Q3+1.5\cdot(Q3-Q1)=5+1.5\cdot(5-2)=9.5$, $0.5 \leq 8 \leq 9.5$.

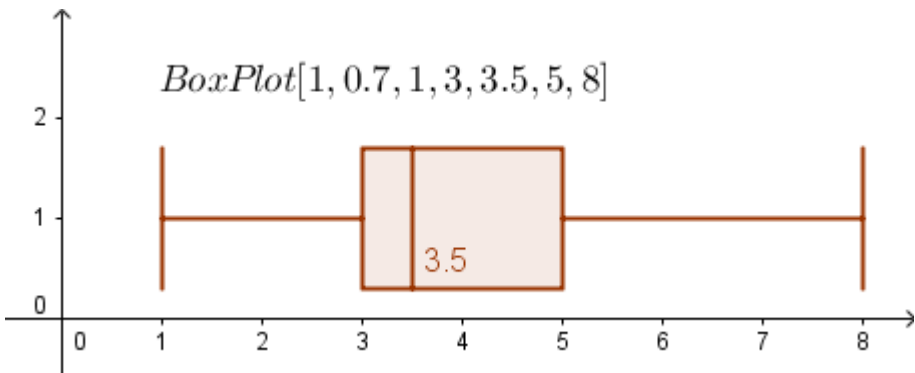


Рис. 31. Диаграмма размаха, выведенная по *BoxPlot* в синтаксисе (30)

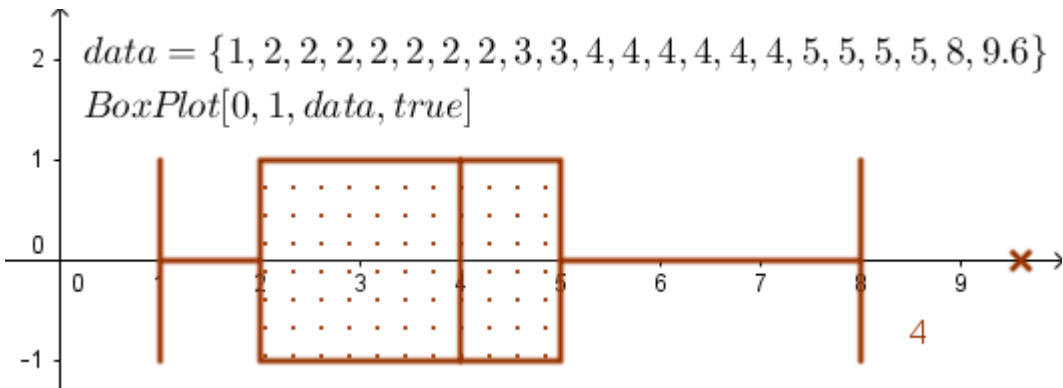


Рис. 32. Диаграмма размаха, выведенная по *BoxPlot* в синтаксисе (31)

IV. *BoxPlot* в форме (32). В данном случае первый и последний аргументы такие же, как и в (31), а аргументу *data* из (31) соответствуют два аргумента *da* и *freq* из (32), где *da* – список всех уникальных элементов выборки, *freq* – список частот соответствующих элементов из *da*. Подобное представление выборки удобно при ее значительной длине и небольшом количестве различных элементов. Выполняется *BoxPlot* в формах (32) и (31) одинаково, то есть подсчитываются требуемые характеристики выборки и т. д.

Если *BoxPlot* для предыдущего примера переписать в форме (32), то он будет выглядеть так: `BoxPlot[0, 1, {1, 2, 3, 4, 5, 8, 9.6}, {1, 7, 2, 6, 4, 1, 1}, true]`.

Замечание. Часто диаграмму размаха представляют в форме, у которой на уровне медианы горизонтальные стороны бокса скошены уголками так, как это показано на рис. 33. В *GeoGebra* такой вывод можно реализовать следую-

щим образом. Вывести обычную диаграмму без уголков. По ней, как по шаблону, провести соответствующие отрезки линий и создать иные объекты, то есть нарисовать требуемую диаграмму с уголками. После чего исходную диаграмму сделать невидимой. Именно так и построена диаграмма рис. 33.

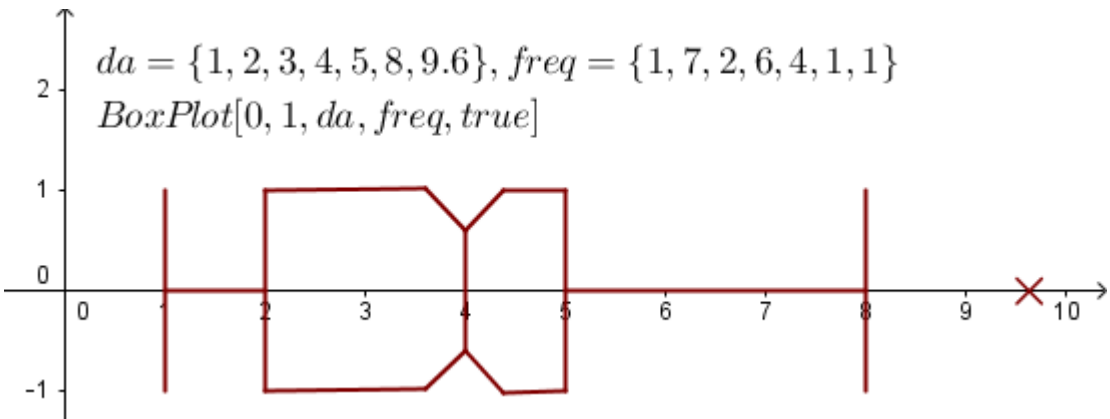


Рис. 33. Диаграмма размаха со скошенными уголками в районе медианы

6.6.3. Частотные точечные графики

К ранее рассмотренным точечным графикам добавим так называемые частотные точечные графики, выводимые командой:

$$DotPlot[lis, boolstack, \mu], \tag{33}$$

где $lis=\{x_1, x_2, \dots, x_k\}$ – список чисел или список строк, $boolstack$ – необязательное булево значение, μ – необязательный масштабный множитель для ординат точек. По умолчанию $boolstack=false$ и $\mu=1$. Аргумент $boolstack$ можно использовать только при числовых списках lis .

I. Пусть lis – числовой список. Тогда при выполнении (33) возможны такие случаи:

a) Аргументы $boolstack$ и μ отсутствуют. Строится список $da=\{a, b, \dots\}$ уникальных и упорядоченных по возрастанию значений lis и список $freq=\{n, m, \dots\}$ частот соответствующих значений из da . По этим спискам формируются и выводятся точки $(a, 1), (a, 2), \dots, (a, n), (b, 1), (b, 2), \dots, (b, m), \dots$ (см. рис. 34).

b) Аргумент μ присутствует. Тогда вместо точек, построенных в случае a , выводятся точки $(a, \mu), (a, 2\mu), \dots, (a, n\mu), (b, \mu), (b, 2\mu), \dots, (b, m\mu), \dots$ (см. рис. 35a).

c) Аргумент $boolstack$ присутствует. Как и в случае a , строятся списки da и $freq$. Далее по этим спискам формируются и выводятся точки $(a, r\cdot\mu), (a, 3r\cdot\mu), \dots, (a, (2n+1)r\cdot\mu), (b, r\cdot\mu), (b, 3r\cdot\mu), \dots, (b, (2m+1)r\cdot\mu), \dots$, где r – поло-

вина установленного диаметра для выводимых точек. Фактически, при отсутствии μ и при $\mu=1$ точки выводятся столбиком друг над другом без промежутков (см. рис. 35b).

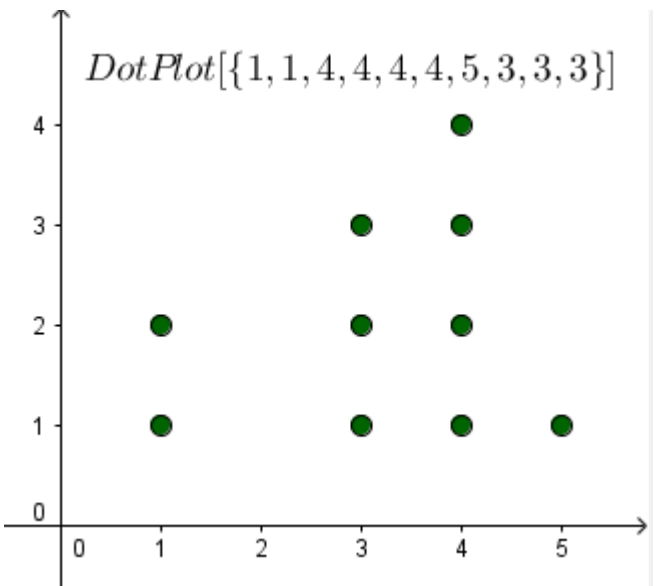


Рис. 34. Вывод частотного точечного графика по числовым данным

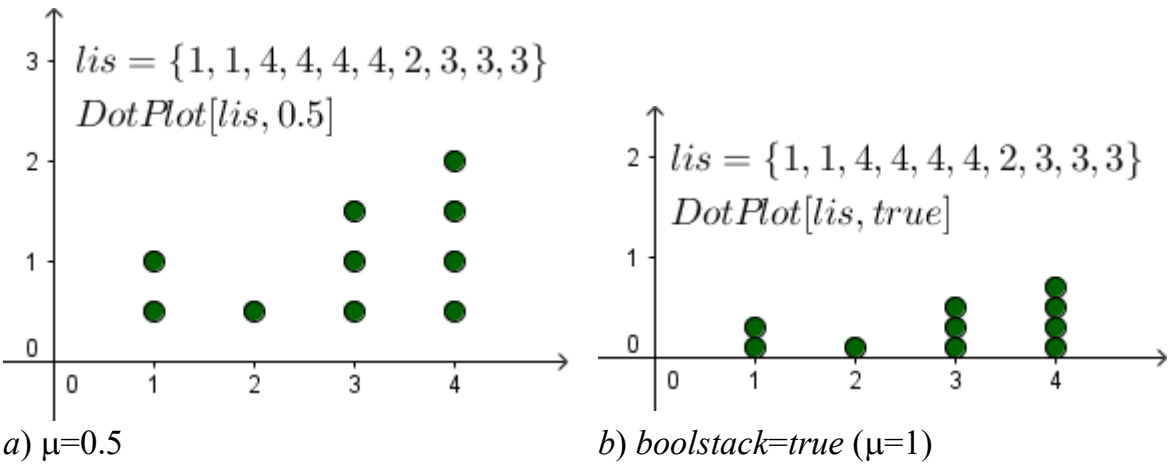


Рис. 35 (a, b). Вывод частотных точечных графиков по числовым данным командой `DotPlot` с дополнительными аргументами

II. Пусть `lis` – список строк. Тогда аргумент `boolstack` присутствовать не должен и при выполнении (33) возможны такие случаи:

a) Аргумент μ отсутствует (то есть $\mu=1$). В этом случае сначала строится список `dastr={a, b, ...}` уникальных строковых элементов `lis` и затем элементы `dastr` лексикографически сортируются в порядке возрастания. Далее по новому списку `dastr` формируется список `da` порядковых номеров элементов `dastr` и список `freq` частот соответствующих элементов `dastr`. Наконец, по `da` и `freq` создаются и выводятся точки графика и делается это так, как описано в пункте I. На рис. 36a представлен пример вывода частотного точечного графика по списку строковых данных. Наклонные надписи создавались через строку ввода дополнительными командами `RotateText["5", 30°]`, `RotateText["мед", 30°]`, `RotateText["сахар", 30°]`.

b) Аргумент μ присутствует. В данном случае списки *da* и *freq* строятся как и в *a*, но точки графика формируются по *da* и $\mu \cdot freq$. На рис. 36b представлен график, построенный по данным примера рис. 36a, но при $\mu=0.5$.

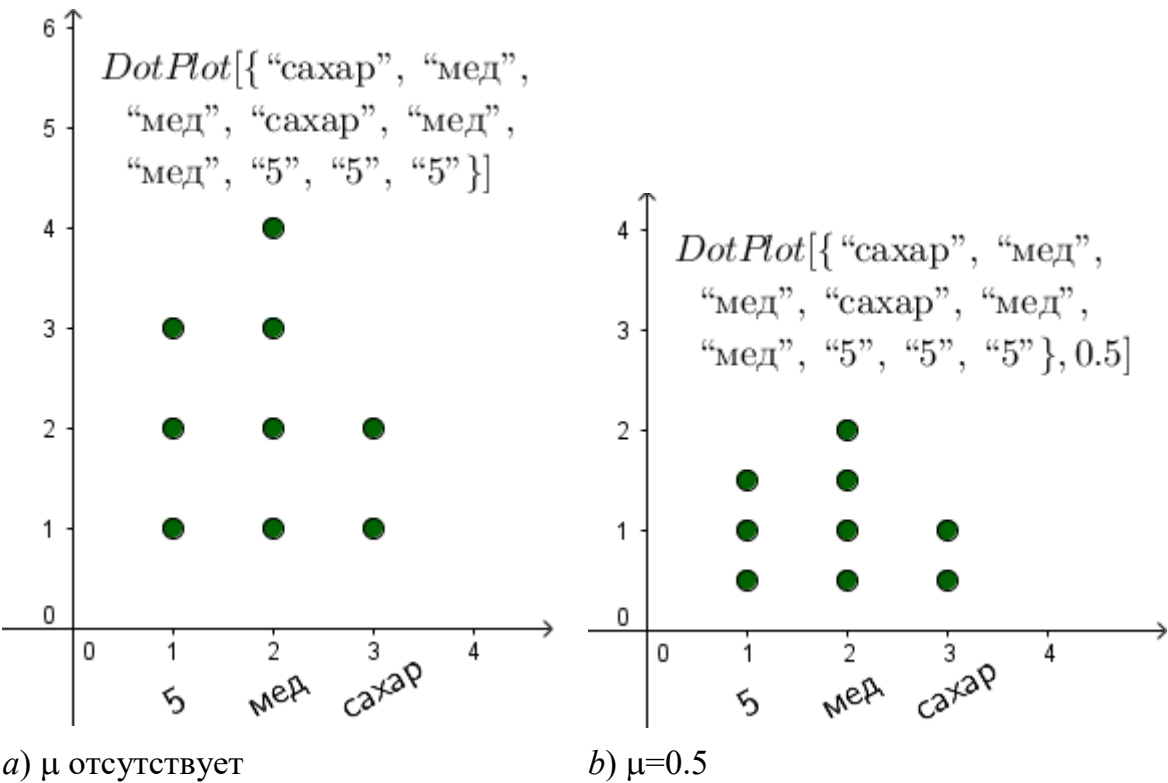


Рис. 36 (a, b). Вывод частотных точечных графиков по строковым данным

6.6.4. Графики частотного полигона

Графики частотного полигона выводятся командой *FrequencyPolygon*, которая имеет следующие разновидности синтаксиса:

$$FrequencyPolygon[lis1, lis2], \tag{34}$$

$$FrequencyPolygon[lis1, data, bool, \mu], \tag{35}$$

$$FrequencyPolygon[bool1, lis1, data, bool, \mu]. \tag{36}$$

Аргументы в (34)-(36) имеют тот же самый смысл, что и в соответствующих командах *Histogram* (см. 2.6.1.2). Поэтому описывать их не будем, а отметим лишь то, что график частотного полигона является ломанной линией, состоящей из отрезков, которые соединяют середины верхних сторон соседних баров в соответствующей гистограмме.

На рис. 37 в одной системе координат одновременно выведены и гистограмма, и частотный полигон. Сделано это вводом команд $lis1=\{0, 3, 4, 6, 7,$

11}, lis2={2, 4, 5, -1, 3}, Histogram[lis1, lis2], FrequencyPolygon[lis1, lis2] (команда в форме (34)). Надпись к графикам сформирована инструментом “ABC Текст” как *LaTeX*-текст вида:

```
lis1=\{0,3,4,6,7,11\}, \; \; lis2=\{2,4,5,-1,3\}, \\\
Histogram[lis1,lis2], \\\
FrequencyPolygon[lis1,lis2]
```

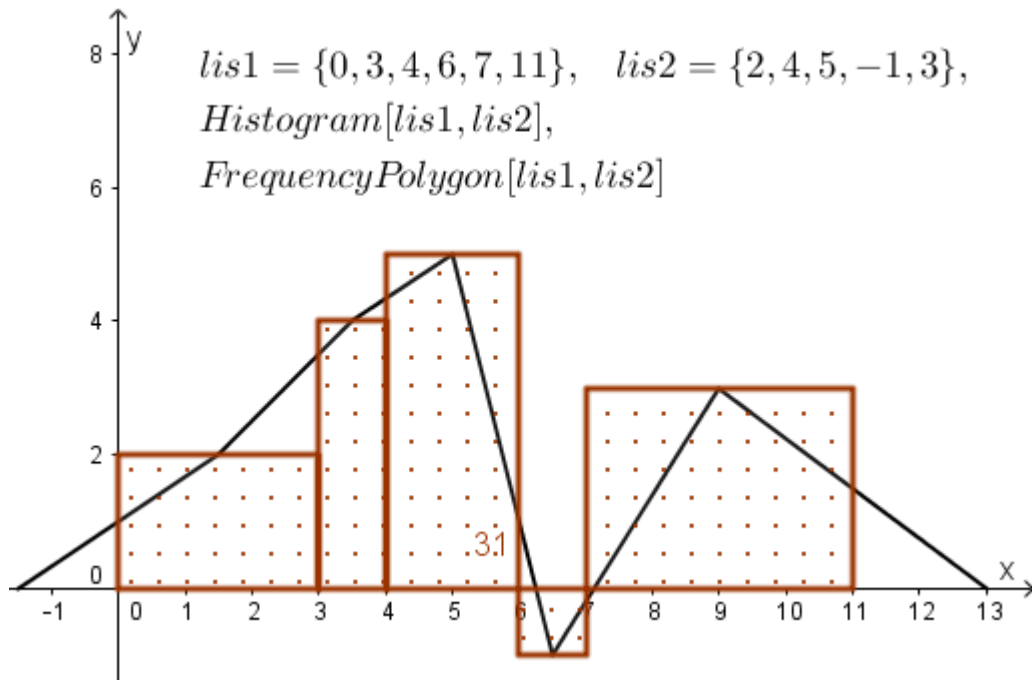


Рис. 37. Вывод гистограммы и частотного полигона

6.6.5. Пошаговые графики

Пошаговые графики выводятся командой *StepGraph*, которая имеет следующие разновидности синтаксиса:

- StepGraph*[lis], (37)
- StepGraph*[lis, boolhv], (38)
- StepGraph*[lis, boolhv, stylepoint], (39)
- StepGraph*[lis1, lis2], (40)
- StepGraph*[lis1, lis2, boolhv], (41)
- StepGraph*[lis1, lis2, boolhv, stylepoint]. (42)

Опишем аргументы и выполнение каждой из команд (37)-(42) в отдельности.

I. *StepGraph* в форме (37). Здесь *lis*={(x₁, y₁), (x₂, y₂), ..., (x_k, y_k)} – список точек. Выполняется команда так. По каждой паре смежных точек (x_s, y_s), (x_{s+1}, y_{s+1}) (s=1, 2, ..., k–1) выводится горизонтальный отрезок между точками

(x_s, y_s) и (x_{s+1}, y_s) ($s=1, 2, \dots, k-1$). Возвращается по (37) сумма ординат точек. Сами точки не прорисовываются.

Заметим, что все графики, представленные на рис. 38a-38j построены при одном и том же списке $lis=\{(1, 1), (3, 2), (4, 3), (5, 1)\}$. График, выведенный по $StepGraph[lis]$, показан на рис. 38a.

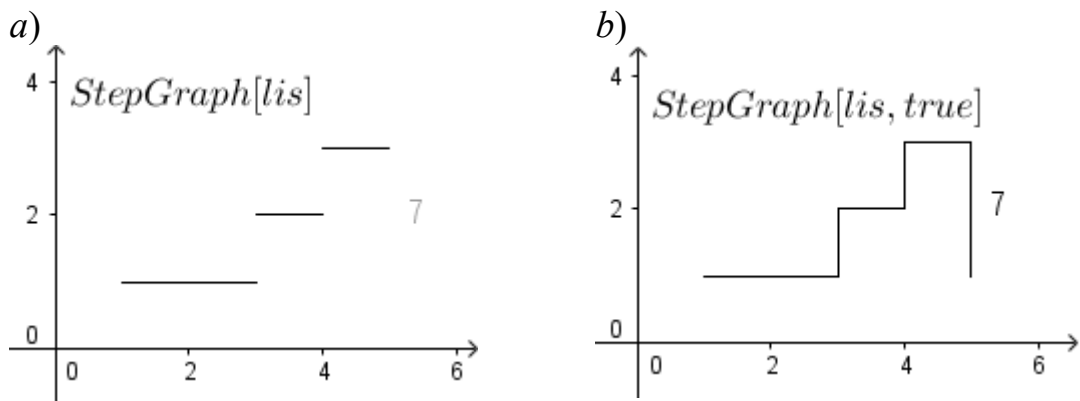
II. *StepGraph* в форме (38). Здесь lis – список точек, как и в I, $boolhv$ – булева переменная. При $boolhv=false$ вывод тот же самый, что и в случае I. При $boolhv=true$ по каждой паре смежных точек (x_s, y_s) , (x_{s+1}, y_{s+1}) выводится горизонтальный отрезок между точками (x_s, y_s) и (x_{s+1}, y_s) , а также вертикальный отрезок между точками (x_{s+1}, y_s) и (x_{s+1}, y_{s+1}) ($s=1, 2, \dots, k-1$). Возвращается по (38) сумма ординат точек. Сами точки не прорисовываются (см. рис. 38b).

III. *StepGraph* в форме (39). Здесь аргументы lis и $boolhv$ такие же, как в II и выводится тот же самый график. Значения *stylepoint* определяют дополнительный вывод точек на концах отрезков и вид этих точек (см. рис. 38c-38j). Возможные значения *stylepoint* и их действие таковы:

- 0 – точки не выводятся;
- 1 – у правых концов отрезков выводятся точки в виде закрашенных кругов;
- 2 – у правых концов отрезков выводятся точки в виде закрашенных кругов, у левых – в виде незакрашенных кругов;
- -1 – у левых концов отрезков выводятся точки в виде закрашенных кругов;
- -2 – у левых концов отрезков выводятся точки в виде закрашенных кругов, у правых – в виде незакрашенных кругов.

Возвращается по (39) сумма ординат точек.

IV-VI. *StepGraph* в форме (40)-(42). В этих командах вместо единого списка точек lis задаются два списка одинаковой длины: $lis1$ – список абсцисс, и $lis2$ – список ординат. Точки строятся по парам элементов, расположенных на одинаковых позициях в этих списках. В остальном выполнение рассматриваемых команд аналогично выполнению соответствующих команд (37)-(39).



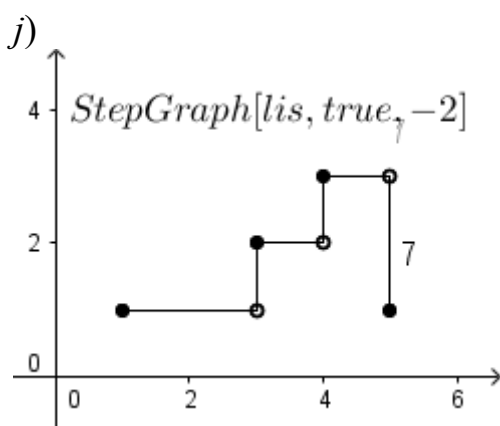
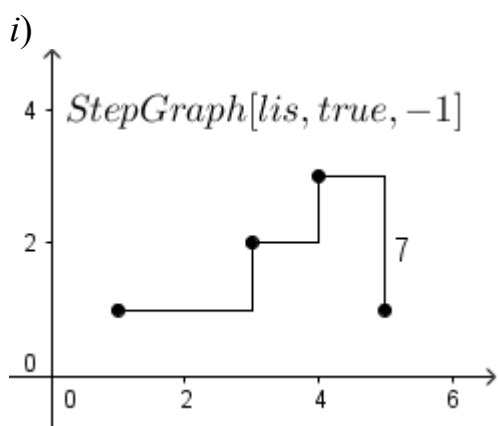
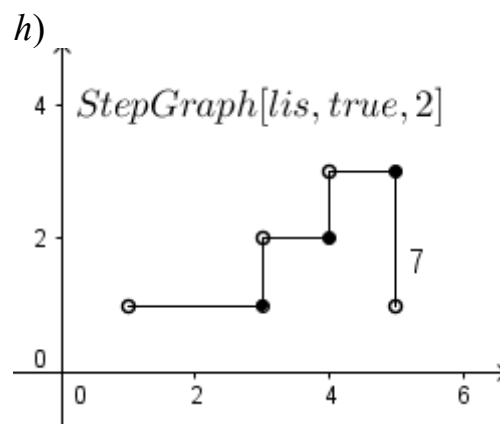
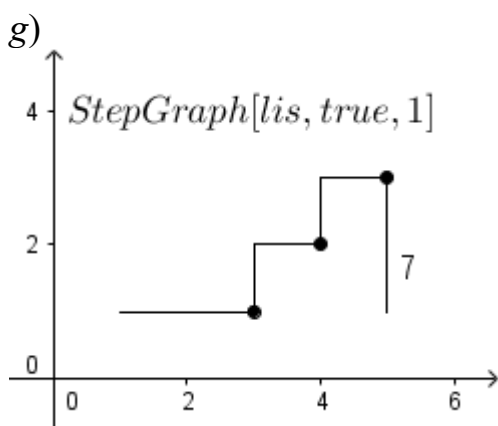
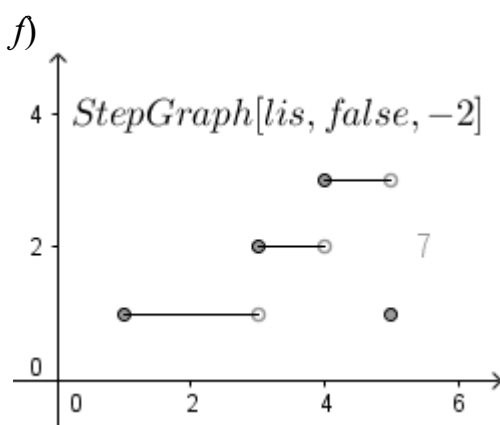
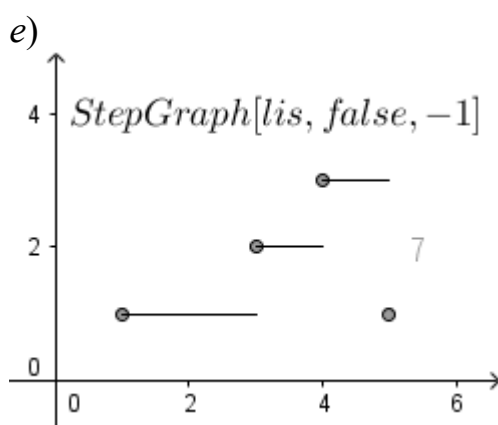
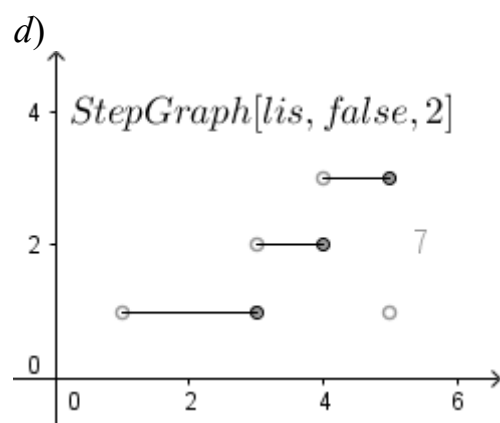
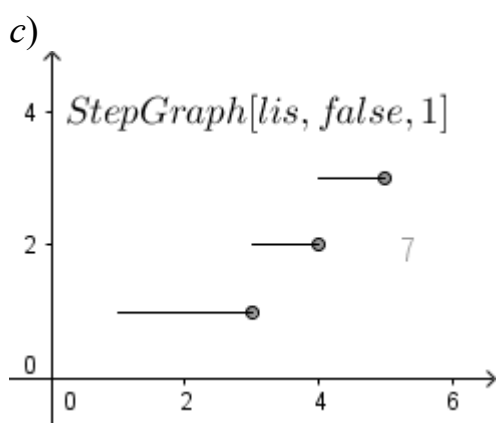


Рис. 38 (a-j). Пошаговые графики, выводимые командами *StepGraph*

6.6.6. Палочные графики

Палочные графики выводятся командой *StickGraph*, которая имеет следующие разновидности синтаксиса:

StickGraph[*lis*], (43)

StickGraph[*lis*, *boolh*], (44)

StickGraph[*lis1*, *lis2*], (45)

StickGraph[*lis1*, *lis2*, *boolh*]. (46)

Опишем аргументы и выполнение каждой из команд (43)-(46) в отдельности.

I. *StickGraph* в форме (43). Здесь $lis = \{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$ – список точек. Выполняется команда так. Выводится каждая точка (x_s, y_s) ($s=1, 2, \dots, k$), а также вертикальный отрезок между точками (x_s, y_s) и $(x_s, 0)$ ($s=1, 2, \dots, k$). Возвращается по (43) сумма ординат точек.

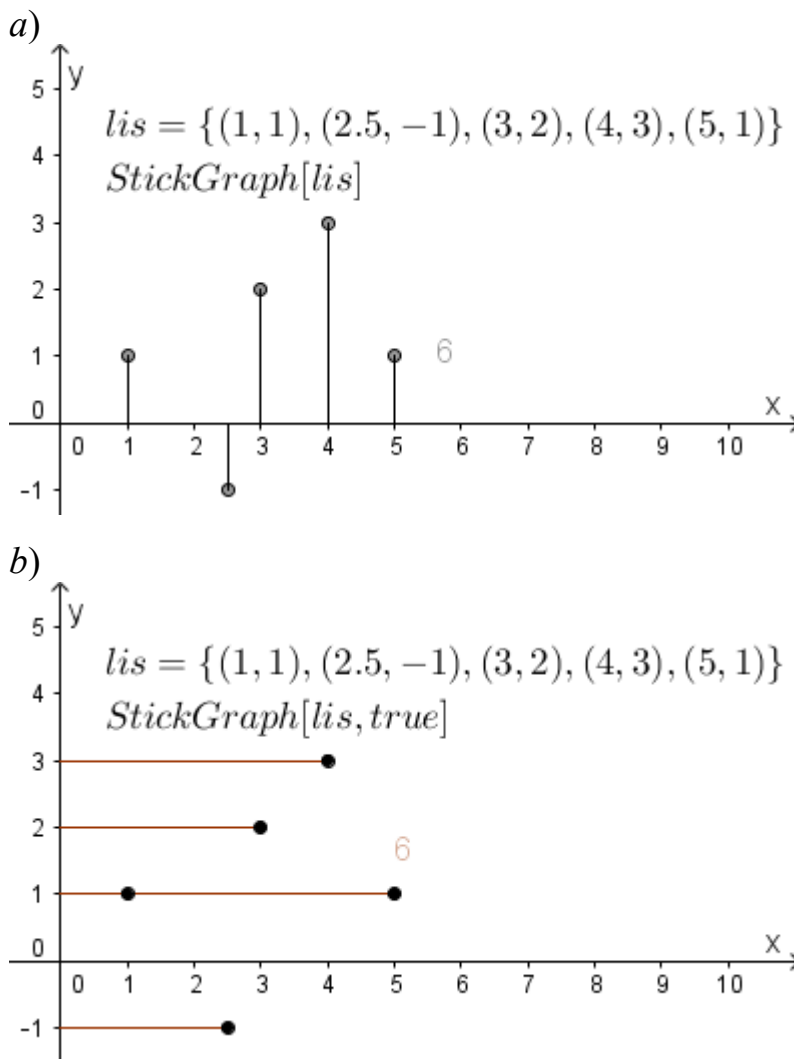


Рис. 39 (a, b). Палочные графики, выводимые командами *StickGraph*

II. *StickGraph* в форме (44). Здесь *lis* – список точек, как и в I, *boolh* – булева переменная. При *boolh=false* вывод тот же самый, что и в случае I. При *boolh=true* выводятся точки (x_s, y_s) ($s=1, 2, \dots, k$), а также горизонтальные отрезки между точками (x_s, y_s) и $(0, y_s)$ ($s=1, 2, \dots, k$). Возвращается по (44) сумма абсцисс точек.

Графики, представленные на рис. 39a и 39b, построены при одном и том же списке *lis*.

III-IV. *StickGraph* в форме (45)-(46). В этих командах вместо единого списка точек *lis* задаются два списка одинаковой длины: *lis1* – список абсцисс, и *lis2* – список ординат. Точки строятся по парам элементов, расположенных на одинаковых позициях в этих списках. В остальном выполнение рассматриваемых команд аналогично выполнению соответствующих команд (43)-(44).

6.6.7. Графики остатков

Графики остатков выводятся командой *ResidualPlot*, которая имеет следующий синтаксис:

$$ResidualPlot[lis, g]. \tag{47}$$

В (47) $lis=\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$ – список точек, $g(x)$ – некоторое выражение от x . Выполняется (47) так. По каждой точке (x_s, y_s) списка *lis* формируется и выводится точка $(x_s, y_s - g(x_s))$ ($s=1, 2, \dots, k$). На рис. 40 показан пример графика остатков.

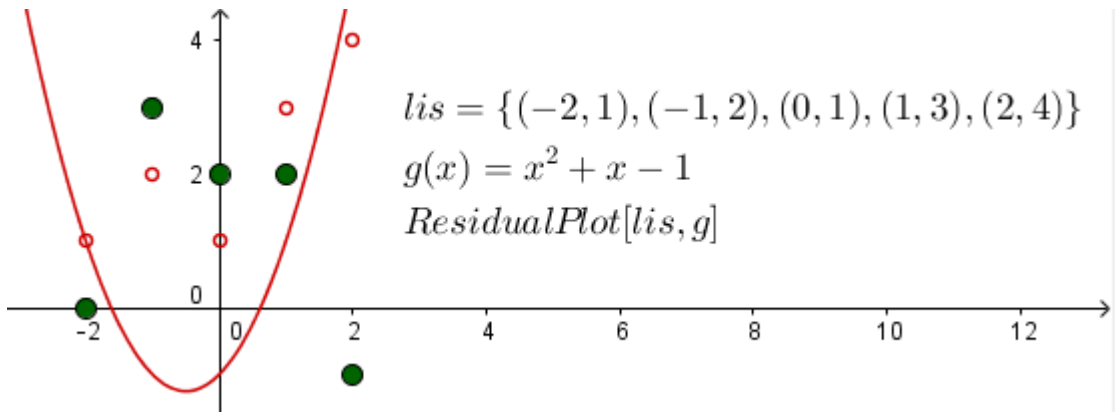


Рис. 40. График остатков (большие точки)

7. Работа со списками

7.1. Простые и вложенные списки

Список – это последовательность объектов, заключенных в фигурные скобки. Например, $\{1, 2, 5, 8\}$ – список действительных чисел, $\{(1,0,0), (0,1,0), (0,0,1)\}$ – список трехмерных точек, $\{\text{“капуста”}, \text{“морковь”}, \text{“свекла”}, \text{“помидоры”}\}$ – строковый список, $\{\}$ – пустой список, то есть список без элементов, $\{poly, (3,5), \sin(x), 8+i, MidPoint[A, B]\}$ – смешанный список из элементов разного типа (*poly* – имя, например, полинома; $(3, 5)$ – точка; $\sin(x)$ – встроенная функция; $8+i$ – комплексное число; $MidPoint[A, B]$ – команда, возвращающая среднюю точку между ранее определенными точками A и B). Объекты списка называют его элементами, а количество элементов списка – его длиной. Элементы в списке пронумерованы от 1 и до n , где n – длина списка. Номер элемента в списке называют также его позицией, или индексом.

Вложенные, гнездовые или многомерные списки – это списки, у которых элементы сами могут быть списками, причем элементы-списки также могут содержать списки и т. д. Простой список – это список, не являющийся вложенным.

Вложенный список удобно интерпретировать деревом, корнем которого является сам список, а от него идут дуги ко всем элементам. Далее от вложенных списков снова идут дуги к их элементам и т. д. Листьями подобного дерева являются объекты, не являющиеся списками, то есть конечные элементы, не имеющие последующих ссылок. Исходя из этого, для гнездовых списков можно использовать многие термины, применяемые для деревьев, такие как листья, уровни вложенности элементов, высота и т. п. Высотой вложенного списка назовем высоту соответствующего ему дерева, то есть максимальное из количеств дуг на путях от корня до листьев. Фактически высота списка есть максимальная глубина его вложенности. Она равна количеству открывающих (или закрывающих) фигурных скобок в записи списка.

$A.Length[lis]$

Длина списка

А) Команда A возвращает длину списка lis , то есть количество элементов на первом уровне вложенности lis .

Примеры 1.

$Length[1,2,3,7] \rightarrow 4;$

$Length[\{9, (3,5), \sin(x), 8+i, MidPoint[(0,0), (1,6)]\}] \rightarrow 5;$

$Length[1, \{2,3\}, \{7,0,1\}] \rightarrow 3.$

Позицией объекта x во вложенном списке lis является последовательность натуральных чисел (i_1, i_2, \dots, i_k) , где i_1 – позиция в списке lis объекта x на его первом уровне вложенности, i_2 – позиция объекта x на втором уровне вложенности, ..., i_k – позиция объекта x на k -ом уровне вложенности. Например, для

вложенного списка $\{2, 1, \{3, \{77, 99, (100,101)\}\}, 7\}$ позиции некоторых его элементов и уровни вложенности представлены в следующей таблице:

Элемент	Позиция	Уровень вложенности
1	2	1
$\{3, \{77, 99, (100,101)\}\}$	3	1
7	4	1
3	(3, 1)	2
$\{77, 99, (100,101)\}$	(3, 2)	2
77	(3, 2, 1)	3
(100, 101)	(3, 2, 3)	3

Отметим, что:

- в списке не может быть присваивания значений переменным или определения функций;
- если речь идет об элементах списка без указания уровня их вложенности, то имеются в виду элементы первого (верхнего) уровня;
- если во вложенном списке *lis* длины *n* два уровня вложенности, и на первом из них находятся простые списки одинаковой длины *m* с действительными числовыми элементами, то при вводе *lis* создается числовая матрица размером *n*×*m* (*n* строк, *m* столбцов). Например,

$$lis = \{\{1,2,3\}, \{9,4,5\}\} \rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 9 & 4 & 5 \end{pmatrix}.$$

При этом по команде *Dimension[*lis*]* возвращается двухэлементный список $\{n, m\}$, где *n* – количество строк, а *m* – количество столбцов матрицы.

- вложенные списки являются обобщениями гнездовых массивов [6,10].

Матрицы являются частным случаем списков. Рассматриваемые ниже средства для работы со списками могут использоваться и при работе с матрицами. Но для матриц имеется много дополнительных средств обработки.

7.2. Создание списков

A. <i>Sequence</i> [<i>expr</i> (<i>k</i>), <i>k</i> , <i>b</i> , <i>e</i>]	C. <i>Sequence</i> [<i>e</i>]	E. <i>b</i> .. <i>e</i>
B. <i>Sequence</i> [<i>expr</i> (<i>k</i>), <i>k</i> , <i>b</i> , <i>e</i> , <i>h</i>]	D. <i>Sequence</i> [<i>b</i> , <i>e</i>]	

Создание списка объектов (1)

Здесь: *expr*(*k*) – выражение от одной переменной, *k* – переменная, *b* и *e* – начальное и конечное значения *k*, *h* – шаг изменения переменной *k*.

А) По команде *A* при *b*≤*e* из выражения *expr*(*k*) создается список объектов вида $\{expr(b), expr(b+1), expr(b+2), \dots, expr(b+m)\}$ (где *m* – натуральное, *b*+*m*≤*e*, *b*+*m*+1>*e*). Формируемые списки на панели “Объекты” по умолчанию

получают имена “*список1*”, “*список2*”, ... (*list1*, *list2*, ...). В дальнейшем эти имена могут быть изменены. Впрочем, можно сразу именовать создаваемые списки по своему усмотрению, вводя вместо *A* выражение вида “*name=Sequence[expr(k), k, b, e]*”. Если элементы сформированного списка являются объектами, отличными от действительных чисел, то они визуализируются на панели “*Полотно*”. При $b > e$ по *A* возвращается пустой список {}.

B) Команда *B* выполняется аналогично *A*, но здесь переменная *k* может принимать такие значения:

- a*) при $b \leq e, h > 0 \rightarrow b, b+h, b+2h, \dots, b+m \cdot h$
(*m* – натуральное, $b+m \cdot h \leq e, b+(m+1) \cdot h > e$);
- b*) при $b > e, h < 0 \rightarrow b, b+h, b+2h, \dots, b+m \cdot h$
(*m* – натуральное, $b+m \cdot h \geq e, b+(m+1) \cdot h < e$).
- c*) в остальных случаях возвращается пустой список.

C) По команде *C* при $e < 0.5$ выводится пустой список, а при $e \geq 0.5$ – список целых чисел вида: {1, 2, ..., *round*(*e*)}, где *round* – функция округления.

