

ФГОС

9



Л. Л. Босова
А. Ю. Босова

ИНФОРМАТИКА



ИЗДАТЕЛЬСТВО

БИНОМ

Л. Л. Босова, А. Ю. Босова

ИНФОРМАТИКА

**Учебник
для 9 класса**

**Рекомендовано
Министерством образования и науки
Российской Федерации
к использованию в образовательном процессе
в имеющих государственную аккредитацию
и реализующих образовательные программы
общего образования образовательных учреждениях**



**Москва
БИНОМ. Лаборатория знаний
2013**

Введение

Уважаемые девятиклассники!

Впереди у вас целый учебный год работы, после чего вы станете выпускниками основной школы.

Вам предстоит:

- систематизировать свои представления об информационном моделировании как основном методе приобретения знаний;
- расширить знания и укрепить навыки использования средств информационных и коммуникационных технологий (ИКТ) для сбора, хранения, преобразования и передачи различных видов информации;
- овладеть такими универсальными умениями информационного характера, как постановка и формулирование проблемы; поиск и выделение необходимой информации, структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных условий; самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера.

Очень важно, чтобы вы смогли увязать учебное содержание с собственным жизненным опытом, понять роль подготовки в области информатики в современном мире, испытывали чувство личной ответственности за качество окружающей информационной среды, были готовы к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и ИКТ.

Как и в предыдущих учебниках, в учебнике для 9 класса кроме основной информации содержатся многочисленные ссылки на образовательные ресурсы сети Интернет, в том числе на такие порталы, как:

- 1) Единая коллекция цифровых образовательных ресурсов (<http://sc.edu.ru/>);
- 2) Федеральный центр информационных образовательных ресурсов (<http://fcior.edu.ru/>);
- 3) сайт методической службы издательства (<http://metodist.Lbz.ru>).

На страницах учебника подробно рассмотрены решения типовых задач по каждой изучаемой теме. В конце каждой главы учебника приведены тестовые задания, которые помогут вам оценить, хорошо ли вы освоили теоретический материал и можете ли применять свои знания для решения возникающих проблем.

Изучая теоретический материал, работая с дополнительными материалами, отвечая на вопросы, решая задачи и выполняя практические задания на компьютере, вы сможете полностью подготовиться к сдаче выпускного экзамена по курсу информатики в форме государственной итоговой аттестации (ГИА), требования к которому размещены на сайте <http://fipi.ru/>.

В работе с учебником вам помогут навигационные значки:



— важное утверждение или определение;



— интересная информация;



— пример решения задачи;



— информация, полезная для решения практических задач;



— ссылка на ресурс в Интернете;



— дополнительный материал к параграфу, содержащийся в электронном приложении к учебнику (<http://methodist.Lbz.ru/>);



— вопросы в тексте параграфа, вопросы и задания для самоконтроля;



— задания для подготовки к итоговой аттестации;



— домашний проект или исследование;



— задания для практических работ на компьютере.

Желаем успехов в изучении информатики!

Глава 1

МОДЕЛИРОВАНИЕ И ФОРМАЛИЗАЦИЯ

§ 1.1

Моделирование как метод познания

Ключевые слова:

- модель
- моделирование
- цель моделирования
- натурная (материальная) модель
- информационная модель
- формализация
- классификация информационных моделей

1.1.1. Модели и моделирование

Человек стремится познать объекты (предметы, процессы, явления) окружающего мира, т. е. понять, как устроен конкретный объект, каковы его структура, основные свойства, законы развития и взаимодействия с другими объектами. Для решения многих практических задач важно знать:

- как изменятся характеристики объекта при определённом воздействии на него со стороны других объектов («Что будет, если...?»);
- какое надо произвести воздействие на объект, чтобы изменить его свойства в соответствии с новыми требованиями («Как сделать, чтобы...?»);
- какое сочетание характеристик объекта является наилучшим в заданных условиях («Как сделать лучше?»).

Одним из методов познания объектов окружающего мира является моделирование, состоящее в создании и исследовании упрощённых заменителей реальных объектов. Объект-заменитель принято назы-

вать моделью, а исходный объект — прототипом или оригиналом. Примеры моделей приведены на рис. 1.1.

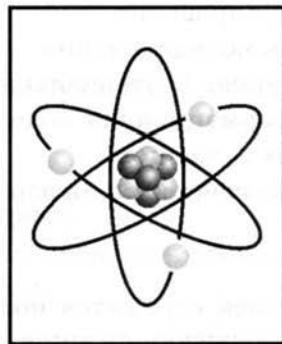
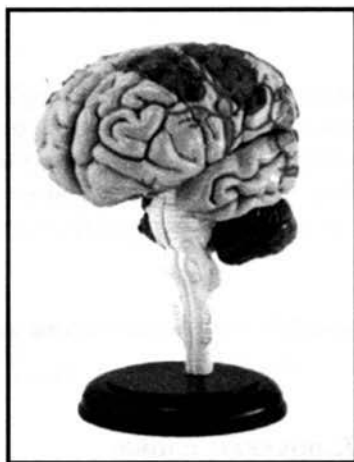
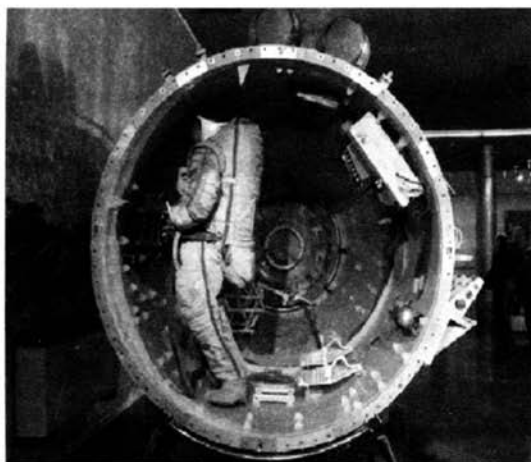


Рис. 1.1. Примеры моделей

К созданию моделей прибегают, когда исследуемый объект слишком велик (Солнечная система) или слишком мал (атом), когда процесс протекает очень быстро (переработка топлива в двигателе внутреннего сгорания) или очень медленно (геологические процессы), когда исследование объекта может оказаться опасным для окружающих (атомный взрыв), привести к разрушению его самого (проверка сейсмических свойств высотного здания) или когда создание реального объекта очень дорого (новое архитектурное решение) и т. д.

Модель не является точной копией объекта-оригинала: она отражает только часть его свойств, отношений и особенностей поведения.

Чем больше признаков объекта отражает модель, тем она полнее. Однако отразить в модели все признаки объекта-оригинала невозможно, а чаще всего и не нужно. Признаки объекта-оригинала, которые должны быть воспроизведены в модели, определяются целью моделирования — назначением будущей модели. Эти признаки называются **существенными** для данной модели с точки зрения цели моделирования.

Подумайте, какие признаки объекта «театр» будут существенными при создании его модели с точки зрения: 1) строительной компании, занимающейся возведением здания театра; 2) режиссёра, готовящего постановку нового спектакля; 3) кассира, продающего билеты; 4) зрителя, собирающегося посетить представление.



Модель — это новый объект, который отражает существенные с точки зрения цели моделирования признаки изучаемого предмета, процесса или явления.

Моделирование — метод познания, заключающийся в создании и исследовании моделей.



Поскольку любая модель всегда отражает только часть признаков оригинала, можно создавать и использовать разные модели одного и того же объекта. Например: мяч может воспроизвести только одно свойство Земли — её форму, обычный глобус отражает ещё расположение материков, а глобус, входящий в состав действующей модели Солнечной системы, — ещё и траекторию движения Земли вокруг Солнца.

Отразить в модели признаки оригинала можно разными способами.

Во-первых, признаки можно скопировать, воспроизвести. Такую модель называют **натурной (материальной)**. Примерами натуральных моделей являются муляжи и макеты — уменьшенные или увеличенные копии, воспроизводящие внешний вид моделируемого объекта (глобус), его структуру (модель Солнечной системы) или поведение (радиоуправляемая модель автомобиля).

Во-вторых, признаки оригинала можно описать на одном из языков представления (кодирования) информации — дать словесное описание, привести формулу, схему или чертёж и т. д. Такую модель называют **информационной**. В дальнейшем мы будем рассматривать именно информационные модели.

Информационная модель — описание объекта-оригинала на одном из языков представления (кодирования) информации.



1.1.2. Этапы построения информационной модели

Любая модель строится для решения некоторой задачи. Построение информационной модели начинается с анализа условия этой задачи, выраженного на естественном языке (рис. 1.2).

В результате анализа условия задачи определяется объект моделирования и цель моделирования.

После определения цели моделирования в объекте моделирования выделяются свойства, основные части и связи между ними, существенные с точки зрения именно этой цели. При этом должно быть чётко определено, что дано (какие исходные данные известны, какие данные допустимы) и что требуется найти в решаемой задаче. Также должны быть указаны связи между исходными данными и результатами.

Следующим этапом построения информационной модели является формализация — представление выявленных связей и выделенных существенных признаков объекта моделирования в некоторой форме (словесное описание, таблица, рисунок, схема, чертёж, формула, алгоритм, компьютерная программа и т. д.).

Формализация — это замена реального объекта его формальным описанием, т. е. его информационной моделью.



Рис. 1.2. Этапы создания информационной модели

Пример. Ученик 9 класса к уроку литературы должен выучить наизусть три первые строфы первой главы романа А. С. Пушкина «Евгений Онегин», содержащие 42 строки. Сколько ему потребуется времени на выполнение этого задания, если первую строку он может запомнить за 5 секунд, а на запоминание каждой следующей строки ему требуется времени на 2 секунды больше, чем на запоминание предыдущей строки?

В данном случае объектом моделирования является процесс запоминания стихотворения учеником; цель моделирования состоит в том, чтобы получить формулу для расчёта времени, необходимого ученику для заучивания стихотворения.

С точки зрения цели моделирования, существенной является следующая информация: время запоминания первой строки (5 секунд);

разница во времени запоминания очередной и предыдущей строк (2 секунды); количество строк, подлежащих запоминанию (42 строки). Это исходные данные. Результатом должно стать время, необходимое для заучивания всех 42 строк фрагмента романа.

Так как время для заучивания каждой строки, начиная со второй, получается добавлением ко времени, требуемому для заучивания предыдущей строки, постоянного числа, то можно говорить об арифметической прогрессии:

5, 7, 9, 11, ...

Первым членом этой прогрессии является $a_1 = 5$, разность прогрессии $d = 10$, число членов прогрессии $n = 42$.

Из курса алгебры известна формула для вычисления суммы n первых членов арифметической прогрессии:

$$S_n = \frac{2a_1 + d(n-1)}{2} n.$$

Эта формула и является искомой информационной моделью. С её помощью самостоятельно вычислите время, необходимое ученику для заучивания стихотворения.



Информационные модели существуют отдельно от объектов моделирования и могут подвергаться обработке независимо от них. Построив информационную модель, человек использует её вместо объекта-оригинала для исследования этого объекта, решения поставленной задачи.

По адресу <http://earth.google.com/intl/ru/> размещено приложение «Google Планета Земля», предоставляющее возможность путешествовать по нашей планете, не вставая с кресла. Это трёхмерная модель планеты, перемещаясь по которой вы можете: просматривать спутниковые фотографии земной поверхности; осматривать города, отдельные здания и всемирно известные достопримечательности в трёхмерном изображении; исследовать отдалённые галактики, созвездия и планеты; совершать путешествия в прошлое и т. д.



1.1.3. Классификация информационных моделей

Существует множество вариантов классификации информационных моделей. Рассмотрим некоторые из них.

Если взять за основу классификации предметную область, то можно выделить физические, экологические, экономические, социологические и другие модели.

В зависимости от учёта фактора времени выделяют динамические (изменяющиеся с течением времени) и статические (не изменяющиеся с течением времени) модели.

В зависимости от формы представления информации об объекте моделирования различают знаковые, образные и смешанные (образно-знаковые) виды информационных моделей.

Знаковые информационные модели строятся с использованием различных естественных и формальных языков (знаковых систем). Знаковая информационная модель может быть представлена в форме текста на естественном языке или программы на языке программирования, в виде формулы и т. д.

Образные информационные модели (рисунки, фотографии и др.) представляют собой зрительные образы объектов, зафиксированные на каком-либо носителе информации.

В смешанных информационных моделях сочетаются образные и знаковые элементы. Примерами смешанных информационных моделей могут служить географические карты, графики, диаграммы и пр. Во всех этих моделях используются одновременно и графические элементы, и знаки.

САМОЕ ГЛАВНОЕ

Модель — это новый объект, который отражает существенные с точки зрения цели моделирования признаки изучаемого предмета, процесса или явления.

Моделирование — метод познания, заключающийся в создании и исследовании моделей.

Цель моделирования (назначение будущей модели) определяет признаки объекта-оригинала, которые должны быть воспроизведены в модели.

Различают натурные и информационные модели. Натурные модели — реальные предметы, в уменьшенном или увеличенном виде воспроизводящие внешний вид, структуру или поведение моделируемого объекта. Информационные модели — описания объекта-оригинала на одном из языков кодирования информации.

Формализация — процесс замены реального объекта его формальным описанием, т. е. его информационной моделью.

По форме представления различают образные, знаковые и смешанные (образно-знаковые) информационные модели.

Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Что вы можете сказать о формах представления информации в презентации и в учебнике? Какими слайдами вы могли бы дополнить презентацию?
2. Что такое модель? В каких случаях используется моделирование?
3. Подтвердите на примерах справедливость следующих высказываний:
 - а) одному объекту может соответствовать несколько моделей;
 - б) одна модель может соответствовать нескольким объектам.
4. Приведите примеры натуральных и информационных моделей.
5. В приведённом перечне моделей укажите те, которые могут использоваться для:
 - а) представления объектов окружающего мира;
 - б) объяснения известных фактов;
 - в) проверки гипотез и получения новых знаний об исследуемых объектах;
 - г) прогнозирования;
 - д) управления.

Модели: макет застройки жилого района; фотоснимки движения воздушных масс; расписание движения поездов; модель полёта самолёта новой конструкции в аэродинамической трубе; схема строения внутренних органов человека.

6. Приведите пример информационной модели:
 - а) ученика вашего класса;
 - б) игрока баскетбольной команды;
 - в) пациента ветеринарной лечебницы;
 - г) квартиры жилого дома;
 - д) книги в библиотеке;
 - е) диска с аудиозаписями музыкальных произведений;
 - ж) города.
7. Опишите этапы построения информационной модели. В чём суть этапа формализации?
8. Перечислите виды информационных моделей в зависимости от формы представления информации об объекте моделирования. Приведите примеры информационных моделей каждого вида.
9. Ознакомьтесь с 3D-моделями, размещёнными в Единой коллекции цифровых образовательных ресурсов (www.sc.edu.ru/). К какому классу моделей их можно отнести?



§ 1.2

Знаковые модели

Ключевые слова:

- словесные модели
- математические модели
- компьютерные модели

1.2.1. Словесные модели



Словесные модели — это описания предметов, явлений, событий, процессов на естественных языках.

Например, гелиоцентрическая модель мира, которую предложил Коперник, словесно описывалась следующим образом:

- Земля вращается вокруг своей оси и вокруг Солнца;
- все планеты движутся по орбитам, центром которых является Солнце.

Множество словесных моделей содержится в ваших школьных учебниках: в учебнике истории представлены модели исторических событий, в учебнике географии — модели географических объектов и природных процессов, в учебнике биологии — модели объектов животного и растительного мира.

Произведения художественной литературы — это тоже модели, так как они фиксируют внимание читателя на определённых сторонах человеческой жизни. Анализируя литературное произведение, вы выделяете в нём объекты и их свойства, отношения между героями, связи между событиями, проводите параллели с другими произведениями и т. п. Самое непосредственное отношение к понятию модели имеет такой литературный жанр, как басня. Смысл этого жанра состоит в переносе отношений между людьми на отношения между вымышленными персонажами, например животными.

Такие особенности естественного языка, как многозначность, использование слов в прямом и переносном значении, синонимия, омонимия и т. п., придают человеческому общению выразительность, эмоциональность, красочность. Вместе с тем наличие этих особенностей делает естественный язык непригодным для создания информационных моделей во многих сферах профессиональной деятельности (например, в системах «человек — компьютер»).

1.2.2. Математические модели

Основным языком информационного моделирования в науке является язык математики.

Информационные модели, построенные с использованием математических понятий и формул, называются **математическими моделями**.



Язык математики представляет собой совокупность множества формальных языков; с некоторыми из них (алгебраическим, геометрическим) вы познакомились в школе, другие сможете узнать при дальнейшем обучении.

Язык алгебры позволяет формализовать функциональные зависимости между величинами, записав соотношения между количественными характеристиками объекта моделирования. В школьном курсе физики рассматривается много функциональных зависимостей, которые представляют собой математические модели изучаемых явлений или процессов.

Пример 1. Зависимость координаты тела от времени при прямолинейном равномерном движении имеет вид:

$$x = x_0 + v_x t.$$

Изменение координаты тела x при прямолинейном равноускоренном движении в любой момент времени t выражается формулой:

$$x = x_0 + v_{0x} t + \frac{a_x t^2}{2}.$$

С помощью языка алгебры логики строятся логические модели — формализуются (записываются в виде логических выражений) простые и сложные высказывания, выраженные на естественном языке. Путём построения логических моделей удаётся решать логические задачи, создавать логические модели устройств и т. д.





Пример 2. Рассмотрите электрические схемы (рис. 1.3).

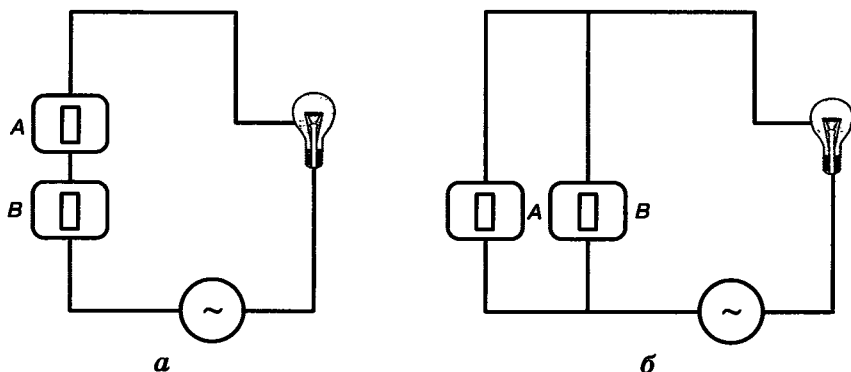


Рис. 1.3. Электрические схемы последовательного (а) и параллельного (б) соединения переключателей

На них изображены известные вам из курса физики последовательное и параллельное соединения переключателей. В первом случае, чтобы лампочка загорелась, должны быть включены оба переключателя. Во втором случае достаточно, чтобы был включён один из переключателей. Можно провести аналогию между элементами электрических схем и объектами и операциями алгебры логики:

Электрическая схема	Алгебра логики
Переключатель	Высказывание
Переключатель включён	1
Переключатель выключен	0
Последовательное соединение переключателей	Конъюнкция
Параллельное соединение переключателей	Дизъюнкция

Спроектируем электрическую цепь, показывающую итог тайного голосования комиссии в составе председателя и двух рядовых членов. При голосовании «за» каждый член комиссии нажимает кнопку. Предложение считается принятым, если члены комиссии проголосуют за него единогласно либо если свои голоса «за» отдадут председатель и один из рядовых членов комиссии. В этих случаях загорается лампочка.