D) По команде *D* выводится:

- a*) при $\text{round}(b) \leq \text{round}(e)$ – список целых последовательных чисел вида: {*round*(*b*), *round*(*b*)+1, ..., *round*(*e*)};
- b*) при $\text{round}(b) > \text{round}(e)$ – список целых последовательных чисел вида: {*round*(*b*), *round*(*b*)-1, ..., *round*(*e*)}.

E) Выражение *E* выполняется аналогично команде *D*, то есть является ее укороченной формой записи.

Примеры 2. Создание списка объектов

- a*) *Sequence*[$k^2, k, 1, 5$] $\rightarrow \{1, 4, 9, 16, 25\}$,
Sequence[$k^2, k, 5, 1$] $\rightarrow \{\}$ – пустой список,
Sequence[$k+2, k, 1.5, 5$] $\rightarrow \{3.5, 4.5, 5.5, 6.5\}$,
Sequence[(*k*, *k*+1), *k*, 1, 5] $\rightarrow \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}\}$ –

создается не только список, но и его элементы-точки визуализируются на панели “*Полотно*”;

- b*) *Sequence*[$k^2, k, 1, 3, 0.5$] $\rightarrow \{1, 2.25, 4, 6.25, 9\}$,
Sequence[$k^2, k, 3, 1, -0.5$] $\rightarrow \{9, 6.25, 4, 2.25, 1\}$,
Sequence[$k^2, k, 1, 3, -0.5$] $\rightarrow \{\}$ – пустой список,
Sequence[$k+2, k, 1.5, 3, 0.5$] $\rightarrow \{3.5, 4.5, 5.5, 6.5\}$;
- c*) *Sequence*[3] $\rightarrow \{1, 2, 3\}$; *Sequence*[0.5] $\rightarrow \{1\}$; *Sequence*[0.49] $\rightarrow \{\}$;
Sequence[2, 5] $\rightarrow \{2, 3, 4, 5\}$; *Sequence*[5, 2] $\rightarrow \{5, 4, 3, 2\}$;
Sequence[0, 0.49] $\rightarrow \{0\}$; *Sequence*[0, 0.5] $\rightarrow \{0, 1\}$;

- A. *Zip*[*expr*(*x*₁, *x*₂, ..., *x*_{*k*}), *x*₁, {*a*₁, *a*₂, ..., *a*_{*k*}}, ..., *x*_{*k*}, {*c*₁, *c*₂, ..., *c*_{*k*}}]
- B. *Zip*[*f*(*a*₁, *a*₂, ..., *a*_{*k*}), *f*, {*f*₁(*x*₁, *x*₂, ..., *x*_{*k*}), ..., *f*_{*m*}(*x*₁, *x*₂, ..., *x*_{*k*})}

Создание списка объектов (2)

А) Здесь: $expr$ – выражение от k переменных x_1, x_2, \dots, x_k ; $\{a_1, a_2, \dots, a_k\}$, $\{b_1, b_2, \dots, b_k\}$, ..., $\{c_1, c_2, \dots, c_k\}$ – списки некоторых объектов. В каждом списке должны быть элементы одного типа. По крайней мере одна переменная и один список в Zip присутствовать должны. По команде A из выражения $expr(x_1, x_2, \dots, x_k)$ и списков $\{a_1, a_2, \dots, a_k\}$, $\{b_1, b_2, \dots, b_k\}$, ..., $\{c_1, c_2, \dots, c_k\}$ создается и возвращается новый “упакованный” список объектов вида:

$$\{expr(a_1, b_1, \dots, c_1), expr(a_2, b_2, \dots, c_2), \dots, expr(a_k, b_k, \dots, c_k)\}. \tag{1}$$

Если в списках-аргументах Zip неодинаковое количество элементов, то построение результирующего списка (1) прекращается, как только оказываются исчерпанными элементы одного из списков аргументов.

В) Здесь: $f(a_1, a_2, \dots, a_k)$ – значение некоторой функции f от k переменных в конкретной точке (a_1, a_2, \dots, a_k) ; f – текущее имя функции; $\{f_1(x_1, x_2, \dots, x_k), f_2(x_1, x_2, \dots, x_k), \dots, f_m(x_1, x_2, \dots, x_k)\}$ – список любых функций от k переменных. По команде B возвращается список вида:

$$\{f_1(a_1, a_2, \dots, a_k), f_2(a_1, a_2, \dots, a_k), \dots, f_m(a_1, a_2, \dots, a_k)\}. \tag{2}$$

Примеры 3. Создание списка объектов

- а) $Zip[x1+y1, x1, \{2,3\}, y1, \{8,11\}] \rightarrow \{10, 14\}$;
- б) $Zip[x1+x2+x3, x1, \{2,3\}, x2, \{8,1\}, x3, \{x, x^2\}] \rightarrow \{2+8+x, 3+1+x^2\}$;
- с) Построение треугольника ABC по точкам A, B и C . Пусть заданы 3 точки A, B и C . Тогда по команде

$$Zip[Segment[X,Y], X, \{A,B,C\}, Y, \{B,C,A\}]$$

строятся три сегмента AB, BC и CA , то есть формируется треугольник ABC (см. рис. 1а) (командой $Segment[X, Y]$ строится отрезок между точками X и Y).

- д) Построение медиан треугольника ABC по точкам A, B и C . Пусть заданы три точки A, B и C . Тогда по команде

$$Zip[Segment[MidPoint[X,Y], Z], X, \{A,B,C\}, Y, \{B,C,A\}, Z, \{C,A,B\}]$$

в треугольнике ABC строятся три медианы (см. рис. 1б). Командой $MidPoint[X,Y]$ строится точка, являющаяся серединой отрезка XY .

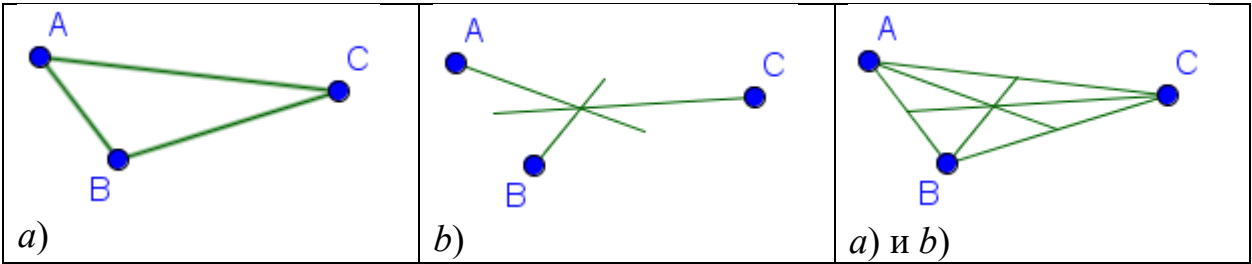


Рис. 1. Результаты выполнения команд Zip из примеров с и d

- e) $Zip[f(2), f, \{x, x^2, x^3\}] \rightarrow \{2, 4, 8\};$
 f) $Zip[g(2), g, Sequence[x^k, k, 1, 7]] \rightarrow \{2, 4, 8, 16, 32, 64, 128\};$
 g) $Zip[h(2, 3), h, Sequence[x^{k+y}, k, 1, 7]] \rightarrow \{5, 7, 11, 19, 35, 67, 131\};$

A. $RootList[\{a_1, a_2, \dots, a_k\}]$
B. $Point[\{a_1, a_2\}]$
C. $PointList[\{a_1, b_1\}, \{a_2, b_2\}, \dots, \{a_k, b_k\}]$

Создание списка объектов (3)

A) Команда A по списку числовых значений $\{a_1, a_2, \dots, a_k\}$ формирует список точек $\{(a_1, 0),$

$(a_2, 0), \dots, (a_k, 0)\}$, каждая из которых выводится на панели “Полотно”.

B) Команда B по списку чисел $\{a_1, a_2\}$ возвращает точку (a_1, a_2) .

C) Команда C по списку $\{\{a_1, b_1\}, \{a_2, b_2\}, \dots, \{a_k, b_k\}\}$ двухэлементных числовых списков возвращает список точек $\{(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)\}$. На верхнем уровне вложенности списка-аргумента могут находиться и элементы, отличные от двухэлементных числовых списков. Но при выполнении C они игнорируются.

Примеры 4. Создание списка точек

a) Командой $RootList[\{1, 2, 4, 3\}]$ формируется список точек $\{(1, 0), (2, 0), (4, 0), (3, 0)\}$, и все созданные точки выводятся на панели “Полотно”.

b) Командой $PointList[\{\{1, 2\}, \{3, 4\}, \{5, 3\}, \{1, 1\}\}]$ формируется список точек $\{(1, 2), (3, 4), (5, 3), (1, 1)\}$, и все созданные точки выводятся на панели “Полотно”.

A. $IterationList[expr(x), a, n]$	C. $IterationList[f, a, n]$
B. $Iteration[expr(x), a, n]$	D. $Iteration[f, a, n]$

Создание списков итерациями (1)

A) Здесь: $expr(\dots)$ – выражение от одной переменной x или y , a – начальное (стартовое) значение переменной, n – количество итераций. По команде A возвращается список из $n+1$ элемента вида: $\{a, expr(a), expr(expr(a)), \dots\}$.

B) Команда B возвращает последний элемент списка, формируемого по A.

C) Здесь: f – имя ранее определенной функции от одной переменной, a – стартовое значение переменной, n – количество итераций. По команде C возвращается список из $n+1$ элемента вида: $\{a, f(a), f(f(a)), \dots\}$. По команде D возвращается последний элемент списка, формируемого по C.

D) Команда D возвращает последний элемент списка, формируемого по C.

A. $IterationList[expr(x_1, x_2, \dots, x_k), x_1, x_2, \dots, x_k, \{a_1, a_2, \dots, a_k\}, n]$
B. $Iteration[expr(x_1, x_2, \dots, x_k), x_1, x_2, \dots, x_k, \{a_1, a_2, \dots, a_k\}, n]$
C. $IterationList[f, x_1, x_2, \dots, x_k, \{a_1, a_2, \dots, a_k\}, n]$
D. $Iteration[f, x_1, x_2, \dots, x_k, \{a_1, a_2, \dots, a_k\}, n]$

Создание списков итерациями (2)

A) Здесь: $\text{expr}(x_1, x_2, \dots, x_k)$ – выражение от k переменных; x_1, x_2, \dots, x_k – имена переменных; $\{a_1, a_2, \dots, a_k\}$ – список стартовых значений для соответствующих переменных; n – количество итераций. По команде A возвращается список из $n+k-1$ элементов, который формируется следующим образом:

$$\{a_1, a_2, \dots, a_k, \text{expr}(a_1, a_2, \dots, a_k) (=b), \\ \text{expr}(a_2, \dots, a_k, b) (=c), \text{expr}(a_3, \dots, a_k, b, c) (=d), \dots \}.$$

B) Команда B возвращает последний из элементов списка, формируемого по A .

C) Здесь: f – имя ранее определенной функции от k переменных, x_1, x_2, \dots, x_k – имена переменных; $\{a_1, a_2, \dots, a_k\}$ – список стартовых значений для соответствующих переменных; n – количество итераций. По команде C возвращается список из $n+k-1$ элементов, который формируется следующим образом:

$$\{a_1, a_2, \dots, a_k, f(a_1, a_2, \dots, a_k) (=b), \\ f(a_2, \dots, a_k, b) (=c), f(a_3, \dots, a_k, b, c) (=d), \dots \}.$$

D) Команда D возвращает последний из элементов списка, формируемого по C .

Примеры 5. Создание списков итерациями

1) Арифметическая прогрессия

a) $\text{IterationList}[x+5, 0, 6] \rightarrow \{0, 5, 10, 15, 20, 25, 30\};$

$\text{Iteration}[x+5, 0, 6] \rightarrow 30.$

b) Определяем $g(b)=b+1$

$\text{IterationList}[g, 0, 6] \rightarrow \{0, 1, 2, 3, 4, 5, 6\};$

$\text{Iteration}[g, 0, 6] \rightarrow 30.$

2) Геометрическая прогрессия

Определяем $g(x)=2*x$

$\text{IterationList}[g, 1, 6] \rightarrow \{1, 2, 4, 8, 16, 32, 64\};$

$\text{Iteration}[g, 0, 6] \rightarrow 64.$

3) Последовательность Фибоначчи

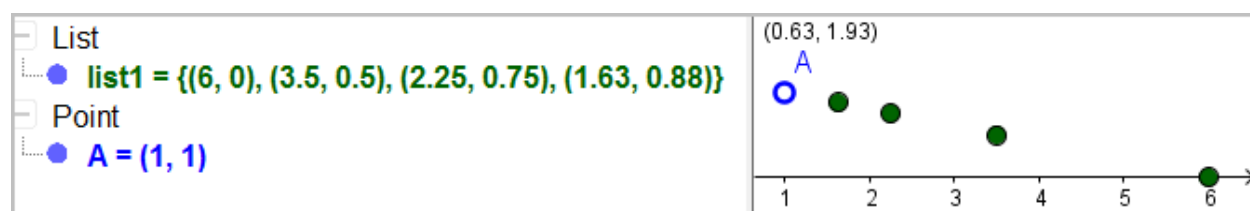
$\text{IterationList}[a+b, a, b, \{1, 1\}, 7] \rightarrow \{1, 1, 2, 3, 5, 8, 13, 21\};$

$\text{Iteration}[a+b, a, b, \{1, 1\}, 5] \rightarrow 21.$

4) Вводим $A=(1, 1)$, а затем:

$\text{Iteration}[\text{Midpoint}[A, B], B, \{(6,0)\}, 3] \rightarrow (1.63, 0.88);$

$\text{IterationList}[\text{Midpoint}[A, B], B, \{(6,0)\}, 3] \rightarrow$ см. результат ниже.



Как видим, создан список из четырех точек и эти точки выведены, причем точка A к выводу не относится. Выполнялась команда так. Сначала в список включена точка $(6, 0)$. Затем в конец списка последовательно добавлялись точки: $(3.5, 0.5)$ – средняя точка отрезка между A и $(6, 0)$, $(2.25, 0.75)$ – средняя точка отрезка между A и $(3.5, 0.5)$, $(1.63, 0.88)$ – средняя точка отрезка между A и $(2.25, 0.75)$.

<div style="border: 1px solid black; padding: 5px;"> <p>A. <i>Coefficients</i>[conic] B. <i>Coefficients</i>[poly]</p> </div>	<p><i>Создание списков коэффициентов коники и полинома</i></p> <p>А) Аргумент <i>conic</i> в A – это имя коники (конического сечения) или ее уравнение с любым порядком следования членов. Команда A возвращает список коэффициентов коники. Если коника на панели “Объекты” представлена в виде: “$h:ax^2+bxy+cy^2+dx+ey=f$”, где a, b, c, d, e и f – действительные числа, то <i>conic</i> – это или имя h, или уравнение $ax^2+bxy+cy^2+dx+ey=f$, а возвращается список вида $\{a, c, -f, b, d, e\}$.</p> <p>В) Аргумент <i>poly</i> в B – это имя полинома от одной переменной x или задающее его выражение с любым порядком следования членов. Команда B возвращает список коэффициентов полинома в порядке убывания степеней переменной. Если полином на панели “Объекты” представлен в виде функции “$r(x)=a_nx^n+a_{n-1}x^{n-1}+...+a_1x+a_0$”, где a_k ($k=0,1, ..., n$) – действительные числа, то <i>poly</i> – это имя r, или выражение $a_nx^n+a_{n-1}x^{n-1}+...+a_1x+a_0$, а возвращается список вида: $\{a_n, a_{n-1}, ..., a_1, a_0\}$.</p>
--	---

Примеры 6. Создание списка коэффициентов коники и полинома

- 1) *Coefficients*[$x^2+x*y+y^2+x=5$] → {1, 1, -5, 1, 1, 0};
- 2) *Coefficients*[x^4+x^2+7] → {1, 0, 1, 0, 7}.

<div style="border: 1px solid black; padding: 5px;"> <p>A. <math>x[<i>lis</i>]</math></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>C. <math>y[<i>lis</i>]</math></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>D. <math>z[<i>lis</i>]</math></p> </div>	<p><i>Создание списков из точек</i></p> <p>Здесь <i>lis</i> – это список двумерных точек $\{(a_1,b_1), (a_2,b_2), ..., (a_n,b_n)\}$ или список трехмерных точек $\{(a_1,b_1,c_1), (a_2,b_2,c_2), ..., (a_n,b_n,c_n)\}$.</p> <p>А) Команда A возвращает список абсцисс двумерных или трехмерных точек списка <i>lis</i>.</p> <p>В) Команда B возвращает список ординат двумерных или трехмерных точек списка <i>lis</i>.</p> <p>С) Команда C возвращает список аппликат трехмерных точек списка <i>lis</i>. При наличии в списке <i>lis</i> и трехмерных, и двумерных точек последние при выполнении C считаются трехмерными с аппlikатой, равной 0.</p>
--	--	--	---

- Примеры 7.*
- 1. $lis=\{(1,4), (2,7), (3,5), (11,0)\}$;
 $x[*lis*] \rightarrow \{1, 2, 3, 11\}$; $y[*lis*] \rightarrow \{4, 7, 5, 0\}$.
 - 2. $li=\{(1,4,77), (2,7,88), (3,5,1), (11,0,1)\}$;
 $z[*li*] \rightarrow \{77, 88, 1, 1\}$.

3. $li1 = \{(1,4,77), (2,7), (3,5,99), (11,0)\};$
 $z[li1] \rightarrow \{77, 0, 99, 0\}.$
4. $x[PolntList[\{\{1,4\}, \{2,7\}, \{3,5\}, \{11,0\}\}]] \rightarrow \{1, 2, 3, 11\}.$

7.3. Операции и функции с числовыми списками

A. $lis1 \otimes lis2$ ($\otimes \in \{+, -, *, \backslash\}$) B. lis^p	C. $f[li]$	D. $GCD[li]$ E. $LCM[li]$
--	------------	------------------------------

Операции и функции с числовыми списками

Здесь lis , $lis1$ и $lis2$ – простые списки одинаковой длины с элементами, являющимися действительными или комплексными числами.

A) По выражению A возвращается список, каждый элемент которого равен результату выполнения бинарной операции \otimes ($\otimes \in \{+, -, *, \backslash\}$) над парой соответствующих друг другу элементов списков $lis1$ и $lis2$. Если длины списков не равны, то по A возвращается пустой список. Выражение $lis1 * lis2$ и команда $Dot[li1, li2]$ выполняются одинаково.

B) По выражению B возвращается список, каждый элемент которого равен результату выполнения унарной операции “^” возведения в степень p . Если над конкретным элементом операция невыполнима, то в результате на этой позиции появляется символ “?”.

C) Для обработки списков можно использовать практически любые встроенные математические функции f одной переменной. Соответствующие действия будут осуществляться над каждым элементом списка.

D) По команде D находится наибольший общий делитель чисел, являющихся элементами списка lis . Если наибольший общий делитель находится для двух чисел a и b , то допустим синтаксис $GCD[a, b]$.

E) По команде E находится наименьшее общее кратное чисел, являющихся элементами списка lis . Если наименьшее общее кратное находится для двух чисел a и b , то допустим синтаксис $LCM[a, b]$.

Замечание. Перечисленные в A-C действия могут выполняться также и для гнездовых списков одинаковой структуры с числовыми элементами на последнем уровне вложенности.

Примеры 8.

- Пусть $lis1 = \{1, 2, 3\}$, $lis2 = \{4, 5, 6\}$. Тогда:
- a) $lis1 + lis2 \rightarrow \{5, 7, 9\}$, $lis1 - lis2 \rightarrow \{-3, -3, -3\};$
 - b) $lis1 * lis2 \rightarrow \{4, 10, 18\}$, $Dot[li1, li2] \rightarrow \{4, 10, 18\},$
 $Dot[li1, li2] == lis1 * lis2 \rightarrow true;$
 - c) $lis1 / lis2 \rightarrow \{0.25, 0.4, 0.5\};$
 - d) $lis1^2 \rightarrow \{1, 4, 9\}$, $\{1, -2, 3\}^{(-2/3)} \rightarrow \{1, ?, 0.48\}.$

Примеры 9.

Пусть $li1=\{1, 2, \{3,4\}, 5, (1,4)\}$, $li2=\{0, 7, \{1,2\}, 3, (2,5)\}$. Тогда:

a) $li1+li2 \rightarrow \{1, 9, \{4,6\}, 8, \{3,9\}\}$, $li1-li2 \rightarrow \{1, -5, \{2,2\}, 2, \{-1,-1\}\}$;

b) $li1*li2 \rightarrow \{0, 14, \{3,8\}, 15, 22\}$. Здесь $(1,4)*(2,5)=22$ – операция выполняется как скалярное произведение векторов;

c) $li1/li2 \rightarrow \{\infty, 0.29, \{3,2\}, 1.67, (0.76,0.1)\}$. Здесь $(1,4)/(2,5)$ – операция выполняется как отношение комплексных чисел $(1+4i)/(2+5i) \approx 0.76+0.1i$.

d) $li1^2 \rightarrow \{1, 4, \{9, 16\}, 25, 17\}$;

$\{1, 2, \{-3,4\}, 5, (1+4*i)\}^{(-2/3)} \rightarrow \{1, 0.63, \{?, 0.4\}, 0.34, 0.25-0.3i\}$.

Примеры 10.

Пусть $li=\{1.45, 2.5, 3.77, 4.1\}$. Тогда:

a) $\sin[li] \rightarrow \{1, 0.63, \{0.48, 0.4\}, 0.34, 0.25-0.3i\}$;

b) $\text{floor}[li] \rightarrow \{1, 2, 3, 4\}$, $\text{ceil}[li] \rightarrow \{2, 3, 4, 5\}$;

c) $\text{Max}[li] \rightarrow 4.1$, $\text{Min}[li] \rightarrow 1.45$;

d) $\text{GCD}[\{30, 42, 66\}] \rightarrow 6$, $\text{LCM}[\{30, 42, 66\}] \rightarrow 2310$.

7.4. Выборки элементов из списков

В этом разделе рассматриваются различного рода выборки из простых или гнездовых списков, элементами которых могут быть любые объекты *GeoGebra*.

A. <i>Element</i> [<i>lis</i> , <i>i</i>] B. <i>lis</i> (<i>i</i>)	C. <i>Element</i> [<i>lis</i> , <i>i</i> , <i>j</i>] D. <i>lis</i> (<i>i</i> , <i>j</i>)	E. <i>Element</i> [<i>lis</i> , <i>i1</i> , <i>i2</i> , ..., <i>ik</i>] F. <i>lis</i> (<i>i1</i> , <i>i2</i> , ..., <i>ik</i>)
---	---	---

Выборка элементов из списков.

В A-F элементы $i, j, i1, i2, \dots, ik$ – целые числа. Если это не так, то перед использованием они округляются до целых чисел функцией *round*.

A-B) Пусть m – длина списка *lis* и $1 \leq i \leq m$. Тогда по командам A и B возвращается элемент, находящийся в i -ой позиции первого уровня вложенности списка *lis*. При иных i возвращается слово “*undefined*” (не определено).

C-D) По командам C и D возвращается элемент x , находящийся в позиции (i, j) списка *lis*. Это означает, что на первом уровне вложенности x расположен в позиции i , а на втором уровне вложенности – в позиции j . При отсутствии в *lis* позиции (i, j) возвращается слово “*undefined*” (не определено).

E-F) По командам E и F возвращается элемент x , находящийся в позиции $(i1, i2, \dots, ik)$ списка *lis*. Это означает, что первом уровне вложенности x расположен в позиции $i1$, а на втором уровне вложенности – в позиции $i2, \dots$, на k -ом уровне вложенности – в позиции ik . При отсутствии в *lis* позиции $(i1, i2, \dots, ik)$ возвращается слово “*undefined*”.

Примеры 11. Выборка элементов, находящихся на различной глубине вложенности списков.

- a) $li = \{1, 2, 31, 41, 51\}$, $Element[li, 3] \rightarrow 31$, $li(3) \rightarrow 31$;
 $Element[li, 6] \rightarrow undefined$, $li(6) \rightarrow undefined$;
- b) $li = \{\{1, 2, 3\}, \{(2, 1), (3, 2)\}, \{1, 7, 5, 6\}\}$,
 $Element[li, 3, 4] \rightarrow 6$, $li(3, 4) \rightarrow 6$, $Element[li, 2, 2] \rightarrow (3, 2)$,
 $Element[li, 2, 3] \rightarrow undefined$;
- c) $li = \{\{1, 2, 3\}, \{2, \{44, \{55, 77\}\}, 1\}, (3, 2), \{1, 7, 5, 6\}\}$,
 $Element[li, 2] \rightarrow \{2, \{44, \{55, 77\}\}, 1\}$, $li(2, 2) \rightarrow \{44, \{55, 77\}\}$,
 $li(2, 2, 2) \rightarrow \{55, 77\}$, $li(2, 2, 2, 2) \rightarrow 77$, $li(2, 2, 2, 3) \rightarrow undefined$.

A. $First[li]$ B. $First[li, k]$	C. $First[text]$ D. $First[text, k]$	Выборка начальных элементов из списков и строк Будем считать k целым числом. Если это не так, то перед использованием k округляется до целых чисел функцией $round$.
-------------------------------------	---	--

- A) Пусть x – первый элемент первого уровня вложенности списка lis . Рассмотрим случаи. Если x – не список или вложенный список, то по A возвращается список $\{x\}$. Если x – простой список, то по A возвращается x , преобразованный в однострочную матрицу.
- B) При $k=1$ команда B выполняется как A . При $2 \leq k \leq n$, где n – длина списка lis , по B возвращается список из k начальных элементов lis . Для иных значений k возвращается слово “ $undefined$ ”.
- C) По команде C возвращается первый символ строки $text$.
- D) Пусть n – длина строки $text$. При $1 \leq k \leq n$ по D возвращаются первые k символов строки $text$. Для иных значений k возвращается слово “ $undefined$ ”.

Примеры 12. Выборка начальных элементов из списков и строк.

- a) $First[\{1, 2, 3, 4\}] \rightarrow \{1\}$, $First[\{(1, 2), 2, 3, 4\}] \rightarrow \{(1, 2)\}$;
- b) $First[\{\{1, 2, 9\}, 2, 3, 4\}] \rightarrow (1 \ 2 \ 9)$ (матрица!);
- c) $First[\{\{1, 2, 9\}, 2, 3, 4\}, 1] \rightarrow (1 \ 2 \ 9)$ (матрица!);
- d) $First[\{\{1, 2, 9\}, 2, 3, 4\}, 2] \rightarrow \{\{1, 2, 9\}, 2\}$;
- e) $First[“анельсин”, 5] \rightarrow “анель”$.

A. $Last[li]$ B. $Last[li, k]$	C. $Last[text]$ D. $Last[text, k]$	<i>Выборка конечных элементов из списков и строк</i> Будем считать k целым числом.
-----------------------------------	---------------------------------------	---

- A) Пусть x – последний элемент первого уровня вложенности списка lis . Рассмотрим случаи. Если x – не список или вложенный список, то по A возвращается $\{x\}$. Если x – простой список, то по A возвращается x , преобразованный в однострочную матрицу.

В) При $k=1$ команда B выполняется как A . При $2 \leq k \leq n$, где n – длина списка lis , по B возвращается список из k последних элементов lis . Для иных значений k возвращается слово “*undefined*”.

С) По команде C возвращается последний символ строки $text$.

Д) Пусть n – длина строки $text$. При $1 \leq k \leq n$ по D возвращаются последние k символов строки $text$. Для иных значений k возвращается слово “*undefined*”.

Примеры 13. Выборка конечных (последних) элементов из списков и строк.

- a) $Last[\{1, 2, 3, 4\}] \rightarrow \{4\}$, $Last[\{1, 2, 3, (4, 5)\}] \rightarrow \{(4, 5)\}$;
- с) $Last[\{1, 2, 3, \{1,2,9\}\}] \rightarrow (1 \ 2 \ 9)$ (матрица!);
- d) $Last[\{1, 2, 3, \{1,2,9\}\}, 1] \rightarrow (1 \ 2 \ 9)$ (матрица!);
- e) $Last[\{1, 2, 3, \{1,2,9\}\}, 2] \rightarrow \{3, \{1,2,9\}\}$;
- e) $Last[“анельсин”, 5] \rightarrow “льсин”$.

A. <i>RandomElement</i> [<i>lis</i>]	C. <i>Sample</i> [<i>lis</i> , <i>n</i>]
B. <i>Shuffle</i> [<i>lis</i>]	D. <i>Sample</i> [<i>lis</i> , <i>n</i> , <i>boolrepeat</i>]

Выборка случайных элементов из списков

А) Здесь lis – список элементов одного типа (числа, точки, списки и т. п.). По команде A возвращается псевдослучайный элемент из первого уровня вложенности lis (при равномерном распределении элементов).

В) Здесь lis – список элементов любого и не обязательно одинакового типа. По команде B возвращается список из всех объектов первого уровня вложенности списка lis , но случайным образом перетасованные.

С) Здесь lis – список элементов любого и не обязательно одинакового типа. По C возвращается список из n случайно выбранных объектов из первого уровня вложенности списка lis . В результирующем списке объекты могут повторяться.

Д) Команда D при переменной $boolrepeat=true$ выполняется аналогично C . Если $boolrepeat=false$, то в результирующий список повторяющиеся объекты не включаются. При $n > Length[*lis*]$ возвращается сообщение *undefined*.

Примеры 14. Выборка случайных элементов списков.

- a) $RandomElement[\{1, 2, 3, 4\}] \rightarrow 3$, $RandomElement[\{1, 2, 3, 4\}] \rightarrow 1$;
- b) $RandomElement[\{(1,2), (3,4), (31,41), (1,5), (3,9)\}] \rightarrow (1, 5)$;
- с) $RandomElement[\{\{1\}, \{2, 3, \{4\}\}, \{\{1, 3\}\}\}] \rightarrow (1 \ 3)$;
- e) $Shuffle[\{1, 2, 3, 4\}] \rightarrow \{4, 2, 1, 3\}$,
 $Shuffle[\{1, 2, 3, 4, \{1, 2, 3, 4\}\}] \rightarrow \{\{1, 2, 3, 4\}, 1, 3, 4, 2\}$;
- f) $Sample[\{1, 2, 3, 4\}, 6] \rightarrow \{4, 1, 4, 1, 1, 2\}$,
 $Sample[\{1, 2, 3, 4\}, 6] \rightarrow \{1, 3, 3, 1, 4, 4\}$;
- g) $Sample[\{1, 2, 3, 4\}, 3, false] \rightarrow \{1, 4, 2\}$,
 $Sample[\{1, 2, 3, 4\}, 6, false] \rightarrow undefined$;

A. <i>IndexOf</i> [<i>obj</i> , <i>lis</i>]	C. <i>IndexOf</i> [<i>te</i> , <i>text</i>]
B. <i>IndexOf</i> [<i>obj</i> , <i>lis</i> , <i>beg</i>]	D. <i>IndexOf</i> [<i>te</i> , <i>text</i> , <i>beg</i>]

Индекс вхождения объекта в список или вхождения строки в строку

Будем считать *beg* целым числом.

A) По команде *A* возвращается индекс первого вхождения объекта *obj* в список *lis* на его первом уровне вложенности. При отсутствии вхождений возвращается слово *undefined*.

B) Команда *B* выполняется как *A*, но поиск ведется, начиная с позиции *beg*.

C) По команде *C* возвращается индекс первого вхождения строки *te* в строку *text*. При отсутствии вхождений возвращается слово *undefined*.

D) Команда *D* выполняется как *C*, но поиск ведется, начиная с позиции *beg*.

Примеры 15. Индекс вхождения.

- a) *IndexOf*[21, {1, 21, 3, 21}] → 2;
- b) *IndexOf*[23, {1, 21, 3, 21}] → *undefined*;
- c) *IndexOf*[23, {1, 21, 3, {21, 23}}] → *undefined*;
- d) *IndexOf*[21, {1, 21, 3, 21}, 3] → 4;
- e) *IndexOf*["бра", "абракадабра"] → 2;
- f) *IndexOf*["бра", "абракадабра", 3] → 9.

7.5. Выборки элементов с помощью логических функций

Здесь аргументы в командах таковы: *lis* – список, *cond* – условие, *var* – переменная. Условие *cond* формируется возможными операторами отношения "<", "≤", ">", "≥", "==" и "!=" ("≠"). Если отношение проверяется для действительных чисел, то условие *cond* может быть любым, иначе в *cond* могут использоваться только операторы "==" (равно) и "!=" (не равно).

A. <i>CountIf</i> [<i>cond</i> , <i>lis</i>]
B. <i>CountIf</i> [<i>cond</i> , <i>var</i> , <i>lis</i>]

Подсчет количества элементов в списке

A) По команде *A* возвращается количество элементов первого уровня вложенности списка *lis*, удовлетворяющих условию *cond*.

B) Это более гибкий вариант команды *A*. Аргументы в команде *B* надо понимать так: условие *cond* в том или ином виде связано с переменной *var*, которая принимает значения последовательных элементов *lis*. Выполнение команды *B* такое же, как и *A*, то есть по *B* возвращается количество элементов первого уровня вложенности *lis*, удовлетворяющих условию *cond*.

Примеры 16. Подсчет количества элементов, удовлетворяющих заданному условию.

- a) Пусть $li = \{1, 7, 2, 4, 0, 6, 7\}$. Тогда:
 $CountIf[x < 4, li] \rightarrow 3$, $CountIf[x \geq 4, li] \rightarrow 4$,
 $CountIf[x \leq 4, li] \rightarrow 4$, $CountIf[x \neq 6, li] \rightarrow 6$, $CountIf[x == 7, li] \rightarrow 2$,
- b) Пусть $li = \{(1,7), 2, 4, 0, (6,7)\}$. Тогда:
 $CountIf[x \neq 6, li] \rightarrow 5$, $CountIf[x == 7, li] \rightarrow 0$, $CountIf[x == (1,7), li] \rightarrow 1$;
- c) $CountIf[x > 5, B1:D5] \rightarrow 7$ (здесь $B1:D5$ – диапазон ячеек электронной таблицы).
- d) $CountIf[x(A) < y(A), A, \{(1,3), (3,1), (5,6), (8,9)\}] \rightarrow 3$ (подсчитывается количество точек списка, у которых абсцисса меньше ординаты);
- e) $CountIf[Slope(U) \leq 1, U, \{x+8, 3*x+1, 0.2*x+11\}] \rightarrow 2$ (подсчитывается количество прямых линий с наклоном, не превосходящим 45°).

<div data-bbox="140 801 506 889" data-label="Text"><p>A. <i>KeepIf</i>[<i>cond</i>, <i>lis</i>] B. <i>KeepIf</i>[<i>cond</i>, <i>var</i>, <i>lis</i>]</p></div>	<div data-bbox="575 784 1369 954" data-label="Text"><p><i>Выборка элементов из простого списка</i> A) По команде <i>A</i> возвращается список элементов первого уровня вложенности списка <i>lis</i>, удовлетворяющих условию <i>cond</i>.</p></div>
---	--

B) Это более гибкий вариант команды *A*. Здесь и в соответствующей команде *CountIf* аргументы имеют одинаковый смысл, то есть условие *cond* в том или ином виде связано с переменной *var*, которая принимает значения последовательных элементов *lis*. А выполнение команды *B* такое же, как и *A*, то есть по *B* возвращается список элементов первого уровня вложенности *lis*, удовлетворяющих условию *cond*.

Примеры 17. Выборка элементов из списка по заданному условию.

- a) Пусть $li = \{1, 7, 2, 4, 0, 6, 7\}$. Тогда:
 $KeepIf[x < 4, li] \rightarrow \{1, 2, 0\}$, $KeepIf[x \geq 4, li] \rightarrow \{7, 4, 6, 7\}$,
 $KeepIf[x \leq 4, li] \rightarrow \{1, 2, 4, 0\}$, $KeepIf[x \neq 6, li] \rightarrow \{1, 7, 2, 4, 0, 7\}$,
 $KeepIf[x == 7, li] \rightarrow \{7, 7\}$;
- b) Пусть $li = \{(1,7), 2, 4, 0, (6,7)\}$. Тогда:
 $KeepIf[x \neq 6, li] \rightarrow \{(1,7), 2, 4, 0, (6,7)\}$, $KeepIf[x == 7, li] \rightarrow \{\}$,
 $KeepIf[x == (1,7), li] \rightarrow \{(1,7)\}$;
- c) $KeepIf[x(A) < y(A), A, \{(1,3), (3,1), (5,6), (8,9)\}] \rightarrow \{(1,3), (5,6), (8,9)\}$;
- d) $KeepIf[Slope(U) \leq 1, U, \{x+8, 3*x+1, 0.2*x+11\}] \rightarrow \{x+8, 0.2*x+11\}$.

7.6. Добавления, удаления, вставки, вырезки

<div data-bbox="140 1892 479 2022" data-label="Text"><p>A. <i>Append</i>[<i>obj</i>, <i>lis</i>] B. <i>Append</i>[<i>lis</i>, <i>obj</i>] C. <i>SetValue</i>[<i>lis</i>, <i>k</i>, <i>obj</i>]</p></div>	<div data-bbox="560 1874 1369 2042" data-label="Text"><p><i>Добавление элементов к спискам</i> A) По команде <i>A</i> возвращается список, полученный добавлением в начало списка <i>lis</i> объекта <i>obj</i>.</p></div>
--	--

В) По команде B возвращается список, полученный добавлением в конец списка lis объекта obj .

С) Пусть n – длина существующего списка lis и $1 \leq k \leq n+1$. По команде C возвращается список, в котором его элемент из позиции k становится равным объекту obj . Если obj замещает некоторый объект ele , то он должен быть того же типа, что и ele .

Примеры 18. Добавление элементов в списки.

- a) $Append[x^3, \{1, 2, 3, 4, 5\}] \rightarrow \{x^3, 1, 2, 3, 4, 5\};$
- b) $Append[\{1, 2, 3, 4, 5\}, x^3] \rightarrow \{1, 2, 3, 4, 5, x^3\};$
- c) $Append[\{x^3\}, \{1, 2, 3, 4, 5\}] \rightarrow \{x^3, \{1, 2, 3, 4, 5\}\};$
- d) Пусть $li = \{1, 2, 3, 4, 5\}$. Тогда $SetValue[li, 2, 9] \rightarrow \{1, 9, 3, 4, 5\};$
 $SetValue[li, 6, 99] \rightarrow \{1, 9, 3, 4, 5, 99\}.$

A. $Remove[li, li]$
 B. $RemoveUndefined[li]$

Удаление элементов из списков
 А) По команде A возвращается список, полученный из первого списка lis удалением элементов, присутствующих во втором списке li . При этом, если элемент a присутствует n раз в lis и m раз в li , то из lis удаляются все элементы a , если $n \leq m$, и первые m элементов a , если $n > m$.

В) По команде B возвращается список, полученный удалением из списка lis всех неопределенных элементов.

Примеры 19. Удаление элементов из списков.

- a) $Remove[\{1, 4, 2, 3, 4, 5, 4\}, \{7, 1, 4, 4, (2,3)\}] \rightarrow \{2, 3, 5, 4\};$
- b) $d=7$
 $RemoveUndefined[\{1, 1/(d-7), 3, (-1)^{(1.1)}, 5\}] \rightarrow \{1, \infty, 3, 5\}.$

A. $Insert[obj, lis, pos]$
 B. $Insert[li, lis, pos]$

Вставка элементов в списки
 А) Пусть $1 \leq pos \leq n+1$, где n – длина списка lis , и объект obj не является списком. Тогда по команде A возвращается список, полученный вставкой в позицию pos списка lis объекта obj . При этом элементы lis , находящиеся в позиции pos и дальше сдвигаются вправо на одну позицию.

В) Пусть $1 \leq pos \leq n+1$, где n – длина списка lis . Тогда по команде B возвращается список, полученный вставкой в список lis последовательных элементов списка li . Вставка проводится в позиции $pos, pos+1, \dots$. При этом собственные элементы lis , находящиеся в позициях pos и дальше, сдвигаются вправо на длину списка li . При иных значениях pos возвращается сообщение “undefined”.

Примеры 20. Вставки элементов в списки.

- a) $Insert[x^3, \{1, 2, 3, 4, 5\}, 3] \rightarrow \{1, 2, x^3, 3, 4, 5\}$
b) $Insert[x^3, \{1, 2, 3, 4, 5\}, 6] \rightarrow \{1, 2, 3, 4, 5, x^3\}$
c) $Insert[x^3, \{1, 2, 3, 4, 5\}, 7] \rightarrow undefined$
d) $A = (3, 11)$
 $Insert[A, \{1, 2, 3, 4, 5\}, 3] \rightarrow \{1, 2, (3, 11), 3, 4, 5\}$
e) $Insert[\{7, 8, "Ab"\}, \{1, 2, 3, 4, 5\}, 3] \rightarrow \{1, 2, 7, 8, "Ab", 3, 4, 5\}$

A. $Take[lis, beg]$	C. $Take[text, beg]$
B. $Take[lis, beg, end]$	D. $Take[text, beg, end]$

Вырезки из списков и текстов

A) Пусть $1 \leq beg \leq n$, где n – длина списка lis . Тогда по команде A возвращается список, состоящий из элементов списка lis , находящихся в позициях beg , $beg+1$ и т. д. до конца списка. При иных значениях beg возвращается сообщение “*undefined*” (не определено).

B) Пусть $1 \leq beg \leq end \leq n$, где n – длина списка lis . Тогда по команде B возвращается список, состоящий из элементов списка lis , находящихся в позициях beg , $beg+1$, ..., end . При иных соотношениях между beg и end возвращается сообщение “*undefined*”.

C) Пусть $1 \leq beg \leq n$, где n – длина строки $text$. Тогда по команде C возвращается строка, состоящая из символов строки $text$, находящихся в позициях beg , $beg+1$ и т. д. до конца строки.

D) Пусть $1 \leq beg \leq end \leq n$, где n – длина строки $text$. По команде D возвращается строка, состоящая из символов строки $text$, находящихся в позициях beg , $beg+1$, ..., end . При иных соотношениях между beg и end возвращается сообщение “*undefined*” (не определено).

Примеры 21. Вырезки из списков и текстов.

- a) $Take[\{1, 2, 3, 4, 5\}, 3] \rightarrow \{3, 4, 5\}$, $Take[\{1, 2, 3, 4, 5\}, 6] \rightarrow undefined$;
b) $Take[\{“a”, “b”, “c”, “d”, “e”, “f”\}, 3, 5] \rightarrow \{“c”, “d”, “e”\}$;
c) $Take[“GeoGebra”, 4] \rightarrow “Gebra”$, $Take[“GeoGebra”, 1, 3] \rightarrow “Geo”$.

7.7. Объединение, соединение и пересечение списков

A. $Union[lis1, lis2, \dots, listk]$	C. $InterSection[lis1, lis2, \dots, listk]$
B. $Join[lis1, lis2, \dots, listk]$	

Объединение, соединение и пересечение списков

А) По команде *A* возвращается список, являющийся пересечением элементов первого уровня вложенности списков-аргументов. Выполнение операции можно представить себе так. В список помещаются все последовательные элементы сначала списка *lis1*, затем списка *lis2*, ..., и, наконец, списка *lisk*. Из полученного списка результирующий список строится удалением повторяющихся элементов.

В) По команде *B* возвращается список, являющийся соединением элементов первого уровня вложенности списков-аргументов. Выполнение операции проводится, как и в случае *A*, но повторяющиеся элементы не удаляются.

С) По команде *C* возвращается список, являющийся пересечением элементов первого уровня вложенности списков-аргументов. Выполнение операции можно представить себе так. В результирующий список помещаются все последовательные элементы списка *lis1*, которые имеются в каждом из списков *lis2*, ..., *lisk*.

Примеры 22. Объединение, соединение и пересечение списков.

- Пусть: $li=\{1, 1, 2, 3, (1,2), \{1\}\}$, $lis=\{2, 3, 5, \{1\}\}$. Тогда:
- a) $Union[li, lis] \rightarrow \{1, 2, 3, (1,2), \{1\}, 5\}$;
 - b) $Join[li, lis] \rightarrow \{1, 1, 2, 3, (1,2), \{1\}, 2, 3, 5, \{1\}\}$;
 - c) $InterSection[li, lis] \rightarrow \{2, 3, \{1\}\}$.

7.8. Суммы и произведения элементов списков

A. <i>Sum</i> [<i>lis</i>]	D. <i>ProductSum</i> [<i>lis</i>]	Суммирование и нахождение произведений элементов списков
B. <i>Sum</i> [<i>lis</i> , <i>n</i>]	E. <i>Product</i> [<i>lis</i> , <i>n</i>]	
C. <i>Sum</i> [<i>lis</i> , <i>lisfreq</i>]	F. <i>Product</i> [<i>lis</i> , <i>lisfreq</i>]	

В командах *Sum* аргумент *lis* – это список, элементами первого уровня вложенности которого могут быть числа, точки, векторы, тексты и функции. Все элементы в *lis* должны быть одного типа. В командах *Product* аргумент *lis* – это список чисел.

А) Рассмотрим возможные случаи:

- 1) $lis=\{a, b, c, \dots\}$, где $a, b, c \dots$ – действительные или комплексные числа. Тогда по *A* возвращается сумма элементов $a, b, c \dots$, то есть значение $a+b+c+\dots$;
- 2) $lis=\{(a1,a2), (b1,b2), \dots\}$, где $a1, b1, a2, b2, \dots$ – действительные числа. Тогда по *A* возвращается точка $(a1+a2+\dots, b1+b2+\dots)$. Точки (a,b) в *lis* можно задавать комплексными числами $a+b\cdot i$, а точки (a, a) – числами a ;
- 3) $lis=\{f1(x), f2(x), \dots\}$, где $f1(x), f2(x), \dots$ – встроенные или пользовательские функции одной переменной. Тогда по *A* на панели “Объекты” формируется описание функции $f1(x)+f2(x)+ \dots$, а на панели “Полотно” выводится ее график;

4) $lis = \{f1(x, y, \dots), f2(x, y, \dots), \dots\}$, где $f1(x, y, \dots), f2(x, y, \dots), \dots$ – встроенные или пользовательские функции многих переменных. Тогда по A на панели “Объекты” формируется описание функции $f1(x, y, \dots) + f2(x, y, \dots) + \dots$;

5) $lis = \{te1, te2, \dots\}$, где $te1, te2, \dots$ – строки. Тогда по A возвращается строка, полученная конкатенацией строк-аргументов.

В) Команда B выполняется аналогично A , но операция суммирования распространяется только на первые $k = \text{round}(n)$ аргументов lis . При k отрицательном или большем длины списка возвращается слово *undefined*.

С) Здесь списки lis и $lisfreq$ одинаковой длины с числовыми элементами. В $lisfreq$ находятся частоты соответствующих элементов lis . По C возвращается сумма произведений соответствующих элементов lis и $lisfreq$ (то есть “скалярное произведение” списков).

D) Команда D возвращает произведение элементов списка lis ;

E) Команда E возвращает произведение первых $k = \text{round}(n)$ элементов списка lis . При k отрицательном или большем длины списка возвращается слово *undefined*.

F) Команда F возвращает произведение элементов списка lis с учетом их кратностей, указанных в соответствующих позициях списка $lisfreq$.

Примеры 23. Суммы и произведения элементов списков.

- a) $Sum[\{1, 1, 2, 3\}] \rightarrow 7, \quad Product[\{1, 1, 2, 3\}] \rightarrow 6;$
- b) $Sum[\{1, 1, 2, 3\}, 3] \rightarrow 4, \quad Product[\{1, 1, 2, 3\}, 3] \rightarrow 2;$
- c) $Sum[\{1, 1, 2, 3\}, 5] \rightarrow undefined;$
- d) $Sum[\{1, 1, 2, 3\}, \{1, 2, 3, 2\}] \rightarrow 15,$
 $Product[\{1, 1, 2, 3\}, \{1, 2, 3, 2\}] \rightarrow 72;$
- e) $Sum[\{(1,1), (2,1), (2,3)\}] \rightarrow (5, 5);$
- f) $Sum[\{(1,1), (2,1), (2,3)\}, 2] \rightarrow (3, 2);$
- g) $Sum[\{x, \sin(x), x^3\}] \rightarrow x + \sin(x) + x^3;$
- h) $Sum[\{x, \sin(x), x^3\}, 2] \rightarrow x + \sin(x);$
- i) $Sum[\{x+y, \sin(x-y), x^3\}] \rightarrow x + y + \sin(x - y) + x^3;$
- j) $Sum[\{“Geo”, “Gebra”, “!”\}] \rightarrow GeoGebra!.$

7.9. Сортировка и связанные с ней команды

A. <i>Sort</i> [<i>lis</i>]	C. <i>Unique</i> [<i>lis</i>]	E. <i>OrdinalRank</i> [<i>lis</i>]
B. <i>Reverse</i> [<i>lis</i>]	D. <i>Frequency</i> [<i>lis</i>]	F. <i>TiedRank</i> [<i>lis</i>]

Сортировка и связанные с ней команды

A) По команде A возвращается список из элементов списка lis , отсортированных в неубывающем порядке. Сортировать можно числа, строки, точки и т. д. Все элементы списка lis должны быть одного типа. Строки сортируются в

лексикографическом порядке. Точки сортируются по первой компоненте, а при равных первых компонентах – по второй компоненте.

В) По команде *B* возвращается список из элементов первого уровня вложенности списка *lis*, записанных в обратном порядке (от конца к началу).

С) По команде *C* возвращается список из уникальных элементов списка *lis*, отсортированных в неубывающем порядке.

Д) По команде *D* возвращается список частот уникальных элементов списка *lis*, то есть элементов списка *Unique[*lis*]*.

Е) По команде *E* возвращается список рангов элементов списка *lis*, то есть их позиций в отсортированном по неубыванию списке *lis*.

Ф) По команде *F* возвращается список связанных или усредненных рангов элементов списка *lis*. Усредненные ранги получаются как средние значения рангов одинаковых элементов *lis*.

Примеры 24. Сортировки и связанные с ней команды.

- a) *Sort*[{5, 1, 2, 5, 3, 2}] → {1, 2, 2, 3, 5, 5};
- b) *Reverse*[{5, 1, 2, 5, 3, 2}] → {2, 3, 5, 2, 1, 5};
- c) *Reverse*[*Sort*[{5, 1, 2, 5, 3, 2}]] → {5, 5, 3, 2, 2, 1};
- d) *Unique*[{5, 1, 2, 5, 3, 2}] → {1, 2, 3, 5};
- e) *Frequency*[{5, 1, 2, 5, 3, 2}] → {1, 2, 1, 2};
- f) *OrdinalRank*[{5, 1, 2, 5, 3, 2}] → {5, 1, 2, 6, 4, 3};
- g) *TiedRank*[{5, 1, 2, 5, 3, 2}] → {5.5, 1, 2.5, 5.5, 4, 2.5};
- h) *Sort*[{"красный", "синий", "зеленый"}] → {"зеленый", "красный", "синий"};
- i) *Sort*[{(4,7), (3,5), (4,1), (2,3), (4,2)}] → {(2,3), (3,5), (4,1), (4,2), (4,7)};

7.10. Сглаживание и другие команды

A. <i>Flatten</i> [<i>lis</i>]	C. <i>Classes</i> [<i>lis</i> , <i>n</i>]	Разные команды
B. <i>PointList</i> [<i>lis</i>]	D. <i>Classes</i> [<i>lis</i> , <i>a</i> , <i>wi</i>]	

А) Сглаживание. По команде *A* реализуется “сглаживание” списка *lis*, то есть перемещение объектов всех его уровней на верхний (первый) уровень.

В) Списки в точки. Здесь *lis* – двухуровневый список, у которого на втором уровне каждый объект является двухэлементным списком из чисел. По команде *B* возвращает список точек, каждая из которых получена из соответствующего элемента *lis*, то есть внутреннего двухэлементного списка.

С) Разбиение элементов списка на классы. Пусть *a* и *b* – минимальный и максимальный элемент числового списка *lis*. По *C* возвращается список границ классов для данных из *lis* в виде {*a*, *a*+*wi*, *a*+2·*wi*, ..., *b*}, где *wi*=(*b*-*a*)/*n*.

Д) Разбиение элементов списка на классы. Пусть *wi*>0 и *b* – максимальный элемент числового списка *lis*. При *a*<*b* по *D* возвращается список границ

классов для данных из *lis* в виде $\{a, a+wi, a+2\cdot wi, \dots, a+k\cdot wi\}$, где k – натуральное и $a+(k-1)\cdot wi < b \leq a+k\cdot wi$. При $a \geq b$ возвращается слово *undefined*.

Примеры 25. Разные команды.

- a) *Flatten*[{"2", 3, (7,8), {5,1}, {{11, {5}}}}] → {"2", 3, (7,8), 5, 1, 11, 5};
- b) *PointList*[{{1,5},{2,6},{3,7}}] → {(1,5), (2,6), (3,7)}.
- c) *Classes*[{1,6,3,7,8,9,2,4}, 4] → {1, 3, 5, 7, 9}.

7.11. Представление списков объектами типа “раскрывающийся” список

Если через строку ввода ввести какой-либо список, то на панели “Объекты” он получит имя, но по умолчанию на панель “Полотно” выводиться не будет. Для того, чтобы получить геометрический объект, представляющий список, требуется выполнить установку:

RClick(описание списка)\Свойства\Basic\☑ *Draw as drop-down list*. (3)

В этом случае список длины n выводится в виде объекта “раскрывающийся” (выпадающий) список с n строками, на которых располагаются элементы первого уровня вложенности исходного списка. Например, при вводе списков {"сахар", "соль", "перец"} и {{1,2,3}, {1,9} 4, (5,6)} и выполнении для каждого из них команды (3) будут сформированы раскрывающиеся списки (см. рис. 2). Они могут быть представлены в свернутом или в раскрытом виде (*Click*(▼)). В заголовочной строке дублируется один из выборов списка. По умолчанию – это первый элемент. Щелчок по любому элементу списка дублирует его в заголовочную строку, то есть его делает текущим.

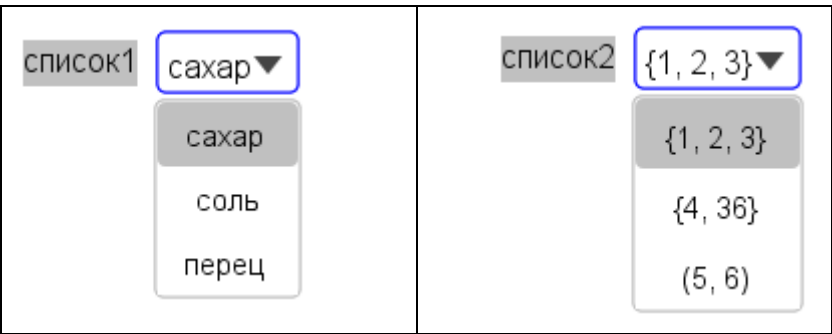


Рис. 2. Представление списков в виде объектов “раскрывающийся список”

A. <i>SelectedElement</i> [<i>lis</i>]
B. <i>SelectedIndex</i> [<i>lis</i>]

Команды для работы с раскрывающимися списками

А) По команде *A* на панель “Полотно” выводится, если он есть, геометрический образ текущего элемента раскрывающегося списка *lis*. Меняя в списке текущий элемент, будем получать актуализацию соответствующего ему образа.

В) По команде *B* выводится индекс текущего элемента раскрывающегося списка *lis*.

7.12. Работа с матрицами и векторами

7.12.1. Работа с матрицами

Мы уже отмечали, что векторы и матрицы создаются с помощью списков. А именно, если *lis* – гнездовой список длины *n* с двумя уровнями вложенности и его элементами являются списки одинаковой длины *m* с числовыми значениями, то при вводе *lis* формируется матрица размером $n \times m$ (*n* – строк и *m* – столбцов).

Примеры 26. В примере *a* определена матрица размером 2×4 , а в примере *b* – матрица размером 2×3 . При этом работа со второй матрицей, если хотя бы одна из величин *a*, *b*, *c* и *d* не определена, может проводиться только на панели символьных вычислений *GAS*.

$$a) \{\{1,2,3,4\}, \{0,7,9,3\}\} \rightarrow \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 7 & 9 & 3 \end{pmatrix}.$$

$$b) \{\{1,2,a\}, \{b,c,d\}\} \rightarrow \begin{pmatrix} 1 & 2 & a \\ b & c & d \end{pmatrix}.$$

Ниже в примерах данного раздела мы будем использовать следующие четыре матрицы:

$$li1 = \{\{7,0,7,8,1\}, \{16,0,16,3,13\}, \{9,4,1,6,12\}, \{7,-4,15,-3,1\}\} \rightarrow$$

$$\begin{pmatrix} 7 & 0 & 7 & 8 & 1 \\ 16 & 0 & 16 & 3 & 13 \\ 9 & 4 & 1 & 6 & 12 \\ 7 & -4 & 15 & -3 & 1 \end{pmatrix}; li2 = \{\{1,8,-2\}, \{3,1,7\}, \{5,3,2\}\} \rightarrow \begin{pmatrix} 1 & 8 & -2 \\ 3 & 1 & 7 \\ 5 & 3 & 2 \end{pmatrix};$$

$$li3 = \{\{1\}, \{3\}, \{2\}\} \rightarrow \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}; li4 = \{\{1,3,2\}\} \rightarrow (1 \ 3 \ 2).$$

A. $c+lis$ B. $lis+c$	C. $c-lis$ D. $lis-c$	E. $c*lis$ F. $lis*c$	G. c/lis H. lis/c	I. $lis1+lis2$ J. $lis1-lis2$	K. $lis1/lis2$ L. $g(lis)$
--------------------------	--------------------------	--------------------------	--------------------------	----------------------------------	-------------------------------

Поэлементные арифметические операции над матрицами

Здесь lis , $lis1$, $lis2$ – матрицы размером $n \times m$, c – скалярное выражение.

A, B) Выражения A и B возвращают матрицу, у которой элемент в позиции (i, j) равен значению $Element[lis, i, j] + c$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операция “+” в данном случае называется сложением матрицы lis со скаляром c .

C, D) Выражения C и D . По C возвращается матрица, у которой элемент в позиции (i, j) равен значению $c - Element[lis, i, j]$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). По D возвращается матрица, у которой элемент в позиции (i, j) равен значению $Element[lis, i, j] - c$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операции “-” в данном случае называются соответственно вычитанием из скаляра c матрицы lis и вычитанием из матрицы lis скаляра c .

E, F) Выражения E и F возвращают матрицу, у которой элемент в позиции (i, j) равен значению $c * Element[lis, i, j]$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операции “*” в данном случае называются умножением матрицы lis на скаляр c .

G, H) По выражению G возвращается матрица, у которой элемент в позиции (i, j) равен значению $c / Element[lis, i, j]$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). По выражению H возвращается матрица, у которой элемент в позиции (i, j) равен значению $Element[lis, i, j] / c$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операции “/” в данном случае называются соответственно делением скаляра c на матрицу lis и делением матрицы lis на скаляр c . Заметим, что при делении неотрицательного числа на 0 возвращается символ “ ∞ ”, а отрицательного – “ $-\infty$ ”.

I) Выражение I возвращает матрицу, у которой элемент в позиции (i, j) равен значению $Element[lis1, i, j] + Element[lis2, i, j]$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операция “+” в данном случае называется сложением матриц $lis1$ и $lis2$.

J) Выражение J возвращает матрицу, у которой элемент в позиции (i, j) равен значению $Element[lis1, i, j] - Element[lis2, i, j]$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операция “-” в данном случае называется вычитанием матриц $lis1$ и $lis2$.

K) Выражение K возвращает матрицу, у которой элемент в позиции (i, j) равен значению $Element[lis1, i, j] / Element[lis2, i, j]$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операция “/” в данном случае называется поэлементным делением матриц $lis1$ и $lis2$.

L) Здесь g – какая-либо встроенная или пользовательская функция с одним числовым аргументом, lis – матрица размером $n \times m$. Выражение L возвращает матрицу, у которой элемент в позиции (i, j) равен значению $g(Element[lis, i, j])$ ($i=1, 2, \dots, n; j=1, 2, \dots, m$). Операция F называется применением функции g к матрице lis .

Примеры 27. Поэлементные арифметические операции над матрицами.

$$a) uu=li2+2 \rightarrow \begin{pmatrix} 3 & 10 & 0 \\ 5 & 3 & 9 \\ 7 & 5 & 4 \end{pmatrix}, vv=li2-2 \rightarrow \begin{pmatrix} -1 & 6 & -4 \\ 1 & -1 & 5 \\ 3 & 1 & 0 \end{pmatrix};$$

$$b) 2*li2 \rightarrow \begin{pmatrix} 2 & 16 & -4 \\ 6 & 2 & 14 \\ 10 & 6 & 4 \end{pmatrix}, li2/2 \rightarrow \begin{pmatrix} 0.5 & 4 & -1 \\ 1.5 & 0.5 & 3.5 \\ 2.5 & 1.5 & 1 \end{pmatrix};$$

$$c) uu+vv \rightarrow \begin{pmatrix} 2 & 16 & -4 \\ 6 & 2 & 14 \\ 10 & 6 & 4 \end{pmatrix}, uu-vv \rightarrow \begin{pmatrix} 4 & 4 & 4 \\ 4 & 4 & 4 \\ 4 & 4 & 4 \end{pmatrix};$$

$$d) \sin(li2) \rightarrow \begin{pmatrix} 0.84 & 0.99 & -0.91 \\ 0.14 & 0.84 & 0.66 \\ -0.96 & 0.14 & 0.91 \end{pmatrix}, \text{floor}[3*\sin(li2)] \rightarrow \begin{pmatrix} 2 & 2 & -3 \\ 0 & 2 & 1 \\ -3 & 0 & 2 \end{pmatrix}.$$

A. $lis1*lis2$	C. lis^0	E. $lis^{(-k)}$
B. lis^k	D. lis^{-1}	

Арифметические операции над матрицами

A) Умножение матриц. Здесь при матрице $lis1$ размером $n \times p$ вторая матрица $lis2$ должна иметь размер $p \times m$, то есть количество столбцов $lis1$ должно быть равно количеству строк $lis2$. В этом случае выражение A возвращает матрицу, у которой элемент в позиции (i,j) равен значению

$$\sum_{k=1}^p Element[li1,i,k] \cdot Element[li1,k,j] \quad (i=1, 2, \dots, n; j=1, 2, \dots, m).$$

Операция “*” в данном случае называется умножением матрицы $lis1$ на матрицу $lis2$.

B) Выражение B для квадратной матрицы lis и натуральном k возвращает матрицу, равную $ma*ma*\dots*ma$ (k умножений).

C) Выражение C для квадратной матрицы lis возвращает единичную матрицу такого же размера, что и lis . Для неквадратных матриц lis возвращается пустой список.

D) Выражение D для квадратной матрицы lis возвращает обратную матрицу, если она существует. Для неквадратных матриц и для квадратных вырожденных матриц lis возвращается пустой список. Отметим, что функцией $Invert[li1]$ в этих случаях возвращается слово *undefined* (см. ниже).

E) Выражение E при натуральном k вычисляется как $(lis^{(-1)})^k$.

Примеры 28. Арифметические операции над матрицами.

a) $li2 * li2 \rightarrow \begin{pmatrix} 15 & 10 & 50 \\ 41 & 46 & 15 \\ 24 & 49 & 15 \end{pmatrix}, li2^2 \rightarrow \begin{pmatrix} 15 & 10 & 50 \\ 41 & 46 & 15 \\ 24 & 49 & 15 \end{pmatrix};$

b) $li2^0 \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, li2^{(-1)} \rightarrow \begin{pmatrix} -0.09 & -0.11 & 0.28 \\ 0.14 & 0.06 & -0.06 \\ 0.02 & 0.18 & -0.11 \end{pmatrix}.$

A. <i>Dimension</i> [lis]	D. <i>Identity</i> [n]	G. <i>ApplyMatrix</i> [lis, P]
B. <i>MatrixRank</i> [lis]	E. <i>Transpose</i> [lis]	H. <i>lis</i> *P
C. <i>Determinant</i> [lis]	F. <i>Invert</i> [lis]	I. <i>ReducedRowEchelonForm</i> [lis]

Команды для работы с матрицами

A) Если *lis* – список, определяющий матрицу, то по команде *A* возвращается двухэлементный список {*n*, *m*}, где *n* и *m* – натуральные числа, равные соответственно количеству строк и количеству столбцов матрицы.

B) По команде *B* вычисляется ранг матрицы, формируемой по списку *lis*.

Примеры 29.

a) $Dimension[li1] \rightarrow \{4,5\}, Dimension[li2] \rightarrow \{3,3\}, Dimension[\{\}] \rightarrow 0;$

b) $MatrixRank[li1] \rightarrow 3, MatrixRank[li2] \rightarrow 3,$
 $MatrixRank[\{\}] \rightarrow undefined.$

C) По команде *C* вычисляется определитель квадратной матрицы, формируемой по списку *lis*.

D) По команде *D* возвращается единичная матрица размером *n*×*n*.

E) По команде *E* возвращается матрица, которая является транспонированной по отношению к матрице, формируемой по списку *lis*. Напомним, что транспонированная матрица из исходной матрицы получается заменой в ней строк на столбцы.

F) По команде *F* для квадратной матрицы, формируемой по списку *lis*, возвращается обратная матрица, если она существует. Для неквадратных матриц и для квадратных матриц *lis* при отсутствии обратной матрицы возвращается слово *undefined*. Условие существования обратной матрицы формулируется так: “определитель исходной матрицы отличен от нуля (*Determinant*[*lis*]≠0)”, или “ранг исходной матрицы равен количеству ее строк (столбцов)”. Если для матрицы *A* существует обратная матрица *B*, то *A* называют невырожденной матрицей, при этом $A \cdot B = B \cdot A = E$, где *E* – единичная матрица такого же размера, что и *A*.

G-H) Матрицу *ma* размером $n \times n$ ($n=2$ или $n=3$) можно умножать на матрицу *v* с одним столбцом размером $n \times 1$, то есть на вектор. При этом возвращается матрица с одним столбцом размером $n \times 1$, то есть снова вектор. Например:

$$\{\{1,3\}, \{5,2\}\} * \{\{1\}, \{2\}\} \rightarrow \begin{pmatrix} 7 \\ 9 \end{pmatrix} \quad (\{\{7\}, \{9\}\} \rightarrow \begin{pmatrix} 7 \\ 9 \end{pmatrix}).$$

Далее, матрицу *ma* нельзя умножить на матрицу *v* с одной строкой размером $1 \times n$. Однако на *n*-мерную точку *P* при $n=2$ или $n=3$ матрицу *ma* “умножать” можно, например,

$$\{\{1,3,0\}, \{5,2,1\}, \{1,1,3\}\} * (2,1,1) \rightarrow (5, 13, 6) \quad (\text{это точка!})$$

или

$$\text{ApplyMatrix}[\{\{1,3,0\}, \{5,2,1\}, \{1,1,3\}\}, (2,1,1)] \rightarrow (5, 13, 6).$$

Выполнение такого умножения можно представлять себе так:

при $n=2$

$$Q = ma * \{\{x(P)\}, \{y(P)\}\}, \\ (Element[Q, 1, 1], Element[Q, 2, 1]) \rightarrow 2D \text{ точка};$$

при $n=3$

$$Q = ma * \{\{x(P)\}, \{y(P)\}, \{z(P)\}\}, \\ (Element[Q, 1, 1], Element[Q, 2, 1], Element[Q, 3, 1]) \rightarrow 3D \text{ точка}.$$

Замечание. Команда *G* может применяться и в иных случаях, которые мы не описывали.

Примеры 30. Команды для работы с матрицами.

$$a) \text{Determinant}[li1] \rightarrow \text{undefined}, \text{Determinant}[li2] \rightarrow 205;$$

$$b) \text{Identity}[2] \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \quad c) \text{Transpose}[li2] \rightarrow \begin{pmatrix} 1 & 3 & 5 \\ 8 & 1 & 3 \\ -2 & 7 & 2 \end{pmatrix};$$

$$d) \text{Invert}[li2] \rightarrow \begin{pmatrix} -0.09 & -0.11 & 0.28 \\ 0.14 & 0.06 & -0.06 \\ 0.02 & 0.18 & -0.11 \end{pmatrix}, \quad li2 \cdot \text{Invert}[li2] \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

$$e) \text{ApplyMatrix}[\{\{1,3\}, \{5,2\}\}, (2,1)] \rightarrow (5, 12),$$

$$ma = \{\{1,3\}, \{5,2\}\} * \{\{2\}, \{1\}\} \rightarrow \begin{pmatrix} 5 \\ 12 \end{pmatrix},$$

$$(Element[ma, 1, 1], Element[ma, 2, 1]) \rightarrow (5, 12) \quad (\text{это точка!})$$

$$f) \text{ApplyMatrix}[\{\{1,3,0\}, \{5,2,1\}, \{1,1,1\}\}, (2,1,3)] \rightarrow (5, 15, 6).$$

Л) По команде *I* для матрицы, формируемой по списку *lis*, возвращается ее верхняя ступенчатая форма с ведущими элементами, приведенными к единице. Вычисления реализуются исключениями Гаусса-Жордана, при которых подстановки делаются во все строки, кроме текущей (ведущей) строки. Заметим, что при исключениях Гаусса подстановки реализуются только в строки с номерами, большими номера текущей строки.

Пример 31. Верхняя ступенчатая форма матрицы

$$\begin{aligned}
 \text{a) } \textit{ReducedRowEchelonForm}[\textit{li1}] &\rightarrow \begin{pmatrix} 1 & 0 & 1 & 0 & 0.94 \\ 0 & 1 & -2 & 0 & 1.93 \\ 0 & 0 & 0 & 1 & -0.7 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \\
 \text{b) } \textit{ReducedRowEchelonForm}[\textit{li2}] &\rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.
 \end{aligned}$$

7.12.2. Работа с векторами

Здесь рассмотрены некоторые команды для работы с векторами. И хотя списки в них не используются, описание этих команд естественно дать в том же разделе, где проведено обсуждение средств для работы с матрицами.

A. <i>Vector</i> [<i>P</i>] B. <i>Vector</i> [<i>P</i> , <i>Q</i>]	C. <i>UnitVector</i> [<i>obj</i>]	D. <i>UnitPerpendicularVector</i> [<i>obj</i>] E. <i>PerpendicularVector</i> [<i>obj</i>]
---	-------------------------------------	--

Команды для работы с векторами

А) По команде *A* возвращается вектор между началом координат и точкой *P*, то есть вектор $\begin{pmatrix} x[P] \\ y[P] \end{pmatrix}$.

В) По команде *A* возвращается вектор между точками *P* и *Q*, то есть вектор $\begin{pmatrix} x[Q] - x[P] \\ y[Q] - y[P] \end{pmatrix}$.

С) По команде *C* возвращается вектор единичной длины, имеющий то же самое направление и ориентацию, что и *obj*. Для аргумента *obj* возможны такие случаи:

- *obj* – вектор. Тогда возвращается вектор, выходящий из начала координат;

- *obj* – прямая или луч между точками *A* и *B*. Тогда возвращается вектор, выходящий из точки *A*. Причем делается это как при видимой, так и при спрятанной точке *A*;

- *obj* – прямая или луч, выведенные командами *Line* или *Ray*. Тогда возвращается вектор, выходящий из начала координат;

- *obj* – отрезок между точками *A* и *B*. Тогда возвращается вектор, выходящий из точки *A*. Причем делается это как при видимой, так и при спрятанной точке *A*;

- *obj* – отрезок, выведенный командой *Segment*. Тогда возвращается вектор, выходящий из начала координат.

D) По команде *D* возвращается вектор, который получается поворотом против часовой стрелки на 90° вектора, возвращаемого по *C*.

E) По команде *E* возвращается вектор, который имеет направление единичного вектора, возвращаемого по *D*, и длину *r*, вычисляемую следующим образом. Пусть *obj* лежит на прямой $a \cdot x + b \cdot y = c$, тогда $r = \sqrt{a^2 + b^2}$.

7.13. Выполнение команд из списков

A. *Execute*["Com1", "Com2", ..., "Comk"]

B. *Execute*["Com1", "Com2", ..., "Comk", t1, t2, ..., tm]

Выполнение команд из списков

Здесь *Com1*, *Com2*, ..., *Comk* – некоторые команды.

A) Элементами списков могут быть команды, не содержащие неопределенных параметров. Для последовательного выполнения команд их следует задать списком в строковом виде в английской нотации как аргумент функции *Execute*. Команды выполняются слева направо в порядке их следования. Например, если ввести функцию $g = x^3 - 10 \cdot x + 1$, а затем команду:

Execute["Limit[g/x^3, +∞]", "Derivative[g]", "Integral[g]"]

(вычислить: предел g/x^3 при $x \rightarrow +\infty$, производную от $g(x)$ и неопределенный интеграл от $g(x)$), то на выходе получим результаты выполнения каждой из команд-аргументов, то есть: сначала предел – 1, затем производную – $(3x^2 - 10)$ и, наконец, интеграл – $(0.25x^4 - 5x^2 + x)$. Отметим, что в командах можно использовать результаты выполнения предыдущих команд. Например, по

Execute["d=Derivative[g]", "Integral[g/d]"]

формируются функции

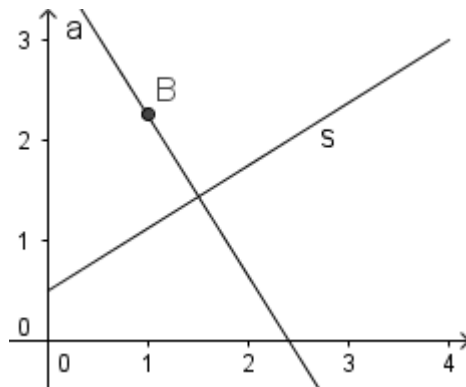
$$d(x) = 3x^2 - 10,$$

$$f(x) = \frac{1}{6}x^2 + \frac{1}{2} \cdot \frac{\ln\left(\left|\frac{6x-2\sqrt{30}}{6x+2\sqrt{30}}\right|\right)}{\sqrt{30}} - \frac{10}{9}\ln(|3x^2-10|),$$

а также выводятся их графики.

В) Элементами списков могут быть команды, содержащие неопределенные параметры, записываемые в виде %1, %2, ... %m. Для последовательного выполнения таких команд их следует задать списком в строковом виде в английской нотации как первый аргумент функции *Execute*. Другими аргументами *Execute* должны быть фактические значения t_1, t_2, \dots, t_m для неопределенных параметров %1, %2, ... %m. Перед выполнением команд все неопределенные параметры в них замещаются их фактическими значениями. Например, на изображении слева показана команда *Execute*, а справа – вывод по ней.

```
Execute[{"s=Segment[%1, %2]",
        "B=Midpoint[%1, (2,4)]",
        "PerpendicularLine[B, s]"},
        (0,0.5), (4,3)]
```



В данном случае выполнение *Execute* протекает так:

- в текст `"s=Segment[%1, %2]"` вместо неопределенных параметров %1 и %2 подставляются соответственно точки (0,0.5) и (4,3). Полученной командой `s=Segment[(0,0.5), (4,3)]` между позициями этих точек выводится отрезок *s* без вывода самих точек;
- в текст `"B=Midpoint[%1, (2,4)]"` вместо неопределенного параметра %1 подставляется точка (0, 0.5). Полученной командой `B=Midpoint[(0,0.5), (2,4)]` на воображаемом отрезке между точками (0, 0.5) и (2, 4) формируется средняя точка *B*;
- командой `PerpendicularLine[B, s]` по результатам предыдущих вычислений (*s* и *B*) из точки *B* проводится перпендикуляр к отрезку *s*.

8. Некоторые задачи дискретной математики

Для дальнейшего изложения нам понадобятся некоторые понятия, связанные с теорией графов. На этом и остановимся. Графом называют совокупность непустого множества вершин (точек) и множества ребер (линий) – наборов некоторых пар вершин. Две вершины на графе называют смежными, если они соединяются общим ребром, а про сами вершины в этом случае говорят, что они инцидентны данному ребру. Граф называют связным, если от любой из его вершин по ребрам можно переместиться к любой другой вершине. Связный граф, в котором между любой парой вершин имеется единственный путь, называют деревом. Граф называется полным, если каждую пару его вершин соединяет ребро. Если на всех ребрах определено направление от одной из вершин к другой вершине, то ребра называют дугами, а граф – ориентированным. Если ни на одном из ребер направление не определено, то граф называют неориентированным. Если в графе имеются как ориентированные, так и неориентированные ребра, то его называют смешанным. Если на ребрах (дугах) графа определена некоторая функция с действительными числовыми значениями, то граф называют сетью. Например, функция, ставящая в соответствие ребрам их реальные евклидовы длины, превращает граф в сеть. В общем случае в сети каждому ребру (дуге) ставится в соответствие некоторое действительное, обычно неотрицательное, число.

8.1. Задача о кратчайшем пути

Будем рассматривать только неориентированные графы. Два ребра называют смежными, если они имеют общую вершину. Путем в графе (сети) между двумя его вершинами U и V будем называть последовательность смежных ребер, только первое из которых содержит U , и только последнее из них содержит V . С помощью графов формулируются многие реальные задачи. Например, если вершинами считать конкретные пункты на местности, а ребрами – дороги между этими пунктами, то можно рассматривать такие постановки задачи о кратчайшем пути.

Задача 1. В графе зафиксированы две вершины U и V (два пункта). Найти кратчайший путь между U и V , то есть путь с минимальным количеством ребер (дорог), по которому можно переместиться из вершины U в вершину V . Заметим, что таких путей может быть несколько.

Задача 2. В сети зафиксированы две вершины U и V (два пункта). Найти кратчайший путь между U и V , то есть путь с минимальным евклидовым расстоянием, по которым можно переместиться из вершины U в вершину V . Заметим, что таких путей может быть несколько.

Для решения задачи о кратчайших путях часто используется алгоритм Дейкстры [50]. По нему кратчайшие пути находятся от одной вершины сразу до всех остальных вершин графа или сети при любой функции на ребрах, принимающей неотрицательные значения.

Существуют и иные постановки задачи о кратчайших путях и алгоритмы для нахождения их решений [22].

A. *ShortestDistance*[*lis*, *beg*, *end*, *bool*]

Задача о кратчайшем пути
 А) Здесь: *lis* – список ребер неориентированного графа; *beg* – стартовая вершина графа; *end* – конечная вершина графа; *bool* – логический управляющий параметр. Рассмотрим возможные варианты:

- 1) *bool=false*. В этом случае решается задача 1.
- 2) *bool=true*. В этом случае решается задача 2.

Пример 1. Командой *Execute* введем вершины графа (точки):

Execute[{"A=(1,1)", "B=(3,5)", "C=(2,4)", "D=(3,0)",
 "E=(5,4)", "F=(7,2)", "G=(6,5)", "H=(3,3)", "I=(5,2)"}].

По точкам A, B, \dots, I командами *Execute* и *Segment* сформируем ребра графа (отрезки):

Execute[{"S1=Segment[A,C]", "S2=Segment[A,D]", "S3=Segment[A,H]",
 "S4=Segment[C,B]", "S5=Segment[C,E]", "S6=Segment[H,E]",
 "S7=Segment[H,I]", "S8=Segment[H,D]", "S9=Segment[D,I]",
 "S10=Segment[D,E]", "S11=Segment[E,G]", "S12=Segment[I,F]",
 "S13=Segment[F,G]", "S14=Segment[D,F]", "S15=Segment[B,G]"}].

Пусть требуется найти кратчайший путь в этом графе между вершинами A и G при *bool=false*. Тогда команда, решающая задачу, может быть, например, такой:

gr=ShortestDistance[
 {S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15}, A, G, false].

На рис. 1а справа показан найденный по команде *ShortestDistance* граф и кратчайший путь между его вершинами длины 3, то есть путь, состоящий из трех звеньев AC, CB и BG . Метки с ребер графа (отрезков) спрятаны. Слева видна часть панели “Объекты” с текущими координатами вершин графа. При неоднозначном решении задачи иногда удается получить новый путь простой

перекомпиляцией кода (*View\Recompile All Objects* или *Ctrl+r*). Далее, все вершины графа – независимые точки. Перемещение вершин по плоскости изменяет их позиции и может привести к другому кратчайшему пути с тем же значением его длины – 3 (см. рис. 1b).