Решение. Пусть голосу председателя соответствует переключатель A , голосам рядовых членов — переключатели B и C . Тогда $F(A, B, C) = A \& B \& C \vee A \& B \vee A \& C$.

Упростим полученное логическое выражение:

$$F(A, B, C) = A \& B \& (C \vee 1) \vee A \& C = A \& B \& 1 \vee A \& C = \\ = A \& B \vee A \& C = A \& (B \vee C).$$

Мы получили логическую модель, позволяющую построить схему проектируемой электрической цепи, изображённую на рис. 1.4.

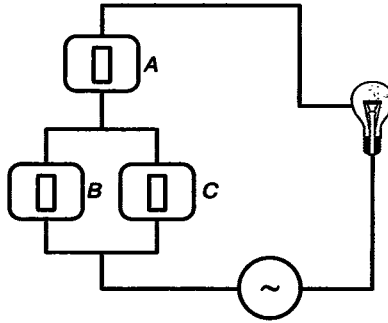


Рис. 1.4. Модель электрической цепи

1.2.3. Компьютерные математические модели

Многие процессы, происходящие в окружающем нас мире, описываются очень сложными математическими соотношениями (уравнениями, неравенствами, системами уравнений и неравенств). До появления компьютеров, обладающих высокой скоростью вычислений, у человека не было возможности проводить соответствующие вычисления, на счёт «вручную» уходило очень много времени.

В настоящее время многие сложные математические модели могут быть реализованы¹ на компьютере. При этом используются такие средства, как:

- системы программирования;
- электронные таблицы;
- специализированные математические пакеты и программные средства для моделирования.

Математические модели, реализованные с помощью систем программирования, электронных таблиц, специализированных математических пакетов и программных средств для моделирования, называются **компьютерными математическими моделями**.



¹ Реализация математической модели — это расчёт состояния (выходных параметров) моделируемой системы по формулам, связывающим её входные и выходные параметры.

Средства компьютерной графики позволяют визуализировать результаты расчётов, получаемых в процессе работы с компьютерными моделями.



С помощью ресурса «Демонстрационная математическая модель» (119324) вы сможете смоделировать полёт снаряда, выпущенного из пушки при различных исходных данных (<http://sc.edu.ru/>).

Особый интерес для компьютерного математического моделирования представляют сложные системы, элементы которых могут вести себя случайным образом. Примерами таких систем являются многочисленные *системы массового обслуживания*: билетные кассы, торговые предприятия, ремонтные мастерские, служба «Скорой помощи», транспортные потоки на городских дорогах и многие другие модели. Многим знакома ситуация, когда, придя в кассу, магазин, парикмахерскую, мы застаём там очередь. Приходится либо вставать в очередь и какое-то время ждать, либо уходить, т. е. покинуть систему необслуженным. Возможны случаи, когда заявок на обслуживание в системе мало или совсем нет; в этом случае она работает с недогрузкой или простаивает. В системах массового обслуживания количество заявок на обслуживание, время ожидания и точное время выполнения заявки заранее предсказать нельзя — это случайные величины.



Имитационные модели воспроизводят поведение сложных систем, элементы которых могут вести себя случайным образом.

Имитационное моделирование — это искусственный эксперимент, при котором вместо проведения натуральных испытаний с реальным оборудованием проводят опыты с помощью компьютерных моделей. Для получения необходимой информации осуществляется многократный «прогон» моделей со случайными исходными данными, генерируемыми компьютером. В результате образуется такой же набор данных, который можно было бы получить при проведении опытов на реальном оборудовании или в реальной системе. Однако имитационное моделирование на компьютере осуществляется гораздо быстрее и обходится значительно дешевле, чем натурные эксперименты.



С помощью ресурса «Демонстрационная имитационная модель» (119425) вы сможете смоделировать ситуацию в системе массового обслуживания — магазине (<http://sc.edu.ru/>).

САМОЕ ГЛАВНОЕ

Словесные модели — это описания предметов, явлений, событий, процессов на естественных языках.

Информационные модели, построенные с использованием математических понятий и формул, называются математическими моделями.

Математические модели, реализованные с помощью систем программирования, электронных таблиц, специализированных математических пакетов и программных средств для моделирования, называются компьютерными математическими моделями.

Имитационные модели воспроизводят поведение сложных систем, элементы которых могут вести себя случайным образом.

Вопросы и задания

1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Что вы можете сказать о формах представления информации в презентации и в учебнике? Какими слайдами вы могли бы дополнить презентацию?
2. Приведите 2–3 собственных примера словесных моделей, рассматриваемых на уроках истории, географии, биологии.
3. Вспомните басни И. А. Крылова: «Волк и ягнёнок», «Ворона и лисица», «Демьянова уха», «Квартет», «Лебедь, Щука и Рак», «Лисица и виноград», «Слон и Моська», «Стрекоза и Муравей», «Тришкин кафтан» и др. Какие черты характера людей и отношения между людьми смоделировал в них автор?
4. Решите, составив математическую модель, следующую задачу. Теплоход прошёл 4 км против течения реки, а затем прошёл ещё 33 км по течению, затратив на весь путь один час. Найдите собственную скорость теплохода, если скорость течения реки равна 6,5 км/ч.
5. Требуется спроектировать электрическую цепь, показывающую итог тайного голосования комиссии в составе трёх членов. При голосовании «за» член комиссии нажимает кнопку. Предложение считается принятым, если оно собирает большинство голосов. В этом случае загорается лампочка.





6. Решите, составив логическую модель, следующую задачу.

На международных соревнованиях по прыжкам в воду первые пять мест заняли спортсмены из Германии, Италии, Китая, России и Украины. Ещё до начала соревнований эксперты высказали свои предположения об их итогах:

- 1) Первое место займёт спортсмен из Китая, а спортсмен из Украины будет третьим.
- 2) Украина будет на последнем месте, а Германия — на предпоследнем.
- 3) Германия точно будет четвёртой, а первое место займёт Китай.
- 4) Россия будет первой, а Италия — на втором месте.
- 5) Италия будет пятой, а победит Германия.

По окончании соревнований выяснилось, что каждый эксперт был прав только в одном утверждении. Какие места в соревновании заняли участники?



7. В середине прошлого века экономисты оценили ежегодный объём вычислений, необходимых для эффективного управления народным хозяйством страны. Он составил 10^{17} операций. Можно ли справиться с таким объёмом вычислений за год, если привлечь к работе миллион вычислителей, каждый из которых способен выполнять одну операцию в секунду?



8. Приведите примеры использования компьютерных моделей. Найдите соответствующую информацию в сети Интернет.



9. В Единой коллекции цифровых образовательных ресурсов найдите лабораторную работу «Изучение закона сохранения импульса». В её основу положена математическая модель, описывающая движение тела, брошенного под углом к горизонту, с последующим делением тела на два осколка. Экспериментально проверьте закон сохранения импульса, выполнив работу согласно имеющемуся в ней описанию.



10. В Единой коллекции цифровых образовательных ресурсов найдите игру «Равноплечий рычаг». Изучите правила игры. Вспомните физическую закономерность, положенную в её основу. Попытайтесь «победить» компьютер и сформулировать выигрышную стратегию.

Графические информационные модели

Ключевые слова:

- схема
- карта
- чертёж
- график
- диаграмма
- граф
- сеть
- дерево

1.3.1. Многообразие графических информационных моделей

В графических информационных моделях для наглядного отображения объектов используются условные графические изображения (образные элементы), зачастую дополняемые числами, символами и текстами (знаковыми элементами). Примерами графических моделей могут служить всевозможные схемы, карты, чертежи, графики и диаграммы.

Схема — это представление некоторого объекта в общих, главных чертах с помощью условных обозначений. С помощью схем может быть представлен и внешний вид объекта, и его структура. Схема как информационная модель не претендует на полноту предоставления информации об объекте. С помощью особых приёмов и графических обозначений на ней более рельефно выделяется один или несколько признаков рассматриваемого объекта. Примеры схем приведены на рис. 1.5.

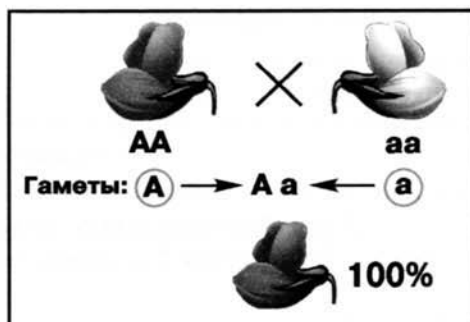
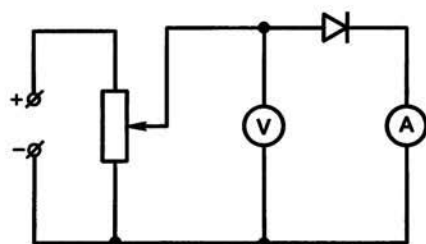


Рис. 1.5. Примеры схем, используемых на уроках физики, биологии, истории

Уменьшенное обобщённое изображение поверхности Земли на плоскости в той или иной системе условных обозначений даёт нам географическая карта.

Чертёж — условное графическое изображение предмета с точным соотношением его размеров, получаемое методом проецирования. Чертёж содержит изображения, размерные числа, текст. Изображения дают представления о геометрической форме объекта, числа — о величине объекта и его частей, надписи — о названии, масштабе, в котором выполнены изображения.

График — графическое изображение, дающее наглядное представление о характере зависимости одной величины (например, пути) от другой (например, времени). График позволяет отслеживать динамику изменения данных.

Диаграмма — графическое изображение, дающее наглядное представление о соотношении каких-либо величин или нескольких значений одной величины, об изменении их значений. Более подробно

типы диаграмм и способы их построения будут рассмотрены при изучении электронных таблиц.

1.3.2. Графы

Если объекты некоторой системы изобразить вершинами, а связи между ними — линиями (рёбрами), то мы получим информационную модель рассматриваемой системы в форме графа. Вершины графа могут изображаться кругами, овалами, точками, прямоугольниками и т. д.

Граф называется взвешенным, если его вершины или рёбра характеризуются некоторой дополнительной информацией — весами вершин или рёбер.

На рис. 1.6 с помощью взвешенного графа изображены дороги между пятью населёнными пунктами *A*, *B*, *C*, *D*, *E*; веса рёбер — протяжённость дорог в километрах.

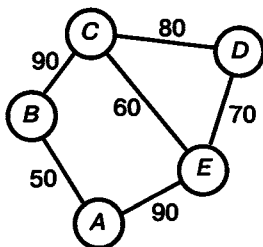


Рис. 1.6. Взвешенный граф

Путь по вершинам и рёбрам графа, в который любое ребро графа входит не более одного раза, называется цепью. Цепь, начальная и конечная вершины которой совпадают, называется циклом.

Граф с циклом называется сетью. Если героев некоторого литературного произведения представить вершинами графа, а существующие между ними связи изобразить рёбрами, то мы получим граф, называемый семантической сетью.

Графы как информационные модели находят широкое применение во многих сферах нашей жизни. Например, можно существующие или вновь проектируемые дома, сооружения, кварталы изображать вершинами, а соединяющие их дороги, инженерные сети, линии электропередач и т. п. — рёбрами графа. По таким графам можно планировать оптимальные транспортные маршруты, кратчайшие объездные пути, расположение торговых точек и других объектов.



Дерево — это граф, в котором нет циклов, т. е. в нём нельзя из некоторой вершины пройти по нескольким различным рёбрам и вернуться в ту же вершину. Отличительной особенностью дерева является то, что между любыми двумя его вершинами существует единственный путь.

Всякая иерархическая система может быть представлена с помощью дерева. У дерева выделяется одна главная вершина, называемая его **корнем**. Каждая вершина дерева (кроме корня) имеет только одного предка, обозначенный предком объект входит в один класс¹ высшего уровня. Любая вершина дерева может порождать несколько потомков — вершин, соответствующих классам нижнего уровня. Такой принцип связи называется «один-ко-многим». Вершины, не имеющие порождённых вершин, называются **листьями**.

Родственные связи между членами семьи удобно изображать с помощью графа, называемого **генеалогическим** или **родословным** деревом.



Ресурс «Живая Родословная» (145555) — инструмент для формирования и анализа генеалогических деревьев, содержащий примеры родословных. С его помощью вы можете изучить генеалогические деревья многих известных семей и построить генеалогическое дерево своей семьи (<http://sc.edu.ru/>).

1.3.3. Использование графов при решении задач

Графы удобно использовать при решении некоторых классов задач.



Пример 1. Для того чтобы записать все трёхзначные числа, состоящие из цифр 1 и 2, можно воспользоваться графом (деревом) на рис. 1.7.

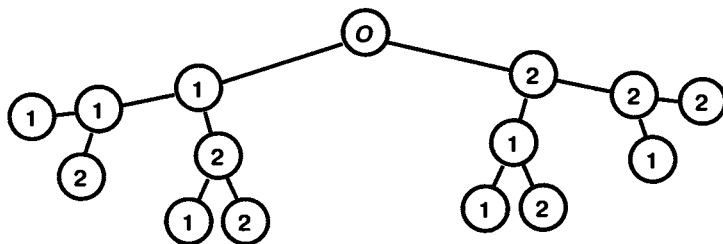


Рис. 1.7. Дерево для решения задачи о записи трёхзначных чисел

¹ Класс — множество объектов, обладающих общими признаками.

Дерево можно не строить, если не требуется выписывать все возможные варианты, а нужно просто указать их количество. В этом случае рассуждать нужно так: в разряде сотен может быть любая из цифр 1 и 2, в разряде десятков — те же два варианта, в разряде единиц — те же два варианта. Следовательно, число различных вариантов: $2 \cdot 2 \cdot 2 = 8$.

В общем случае, если известно количество возможных вариантов выбора на каждом шаге построения графа, то для вычисления общего количества вариантов нужно все эти числа *перемножить*.



Пример 2. Рассмотрим несколько видоизменённую классическую задачу о переправе.



На берегу реки стоит крестьянин (*К*) с лодкой, а рядом с ним — собака (*С*), лиса (*Л*) и гусь (*Г*). Крестьянин должен переправиться сам и перевезти собаку, лису и гуся на другой берег. Однако в лодку кроме крестьянина помещается либо только собака, либо только лиса, либо только гусь. Оставлять же собаку с лисой или лису с гусем без присмотра крестьянина нельзя — собака представляет опасность для лисы, а лиса — для гуся. Как крестьянин должен организовать переправу?

Для решения этой задачи составим граф, вершинами которого будут исходное размещение персонажей на берегу реки, а также всевозможные промежуточные состояния, достигаемые из предыдущих за один шаг переправы. Каждую вершину-состояние переправы обозначим овалом и свяжем рёбрами с состояниями, образованными из неё (рис. 1.8).

Недопустимые по условию задачи состояния выделены пунктирной линией; они исключаются из дальнейшего рассмотрения. Начальное и конечное состояния переправы выделены жирной линией.

На графе видно, что существуют два решения этой задачи. Приведём соответствующий одному из них план переправы:

- 1) крестьянин перевозит лису;
- 2) крестьянин возвращается;
- 3) крестьянин перевозит собаку;
- 4) крестьянин возвращается с лисой;
- 5) крестьянин перевозит гуся;
- 6) крестьянин возвращается;
- 7) крестьянин перевозит лису.

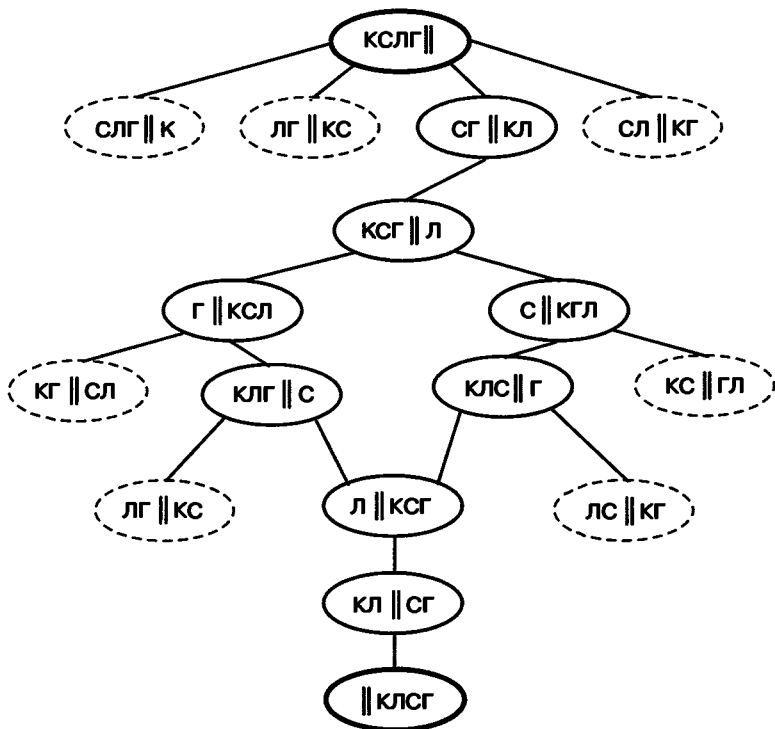


Рис. 1.8. Граф переправы

Пример 3. Рассмотрим следующую игру: сначала в кучке лежат 5 спичек; два игрока убирают спички по очереди, причём за 1 ход можно убрать 1 или 2 спички; выигрывает тот, кто оставит в кучке 1 спичку. Выясним, кто выигрывает при правильной игре — первый (I) или второй (II) игрок.

Игрок I может убрать одну спичку (в этом случае их останется 4) или сразу 2 (в этом случае их останется 3).

Если игрок I оставил 4 спички, игрок II может своим ходом оставить 3 или 2 спички. Если же после хода первого игрока осталось 3 спички, второй игрок может выиграть, взяв две спички и оставив одну.

Если после игрока II осталось 3 или 2 спички, то игрок I в каждой из этих ситуаций имеет шанс на выигрыш.

Таким образом, при правильной стратегии игры всегда выигрывает первый игрок. Для этого своим первым ходом он должен взять одну спичку.

На рис. 1.9 представлен граф, называемый деревом игры; на нём отражены все возможные варианты, в том числе ошибочные (проигрышные) ходы игроков.

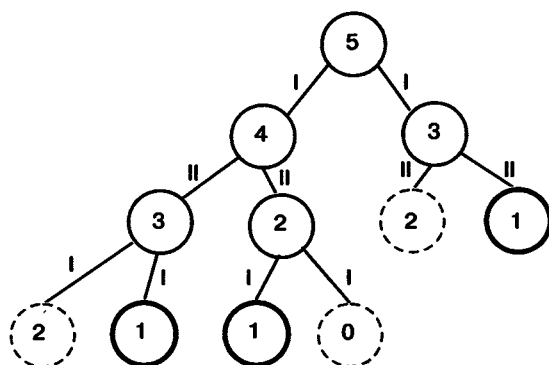


Рис. 1.9. Дерево игры

САМОЕ ГЛАВНОЕ

В графических информационных моделях для наглядного отображения объектов используются условные графические изображения (образные элементы), зачастую дополняемые числами, символами и текстами (знаковыми элементами). Примерами графических моделей могут служить всевозможные схемы, карты, чертежи, графики и диаграммы, графы.