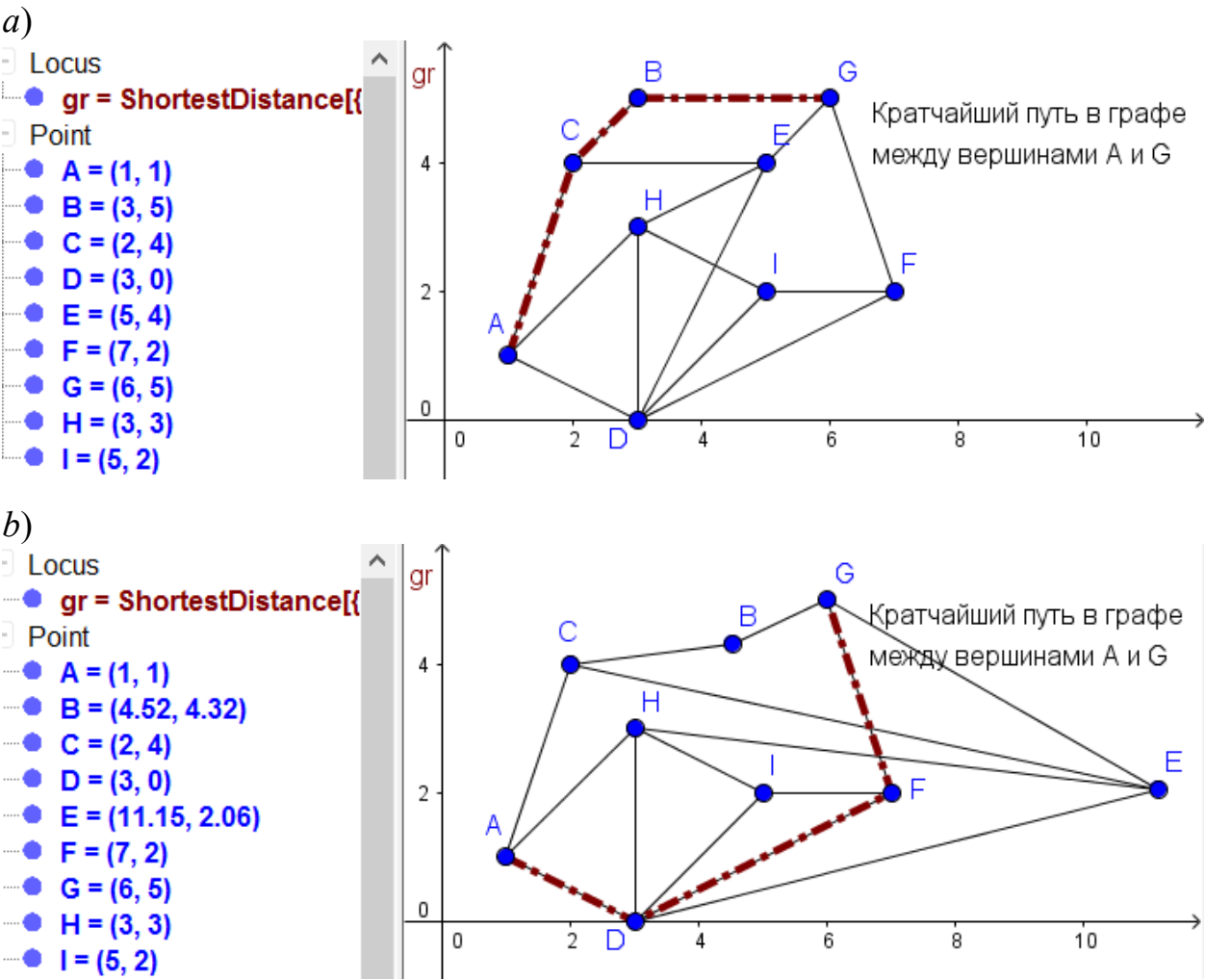


Рис. 1 (a, b). Кратчайший путь в графе между вершинами A и G (длина ребра 1)

Теперь решим ту же самую задачу при *bool=true*. В новом документе введем тот же самый граф (те же вершины и ребра) и затем команду.

```
gr=ShortestDistance[
  {S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15}, A, G, true].
```

На рис. 2a показан найденный по команде *ShortestDistance* кратчайший путь между вершинами A и G. Длину этого пути можно подсчитать с помощью функции *Distance*: $r=Distance[A,H]+Distance[H,E]+Distance[E,G]$. В результате получим $r=6.48$. Если решение задачи неоднозначное, то новый путь с тем же расстоянием можно получить простой перекомпиляцией кода. Далее, поскольку все вершины сети – независимые точки, то перемещение некоторых из них изменяет сеть и может привести к другому кратчайшему пути (см. рис. 2b).

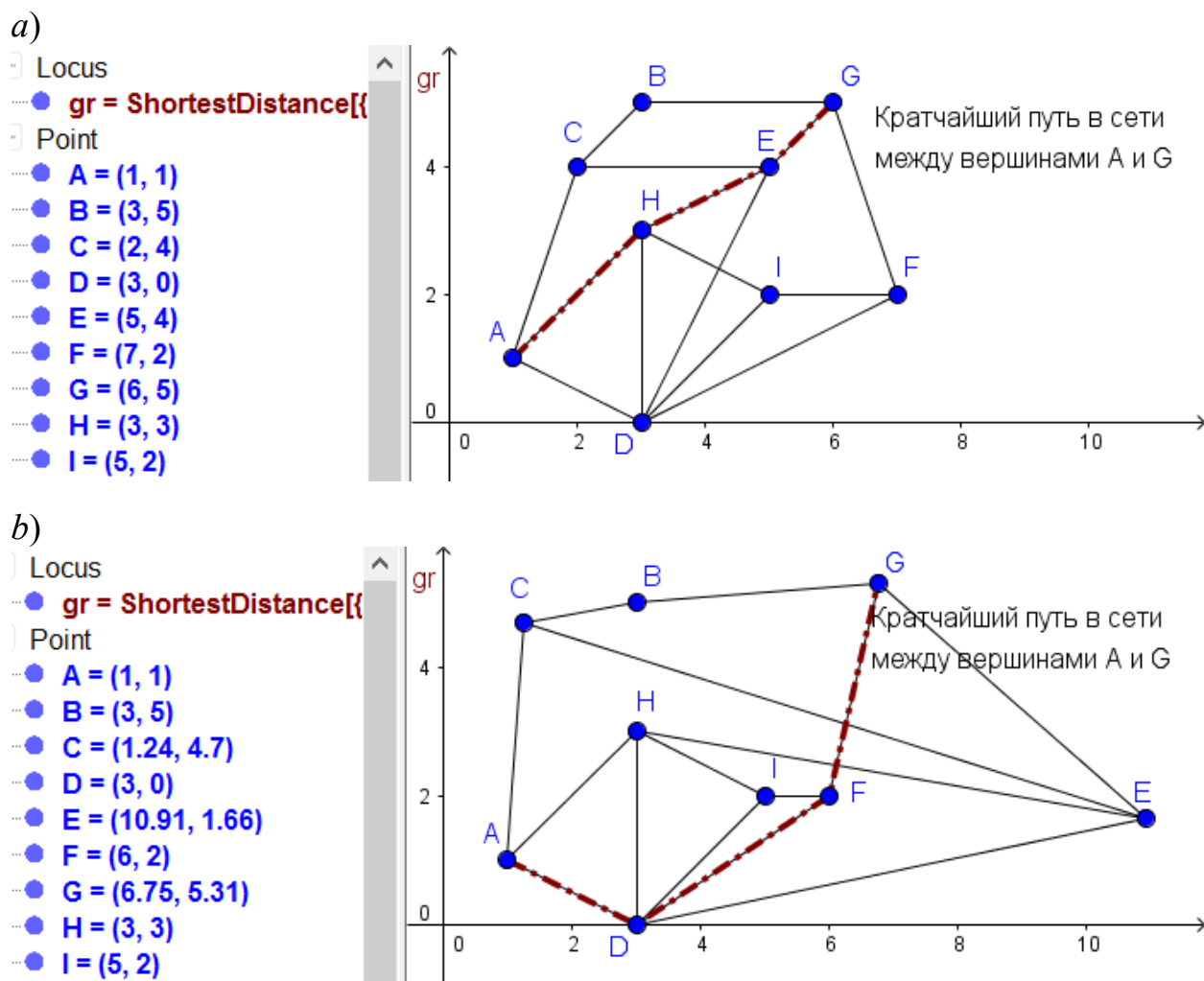


Рис. 2 (a, b). Кратчайший путь в сети между вершинами A и G (при евклидовой длине ребра)

Замечания.

1. Командой *ShortestDistance* в версии *GeoGebra* 5.0.212 нельзя решить задачу о кратчайшем пути для сетей общего вида с любой функцией на ребрах, принимающей неотрицательные значения. Этой командой задача решается лишь тогда, когда функция возвращает евклидовы длины соответствующих ребер.

2. Сеть, то есть вершины и ребра, совсем не обязательно вводить с помощью команд. Ее можно нарисовать на панели “Полотно”, используя инструменты “ Point” и “ Segment”. В этом случае для решения задачи о кратчайших путях можно сразу воспользоваться командой *ShortestDistance*.


8.2. Задача коммивояжера

Задача коммивояжера формулируется так. Пусть имеется n ($n > 1$) населенных пунктов и между каждой парой пунктов есть дорога. Коммивояжер (агент

по сбыту), отправляясь из своего населенного пункта, кратчайшим маршрутом должен посетить $n-1$ других населенных пунктов, побывав в каждом из них по одному разу, и вернуться назад. Задача коммивояжера состоит в отыскании хотя бы одного такого маршрута. Математические модели задачи коммивояжера, как правило, содержат большое количество переменных и ограничений. Поэтому для их решения общие методы линейного программирования обычно не используются. Предпочтение отдается вычислительной схеме, известной под названием “метод ветвей и границ”. При небольших n с задачей коммивояжера можно справиться перебором всех замкнутых маршрутов, проходящих по одному разу через каждый из городов.

A. *TravelingSalesman*[*lis*]

Задача коммивояжера

A) В *GeoGebra* задача коммивояжера формулируется для сети, являющейся полным неориентированным графом, ребрам которого ставятся в соответствие их евклидовы длины. В задаче отыскивается замкнутый путь (маршрут) минимальной длины, проходящий через все вершины по одному разу. Решение задачи коммивояжера можно получить по команде *A*, где *lis* – список вершин сети. Для выполнения *A* предварительно требуется формирование вершин сети. Их как точки можно ввести непосредственно через строку ввода или просто “нарисовать” на панели “Полотно” инструментом “ *Point*”. При этом по *A* выводятся только ребра маршрута, а не все ребра сети. Если же требуется вывести все ребра сети, то об этом необходимо позаботиться отдельно. Например, по списку вершин *lis* все ребра можно вывести командой

$$\text{Sequence}[\text{Sequence}[\text{Segment}[\text{Element}[\text{lis}, i], \text{Element}[\text{lis}, j]], i, 1, \text{Length}[\text{lis}]], j, 1, \text{Length}[\text{lis}]]. \quad (3)$$

Пример 2. Введем вершины сети:

$$\text{Execute}[\{ "A=(1,1)", "B=(3,5)", "C=(2,4)", "D=(3,0)", "E=(5,4)", "F=(7,2)", "G=(6,5)", "H=(3,3)", "I=(5,2)" \}].$$

Команда, решающая задачу коммивояжера, может быть записана, например, так:

$$\text{gra}=\text{TravelingSalesman}[\{A,B,C,D,E,F,G,H,I\}].$$

На рис. 3а справа показаны вершины сети и найденный на ней по команде *TravelingSalesman* кратчайший маршрут коммивояжера. Длина пути равна сумме длин составляющих его звеньев: $\text{Distance}[A,H]+\text{Distance}[H,C]+\dots+\text{Distance}[D,A]=19.53$. Если звеньев в маршруте много, то проще подсчитать его длину, введя список вершин и затем выполнив команду *Sum*:

$$\text{li}=\{A,H,C,B,E,G,F,I,D,A\},$$

$$Sum[Sequence[Segment[Element[li, k], Element[li, k+1]], k, 1, Length[li]-1]].$$

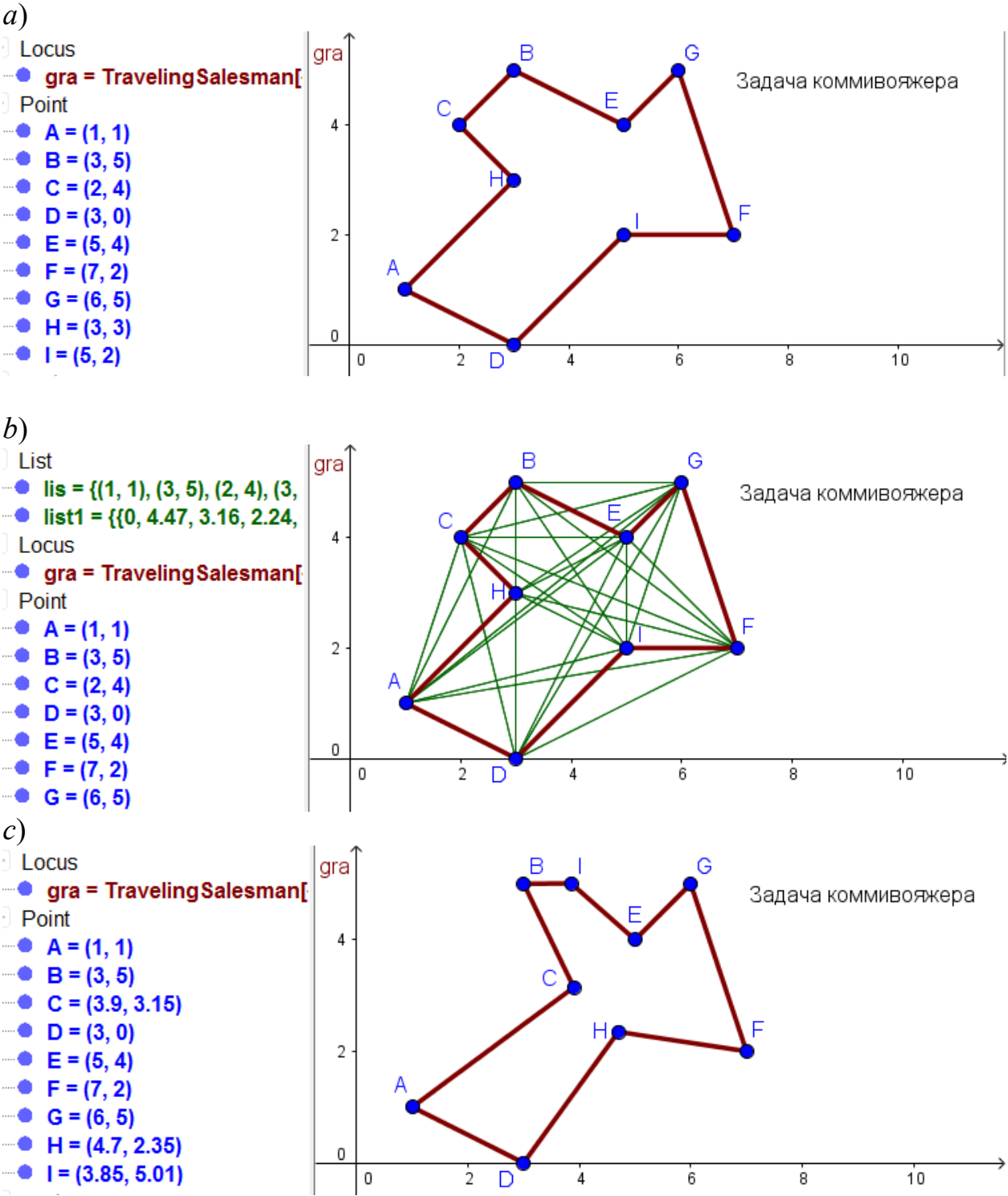



Рис. 3 (a, b, c). Маршрут коммивояжера (при евклидовой длине ребра)

На рис. 3b дополнительно к кратчайшему маршруту выведена сеть. Сделано это формированием списка $lis=\{A,B,C,D,E,F,G,H,I\}$ с последующим выполнением команды (1). Если решение задачи неоднозначное, то новое решение иногда удастся получить простой перекомпиляцией кода. Далее, поскольку все вершины сети – независимые точки, то перемещение некоторых

из них изменяет сеть и может привести к другому кратчайшему маршруту коммивояжера (см. рис. 3с).

Замечание. Вершины сети совсем не обязательно формировать с помощью команд. Их можно вывести на панель “Полотно”, используя инструмент “ Point”. В этом случае для решения задачи о кратчайшем маршруте коммивояжера можно сразу воспользоваться командой *TravelingSalesman*.

8.3. Нахождение выпуклой оболочки точек

Пусть M – непустое конечное множество из n точек на плоскости. Среди M могут быть повторяющиеся (кратные) точки. Выпуклой оболочкой M называют минимальный по площади выпуклый многоугольник P , содержащий все точки M . Напомним, что многоугольник P на плоскости называется выпуклым, если при проведении прямой через любую из его сторон в одной из полученных замкнутых полуплоскостей оказываются все точки P (во второй открытой полуплоскости точек P нет). В некоторых случаях выпуклой оболочкой точек M считают совокупность тех точек M , которые лежат на границе P , или только тех точек, которые являются вершинами P . Если представить себе точки M в виде вбитых в плоскость перпендикулярных штырьков, то выпуклая оболочка M – это многоугольник, границу которого принимает натянутое на штыри резиновое кольцо. Для нахождения выпуклой оболочки точек существует весьма эффективный алгоритм Р. Л. Грэхэма. Этот и иные алгоритмы решения данной задачи описаны в [22,31,35].

A. <i>ConvexHull</i> [<i>lis</i>]

Нахождение выпуклой оболочки точек
A) Командой *A*, где *lis* – список исходных точек плоскости, находится и возвращается выпуклая оболочка этих точек.

Пример 3. Введем точки M :

$$\text{Execute}[\{"A=(1,1)", "B=(3,5)", "C=(2,4)", "D=(3,0)", "E=(5,4)", "F=(7,2)", "G=(6,5)", "H=(3,3)", "I=(5,2)", "J=(5,1)"\}].$$

(4)

Команда, по которой находится выпуклая оболочка этих точек, может быть записана, например, так:

$$gr=ConvexHull[\{A,B,C,D,E,F,G,H,I,J\}].$$

При ее вводе на панелях “Объекты” и “Полотно” получим изображение, представленное на рис. 4.

Все выведенные точки являются независимыми. Перемещая некоторые из них, мы будем получать динамическую картинку – другие задачи для нахождения выпуклой оболочки точек и в реальном времени – ее построение. Если происходит добавление на плоскость новых точек, то для того, чтобы они участвовали в построении выпуклой оболочки, следует заново выполнить команду *ConvexHull*, добавив в ее список-аргумент эти точки.

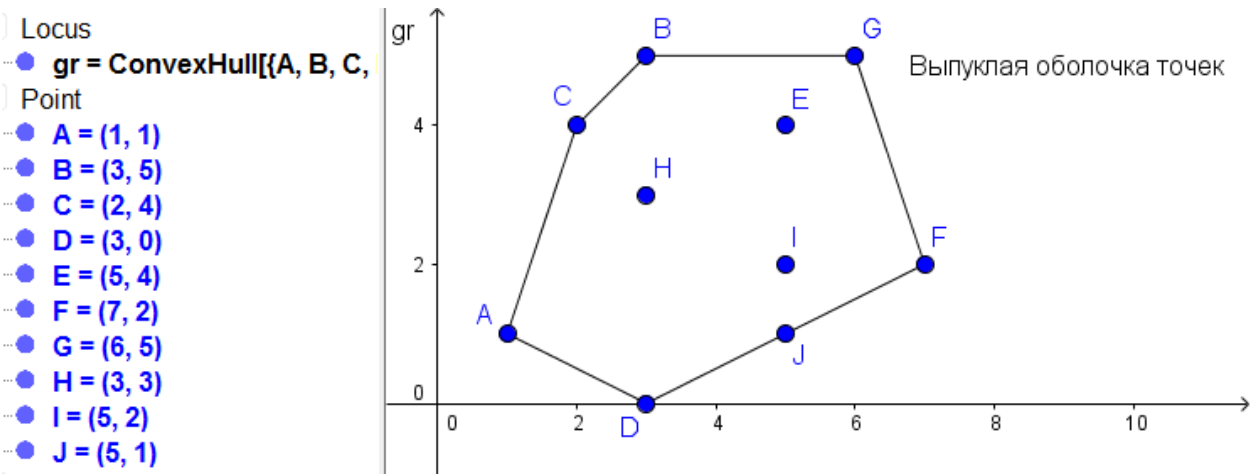


Рис. 4. Нахождение выпуклой оболочки точек

8.4. Диаграмма Вороного

Пусть на плоскости зафиксировано конечное множество несовпадающих точек P . Организовать разбиение плоскости на замкнутые области (ячейки), каждая из которых содержит по одной точке из P , причем в любой из полученных областей расстояние от любой ее позиции Q до своей точки из P не должно превосходить расстояния от Q до любой другой точки из P . Подобное “районирование” плоскости на ячейки впервые предложено российским ученым Г. Ф. Вороным [35]. В его честь подобное разбиение плоскости теперь называют диаграммой или мозаикой Вороного. Приведем конкретный пример. Человеку потребовалось некоторое лекарство и у него есть карта города, на которой точками обозначены все аптеки. Он может расчертить карту на ячейки так, чтобы внутри каждой ячейки находилось только одна аптека, а для всех остальных точек ячейки именно эта аптека и была бы ближайшей. Полученная картинка и будет диаграммой Вороного для аптек города. Чтобы понять, к какой аптеке двигаться, человеку остается лишь определить на карте свое текущее местоположение.

A. Voronoi[*lis*]

Построение диаграммы Вороного

А) Командой *A*, где *lis* – список различных исходных точек плоскости, формируется и возвращается диаграмма Вороного этих точек.

Пример 4. Введем те же точки P , что и в (4). Команда, по которой находится выпуклая оболочка этих точек, может быть записана, например, так:

$$gr=Voronoi[\{A,B,C,D,E,F,G,H,I,J\}].$$

При ее вводе на панелях “Объекты” и “Полотно” получим изображение, представленное на рис. 5.

Все выведенные точки являются независимыми. Перемещая некоторые из них, мы будем получать динамическую картинку – другие задачи для нахождения диаграмм Вороного и в реальном времени – ее построение.

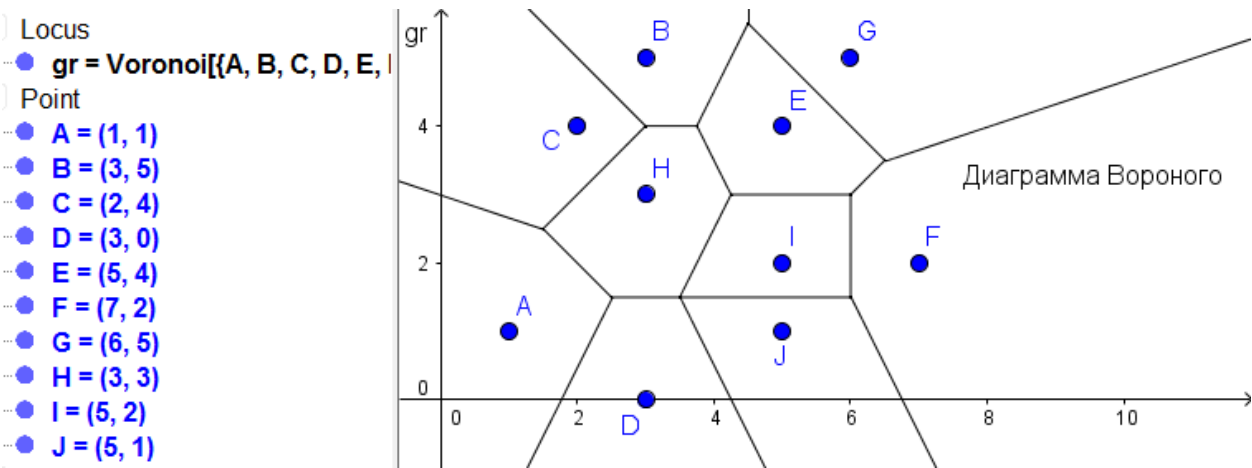


Рис. 5. Построение диаграммы Вороного

8.5. Триангуляция Делоне

Пусть на плоскости зафиксировано конечное множество несовпадающих точек P . Пусть далее некоторые пары точек P соединены отрезками линий так, что все точки оказались вершинами треугольников, которые могут пересекаться друг с другом только по сторонам. В этом случае говорят, что проведена триангуляция по точкам P . В общем случае триангуляций может быть много. Оказывается, что если никакие четыре точки P не лежат на окружности, описанной около любого из треугольников, то триангуляция единственна. Впервые такое разбиение плоскости описано в 1934 г. советским математиком Б. Делоне и в его честь теперь оно называется триангуляцией Делоне [35,36]. Рассмотренные в предыдущем пункте диаграммы Вороного имеют тесную связь с триангуляцией Делоне. Если в диаграмме Вороного точки P для всех соседних областей соединить отрезками, то получим неориентированный граф, который и будет являться триангуляцией Делоне.

A. DelaunayTriangulation[lis]

Проведение триангуляции Делоне


А) Командой A , где lis – список различных исходных точек плоскости, формируется и возвращается триангуляция Делоне для этих точек.

Пример 5. Введем точки P :

```
Execute[{"A=(1,1)", "B=(3,5)", "C=(2,4)", "D=(3,0)", "E=(5,4)",  
        "F=(7,2)", "G=(6,5)", "H=(3,3)", "I=(5,2)", "J=(6,1)"}].
```

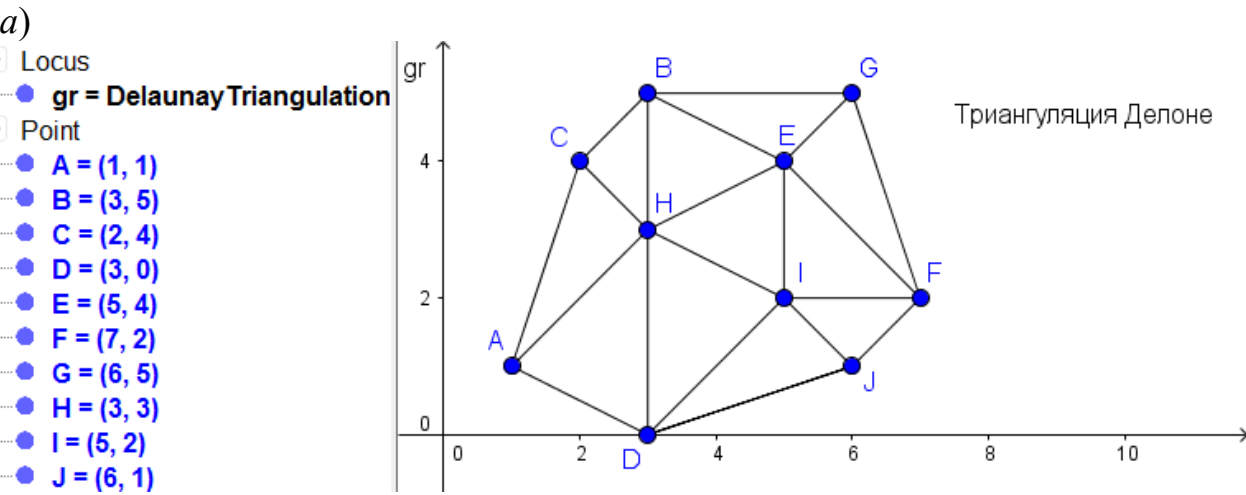
Команда, по которой находится триангуляция Делоне, может быть записана, например, так:

```
gr=DelaunayTriangulation[{A,B,C,D,E,F,G,H,I,J}].
```

При ее вводе на панели “Полотно” получим изображение, представленное на рис. 6а. Изображение на рис. 6б повторяет триангуляцию Делоне для того же самого множества точек P с дополнительным выводом описанных окружностей для некоторых из треугольников. Эти окружности можно провести, например, с помощью инструмента “ Окружность по трем точкам”.

Все выведенные точки являются независимыми. Перемещая некоторые из них, мы будем получать динамическую картинку – другие задачи для нахождения триангуляции Делоне и в реальном времени – ее построение.

Триангуляция Делоне на плоскости обладает рядом интересных свойств. В частности, она имеет минимальную сумму радиусов окружностей, описанных около треугольников, среди всех возможных триангуляций [36].



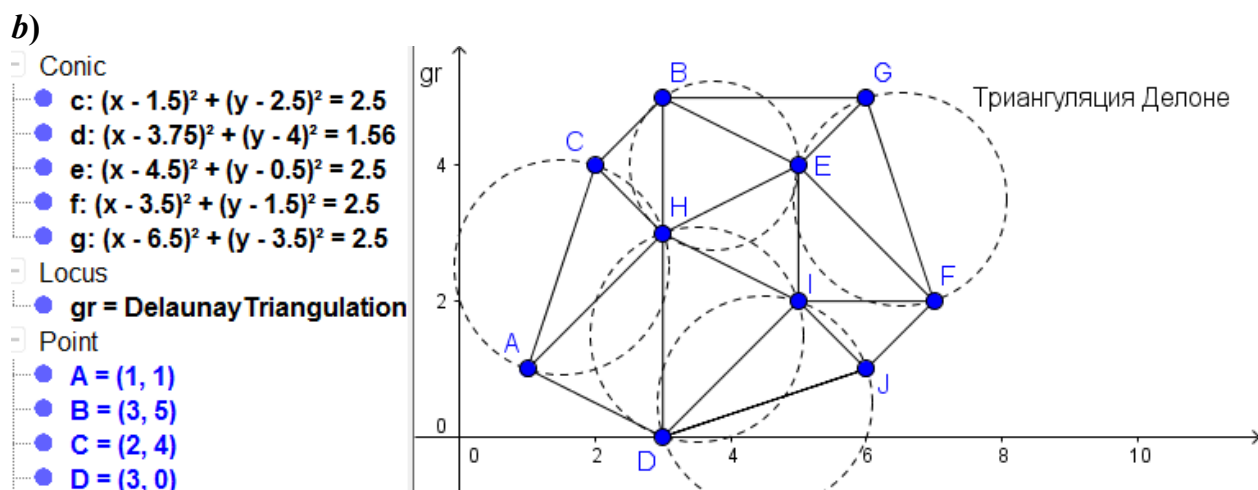


Рис. 6 (a, b). Триангуляция Делоне

8.6. Нахождение минимального остовного дерева

Пусть задан полный неориентированный граф G и на его ребрах определена функция евклидова расстояния между вершинами. Остовным деревом графа G называют такой его подграф, в который входят все вершины G и нет циклов, то есть двигаясь от любой вершины U по ребрам нельзя снова попасть в U , не проходя по какому-либо ребру дважды. Остовные деревья иногда называют покрывающими деревьями, остовами или скелетами графа. Минимальным остовным деревом называют остовное дерево с минимальной суммарной длиной ребер.

A. *MinimumSpanningTree*[lis]

Нахождение минимального остовного дерева

A) Пусть *lis* – список вершин полного неориентированного графа, на ребрах которого определена функция евклидова расстояния между соединяемыми ими вершинами. Командой A находится и возвращается минимальное остовное дерево исходного графа.

Пример 6. Введем вершины графа:

Execute["A=(1,1)", "B=(3,5)", "C=(2,4)", "D=(3,0)", "E=(5,4)",
"F=(7,2)", "G=(6,5)", "H=(3,3)", "I=(3,2)", "J=(6,1)"].

Команда, по которой находится минимальное остовное дерево, может быть записана, например, так:

gr=MinimumSpanningTree[{A,B,C,D,E,F,G,H,I,J}].

При ее вводе на панели “Полотно” получим изображение с вершинами полного графа G и минимальным остовным деревом (см. рис. 7a). Сам граф G не нарисован. Изображение на рис. 7b повторяет остовное дерево рис. 7a, но на нем дополнительно показана также и триангуляция Делоне. Именно на ее ребрах и размещается остовное дерево. И этот факт не случаен. Минимальное (евклидово) остовное дерево всегда гарантированно располагается на триангуляции Делоне.

Все выведенные точки являются независимыми. Перемещая некоторые из них, мы получаем динамическую картинку – другие задачи для нахождения минимального остовного дерева и в реальном времени – построение этого дерева.

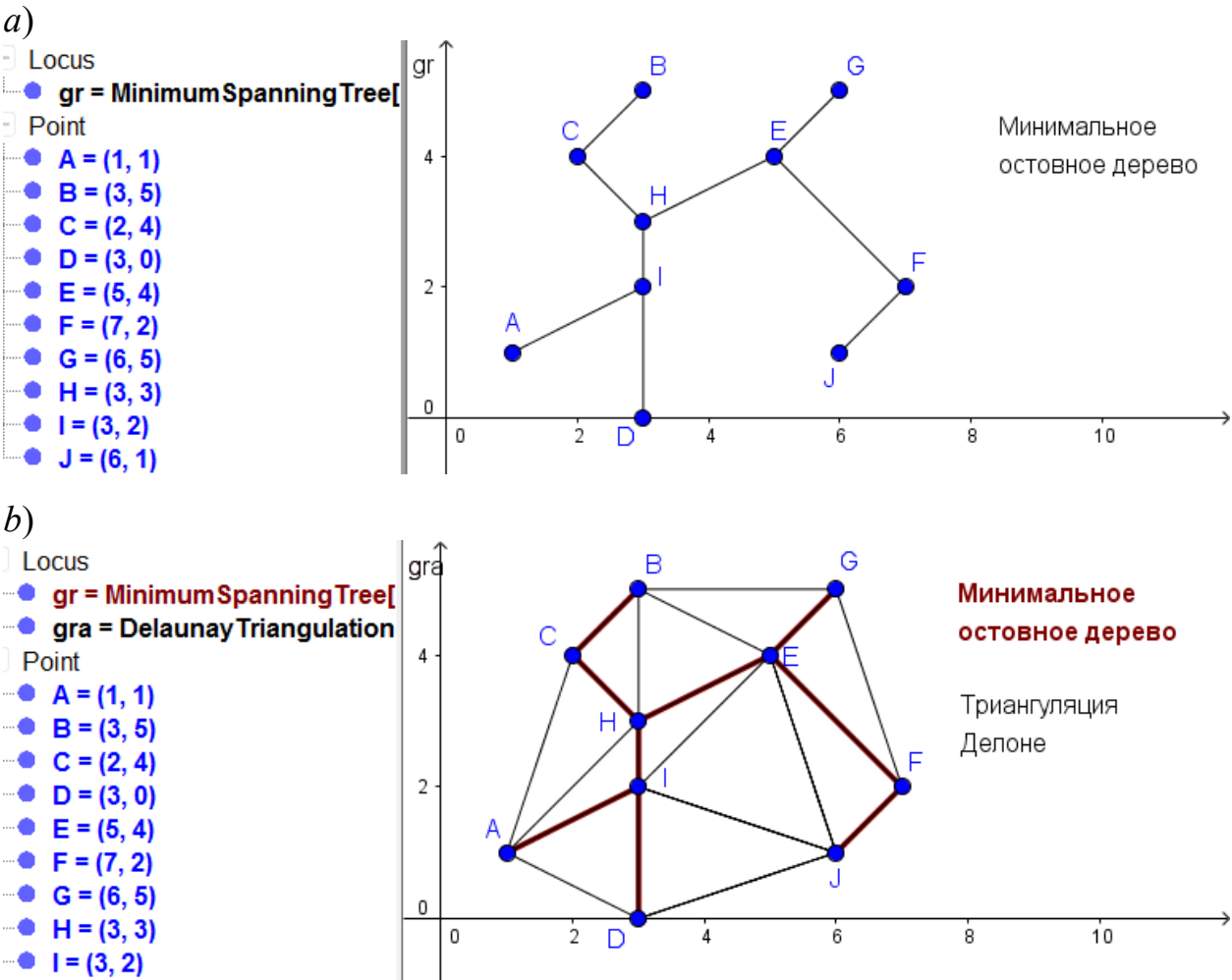





Рис. 7 (a, b). Минимальное остовное дерево и триангуляция Делоне

9. Создание новых инструментов


Если среди предлагаемых системой *GeoGebra* инструментов отсутствует тот, который хотелось бы иметь, то ничто не препятствует пользователю создать такой инструмент. Все средства для этого в системе имеются. Можно сказать, что пользователю предоставлена простая и удобная виртуальная мастерская по производству новых инструментов. Причем доступ к таким инструментам и работа с ними проводятся так же, как и с системными инструментами – через имеющееся пиктографическое меню [52]. По умолчанию новые инструменты встраиваются в это меню с предлагаемыми системой уникальными именами, с одинаковыми пиктограммами (иконками, значками), группируются в отдельном раскрывающемся подменю и сохраняются только в текущей сессии. Однако есть возможность изменить и имя, и пиктограмму инструмента по своему усмотрению, а также сохранить инструмент для работы в последующих сессиях. При этом размещать пользовательские инструменты можно в одном новом подменю, в нескольких новых подменю и даже в имеющихся системных подменю. Пиктограммы задаются в формате графических *png*-файлов, *jpg*-файлов или *bmp*-файлов, а инструменты сохраняются в виде *ggt*-файлов. Вместе с каждым инструментом создается и соответствующая ему команда. Инструментом мы работаем на панели “Полотно” в ручном режиме, а командой, вводимой через строку ввода, эта же работа выполняется автоматически. Имена инструментов и соответствующих им команд могут быть и одинаковыми, и разными. Поясним формирование новых инструментов на трех конкретных примерах.

9.1. Инструмент “Треугольник и углы”

Попробуем создать инструмент для построения треугольников с помеченными углами. Сформировать просто треугольник системными средствами можно двумя путями. Первый из них – поместить на панель “Полотно” три точки (инструмент “ Точка”) и соединить их отрезками линий (инструмент “ Отрезок”), второй – поочередно указать три позиции и повторно щелкнуть по первой из них (инструмент “ Многоугольник”). Пусть через *Триа* обозначен инструмент для формирования треугольников с помеченными и именованными углами. Приступим к его созданию (пункты 1-6).

1. Решение конкретной задачи:

- потроим $\triangle ABC$ одним из указанных выше способов, например, первым;

- инструментом “ Угол” последовательно отметим в треугольнике три угла α , β и γ , указывая для каждого из них по три точки ($B, A, C \rightarrow \alpha, \dots$);
- изменим свойства объектов α , β и γ . Для этого выделим их на панели “Объекты” и через контекстное меню ($RClick \backslash Свойства \backslash \dots$) откроем панель “Настройки”. На странице “Основные” в независимом переключателе “Показывать обозначение” укажем “Имя”. Это обеспечит показ имен углов, то есть α , β и γ без указания их текущих значений. Далее, на странице “Стиль” увеличим расстояние от вершин, на котором проводятся дуги углов, изменив значение ползунка “Размер” с 30 на 40.

2. Создание инструмента *Tria*. Предварительная работа завершена и можно приступать к созданию инструмента *Tria*:

- в меню “Инструменты” фиксируем выбор “Создать инструмент” и в результате открывается одноименная панель с тремя страницами, имеющими закладки: “Выходные объекты”, “Входные объекты” и “Имя и значок” (см. рис. 1а). На первых двух страницах создаются описания объектов и делается это щелчками левой кнопки мыши по существующим объектам панелей “Объекты” или “Полотно”. Если на странице оказывается много описаний объектов, то навигация по ним реализуется кнопками “▲” и “▼”. Если описание попало на страницу по ошибке, то его можно удалить кнопкой “X”. Перемещение по страницам реализуется щелчками по их закладкам или кнопками “Назад” и “Следующий”;
- на странице “Выходные объекты” необходимо указать треугольник, то есть три его стороны и три угла. И если их там еще нет, то требуется лишь щелкнуть кнопкой мыши по соответствующим объектам на панели “Полотно” или по строкам их описания на панели “Объекты” (см. рис. 1а);

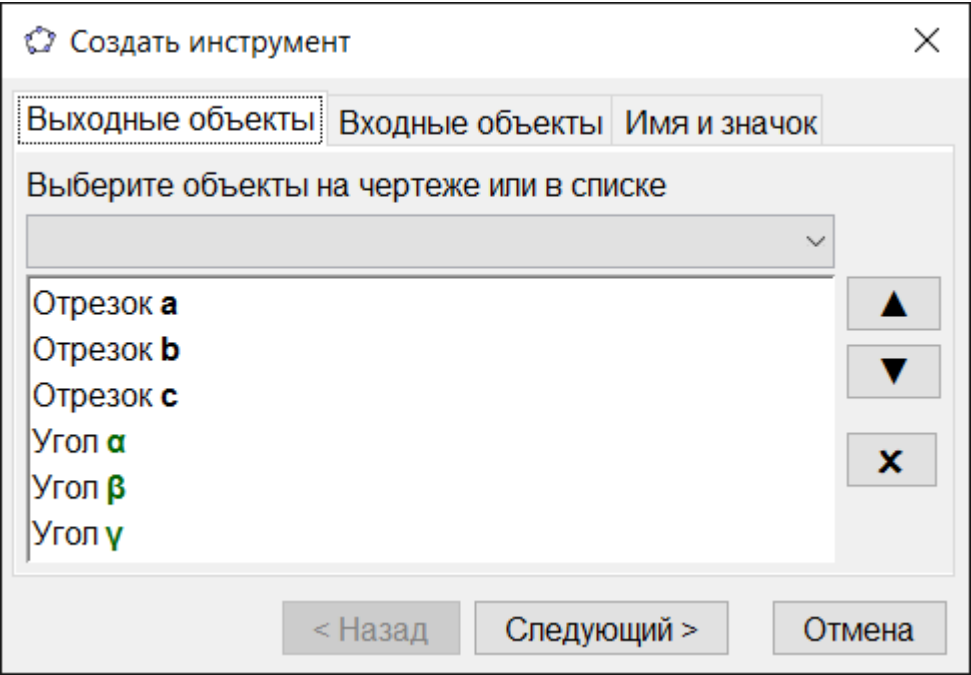


Рис. 1а. Страница “Выходные обхекты” панели создания нового инструмента

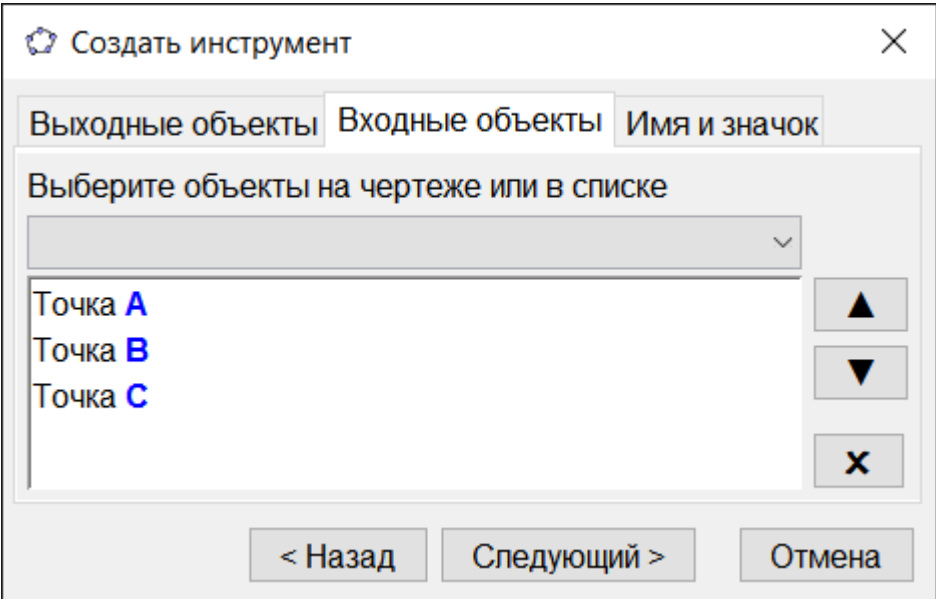

- аналогичным образом на странице “Входные объекты” необходимо указать три точки A , B и C (см. рис. 1b);
- активируем страницу “Имя и значок” (см. рис. 1c). Зададим одинаковые имена $Tria$ для инструмента и соответствующей ему команды, хотя разрешается им быть и разными. В дальнейшем действие инструмента можно будет осуществлять непосредственно, а также с помощью команд вида $Tria(A, B, C)$. Если задать описание, то оно будет текстом всплывающей подсказки для инструмента $Tria$, появляющейся при наведении курсора на его пиктограмму. По умолчанию пиктограмма представляется в виде “

Рис. 1b. Страница “Входные объекты” панели создания нового инструмента

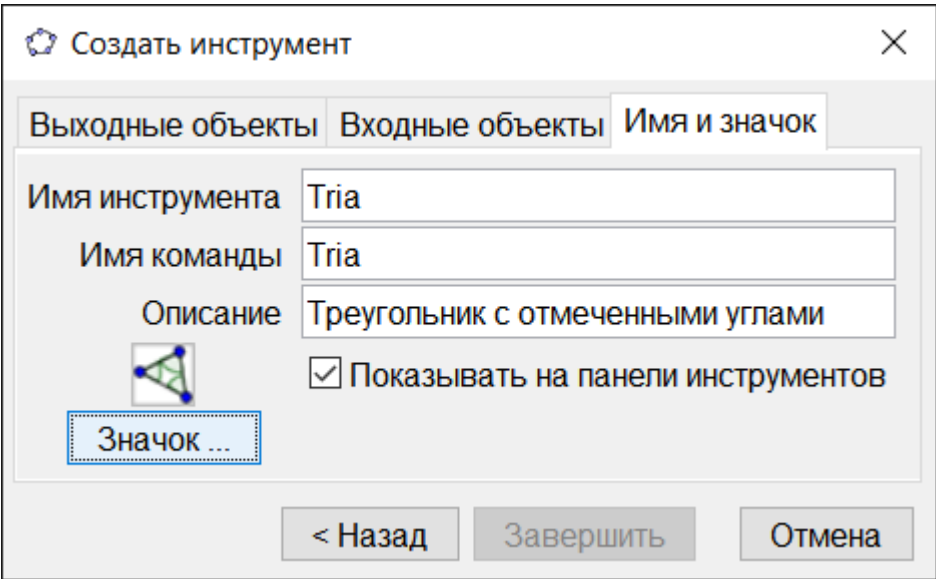




Рис. 1c. Страница “Имя и значок” панели создания нового инструмента

3. Работа с инструментом *Tria* в текущей сессии. Инструмент *Tria* создан и им уже можно пользоваться в любых новых документах текущей сессии. Если через меню открыть новый документ (*Файл/Создать/...*), то в нем инструмент *Tria* и команда $Tria(A,B,C)$ оказываются доступными, причем даже в том случае, когда мы не сохранили *ggb*-файл документа с созданным инструментом *Tria*. Для эксперимента решим такие контрольные задачи:

Z1. Инструментом “ Треугольник с отмеченными углами” сформируем треугольник, у которого отмечены внутренние углы. Для решения задачи при создании точек следует щелкать по позициям панели “Полотно” в направлении, обратном ходу часовой стрелки (см. рис. 2a).