Граф состоит из вершин, связанных линиями — рёбрами. Граф называется взвешенным, если его вершины или рёбра характеризуются некоторой дополнительной информацией — весами вершин (рёбер).

Граф иерархической системы называется деревом. Отличительной особенностью дерева является то, что между любыми двумя его вершинами существует единственный путь.

Вопросы и задания

1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Что вы можете сказать о формах представления информации в презентации и в учебнике? Какими слайдами вы могли бы дополнить презентацию?
2. Какие информационные модели относят к графическим?



3. Приведите примеры графических информационных моделей, с которыми вы имеете дело:

- а) при изучении других предметов;
- б) в повседневной жизни.



4. Что такое граф? Что является вершинами и рёбрами графа на рис. 1.6? Приведите примеры цепей и циклов, имеющих в этом графе. Определите, какие два пункта наиболее удалены друг от друга (два пункта считаются самыми удалёнными, если длина кратчайшего пути между ними больше, чем длина кратчайшего пути между любыми другими двумя пунктами). Укажите длину кратчайшего пути между этими пунктами.



5. Приведите пример системы, модель которой можно представить в форме графа. Изобразите соответствующий граф.

6. Грунтовая дорога проходит последовательно через населённые пункты *A*, *B*, *C* и *D*. При этом длина грунтовой дороги между *A* и *B* равна 40 км, между *B* и *C* — 25 км, и между *C* и *D* — 10 км. Между *A* и *D* дороги нет. Между *A* и *C* построили новое асфальтовое шоссе длиной 30 км. Оцените минимально возможное время движения велосипедиста из пункта *A* в пункт *B*, если его скорость по грунтовой дороге — 20 км/ч, по шоссе — 30 км/ч.



7. Составьте семантическую сеть по русской народной сказке «Колобок».



8. Что такое дерево? Моделями каких систем могут служить деревья? Приведите пример такой системы.

9. Сколько трёхзначных чисел можно записать с помощью цифр 2, 4, 6 и 8 при условии, что в записи числа не должно быть одинаковых цифр?

10. Сколько существует трёхзначных чисел, все цифры которых различны?



11. Для составления цепочек используются бусины, помеченные буквами *A*, *B*, *C*, *D*, *E*. На первом месте в цепочке стоит одна из бусин *A*, *C*, *E*. На втором — любая гласная, если первая буква гласная, и любая согласная, если первая согласная. На третьем месте — одна из бусин *C*, *D*, *E*, не стоящая в цепочке на первом месте. Сколько цепочек можно создать по этому правилу?



12. Два игрока играют в следующую игру. Перед ними лежит куча из 6 камней. Игроки берут камни по очереди. За один ход можно взять 1, 2 или 3 камня. Проигрывает тот, кто забирает последний камень. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Табличные информационные модели

Ключевые слова:

- таблица
- таблица «объект—свойство»
- таблица «объект—объект»

В табличных информационных моделях информация об объектах представляется в виде прямоугольной таблицы, состоящей из столбцов и строк.

Вам хорошо известно табличное представление расписания уроков, в табличной форме представляются расписания движения автобусов, самолётов, поездов и многое другое.

Представленная в таблице информация наглядна, компактна и легко обозрима.

1.4.1. Представление данных в табличной форме

В качестве информационных моделей объектов, обладающих одинаковыми наборами свойств, как правило, используются таблицы типа «объект—свойство».

Например, информацию о регионах нашей страны можно представить с помощью таблицы, фрагмент которой приведён в табл. 1.1.

В этой таблице каждая строка содержит информацию об одном объекте — регионе; столбцы — отдельные характеристики (свойства) рассматриваемых объектов: название, дата образования, площадь и т. д. Такие таблицы могут содержать числовую, текстовую и графическую информацию.

Таблица 1.1

Регионы Российской Федерации

Название	Дата образования (ДД.ММ.ГГ)	Площадь (тыс. км ²)	Население (тыс. чел.)
Астраханская область	27.12.1943	44,1	1006,3
Архангельская область	23.09.1937	587,4	1336,5
Белгородская область	06.01.1954	27,1	1511,6
Владимирская область	14.08.1944	29,0	1524,0
Вологодская область	23.09.1937	145,7	1269,6
Воронежская область	13.06.1934	52,4	2378,8
Калужская область	05.07.1947	29,9	1041,6



В пустую строку таблицы вы можете записать информацию о своём регионе.

В таблицах типа «объект—объект» отражается взаимосвязь между объектами одного или нескольких классов. Например, в школьных журналах есть таблица «Сведения о количестве уроков, пропущенных обучающимися»; её фрагмент представлен в табл. 1.2.

Таблица 1.2

Сведения о пропусках уроков

№	Учащиеся	Месяц: январь					
		Число					
		14	15	16	17	18	19
1	Акуленко Иван						
2	Баранов Владимир	6	6	6			
3	Варнаков Олег						
4	Егорова Виктория				5	6	1
5	Машкова Карина						6

В этой таблице отражена связь «количество пропущенных уроков» между объектами класса «Учащиеся» и объектами класса «Число».

В таблице «Расстояния между городами» (табл. 1.3) представлены расстояния между парами объектов, принадлежащих одному классу «Город». Создайте эту таблицу в текстовом редакторе и добавьте в свободные строки и столбец информацию о своём населённом пункте.



Таблица 1.3

Расстояния между городами (км)

Город	Город			
	Москва	Петрозаводск	Самара	Казань
Москва		1076	1069	815
Петрозаводск	1076		2145	1891
Самара	1069	2145		631
Казань	815	1891	631	

В форме таблицы «объект–объект» можно представить информацию о наличии границ (сухопутной, морской, озёрной, речной) России с другими странами; её фрагмент представлен в табл. 1.4.

Таблица 1.4

Граница Российской Федерации

Страна	Граница			
	сухопутная	речная	озёрная	морская
Норвегия	1	1	0	1
Финляндия	1	1	1	1
Латвия	1	1	1	0
Корея	0	1	0	1
Япония	0	0	0	1

Если граница соответствующего вида есть, то в нужную ячейку ставится 1, а если нет — 0.

Важная особенность этой таблицы состоит в том, что в ней фиксируются не количественные («Сколько?»), а качественные свойства (наличие/отсутствие связи между объектами).

1.4.2. Использование таблиц при решении задач

Рассмотрим несколько примеров задач, которые удобно решать с помощью табличных информационных моделей.

Пример 1. Два игрока играют в следующую игру. Перед ними лежат две кучи камней, в первой из которых 3 камня, а во второй — 2 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или увеличивает в 3 раза число камней в какой-то куче, или добавляет 1 камень в какую-то кучу. Выигрывает игрок, после хода которого общее число камней в двух кучах становится не менее 16. Кто выигрывает при безошибочной игре — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Ранее мы рассмотрели способ записи решения подобных задач с помощью дерева. Сейчас оформим решение в виде таблицы (табл. 1.5).

Таблица 1.5

Таблица игры

Исходное положение	1-й игрок — 1-й ход	2-й игрок — 1-й ход	1-й игрок — 2-й ход	2-й игрок — 2-й ход	
1	2	3	4	5	
3,2,5	9,2,11	27,2,29 ✓			
	3,6,9	3,18,21 ✓			
	4,2,6	12,2,14	36,2,38 ✓		
			4,6,10	12,6,18 ✓	
			5,2,7	15,2,17 ✓	
		4,3,7	12,3,15	36,3,39 ✓	
			4,9,13	12,9,21 ✓	
			5,3,8	15,3,18 ✓	
			4,4,8	12,4,16 ✓	
	3,3,6	9,3,12	27,3,12 ✓		
4,3,7 ¹					

¹ Вариант (как повторный) исключается из дальнейшего рассмотрения.

Три числа в каждой ячейке таблицы обозначают соответственно количество камней в кучах и их сумму. В первом столбце зафиксировано распределение камней перед игрой (исходное положение).

Во втором столбце рассмотрены все возможные варианты ходов первого игрока; победить с первого хода он не может.

В третьем столбце рассмотрены имеющиеся выигрышные варианты ходов второго игрока (отмечены «галочкой»). При безошибочной игре первого игрока такие ситуации возникнуть не должны. Поэтому рассматриваем все возможные ходы второго игрока в случаях, когда у него нет выигрышного хода. Если получены одинаковые варианты, то все из них, кроме одного, исключаем из дальнейшего рассмотрения.

В четвёртом столбце отмечены имеющиеся выигрышные варианты второго хода первого игрока. При безошибочной игре второго игрока такие ситуации возникнуть не должны. Поэтому рассматриваем все возможные ходы первого игрока в случае, когда у него нет выигрышного хода.

В пятом столбце отмечены выигрышные ходы второго игрока, имеющиеся при всех вариантах хода первого игрока.

Таким образом, при безошибочной игре соперников побеждает второй игрок. Его первый ход должен быть таким, чтобы в кучах стало 4 и 3 камня.

Пример 2. С помощью взвешенного графа на рис. 1.6 представлена схема дорог, соединяющих населённые пункты *A, B, C, D, E*. Построим таблицу, соответствующую этому графу (рис. 1.10).

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	×	50			90
<i>B</i>	50	×	90		
<i>C</i>		90	×	80	60
<i>D</i>			80	×	70
<i>E</i>	90		60	70	×

Рис. 1.10. Весовая матрица

Если между парой населённых пунктов существует дорога, то в ячейку на пересечении соответствующих строки и столбца записывается число, равное её длине. Имеющиеся в таблице пустые клетки означают, что дорог между соответствующими населёнными пунктами нет. Построенная таким образом таблица называется **весовой матрицей**.



Для решения некоторых задач бывает удобно по имеющейся таблице строить граф. При этом одной и той же таблице могут соответствовать графы, внешне не похожие друг на друга. Например, рассмотренной выше таблице кроме графа на рис. 1.6 соответствует граф на рис. 1.11.

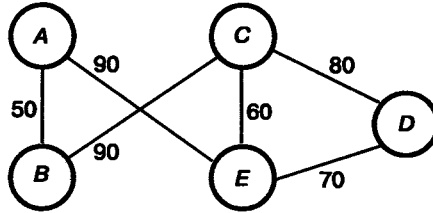


Рис. 1.11. Вариант графа, представляющего схему дорог



Пример 3. Таблицы типа «объект—объект» удобно использовать для решения логических задач, в которых требуется установить взаимно однозначное соответствие между объектами нескольких классов. Рассмотрим задачу, в которой объекты связаны тремя парами отношений.

Три подружки — Аня, Света и Настя — купили различные молочные коктейли в белом, голубом и зелёном стаканчиках. Ане достался не белый стаканчик, а Свете — не голубой. В белом стаканчике не банановый коктейль. В голубой стаканчик налит ванильный коктейль. Света не любит клубничный коктейль.

Требуется выяснить, какой коктейль и в каком стаканчике купила каждая из девочек.

Создадим три следующие таблицы:

Стаканчик	Девочка		
	Аня	Света	Настя
Белый			
Голубой			
Зелёный			

Стаканчик	Коктейль		
	Банановый	Ванильный	Клубничный
Белый			
Голубой			
Зелёный			

Коктейль	Девочка		
	Аня	Света	Настя
Банановый			
Ванильный			
Клубничный			

Отметим в таблицах информацию, содержащуюся в условии задачи:

Стаканчик	Девочка		
	Аня	Света	Настя
Белый	0		
Голубой		0	
Зелёный			

Стаканчик	Коктейль		
	Банановый	Ванильный	Клубничный
Белый	0		
Голубой		1	
Зелёный			

Коктейль	Девочка		
	Аня	Света	Настя
Банановый			
Ванильный			
Клубничный		0	

Имеющейся во второй таблице информации достаточно для того, чтобы заполнить всю эту таблицу:

Стаканчик	Коктейль		
	Банановый	Ванильный	Клубничный
Белый	0	0	1
Голубой	0	1	0
Зелёный	1	0	0

Используя факты, что Света купила не клубничный коктейль и что этот коктейль был налит в белый стаканчик, заполняем всю первую таблицу:

Стаканчик	Девочка		
	Аня	Света	Настя
Белый	0	0	1
Голубой	1	0	0
Зелёный	0	1	0

На основании информации в первой и второй таблицах можем заполнить всю третью таблицу:

Коктейль	Девочка		
	Аня	Света	Настя
Банановый	0	1	0
Ванильный	1	0	0
Клубничный	0	0	1

Ответ: Аня купила ванильный коктейль в голубом стаканчике, Света — банановый коктейль в зелёном стаканчике, Настя — клубничный коктейль в белом стаканчике.

САМОЕ ГЛАВНОЕ

В табличных информационных моделях информация об объекте или процессе представляется в виде прямоугольной таблицы, состоящей из столбцов и строк. Представленная в таблице информация наглядна, компактна и легкообозрима.

Таблица типа «объект—свойство» — это таблица, содержащая информацию о свойствах отдельных объектов, принадлежащих одному классу.

Таблица типа «объект—объект» — это таблица, содержащая информацию о некотором одном свойстве пар объектов, чаще всего принадлежащих разным классам.

Вопросы и задания

1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Что вы можете сказать о формах представления информации в презента-

ции и в учебнике? Какими слайдами вы могли бы дополнить презентацию?

2. Какие преимущества обеспечивают табличные информационные модели по сравнению со словесными описаниями? Приведите пример.
3. Приведите примеры табличных информационных моделей, с которыми вы имеете дело:
 - а) на уроках в школе;
 - б) в повседневной жизни.
4. К какому типу относится таблица «Табель успеваемости», расположенная в конце вашего дневника?
5. Узнайте, в каких случаях в ячейку таблицы ставится знак «×». Почему мы использовали этот знак в таблице (пример 2)?
6. Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 1 камень, а во второй — 2 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или увеличивает в 3 раза число камней в какой-то куче, или добавляет 2 камня в какую-то кучу. Выигрывает игрок, после хода которого общее число камней в двух кучах становится не менее 17. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.
7. Таблица стоимости перевозок устроена следующим образом: числа, стоящие на пересечениях строк и столбцов таблиц, означают стоимость перевозок между соответствующими соседними станциями. Если пересечение строки и столбца пусто, то станции не являются соседними. Стоимость перевозок по маршруту складывается из стоимостей перевозок между соседними станциями. Перевозки между населёнными пунктами *A*, *B*, *C*, *D*, *E* осуществляют три компании, представившие стоимость своих услуг в табличной форме. Какая компания обеспечивает минимальную стоимость проезда из *A* в *B*?

1)

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	×		3	1	
<i>B</i>		×	4		2
<i>C</i>	3	4	×		2
<i>D</i>	1			×	
<i>E</i>		2	2		×

2)

	A	B	C	D	E
A	×		3	1	1
B		×	4		
C	3	4	×		2
D	1			×	
E	1		2		×

3)

	A	B	C	D	E
A	×		3	1	4
B		×	4		2
C	3	4	×		2
D	1			×	
E	4	2	2		×

8. Соревнования по плаванию были в самом разгаре, когда стало ясно, что первые четыре места займут мальчики из пятёрки лидеров. Их имена: Валерий, Николай, Михаил, Игорь, Эдуард, фамилии: Симаков, Чигрин, Зимин, Копылов, Блинов (имена и фамилии названы в произвольном порядке). Нашлись знатоки, которые предсказали, что первое место займёт Копылов, второе — Валерий, третье — Чигрин, четвёртое — Эдуард. Но ни один из ребят не занял того места, какое ему предсказывали. На самом деле первое место завоевал Михаил, второе — Симаков, третье — Николай, четвёртое — Блинов, а Чигрин не попал в четвёрку сильнейших. Назовите имя и фамилию каждого из лидеров.
9. В Норильске, Москве, Ростове и Пятигорске живут четыре супружеские пары (в каждом городе — одна пара). Имена этих супругов: Антон, Борис, Давид, Григорий, Ольга, Мария, Светлана, Екатерина. Антон живёт в Норильске, Борис и Ольга — супруги, Григорий и Светлана не живут в одном городе, Мария живёт в Москве, Светлана — в Ростове. В каком городе живёт каждая из супружеских пар?
10. Постройте граф, отражающий разновидности информационных моделей.

§ 1.5

База данных как модель предметной области

Ключевые слова:

- информационная система
- база данных
- иерархическая база данных
- сетевая база данных
- реляционная база данных
- запись
- поле
- ключ

1.5.1. Информационные системы и базы данных

Современный человек в своей практической деятельности всё чаще и чаще использует различные информационные системы, обеспечивающие хранение, поиск и выдачу информации по его запросам. Примерами информационных систем являются:

- справочная адресная служба большого города;
- транспортная информационная система, обеспечивающая не только возможность получения справочной информации о расписании поездов и самолётов, но и покупку железнодорожных и авиабилетов;
- информационно-поисковая система, содержащая информацию правового характера.

Центральной частью любой информационной системы является база данных.



База данных (БД) — совокупность данных, организованных по определённым правилам, отражающая состояние объектов и их отношений в некоторой предметной области (транспорт, медицина, образование, право и т. д.), предназначенная для хранения во внешней памяти компьютера и постоянного применения.

Базу данных можно рассматривать как информационную модель предметной области.

Основными способами организации данных в базах данных являются иерархический, сетевой и реляционный (рис. 1.12).

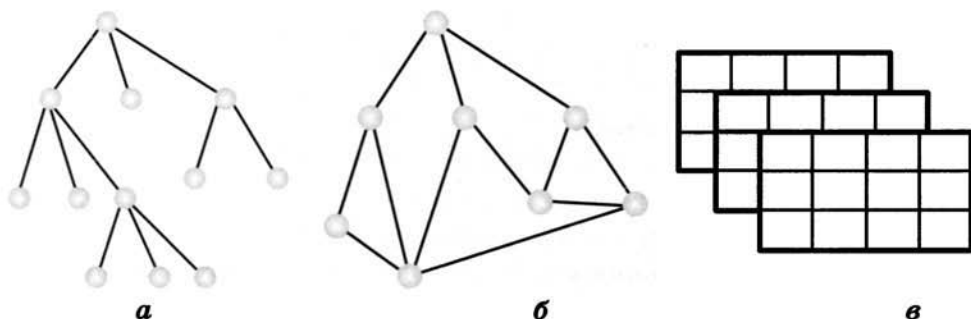


Рис. 1.12. Способы организации данных в БД: а) иерархический, б) сетевой, в) реляционный

В иерархической базе данных существует упорядоченность объектов по уровням. Между объектами существуют связи: каждый объект может быть связан с объектами более низкого уровня. Говорят, что такие объекты находятся в отношении предка к потомку. Иерархический способ организации данных реализован в системе папок операционной системы Windows. Верхний уровень занимает папка *Рабочий стол*. Папки второго уровня *Мой компьютер*, *Корзина* и *Сетевое окружение* являются её потомками. Папка *Мой компьютер* является предком для папок *Диск А*, *Диск С* и т. д. Поиск какого-либо объекта в такой базе данных может оказаться довольно трудоёмким из-за необходимости последовательно проходить несколько предшествующих иерархических уровней.

В сетевой базе данных не накладывается никаких ограничений на связи между объектами: в ней могут быть объекты, имеющие более одного предка. Сетевой способ организации данных реализован во Всемирной паутине глобальной компьютерной сети Интернет.

Наибольшее распространение получили реляционные базы данных. Их мы рассмотрим более подробно.

1.5.2. Реляционные базы данных

В реляционной базе данных (РБД) используется реляционная модель данных, основанная на представлении данных в виде таблиц.

Реляционная БД может состоять из одной или нескольких взаимосвязанных прямоугольных таблиц.

Строка таблицы РБД называется записью, столбец — полем (рис. 1.13).

Имя поля 1	Имя поля 2	Имя поля 3	Имя поля 4

Запись

Поле

Рис. 1.13. Структура таблицы реляционной БД

Запись содержит информацию об одном объекте, описываемом в базе данных: об одном товаре, продаваемом в магазине; об одной книге, имеющейся в библиотеке; об одном сотруднике, работающем на предприятии, и т. п.

Поле содержит информацию о значениях только одной из характеристик (атрибутов, свойств) объекта: названия товара; стоимости товара; количества имеющихся в наличии товаров; названия книги; автора книги; года издания; фамилии, имени, отчества сотрудника; даты рождения; специальности и т. п. Значения полей в одном столбце относятся к одной характеристике объекта.

Поле базы данных имеет имя, тип и длину.

Все имена полей таблицы должны быть разными.

Тип поля определяется типом данных, которые поле содержит. Основные типы полей:

- числовой — для полей, содержащих числовую информацию;
- текстовый — для полей, содержащих всевозможные последовательности символов;
- логический — для полей, данные в которых могут принимать всего два значения: ДА (ИСТИНА, TRUE, 1) и НЕТ (ЛОЖЬ, FALSE, 0);
- дата — для полей, содержащих календарные даты (в нашей стране принято писать день, а потом месяц и год).

Длина поля — это максимальное количество символов, которые могут содержаться в поле.

Для записи структуры таблицы можно применять следующую форму:

ИМЯ_ТАБЛИЦЫ (ИМЯ ПОЛЯ 1, ИМЯ ПОЛЯ 2, ...)

Например, описать однотабличную базу данных «Календарь погоды» можно так:

КАЛЕНДАРЬ_ПОГОДЫ (ДЕНЬ, ТЕМПЕРАТУРА, ВЛАЖНОСТЬ, ДАВЛЕНИЕ, НАПРАВЛЕНИЕ ВЕТРА, СКОРОСТЬ ВЕТРА).

Здесь поле **ДЕНЬ** будет иметь тип «дата», поля **ТЕМПЕРАТУРА**, **ВЛАЖНОСТЬ**, **ДАВЛЕНИЕ**, **СКОРОСТЬ ВЕТРА** — числовой тип; поле **НАПРАВЛЕНИЕ ВЕТРА** — текстовый тип.

В таблице не должно быть совпадающих записей. Иначе говоря, должны быть поле или совокупность полей, значения которых для всех записей разные.

Например, значения поля **ДЕНЬ** базы данных «Календарь погоды» всегда будут разными в разных записях.

В базе данных

УЧЕНИК (ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ДАТА РОЖДЕНИЯ, СЕРИЯ СВИДЕТЕЛЬСТВА О РОЖДЕНИИ, НОМЕР СВИДЕТЕЛЬСТВА О РОЖДЕНИИ, КЛАСС)

наверняка не будут совпадать только значения совокупности таких полей, как **СЕРИЯ СВИДЕТЕЛЬСТВА О РОЖДЕНИИ** и **НОМЕР СВИДЕТЕЛЬСТВА О РОЖДЕНИИ**.

Поле или совокупность полей, значения которых в записях не повторяются (являются уникальными), называют ключом таблицы базы данных.

САМОЕ ГЛАВНОЕ

База данных (БД) — совокупность данных, организованных по определённым правилам, отражающая состояние объектов и их отношений в некоторой предметной области (транспорт, медицина, образование, право и т. д.), предназначенная для хранения во внешней памяти компьютера и постоянного применения. Базу данных можно рассматривать как информационную модель предметной области.

Основными способами организации данных в базах данных являются иерархический, сетевой и реляционный. В реляционных базах данных (РБД) используется реляционная модель данных, основанная на представлении данных в виде таблиц.

Строка таблицы РБД называется записью, столбец — полем. Поле или совокупность полей, значения которых в разных записях не повторяются (являются уникальными), называют ключом таблицы базы данных.

Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Используйте эти материалы при подготовке ответов на вопросы и выполнении заданий.
2. Что такое информационная система? Приведите пример информационной системы.
3. Что такое база данных? Как вы считаете, можно ли в широком смысле назвать базами данных телефонный справочник, записную книжку, библиотечный каталог и другие средства, позволяющие нам хранить данные в упорядоченном виде?
4. Назовите основные способы организации данных в базах данных.
5. Какие базы данных называются реляционными?
6. Что такое запись? Какую информацию она содержит?
7. Что такое поле? Какую информацию оно содержит?
8. Перечислите основные типы полей РБД.
9. Для полей однотабличной базы данных
КОЛЛЕКЦИЯ (КОД, НАЗВАНИЕ ЭКСПОНАТА, АВТОР,
МЕСТО ИЗГОТОВЛЕНИЯ, ГОД ИЗГОТОВЛЕНИЯ,
ФИО ПРЕДЫДУЩЕГО ВЛАДЕЛЬЦА, ДАТА
ПРИБРЕТЕНИЯ, СТОИМОСТЬ ЭКСПОНАТА,
УПОМИНАНИЕ В КАТАЛОГАХ (да/нет))
укажите тип каждого поля.
10. Что такое ключ таблицы базы данных? Что может служить ключом в базе данных КОЛЛЕКЦИЯ (см. задание 9)?
11. Продумайте состав, типы полей и ключ однотабличной базы данных:
 - а) ТУРАГЕНТСТВО;
 - б) ВИДЕОТЕКА;
 - в) АВТОСАЛОН;
 - г) РЕГИОНЫ РФ.



§ 1.6

Система управления базами данных

Ключевые слова:

- СУБД
- таблица
- форма
- запрос
- условие выбора
- отчёт

1.6.1. Что такое СУБД



Программное обеспечение для создания баз данных, хранения и поиска в них необходимой информации называется **системой управления базами данных (СУБД)**.

С помощью СУБД пользователь может:

- создавать структуру базы данных;
- заполнять базу данных информацией;
- редактировать (исправлять, дополнять) структуру и содержание базы данных;
- выполнять сортировку (упорядочение) данных;
- осуществлять поиск информации в базе данных;
- выводить нужную информацию на экран монитора, в файл и на бумажный носитель;
- устанавливать защиту базы данных.

Именно наличие СУБД превращает огромный объём хранимых в компьютерной памяти сведений в мощную справочную систему, способную быстро производить поиск и отбор необходимой нам информации.

1.6.2. Интерфейс СУБД

Существуют СУБД, с помощью которых создаются крупные промышленные информационные системы. Для работы с этими системами нужны специальные знания, в том числе владение специализированными языками программирования.

Для ведения личных баз данных, а также баз данных небольших организаций используются более простые СУБД, работать с которыми могут обычные пользователи. Наиболее распространёнными СУБД такого типа являются Microsoft Access и OpenOffice.org Base. При запуске любой из них на экран выводится окно, имеющее строку заголовка, строку меню, панели инструментов, рабочую область и строку состояния (рис. 1.14).

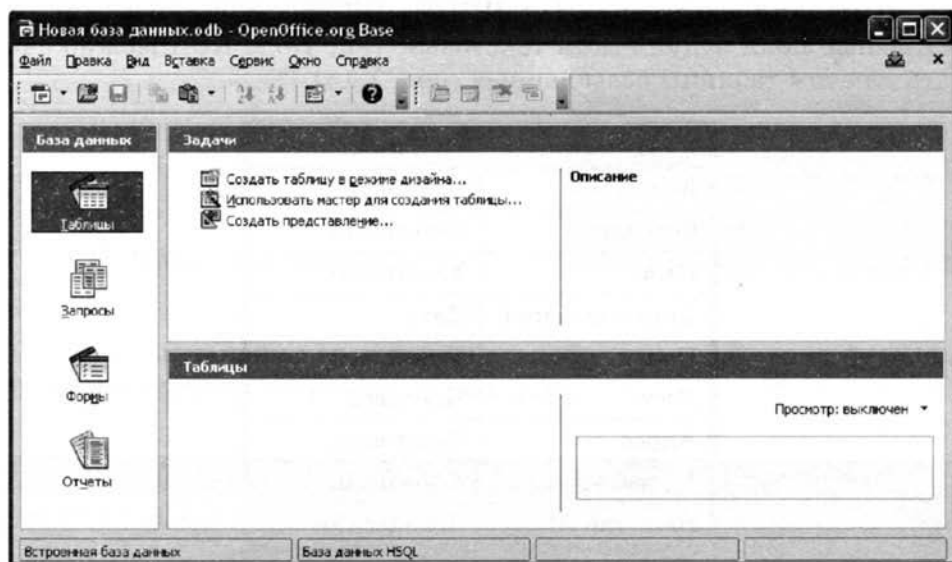


Рис. 1.14. Среда OpenOffice.org Base

Основными объектами СУБД являются таблицы, формы, запросы, отчёты.

Таблицы — это главный тип объектов. С ними вы уже знакомы. В таблицах хранятся данные. Реляционная база данных может состоять из множества взаимосвязанных таблиц.

Формы — это вспомогательные объекты. Они создаются для того, чтобы сделать более удобной работу пользователя при вводе, просмотре и редактировании данных в таблицах.

Запросы — это команды и их параметры, с которыми пользователь обращается к СУБД для поиска данных, сортировки, добавления, удаления и обновления записей.

Отчёты — это документы, сформированные на основе таблиц и запросов и предназначенные для вывода на печать.

1.6.3. Создание базы данных



В качестве примера рассмотрим процесс создания базы данных «Наш класс». Она будет состоять из одной таблицы, имеющей следующую структуру:

СПИСОК (КОД, ФАМИЛИЯ, ИМЯ, ДАТА РОЖДЕНИЯ, ПОЛ, РОСТ, АДРЕС, УВЛЕЧЕНИЕ, НАЛИЧИЕ ПК).

Поля КОД и РОСТ будут числовыми; поле ДАТА РОЖДЕНИЯ будет иметь тип дата; поле НАЛИЧИЕ ПК будет логическим; все остальные поля будут иметь текстовый тип. Поле КОД можно считать ключом таблицы базы данных (рис. 1.15).

Имя поля	Тип поля
Код	Числовой
Фамилия	Текстовый
Имя	Текстовый
Дата рождения	Дата
Пол	Текстовый
Рост	Числовой
Адрес	Текстовый
Увлечение	Текстовый
Наличие ПК	Логический

Рис. 1.15. Имена и типы полей БД «Наш класс»



Создание базы данных начинается с открытия файла, в котором она будет храниться. Для этого нужно после запуска программы OpenOffice.org Base следовать указаниям мастера баз данных:

- 1) создать новую базу данных;
- 2) зарегистрировать базу данных (указать путь и имя файла).

Далее следует описать структуру таблицы (указать имена и типы всех полей) и ввести данные в таблицу.

Данные можно вводить непосредственно в таблицу (рис. 1.16), а можно создать для этого специальный шаблон — форму (рис. 1.17).

Код	Фамилия	Имя	Дата рождения	Пол	Рост	Адрес	Увлечение	Наличие ПК
								<input type="checkbox"/>

Рис. 1.16. Таблица для ввода данных

Код

Фамилия

Имя

Дата рождения

Пол

Рост

Адрес

Увлечение

Наличие ПК

Код

Фамилия

Имя

Дата рождения

Пол

Рост

Адрес

Увлечение

Наличие ПК

Рис. 1.17. Формы для ввода данных

После выполнения всех перечисленных выше действий будет получен следующий результат — рис. 1.18.

Список - Наш класс.odf - OpenOffice.org Table Data View

Файл Правка Вид Вставка Сервис Опции Справка

Код	Фамилия	Имя	Дата рождения	Пол	Рост	Адрес	Увлечение	Наличие ПК
1	Гриднев	Михаил	23.05.96	м	152,0	Первомайская 16-8	футбол	<input checked="" type="checkbox"/>
2	Дементьева	Анастасия	08.04.96	ж	154,0	Школьная 5-2	танцы	<input checked="" type="checkbox"/>
3	Жихорев	Алексей	12.12.96	м	160,0	Садская 10-14	футбол	<input checked="" type="checkbox"/>
4	Кочергина	Ольга	01.11.95	м	164,0	Садская 10-56	плавание	<input type="checkbox"/>
5	Новиков	Михаил	12.08.96	м	158,0	Школьная 12-24	футбол	<input type="checkbox"/>
6	Патрина	Ирина	09.05.96	ж	154,0	Первомайская 2-10	музыка	<input checked="" type="checkbox"/>
7	Патрина	Ольга	09.05.96	ж	156,0	Первомайская 2-10	танцы	<input checked="" type="checkbox"/>
8	Торопкин	Сергей	14.04.96	м	162,0	Первомайская 12-3	плавание	<input checked="" type="checkbox"/>
9	Шашков	Иван	13.11.95	м	156,0	Школьная 4-4	танцы	<input checked="" type="checkbox"/>
10	Юсуфова	Диана	01.09.95	ж	168,0	Школьная 3-15	музыка	<input checked="" type="checkbox"/>

Запись 11 из 11

Рис. 1.18. Таблица «Список» базы данных «Наш класс»

Созданная и сохранённая база данных в дальнейшем может быть открыта для добавления новых записей, исправления и удаления существующих, изменения содержимого отдельных полей и структуры всей таблицы.

Данные из таблиц можно упорядочить по некоторому признаку. Например, фамилии учеников в классном журнале записывают в алфавитном порядке; телепередачи в программе — в соответствии со временем их выхода в эфир; уроки в расписании — по возрастанию их порядковых номеров.

Упорядочение данных по возрастанию или убыванию значений некоторого признака называют **сортировкой**. Для выполнения сортировки указывают имя поля (имена полей), по которому будет произведена сортировка, и её порядок (возрастание или убывание значений поля).

1.6.4. Запросы на выборку данных

После того как база данных создана, её можно использовать в качестве справочной системы.

Таблица, содержащая интересующие пользователя сведения, извлечённые из базы данных, называется **справкой** или **запросом**; она содержит только те записи и их поля, которые содержатся в запросах на выборку данных, удовлетворяющих заданным условиям (условиям выбора).

В командах СУБД условия выбора записываются в форме логических выражений, сформированных из высказываний на естественном языке (табл. 1.6).

В логических выражениях имена полей базы данных связываются с определёнными значениями этих полей операциями отношений:

- = равно;
- <> не равно;
- < меньше;
- > больше;
- <= меньше или равно (не больше);
- >= больше или равно (не меньше).

На уроках математики вы применяете эти операции, составляя и решая числовые равенства, неравенства и их системы.

Операции отношений применимы и к текстовым полям. Их сравнение построено на лексикографическом принципе: из двух слов меньшим считается то слово, первая буква которого идёт по алфавиту раньше; если первые несколько букв двух слов одинаковы, то сравнение производится по первой различающейся букве; если более короткое слово совпадает с началом более длинного слова, то первое считается меньшим.

Таблица 1.6

Условия выбора — простые логические выражения

№	Высказывание	Логическое выражение	Номер записи	Значение
1	Рост ученика не превышает 160 см	РОСТ<=160	1	Истина
			4	Ложь
2	Ученик увлекается футболом	УВЛЕЧЕНИЕ='футбол'	1	Истина
			2	Ложь
3	Фамилия ученика — Патрина	ФАМИЛИЯ='Патрина'	6	Истина
			1	Ложь
4	Ученик не увлекается танцами	УВЛЕЧЕНИЕ<>'танцы'	2	Ложь
			1	Истина
5	Ученик родился в 1996 году или позже	ДАТА>#31.12.95#	8	Истина
			10	Ложь
6	Ученик имеет персональный компьютер	НАЛИЧИЕ ПК=1	7	Истина
			9	Ложь

Значение поля текстового типа и некоторая текстовая величина равны, если они содержат одинаковое количество символов и все их символы, стоящие в позициях с одинаковыми номерами, совпадают.

При сравнении текстовых величин следует иметь в виду, что пробел — это тоже символ, хотя он и «меньше» любой буквы.

Сравнение дат построено так: одна дата считается меньше другой, если она относится к более раннему времени. Например, истинными будут следующие отношения:

$$\begin{aligned} 01.11.95 < 02.11.95; \\ 29.11.95 < 02.12.95; \\ 29.11.95 < 01.11.96. \end{aligned}$$

Условия выбора могут задаваться не только простыми, но и сложными логическими выражениями, содержащими логические операции (табл. 1.7). С основными логическими операциями И, ИЛИ, НЕ вы познакомились в 8 классе.

Условия выбора — сложные логические выражения

№	Высказывание	Логическое выражение	Номер записи	Значение
1	Рост ученика больше 160 см, и ученик увлекается плаванием	РОСТ>160 И УВЛЕЧЕНИЕ='плавание'	4	Истина
			10	Ложь
2	Рост ученика больше 160 см или ученик увлекается плаванием	РОСТ>160 ИЛИ УВЛЕЧЕНИЕ='плавание'	10	Истина
			1	Ложь
3	День рождения Ольги не 09.05.96	ИМЯ='Ольга' И ДАТА#09.05.96#	4	Истина
			7	Ложь

С помощью запросов пользователь может быстро найти в базе данных и вывести на экран компьютера интересующую его информацию. Но для решения большинства практических задач найденную информацию необходимо представить в определённой форме и подготовить к выводу на печать. Этот этап работы называется подготовкой отчёта.

САМОЕ ГЛАВНОЕ

Программное обеспечение для создания баз данных, хранения и поиска в них необходимой информации называется системой управления базами данных (СУБД).

Основными объектами СУБД являются таблицы, формы, запросы, отчёты.

С помощью запросов на выборку данных, удовлетворяющих заданным условиям (условиям выбора), пользователь получает из базы данных только те записи и их поля, которые ему нужны. В командах СУБД условия выбора записываются в форме логических выражений.



Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Используйте эти материалы при подготовке ответов на вопросы и выполнении заданий.