Z2. Инструментом “ Треугольник с отмеченными углами” сформируем треугольник, у которого отмечены внешние углы. Для решения задачи при создании точек следует щелкать по позициям панели “Полотно” в направлении по ходу часовой стрелки (см. рис. 2b).

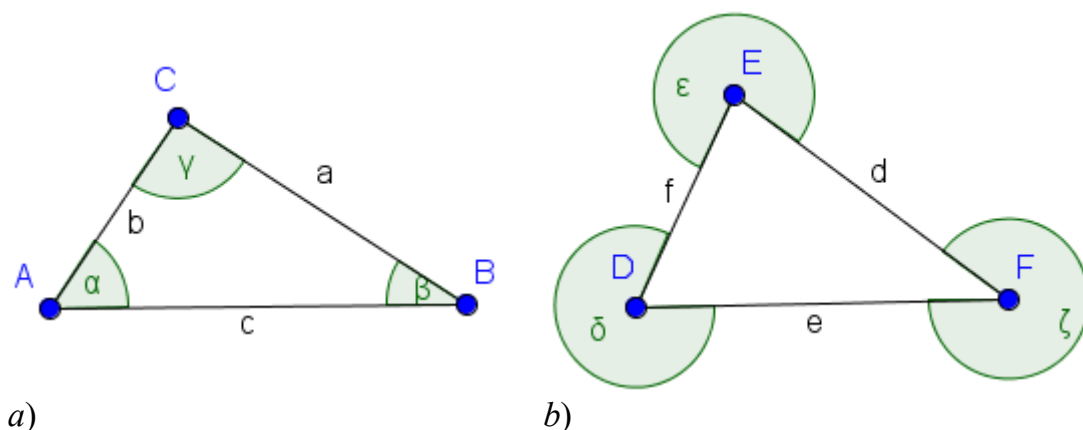


Рис. 2. Варианты действия инструмента *Tria* и команды $Tria[...]$

Замечание. Если вместо инструмента *Tria* используется команда $Tria[...]$, то, во-первых, точки на полотне уже должны быть заданы, а, во-вторых, вне зависимости от направления их вывода можно получить любой из треугольников рис. 2. Если сформирован не тот треугольник, то его следует удалить и заново вывести командой $Tria[...]$, изменив в ней порядок аргументов на обратный.

Инструмент и команду *Tria* можно использовать и в открываемых новых документах. Но, если текущую сессию *GeoGebra* закрыть, то в новой сессии *Tria* станет недоступным. Сохранить эту доступность для последующих сессий можно, выполнив через меню команду-установку “*Настройки/Сохранить настройки*”. Впрочем, также просто от нее и освободиться установкой “*Настройки/Сбросить настройки*”. Но действовать она начнет лишь со следующей сессии.

4. *Сохранение инструмента Tria и команды Tria(...)*. Как мы уже отмечали, для работы с *Tria* в последующих сессиях достаточно было выполнить команду “*Настройки\Сохранить настройки*”. Чтобы после выполнения команды “*Настройки\Сбросить настройки*” инструмент не оказывался утерянным, его следует сохранить под каким-нибудь именем в виде *ggt*-файла. Сделать это можно из любого *ggb*-документа с данным инструментом командой меню “*Инструменты\Управление инструментами\Выбор инструмента (▲,▼)\Сохранить как\Имя для инструмента*”. Имя файла может быть, например, таким – *Tria.ggt*.

5. *Встраивание пользовательских инструментов в пиктографическое меню*. Если сохранены *ggt*-файлы, то встроить их инструменты в пиктографическое меню открытого пустого документа *GeoGebra* можно так. Через меню открываем *ggt*-файлы или последовательными командами “*Файл\Открыть\...*”, или одной такой командой. В последнем случае все требуемые *ggt*-файлы предварительно должны быть выделены в “*Проводнике*” с помощью *Ctrl*- и (или) *Shift*-технологии. После таких действий соответствующие пользовательские инструменты окажутся собранными в единое новое пиктографическое подменю. Кроме того, открыть документ *Geogebra* с конкретным пользовательским инструментом можно непосредственно из *Проводника* щелчком по *ggt*-файлу.



6. *Обустройство пиктографического подменю*. Командой меню “*Инструменты\Настройка*” открывается панель “*Настройка*”, на которой можно проводить различные манипуляции с любым подменю текущего пиктографического меню и его выборами (компонентами, элементами). Сказанное касается и пользовательского подменю. Возможны такие действия:


- какие-бы операции с текущим пиктографическим меню мы не совершали, всегда есть возможность возврата к тому виду меню, которое принято по умолчанию. И этот возврат реализуется щелчком по кнопке “*Восстановить по умолчанию*”;
- выделив подменю, можно перемещать его относительно других подменю вверх и вниз (▲,▼);
- раскрыв список выборов конкретного подменю и выделив выбор, можно перемещать его вверх и вниз относительно других выборов, но в пределах данного подменю (▲,▼);
- для удаления конкретного подменю или конкретного выбора подменю его следует переместить с левого окна панели в правое окно. Для этого следует выделить подменю или выбор подменю, а затем щелкнуть по кнопкам “*Удалить>*” и “*Применить*”;
- выборы можно перемещать из одного подменю в другое подменю. Для этого перемещаемый выбор необходимо перекинуть в правое окно и там его выделить. Далее в левом окне необходимо раскрыть требуемое подменю и выделить в нем выбор, за которым должна быть организована вставка. Для завершения операции остается щелкнуть по кнопке “*<Вставить*”. Заметим,





что таким способом можно перестраивать системные подменю, размещать в системных подменю пользовательские инструменты, а также создавать несколько пользовательских подменю;

- в правом окне всегда присутствует специальный выбор меню, называемый разделителем (---*Разделитель*). Он служит для разбиения выборов подменю на отдельные группы и представляется там реальным разделяющим отрезком. С подобными выборами на панели “*Настройка*” можно проводить те же самые действия, что и с обычными выборами.


9.2. Инструмент “Окружность и центр”

Системным инструментом “ Окружность по трем точкам” строятся окружности по трем точкам. Сформируем аналог этого инструмента для построения окружностей и их центров по трем точкам. Иконку для пиктограммного меню в данном случае можно взять в виде “”. Итак, создаем инструмент *circleO*.



1. Решим поставленную задачу с помощью инструмента “”:

- в новом документе инструментом “ Окружность по трем точкам” строим окружность *c* по точкам *A*, *B* и *C*;
- инструментом “ Отрезок” соединяем отрезками пары точек *A*, *B* и *A*, *C*;
- инструментом “ Срединный перпендикуляр” проводим к серединам отрезков *AB* и *AC* перпендикуляры;
- инструментом “ Пересечение” устанавливаем точку *D* в позицию пересечения срединных перпендикуляров.

2. Создание инструмента *circleO*. В меню “*Инструменты*” зафиксируем выбор “*Создать инструмент*”. В результате открывается одноименная панель с тремя страницами, имеющими закладки: “*Выходные объекты*”, “*Входные объекты*” и “*Имя и значок*”. Выполним такие действия:


- на странице “*Выходные объекты*” укажем окружность *c* и точку *D*;
- на странице “*Входные объекты*” укажем точки *A*, *B* и *C*;
- активируем страницу “*Имя и значок*” и зададим одинаковые имена инструмента и команды – *circleO*. В дальнейшем действие инструмента можно будет осуществлять непосредственно, а также с помощью команд вида *circleO(A,B,C)*. Для всплывающей подсказки сформируем дописание “*Окружность и центр по трем точкам*” и укажем для иконки пиктографического меню заранее созданный файл с рисунком “”.

3. *Сохранение нового инструмента*. Созданный инструмент сохраним под именем *circleO.ggt*. Для этого через меню и панели “*Управление инструментами*” и “*Сохранить*” выполним команду “*Инструменты\Управление*


инструментами\Выбор инструмента (, )\Сохранить как\...\circleO.ggt". Инструмент *circleO* и команду *circle[A,B,C]* можно использовать в текущей сессии и при необходимости через меню открывать в новых сессиях (Файл\Открыть\...);

4. Если инструмент *circleO* и команду *circle[A,B,C]* требуется сделать доступными для последующих сессий без открытия ggt-файла, то дополнительно следует выполнить команду “Настройки\Сохранить настройки”.


9.3. Инструмент “Деление отрезка на равные части”

Сформируем новый инструмент *segmentN*, по которому отрезок *AB* делится на *n* равных частей, где *AB* задан точками *A* и *B*, а число *n* вводится с клавиатуры. Иконку для пиктограммного меню в данном случае можно взять в виде . Итак, создаем инструмент *segmentN*.

1. Решим поставленную задачу для $n=2$, но так, чтобы она аналогично решалась и при $n=3, 4, \dots$:

- инструментом  “Отрезок” создаем отрезок *AB* (*a*);
- вводим через строку ввода число $n=2$;
- вводим через строку ввода выражение $Sequence[(A \cdot (n-k) + k \cdot B)/n, k, 1, n-1]$. Фактически создается список из $n-1$ точки, лежащей между *A* и *B*, причем расстояния между любыми смежными точками на *AB*, включая *A* и *B*, являются равными (список1). Теперь отрезок *AB* точками разделен на *n* равных частей, где $n=2$;

2. Создание инструмента *segmentN*. В меню “Инструменты” зафиксируем выбор “Создать инструмент” и выполним такие действия:

- на странице “Выходные объекты” укажем список “список1” и отрезок *a*;
- на странице “Входные объекты” укажем точки *A*, *B* и число *n*;
- активируем страницу “Имя и значок” и на ней зададим одинаковые имена *segmentN* для инструмента и команды, сформируем описание “Деление отрезка на *n* равных частей” для всплывающей подсказки и укажем заранее созданный файл с рисунком  для иконки пиктографического меню.

3. Сохранение нового инструмента. Созданный инструмент сохраним под именем *segmentN.ggt*. Для этого через меню и панели “Управление инструментами” и “Сохранить” выполним команду “Инструменты\Управление инструментами\Выбор инструмента\Сохранить как\...\segmentN.ggt”. Инструмент *segmentN* и команду *segment[A,B,n]* можно использовать в текущей сессии и при необходимости открывать в новых сессиях.

4. Если инструмент *segmentN* и команду *segment[A,B,n]* требуется сохранить доступными для последующих сессий без дополнительного открытия

ggt-файла, то дополнительно следует выполнить команду “*Настройки\Сохранить настройки*”.

9.4. Задачи для создания новых инструментов

Приведенные примеры создания новых инструментов показывают, что формировать их несложно, и демонстрируют гибкость и универсальность системы *GeoGebra*. Но как бы просто не создавались пользовательские инструменты, без определенного тренинга на этот счет обойтись вряд ли удастся. И здесь можно было бы предложить задачи на формирование инструментов двух разных типов. Если нашей целью является создание широко используемых инструментов, то по ним должны выводиться отдельные небольшие тщательно продуманные фрагменты изображений, часто встречающиеся в различных динамических чертежах. Если же речь идет об обучении созданию инструментов, то по ним должны выводиться достаточно сложные фрагменты изображений. Это позволит быстро научиться правильно вычленять для инструментов входные и выходные объекты. На задачах второго типа мы и сосредоточимся. А перед их формулированием напомним названия некоторых замечательных точек в треугольнике.

Центроид – центр тяжести, центр масс, точка пересечения медиан.

Ортоцентр – центр пересечения высот.

Центр описанной окружности – точка пересечения “серединных” перпендикуляров, то есть перпендикуляров к серединам сторон треугольника.


Инцентр – центр вписанной в треугольник окружности, точка пересечения биссектрис.

Точки Фейербаха – середины отрезков высот, соединяющих ортоцентр с вершинами треугольника.

Центр вневписанной окружности. Вневписанная окружность – это окружность, касающаяся одной из сторон треугольника и продолжений двух других его сторон. Центром любой из трех вневписанных окружностей является точка пересечения биссектрисы внутреннего угла и двух биссектрис внешних углов треугольника, не смежных с этим внутренним углом.

9.4.1. Теорема о “бабочке”

Пусть имеется окружность с центром в точке A , BC – хорда окружности и D – середина хорды (см. рис. 3). Прикрепим к окружности точки E и F и проведем две другие хорды EH и FG , проходящие через точку D . Если через I и J обозначить точки пересечения соответственно хорд EG и FH с BC , то точка D окажется серединой отрезка IJ , то есть $ID=DJ$. Это и есть теорема о бабочке.

Создать инструмент *Butterfly*, по которому выводится иллюстрация к теореме о бабочке в виде чертежа и текста, показанная на рис. 3. Точки B , C , E и F должны быть независимыми. Каждая из трех строк текста должна формироваться инструментом “ Текст” и состоять из постоянной и переменной части – некоторого выражения и его текущего значения. Задаются эти строки через автоматически раскрываемую панель “Текст”, где переменные части необходимо размещать в пустых боксах, вводимых из списка “Объекты”:

$$ID = \boxed{ID}, DJ = \boxed{DJ}, ID = DJ \rightarrow \boxed{ID == DJ}.$$

При таком задании текстов первые две строки будут демонстрировать текущее значение отрезков ID и DJ , а третья строка – показывать, равны эти отрезки (*true*) или не равны (*false*).

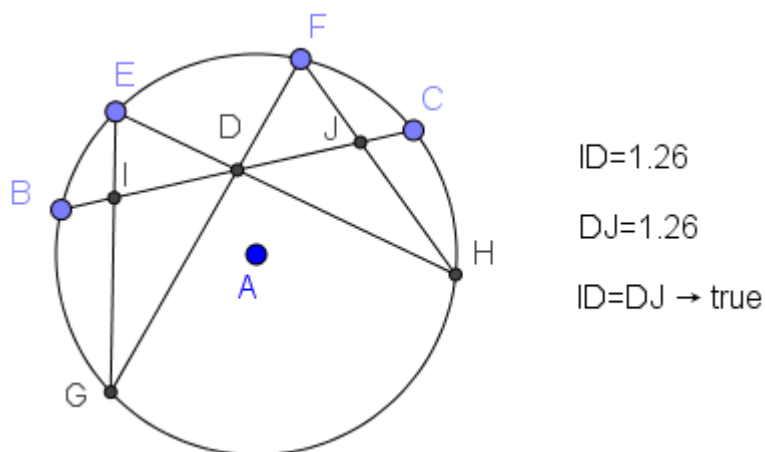



Рис. 3. Иллюстрация к задаче “Теорема о бабочке”
(инструмент *Butterfly* и команда *Butterfly[A]*)

Замечания. 1. При создании инструмента *Butterfly* потребуется указать такие данные:

- выходные параметры:
 - окружность;
 - точки: $B, C, D, E, F, G, H, I, J$;
 - отрезки: BC, EH, FG, EG, FH ;
- входные параметры – точка A (если окружность формировалась инструментом “ Окружность по центру и радиусу”).

В изображении, получаемом инструментом *Butterfly*, точки B, C, E и F являются независимыми. Можно убедиться, что надписи к рисунку ведут себя в полном соответствии с теоремой о бабочке как при изменении позиции начальной хорды BC (перемещение концевых точек B и C хорды по окружности), так и при изменении позиций точек E и F , а вместе с ними и отрезка IJ (перемещение точек E и F по окружности, не переходя через точки B и C).

2. Считают, что теорему о бабочке сформулировал и доказал математик У. Дж. Горнер, ставший широко известным благодаря предложенному им алгоритму вычисления многочленов от одной переменной в конкретной точке, который известен сегодня как “схема Горнера”. Задача о бабочке впервые появилась в журнале “*Gentleman’s Dairy*” (Англия, 1815 г.), и на сегодняшний день известно более десятка ее доказательств. Весьма простое из них получается, если в треугольниках IED , DFJ , GID и HDJ из вершин I и J провести перпендикуляры к противоположным сторонам и рассмотреть четыре пары подобных треугольников [21, с. 59-60].

9.4.2. Теорема Эйлера

Создать инструмент Eu_inout , по которому выводятся треугольник, его вписанная и описанная окружности и отрезок между их центрами (см. рис. 4). Используйте инструмент Eu_inout для экспериментальной проверки теоремы Эйлера, утверждающей, что $d^2 = R(R - 2r)$ (или $\frac{1}{R-d} + \frac{1}{R+d} = \frac{1}{r}$), где R и r – радиусы вписанной и описанной окружностей, d – расстояние между их центрами [23, с.148; 71]. Проверку равенства $d^2 = R(R - 2r)$ можно осуществлять выводом текста с постоянной и переменной частями (см. А). Например, основная часть текста может быть задана так:

$$d^2 = R(R - 2r) \rightarrow \boxed{GK \wedge 2 == KC * (KC - 2 * GH)}.$$

В этом случае выражение из прямоугольника будет возвращать логическое значение – *true*, если $d^2 = R(R - 2r)$, и логическое значение *false*, если $d^2 \neq R(R - 2r)$.

На рис. 4 показан вывод, который должен осуществляться инструментом Eu_inout или командой $Eu_inout[A,B,C]$, где $r=GH$, $R=KC$, $d=GK$.

Замечания. 1. Будем считать, что при создании инструмента Eu_inout треугольник ABC формируется “вручную”, то есть заданием трех точек с последующим соединением их отрезками прямых линий. В этом случае потребуется указать такие данные:

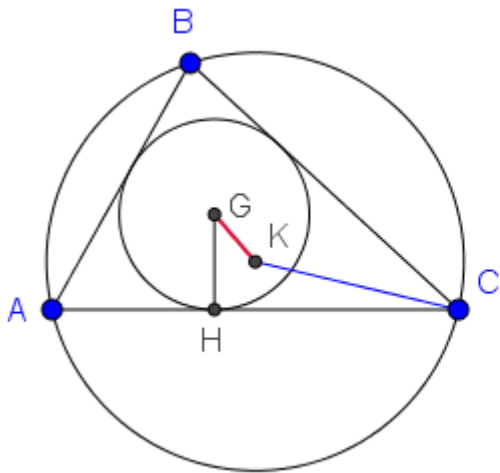
- выходные параметры:
 - отрезки: AB , BC , CA , $GH(r)$, $KC(R)$, $GK(d)$;
 - окружности: вписанная и описанная;
 - точки: G , H , K ;
 - тексты: 4 надписи (см. рис. 4);
- входные параметры – точки A , B и C .

В изображении, получаемом инструментом Eu_inout , точки A , B и C являются независимыми. Протаскивая их по полотну, мы будем менять вели-

чины r , R и d . При этом можно видеть, что во второй и третьей строках текста значения всегда будут совпадать, а потому в четвертой строке выводится одно и то же логическое значение *true*, что экспериментально подтверждает справедливость формулы Эйлера.

2. Из теоремы Эйлера следует неравенство Эйлера: $R \geq 2r$. Имеют место и такие неравенства [59, с. 197-209]:

$$\frac{R}{r} \geq \frac{a^3 + b^3 + c^3 + abc}{2abc} \geq \frac{a}{b} + \frac{b}{c} + \frac{c}{a} - 1 \geq \frac{2}{3} \left(\frac{a}{b} + \frac{b}{c} + \frac{c}{a} \right) \geq 2.$$



$$\begin{aligned} r &= 0.95 \quad R = 2.09 \quad d = 0.62 \\ d^2 &= 0.39 \\ R(R - 2r) &= 0.39 \\ d^2 &= R(R - 2r) \rightarrow true \end{aligned}$$

Рис. 4. Иллюстрация к задаче “Теорема Эйлера”
(инструмент *Eu_inout* и команда *Eu_inout[A,B,C]*)

9.4.3. Аналог теоремы Эйлера

Для треугольника с описанной и одной из внеписанных окружностей справедлив аналог теоремы Эйлера (см. предыдущий пункт), который можно записать так: $d^2 = R(R + 2r_e)$, где R и r_e – радиусы описанной и внеписанной окружностей, d – расстояние между их центрами [71]. Создайте инструмент *Eu_outexc*, по которому выводятся треугольник с описанной и одной из внеписанных окружностей. Используйте его для экспериментальной проверки справедливости приведенной теоремы.

Требуемый инструмент *Eu_outexc* может быть построен по аналогии с ранее созданным инструментом *Eu_inout*. На рис. 5 показан вывод, который должен осуществляться инструментом *Eu_outexc* или командой *Eu_outexc[A,B,C]*, где $R=FC$, $r_e=GH$, $d=FG$.

Замечание. Будем считать, что при создании инструмента *Eu_outexc* треугольник ABC , как и в предыдущей задаче, формируется “вручную”, то есть заданием трех точек с последующим соединением их отрезками прямых линий. В этом случае потребуется указать такие данные:

- выходные параметры:
 - отрезки: AB , BC , CA , FC (R), GH (r_e), FG (d);

- окружности: описанная и внеписанная;
- точки: F , G , H ;
- тексты: 4 надписи (см. рис. 5);
- входные параметры – точка A , B и C .

В изображении, получаемом инструментом *Eu_outexc*, точки A , B и C являются независимыми. Протаскивая их по полотну, мы будем менять величины r , R и d . При этом выводимые надписи к рисунку экспериментально подтверждают справедливость аналога формулы Эйлера.

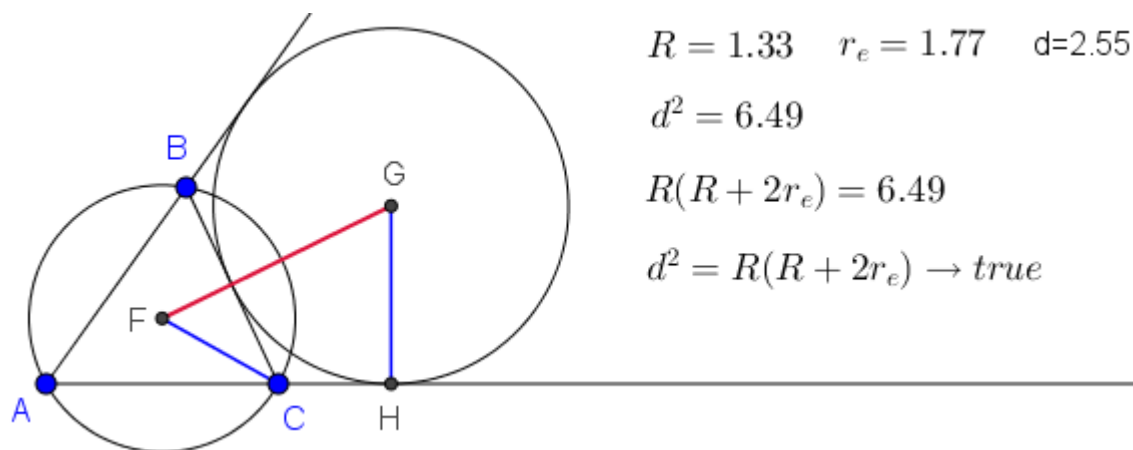


Рис. 5. Иллюстрация к задаче “Аналог теоремы Эйлера”
(инструмент *Eu_outexc* и команда *Eu_outexc[A,B,C]*)

9.4.4. Интересное соотношение

Пусть в треугольнике ABC радиус описанной окружности равен R . Экспериментально подтвердите, что сумма квадратов расстояний от центра описанной окружности до центров вписанной и трех внеписанных окружностей равна $12R^2$ [23, с. 88]. Для этого создайте инструмент *SumSquare* для вывода соответствующего рисунка и тестирующих надписей, наподобие тех, которые использовались в предыдущих задачах. На рис. 6 показан вывод, который должен осуществляться инструментом *SumSquare* или командой *SumSquare[A,B,C]*.

9.4.5. Теорема Мансиона

Экспериментально подтвердите справедливость теоремы П. Мансиона, утверждающей, что если в треугольнике соединить центр вписанной окружности с центрами внеписанных окружностей, то середины полученных отрезков будут лежать на описанной окружности [23, с. 81]. Для этого создайте инструмент *Middles* для вывода соответствующего рисунка и тестирующих надписей, наподобие тех, которые использовались в предыдущих задачах. На рис. 7 показан вывод, который должен осуществляться инструментом *Middles* или командой *Middles[A,B,C]*.

Замечание. Теорема Мансиона имеет еще и другую формулировку. Середина дуги AB (см. рис. 7), не содержащая C , равноудалена от вершин A и B треугольника, центра D вписанной окружности и центра F внеписанной окружности. Середина дуги AB , содержащая C , равноудалена от вершин A и B треугольника и от центров G и H внеписанных окружностей.

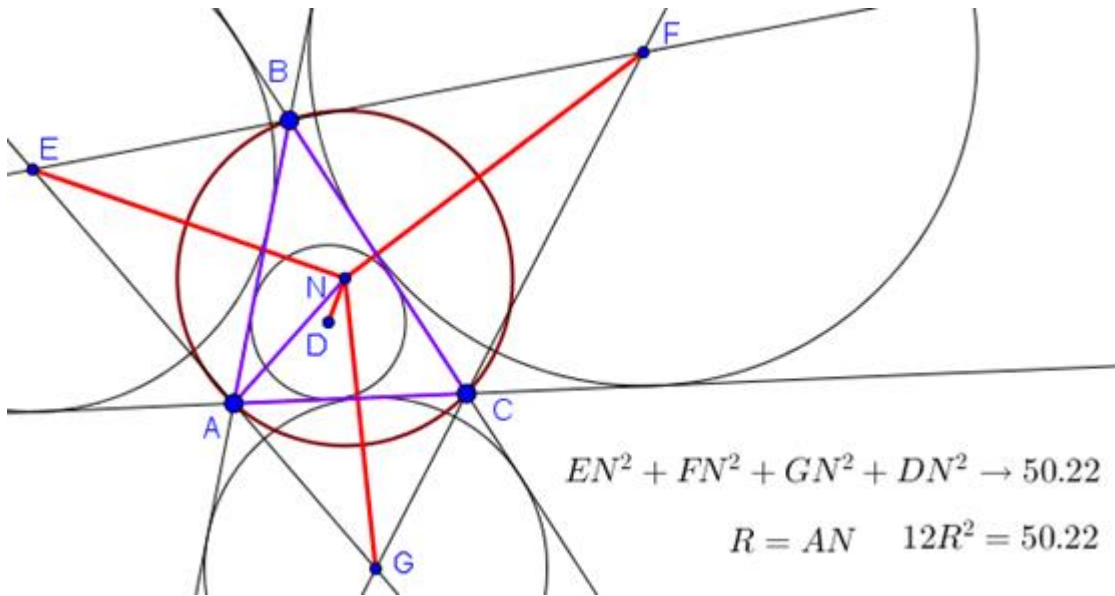


Рис. 6. Иллюстрация к задаче “Интересное соотношение”
(инструмент *SumSquare* и команда *SumSquare[A,B,C]*)

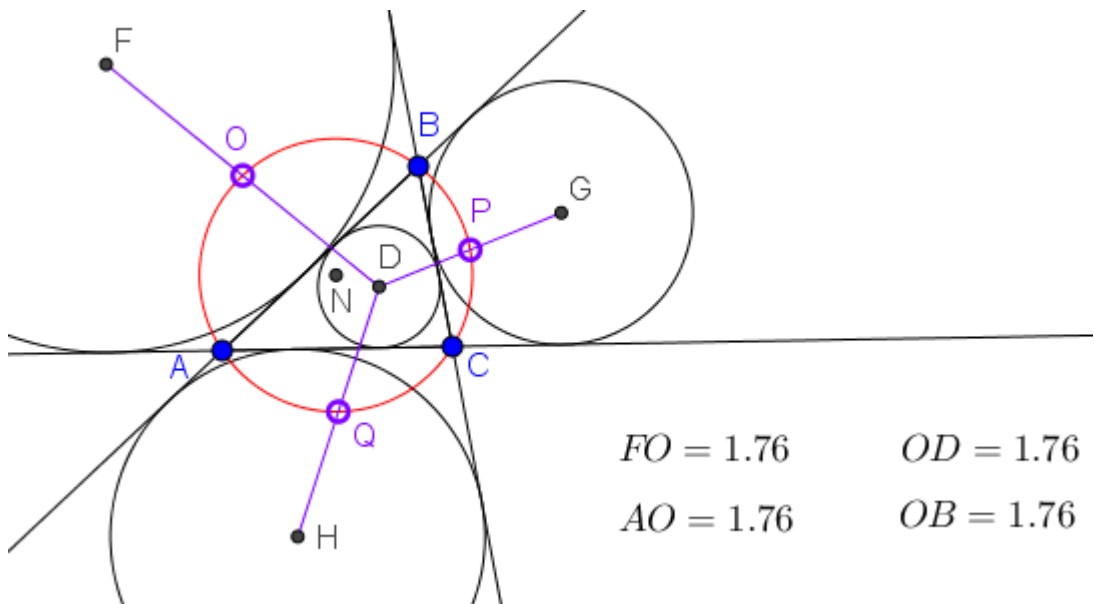


Рис 7. Иллюстрация к задаче “Теорема Мансиона”
(инструмент *Middles* и команда *Middles[A,B,C]*)

9.4.6. Прямая Эйлера

В 1765 г. Эйлер установил, что в любом треугольнике центроид, ортоцентр и центр описанной окружности лежат на одной прямой, которую теперь принято называть прямой Эйлера. Заметим, что для равностороннего треугольника центроид, ортоцентр и центр описанной окружности совпадают. Решите такие задачи.

F1. Создайте инструменты для нахождения в треугольнике центроида (M), ортоцентра (O) и центра описанной окружности (P).

F2. Создайте единый инструмент для нахождения в треугольнике центроида (M), ортоцентра (O) и центра описанной окружности (P).

Используйте инструменты задач *F1* и *F2* для построения в треугольнике перечисленных выше точек. Меняя позиции вершин треугольника, экспериментально убедитесь в справедливости теоремы Эйлера о расположении на одной прямой центроида, ортоцентра и центра описанной окружности (см. рис. 8).

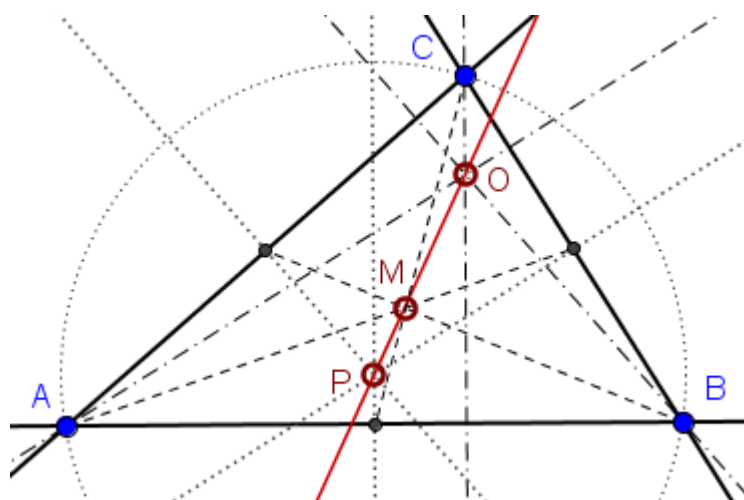


Рис. 8. Иллюстрация к задаче “Прямая Эйлера”
(M – центроид, O – ортоцентр, P – центр описанной окружности)

9.4.7. Окружность 9 точек

Известно, что в произвольном треугольнике основания медиан, основания высот и точки Фейербаха лежат на одной и той же окружности (Ж-В. Понселе, Ч. Брианшон и др. двадцатые годы XIX века). Позже эту окружность стали называть окружностью девяти точек. Решите следующие задачи.

G1. Создайте инструмент для построения окружности, проходящей через основания медиан треугольника. Экспериментально проверьте, что это и есть окружность девяти перечисленных выше точек (см. рис. 9). Меняя пози-

ции вершин треугольника убедитесь, что указанные точки остаются на одной окружности.

G2. Убедитесь, что центр окружности девяти точек лежит на прямой Эйлера и делит пополам отрезок между ортоцентром O и центром описанной окружности P (см. рис. 9).

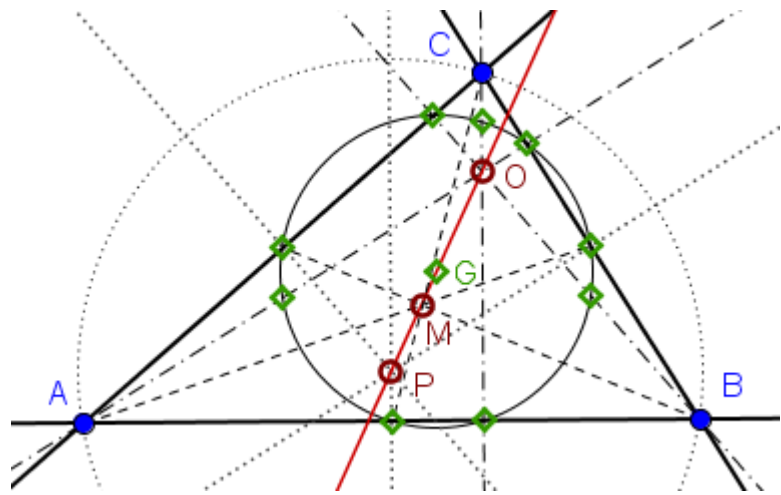


Рис. 9. Иллюстрация к задаче “Окружность девяти точек”

9.4.8. Теорема Фейербаха

Используя инструмент для вывода окружности девяти точек, экспериментально проверьте справедливость теоремы Фейербаха, утверждающей, что окружность девяти точек произвольного треугольника касается вписанной и всех трёх его внеписанных окружностей.

На рис. 10 на окружности девяти точек указаны 9 “стандартных” точек, а также точки ее касания с вписанной окружностью и тремя внеписанными окружностями. Чтобы не загромождать чертеж, имена этих точек спрятаны. Кроме того, выведена прямая Эйлера с центроидом (M), ортоцентром (O), центром описанной окружности (P) и центром окружности девяти точек (G). Изменяя позиции вершин треугольника ABC можно наблюдать за смещениями по окружности всех перечисленных выше точек. Заметим, что иногда окружность девяти точек называют окружностью 12 точек. В связи с данной задачей, наверное, ее стоило бы называть окружностью 13 точек.

9.4.9. Пересечение четырех окружностей девяти точек

Используя инструмент для вывода окружности девяти точек убедитесь, что в произвольном выпуклом четырехугольнике $ABCD$ окружности девяти точек треугольников ABC , BCD , CDA и DAB , на которые его разбивают две диагонали, пересекаются в одной точке.

Для решения задачи можно создать инструмент для вывода окружности девяти точек без указания самих точек. Можно строить ее, например, по трем серединам сторон треугольника. Далее останется применить этот инструмент к каждому из треугольников ABC , BCD , CDA и DAB . В результате получим чертеж, показанный на рис. 11. Меняя позиции вершин треугольника, экспериментально убедитесь в том, что точка пересечения четырех окружностей девяти точек может находиться как внутри, так и вне четырехугольника $ABCD$.

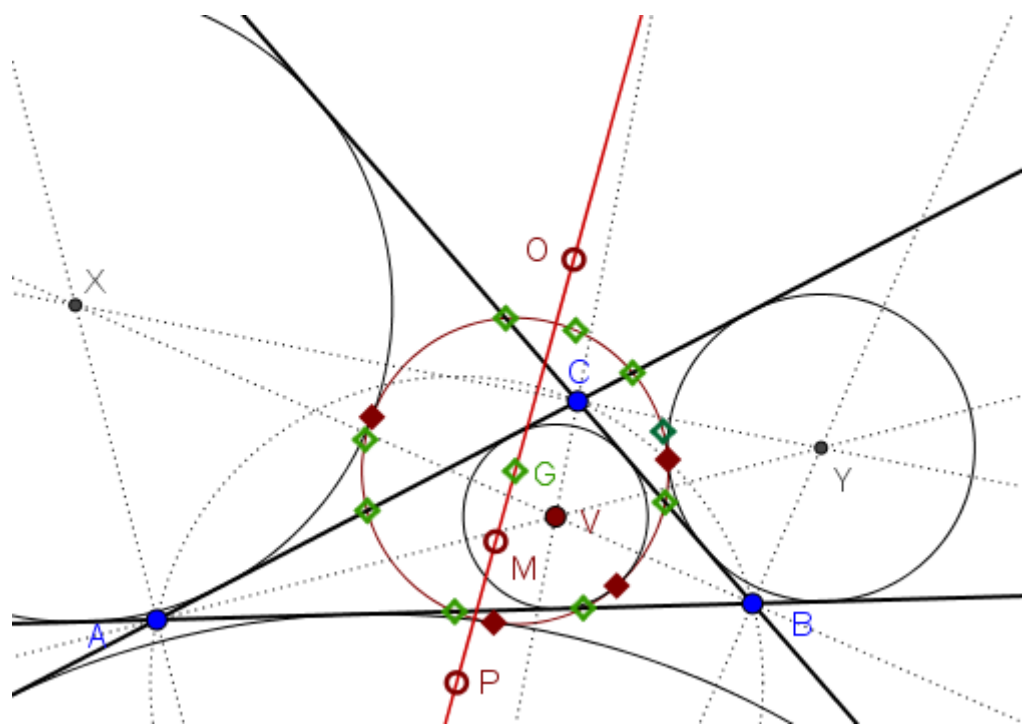


Рис. 10. Иллюстрация к задаче “Теорема Фейербаха”
(касание окружности девяти точек вписанной и внеписанных окружностей)

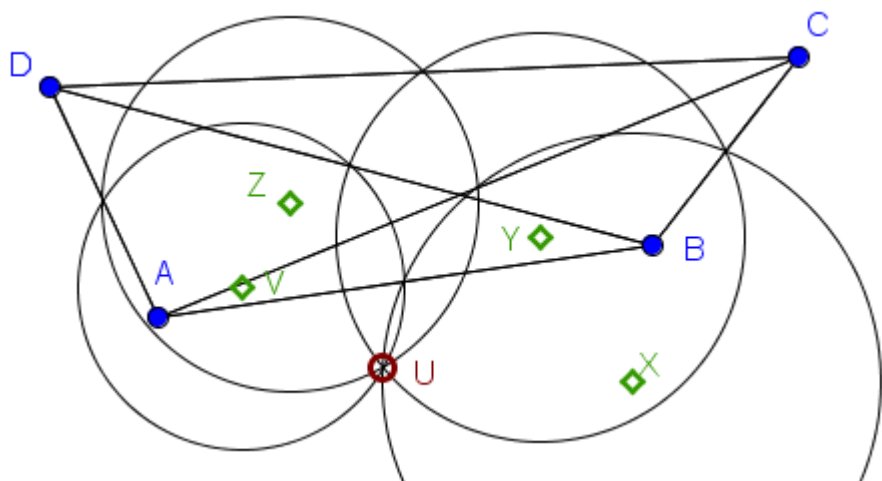


Рис. 11. Иллюстрация к задаче
“Пересечение четырех окружностей девяти точек”

10. Электронные таблицы

Электронные таблицы представляют собой удобный инструмент для автоматизации многих типов вычислений в области бухгалтерского учёта, а также при решении ряда экономических, математических, технических и иных задач. Электронные таблицы называют также табличными процессорами. Друг от друга они отличаются не только набором выполняемых функций, но и удобствами интерфейса с пользователем. Электронные таблицы позволяют проводить вычисления с данными, представленными в виде двумерных массивов. Впрочем, часто такие таблицы организуются в виде серии рабочих листов или связанных друг с другом окон, что фактически означает выход в третье измерение. Первую электронную таблицу создали Дэн Бриклин и Боб Фрэнкстон в 1979 г. (США). Их программа называлась *VisiCalc*. Вслед за ней появились многочисленные программные продукты этого класса. К настоящему времени явно лидирующее положение на рынке табличных процессоров занимают электронные таблицы *Excel* фирмы *Microsoft* [38-40].