2. Что такое СУБД?
3. Какая СУБД установлена на компьютерах в вашем классе?
4. С чего начинается создание БД?
5. Перечислите основные объекты СУБД. Какие функции они выполняют?
6. Ниже в табличной форме представлены характеристики ноутбуков, имеющих в продаже в компьютерном салоне:



№	Название	Жёсткий диск (Гб)	Оперативная память (Мб)
4	Sony Vaio AW2X	500	4096
3	Lenovo S10e	250	3072
5	Asus F70SL	250	2048
1	Acer E525	160	2048
2	Samsung NC20	160	1024
6	Roverbook V212	120	1024

- а) Какую строку будет занимать запись, содержащая сведения о ноутбуке **Asus F70SL**, после сортировки данных по возрастанию значений поля **НАЗВАНИЕ**?
- б) Какую строку будет занимать запись, содержащая сведения о ноутбуке **Asus F70SL**, после сортировки данных по убыванию значений поля **ЖЁСТКИЙ ДИСК**?
- в) Какую строку будет занимать запись, содержащая сведения о ноутбуке **Asus F70SL**, после сортировки данных сначала по убыванию значений поля **ОПЕРАТИВНАЯ ПАМЯТЬ**, затем по возрастанию значений поля **ЖЁСТКИЙ ДИСК**?
7. Как будет выглядеть список (фамилия, имя) учеников после сортировки данных по возрастанию значений поля **ДАТА РОЖДЕНИЯ** базы данных «Наш класс» (рис. 1.18)?
8. Укажите все записи базы данных «Наш класс» (рис. 1.18), для которых будут истинными простые логические выражения 1–6 (табл. 1.6).
9. Укажите все записи базы данных «Наш класс» (рис. 1.18), для которых будут истинными сложные логические выражения 1–3 (табл. 1.7).
10. Какова цель запроса на выборку?





11. Ниже в табличной форме представлен фрагмент базы данных с годовыми оценками учащихся:

Фамилия	Пол	Алгебра	Геометрия	Информатика	Физика
Алексеева	Ж	3	3	4	3
Воронин	М	4	4	4	3
Ильин	М	4	3	3	4
Костин	М	5	4	5	4
Сизова	Ж	5	5	5	4
Школина	Ж	5	5	5	5

Сколько записей в данном фрагменте удовлетворяет следующему условию?

- а) АЛГЕБРА > 3 И ИНФОРМАТИКА > 4 И ПОЛ = 'М'
- б) (АЛГЕБРА > 4 ИЛИ ИНФОРМАТИКА > 4) И ПОЛ = 'Ж'
- в) ФИЗИКА = 3 ИЛИ АЛГЕБРА = 3 ИЛИ ГЕОМЕТРИЯ = 3 ИЛИ ИНФОРМАТИКА = 3
- г) (ФИЗИКА = 3 ИЛИ АЛГЕБРА = 3) И (ГЕОМЕТРИЯ = 3 ИЛИ ИНФОРМАТИКА = 3)



12. Ниже в табличной форме представлен фрагмент базы данных с результатами олимпиады по информатике:

Фамилия	Пол	Задача1	Задача2	Задача3	Сумма
Жариков	М	15	20	25	60
Костин	М	10	10	10	30
Кузнецов	М	20	25	30	75
Михайлова	Ж	25	20	10	55
Сизова	Ж	30	30	30	90
Старовойтова	Ж	20	25	25	70
Школина	Ж	30	25	25	80

Сколько записей в данном фрагменте удовлетворяет следующему условию?

- а) ПОЛ = 'М' И СУММА > 55
- б) (ЗАДАЧА1 < ЗАДАЧА2) И (ЗАДАЧА2 < ЗАДАЧА3)
- в) ЗАДАЧА1 = 30 ИЛИ ЗАДАЧА2 = 30 ИЛИ ЗАДАЧА3 = 30
- г) ЗАДАЧА1 = 30 И ЗАДАЧА2 = 30 И ЗАДАЧА3 = 30

Тестовые задания для самоконтроля



1. Выберите верное утверждение:

- а) Один объект может иметь только одну модель
- б) Разные объекты не могут описываться одной моделью
- в) Электрическая схема — это модель электрической цепи
- г) Модель полностью повторяет изучаемый объект

2. Выберите неверное утверждение:

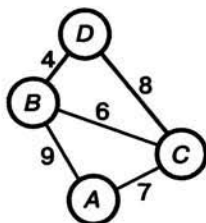
- а) Натурные модели — реальные объекты, в уменьшенном или увеличенном виде воспроизводящие внешний вид, структуру или поведение моделируемого объекта
- б) Информационные модели описывают объект-оригинал на одном из языков кодирования информации
- в) Динамические модели отражают процессы изменения и развития объектов во времени
- г) За основу классификации моделей может быть взята только предметная область, к которой они относятся

3. Какие признаки объекта должны быть отражены в информационной модели ученика, позволяющей получать следующие сведения: возраст учеников, увлекающихся плаванием; количество девочек, занимающихся танцами; фамилии и имена учеников старше 14 лет?

- а) имя, фамилия, увлечение
- б) имя, фамилия, пол, пение, плавание, возраст
- в) имя, увлечение, пол, возраст
- г) имя, фамилия, пол, увлечение, возраст

4. Выберите элемент информационной модели учащегося, существенный для выставления ему оценки за контрольную работу по информатике:
 - а) наличие домашнего компьютера
 - б) количество правильно выполненных заданий
 - в) время, затраченное на выполнение контрольной работы
 - г) средний балл за предшествующие уроки информатики
5. Замена реального объекта его формальным описанием — это:
 - а) анализ
 - б) моделирование
 - в) формализация
 - г) алгоритмизация
6. Выберите знаковую модель:
 - а) рисунок
 - б) схема
 - в) таблица
 - г) формула
7. Выберите образную модель:
 - а) фотография
 - б) схема
 - в) текст
 - г) формула
8. Выберите смешанную модель:
 - а) фотография
 - б) схема
 - в) текст
 - г) формула
9. Описания предметов, ситуаций, событий, процессов на естественных языках — это:
 - а) словесные модели
 - б) логические модели
 - в) геометрические модели
 - г) алгебраические модели

10. Модели, реализованные с помощью систем программирования, электронных таблиц, специализированных математических пакетов и программных средств для моделирования, называются:
- а) математическими моделями
 - б) компьютерными моделями
 - в) имитационными моделями
 - г) экономическими моделями
11. Файловая система персонального компьютера наиболее адекватно может быть описана в виде:
- а) математической модели
 - б) табличной модели
 - в) натурной модели
 - г) иерархической модели
12. Графической моделью иерархической системы является:
- а) цепь
 - б) сеть
 - в) генеалогическое дерево
 - г) дерево
13. Расписание движения электропоездов может рассматриваться как пример:
- а) табличной модели
 - б) графической модели
 - в) имитационной модели
 - г) натурной модели
14. Какая тройка понятий находится в отношении «объект – натурная модель — информационная модель»?
- а) человек — анатомический скелет — манекен
 - б) человек — медицинская карта — фотография
 - в) автомобиль — рекламный буклет с техническими характеристиками автомобиля — атлас автомобильных дорог
 - г) автомобиль — игрушечный автомобиль — техническое описание автомобиля
15. На схеме изображены дороги между населёнными пунктами А, В, С, D и указаны протяжённости этих дорог.



Определите, какие два пункта наиболее удалены друг от друга. Укажите длину кратчайшего пути между ними.

- а) 17
- б) 15
- в) 13
- г) 9



16. Населённые пункты A, B, C, D соединены дорогами. Время проезда на автомобиле из города в город по соответствующим дорогам указано в таблице:

	A	B	C	D
A	×	2	4	4
B	2	×	5	3
C	4	5	×	1
D	4	3	1	×

Турист, выезжающий из пункта A , хочет посетить все города за кратчайшее время. Укажите соответствующий маршрут.

- а) $ABCD$
- б) $ACBD$
- в) $ADCB$
- г) $ABDC$



17. В школе учатся четыре ученика — Андреев, Иванов, Петров, Сидоров, имеющие разные увлечения. Один из них увлекается теннисом, другой — бальными танцами, третий — живописью, четвёртый — пением. О них известно:

- Иванов и Сидоров присутствовали на концерте хора, когда пел их товарищ;

- Петров и теннисист позировали художнику;
- теннисист дружит с Андреевым и хочет познакомиться с Ивановым.

Чем увлекается Андреев?

- а) теннисом
- б) живописью
- в) танцами
- г) пением

18. Два игрока играют в следующую игру. Перед ними лежат три кучки камней, в первой из которых 2 камня, во второй — 3 камня, в третьей — 4 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или удваивает число камней в какой-то куче, или добавляет по два камня в каждую из куч. Выигрывает игрок, после хода которого либо в одной из куч становится не менее 15 камней, либо общее число камней во всех трёх кучах становится не менее 25. Кто выигрывает при безошибочной игре обоих игроков?



- а) игрок, делающий первый ход
- б) игрок, делающий второй ход
- в) каждый игрок имеет одинаковый шанс на победу
- г) для этой игры нет выигрышной стратегии

19. База данных — это:

- а) набор данных, собранных на одном диске
- б) таблица, позволяющая хранить и обрабатывать данные и формулы
- в) прикладная программа для обработки данных пользователя
- г) совокупность данных, организованных по определённым правилам, предназначенная для хранения во внешней памяти компьютера и постоянного применения

20. Какая база данных основана на табличном представлении информации об объектах?

- а) иерархическая
- б) сетевая
- в) распределённая
- г) реляционная

21. Строка таблицы, содержащая информацию об одном конкретном объекте, — это:
- а) поле
 - б) запись
 - в) отчёт
 - г) форма
22. Столбец таблицы, содержащий определённую характеристику объекта, — это:
- а) поле
 - б) запись
 - в) отчёт
 - г) ключ
23. Системы управления базами данных используются для (выберите наиболее полный ответ):
- а) создания баз данных, хранения и поиска в них необходимой информации
 - б) сортировки данных
 - в) организации доступа к информации в компьютерной сети
 - г) создания баз данных
24. Какое из слов НЕ является названием базы данных?
- а) Microsoft Access
 - б) OpenOffice.org Base
 - в) OpenOffice.org Writer
 - г) FoxPro
25. В табличной форме представлен фрагмент базы данных:

№	Наименование товара	Цена	Количество
1	Монитор	7654	20
2	Клавиатура	1340	26
3	Мышь	235	10
4	Принтер	3770	8
5	Колонки акустические	480	16
6	Сканер планшетный	2880	10

На какой позиции окажется товар «Сканер планшетный», если произвести сортировку данных по возрастанию столбца КОЛИЧЕСТВО?

- а) 5
- б) 2
- в) 3
- г) 6

26. В табличной форме представлен фрагмент базы данных:



Наименование	Цена	Продано
Карандаш	5	60
Линейка	18	7
Папка	20	32
Ручка	25	40
Тетрадь	15	500

Сколько записей в данном фрагменте удовлетворяет условию ЦЕНА > 20 ИЛИ ПРОДАНО < 50?

- а) 1
- б) 2
- в) 3
- г) 4

Для проверки знаний и умений по теме «Моделирование и формализация» вы можете воспользоваться интерактивным тестом к главе 1, содержащимся в электронном приложении к учебнику.



Глава 2

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

§ 2.1

Решение задач на компьютере

Ключевые слова:

- постановка задачи
- формализация
- алгоритмизация
- программирование
- отладка и тестирование

Чтобы решать задачи на компьютере, необходимо владеть языком программирования, обладать знаниями в области информационного моделирования и алгоритмизации.

2.1.1. Этапы решения задачи на компьютере

Решение задачи с использованием компьютера включает в себя этапы, показанные на рис. 2.1.

На *первом этапе* обычно осуществляется постановка задачи, происходит осознание её условия. При этом должно быть чётко определено, что дано (какие исходные данные известны, какие данные допустимы) и что требуется найти в решаемой задаче. Также должны быть чётко выделены существенные свойства рассматриваемого объекта, указаны связи между исходными данными и результатами.

На *втором этапе* описательная информационная модель формализуется, т. е. записывается с помощью некоторого формального языка.

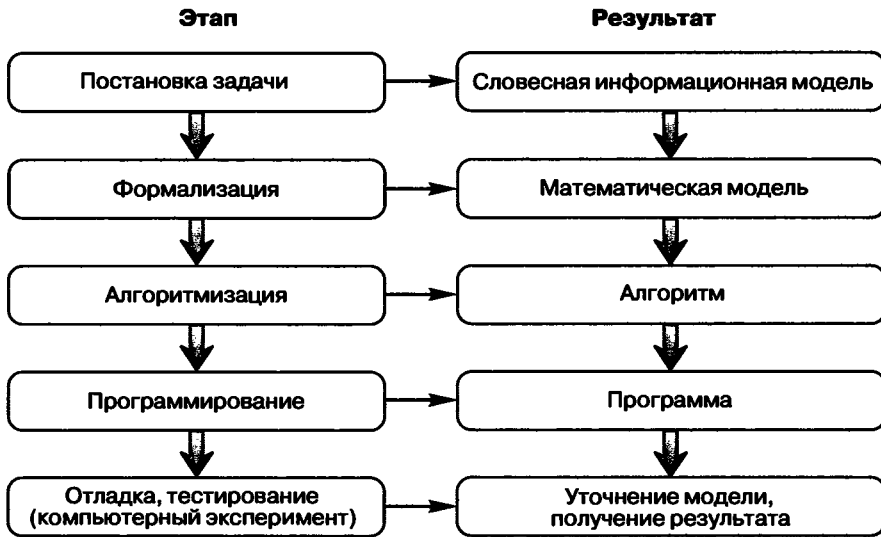


Рис. 2.1. Этапы решения задачи на компьютере

Для этого требуется:

- понять, к какому классу принадлежит рассматриваемая задача;
- записать известные связи между исходными данными и результатами с помощью математических соотношений;
- выбрать наиболее подходящий способ для решения задачи.

На *третьем этапе* осуществляется построение алгоритма — чёткой инструкции, задающей необходимую последовательность действий для решения задачи. Алгоритм чаще всего представляется в форме блок-схемы ввиду её наглядности и универсальности.

На *четвёртом этапе* алгоритм записывается на одном из языков программирования. Вы учитесь записывать программы на языке Паскаль.

На *пятом этапе* осуществляется отладка и тестирование программы. Этап отладки и тестирования также называют компьютерным экспериментом.

Отладка программы — это процесс проверки работоспособности программы и исправления обнаруженных при этом ошибок. Ошибки могут быть связаны с нарушением правил записи программы на конкретном языке программирования. Их программисту помогает найти используемая система программирования; она выдаёт на экран сообщения о выявленных ошибках.

Проверка правильности разработанной программы осуществляется с помощью тестов. Тест — это конкретный вариант значений исходных данных, для которого известен ожидаемый результат.

О правильности разработанной программы свидетельствует также соответствие полученных данных экспериментальным фактам, теоретическим положениям и т. д. При этом может возникнуть необходимость уточнить разработанную математическую модель, полнее учесть особенности изучаемого объекта или процесса. По уточнённой математической модели снова составляется программа, анализируются результаты её выполнения. Так продолжается до тех пор, пока полученные результаты не будут достаточно точно соответствовать изучаемому объекту.

2.1.2. Задача о пути торможения автомобиля



Рассмотрим последовательность прохождения этапов решения задачи на компьютере (см. рис. 2.1) на примере простой задачи.

Водитель автомобиля, движущегося с некоторой постоянной скоростью, увидев красный свет светофора, нажал на тормоз. После этого скорость автомобиля стала уменьшаться каждую секунду на 5 метров. Требуется найти расстояние, которое автомобиль пройдёт до полной остановки.

Первый этап. Дано:

v_{0x} — начальная скорость;

v_x — конечная скорость (равна нулю, так как автомобиль остановился);

a_x — ускорение (равно -5 м/с).

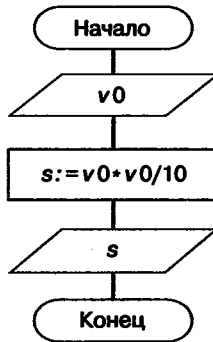
Требуется найти: s_x — расстояние, которое автомобиль пройдёт до полной остановки.

Второй этап. В данной ситуации мы имеем дело с прямолинейным равноускоренным движением тела. Формула для перемещения при этом имеет вид:

$$s_x = \frac{v_{0x}(v_x - v_{0x})}{a_x} + \frac{a_x}{2} \left(\frac{v_x - v_{0x}}{a_x} \right)^2.$$

Упростим эту формулу с учётом того, что конечная скорость равна нулю: $s_x = -\frac{v_{0x}^2}{2a_x}$. При $a_x = -5$ м/с получим: $s_x = \frac{v_{0x}^2}{10}$.

Третий этап. Представим алгоритм решения задачи в виде блок-схемы:



Четвёртый этап. Запишем данный алгоритм на языке программирования Паскаль:

```

program n_1;
  var v0, s: real;
begin
  writeln('Вычисление длины пути торможения автомобиля');
  write('Введите начальную скорость (м/с)>>');
  readln (v0);
  s:=v0*v0/10;
  writeln ('До полной остановки автомобиль пройдёт ',
          s:8:4, ' м. ');
end.
  
```

Пятый этап. Протестировать составленную программу можно, используя информацию, что при скорости 72 км/ч с начала торможения до полной остановки автомобиль проходит 40 метров.

Выполнив программу несколько раз при различных исходных данных, можно сделать вывод: чем больше начальная скорость автомобиля, тем большее расстояние он пройдёт с начала торможения до полной остановки.

Применяя компьютер для решения задач, всегда следует помнить, что наряду с огромным быстродействием и абсолютной исполнительностью у компьютера отсутствуют интуиция и чувство здравого смысла, и он способен решать только ту задачу, программу решения которой ему подготовил человек.

САМОЕ ГЛАВНОЕ

Этапы решения задачи с использованием компьютера:

- 1) постановка задачи;
- 2) формализация;
- 3) алгоритмизация;
- 4) программирование;
- 5) компьютерный эксперимент.

Для решения задач на компьютере необходимо владеть языком программирования, обладать знаниями в области информационного моделирования и алгоритмизации.



Вопросы и задания

1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Какими слайдами вы могли бы дополнить презентацию?
2. Перечислите основные этапы решения задачи с использованием компьютера.
3. Что происходит на этапе постановки задачи? Что является результатом этого этапа?
4. Что происходит на этапе формализации? Что является результатом этого этапа?
5. Что происходит на этапе алгоритмизации? Что является результатом этого этапа?
6. Что происходит на этапе программирования? Что является результатом этого этапа?
7. Что происходит на этапе компьютерного эксперимента? Что является результатом этого этапа?
8. Какой этап решения задачи на компьютере, по вашему мнению, является наиболее трудоёмким?
9. Как вы считаете, по силам ли одному специалисту реализация всех этапов решения сложной практической задачи? Обоснуйте свою точку зрения.
10. Как правило, сложные практические задачи решаются большими коллективами разработчиков. Отдельные группы в этих коллективах специализируются на выполнении одного или нескольких этапов решения задачи. Нужно ли в таком случае им



иметь представление обо всех этапах решения задачи с использованием компьютера? Обоснуйте свою точку зрения.

11. Может ли пригодиться в жизни представление об этапах решения задачи с использованием компьютера? Обоснуйте свою точку зрения.



12. Уличный продавец газет получает a рублей с продажи каждой из первых 50 газет. С продажи каждой из последующих газет он получает на 20% больше.

Разработайте программу, которая вычислит заработок продавца, если он продаст за день 200 газет. Зафиксируйте свои действия на каждом из этапов решения этой задачи.

13. В аэробусе, вмещающем 160 пассажиров, три четверти мест находятся в салонах экономического класса и одна четверть мест — в салоне бизнес-класса. Стоимость билета в салоне бизнес-класса составляет x рублей, что в два раза выше стоимости билета в салонах экономического класса.

Разработайте программу, которая вычислит сумму денег, полученную авиакомпанией от продажи билетов на этот рейс, если известно, что остались нераспроданными a билетов бизнес-класса и b билетов экономического класса. Выделите все этапы решения этой задачи и опишите свои действия на каждом из них.

§ 2.2

Одномерные массивы целых чисел

Ключевые слова:

- массив
- описание массива
- заполнение массива
- вывод массива
- обработка массива
- последовательный поиск
- сортировка

До сих пор мы работали с простыми типами данных. При решении практических задач данные часто объединяются в различные структуры данных, например в массивы. В языках программирования массивы используются для реализации таких структур данных, как последовательности¹ (одномерные массивы) и таблицы (двумерные массивы).



Упорядоченное множество однотипных переменных (элементов массива), которым можно присвоить общее имя, различающихся номерами (индексами), называют **массивом**.

Мы будем рассматривать одномерные массивы.

Решение разнообразных задач, связанных с обработкой массивов, базируется на использовании таких типовых алгоритмов, как:

- суммирование значений элементов массива;
- поиск элемента с заданными свойствами;
- сортировка массива.

¹ Например, числовые последовательности в математике.

2.2.1. Описание массива

Перед использованием в программе массив должен быть описан, т. е. должно быть указано имя массива, количество элементов массива и их тип. Это необходимо для того, чтобы выделить участок памяти нужного размера для хранения массива. Общий вид описания одномерного массива:

```
var <имя_массива>: array [<мин_знач_индекса> ..
    <макс_знач_индекса>] of <тип_элементов>;
```

Пример

```
var a: array [1..10] of integer;
```

Здесь описан массив *a* из 10 целочисленных значений. При выполнении этого оператора в памяти компьютера будет выделено место для хранения десяти целочисленных переменных.

Массив, элементы которого имеют заданные начальные значения, может быть описан в разделе описания констант:

```
const b: array [1..5] of integer = (1, 2, 3, 5, 7);
```

В этом случае не просто выделяются последовательные ячейки памяти — в них сразу же заносятся соответствующие значения.

2.2.2. Заполнение массива

Заполнять массив можно либо вводя значение каждого элемента с клавиатуры, либо присваивая элементам некоторые значения в программе. При этом может использоваться цикл с параметром.

Например, для ввода с клавиатуры значений элементов описанного выше массива *a* используется следующий цикл с параметром:

```
for i:=1 to 10 do read (a[i]);
```

Задавать значения элементов массива можно с помощью оператора присваивания. Например:

```
for i:=1 to 10 do a[i]:=i;
```

В следующем фрагменте программы организовано заполнение целочисленного массива *a*, состоящего из 10 элементов, случайными числами, значения которых изменяются в диапазоне от 0 до 99:

```
randomize;
for i:=1 to 10 do a[i]:=random(100);
```



2.2.3. Вывод массива

Во многих случаях бывает полезно вывести значения элементов массива на экран. Так, если значения массива генерировались случайным образом, то необходимо знать, каков исходный массив. Также нужно знать, каким стал массив после обработки.



Значения элементов массива можно вывести в строку, разделив их пробелом:

```
for i:=1 to 10 do write (a[i], ' ');
```

Более наглядным является следующий вариант вывода с комментариями:

```
for i:=1 to 10 do writeln ('a[' , i, ']=' , a[i]);
```

На основании рассмотренных примеров запишем программу, в которой осуществляется: заполнение целочисленного массива a , состоящего из 10 элементов, случайными числами, значения которых изменяются в диапазоне от 0 до 99; вывод массива a на экран.

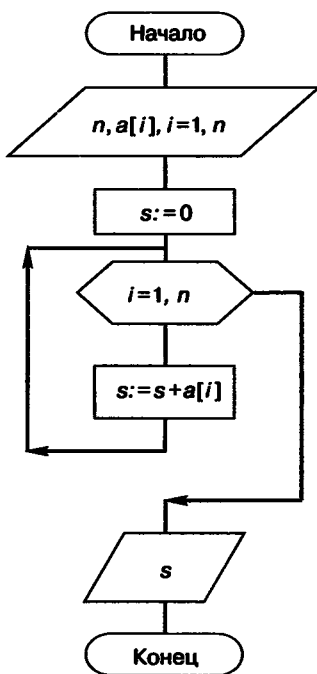
<code>program n_2;</code>	Заголовок программы
<code>var</code> <code> i: integer;</code> <code> a: array [1..10] of integer;</code>	Блок описания переменных
<code>begin</code> <code> randomize;</code> <code> for i := 1 to 10 do</code> <code> a[i]:= random(100);</code> <code> for i := 1 to 10 do</code> <code> writeln ('a[' , i, ']=' , a[i])</code> <code>end.</code>	Программный блок
	----- Заполнение массива -----
	----- Вывод массива -----

2.2.4. Вычисление суммы элементов массива



Пример. В некотором населённом пункте n домов. Известно, сколько людей проживает в каждом из домов. Составим алгоритм подсчёта количества жителей населённого пункта.

Исходные данные (количество жильцов) здесь представлены с помощью одномерного массива a , содержащего n элементов: $a[1]$ — количество жильцов дома 1, $a[2]$ — количество жильцов дома 2, ..., $a[n]$ — количество жильцов дома n . В общем случае $a[i]$ — количество жильцов дома i , где i принимает целочисленные значения от 1 до n ($i = 1, n$). Результат работы алгоритма обозначен через s .



Суммирование элементов массива осуществляется по тому же принципу, что и суммирование значений простых переменных: за счёт поочерёдного добавления слагаемых:

- 1) определяется ячейка памяти (переменная s), в которой будет последовательно накапливаться результат суммирования;
- 2) переменной s присваивается начальное значение 0 — число, не влияющее на результат сложения;
- 3) для каждого элемента массива из переменной s считывается её текущее значение и складывается со значением элемента массива; полученный результат присваивается переменной s .

Описанный процесс наглядно можно изобразить так:

$s := 0$	$s = 0$
$s := s + a[1]$	$s = 0 + a[1]$
$s := s + a[2]$	$s = 0 + a[1] + a[2]$
$s := s + a[3]$	$s = 0 + a[1] + a[2] + a[3]$
...	...
$s := s + a[n]$	$s = 0 + a[1] + a[2] + a[3] + \dots + a[n]$

Запишем соответствующую программу на языке Паскаль.

```
program n_3;
```

```
const
```

```
  n=20;
```

```
var
```

```
  i, s: integer;
```

```
  a: array [1..n] of integer;
```

```
begin
```

```
  randomize;
```

```
  for i := 1 to n do
```

```
  begin
```

```
    a[i]:= random(100)+50;
```

```
    writeln ('a[' , i, ']=' , a[i])
```

```
  end;
```

```
  s := 0;
```

```
  for i :=1 to n do
```

```
    s := s + a[i];
```

```
  writeln ('s=' , s)
```

```
end.
```

Заголовок программы

Блок описания
используемых данных

Программный блок

Заполнение и вывод
массива

Вычисление суммы
элементов массива

Вывод результата



Сравните программы n_2 и n_3. Выделите в них общие блоки. Обратите внимание на различия.

2.2.5. Последовательный поиск в массиве

В программировании поиск — одна из наиболее часто встречающихся задач невычислительного характера.

Можно выделить следующие типовые задачи поиска:

- 1) найти наибольший (наименьший) элемент массива;
- 2) найти элемент массива, значение которого равно заданному значению.

Для решения таких задач в программе необходимо организовать последовательный просмотр элементов массива и сравнение значения очередного просматриваемого элемента с неким образцом.



Рассмотрим подробно решение задач первого типа: нахождение наибольшего (наименьшего) элемента.

Представим себе одномерный массив в виде стопки карточек, на каждой из которых написано число. Тогда идея поиска наибольшего элемента массива может быть представлена следующим образом:

- 1) возьмём верхнюю карточку (первый элемент массива), запомним имеющееся на карточке число

(запишем его мелом на доске) как наибольшее из просмотренных; уберём карточку в сторону;

2) возьмём следующую карточку; сравним числа, записанные на карточке и на доске; если число на карточке больше, то сотрём число, записанное на доске, и запишем там то же число, что и на карточке; если же новое число не больше, то на доске оставим имеющуюся запись; уберём карточку в сторону;

3) повторим действия, описанные в п. 2, для всех оставшихся карточек в стопке.

В итоге на доске будет записано самое большое значение элемента просмотренного массива.

Так как доступ к значению элемента массива осуществляется по его индексу, то при организации поиска наибольшего элемента в одномерном массиве можно искать его индекс. Обозначим искомый индекс *imax*. Тогда описанный выше алгоритм в сформированном нами массиве *a* на языке Паскаль можно записать так:

<code>program n_4;</code>	Заголовок программы
<code>var</code> <code> i, imax: integer;</code> <code> a: array [1..10] of integer;</code>	Блок описания переменных
<code>begin</code> <code> randomize;</code> <code> for i := 1 to 10 do</code> <code> begin</code> <code> a[i] := random(100);</code> <code> writeln ('a[' , i, ']=', a[i])</code> <code> end;</code> <code> imax := 1;</code> <code> for i := 2 to 10 do</code> <code> if a[i] > a[imax] then imax := i;</code> <code> writeln ('Наибольший элемент</code> <code> массива', a[imax])</code> <code>end.</code>	Программный блок Заполнение и вывод массива Поиск наибольшего элемента массива Вывод результата

Если в массиве несколько элементов, значения которых равны максимальному значению, то данная программа найдёт первый из них (первое вхождение). Подумайте, что следует изменить в программе, чтобы в ней находился последний из максимальных элементов. Как следует преобразовать программу, чтобы с её помощью можно было найти минимальный элемент массива?



Результатом решения задачи второго типа (нахождение элемента массива, значение которого равно заданному значению) может быть:

- n — индекс элемента массива такой, что $a[n] = x$, где x — заданное число;
 - сообщение о том, что искомого элемента в массиве не обнаружено.
- Программа поиска в сформированном нами массиве a значения, равного x , может выглядеть так:

<code>program n_5;</code>	Заголовок программы
<code>var</code> <code> i, n, x: integer;</code> <code> a: array [1..10] of integer;</code>	Блок описания переменных
<code>begin</code> <code> randomize;</code> <code> for i := 1 to 10 do</code> <code> begin</code> <code> a[i]:= random(100);</code> <code> writeln ('a[' , i, ']=', a[i])</code> <code> end;</code> <code> writeln ('x=');</code> <code> readln (x);</code> <code> n := 0;</code> <code> for i := 1 to 10 do</code> <code> if a[i] = x then n := i;</code> <code> if n = 0</code> <code> then writeln ('Элемента со значением,</code> <code> равным заданному,</code> <code> в массиве нет')</code> <code> else writeln ('Индекс элемента,</code> <code> равного заданному, ', n)</code> <code>end.</code>	Программный блок Заполнение и вывод массива Ввод значения x Поиск в массиве элемента, равного x Вывод результата

В этой программе последовательно просматриваются все элементы массива. Если в массиве несколько элементов, значения которых равны заданному числу, то программа найдёт последний из них.

Во многих случаях требуется найти первый из элементов, имеющих соответствующее значение, и дальнейший просмотр массива прекратить. Для этой цели можно использовать следующую программу:

```

i:=0;
repeat
  i:=i+1;
until (a[i]=x) or (i=10);
if a[i]=x then write(i) else write('Нет')

```

Здесь выполнение алгоритма будет прервано в одном из двух случаев: 1) в массиве найден первый из элементов, равный заданному; 2) все элементы массива просмотрены.

Запишите полный текст программы и выполните её на компьютере. 

Зачастую требуется определить количество элементов, удовлетворяющих некоторому условию. В этом случае вводится переменная, значение которой увеличивается на единицу каждый раз, когда найден нужный элемент.


Определите, количество каких элементов подсчитывается в следующем фрагменте программы. 

```
k:=0;
for i:=1 to 10 do
  if a[i]>50 then k:=k+1;
write('k=', k)
```

Если требуется определить сумму значений элементов, то вводят переменную, к значению которой прибавляют значение найденного элемента массива.

Определите, какому условию удовлетворяют элементы массива, значения которых суммируются в следующем фрагменте программы. 

```
s:=0;
for i:=1 to 10 do
  if (a[i]>50) and (a[i]<60) then s:=s+a[i];
write('s=', s)
```

Запишите полные тексты двух последних программ и выполните их на компьютере. 

2.2.6. Сортировка массива

Под сортировкой (упорядочением) массива понимают перераспределение значений его элементов в некотором определённом порядке.

Порядок, при котором в массиве первый элемент имеет самое маленькое значение, а значение каждого следующего элемента не меньше значения предыдущего элемента, называют **неубывающим**.

Порядок, при котором в массиве первый элемент имеет самое большое значение, а значение каждого следующего элемента не больше значения предыдущего элемента, называют **невозрастающим**.

Цель сортировки — облегчить последующий поиск элементов: искать нужный элемент в упорядоченном массиве легче.

Вы уже встречались с сортировкой при работе с базами данных. Сейчас мы рассмотрим один из возможных вариантов¹ реализации механизма этой операции — сортировку выбором.

Сортировка выбором (например, по невозрастанию) осуществляется следующим образом:

- 1) в массиве выбирается максимальный элемент;
- 2) максимальный и первый элементы меняются местами (первый элемент считается отсортированным);
- 3) в неотсортированной части массива снова выбирается максимальный элемент; он меняется местами с первым неотсортированным элементом массива;
- 4) действия, описанные в п. 3, повторяются с неотсортированными элементами массива до тех пор, пока не останется один неотсортированный элемент (его значение будет минимальным).

Рассмотрим процесс сортировки выбором на примере массива $a = \{0, 1, 9, 2, 4, 3, 6, 5\}$.

Индекс	1	2	3	4	5	6	7	8	
Значение	0	1	9	2	4	3	6	5	
Шаги	1	0	1	9	2	4	3	6	5
	2	9	1	0	2	4	3	6	5
	3	9	6	0	2	4	3	1	5
	4	9	6	5	2	4	3	1	0
	5	9	6	5	4	2	3	1	0
	6	9	6	5	4	3	2	1	0
	7	9	6	5	4	3	2	1	0
	Итог:	9	6	5	4	3	2	1	0



В этом массиве из восьми элементов операцию выбора максимального элемента мы проводили 7 раз. В массиве из n элементов такая операция будет проводиться $n-1$ раз. Объясните почему.

¹ С другими способами сортировки вы познакомитесь на уроках информатики в 10–11 классах.

Приведём фрагмент программы, реализующий описанный алгоритм:

```

for i:=1 to n-1 do
begin
    imax:=i;
    for j:=i+1 to n do if a[j]>a[imax] then imax:=j;
    x:=a[i];
    a[i]:=a[imax];
    a[imax]:=x
end;
    
```

Здесь мы использовали один цикл внутри другого. Такая конструкция называется вложенным циклом.

Запишите полный текст программы и выполните её на компьютере для рассмотренного в примере массива *a*.

На сайте «Интерактивные демонстрации по программированию» (<http://informatika.kspu.ru/flashprog/demos.php>) вы сможете поработать с интерактивными наглядными пособиями для того, чтобы более полно представить процесс сортировки выбором и другими способами.

САМОЕ ГЛАВНОЕ

Массив — это поименованная совокупность однотипных элементов, упорядоченных по индексам, определяющим положение элементов в массиве. В языках программирования массивы используются для реализации таких структур данных, как последовательности и таблицы.

Перед использованием в программе массив должен быть описан. Общий вид описания одномерного массива:

```

var <имя_массива>: array [<мин_знач_индекса> ..
    <макс_знач_индекса>] of тип_элементов;
    
```

Заполнять массив можно либо вводя значение каждого элемента с клавиатуры, либо присваивая элементам некоторые значения в программе. При заполнении массива и его выводе на экран используется цикл с параметром.

При решении разнообразных задач, связанных с обработкой массивов, используются такие типовые алгоритмы, как: суммирование элементов массива; поиск элемента с заданными свойствами; сортировка массива.





Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Какими слайдами вы могли бы дополнить презентацию?
2. Может ли массив одновременно содержать целые и вещественные значения?
3. Для чего необходимо описание массива?
4. Что вы можете сказать о массиве, сформированном следующим образом?



- a) `for i:=1 to 10 do a[i]:=random(101)-50;`
- b) `for i:=1 to 20 do a[i]:=i;`
- в) `for i:=1 to 5 do a[i]:=2*i-1;`



5. Запишите на языке Паскаль программу решения задачи, рассмотренной в примере пункта 2.2.4. Считайте количество жильцов дома случайным числом из диапазона от 50 до 200 человек, а число домов $n = 30$.



6. Напишите программу, которая вычисляет среднюю за неделю температуру воздуха. Исходные данные вводятся с клавиатуры.

Пример входных данных	Пример выходных данных
Введите температуру Понедельник>>12 Вторник>>10 Среда>>16 Четверг>>18 Пятница>>17 Суббота>>16 Воскресенье>>14	Средняя температура за неделю: 14.71



7. Дан массив из десяти целых чисел. Определите, сколько элементов этого массива имеют максимальное значение.



8. В классе 20 учеников писали диктант по русскому языку. Напишите программу, подсчитывающую количество двоек, троек, четвёрок и пятёрок, полученных за диктант.



9. Объявлен набор в школьную баскетбольную команду. Известен рост каждого из n учеников, желающих попасть в эту команду. Составьте алгоритм подсчёта количества претендентов,

имеющих шанс попасть в команду, если рост игрока команды должен быть не менее 170 см. Запишите на языке Паскаль программу. Считайте рост претендента в команду случайным числом из диапазона от 150 до 200 см, а число претендентов $n = 50$.

10. В целочисленных массивах a и b содержатся длины катетов десяти прямоугольных треугольников ($a[i]$ — длина первого катета, $b[i]$ — длина второго катета i -го треугольника). Найдите треугольник с наибольшей площадью. Выведите его номер, длины катетов и площадь. Предусмотрите случай, когда таких треугольников несколько.
11. Занесите информацию о десяти европейских странах в массивы n (название страны), k (численность населения), s (площадь страны). Выведите названия стран в порядке возрастания плотности их населения.

§ 2.3

Конструирование алгоритмов

Ключевые слова:

- последовательное построение алгоритма
- вспомогательный алгоритм
- формальные параметры
- фактические параметры
- рекурсивный алгоритм

2.3.1. Последовательное построение алгоритма

Существуют различные методы конструирования (разработки, построения) алгоритмов. Мы познакомимся с одним из них — методом последовательного построения (уточнения) алгоритма. Иначе он называется методом разработки «сверху вниз», нисходящим методом или методом пошаговой детализации.

Процесс последовательного построения алгоритма выглядит следующим образом.

На первом шаге мы считаем, что перед нами совершенный исполнитель, который «всё знает и всё умеет». Поэтому достаточно определить исходные данные и результаты алгоритма, а сам алгоритм представить в виде единого предписания — постановки задачи (рис. 2.2).