Электронные таблицы *GeoGebra* во многом устроены аналогично другим табличным процессорам, но есть и отличия. При небольших размерах они имеют хорошо продуманный интерфейс с пользователем и тесно связаны с некоторыми средствами *GeoGebra*, такими как панели “Объекты”, *CAS*, “Полотно” и т. д. и инструменты “Калькулятор вероятностей”, “Регрессионный анализ” и т. д. За счет этой связи мощь электронных таблиц усиливается, позволяя, например, легко решать многие важные задачи теории вероятностей и статистики в соответствии с принципами наглядности, доступности и систематичности. Все это способствует успешному и быстрому обучению навыкам работы с электронными таблицами на нетрадиционном дидактическом материале.


10.1. Инструменты панели “Таблица”

Панель “Таблица” открывается и закрывается через основное меню командой “Вид\Таблица” или ключом *Ctrl+Shift+s*. На рис. 1 слева показан вид иерархического пиктографического меню встроенных инструментов панели “Таблица” в раскрытом виде, а справа – название выборов каждого подменю. Это меню автоматически появляется при открытии или активизации панели “Таблица”. В последующих пунктах этого раздела кратко описывается работа с каждым инструментом подменю “Создание списков” и “Суммы”. Что касается инструментов подменю “Анализ данных”, то их описание дается в разделе “Вероятность и статистика” второй части пособия.



Перемещение Объектов	Анализ Данных	Создание списков	Суммы
Перемещать	Анализ одной переменной	Создать список	Сумма
		Создать список точек	
	Регрессионный анализ		Среднее арифметическое
		Создать матрицу	
	Анализ нескольких переменных		Подсчитать количество ячеек
		Создать таблицу	
	Калькулятор вероятностей		Максимум
		Создать ломаную	
			Минимум

Рис. 1. Пиктографическое меню панели “Таблица” в раскрытом виде


10.2. Открытие и закрытие таблицы

- A. Вид\Таблица()

B. Ctrl+Shift+s

C. Click()  Spreadsheet

Открытие и закрытие электронной таблицы

A-C) Командой A из меню или ключом B¹ в главном окне *GeoGebra* открывается или закрывается панель электронной таблицы. Эта панель так и называется – “Таблица” (*Spreadsheet*) (см. рис. 2). Таблица представляет собой прокручиваемую в обоих направлениях область, разделенную серией горизонтальных и вертикальных отрезков линий на прямоугольные ячейки. Открыть такую таблицу можно и командой C, выполняемой из правой боковой панели главного окна системы. Закрытие (спрятывание) окна текущей электронной таблицы осуществляется или командой A, или нажатием на кнопку “”, расположенную в правой части строки заголовка таблицы. При этом окно таблицы исчезает, но ее содержимое сохраняется. И это можно увидеть, повторно выполнив любую из команд A-C.

¹ Мы везде приводим горячие ключи, которые используются в *GeoGebra*, работающей под управлением *OS Windows*. В *GeoGebra*, запускаемой в *Mac OS(X)*, ключи могут быть иными. Кроме того, ключи могут переопределяться многими открываемыми приложениями. Например, если запустить программу *HyperSnap* – “снятия” изображений с экрана монитора, то ключ B будет переопределен.

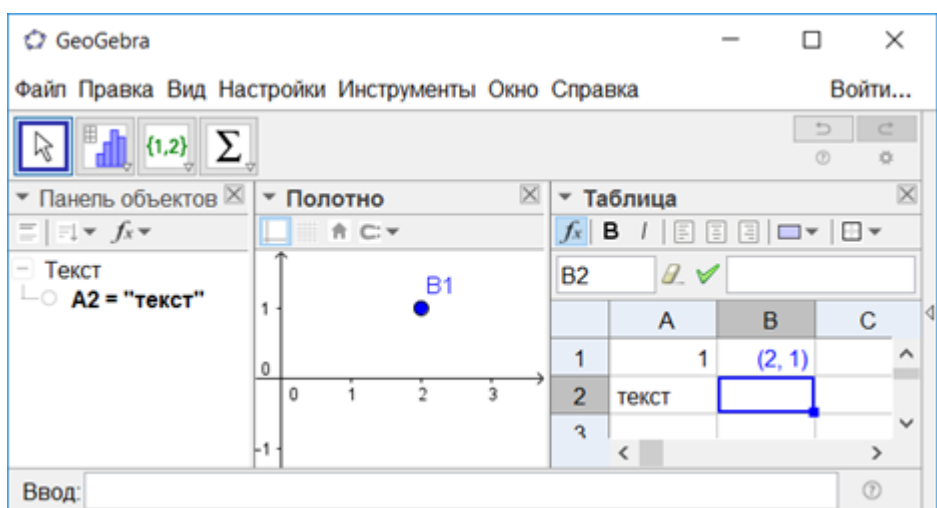


Рис. 2. Вид электронной таблицы, открытой в главном окне *GeoGebra*

Обратите внимание, что ввод объектов в ячейки электронной таблицы приводит к появлению их образов на других открытых панелях. На рис. 2 мы видим, что введенная в ячейку *B1* точка $(2,1)$ появилась на панели “*Полотно*” в виде реальной именованной точки в прямоугольной системе координат, а введенный в ячейку *A2* текст “*текст*” получил описание на панели “*Панели объектов*”. При этом имена создаваемых объектов вне электронной таблицы формируются по именам ячеек размещения соответствующих оригинальных объектов электронной таблицы.

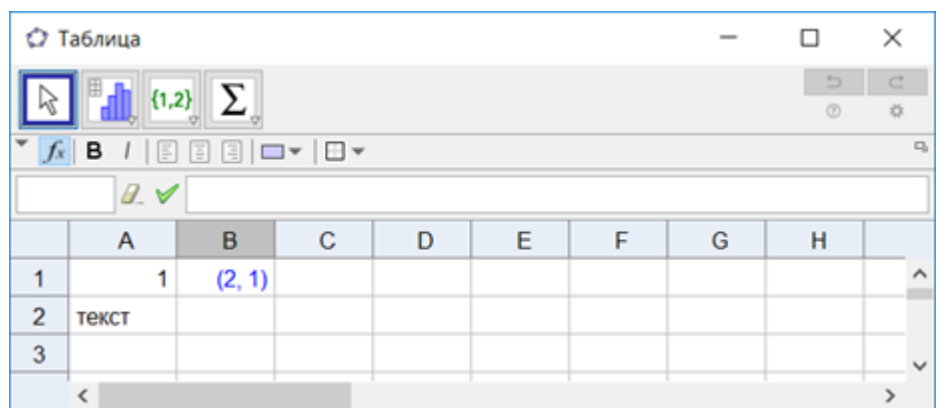







Рис. 3. Вид электронной таблицы, расположенной в отдельном окне

Для того, чтобы “*вытянуть*” электронную таблицу из главного окна *GeoGebra* и разместить ее в отдельном окне, достаточно выполнить команду *Click*() (см. рис. 3). Кнопка “” расположена в правой части заголовка таблицы и становится видимой при наведении курсора мыши на позицию перед кнопкой “”. И наоборот, для того, чтобы отдельное окно электронной таблицы встроить в виде панели в основное окно *GeoGebra*, требуется выполнить команду *Click*() . Кнопка “” расположена в конце правой части строки таблицы с пиктографическим меню.

10.3. Имена столбцов, строк и ячеек таблицы

В электронной таблице столбцы (колонки), строки и ячейки имеют уникальные имена, которые можно использовать в формулах и командах. Имена могут быть такими:

- для столбцов – прописные латинские буквы: $A, B, \dots, Z, AA, AB, \dots, ZZ, \dots, AAA, \dots, NTO$ или соответствующие им арабские десятичные числа: 1, 2, ... 9999. Буквенные имена используются в формулах, а цифровые имена, называемые также номерами столбцов, используются лишь в некоторых командах;
- для строк – арабские десятичные числа 1, 2, ... 9999;
- для ячеек – $A1, A5, B12, \dots, Z77, AB88$. Иными словами, ячейка, находящаяся на пересечении столбца Ω ($\Omega \in \{A, B, \dots, NTO\}$) и строки с номером $\#$ ($\# \in \{1, 2, \dots, 9999\}$) имеет имя $\Omega\#$.

Про указанные имена столбцов, строк и ячеек говорят, что они относительные. Если перед именем столбца или строки поставить символ \$, то получим абсолютный адрес столбца или абсолютный адрес строки. Если в имени ячейки перед именами столбца и строки поставить символы \$, то получим абсолютный адрес ячейки, например, $\$A\$1, \$B\7 . Кроме того, символ \$ в имени ячейки может стоять только перед именем столбца или только перед именем строки, например, $\$A1, B\7 . В этом случае говорят о смешанных адресах ячеек. Отметим, что использование относительных и абсолютных адресов ячеек приводит к разным типам копирований содержимого ячеек (см. 10.8).

10.4. Ввод чисел, точек и формул

В ячейки электронной таблицы можно вводить не только числа, но и любые объекты, поддерживаемые системой *GeoGebra*. Введенный в ячейку объект по возможности автоматически визуализируется в окне “Полотно”, если оно открыто. Итак, в ячейки могут быть введены:

- числа: 5, 7.67, -9, $2/3$, ...;
- точки в виде (абсцисса, ордината): (1, 2), (7.1, -1.3), ... или в виде комплексного числа $a+b \cdot i$, где a, b – действительные числа. При вводе в ячейку точки выводятся на графике, то есть на панели “Полотно”, если она открыта. Выводимые точки получают имена ячеек, в которых они расположены. Обратно, при формировании на полотне точки с именем, которое является именем ячейки таблицы, в соответствующей ячейке также появляется точка. Поэтому изменение имен точек на панели “Полотно” может приводить к изменению содержимого ячеек таблицы. Протаскивание точки по графику актуализирует значения ее координат в таблице. Через контекстное меню у точки на

графике можно менять свойства: имя, цвет, размер, стиль (вид) и т. п., а также изменить координатную систему (декартова ↔ полярная) с автоматическим изменением координат точки в таблице. Кроме того, можно изменить форму записи точки – $(a,b) \leftrightarrow a+b\cdot i$;

- тексты: *Ускорение, Глубина озера, Объем продаж* и т. п.
- формулы – выражения, которые могут содержать имена ячеек и встроенные функции. В большинстве электронных таблиц ввод формулы начинается с символа “=” (без кавычек). В таблицах *GeoGebra* знак “=” при вводе формул можно не писать. Однако, имеется установка, при которой знак “=” для формул становится обязательным. Реализовать ее можно в окне “*Настройки*”, где на вкладке “*Настройки-Таблица*” перед выбором “*Требовать “=” перед командами*” надо установить галочку “☒”. Сделать это можно разными способами, например, так: *RClick(таблица)\Настройка таблицы\✓*. Примеры записи формул: $=7+A1$, $=(B1+P2)*C3$, $=(7, 3)+D3$, Значением такого выражения должно быть число, точка или текст. Если значением выражения является точка, то она автоматически визуализируется на графике;


- списки: $\{5, 6, 99\}$, $\{7, 7, 11, 8, 9, 0\}$, ...;
- матрицы: $\{\{6,7\}, \{1,9\}, \{0,5\}\}$, ...;
- ...

Данные могут вводиться непосредственно в ячейки электронной таблицы или с помощью двухстрочной панели управления вводом данных (см. рис. 4).

В первой строке панели расположена кнопка “ f_x ” индикации и скрытия второй строки панели, а также кнопки, с помощью которых напрямую или через раскрывающиеся списки реализуются те или иные команды форматирования данных в ячейках таблицы. Вторая строка обеспечивает средства для непосредственного ввода данных в ячейки таблицы.



Рис. 4. Панель управления вводом и редактированием данных

A. Click(f_x)	C. Click(<input checked="" type="checkbox"/>)
B. Click()	D. Enter

Ввод данных в ячейки таблицы

А) Командой А выводится или прячется строка ввода данных в ячейки таблицы (см. рис. 4). В левом поле строки ввода создается адрес ячейки, а в правом – содержимое ячейки. Адрес ячейки формируется прямым вводом имени в поле ввода или щелчком по соответствующей ячейке. Содержимое ячейки формируется или прямым набором символов в правое поле ввода, или вставкой в него данных из буфера памяти. Допустим ввод и непосредственно в ячейки таблицы.

В) Командой В из второго поля ввода вычищаются уже имеющиеся там данные. Поле ввода становится пустым и ввод данных можно осуществлять в ту же самую ячейку заново.

С-D) Ввод данных в ячейку должен завершаться или командой *С*, или ключом *D*. При этом, если данные формируются в строке ввода, то завершать ввод можно и по *С*, и по *D*. Если же данные вводятся непосредственно в ячейку, то завершать ввод требуется ключом *D*. В любом случае текущей становится ячейка того же столбца, но следующей строки.

A. *Click*(вертикальная граница столбца)\Тащи и бросай
B. *Click*(горизонтальная граница строки)\Тащи и бросай

Изменение размеров ячеек. На панели “Таблица” можно изменять ширину ячеек конкретного столбца или высоту ячеек конкретной строки. Делается это для того, чтобы можно было видеть данные, не умещающиеся в ячейки стандартных размеров.

A) По команде *A* изменяется ширина ячеек конкретного столбца. Щелчок следует производить на правой вертикальной границе столбца в районе его имени. Протаскивая границу влево или вправо при нажатой левой клавише мыши, мы будем соответственно уменьшать или увеличивать ширину левого столбца ячеек.

B) По команде *B* изменяется высота ячеек конкретной строки. Щелчок следует производить на нижней горизонтальной границе строки в районе ее имени (номера). Протаскивая границу влево или вправо при нажатой левой клавише мыши, мы будем соответственно уменьшать или увеличивать высоту верхней строки ячеек.

A. <i>Click</i> (П) B. <i>Click</i> (K) C. <i>Click</i> (☐ или ☐ или ☐)	D. <i>Click</i> (☐▼) E. <i>Click</i> (☐▼)	<i>Форматирование данных в ячейках таблицы</i> A) Командой <i>A</i> шрифт с обычного начертания пере-
---	--	--

ключается на полужирное начертание и наоборот.

B) Командой *B* шрифт с прямого начертания переключается на курсив и наоборот.

C) Командой *C* содержимое ячейки выравнивается соответственно по левому краю, по центру и по правому краю. По умолчанию тексты выравниваются по левому краю, а числа и точки – по правому краю.

D) Командой *D* открывается палитра для установки цвета фона текущей ячейки, выделенной строки, выделенного столбца или выделенного диапазона ячеек (см. рис. 5a).

E) Командой *E* открывается палитра для установки типа границы текущей ячейки, выделенной строки, выделенного столбца или выделенного диапазона ячеек (см. рис. 5b).

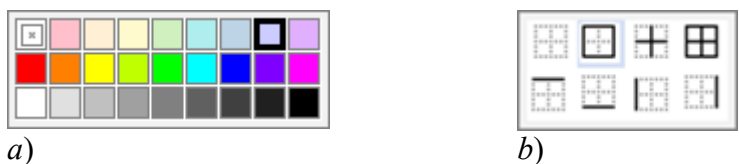


Рис. 5 (а, б). Палитры для установки цвета фона и типа границы ячеек

10.5. Навигация по ячейкам

Щелчок по ячейке таблицы делает ее активной или, по-другому, текущей. Граница такой ячейки становится утолщенной, а в ее нижнем правом углу появляется черный квадратный маркер. В активной ячейке непосредственно или через поле ввода можно формировать и (или) редактировать данные. Ниже приводятся ключи, позволяющие изменять позицию активной ячейки.

<p>A. \uparrow, \downarrow, B. \leftarrow или <i>Shift+Tab</i>, \rightarrow или <i>Tab</i> C. <i>Home</i>, <i>Ctrl+Home</i> D. <i>End</i></p>	<p>E. <i>Ctrl</i>+\uparrow, <i>Ctrl</i>+\downarrow F. <i>Ctrl</i>+\leftarrow, <i>Ctrl</i>+\rightarrow G. <i>PgUp</i>, <i>PgDown</i> H. <i>Ctrl</i>+<i>PgUp</i>, <i>Ctrl</i>+<i>PgDown</i></p>
--	--

Навигация по ячейкам

A) Ключи A делают активной соседнюю ячейку, расположенную в текущем столбце соответственно выше или ниже текущей ячейки, если такие ячейки имеются.

B) Ключи B делают активной соседнюю ячейку, расположенную в текущей строке соответственно левее или правее текущей ячейки, если такие ячейки имеются.

C) Ключи C делают активной соответственно начальную ячейку текущей строки (*Home*) и начальную ячейку A1 таблицы (*Ctrl+Home*).

D) Ключ D делает активной ячейку, находящуюся на пересечении последней непустой строки и последнего непустого столбца.

E) Пусть текущая ячейка пустая. Тогда ключи E делают активной ближайшую непустую ячейку текущего столбца, расположенную соответственно выше или ниже текущей ячейки. Если выше текущей ячейки пустых ячеек нет, то по “*Ctrl*+ \uparrow ” активной становится начальная ячейка текущего столбца. Если текущая ячейка находится в строке *k* и ниже ее пустых ячеек нет, то по “*Ctrl*+ \downarrow ” при $k < 100$ активной становится ячейка текущего столбца и 100-ой строки, а при $k \geq 100$ – ячейка текущего столбца, лежащая в последней строке, с которой в текущей сессии уже была работа (или в следующей за ней строке). Если текущая ячейка не пуста, то ключи E оставляют ее текущей.

F) Пусть текущая ячейка пустая. Тогда ключи F делают активной ближайшую непустую ячейку текущей строки, расположенную соответственно левее или правее текущей ячейки. Если левее текущей ячейки пустых ячеек нет, то по “*Ctrl*+ \leftarrow ” активной становится начальная ячейка текущей строки. Если

текущая ячейка находится в столбце ζ и правее ее непустых ячеек нет, то по “Ctrl+→” при $\zeta < Z$ активной становится ячейка текущей строки столбца Z , а при $\zeta \geq Z$ – ячейка текущей строки, лежащая в последнем столбце, с которым в текущей сессии уже была работа (или в следующем за ним столбце). Если текущая ячейка не пуста, то ключи E оставляют ее текущей.

G) Ключи G делают активной ячейку, расположенную от текущей ячейки соответственно на одну страницу вверх ($PgUp$) или одну страницу вниз ($PgDown$). Если сместиться вверх на целую страницу нельзя, то активной становится начальная ячейка текущего столбца. Если сместиться вниз на целую страницу нельзя, то активной становится “последняя” ячейка текущего столбца (см. E).

H) Ключи H делают активной ячейку, расположенную от текущей ячейки соответственно на одну страницу влево ($Ctrl+PgUp$) или одну страницу вправо ($Ctrl+PgDown$). Если сместиться влево на целую страницу нельзя, то активной становится начальная ячейка текущей строки. Если сместиться вправо на целую страницу нельзя, то активной становится “последняя” ячейка текущей строки (см. F).

10.6. Активные ячейки и активные группы ячеек

Наряду с понятием активной или текущей ячейки, можно вести речь и об активной или текущей группе ячеек. Активной группой ячеек может быть любая прямоугольная область таблицы из смежных друг другу ячеек. Оформляется она также, как и активная ячейка, то есть появляется утолщенная граница, а в правом нижнем углу – черный квадратный маркер. Ниже приводится ряд команд, позволяющих создавать активные группы ячеек.

A. Shift+→ B. Shift+←	C. Shift+↓ D. Shift+↑	E. Shift+Click(ячейка) F. Shift+Home	G. Shift+Ctrl+Home H. Shift+End
--------------------------	--------------------------	---	------------------------------------

Создание активной группы ячеек

A) Ключом A в активную группу объединяются активная ячейка и ячейка справа от нее. Если перед выполнением A уже создана некоторая активная прямоугольная группа ячеек, то ключом A к ней добавляются ячейки столбца справа от нее в пределах строк группы.

B) Ключом B в активную группу объединяются активная ячейка и ячейка слева от нее, если она имеется. Если перед выполнением B уже создана активная прямоугольная группа ячеек, то возможны такие случаи. Если группа состоит из одного столбца, то по ключу B к ней добавляются ячейки соседнего столбца слева от нее в пределах строк группы. Если группа состоит из нескольких столбцов, то по B от нее урезается правый столбец.

С) Ключом C в активную группу объединяются активная ячейка и соседняя ячейка снизу. Если перед выполнением C уже создана активная прямоугольная группа ячеек, то по ключу C к ней добавляются ячейки строки ниже нее в пределах столбцов группы.

Д) Ключом D в активную группу объединяются активная ячейка и соседняя ячейка сверху, если она имеется. Если перед выполнением D уже создана активная прямоугольная группа ячеек, то возможны такие случаи. Если группа состоит из одной строки, то ключом D к ней добавляются ячейки соседней строки выше нее в пределах столбцов группы. Если группа состоит из нескольких строк, то по D от нее урезается нижняя строка.

Е) Команда E выполняется так: при нажатой клавише *Shift* производится щелчок левой кнопкой мыши по любой ячейке α таблицы. Тем самым активной становится группа ячеек, у которой в противоположных углах находятся бывшая активная ячейка и ячейка α . Если уже создана активная прямоугольная группа ячеек, то по E формируется новая активная группа, у которой в противоположных углах находится вся или часть бывшей активной группы и ячейка α (здесь несколько возможных случаев).

Ф) Ключом F в активную группу объединяются бывшая активная ячейка и все ячейки текущей строки левее нее. Если перед выполнением F уже создана активная прямоугольная группа ячеек, то по ключу F в новую активную группу объединяются верхняя левая ячейка группы и все ячейки текущей строки левее нее.

Г) Ключом G создается активная группа, у которой верхний левый угол и правый нижний угол занимают соответственно ячейка $A1$ и бывшая активная ячейка. Если уже создана активная прямоугольная группа ячеек, то по ключу G формируется новая активная группа, у которой левый верхний и правый нижний углы занимают соответственно ячейка $A1$ и ячейка левого верхнего угла бывшей активной группы.

Н) Пусть β – ячейка, расположенная на пересечении последнего непустого столбца и последней непустой строки. Рассмотрим возможные случаи. Пусть α – активная ячейка. Если α расположена не ниже и не правее β , то ключом H формируется активная группа, у которой верхняя левая ячейка – α , а нижняя правая ячейка – β . Если активная ячейка α расположена ниже, но не правее β , то ключом H формируется активная строка с начальной ячейкой α и конечной ячейкой с тем же столбцом, что и β . Если активная ячейка α расположена правее, но не ниже β , то ключом H формируется активный столбец с начальной ячейкой α и конечной ячейкой с той же строкой, что и β . В иных случаях при активной ячейке α по H никаких действий не производится. Описывать варианты действия ключа H при уже созданной активной группе ячеек мы не будем, предоставляя читателю возможность разобраться в этом самостоятельно.

- | |
|---|
| A. <i>Click(имя столбца)</i>
B. <i>Click(имя столбца1)\Shift+Click(имя столбца2)</i>
C. <i>Click(номер строки)</i>
D. <i>Click(номер строки1)\Shift+Click(номер строки2)</i> |
|---|

Создание активных групп из целых строк и столбцов

- A) Пусть k – номер последней строки таблицы, в которой была какая-либо работа. Ключом A в активную группу выделяются ячейки строк от 1 до $\max(100, k)$ того столбца, по имени которого был произведен щелчок мышью.
- B) Командой B в активную группу выделяются все ячейки столбцов, по именам которых были произведены щелчки мышью, а также все ячейки всех столбцов между ними.
- C) Пусть ζ – имя последнего столбца таблицы, в котором была какая-либо работа. Ключом C в активную группу выделяются все ячейки столбцов от A до $\max(\zeta, Z)$ той строки, по номеру которой был произведен щелчок мышью.
- D) Командой D в активную группу выделяются все ячейки строк, по номерам которых были произведены щелчки мышью, а также все ячейки всех строк между ними.

10.7. Выделенные ячейки

Наряду с активными ячейками и активными группами ячеек рассматривают также так называемые выделенные ячейки. Такие ячейки по умолчанию имеют светло-голубой фон. Отметим, что выделенными являются непустая активная ячейка, а также все непустые ячейки активной группы ячеек. Однако выделять можно и иные совокупности ячеек.

- | |
|--|
| A. <i>Ctrl+a</i>
B. <i>Правка\Выбрать все</i>
C. <i>Click(α_1)\Ctrl+Click(α_2)\...\Ctrl+Click(α_n)</i> |
|--|

Выделение ячеек
A-B) По ключу A или команде B , выполняемой через основное меню, выделяются

все непустые ячейки таблицы.


C) По команде C выделяются все непустые ячейки из совокупности ячеек $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$. Щелчки мышью можно осуществлять также по именам столбцов и именам (номерам) строк. В этом случае к уже выделенным ячейкам будут добавляться сразу все непустые ячейки соответствующих строк и столбцов.

- | | |
|---|--|
| A. <i>Ctrl+Shift+j</i>
B. <i>Правка\Выбрать потомков</i> | C. <i>Ctrl+j</i>
D. <i>Правка\Выбрать предков</i> |
|---|--|

Выделение ячеек предков и ячеек потомков

A-B) По ключу *A* или команде *B*, выполняемой через основное меню, для предварительно выделенных ячеек α дополнительно выделяются все их потомки, то есть ячейки, в которых есть формулы со ссылками на какие-либо ячейки α .

C-D) По ключу *C* или команде *D*, выполняемой через основное меню, для предварительно выделенных ячеек α дополнительно выделяются все их предки, то есть ячейки, на которые имеются ссылки в формулах каких-либо ячеек α .

A. *RClick(ячейка)\Настройка табл.\Свойства0)*

B. *Ctrl+l*

C. *Правка\Выбрать текущий слой*

D. *Ctrl+i*

E. *Правка\Инвертировать текущий слой*

Выделение ячеек в слоях

A) По умолчанию все данные вводятся для их визуализации на панели “Полотно” в графическом слое 0. Всего слоев для вывода может быть 10 с номерами: 0, 1, ..., 9. Командой *A*, выполняемой через контекстное меню, можно сменить номер текущего слоя для вывода изображений. Однако следует помнить, что вновь вводимые данные всегда визуализируются в слое с наибольшим номером.

B-C) По ключу *B* или команде *C*, выполняемой через основное меню, выделяются все непустые ячейки текущего слоя таблицы.

D-E) По ключу *D* или команде *E*, выполняемой через основное меню, выделение ячеек таблицы инвертируется, то есть все выделенные ячейки становятся невыделенными, а все невыделенные ячейки становятся выделенными.

<div> <div>A. <i>Delete</i></div> <div>B. <i>Backspace</i></div> </div>	<div> <div>C. <i>RClick(Ω)\Удалить объекты</i></div> <div>D. <i>Правка\Удалить</i></div> </div>
---	---

Удаление содержимого активной группы ячеек

A-D) По командам *A-D* содержимое активной группы ячеек Ω вычищается, то есть все ячейки Ω становятся пустыми. Команды *A* и *B* выполняются нажатием соответствующих клавиш, команда *C* – через контекстное меню и команда *D* – через основное меню.

A. *Правка/Удалить*

Удаление содержимого выделенной группы ячеек

A) Как мы видели, по данной команде удаляется содержимое активной группы ячеек. Однако по ней можно удалять и любую выделенную группу ячеек.

10.8. Копирования, вырезки и вставки данных

Копирования и вырезки данных в буфер памяти и вставки их из буфера в таблицу являются одними из самых используемых действий, которые применяются к активным ячейкам, столбцам, строкам и группам ячеек. Вместе с перемещением и удалением данных копирования и вставки являются базовыми операциями, без которых работа с таблицами невозможна, или, по крайней мере, сильно затруднена.

A. *Ctrl+c*
B. *RClick(Ω)\Копировать*
C. *Правка\Копировать*

Копирование содержимого активного объекта в буфер памяти

А) По команде А содержимое активного объекта Ω помещается в буфер памяти. Копировать можно:

- ячейку (*Click(ячейка)*);
- столбец ячеек (*Click(заголовок столбца)*);
- строку ячеек (*Click(номер строки)*);
- группу ячеек (*Click(ячейка1)\Shift+Click(ячейка2)*);

В) По команде В, выполняемой через контекстное меню, содержимое активного объекта Ω помещается в буфер памяти.

С) По команде С, выполняемой через основное меню, содержимое активного объекта Ω помещается в буфер памяти.

Замечания. 1. При копированиях А-С кроме содержимого ячеек сохраняется его цвет и цвет фона ячеек. Другое форматирование (границы ячеек, выравнивание содержимого и т. п.) не сохраняется.

A. *Ctrl+x*
B. *RClick(Ω)/Вырезать*

Вырезание содержимого активного объекта в буфер памяти

А-В) Команды А-В выполняются так. Содержимое активного объекта Ω помещается в буфер памяти без какого либо форматирования и удаляется из исходной позиции электронных таблиц. Цвет фона и границы ячеек на исходной позиции остаются неизменными. Если вырезалась формула, то в буфер помещается не она, а лишь вычисленное по ней значение. Заметим, что при копировании формулы, в буфер помещается именно формула, возможно преобразованная.

A. <i>Click(ячейка)\Ctrl+v</i> B. <i>RClick(ячейка)\Вставить</i>	C. <i>Click(ячейка)\Правка\Вставить</i>
---	---

Вставка содержимого буфера памяти

А) По команде А содержимое буфера памяти (ячейка, столбец, строка, группа ячеек) вставляется как есть, начиная с текущей ячейки памяти.

В) По команде *B*, выполняемой через контекстное меню, содержимое буфера памяти вставляется, начиная с текущей ячейки памяти.

С) По команде *C*, выполняемой через основное меню, содержимое буфера памяти вставляется, начиная с текущей ячейки памяти.

Замечания. 1. Если при вставках (копированиях) *A-C* в буфере находится некоторая формула, содержащая адреса ячеек, то эти адреса преобразуются по определенным правилам (см. ниже).

2. Если данные необходимо переместить как есть из одной области таблицы в другую, то их следует скопировать в требуемое место через буфер памяти, а затем удалить из исходного места. Если в перемещаемых данных нет формул с адресами ячеек, то вырезка и копирование также подходят для выполнения этой операции.

Преобразование адресов в формулах при копированиях-вставках

Теперь поговорим о том, как преобразуются относительные, абсолютные и смешанные адреса ячеек в формулах при их вставках из буфера памяти в новые позиции таблицы. Начнем с описания преобразований относительных адресов. Пусть в ячейке α находится формула f , содержащая адреса некоторых ячеек, и f копируется из α в некоторую новую ячейку β , отстоящую от α на s ($s=0, 1, 2, \dots$) ячеек вправо по столбцам и t ячеек вверх или вниз по строкам ($t=0, 1, 2, \dots$). Тогда все относительные адреса f в β преобразуются по тому же самому правилу, то есть номера столбцов увеличиваются на s ($s=0, 1, 2, \dots$), а номера строк соответственно уменьшаются или увеличиваются на t . Пусть, далее, f копируется из α в ячейку β , находящуюся слева от текущего столбца и по строкам на t ячеек выше или ниже α ($t=0, 1, 2, \dots$). Тогда все относительные адреса f в β преобразуются так: номера столбцов не изменяются, а номера строк соответственно уменьшаются или увеличиваются на t . Если при копировании f в β формируется несуществующий адрес таблицы, то преобразование отменяется и этот адрес в f остается без изменений. Если по преобразованной формуле вычисления не могут быть проведены, то в β появляется значение “?”.

Абсолютные адреса ($\$ \Omega \$ n$, $\Omega \in \{A, B, \dots\}$, $n \in \{1, 2, \dots\}$) ячеек из формулы f при ее копировании вообще не меняются. В смешанных адресах вида $\$ \Omega n$ при копировании f номера столбцов остаются неизменными, а номера строк изменяются как и для относительных адресов. В смешанных адресах вида $\Omega \$ n$ при копировании f номера столбцов изменяются как в относительных адресах, а номера строк остаются неизменными.

10.9. Копирования данных с помощью маркеров

Имеется еще один способ копирования активных групп ячеек протаскиванием их маркеров вправо, влево, вверх или вниз. Рассмотрим отдельно случаи, когда в активной группе ячеек нет текстовых значений и когда они есть.

А. В ячейках активной группы нет текстовых значений. Здесь возможны такие случаи:

1. *активна одна ячейка α* . При протягивании маркера α превращается в активную группу ячеек Ω и содержимое α без какого-либо форматирования копируется во все новые ячейки Ω . Форматирование самой ячейки α на другие ячейки Ω не распространяется. Если в α расположена формула с адресами ячеек, то при копировании эти адреса преобразуются так, как это описано в предыдущем разделе;

2. *активна группа ячеек β с n ($n \neq 2$) строками и m ($m \neq 2$) столбцами*. При протягивании маркера копируются ячейки ближайшей границы β , в сторону от которой перемещался маркер. При этом каждая ячейка границы копируется как одна активная ячейка;

3. *активна группа ячеек β с двумя строками и m ($m \neq 2$) столбцами*. При протягивании маркера вправо или влево копирование проводится также, как и в случае 2. При протягивании маркера вниз или вверх каждый столбец β формирует значения для копирования по-своему. Если в столбце не оба значения являются числами, то копирование проводится также, как и в случае 2. Если в столбце оба значения – числа (сверху a , снизу b), то при копировании в верхние и нижние ячейки помещаются соответственно последовательные числа $a-\delta$, $a-2\delta$, ... и $b+\delta$, $b+2\delta$, ... ($\delta=b-a$). Иными словами, формируются арифметические прогрессии;

4. *активна группа ячеек β с n ($n \neq 2$) строками и двумя столбцами*. При протягивании маркера вниз или вверх копирование проводится так же, как и в случае 2. При протягивании маркера вправо или влево каждая строка β формирует значения для копирования по-своему. Если в строке не оба значения являются числами, то копирование проводится так же, как и в случае 2. Если в строке оба значения – числа (слева a , справа b), то при копировании в левые и правые ячейки помещаются соответственно последовательные числа $a-\delta$, $a-2\delta$, ... и $b+\delta$, $b+2\delta$, ... ($\delta=b-a$). Иными словами, формируются арифметические прогрессии.

5. *активна группа ячеек с двумя строками и двумя столбцами*. При протягивании маркера вниз или вверх каждый столбец β формирует значения для копирования как в 3 при протягивании маркера в этом же направлении. При протягивании маркера влево или вправо каждая строка β формирует значения для копирования как в 4 при протягивании маркера в этом же направлении.

В. В выделенных ячейках есть текстовые значения. Пусть выделена ячейка или группа ячеек α . Тогда при протягивании маркера в любую сторону в эту же сторону копируются и ближайшие граничные ячейки α . При этом, формулы с адресами ячеек копируются обычным образом, а текстовые значения, числовые значения и точки (пары числовых значений) копируются как есть.

На рис. 6 показаны примеры копирования содержимого областей по разным направлениям протяжкой маркера. Серым цветом показаны выделенные области.