Если исполнитель не обучен исполнять заданное предписание, то необходимо представить это предписание в виде совокупности более простых предписаний (команд). Для этого:

- задачу разбивают на несколько частей, каждая из которых проще всей задачи;
- решение каждой части задачи формулируют в отдельной команде, которая также может выходить за рамки системы команд исполнителя;



Рис. 2.2. Линейный алгоритм, являющийся результатом первого этапа детализации задачи

- при наличии в алгоритме предписаний, выходящих за пределы возможностей исполнителя, такие предписания вновь представляются в виде совокупности ещё более простых предписаний. Процесс продолжается до тех пор, пока все предписания не будут понятны исполнителю.

Объединяя полученные предписания в единую совокупность выполняемых в определённой последовательности команд, получаем требуемый алгоритм решения исходной задачи.

2.3.2. Разработка алгоритма методом последовательного уточнения для исполнителя Робот

Вы уже знакомы с исполнителем Робот. Он действует на клетчатом поле, между клетками которого могут быть стены.

Система команд исполнителя Робот:

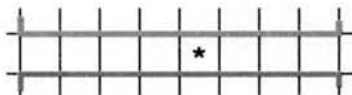
Команда	Описание команды
Вверх	Робот перемещается в соседнюю клетку в указанном направлении. Если в этом направлении между клетками стоит стена, то Робот разрушается
Вниз	
Вправо	
Влево	
Закрасить	Робот закрашивает ту клетку, в которой находится

Команда	Описание команды
Сверху свободно	Проверка истинности условия отсутствия стены у соответствующей стороны той клетки, где находится Робот: стены нет – «истина», иначе «ложь»
Снизу свободно	
Слева свободно	
Справа свободно	
Сверху стена	Проверка истинности условия наличия стены у соответствующей стороны той клетки, где находится Робот: стена есть – «истина», иначе «ложь»
Снизу стена	
Слева стена	
Справа стена	
Клетка закрашена	Проверка истинности условия: клетка закрашена – «истина», иначе «ложь»
если <условие> то <последовательность команд> все	Организация ветвления: если <условие> верно, то выполняется <последовательность команд>
идь пока < условие > <последовательность команд> кц	Организация цикла: пока <условие> верно, выполняется <последовательность команд>

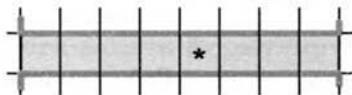
В одном условии можно использовать несколько команд, применяя логические операции И, ИЛИ, НЕ.



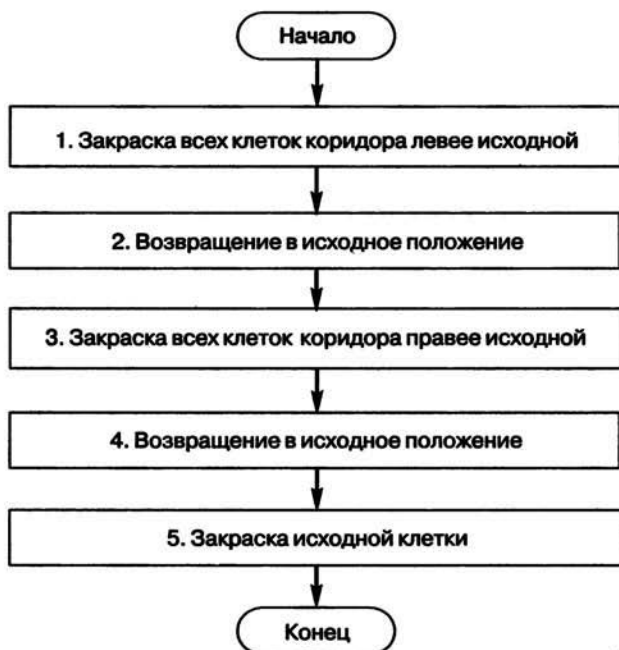
Известно, что Робот находится где-то в горизонтальном коридоре. Ни одна из клеток коридора не закрашена.



Составим алгоритм, под управлением которого Робот закрасит все клетки этого коридора и вернётся в исходное положение.



Представим план действий Робота следующими укрупнёнными шагами (модулями):



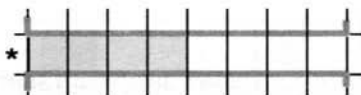
Детализируем каждый из пяти модулей.

1. Чтобы закрасить все клетки коридора, находящиеся левее Робота, прикажем Роботу шагнуть влево и выполнить цикл-ПОКА:

```

влево
нц пока сверху стена и снизу стена
    закрасить; влево
кц
    
```

Под управлением этого алгоритма Робот закрасит все клетки коридора, находящиеся левее от него, и окажется на клетке рядом с левой границей коридора.

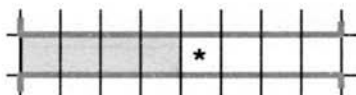


2. Командой вправо вернём Робота в коридор. Наша задача — вернуть Робота в исходную точку. Эта точка имеет единственный отличительный признак — она не закрашена. Поэтому пока занимаемая

Роботом клетка оказывается закрашенной, будем перемещать его вправо.

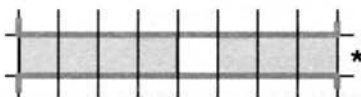
```
вправо
  нц пока клетка закрашена
    вправо
  кц
```

Под управлением этого алгоритма Робот окажется в исходной клетке.



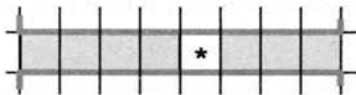
3. Выполнив команду вправо, Робот пройдёт исходную клетку и займёт клетку правее исходной. Теперь можно закрашивать клетки коридора, расположенные правее исходной.

```
вправо
  нц пока сверху стена и снизу стена
    закрасить; вправо
  кц
```



4. Так как, выполнив предыдущий алгоритм, Робот оказался правее коридора, командой влево вернём его в коридор. Возвращение в исходную точку обеспечивается алгоритмом:

```
влево
  нц пока клетка закрашена
    влево
  кц
```



5. По команде закрасить Робот закрашивает исходную клетку.

Полностью программа управления Роботом выглядит так:

```
алг
нач
  влево
```



```

нц пока сверху стена и снизу стена
    закрасить; влево
кц
    вправо
нц пока клетка закрашена
    вправо
кц
    вправо
нц пока сверху стена и снизу стена
    закрасить; вправо
кц
    влево
нц пока клетка закрашена
    влево
кц
    закрасить
кон
    
```



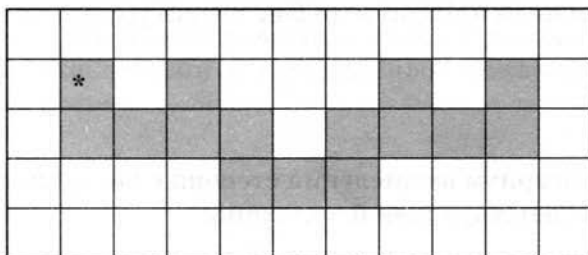
2.3.3. Вспомогательные алгоритмы

При построении новых алгоритмов нередко возникают ситуации, когда в разных местах алгоритма необходимо выполнение одной и той же последовательности шагов обработки данных. Для такой последовательности шагов создают отдельный алгоритм, называемый вспомогательным. В качестве вспомогательных могут использоваться алгоритмы, ранее разработанные для решения других задач.

Вспомогательный алгоритм — алгоритм, целиком используемый в составе другого алгоритма.



Пример 1. В среде КуМир составим алгоритм для исполнителя Робот, под управлением которого он нарисует узор:



Начальное положение Робота отмечено звёздочкой. В алгоритме использован вспомогательный алгоритм фигура.

использовать Робот

алг узор

нач

 фигура

 вправо; вниз

 фигура

 вправо; вниз

 фигура

кон

алг фигура

нач

 закрасить; вниз

 закрасить; вправо; закрасить; вправо; закрасить

 вверх; закрасить

кон

При представлении алгоритмов с помощью блок-схем для обозначения команды вызова вспомогательного алгоритма используется блок «предопределённый процесс» (рис. 2.3), внутри которого записывается название (имя) вспомогательного алгоритма, после которого в скобках перечисляются параметры — входные данные и результаты.

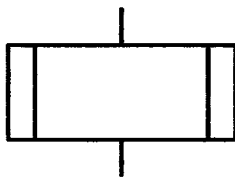


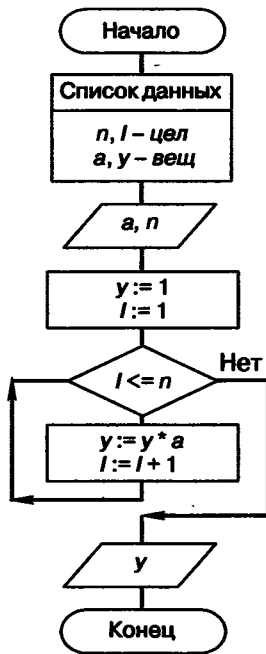
Рис. 2.3. Блок «предопределённый процесс»

Вспомогательный алгоритм делает структуру алгоритма более понятной.



Пример 2.

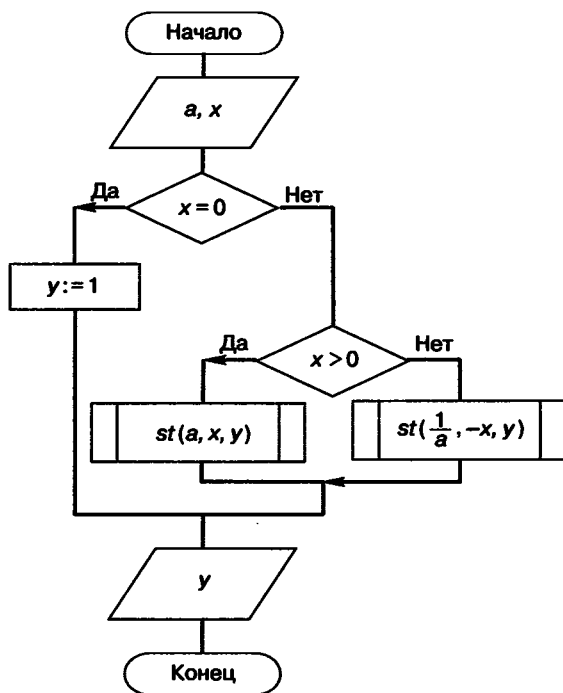
Вспомним алгоритм вычисления степени с натуральным показателем $y = a^n$. Соответствующая блок-схема:



Степень с целым показателем $y = a^x$, где x – целое число, $a \neq 0$ вычисляется так:

$$y = \begin{cases} 1, & \text{при } x = 0, \\ a^x, & \text{при } x > 0, \\ \left(\frac{1}{a}\right)^{-x}, & \text{при } x < 0. \end{cases}$$

В приведённой записи дважды фигурирует вычисление степени с натуральным показателем. Поэтому в алгоритм вычисления степени с целым показателем можно включить вызов вспомогательного алгоритма вычисления степени с натуральным показателем. Соответствующая блок-схема:



Алгоритм, представленный на блок-схеме, является основным по отношению к вызываемому в нём вспомогательному алгоритму.

Параметрами используемого вспомогательного алгоритма являются величины a , n , y . Это формальные параметры, они используются при описании алгоритма. При конкретном обращении к вспомогательному алгоритму формальные параметры заменяются фактическими параметрами, т. е. именно теми величинами, для которых будет исполнен вспомогательный алгоритм. Типы, количество и порядок следования формальных и фактических параметров должны совпадать.

Команда вызова вспомогательного алгоритма выполняется следующим образом (рис. 2.4):

- 1) формальные входные данные вспомогательного алгоритма заменяются значениями фактических входных данных, указанных в команде вызова вспомогательного алгоритма;
- 2) для заданных входных данных исполняются команды вспомогательного алгоритма;
- 3) полученные результаты присваиваются переменным с именами фактических результатов;
- 4) осуществляется переход к следующей команде основного алгоритма.

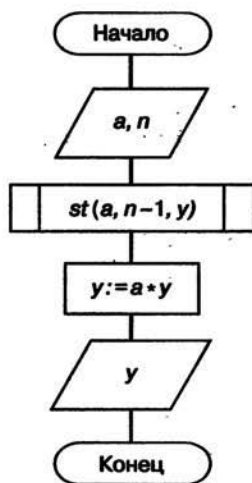


Рис. 2.4. Схема выполнения команды вызова вспомогательного алгоритма

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют **рекурсивным**.

Рассмотрим несколько примеров рекурсивных алгоритмов.

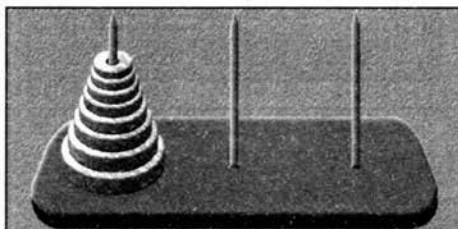
Пример 3. Алгоритм вычисления степени с натуральным показателем n для любого вещественного числа a можно представить в виде рекурсивного:



n -я степень числа a есть не что иное, как произведение $a^{n-1} \cdot a$; в свою очередь, $a^{n-1} = a^{n-2} \cdot a$ и т. д.



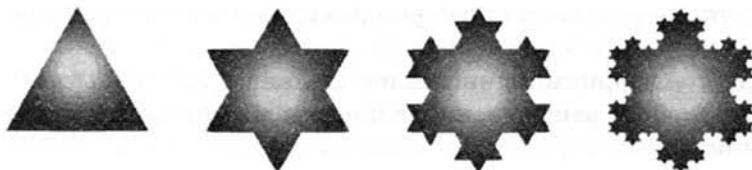
Пример 4. Рекурсивный алгоритм положен в основу эффективно-го решения головоломки «Ханойская башня».



Интерактивная игра «Ханойские башни» (195747) поможет вам вспомнить условие и алгоритм решения головоломки (<http://sc.edu.ru/>).



Пример 5. Рассмотрим алгоритм построения геометрической фигуры, которая называется снежинкой Коха. Шаг процедуры построения состоит в замене средней трети каждого из имеющихся отрезков двумя новыми такой же длины, как показано на рисунке:



Начальное состояние Первый шаг Второй шаг Третий шаг

С каждым шагом фигура становится всё причудливее. Граница снежинки Коха — положение кривой после выполнения бесконечно-го числа шагов.



Попробуйте подсчитать, сколько рёбер в границе снежинки Коха после четвёртого шага; после пятого шага.

САМОЕ ГЛАВНОЕ

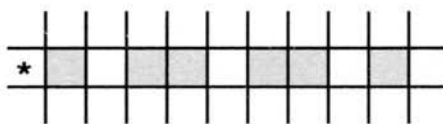
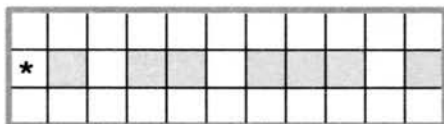
Один из основных методов конструирования алгоритмов — метод последовательного построения алгоритма. Его суть состоит в том, что: исходная задача разбивается на несколько частей, каждая из которых проще всей задачи, и решение каждой части формулируется в отдельной команде; если получаются команды, выходящие за пределы возможностей исполнителя, то они представляются в виде совокупности ещё более простых предписаний. Процесс продолжается до тех пор, пока все предписания не будут понятны исполнителю.

Вспомогательный алгоритм — алгоритм, целиком используемый в составе другого алгоритма.

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют рекурсивным.

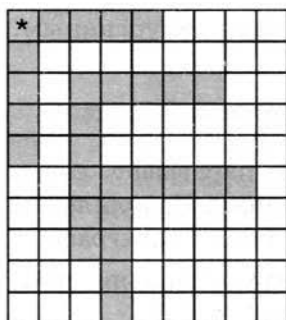
Вопросы и задания

1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Дополняет ли презентация информацию, содержащуюся в тексте параграфа?
2. Почему при решении сложной задачи затруднительно сразу конкретизировать все необходимые действия?
3. В чём заключается метод последовательного уточнения при построении алгоритма?
4. Какая связь между методом последовательного построения алгоритма и такими процессами, как написание сочинения или подготовка к многодневному туристическому походу?
5. Известен рост каждого из n учеников 9А класса и m учеников 9Б класса. Опишите укрупнёнными блоками алгоритм сравнения среднего роста учеников этих классов.
6. В ряду из десяти клеток правее Робота некоторые клетки закрашены. Последняя закрашенная клетка может примыкать к стене. Составьте алгоритм, который закрашивает клетки выше и ниже каждой закрашенной клетки. Проверьте работу алгоритма в следующих случаях:

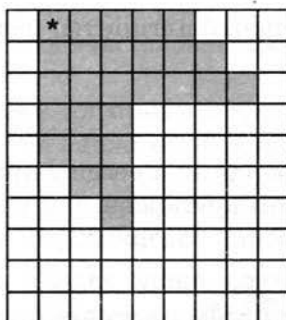


7. Для чего нужны вспомогательные алгоритмы?
8. Опишите процесс выполнения команды вызова вспомогательного алгоритма в основном алгоритме.
9. Сталкивались ли вы с идеей формальных и фактических параметров при изучении математики и физики? Приведите пример.
10. Какие алгоритмы называют рекурсивными? Приведите пример рекурсии из жизни.
11. Составьте алгоритмы, под управлением которых Робот закрасит указанные клетки. При необходимости используйте вспомогательный алгоритм.

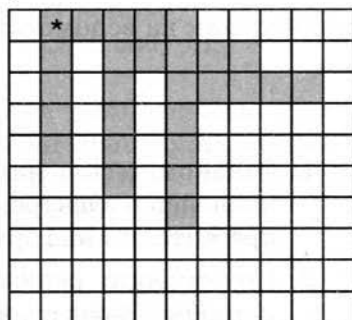
а)



б)



в)



§ 2.4

Запись вспомогательных алгоритмов на языке Паскаль

Ключевые слова:

- подпрограмма
- процедура
- функция
- рекурсивная функция

Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью подпрограмм. В Паскале подпрограмма является частью основной программы. Её описание располагается между разделом `var` и программным блоком главной программы. Если подпрограмм несколько, то их описания располагаются в произвольном порядке одно за другим.

Структура описания подпрограммы аналогична структуре главной программы. Описание подпрограммы начинается с заголовка и заканчивается оператором `end`.

В Паскале различают два вида подпрограмм: процедуры и функции.

2.4.1. Процедуры

Процедура — подпрограмма, имеющая произвольное количество входных и выходных данных.

Описание процедуры имеет вид:

```
procedure <имя_процедуры> (<описание параметров-значений>;  
    var: <описание параметров-переменных>);  
begin  
    <операторы>  
end;
```

В заголовке процедуры после её имени приводится перечень формальных параметров и их типов. Входные параметры, значения которых не изменяются в программе, должны быть параметрами-значениями. Выходные (результатирующие) параметры должны быть параметрами-переменными.

Для вызова процедуры достаточно указать её имя со списком фактических параметров. В качестве параметров-значений можно указывать имена переменных, константы и выражения.

Например, заголовок процедуры вычисления наибольшего общего делителя может быть описан так:

```
procedure nod (a, b: integer; var c: integer);
```

Возможны следующие варианты вызова этой процедуры:

```
nod (36, 15, z)      – в качестве параметров-значений использованы константы;
nod (x, y, z)       – в качестве параметров-значений использованы имена переменных;
nod (x+y, 15, z)    – в качестве параметров-значений использованы выражение и константа;
```

В любом случае между фактическими и формальными параметрами должно быть полное соответствие по количеству, порядку следования и типу.