<table><tr><td></td><td></td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td>3</td><td></td><td></td></tr><tr><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td></tr><tr><td></td><td></td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td>3</td><td></td><td></td></tr></table> <p>i)</p>			3					3			3	3	3	3	3			3					3					3			<table><tr><td></td><td></td><td>-3</td><td></td><td></td></tr><tr><td></td><td></td><td>-1</td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td></tr><tr><td></td><td></td><td>5</td><td></td><td></td></tr><tr><td></td><td></td><td>7</td><td></td><td></td></tr></table> <p>ii)</p>			-3					-1			1	1	1	1	1	3	3	3	3	3			5					7			<table><tr><td></td><td></td><td>1</td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td>1</td><td>3</td><td></td><td></td></tr><tr><td>-3</td><td>-1</td><td>1</td><td>3</td><td>5</td><td>7</td></tr><tr><td></td><td></td><td>1</td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td>1</td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td>1</td><td>3</td><td></td><td></td></tr></table> <p>iii)</p>			1	3					1	3			-3	-1	1	3	5	7			1	3					1	3					1	3																									
		3																																																																																																																							
		3																																																																																																																							
3	3	3	3	3																																																																																																																					
		3																																																																																																																							
		3																																																																																																																							
		3																																																																																																																							
		-3																																																																																																																							
		-1																																																																																																																							
1	1	1	1	1																																																																																																																					
3	3	3	3	3																																																																																																																					
		5																																																																																																																							
		7																																																																																																																							
		1	3																																																																																																																						
		1	3																																																																																																																						
-3	-1	1	3	5	7																																																																																																																				
		1	3																																																																																																																						
		1	3																																																																																																																						
		1	3																																																																																																																						
<table><tr><td></td><td>3</td><td>2</td><td>7</td><td>6</td><td></td></tr><tr><td></td><td>2</td><td>2</td><td>5</td><td>5</td><td></td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>4</td></tr><tr><td>0</td><td>0</td><td>2</td><td>1</td><td>3</td><td>3</td></tr><tr><td></td><td>-1</td><td>2</td><td>-2</td><td>2</td><td></td></tr><tr><td></td><td>-2</td><td>2</td><td>-3</td><td>1</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <p>iv)</p>		3	2	7	6			2	2	5	5		1	1	2	3	4	4	0	0	2	1	3	3		-1	2	-2	2			-2	2	-3	1								<table><tr><td></td><td>1</td><td>2</td><td></td><td></td></tr><tr><td></td><td>1</td><td>2</td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td><td>4</td><td>6</td></tr><tr><td></td><td>0</td><td>2</td><td></td><td></td></tr><tr><td></td><td>0</td><td>2</td><td></td><td></td></tr></table> <p>v)</p>		1	2				1	2			0	1	2	3	4	5	4	3	2	1	-2	0	2	4	6		0	2				0	2			<table><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>4</td></tr><tr><td>4</td><td>4</td><td>3</td><td>2</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>2</td><td>1</td><td>3</td><td>3</td></tr><tr><td></td><td>0</td><td>2</td><td>1</td><td>3</td><td></td></tr><tr><td></td><td>0</td><td>2</td><td>1</td><td>3</td><td></td></tr></table> <p>vi)</p>		1	2	3	4			1	2	3	4		1	1	2	3	4	4	4	4	3	2	1	1	0	0	2	1	3	3		0	2	1	3			0	2	1	3	
	3	2	7	6																																																																																																																					
	2	2	5	5																																																																																																																					
1	1	2	3	4	4																																																																																																																				
0	0	2	1	3	3																																																																																																																				
	-1	2	-2	2																																																																																																																					
	-2	2	-3	1																																																																																																																					
	1	2																																																																																																																							
	1	2																																																																																																																							
0	1	2	3	4																																																																																																																					
5	4	3	2	1																																																																																																																					
-2	0	2	4	6																																																																																																																					
	0	2																																																																																																																							
	0	2																																																																																																																							
	1	2	3	4																																																																																																																					
	1	2	3	4																																																																																																																					
1	1	2	3	4	4																																																																																																																				
4	4	3	2	1	1																																																																																																																				
0	0	2	1	3	3																																																																																																																				
	0	2	1	3																																																																																																																					
	0	2	1	3																																																																																																																					
<table><tr><td></td><td></td><td>(-3,6)</td><td></td><td></td></tr><tr><td></td><td></td><td>(-1,4)</td><td></td><td></td></tr><tr><td>(1,2)</td><td>(1,2)</td><td>(1,2)</td><td>(1,2)</td><td>(1,2)</td></tr><tr><td>(3,0)</td><td>(3,0)</td><td>(3,0)</td><td>(3,0)</td><td>(3,0)</td></tr><tr><td></td><td></td><td>(5,-2)</td><td></td><td></td></tr><tr><td></td><td></td><td>(7,-4)</td><td></td><td></td></tr></table> <p>vii)</p>			(-3,6)					(-1,4)			(1,2)	(1,2)	(1,2)	(1,2)	(1,2)	(3,0)	(3,0)	(3,0)	(3,0)	(3,0)			(5,-2)					(7,-4)			<table><tr><td></td><td>(1,2)</td><td>(3,0)</td><td></td><td></td></tr><tr><td></td><td>(1,2)</td><td>(3,0)</td><td></td><td></td></tr><tr><td>(-1,4)</td><td>(1,2)</td><td>(3,0)</td><td>(5,-2)</td><td>(7,-4)</td></tr><tr><td></td><td>(1,2)</td><td>(3,0)</td><td></td><td></td></tr><tr><td></td><td>(1,2)</td><td>(3,0)</td><td></td><td></td></tr><tr><td></td><td>(1,2)</td><td>(3,0)</td><td></td><td></td></tr></table> <p>viii)</p>		(1,2)	(3,0)				(1,2)	(3,0)			(-1,4)	(1,2)	(3,0)	(5,-2)	(7,-4)		(1,2)	(3,0)				(1,2)	(3,0)				(1,2)	(3,0)																																																														
		(-3,6)																																																																																																																							
		(-1,4)																																																																																																																							
(1,2)	(1,2)	(1,2)	(1,2)	(1,2)																																																																																																																					
(3,0)	(3,0)	(3,0)	(3,0)	(3,0)																																																																																																																					
		(5,-2)																																																																																																																							
		(7,-4)																																																																																																																							
	(1,2)	(3,0)																																																																																																																							
	(1,2)	(3,0)																																																																																																																							
(-1,4)	(1,2)	(3,0)	(5,-2)	(7,-4)																																																																																																																					
	(1,2)	(3,0)																																																																																																																							
	(1,2)	(3,0)																																																																																																																							
	(1,2)	(3,0)																																																																																																																							
<table><tr><td></td><td>tex</td><td>1</td><td></td><td></td></tr><tr><td></td><td>tex</td><td>1</td><td></td><td></td></tr><tr><td>tex</td><td>tex</td><td>1</td><td>1</td><td>1</td></tr><tr><td>2</td><td>2</td><td>3</td><td>3</td><td>3</td></tr><tr><td>H</td><td>h</td><td>5</td><td>5</td><td>5</td></tr><tr><td></td><td>7</td><td>8</td><td></td><td></td></tr><tr><td></td><td>7</td><td>8</td><td></td><td></td></tr><tr><td></td><td>7</td><td>8</td><td></td><td></td></tr></table> <p>ix)</p>		tex	1				tex	1			tex	tex	1	1	1	2	2	3	3	3	H	h	5	5	5		7	8				7	8				7	8			<table><tr><td>3</td><td>2</td><td>текст</td><td>(1,10)</td><td>3</td><td></td></tr><tr><td>11</td><td></td><td>текст</td><td>(1,10)</td><td>3</td><td></td></tr><tr><td>текст</td><td>текст</td><td>текст</td><td>(1,10)</td><td>3</td><td>3</td></tr><tr><td>h</td><td>h</td><td>h</td><td>2</td><td>A1+3</td><td>B1+3</td></tr><tr><td></td><td></td><td>h</td><td>2</td><td>6</td><td>5</td></tr><tr><td></td><td></td><td>h</td><td>2</td><td>A2+3</td><td></td></tr><tr><td></td><td></td><td>h</td><td>2</td><td>14</td><td></td></tr><tr><td></td><td></td><td>h</td><td>2</td><td>A3+3</td><td></td></tr></table> <p>x)</p>	3	2	текст	(1,10)	3		11		текст	(1,10)	3		текст	текст	текст	(1,10)	3	3	h	h	h	2	A1+3	B1+3			h	2	6	5			h	2	A2+3				h	2	14				h	2	A3+3																																	
	tex	1																																																																																																																							
	tex	1																																																																																																																							
tex	tex	1	1	1																																																																																																																					
2	2	3	3	3																																																																																																																					
H	h	5	5	5																																																																																																																					
	7	8																																																																																																																							
	7	8																																																																																																																							
	7	8																																																																																																																							
3	2	текст	(1,10)	3																																																																																																																					
11		текст	(1,10)	3																																																																																																																					
текст	текст	текст	(1,10)	3	3																																																																																																																				
h	h	h	2	A1+3	B1+3																																																																																																																				
		h	2	6	5																																																																																																																				
		h	2	A2+3																																																																																																																					
		h	2	14																																																																																																																					
		h	2	A3+3																																																																																																																					

Рис. 6. Копирования протяжкой маркера областей с числами, точками, текстами и формулами с адресами ячеек

В примерах рис. 6 демонстрируется:

- i. копирование из одной активной ячейки с числовым значением;
- ii. копирование из активной двухстрочной группы ячеек с числовыми значениями. При протягивании маркера вверх или вниз формируются арифметические прогрессии. При протягивании маркера влево или вправо копируются значения из соответствующих строк группы;

iii. копирование из активной двухстолбцевой группы ячеек с числовыми значениями. При протягивании маркера влево или вправо формируются арифметические прогрессии. При протягивании маркера вверх или вниз копируются значения из соответствующих столбцов группы;

iv. копирование из активной двухстрочной группы ячеек с числовыми значениями. При протягивании маркера вверх или вниз формируются арифметические прогрессии (см. ii). При протягивании маркера влево или вправо копируются значения из соответствующих ближайших граничных ячеек группы;

v. копирование из активной двухстолбцевой группы ячеек с числовыми значениями. При протягивании маркера влево или вправо формируются арифметические прогрессии (см. iii). При протягивании маркера вверх или вниз копируются значения из соответствующих ближайших граничных ячеек группы;

vi. копирование из активной группы ячеек с числовыми значениями при n строках и m столбцах ($n, m \neq 2$). При протягивании маркера в любом допустимом направлении копируются значения из соответствующих ближайших граничных ячеек группы;

vii. копирование из активной двухстрочной группы ячеек с точками. При протягивании маркера в любом допустимом направлении копируются точки, причем каждая компонента точки преобразуется как отдельное число (см. ii);

viii. копирование из активной двухстолбцевой группы ячеек с точками. При протягивании маркера в любом допустимом направлении копируются точки, причем каждая компонента точки преобразуется как отдельное число (см. iii);

ix. копирование из активной группы ячеек при наличии в ней ячеек с текстами;

x. копирование из активной группы ячеек при наличии в ней формул, содержащих адреса некоторых ячеек. Здесь в ячейке $E3$ находится формула “ $=A1+3$ ” с относительным адресом $A1$. При протягивании маркера вниз и вправо показано, как меняется эта формула (верхняя строка ячеек) и выводится вычисленное по ней значение (нижняя строка ячеек). В ячейке $E6$ оказывается формула “ $=A3+3$ ”. Относительно значения, вычисляемого по ней, можно сказать следующее. Если копирование влево действительно производилось, то в $A3$ окажется значение “текст” и по формуле “ $=A3+3$ ” будет вычислено “текст3” – соединение текста и числа. Если копирование влево не производилось, то $A3$ останется не определенным и будет выведен символ “?”.

На рис. 7a-7b приведены дополнительные примеры преобразования относительных адресов в формулах при копировании содержимого ячеек протаскиванием маркера в различных допустимых направлениях. Ниже формул показаны вычисленные по ним значения. В обоих случаях перед протаскиванием маркера должны быть сформированы числовые ячейки $A2, B2, C2, D2, E2, B3, B4$ и ячейка $F5$ с формулой $B2+3$. Маркер протягивается у активной ячейки $F5$. Обратите внимание, что для точек вместо $(a, b)+(c, c)$ можно писать $(a, b)+c$.

	A	B	C	D	E	F	G	H
1		?				B2+3 4		
2	A2 10	B2 1	C2 2	D2 5	E2 4	B2+3 4		
3		B3 3				B2+3 4		
4		B4 9				B1+3 ?		
5	B2+3 4	B2+3 4	B2+3 4	B2+3 4	B2+3 4	B2+3 4	C2+3 5	D2+3 8
6						B3+3 6		
7						B4+3 12		

Рис. 7а. Копирование формулы “=B2+3” с относительным адресом ячейки

	A	B	C	D	E	F	G	H
1		(?,?)				B2+3 (4,6)		
2	A2 (10,1)	B2 (1,3)	C2 (2,5)	D2 (5,2)	E2 (4,4)	B2+3 (4,6)		
3		B3 (3,0)				B2+3 (4,6)		
4		B4 (9,7)				B1+3 (?,?)		
5	B2+3 (4,6)	B2+3 (4,6)	B2+3 (4,6)	B2+3 (4,6)	B2+3 (4,6)	B2+3 (4,6)	C2+3 (5,8)	D2+3 (8,5)
6						B3+3 (6,3)		
7						B4+3 (12,10)		

Рис. 7б. Копирование формулы “=B2+3” с относительным адресом ячейки

10.10. Команды для работы с таблицами

Приведем некоторые команды по обустройству интерфейса, связанные со строкой ввода команд и справки по командам

A. <i>RClick(таблица)\Настройки таблицы\Показывать строку ввода</i> (✓)
B. <i>Вид\Строка ввода</i>
C. <i>Click</i> (?)

Индикация строки ввода команд и справки по командам

А) По команде А, выполняемой через контекстное меню, реализуется установка, в соответствии с которой строка ввода команд (*Input Bar*) оказывается всегда видимой, начиная с запуска системы. Появляется она с правой стороны внизу главного окна системы.

В) Команда *B*, выполняемая через основное меню, работает как двухпозиционный переключатель, вставляя (✓) в главное окно системы и удаляя из него строку ввода команд.

С) Команда *C* выполняется щелчком по кнопке “?”, находящейся справа от строки ввода. Эта кнопка работает как двухпозиционный переключатель, вставляя в главное окно системы и удаляя из него панель “Список команд”.

Ниже рассматриваются некоторые команды, используемые при работе с электронными таблицами. Они могут содержать один или два аргумента и записываются так:

Имя_команды[*аргумент*] или *Имя_команды*[*аргумент1*, *аргумент2*].

A. <i>Row</i> [ячейка]	E. <i>FillColumn</i> [№ столбца, список]
B. <i>Column</i> [ячейка]	F. <i>FillRow</i> [№ строки, список]
C. <i>ColumnName</i> [ячейка]	G. <i>CellRange</i> [ячейка1, ячейка2]
D. ζ = <i>Cell</i> [№ столбца, № строки]	

Команды для работы с электронными таблицами (1)

- А) Команда *A* возвращает номер строки ячейки-аргумента.
- В) Команда *B* возвращает номер столбца ячейки-аргумента.
- С) Команда *C* возвращает имя столбца ячейки-аргумента.

Замечание. Если *A*, *B* или *C* используется как отдельное выражение, то результат вычислений присваивается переменной, имя которой выбирает система (*a*, *b*, ...). Команды *A*, *B* и *C* можно задавать в виде *label*=*Row*[ячейка], где *label* – имя пользовательской переменной (*xx*, *pqr*, ...) или имя какой-либо ячейки (*A5*, *W7*, ...). В этом случае результат выполнения команды становится значением переменной *label* или ячейки.

Д) Команда *D* копирует содержимое ячейки, заданной аргументами в новую ячейку ζ . Например, по команде *A5=Cell*[2,10] содержимое ячейки *B10* будет скопировано в ячейку *A5*.

Е) По команде *E* ячейки столбца с данным номером, начиная с первой строки заполняются последовательными элементами списка.

Ф) По команде *F* ячейки строки с данным номером, начиная с первого столбца заполняются последовательными элементами списка.

Г) По команде *G* возвращается список с данными из диапазона ячеек, расположенных между ячейками с именами “ячейка1” и “ячейка 2” (без кавычек, см. замечание к *A*). В список данные попадают из последовательных столбцов диапазона, начиная с его начального столбца.

H. <i>FillCells</i> [ячейка, список]
I. <i>FillCells</i> [ячейка, матрица]
J. <i>FillCells</i> [диапазон ячеек, объект]

Команды для работы с электронными таблицами (2)

Н) Команда *H* заполняет ячейки таблицы последовательными элементами списка “список”. Заполнение начинается от ячейки “ячейка” и далее последовательно по строке направо по одному элементу на ячейку.


Л) Команда *I* заполняет ячейки таблицы элементами матрицы “матрица” по одному элементу на ячейку. Если размеры матрицы есть $n \times m$, то матрица записывается как есть в группу ячеек, у которой n строк и m столбцов, а в левом верхнем углу находится ячейка “ячейка”.


М) Команда *J* размещает в диапазоне ячеек “диапазон ячеек” указанный объект “объект”. Этот объект может быть любым. Приведем пример:

- по команде *FillCells*[A1:A1, Circle[(0,1), 2]] – в ячейку A1 помещается уравнение окружности $x^2+(y-1)^2=4$;
- по команде *FillCells*[A15:G15, RandomBetween[0,8]] последовательные ячейки строки от A15 и до G15 заполняются генерируемыми псевдослучайными целыми числами от 0 до 7 при равномерном распределении;
- по команде *FillCells*[A17:D19, RandomUniform[0,8,12]] ячейки диапазона A17:D19 заполняются 12 генерируемыми псевдослучайными действительными числами из промежутка [0,8) при равномерном распределении.

10.11. Создание некоторых объектов

В этом разделе описываются инструменты и команды для создания некоторых объектов из наборов данных, размещенных в выделенных диапазонах ячеек электронных таблиц. Речь идет о таких объектах, как список, список точек, матрица, таблица или ломаная.

Инструмент “ Создать список” позволяет по данным из диапазона ячеек электронной таблицы сформировать на панели объектов именованный список. Для этого необходимо:

- протяжкой курсора мыши или иным способом выделить диапазон ячеек. В нем могут быть и пустые ячейки, но в формировании списка они принимать участия не будут;
- инструментом “ Создать список” вывести одноименную панель, представленную на рис. 8. На ней можно задать имя списка и тип выборки данных в список из выделенного диапазона по строкам или по столбцам. Кроме того, с помощью радиокнопок создаваемый список можно сделать свободным или зависимым объектом. В первом случае он не будет связан с содержимым ячеек выделенного диапазона, а во втором – автоматически будет актуализироваться при любых изменениях данных в ячейках этого диапазона. При щелчке по кнопке “Create” панель “Создать список” закрывается и новый список с учетом сделанных установок создан.

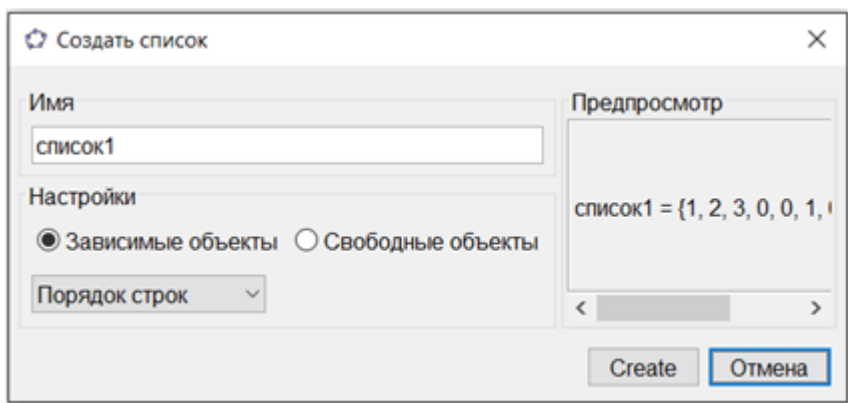




Рис. 8. Панель для установки свойств создаваемого списка

Инструмент “ Создать список точек” позволяет по данным из диапазона ячеек электронной таблицы сформировать на панели объектов именованный список точек. Для этого необходимо:

- протяжкой курсора мыши или иным способом выделить диапазон ячеек с двумя столбцами или двумя строками. Если выделен диапазон ячеек с двумя столбцами, то компоненты последовательных точек списка будут формироваться из элементов последовательных строк диапазона. Если выделен диапазон ячеек с двумя строками, то компоненты последовательных точек списка будут формироваться из элементов последовательных столбцов диапазона. В любом случае строки (столбцы) с одной или двумя пустыми ячейками допустимы, но в формировании точек они принимать участия не будут;
- инструментом “ Создать список точек” вывести одноименную панель, представленную на рис. 9. На ней можно задать имя списка и порядок выборки данных для точек типа $X \rightarrow Y$ – сначала абсцисса, потом ордината или $Y \rightarrow X$ – сначала ордината, потом абсцисса. Кроме того, с помощью радиокнопок создаваемый список можно сделать свободным или зависимым объектом. При щелчке по кнопке “Create” панель “Создать список точек” закрывается и новый список точек с учетом сделанных установок создан.

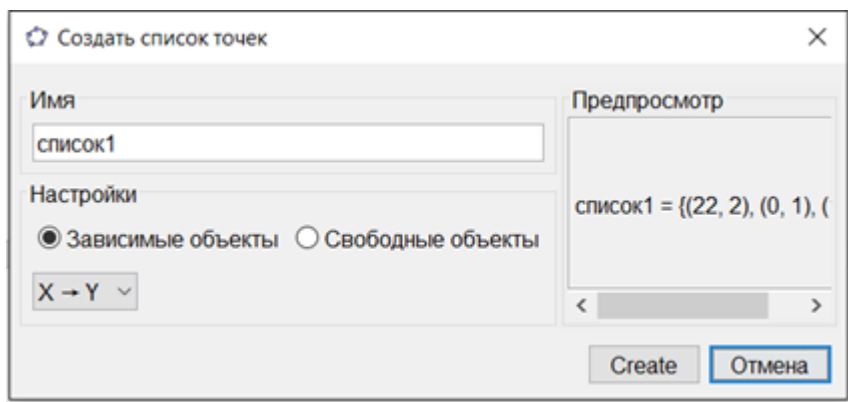


Рис. 9. Панель для установки свойств создаваемого списка

Замечание. Для создания точек в пространстве, то есть с тремя компонентами, необходимо выделять диапазон ячеек с тремя столбцами (строками).

Точки будут формироваться по данным последовательных строк (столбцов) диапазона. При этом абсцисса, ордината и аппликата точек всегда будут выбираться соответственно из 1, 2 и 3 столбца (строки).

Инструмент “^{1 2}_{3 4} Создать матрицу” позволяет по данным из диапазона непустых ячеек электронной таблицы сформировать на панели объектов именованную матрицу. Для этого необходимо:

- протяжкой курсора мыши или иным способом выделить диапазон непустых ячеек;
- инструментом “^{1 2}_{3 4} Создать матрицу” вывести одноименную панель, представленную на рис. 10. На ней можно задать имя матрицы и с помощью радиокнопок сделать матрицу свободным или зависимым объектом. Включением независимого переключателя можно предусмотреть создание транспонированной матрицы. Если все ячейки выделенного диапазона были числовыми, то матрица выводится окаймленной круглыми скобками, что можно видеть в окне предпросмотра. Если в диапазоне имеются текстовые ячейки, то вывод реализуется в виде вложенного списка. Если, например, первый элемент второй строки матрицы рис. 10 заменить на “текст”, то матрица будет выведена в виде $\{\{2,3,0\}, \{\text{“hhh”},0,2\}, \{1,2,1\}, \{33,44,55\}\}$. При щелчке по кнопке “Create” панель “Создать матрицу” закрывается и новая матрица с учетом сделанных установок создана.

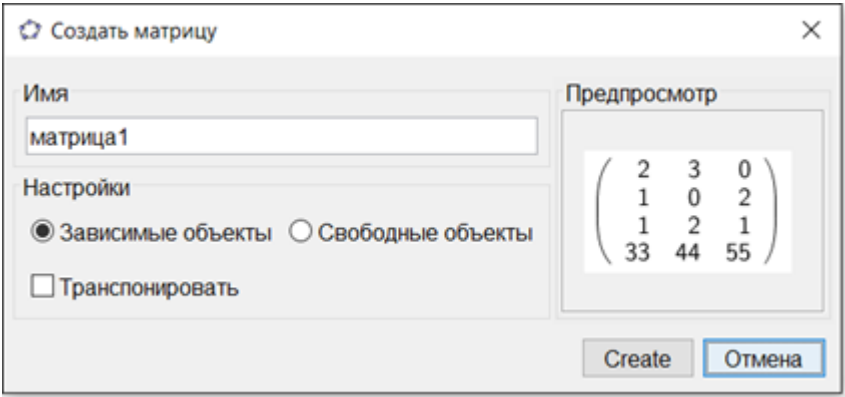


Рис. 10. Панель для установки свойств создаваемой матрицы

Инструмент “^{1 2}_{3 4} Создать таблицу” позволяет по данным из диапазона непустых ячеек электронной таблицы сформировать на панели “Полотно” именованную таблицу. Для этого необходимо:

- протяжкой курсора мыши или иным способом выделить диапазон непустых ячеек;
- инструментом “^{1 2}_{3 4} Создать таблицу” вывести одноименную панель, представленную на рис. 11. На ней можно задать имя таблицы и вывод ее или в виде выделенного диапазона (см. окно “Предпросмотр” рис. 11), или в транспонированном виде. Кроме того, радиокнопками создаваемую таблицу можно

сделать свободным или зависимым объектом. При щелчке по кнопке “*Create*” панель “*Создать таблицу*” закрывается и новая таблица с учетом сделанных установок создана. Появляется она в левом верхнем углу панели “*Полотно*” и затем ее можно перетащить в любую другую позицию.

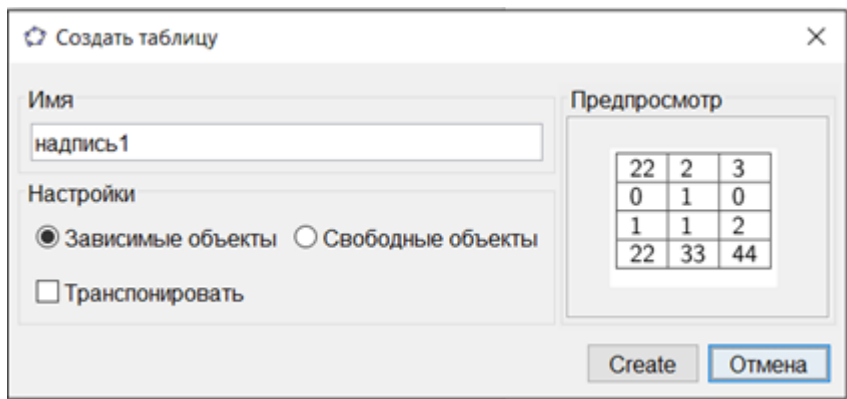




Рис. 11. Панель для установки свойств создаваемой таблицы

Инструмент “ *Создать ломаную*” позволяет по данным из диапазона ячеек электронной таблицы сформировать ломаную с выводом ее на панель “*Полотно*”. Для этого необходимо:

- протяжкой курсора мыши или иным способом выделить диапазон ячеек с двумя столбцами или двумя строками. Если выделен диапазон ячеек с двумя столбцами, то позиции узлов ломаной (точек) будут формироваться из элементов последовательных строк выделенного диапазона. Если выделен диапазон ячеек с двумя строками, то позиции узлов ломаной будут формироваться из элементов последовательных столбцов выделенного диапазона. В любом случае строки (столбцы) с одной или двумя пустыми ячейками допустимы, но в формировании точек они принимать участия не будут;
- инструментом “ *Создать ломаную*” вывести одноименную панель, представленную на рис. 12. На ней можно задать имя ломаной и порядок выборки данных для позиций ее узлов типа: $X \rightarrow Y$ – сначала абсцисса, потом ордината или $Y \rightarrow X$ – сначала ордината, потом абсцисса. Кроме того, с помощью радиокнопок создаваемую ломаную можно сделать свободным или зависимым объектом. При щелчке по кнопке “*Create*” панель “*Создать ломаную*” закрывается и новая ломаная с учетом сделанных установок создана.

Что касается команд для создания перечисленных выше объектов, то все они ранее уже были рассмотрены в соответствующих разделах.

Замечание. Для создания ломаной в пространстве, то есть по точкам с тремя компонентами, необходимо выделять диапазон из трех столбцов (или трех строк). В этом случае позиции для узлов ломаной будут формироваться по данным последовательных строк (столбцов) диапазона. При этом абсцисса, ордината и аппликата позиций всегда будут выбираться соответственно из 1, 2 и 3 столбца (строки).

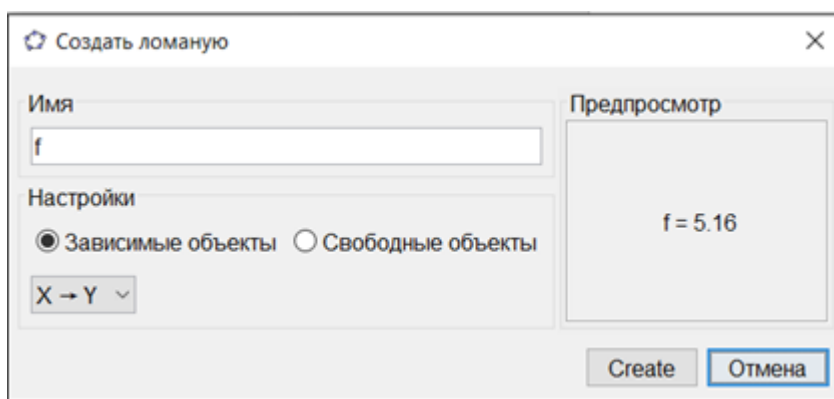


Рис. 12. Панель для установки свойств создаваемой ломаной

10.12. Формирование характеристик элементов таблицы

В этом разделе описываются инструменты и команды для формирования простейших характеристик для наборов данных, размещенных в выделенных диапазонах ячеек электронных таблиц. Речь идет о таких характеристиках как “сумма”, “среднее арифметическое”, “количество элементов”, “максимальный элемент” и “минимальный элемент”.

Инструмент “ Σ Сумма” для выделенного диапазона ячеек позволяет определить сумму их значений или суммы значений ячеек каждого столбца или суммы значений ячеек каждой строки. Здесь ячейки должны быть числовыми или пустыми. При вычислениях значения в пустых ячейках диапазона считаются равными нулю. При наличии в диапазоне нечисловых ячеек появляется сообщение об ошибке. Рассмотрим два возможных случая:

1. Для нахождения суммы значений ячеек выделенного диапазона требуется:


- щелчком левой кнопкой мыши по ячейке указать, где должен размещаться результат вычислений;
- щелчком левой кнопкой мыши активировать инструмент “ Σ Сумма”. При этом граница ячейки результата становится точечной, а в ячейке появляется знак вопроса;
- протяжкой курсора мыши или иным способом выделить диапазон ячеек.


В результате этих действий сумма значений ячеек указанного диапазона помещается в ячейку результата.





Заметим, что в данном случае инструмент “ Σ Сумма” действует как команда $\lambda = \text{Sum}[\alpha:\beta]$, где $\alpha:\beta$ – диапазон ячеек, λ – адрес ячейки результата.


2. Для нахождения суммы значений ячеек по столбцам или строкам выделенного диапазона требуется:

- протяжкой курсора мыши или иным способом выделить диапазон более чем из одной ячейки;

- щелчком левой кнопки мыши активировать инструмент “ Сумма”.


В результате этих действий возможны такие случаи. Если в диапазоне несколько строк, то вычисляются суммы значений ячеек каждого столбца и результаты появляются в ячейках ниже соответствующих столбцов. Если в диапазоне одна строка, то вычисляется сумма значений ее ячеек и результат появляется в ячейке справа от строки. Если при щелчке по кнопке “ ” держать нажатой клавишу *Shift*, то будут отыскиваться суммы значений ячеек каждой строки выделенного диапазона, а результаты будут размещаться справа от этих строк.

Инструмент “ Среднее арифметическое” для выделенного диапазона числовых ячеек позволяет определить среднее арифметическое их значений или средние арифметические значения ячеек каждого столбца, или средние арифметические значения ячеек каждой строки. Выполняется данный инструмент аналогично инструменту “ Сумма”. Описание действия инструмента “ Среднее арифметическое” получается из описания действия инструмента “ Сумма” простой заменой в нем слов “сумма” и “суммы” соответственно на “среднее арифметическое” и “средние арифметические”.

Инструмент “ Подсчитать количество ячеек” позволяет определить количество непустых ячеек выделенного диапазона. Причем ячейки не обязаны быть числовыми. Рассмотрим два возможных случая:

1. Для определения количества непустых ячеек выделенного диапазона требуется:

- щелчком левой кнопкой мыши по ячейке указать, где должен размещаться результат вычислений;


- щелчком левой кнопкой мыши активировать инструмент “ Подсчитать количество ячеек”. При этом граница ячейки результата становится точечной, а в ячейке появляется знак вопроса;

- протяжкой курсора мыши или иным способом выделить диапазон ячеек.


В результате этих действий подсчитывается количество непустых ячеек указанного диапазона и помещается в ячейку результата.

2. Для определения количества непустых ячеек по столбцам или строкам выделенного диапазона требуется:

- протяжкой курсора мыши или иным способом выделить диапазон более чем из одной ячейки;


- щелчком левой кнопки мыши активировать инструмент “ Подсчитать количество ячеек”.

В результате этих действий возможны такие случаи. Если в диапазоне несколько строк, то вычисляется количество непустых ячеек каждого столбца и результаты появляются в ячейках ниже соответствующих столбцов. Если в диапазоне одна строка, то вычисляется количество ее непустых ячеек и результат появляется в ячейке справа от строки. Если при выполнении команды держать нажатой клавишу *Shift*, то будет находиться количество непустых ячеек каждой строки выделенного диапазона и результаты размещаться справа от этих строк.

Инструмент “ Максимум” позволяет определить максимальное значение в ячейках выделенного диапазона или максимальные значения в ячейках каждого столбца или каждой строки этого диапазона. Рассмотрим два возможных случая.


1. Для нахождения наибольшего значения ячеек выделенного диапазона требуется:

- щелчком левой кнопкой мыши по ячейке указать, где должен размещаться результат вычислений;

- щелчком левой кнопкой мыши активировать инструмент “ Максимум”. При этом граница ячейки результата становится точечной, а в ячейке появляется знак вопроса;


- протяжкой курсора мыши или иным способом выделить диапазон ячеек.

В результате этих действий максимальное значение из ячеек указанного диапазона дублируется в ячейку результата. При наличии в диапазоне нечисловых ячеек в ячейку результата помещается знак “?”.

Заметим, что в данном случае инструмент “ ” действует как команда $\lambda = \text{Max}[\alpha:\beta]$, где $\alpha:\beta$ – диапазон ячеек, λ – адрес ячейки результата.





2. Для нахождения наибольшего значения ячеек в столбцах или в строках выделенного диапазона требуется:

- протяжкой курсора мыши или иным способом выделить диапазон более чем из одной ячейки;

- щелчком левой кнопки мыши активировать инструмент “ Максимум”.





В результате этих действий возможны такие случаи. Если в диапазоне несколько строк, то вычисляются максимальные значения ячеек каждого столбца и результат появляется в ячейках ниже соответствующих столбцов. Если в диапазоне одна строка, то вычисляется максимальное значение из ее ячеек и


результат появляется в ячейке справа от строки. Если при выполнении команды держать нажатой клавишу *Shift*, то будут отыскиваться максимальные значения ячеек каждой строки выделенного диапазона, а максимумы по строкам будут размещаться справа от этих строк. При наличии в столбце (строке) нечисловых ячеек в соответствующую ячейку результата помещается знак “?”.





Инструмент “ Минимум” позволяет определить минимальное значение в ячейках выделенного диапазона или минимальные значения в ячейках каждого столбца или каждой строки этого диапазона. Выполняется данный инструмент аналогично инструменту “ Максимум”. Описание действия инструмента “ Минимум” получается из описания действия инструмента “ Максимум” простой заменой в нем слов “максимум”, “наибольшее” и “наибольшие” соответственно на “минимум”, “наименьшее” и “наименьшие”.

Пример 1. Числовая таблица размещена в диапазоне ячеек *C3:F5*. Сформировать в ячейках *A1:A4* соответствующими инструментами сумму значений, среднее значение, максимальное значение и минимальное значение элементов *C3:F5*.

Решение. Последовательность действий для решения данного примера может быть такой:

- сформируем исходные данные в диапазоне ячеек *C3:F5* (см. рис. 13);
- щелкаем левой кнопкой мыши по ячейке *A1*, активируем инструмент “ Сумма” и протяжкой курсора мыши выделяем в таблице диапазон ячеек *C3:F5*. В ячейке *A1* оказывается сформированной сумма значений ячеек указанного диапазона;
- щелкаем левой кнопкой мыши по ячейке *A2*, активируем инструмент “ Среднее арифметическое” и протяжкой курсора мыши выделяем в таблице диапазон ячеек *C3:F5*. В ячейке *A2* оказывается сформированным среднее арифметическое значений ячеек указанного диапазона;
- щелкаем левой кнопкой мыши по ячейке *A3*, активируем инструмент “ Максимум” и протяжкой курсора мыши выделяем в таблице диапазон ячеек *C3:F5*. В ячейке *A3* оказывается сформированным наибольшее из значений ячеек указанного диапазона;
- щелкаем левой кнопкой мыши по ячейке *A4*, активируем инструмент “ Минимум” и протяжкой курсора мыши выделяем в таблице диапазон ячеек *C3:F5*. В ячейке *A4* оказывается сформированным наименьшее из значений ячеек указанного диапазона;
- через панель управления вводом и редактированием данных сформируем окаймление исходной таблицы и светло-серый фон ячеек результата *A1:A4* (см. рис. 13).

- активируем инструмент “ Сумма” и в ячейке G6 получаем “сумму сумм по строкам”, то есть сумму значений ячеек всей таблицы;
- через панель управления вводом и редактированием данных сформируем светло-серый фон ячеек результата G3:G6 и C6:F6 (см. рис. 2).

Завершая описание работы с электронными таблицами, отметим, что в *GeoGebra* наиболее значимое их применение связано с темами “Вероятность” и “Статистика” при работе с инструментами: “ Анализ одной переменной”, “ Регрессионный анализ”, “ Анализ нескольких переменных” и “ Калькулятор вероятностей”. И эти вопросы рассмотрены отдельно во второй части пособия.

Приложение.

Встроенные (предопределенные) математические функции GeoGebra

Константа, Функция	Описание
<i>Константы</i>	
e	Основание натуральных логарифмов (ввод: $Alt+e$). $e=2.7182818284590452353602874713527...$
π	Число π (ввод имени: π , ввод значения $Alt+p$). $\pi=3,1415926535897932384626433832795...$
i	Мнимая единица: $\sqrt{-1}$ (ввод имени: i или $Alt+i$)
<i>Часто используемые функции</i>	
random()	Псевдослучайное число. Генерируется псевдослучайное число a между 0 и 1 ($0<a<1$) при равномерном распределении
abs(x)	Абсолютное значение или модуль. При действительном x возвращает абсолютное значение, при комплексном $x=a+bi$ – модуль числа, то есть $\sqrt{a^2 + b^2}$. Примеры: $abs(-5) \rightarrow 5$, $abs(5+3i) \rightarrow 5.83$, $sqrt(5+3i) \rightarrow 5.83$
sqrt(x)	Корень квадратный. Возвращается главное значение корня квадратного из действительного числа x ($x\geq0$) или комплексного числа $x=a+bi$. Примеры: $sqrt(4) \rightarrow 2$, $sqrt(1+i) \rightarrow 1.1+0.46i$
cbrt(x)	Корень кубический. Возвращается главное значение корня кубического из действительного или комплексного числа x . Примеры: $cbrt(27) \rightarrow 3$, $cbrt(-27) \rightarrow -3$, $cbrt(3+2i) \rightarrow 1.5+0.3i$
nroot(x,n)	Корень n -ой степени из числа x . Здесь: x – действительное или комплексное число, n – действительное число. $\sqrt[n]{x}=nroot(x,n)$ – это (главное) решение t уравнения $t^n=x$. Примеры: $nroot(16,2) \rightarrow 4$, $nroot(16,-2) \rightarrow 0.25$, $nroot(16+i,2) \rightarrow 4+0.12i$, $nroot(16+i,5) \rightarrow 1.744+0.02i$
sign(x) или sgn(x)	Знак действительного числа. Функция знака определяется так: $sign(x) = \begin{cases} 1, & \text{если } x > 0; \\ 0, & \text{если } x=0; \\ -1, & \text{если } x<0. \end{cases}$ Примеры: $sign(5)=5$, $sign(-3)=1$, $sign(0)=0$

$\text{arg}(x)$	<p><i>Аргумент.</i> Для комплексного числа x ($x \neq 0+0i$) возвращается главное значение угла φ в градусах, то есть наименьший по модулю угол, на который необходимо повернуть ось абсцисс вокруг начала координат до встречи с радиус-вектором точки x. Поворот оси абсцисс в направлении, противоположном движению часовой стрелки, считается положительным, по часовой стрелке – отрицательным. Угол φ измеряется в градусах, $-180^\circ < \varphi \leq 180^\circ$ и $\text{arg}(0+0i)=0^\circ$, $\text{arg}(a)=0$ ($a \geq 0$), $\text{arg}(a)=-\pi$ ($a < 0$). Для точек $\text{arg}(x)$ определяется так: $\text{arg}((a,b))=\text{arg}((a,b,c))=\text{arg}(a+bi)$. <i>Примеры:</i> $\text{arg}(1+i)=45^\circ$, $\text{arg}((1,1))=45^\circ$, $\text{arg}(-7)=-\pi$</p>
$\text{floor}(x)$	<p><i>Пол.</i> Для действительных x функция $\text{floor}(x)$ возвращает наибольшее целое, меньшее или равное x. Если $x=a+bi$, то $\text{floor}(x)=\text{floor}(a)+i \cdot \text{floor}(b)$. <i>Примеры:</i> $\text{floor}(5)=5$, $\text{floor}(5.9)=5$, $\text{floor}(-5.9)=-6$</p>
$\text{ceil}(x)$	<p><i>Потолок.</i> Для действительных x функция $\text{ceil}(x)$ возвращает наименьшее целое, большее или равное x. Если $x=a+bi$, то $\text{ceil}(x)=\text{ceil}(a)+i \cdot \text{ceil}(b)$. <i>Примеры:</i> $\text{ceil}(5)=5$, $\text{ceil}(5.9)=6$, $\text{ceil}(-5.9)=-5$</p>
$\text{round}(x)$	<p><i>Округление.</i> Если выражение x имеет действительное значение, то возвращается его округление до целых чисел. При неоднозначности результата округление проводится вправо, то есть в сторону больших значений. Если $x=a+bi$, то $\text{round}(x)=\text{round}(a)+i \cdot \text{round}(b)$. <i>Примеры:</i> $\text{round}(6)=6$, $\text{round}(6.3)=3$, $\text{round}(6,5)=7$, $\text{round}(-6,5)=-6$, $\text{round}(7.5)=8$</p>
$\text{fractional-Part}(x)$	<p><i>Дробная часть.</i> При целых действительных x функция равна 0. При $x=\alpha.\beta$, где α, β – действительные числа, функция возвращает β со знаком x. При $x=\alpha.\beta+\gamma$, где α, β и γ – действительные числа, функция возвращает β, если $\alpha.\beta > 0$; $1-\beta$, если $\alpha.\beta < 0$; 0, если $\alpha.\beta = 0$. <i>Примеры:</i> $\text{fractionalPart}(7)=0$, $\text{fractionalPart}(7.3)=0.3$, $\text{fractionalPart}(-7.3)=-0.3$, $\text{fractionalPart}(-7.3+2i)=0.7$</p>
$\text{real}(x)$	<p><i>Действительная часть.</i> Для действительного или комплексного числа x возвращается его действительная часть. <i>Примеры:</i> $\text{real}(5)=5$, $\text{real}(0)=0$, $\text{real}(-3+i)=-3$</p>
$\text{imaginary}(x)$	<p><i>Комплексная часть.</i> Для действительного или комплексного числа x возвращается действительный множитель при комплексной единице. <i>Примеры:</i> $\text{imaginary}(5)=0$, $\text{imaginary}(0)=0$, $\text{imaginary}(-3+i)=1$</p>
$\text{conjugate}(x)$	<p><i>Сопряженное число.</i> Для действительного числа x возвращается x, для комплексного числа $x=a+bi$ возвращается сопряженное число $a-bi$.</p>

	<i>Примеры: conjugate(5)=5, conjugate(2-3i)=2+3i</i>
a) atan2(y,x) b) atan2d(y,x)	<i>Артангенс2. Здесь: y и x – действительные числа. Возвращается наименьший угол α в радианах ($-\pi < \alpha \leq \pi$) или в градусах ($-180^\circ < \alpha \leq 180^\circ$) между осью абсцисс и вектором с началом в точке (0,0) и концом в точке (x,y). Знак результата показывает направление отсчета угла: “+” – против часовой стрелки, “-” – по часовой стрелке. Имеют место соотношения: $\text{atan2}(y,x) = \arg(x+iy) \cdot \pi/180$. Примеры: 1) $\text{atan2}(2,1)=1.11$, $\arg(1+i*2) \cdot \pi/180=1.11$, $\text{atan}(2/1)=1.11$ 2) $\text{atan2}(-2,-1)=-2.03$, $\arg(-1-2*i) \cdot \pi/180=-2.03$, $\text{atan}(-2/1)=1.11$</i>
x[P]	<i>Абсцисса точки P. Примеры: x[(2,1)]=2, x[(2,1,7)]=2</i>
y[P]	<i>Ордината точки P. Примеры: y[(2,1)]=1, y[(2,1,7)]=1</i>
z[P]	<i>Аппликата точки P. Примеры: z[(2,1)]=0, y[(2,1,7)]=7</i>
<i>Экспонента и логарифмы</i>	
exp(x)	<i>Экспонента. Аргумент x может быть любым комплексным числом</i>
ln(x)	<i>Логарифм. Натуральный логарифм (по основанию e). Аргумент $x \neq 0$ может быть комплексным. Для комплексного числа x: $\ln(x) = \ln(\text{abs}(x)) + i * \arg(x)$ ($-\pi < \arg(z) \leq \pi$). Это же относится и к lg(x) и log(b,x)</i>
lg(x)	<i>Логарифм. Десятичный логарифм (по основанию 10)</i>
log(b,x)	<i>Логарифм. Логарифм по основанию b ($x \neq 0$, $b > 0$, $b \neq 1$)</i>
ld(x)	<i>Логарифм. Логарифм x по основанию 2. Число $x \neq 0$ может быть только действительным</i>
<i>Тригонометрические функции</i>	
sin(x)	<i>Синус. По умолчанию аргумент x измеряется в радианах и может иметь комплексное значение. Это же утверждение относится и к другим тригонометрическим функциям cos(x), tg(x), ctg(x), sec(x), cosec(x). Примеры: $\sin(0)=0$, $\sin(1+i)=1.3+0.63i$, $\sin(\pi/2)=1$</i>
cos(x)	<i>Косинус. Примеры: $\cos(0)=1$, $\cos(2+i)=-0.64+1.07i$</i>
tg(x)	<i>Тангенс. Здесь $x \neq \pi/2 + \pi k$ ($k=0, \pm 1, \pm 2, \dots$). Английская запись – tan(x). Примеры: $\text{tg}(1)=1.56$, $\text{tg}(3+i)=-0.06+0.77i$</i>
ctg(x)	<i>Котангенс. Здесь $x \neq \pi k$ ($k=0, \pm 1, \pm 2, \dots$). Английская запись – cot(x). Примеры: $\text{ctg}(1)=0.64$, $\text{ctg}(3-i)=-0.01+1.29i$</i>
sec(x)	<i>Секанс. Здесь $x \neq \pi/2 + \pi k$ ($k=0, \pm 1, \pm 2, \dots$).</i>
cosec(x)	<i>Косеканс. Здесь $x \neq \pi k$ ($k=0, \pm 1, \pm 2, \dots$)</i>
<i>Гиперболические функции</i>	

$\text{sh}(x)$	<i>Гиперболический синус: $\text{sh}(x)=(e^x-e^{-x})/2$. Аргумент x – действительное или комплексное число. Английская запись – $\sinh(x)$.</i>
$\text{ch}(x)$	<i>Гиперболический косинус: $\text{ch}(x)=(e^x+e^{-x})/2$. Аргумент x – действительное или комплексное число. Английская запись – $\cosh(x)$.</i>
$\text{th}(x)$	<i>Гиперболический тангенс: $\text{th}(x)=(e^x-e^{-x})/(e^x+e^{-x})$. Аргумент x – действительное или комплексное число. Английская запись – $\tanh(x)$.</i>
$\text{cth}(x)$	<i>Гиперболический котангенс: $(e^x+e^{-x})/(e^x-e^{-x})$. Аргумент x – действительное или комплексное число. Английская запись – $\coth(x)$.</i>
$\text{sech}(x)$	<i>Гиперболический секанс: $\text{sech}(x)=2/(e^x+e^{-x})$. Аргумент x – действительное или комплексное число.</i>
$\text{cosech}(x)$	<i>Гиперболический косеканс: $\text{cosech}(x)=2/(e^x-e^{-x})$. Аргумент x – действительное или комплексное число.</i>
<i>Обратные тригонометрические и гиперболические функции</i>	
$\arcsin(x)$	<i>Арксинус. Аргумент x ($x \leq 1$) и результат – действительные числа. Английская нотация: $\text{asin}(x)$ – для результата в радианах, и $\text{asind}(x)$ – для результата в градусах.</i>
$\arccos(x)$	<i>Арккосинус. Аргумент x ($x \leq 1$) и результат – действительные числа. Английская нотация: $\text{acos}(x)$ – для результата в радианах, и $\text{acosd}(x)$ – для результата в градусах.</i>
$\arctg(x)$	<i>Арктангенс. Аргумент x и результат – действительные числа. Английская нотация: $\text{atan}(x)$ – для результата в радианах, и $\text{atand}(x)$ – для результата в градусах.</i>
$\text{arcsh}(x)$	<i>Арксинус гиперболический: $\text{arcsh}(x)=\ln\left(x+\sqrt{x^2+1}\right)$. Аргумент x – действительное число. Английская нотация – $\text{asinh}(x)$</i>
$\text{arcch}(x)$	<i>Арккосинус гиперболический: $\text{arcch}(x)=\ln\left(x+\sqrt{x^2-1}\right)$ ($x \geq 1$). Аргумент x – действительное число. Английская нотация – $\text{acosh}(x)$.</i>
$\text{arcth}(x)$	<i>Арктангенс гиперболический: $\text{arcth}(x)=\frac{1}{2}\ln\left(\frac{1+x}{1-x}\right)$. Аргумент x – действительное число. Английская нотация – $\text{atanh}(x)$.</i>
<i>Разные функции</i>	
$n!$	Факториал целого неотрицательного n

gamma(z)	<p><i>Гамма-функция.</i> По определению $\text{gamma}(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$ - несобственный интеграл, зависящий от комплексного параметра z ($z \neq 0, -1, -2, \dots$). Эту функцию называют также интегралом Эйлера второго рода. Для целых неотрицательных z: $\text{gamma}(z+1)=z!$. В математике обычное обозначение $\text{gamma}(z) - \Gamma(z)$.</p>
gamma(a,z)	<p><i>Неполная гамма-функция.</i> По определению $\text{gamma}(a,z) = \int_a^{\infty} t^{z-1} e^{-t} dt$ ($a \geq 0$). В частности, $\text{gamma}(0,z)=\text{gamma}(z)$</p>
gamma-Regularized(a,z)	<p><i>Нормализованная неполная гамма-функция.</i> По определению $\text{gammaRegularized}(a,z) = \frac{\text{gamma}(a,z)}{\text{gamma}(a)}$</p>
polyGamma(m,z)	<p><i>Полигамма-функция.</i> По определению полигамма-функция есть $(m+1)$ производная от натурального логарифма гамма-функции, то есть</p> $\text{polyGamma}(m,z) = \frac{d^{m+1}}{dx^{m+1}} \ln \left(\int_0^{\infty} t^{z-1} e^{-t} dt \right) \quad (m=0,1, \dots)$
psi(z)	<p><i>Дигамма-функция.</i> Эта функция определяется так: $\text{psi}(z) = \frac{d}{dx} \ln(\text{gamma}(z))$, то есть $\text{psi}(z)$ является частным случаем полигамма-функции при $m=0$</p>
erf(x)	<p><i>Гауссова функция ошибок.</i> По определению для действительных x: $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$</p>
beta(a,b)	<p><i>Бета-функция.</i> По определению</p> $\text{beta}(a,b) = \int_0^1 t^{a-1} \cdot (1-t)^{b-1} dt \quad (\text{real}(a) > 0, \text{real}(b) > 0) -$ <p>несобственный интеграл, зависящий от двух комплексных параметров a и b. Эту функцию называют также интегралом Эйлера первого рода. В математике ее обычное обозначение – $B(a,b)$.</p>
beta(a,b,x)	<p><i>Неполная beta-функция.</i> По определению</p> $\text{beta}(a,b,x) = \int_0^x t^{a-1} \cdot (1-t)^{b-1} dt \quad (0 \leq x \leq 1).$ <p>В частности, $\text{beta}(a,b,1)=\text{beta}(a,b)$</p>

betaRegularized(a,b,x)	<p>Нормализованная неполная бета-функция. По определению $\text{betaRegularized}(a,b,x) = \frac{\text{beta}(a,b,x)}{\text{beta}(a,b)}$</p>
zeta(z)	<p>Дзета-функция Римана. Для действительных $z > 1$ определена сходящимся рядом Дирихле $\sum_{n=1}^{\infty} \frac{1}{n^z}$, допускающим аналитическое продолжение на всю комплексную плоскость, кроме точки $z=1$. В комплексной плоскости функция $\text{zeta}(z)$ ($z \neq 1$) удовлетворяет функциональному уравнению Римана:</p> $\text{zeta}(z) = 2^z \cdot \pi^{z-1} \cdot \sin\left(\frac{\pi \cdot z}{2}\right) \cdot \text{gamma}(1-z) \cdot \text{zeta}(1-z).$ <p>В математике обычное обозначение $\text{zeta}(z) - \zeta(z)$.</p>
sinIntegral(x)	<p>Интегральный синус. По определению $\text{sinIntegral}(x) = \int_0^x \frac{\sin(t)}{t} dt$, где x – любое действительное число. В математике обычное обозначение интегрального синуса – $\text{Si}(x)$</p>
cosIntegral(x)	<p>Интегральный косинус. По определению $\text{cosIntegral}(x) = -\int_x^{\infty} \frac{\cos(t)}{t} dt$, где x – действительное число ($x > 0$). В математике обычное обозначение интегрального косинуса – $\text{Ci}(x)$</p>
expIntegral(x)	<p>Интегральная экспонента (интегральная показательная функция). По определению для любого комплексного z $\text{expIntegral}(z) = \int_{-\infty}^z \frac{e^t}{t} dt$.</p>

Литература

1. Абрамов С. А. Задачи по программированию / С. А. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина, М. И. Селюн. – М.: Наука, 1988. -224 с.
2. Акопян А. В. Геометрические свойства кривых второго порядка / А. В. Акопян, А. А. Заславский - 2-е изд., дополн. - М.: МЦНМО, 2011. -148 с.
3. Арсак Ж. Программирование игр и головоломок / Ж. Арсак.– М.: Наука, 1990. -224 с.
4. Балдин Е. М. Компьютерная типография LaTeX / Е. М. Балдин; –БХВ-Петербург, 2008. -304 с.
5. Безумова О. Л. Обучение геометрии с использованием возможностей *GeoGebra* / О. Л. Безумова, Р. П. Овчинникова и др. Архангельск, Изд. ООО “Кира”, 2011, -140 с.
6. Григорьев С. Г. Поиск и замещение в гнездовых массивах, С. Г. Григорьев, А. Р. Есаян. Вестник Российского университета дружбы народов. Серия “Информатизация образования” № 3 2016. с. 26-39
7. Григорьев С. Г. Параметрические функции для обработки гнездовых массивов, С. Г. Григорьев, Н. М. Добровольский, А. Р. Есаян. Вестник Московского городского педагогического университета. Серия “Информатика и информатизация образования” М.: Изд. Центр МГПУ, № 3 (37) 2016, с. 8-29
8. Душкин Р. В. Функциональное программирование на языке Haskell. Лучшая практическая реализация функциональной парадигмы; –М.: ДМК Пресс, 2007. -608 с.
9. Есаян А. Р. Maxima. Данные и графика / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин. – Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2011. -367 с.
10. Есаян А. Р. РТС Mathcad Prime 3.1 / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин, М. М. Абдуразаков. –Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2016 –400 с.
11. Есаян А. Р. Обучение алгоритмизации на основе рекурсии: Учеб. пособие для студентов пед. вузов / А. Р. Есаян. – Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2001. -215 с.
12. Есаян А. Р. Подготовка документов в LaTeX2e / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин, Учеб. пос. - Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2013 г. -390 с.
13. Есаян А. Р. Программирование в *Maxima* / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин. – Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2012. -351 с.
14. Есаян А. Р. Творческая лаборатория *Mathematica*: Программирование, функции алгебры и анализа. В 2 ч., Ч. 2 / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский. -Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2005. -258 с.
15. Есаян А. Р. Творческая лаборатория *Mathematica*: Система, данные, графика. В 2 ч., Ч. 1 / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский. -Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2005. -296 с.
16. Есаян, А. Р. Верстка в LyX / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин. –Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2013. -242 с.

17. Зиатдинов Р. А. Геометрическое моделирование и решение задач проективной геометрии в системе GeoGebra // Материалы конференции «Молодежь и современные информационные технологии», Томский политехнический университет, г. Томск, 2010, С. 168-170.
18. Зиатдинов Р. А., Системы динамической геометрии как средство компьютерного моделирования в системе современного математического образования / Р. А. Зиатдинов, В. М. Ракута. European Journal of Contemporary Education, 1(1), 93-100, 2012.
19. Зиатдинов Р. А.. О возможностях использования интерактивной геометрической среды Geogebra 3.0 в учебном процессе // Материалы 10-й Международной конференции “Системы компьютерной математики и их приложения”, СмолГУ, г. Смоленск, 2009, С. 39-40.
20. Коксетер Г. С. М. Введение в геометрию. М.: Наука, главная ред. физ.- мат. лит., 1966. -648 с.
21. Коксетер Г. С. М., Новые встречи с геометрией / Коксетер Г.С.М, Крейцер С.: Библиотека математического кружка, 1978. -224 с.
22. Кормен Т. Алгоритмы: построение и анализ = Introduction to Algorithms / Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. – 2-е изд. –М.: “Вильямс”, 2006.
23. Куланин А. Д., Геометрия треугольника в задачах / А. Д. Куланин, С. Н. Федин. - М.: Книжный дом “ЛИБРОКОМ”, 2009. -208 с.
24. Ларин С. В. Компьютерная анимация в среде *GeoGebra* на уроках математики, Легион, –Ростов-на-Дону, 2015. -192 с.
25. Мякишев А. Г. Элементы геометрии треугольника. Библиотека “Математическое просвещение”, вып. 19, М.: МЦНМО, 2009.
26. Окулов С. М. Алгоритмы обработки строк. –М.: БИНОМ. Лаборатория знаний, 2009. -256 с.
27. Окулов С. М. Динамическое программирование / С. М. Окулов, О. А. Пестов. –М.: БИНОМ. Лаборатория знаний, 2012. -296 с.
28. Окулов С. М. Дискретная математика. Теория и практика решения задач по информатике / С. М. Окулов. –М.: БИНОМ. Лаборатория знаний, 2008. -422 с.
29. Окулов С. М. Основы программирования. –М.: БИНОМ. Лаборатория знаний, 2006. -440 с.
30. Окулов С. М. Программирование в алгоритмах. –М.: БИНОМ. Лаборатория знаний, 2013. -384 с.
31. Половинкин Е. С. Элементы выпуклого и сильно выпуклого анализа / Е. С. Половинкин, М. В. Балашов – М.: Физматлит, 2004. -416 с.
32. Понарин Я. П. Элементарная геометрия. Т.1. –М.: МЦНМО, 2004. -312 с.
33. Прасолов В. В. Задачи по планиметрии –М.: МЦНМО, 2007.
34. Прасолов В. В. Точки Брокера и изогональное сопряжение. Библиотека “Математическое просвещение” вып. 4. М.: МЦНМО, –М: 2000, с. 1-24
35. Препарата Ф. Вычислительная геометрия: Введение = Computational Geometry An introduction / Ф. Препарата, М. Шеймос. – М.: Мир, 1989. -478.
36. Скворцов А. В. Триангуляция Делоне и её применение. — Томск: Изд-во Томского университета, 2002. -128 с.
37. Таха Х. Введение в исследование операций / Х. Таха, М.: Мир, 1985. В 2 т. – Т.1. -479 с.; Т.2. -496 с.
38. Уокенбах, Дж. Excel 2013. Профессиональное программирование на VBA. –М.: Изд. “ООО Вильямс”, 2014. -960 с.

39. Уокенбах, Дж. Excel 2013. Трюки и советы Джона Уокенбаха. –СПб.: Питер, 2014 г. -336 с.
40. Уокенбах, Дж. Microsoft Excel 2013. Библия пользователя. Исчерпывающее руководство. –М.: Изд. “ООО Вильямс”, 2015. -918 с.
41. Уокенбах, Дж. Формулы в Microsoft Excel 2013. –М.: Изд. “ООО Вильямс”, 2014. -720 с.
42. Химмельблау, Д. Прикладное нелинейное программирование / Д. Химмельблау -М.: Мир, 1975.
43. Холл М. Комбинаторика. М.: –Мир, 1970. -424 с.
44. Шарыгин И. Ф. Геометрия 7–9 классы. –М.: Дрофа, 2002. -368 с.
45. Шарыгин И. Ф. Геометрия 9–11 классы. От учебной задачи к творческой. – М.: Дрофа, 1996. –398 с.
46. Шень А. Х. Программирование: Теоремы и задачи / А. Х. Шень. –М.: МЦНМО, 2001.
47. Barbara K. D'Ambrosia, Carl R. Spitznagel, John Carroll / ICTCM - 26th International Conference on Technology in Collegiate Mathematics / University Department of Mathematics and Computer Science, Cleveland, OH, USA / <http://archives.math.utk.edu/ICTCM/VOL26/S030/paper.pdf>, 2014.
48. Barbara K. D'Ambrosia, Carl R. Spitznagel, John Carroll / University Department of Mathematics and Computer Science, Cleveland, OH, USA / <http://tube.geogebra.org/student/msOd3EhOt>, 2014.
49. Bryan O'Sullivan, John Goerzen, Don Stewart. Real World Haskell – O'Reilly, 2008 -710с.
50. Dijkstra E. W. A note on two problems in connexion with graphs // Numer. Math — Springer Science+Business Media, 1959. – vol. 1, Iss. 1. –p. 269-271
51. Duval, R., Representation, vision and visualization: Cognitive functions in mathematical thinking. Basic issues for learning. In: Proceedings of the twenty-first annual meeting of the North American Chapter of the International group for the Psychology of Mathematics Education. 1999. p. 3-26
52. GeoGebra Manual. The official manual of GeoGebra. PDF generated at: Tue, 28 Apr 2015 19:24:47 CEST. -325 p.
53. Haftendorn D. Dynamische Mathematik – Bewegung beflügelt Verstehen (English: Dynamic mathematics - movement fosters understanding). In: Beiträge zum Mathematikunterricht. Franzbecker. 2005, p. 235-238
54. Hohenwarter M., GeoGebra – didaktische Materialien und Anwendungen für den Mathematikunterricht (English: GeoGebra - educational material and applications for mathematics teaching). PhD thesis, University of Salzburg. 2006. -334 p. http://www.geogebra.org/publications/mhohen_diss.pdf
55. Hohenwarter M. GeoGebra – ein Softwaresystem für dynamische Geometrie und Algebra der Ebene (English: GeoGebra – a software system for dynamic geometry and algebra in the plane). Master's thesis, University of Salzburg. 2002. -235 p. http://www.geogebra.org/publications/diplomarbeit_geogebra.pdf
56. Kimberling, C. Central Points and Central Lines in the Plane of a Triangle, Mathematics Magazine 67 (3), 1994, 163–187.
57. Kimberling, C. Triangle Centers and Central Triangles, Congr. Numer. 129, 1998. p. 1-295.
58. Model-Centered Learning. Pathways to Mathematical Understanding Using GeoGebra. Volume 6. Sense Publishers Rotterdam/Boston/Taipei, 2011. -257 p.

59. Svrtan D., Non-Euclidean versions of some classical triangle inequalities, D. Svrtan, D. Veljan, Forum Geometricorum T. 12, 2012.
60. Venema G. A. Exploring Advanced Euclidean Geometry with GeoGebra. The Mathematical Association of America, MAA Service Center, Washington, 2013. -129 p.
61. Venema G. A. Foundations of Geometry, 2nd ed. Department of Mathematics and Statistics Calvin College, Pearson Education, Inc. 2012, -407 p.

Интернет-ресурсы

62. <http://www.geogebra.org/> – сайт GeoGebra
63. https://en.wikipedia.org/wiki/List_of_interactive_geometry_software – список интерактивного программного обеспечения геометрии.
64. <http://geogebra.org/> – сайт GeoGebra института (Калифорния)
65. <http://standards.nctm.org/> – Principles and Standards for School Mathematics. National Council of Teacher of Mathematics (NCTM).
66. <http://www.xmarks.com/topic/geogebra> – Top Sites about GeoGebra
67. <https://ru.wikipedia.org/wiki/Треугольник> – Википедия о треугольнике
68. https://en.wikipedia.org/wiki/List_of_triangle_topics – Wikipedia, List of triangle topics
69. https://www.geogebra.org/wiki/en/List_Commands – команды GeoGebra.
70. https://www.geogebra.org/manual/en/List_Commands – список команд GeoGebra.
71. [https://ru.wikipedia.org/wiki/Теорема_Эйлера_\(планиметрия\)](https://ru.wikipedia.org/wiki/Теорема_Эйлера_(планиметрия))
72. <http://mathworld.wolfram.com/> – a free resource from Wolfram Research built with Mathematica technology.
73. <http://mathworld.wolfram.com/KimberlingCenter.html> – Wolfram MathWorld, the web's most extensive mathematics resource.
74. <http://mathworld.wolfram.com/search/?q=Kimberling+Center> – the web's most extensive Mathematics resource.
75. <https://www.geogebra.org/materials/> – проекты GeoGebra.
76. <http://vuz.exponenta.ru/PDF/NAUKA/Sbornik12ito.pdf> – международная научно-практическая конф. “Информационные технологии в образовании и науке – Итон 2012”
77. <http://faculty.evansville.edu/ck6/encyclopedia/ETC.html> - Kimberling C. Encyclopedia of Triangle Centers.
78. https://en.wikipedia.org/wiki/Encyclopedia_of_Triangle_Centers – энциклопедия треугольных центров.
79. https://en.wikipedia.org/wiki/List_of_triangle_topics – Wikipedia, List of triangle topics.
80. <http://www.dic.academic.ru/dic.nsf/ruwiki/1638040> – список систем интерактивной геометрии.
81. http://www.wikiwand.com/ru/Список_систем_интерактивной_геометрии
82. http://www.geogebra.org/manual/en/Release_Notes_GeoGebra_5.0
83. http://www.geogebra.org/manual/en/Reference:Changelog_5.0 – изменения в версиях 5.0 и выше
84. [https://ru.wikipedia.org/wiki/Теорема_Эйлера_\(планиметрия\)](https://ru.wikipedia.org/wiki/Теорема_Эйлера_(планиметрия))
85. <http://www.professores.uff.br/hjbortol/disciplinas/2005.1/gma06077/arquivos/ellipse.pdf>

Алфавитный указатель команд, функций, операторов и других средств *GeoGebra*

- – 323,335	<code>\binom</code> – 148	<code>\overleftarrow</code> – 149
<code>&</code> – 149,159	<code>\breve</code> – 149	<code>\overline</code> – 149
\neq – 121	<code>\cdots</code> – 147	<code>\overrightarrow</code> – 149
\leq – 121	<code>\check</code> – 149	<code>\qquad</code> – 147
\geq – 121	<code>\ddot</code> – 149	<code>\quad</code> – 147
\in – 121	<code>\ddots</code> – 147	<code>\rangle</code> – 153
\perp – 121	<code>\dot</code> – 149	<code>\rceil</code> – 153
\neg – 121	<code>\dots</code> – 147	<code>\rfloor</code> – 153
\surd – 121	<code>\end</code> – 150	<code>\right</code> – 153
\rightarrow – 19,38,377	<code>\frac</code> – 148	<code>\right)</code> – 149
\leftarrow – 19,38,337	<code>\grave</code> – 149	<code>\right]</code> – 149
\downarrow – 19,38,377	<code>\hat</code> – 149	<code>\right </code> – 149
\uparrow – 19,38,377	<code>\hspace</code> – 147	<code>\right\}</code> – 149
π – 399	<code>\infty</code> – 149	<code>\slash</code> – 147,153
$*$ – 323,335,337	<code>\int</code> – 148	<code>\sqrt</code> – 148
\cdot – 317	<code>\langle</code> – 153	<code>\substack</code> – 156
$/$ – 153,335	<code>\lceil</code> – 153	<code>\sum</code> – 148
\backslash – 153,323	<code>\ldots</code> – 147	<code>\tilde</code> – 149
$!$ – 147	<code>\left</code> – 153	<code>\to</code> – 149
<code>\\$ (\&,\%,_,\sim,\{,\})</code> – 147	<code>\left(</code> – 149	<code>\underbrace</code> – 149,156
$\backslash,$ – 147	<code>\left[</code> – 149	<code>\underleftrightharpoonrightarrow</code> – 149
$\backslash:$ – 147	<code>\left\{</code> – 149	<code>\underleftarrow</code> – 149
$\backslash;$ – 147,150	<code>\left </code> – 149	<code>\underline</code> – 149
$\backslash\backslash$ – 147	<code>\lfloor</code> – 153	<code>\underrightarrow</code> – 149
$\backslash\{$ – 153	<code>\lim</code> – 149,152	<code>\vdots</code> – 147
$\backslash $ – 147,153	<code>\limits</code> – 152	<code>\vec</code> – 149
$\backslash\}$ – 153	<code>\mathbb</code> – 150	<code>\vphantom</code> – 155
<code>\acute</code> – 149	<code>\mathcal</code> – 150	<code>\vspace</code> – 147
<code>\backslash</code> – 147	<code>\mathfrak</code> – 150	<code>\widetilde</code> – 149
<code>\backslash</code> – 153	<code>\mathring</code> – 149	<code>\пробел</code> – 147
<code>\bar</code> – 149	<code>\mathtt</code> – 150	\wedge – 121,148,323,337
<code>\begin</code> – 149	<code>\multicolumn</code> – 159	$_$ – 148
<code>\big</code> – 155	<code>\nolimits</code> – 152	\parallel – 121,159
<code>\Big</code> – 155	<code>\oint</code> – 149	$+$ – 323,335
<code>\bigg</code> – 155	<code>\overbrace</code> – 156	$<$ – 121
<code>\Bigg</code> – 155	<code>\overleftharpoonrightarrow</code> – 149	$== (\stackrel{?}{=})$ – 121


> – 121	betaRegularized – 403	CountIf – 327
3D Graphics – 9,35,281	Black – 135	Crimson – 135
	Blue – 135	ctg – 401
A	BoxPlot – 281,306	cth – 402
	Brown – 135	Ctrl – 19,38
abs – 399	Button – 134	Ctrl+↓ – 377
acos – 402		Ctrl+← – 377
acosh – 402	C	Ctrl+→ – 377
Algebra – 8		Ctrl+↑ – 377
Alt – 19,38	c – 159	Ctrl+a – 18,380
Angle – 96	CAS – 8,35	Ctrl+c – 18
AngleBisector – 55	cbrt – 399	Ctrl+Home – 377
Append – 328	ceil – 400	Ctrl+i – 18,381
ApplyMatrix – 338	Cell – 388	Ctrl+j – 17,380
Aqua – 135	CellRange – 388	Ctrl+l – 18,381
arcch – 402	Center – 47	Ctrl+PgDown – 377
arccos – 402	Centroid – 167	Ctrl+PgUp – 377
arcsh – 402	ch – 402	Ctrl+Shift+j – 17,380
arcsin – 402	Checkbox – 133	Ctrl+Shift+s – 371, 372
arctg – 402	Circle – 72	Ctrl+v – 19,382
arth – 402	CircularArc – 74	Ctrl+x – 382
Area – 98	CircularSector – 74	Ctrl+y – 12
AreCollinear – 123	Circumcenter – 167	Ctrl+z – 12
AreConcurrent – 123	CircumCircularArc – 74	Ctrl+c – 382
AreConcyclic – 123	CircumCircularSector-74	Ctrl-технология – 17
AreCongruent – 123	Classes – 333	Curve – 281,289,291
AreEqual – 123	Coefficients – 322	Cyan – 135
AreParallel – 123	Column – 388	
ArePerpendicular – 123	ColumnName – 388	D
arg – 400	Conic – 90	
array – 149,150,157	conjugate – 400	Dark Blue – 135
asin – 402	Construction Protocol – 9	Dark Gray – 135
asinh – 402	ContinuedFraction – 162	Dark Green – 135
atan – 402	ConvexHull – 349	DClick – 21
atan2 – 401	cos – 401	DelaunayTriangula...-351
	cosec – 401	Delete – 18,381
B	cosech – 402	Design Science – 5
	cosh – 402	Determinant – 338
BackSpace – 18,381	cosIntegral – 404	Dilate – 108
BarChart – 281,297	cot – 401	Dimension – 338
beta – 403	coth – 402	Distance – 97

DotPlot – 308	gamma – 403	IsInteger – 124
	gammaRegularized – 404	IsPrime – 124
E	GCD – 323	Iteration – 320
	Ge – 167	IterationList – 320
e – 399	GeoGebra – 3-7,9	
Element – 324	Gergonne point – 217	J
Ellipse – 81	Gold – 135	
End – 377	Graphics – 8,35,281	Join – 330
erf – 403	Graphics 2 – 9,35,281	
Excel – 371	Gray – 135	K
excenters – 167	Green – 135	
Execute – 341		K – 167
exp – 401	H	KeepIf – 328
expIntegral – 404		Kimberling – 166
Extremum – 49	H – 167	
	Harmonic conjugate–168	L
F	Histogram – 281,301	
	HistogramRight–281,303	l – 159
F – 168	Home – 377	Last – 325
f[список] – 323	Hyperbola – 85	LaTeX – 58,143
false – 122		LaTeX2e – 143
Fermat point – 251	I	LCM – 323
Feuerbach point – 241		ld – 401
FillCells – 388	I – 167	Length – 65,98,316
FillColumn – 388	i – 399	LetterToUnicode – 160
FillRow – 388	Identity – 338	lg – 401
First – 65,325	imaginary – 400	Light Blue – 135
FitLine – 61	ImplicitCurve – 281,286	Light Gray – 135
Flatten – 333	Incenter – 167	Light Green – 135
floor – 400	IndexOf – 327	Light Orange – 135
FormulaText – 163	Indigo – 135	Light Purple – 135
fractionalPart – 400	InputBox – 140	Light Violet – 135
FractionText – 160	Insert – 329	Light Yellow – 135
Frequency – 332	IntegralBetween – 99	Lime – 135
FrequencyPolygon – 310	Intersect – 47	Line – 50
Function – 281,284,293	InterSection – 330	ln – 401
	IntersectPath – 47	Locus – 63,281
G	Invert – 338	log – 401
	IsDefined – 124	LyX – 58,143
G – 167	IsInRegion – 124	

M	Polar – 61	Segment – 51
	polyGamma – 403	SelectedElement – 334
M – 168	Polygon – 52,67	SelectedIndex – 334
Magenta – 135	Probability Calculator– 9	Semicircle – 74
Markus Hohenwater – 6	Product – 331	Sequence – 281,282,317
Maroon – 135	Prove – 125	SetColor – 136,137
Mathtype – 5	ProveDetails – 125	SetDynamicColor – 137
MatrixRank – 338	psi – 403	SetValue – 328
Microsoft – 371	Purple – 135	sgn – 399
Midpoint – 47		sh – 402
MinimumSpanning...-353	R	Shear – 108
Mittenpunkt – 168,229		Shift – 19,38
	r – 159	Shift+→ – 378
N	random – 399	Shift+← – 378
	RandomElement – 326	Shift+↓ – 378
N – 167,169	Ray – 52	Shift+↑ – 378
N' – 169	RClick – 21	Shift+Click – 378
n! – 402	real – 400	Shift+Ctrl+Home – 378
Na – 168	Red – 135	Shift+Ctrl-технол... – 17
Nagel point – 222	ReduceRowEche...– 338	Shift+End – 378
nroot – 399	Reflect – 102,103,105	Shift+Home – 378
	Relation – 123	Shift+Tab – 16
O	Remove – 329	Shift+Tab – 377
	RemoveUndefined – 329	Shift-технология – 17
O – 167	ResidualPlot – 281,315	ShortestDistance – 344
Orange – 135	Reverse – 332	Shuffle – 326
OrdinalRank – 332	RootList – 320	sign – 399
Orthocenter – 167	Roots – 49	Silver – 135
	Rotate – 39,106	sin – 401
P	RotateText – 39,164,309	sinh – 402
	round – 400	sinIntegral – 404
Parabola – 89	Row – 388	Slider – 130
Perimeter – 65		Slope – 100
PerpendicularBisector-55	S	Sort – 332
PerpendicularLine – 54		Sp – 168
Perpendi...Vector-54,340	S – 168	Spieker center – 235
PgDown – 377	S' – 169	Spline – 282
PgUp – 377	Sample – 326	Spreadsheet – 8,35,372
Pink – 135	ScientificText – 160	sqrt – 399
Point – 320	sec – 401	StepGraph – 281,311
PointList – 320,333	sech – 402	StickGraph – 281,314

Stretch – 108	Union – 330	X(15) – 168,261
Sum – 331	Unique – 332	X(16) – 168,261
SurdText – 161	UnitPerpendicula... – 340	X(17) – 169,267
	UnitVector – 340	X(18) – 169,267
T		X(2) – 167,175
	V	X(3) – 167,180
Tab – 16,377		X(4) – 167,190
TableText – 164	Vector – 52,340	X(5) – 167,201
Take – 330	VerticalText – 164	X(6) – 167,210
tan – 401	Violet – 135	X(7) – 167,217
Tangent – 59	Virtual Keyboard – 9	X(8) – 168
tanh – 402	VisiCalc – 371	X(9) – 168,229
Text – 162	Voronoi – 350	x[P] – 401
TextToUnicode – 160		x[список] – 322
tg – 401	W	
th – 402		Y
TiedRank – 332	White – 135	
Translate – 107		y[P] – 401
Transpose – 338	U	y[список] – 322
TravelingSalesman – 347		Yellow – 135
TriangleCenter – 166,224	X – 168	
true – 122	X' – 168	Z
Turquoise – 135	X(1) – 167,169	
	X(10) – 168,235	z[P] – 401
U	X(11) – 168,241	z[список] – 322
	X(12) – 168,249	zeta – 404
UnicodeToLetter – 160	X(13) – 168,251	Zip – 318
UnicodeToText – 160	X(14) – 168,251	

Оглавление

Предисловие	3
Введение	5
I. Среда разработки приложений	8
1.1. Основные панели	8
1.2. Элементы интерфейса	9
2. Панели “Объекты” и “Полотно”	12
2.1. Связь алгебры и геометрии	12
2.2. Объекты	13
2.3. Выделение объектов	16
2.4. Некоторые операции с объектами	18
2.5. Изменение свойств объектов	20
2.6. Создание динамических моделей	25
2.6.1. Использование алгебраических средств	26
2.6.2. Использование геометрических средств	29
2.6.3. Панель “Протокол”	31
3. Средства для работы с панелями “Объекты” и “Полотно”	35
3.1. Пиктографические меню панелей	35
3.2. Пиктографические меню панели “Полотно”	36
3.3. Инструменты и команды	37
3.3.1. Перемещение объектов	37
3.3.2. Перемещение и масштабирование чертежа	40
3.3.3. Точки	43
3.3.4. Прямые	49
3.3.5. Специальные прямые	53
3.3.6. Многоугольники	65
3.3.7. Окружности	68
3.3.8. Конические сечения	75
3.3.9. Углы, длины, площади	90
3.3.10. Линейные и нелинейные преобразования объектов	101
3.3.11. Тексты и рисование	114
3.3.12. Логические выражения	120
3.3.13. Управляющие элементы	127
3.4. Пример использования панели “Полотно 2”	140
4. Вставка текстов и формул	143
4.1. Формирование текстов инструментом “  Текст”	143
4.1.1. Управляющие элементы панели “Текст”	144
4.1.2. Символы Юникода списка “Символы”	145
4.1.3. Список “ <i>LaTeX-формула</i> ” и его подписки	148
4.1.4. Дроби, степени, корни, индексы	150
4.1.5. Пределы операторов	152
4.1.6. Скобки переменных размеров	153

4.1.7. Парные скобки на разных строках	154
4.1.8. Скобки предопределенных размеров	155
4.1.9. Дополнительные диакритические знаки	156
4.1.10. Матрицы	157
4.2. Команды для работы с текстами	159
5. Центры треугольника и их свойства	166
5.1. Некоторые центры треугольников	166
5.2. Инцентр	169
5.3. Центроид	175
5.4. Центр описанной окружности	180
5.5. Ортоцентр	190
5.6. Центр окружности 9 точек	201
5.7. Точка Лемуана	210
5.8. Точка Жергонна	217
5.9. Точка Нагеля	222
5.10. Средняя точка	229
5.11. Центр Шпикера	235
5.12. Точки Фейербаха	241
5.13. Точка, гармонически сопряженная точке Фейербаха	249
5.14. Точки Ферма	251
5.15. Точки Аполлония	261
5.16. Точки Наполеона	266
5.17. Обобщенные точки Наполеона	272
6. 2D-графики	281
6.1. Точечные графики	281
6.2. Графики функций одной переменной	284
6.3. Графики неявных функций	286
6.4. Графики параметрических функций	288
6.5. Графики функций в полярных координатах	290
6.5.1. Алгебраический подход	291
6.5.2. Геометрический подход	293
6.6. Специальные графики	297
6.6.1. Гистограммы	297
6.6.1.1. Вывод гистограмм командой <i>BarChart</i>	297
6.6.1.2. Вывод гистограмм командой <i>Histogram</i>	301
6.6.1.3. Вывод гистограмм командой <i>HistogramRight</i>	303
6.6.2. Диаграммы размаха	304
6.6.3. Частотные точечные графики	308
6.6.4. Графики частотного полигона	310
6.6.5. Пошаговые графики	311
6.6.6. Палочные графики	314
6.6.7. Графики остатков	315
7. Работа со списками	316
7.1. Простые и вложенные списки	316

7.2. Создание списков	317
7.3. Операции и функции с числовыми списками	323
7.4. Выборки элементов из списков	324
7.5. Выборки элементов с помощью логических функций	327
7.6. Добавления, удаления, вставки, вырезки	328
7.7. Объединение, соединение и пересечение списков	330
7.8. Суммы и произведения элементов списков	331
7.9. Сортировка и связанные с ней команды	332
7.10. Сглаживание и другие команды	333
7.11. Представление списков объектами типа “раскрывающийся список”	334
7.12. Работа с матрицами и векторами	335
7.12.1. Работа с матрицами	335
7.12.2. Работа с векторами	340
7.13. Выполнение команд из списков	341
8. Некоторые задачи дискретной математики	343
8.1. Задача о кратчайшем пути	343
8.2. Задача коммивояжера	346
8.3. Нахождение выпуклой оболочки точек	349
8.4. Диаграмма Вороного	350
8.5. Триангуляция Делоне	351
8.6. Нахождение минимального остовного дерева	353
9. Создание новых инструментов	355
9.1. Инструмент “Треугольник и углы”	355
9.2. Инструмент “Окружность и центр”	360
9.3. Инструмент “Деление отрезка на равные части”	361
9.4. Задачи для создания новых инструментов	362
9.4.1. Теорема о “бабочке”	362
9.4.2. Теорема Эйлера	364
9.4.3. Аналог теоремы Эйлера	365
9.4.4. Интересное соотношение	366
9.4.5. Теорема Мансиона	366
9.4.6. Прямая Эйлера	368
9.4.7. Окружность 9 точек	368
9.4.8. Теорема Фейербаха	369
9.4.9. Пересечение четырех окружностей девяти точек	369
10. Электронные таблицы	371
10.1. Инструменты панели “Таблица”	371
10.2. Открытие и закрытие таблицы	372
10.3. Имена столбцов, строк и ячеек таблицы	374
10.4. Ввод чисел, точек и формул	374
10.5. Навигация по ячейкам	377
10.6. Активные ячейки и активные группы ячеек	378
10.7. Выделенные ячейки	380

10.8. Копирования, вырезки и вставки данных	382
10.9. Копирования данных с помощью маркеров	383
10.10. Команды для работы с таблицами	387
10.11. Создание некоторых объектов	389
10.12. Формирование характеристик элементов таблицы	393
Приложение. Встроенные (предопределенные)	399
математические функции <i>GeoGebra</i>	
Литература и интернет-ресурсы	405
Алфавитный указатель команд, функций, операторов	
и других средств <i>GeoGebra</i>	409

Учебное издание

**ЕСАЯН Альберт Рубенович,
ДОБРОВОЛЬСКИЙ Николай Михайлович,
СЕДОВА Елена Александровна,
ЯКУШИН Алексей Валериевич**

**Динамическая математическая
образовательная среда
GEOGEBRA**

Учебное пособие

Часть I

Подписано в печать 10.02.2017. Формат 60×90/8.
Бумага офсетная. Печать трафаретная.
Усл. печ. л. 52,25. Тираж 100 экз. (1-й завод – 50 экз.).
«С» 1712. Заказ 16/075.

Издательство Тульского государственного педагогического университета
им. Л. Н. Толстого. 300026, Тула, просп. Ленина, 125.

Отпечатано с оригинал-макета, предоставленного авторами,
в Издательском центре ТГПУ им. Л. Н. Толстого.
300026, Тула, просп. Ленина, 125.