Пример 1. Напишем процедуру для нахождения наибольшего общего делителя двух чисел с помощью алгоритма Евклида. Используем её для нахождения наибольшего общего делителя следующих шести чисел: 16, 32, 40, 64, 80 и 128.

<code>program n_6;</code>	Заголовок главной программы
<code>const m: array [1..6] of integer = (16, 32, 40, 64, 80, 128);</code>	Раздел описания констант
<code>var i, x, y, z: integer;</code>	Раздел описания переменных
<code>procedure nod (a, b: integer; var c: integer); begin while a<>b do if a>b then a:=a-b else b:=b-a; c:=a end;</code>	Раздел описания подпрограмм

```
begin
  x:=m[1];
  for i:=2 to 6 do
    begin
      y:=m[i];
      nod (x, y, z);
      x:=z
    end;
  writeln ('НОД=', x)
end.
```

Раздел операторов
главной программы

Измените программу так, чтобы с её помощью можно было найти:

- наибольший общий делитель следующих пяти чисел: 12, 24, 30, 48 и 51;
- наибольший общий делитель произвольных десяти целых двузначных чисел.



2.4.2. Функции

Описание функции имеет вид:

```
function <имя_функции> (<описание входных данных>):
  <тип_функции>;
begin
  <операторы>;
  <имя_функции> := <результат>
end;
```

В заголовке функции после её имени приводится описание входных данных — указывается перечень формальных параметров и их типов. Там же указывается тип самой функции, т. е. тип результата.

Функция — подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем функции. Поэтому в блоке функции обязательно должен присутствовать оператор `<имя_функции>:=<результат>`.

Для вызова функции достаточно указать её имя со списком фактических параметров в любом выражении, в условиях (после слов `if`, `while`, `until`) или в операторе `write` главной программы.

Пример 2. Напишем программу нахождения максимального из четырёх целых чисел, использующую функцию поиска максимального из двух чисел:



<code>program n_7;</code>	Заголовок главной программы
<code>var a, b, c, d, f: integer;</code>	Раздел описания переменных
<code>function max (x, y: integer): integer;</code> <code>begin</code> <code>if x>y then max:=x else max:=y;</code> <code>end;</code>	Раздел описания подпрограмм
<code>begin</code> <code>readln (a, b, c, d);</code> <code>f:=max(max(a, b), max(c, d));</code> <code>writeln ('f=', f);</code> <code>end.</code>	Раздел операторов главной программы



Измените программу так, чтобы с её помощью можно было найти:

- а) максимальное из чисел a, b, c ;
- б) максимальное из чисел b, c, d ;
- в) минимальное из четырёх чисел;
- г) разность максимального и минимального из четырёх чисел.



Пример 3. В январе Саши подарили пару новорождённых кроликов. Через два месяца они дали первый приплод — новую пару кроликов, а затем давали приплод по паре кроликов каждый месяц. Каждая новая пара также даёт первый приплод (пару кроликов) через два месяца, а затем — по паре кроликов каждый месяц. Сколько пар кроликов будет у Саши в декабре?

Составим математическую модель этой задачи. Обозначим через $f(n)$ количество пар кроликов в месяце с номером n . По условию задачи, $f(1) = 1, f(2) = 1, f(3) = 2$. Из двух пар, имеющих в марте, дать приплод в апреле сможет только одна: $f(4) = 3$. Из пар, имеющих в апреле, дать приплод в мае смогут только пары, родившиеся в марте и ранее: $f(5) = f(4) + f(3) = 3 + 2 = 5$. В общем случае: $f(n) = f(n - 1) + f(n - 2), n \geq 3$.



Числа 1, 1, 2, 3, 5, 8, ... образуют так называемую последовательность **Фибоначчи**, названную в честь итальянского математика, впервые решившего соответствующую задачу ещё в начале XIII века.

Оформим в виде функции вычисление члена последовательности Фибоначчи.

```
function f (n: integer): integer;
begin
  if (n=1) or (n=2) then f:=1
  else f:=f(n-1)+f(n-2)
end;
```

Полученная функция считается рекурсивной — в ней реализован способ вычисления очередного значения функции через вычисление её предшествующих значений.

Напишите программу, вычисляющую и выводящую 10 первых членов последовательности Фибоначи.



САМОЕ ГЛАВНОЕ

Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью подпрограмм. В Паскале различают два вида подпрограмм: процедуры и функции.

Процедура — подпрограмма, имеющая произвольное количество входных и выходных данных.

Функция — подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем функции.

Вопросы и задания

1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Дополняет ли презентация информацию, содержащуюся в тексте параграфа?
2. Для чего используются подпрограммы?
3. В чём основное различие процедур и функций?
4. Напишите программу вычисления наименьшего общего кратного следующих четырёх чисел: 36, 54, 18 и 15. Используйте процедуру вычисления наибольшего общего делителя двух чисел.



5. Напишите программу перестановки значений переменных a , b , c в порядке возрастания, т. е. так, чтобы $a < b < c$. Используйте процедуру swap.

```

procedure swap (var x, y: integer);
    var m: integer;
begin
    m:=x;
    x:=y;
    y:=m
end;
    
```

Исходные данные вводятся с клавиатуры.

Пример входных данных	Пример выходных данных
1 2 3	1 2 3
2 1 3	1 2 3
3 1 2	1 2 3
2 3 1	1 2 3

6. Видоизмените программу сортировки массива выбором так, чтобы в ней использовалась процедура выбора наибольшего элемента массива.
7. Напишите программу вычисления выражения:
 $s = 1! + 2! + 3! + \dots + n!$
 Здесь $n!$ — факториал числа n . $n! = 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n$.
 Используйте функцию вычисления факториала.
8. Напишите программу вычисления выражения:
 $s = x^3 + x^5 + x^n$, где x и n вводятся с клавиатуры. Используйте подпрограмму вычисления степени.
9. Напишите функцию, вычисляющую длину отрезка по координатам его концов. С помощью этой функции напишите программу, вычисляющую периметр треугольника по координатам его вершин.
10. Напишите функцию, вычисляющую площадь треугольника по целочисленным координатам его вершин. С помощью этой функции вычислите площадь четырёхугольника по координатам его вершин.

Алгоритмы управления

Ключевые слова:

- управление
- алгоритм управления
- обратная связь

2.5.1. Управление

Управление — это процесс целенаправленного воздействия на объект; осуществляется для организации функционирования объекта по заданной программе.



В середине прошлого века выдающийся американский учёный Норберт Винер (1894—1964), изучавший различные технические и биологические системы, установил, что управление в них осуществляется по общей схеме. Винер считается основоположником науки об управлении — **кибернетики**.



Управляемым объектом (объектом управления) может быть техническое устройство (например, автомобиль), один человек (например, ученик, солдат) или коллектив (например, оркестр, работники предприятия).

Управляющим объектом (управляющей системой) может быть человек (например, шофёр, дирижёр оркестра, учитель, директор), коллектив (например, правительство, парламент), а может быть и техническое устройство (например, автоматический регулятор, компьютер).



Последовательность команд по управлению объектом, приводящая к заранее поставленной цели, называется **алгоритмом управления**.

Простейшие алгоритмы управления могут состоять из одной команды или представлять собой линейную последовательность команд. Более сложные алгоритмы управления содержат ветвления и циклы.

2.5.2. Обратная связь

Для управления нужна информация. Во-первых, управляющий объект должен получить информацию о том, что ему нужно, т. е. он должен знать цель своих действий. Во-вторых, управляющий объект должен знать, как можно достичь поставленной цели. Важно, что информация о цели и способах её достижения должна быть известна управляющему объекту до начала процесса управления.



Пример 1. Рассмотрим управление движением автомашин (объект управления) на перекрёстке с помощью светофора (управляющий объект). В этой ситуации управляющее воздействие формируется в зависимости от заложенной в управляющем объекте исходной информации. Светофор не воспринимает текущую информацию о состоянии движения на перекрёстке, он не изменяет алгоритм управления от того, что с какой-то стороны скопилось очень много машин и образовалась «пробка».



Обратная связь — это процесс передачи информации о состоянии объекта управления в управляющую систему.

Обратная связь позволяет корректировать управляющие воздействия управляющей системы на объект управления в зависимости от состояния объекта управления (рис. 2.5). Обратная связь предусмотрена в ряде бытовых приборов (например, утюг с терморегулятором, холодильник, кастрюля-скороварка), в живых организмах, в обществе.

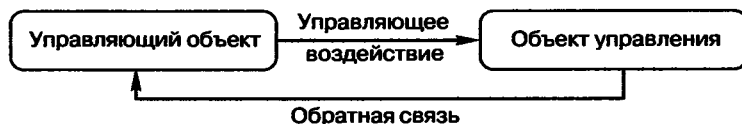


Рис. 2.5. Кибернетическая модель управления

В настоящее время очень часто роль управляющей системы отводится компьютеру, в память которого заложена программа управления, предусматривающая все варианты информации, которые могут быть получены по обратной связи.

Пример 2. Если вместо обычного светофора на дорожном перекрёстке будет установлен «интеллектуальный» светофор — высокотехнологичное устройство, оснащённое датчиками, фиксирующими скорость движения на дороге и плотность транспортных потоков, то управление движением станет более рациональным за счёт учёта информации, поступающей от объекта управления.



САМОЕ ГЛАВНОЕ

Управление — процесс целенаправленного воздействия на объект; осуществляется для организации функционирования объекта по заданной программе.

Последовательность команд по управлению объектом, приводящая к заранее поставленной цели, называется алгоритмом управления.

Вопросы и задания

1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Какими слайдами вы могли бы дополнить презентацию?
2. Что такое управление? Приведите примеры управляющих систем и управляемых ими объектов.
3. Что такое алгоритм управления? Приведите примеры ситуаций, в которых имеют место линейные, разветвляющиеся и циклические алгоритмы управления.
4. Что изучает наука кибернетика?
5. Какая информация нужна для управления? Приведите пример.
6. Что такое обратная связь?
7. Опишите кибернетическую модель управления.



Тестовые задания для самоконтроля

1. Что является результатом этапа «формализация» решения задачи на компьютере?
 - а) словесная информационная модель
 - б) математическая модель
 - в) алгоритм
 - г) программа
2. Имеется описание:

```
var c: array [1..20] of integer;
```

Для хранения массива *c* будет отведено... ячеек памяти объёмом... байтов.
 - а) 40, 20
 - б) 20, 320
 - в) 20, 40
 - г) 20, 20
3. Чему равна сумма элементов *a[1]* и *a[4]* массива, сформированного следующим образом?

```
for i:=1 to 5 do a[i]:=i*(i+1);
```

 - а) 30
 - б) 5
 - в) 22
 - г) 40
4. Массив описан следующим образом:

```
const b: array [1..5] of integer = (1, 2, 3, 5, 11);
```

Значение выражения $b[5]*b[4]-b[2]-b[3]*b[1]$ равно:
 - а) 50
 - б) 15
 - в) 11
 - г) 22

5. Для записи вспомогательных алгоритмов в языке Паскаль используются:
- а) массивы
 - б) составные операторы
 - в) процедуры и функции
 - г) операторы и операнды
6. Между формальными и фактическими параметрами следует соблюдать соответствие:
- а) по типу параметров
 - б) по количеству параметров
 - в) по порядку следования параметров
 - г) по всему, перечисленному в п. а)–в)
7. Алгоритм, целиком используемый в составе другого алгоритма, называется:
- а) рекурсивным
 - б) вспомогательным
 - в) основным
 - г) дополнительным
8. Подпрограмма, имеющая произвольное количество входных и выходных данных, называется:
- а) процедурой
 - б) функцией
 - в) вспомогательным алгоритмом
9. Что такое управление? Выберите самое полное определение.
- а) перевод объекта из одного состояния в другое
 - б) удержание объекта в существующем состоянии
 - в) процесс целенаправленного воздействия одних объектов на другие объекты
 - г) регулирование движения автомашин на перекрёстке
10. Кто является основоположником кибернетики?
- а) Норберт Винер
 - б) Джон фон Нейман
 - в) Платон
 - г) И. П. Павлов

Для проверки знаний и умений по теме «Алгоритмизация и программирование» вы можете воспользоваться интерактивным тестом к главе 2, содержащимся в электронном приложении к учебнику.



Глава 3

ОБРАБОТКА ЧИСЛОВОЙ ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

§ 3.1

Электронные таблицы

Ключевые слова:

- электронные таблицы
- табличный процессор
- столбец
- строка
- ячейка
- диапазон ячеек
- лист
- книга

Сотни лет в деловой сфере при выполнении громоздких однотипных расчётов используются таблицы. С их помощью рассчитывается заработная плата, ведутся различные системы учёта материальных ценностей, просчитывается стоимость новых товаров и услуг, прогнозируется размер прибыли и т. д. Такие расчёты многие специалисты до конца прошлого века выполняли с помощью калькуляторов, вручную занося полученные результаты в соответствующие графы таблиц. Такая работа требовала больших временных затрат; на исправление незначительной ошибки, допущенной расчётчиком, уходили недели и даже месяцы.

Ситуация кардинально изменилась с появлением электронных таблиц, позволивших за счёт изменения исходных данных быстро решать большое количество типовых расчётных задач.



Электронные таблицы (табличный процессор) — это прикладная программа, предназначенная для организации табличных вычислений на компьютере. Электронными также называют и таблицы, созданные с помощью одноименных программ в памяти компьютера.

В наши дни электронные таблицы являются одним из программных продуктов, наиболее широко используемых на практике. С их помощью пользователи, не обладая специальными знаниями в области программирования, имеют возможность определять последовательность вычислительных операций, выполнять различные преобразования исходных данных, представлять полученные результаты в графической форме.

3.1.1. Интерфейс электронных таблиц

Наиболее распространёнными табличными процессорами являются Microsoft Excel и OpenOffice.org Calc. При запуске любого из них на экран выводится окно, многие элементы которого вам хорошо известны по опыту работы с другими программами (рис. 3.1).

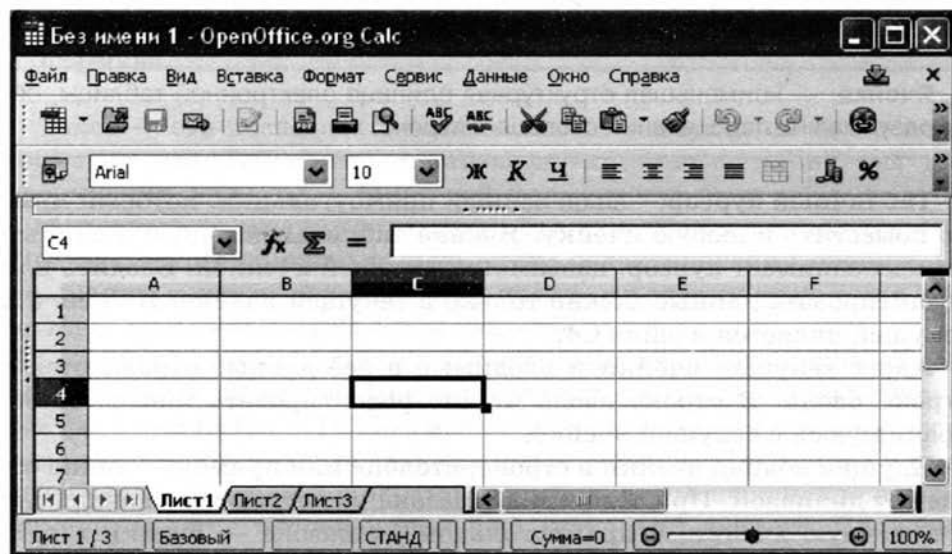


Рис. 3.1. Интерфейс табличного процессора OpenOffice.org Calc

Строка заголовка содержит название документа, название программы и кнопки управления окном.

Строка меню содержит названия групп команд управления электронной таблицей, объединённых по функциональному признаку.

Панели инструментов содержат пиктограммы для вызова наиболее часто выполняемых команд.

Рабочей областью табличного процессора является прямоугольное пространство, разделённое на столбцы и строки. Каждый столбец и каждая строка имеют обозначения (заголовки, имена). Столбцы обозначаются слева направо латинскими буквами в алфавитном порядке; могут использоваться однобуквенные, двухбуквенные и трёхбуквенные имена (А, В, С и т. д.; после 26-го столбца начинаются двухбуквенные сочетания АА, АВ и т. д.). Строки нумеруются сверху вниз. Максимально возможное число строк и столбцов у разных табличных процессоров различно.

На пересечении столбцов и строк образуются ячейки (клетки), в которые могут быть записаны данные или выполняемые над ними операции. Ячейка — наименьшая структурная единица электронной таблицы. Каждая ячейка электронной таблицы имеет имя, составленное из буквенного имени столбца и номера строки, на пересечении которых она располагается. Возможны следующие имена ячеек: Е1, К12, АВ125¹. Таким образом, имя ячейки определяет её адрес в таблице.



Ячейка — наименьшая структурная единица электронной таблицы, образуемая на пересечении столбца и строки.

Табличный курсор — выделенный прямоугольник, который можно поместить в любую ячейку. Ячейка таблицы, которую в данный момент занимает курсор, называется текущей ячейкой. Вводить или редактировать данные можно только в текущей ячейке. На рис. 3.1 текущей является ячейка С4.

Адрес текущей ячейки и вводимые в неё данные отражаются в строке ввода. В строке ввода можно редактировать информацию, хранящуюся в текущей ячейке.

Идущие подряд ячейки в строке, столбце или прямоугольнике образуют диапазон. При задании диапазона указывают его начальную и конечную ячейки, в прямоугольном диапазоне — ячейки левого верхнего и правого нижнего углов. Наибольший диапазон представ-

¹ В современных версиях в Microsoft Excel положение ячейки может обозначаться буквой R, за которой следует номер строки, и буквой C, за которой следует номер столбца, например R1C1.

ляет вся таблица, наименьший — одна ячейка. Примеры диапазонов: A1:A10, B2:C2, B2:D10.

Рабочая область табличного процессора иначе называется листом. Создаваемый и сохраняемый в табличном процессоре документ называется книгой; он может состоять из нескольких листов. Аналогично листам бухгалтерской книги, их можно перелистывать, щёлкая на ярлыках, расположенных внизу окна. Каждому листу книги пользователь может дать имя, исходя из содержимого этого листа.

Лист — рабочая область, состоящая из ячеек.

Книга — документ электронной таблицы, состоящий из листов, объединённых одним именем, и являющийся файлом.



В строке состояния выводятся сообщения о текущем режиме работы таблицы и возможных действиях пользователя.

3.1.2. Данные в ячейках таблицы

Содержимым ячейки может быть:

- текст;
- число;
- формула.

Текст — это последовательность любых символов из компьютерного алфавита. Тексты (надписи, заголовки, пояснения) нужны для оформления таблицы, в текстовой форме могут быть представлены характеристики рассматриваемых объектов. Изменить содержимое ячейки с текстом можно только путём редактирования ячейки. По умолчанию текст выравнивается в ячейке по левому краю — по аналогии со способом письма слева направо.

С помощью чисел задаются количественные характеристики рассматриваемых объектов. При этом используются различные числовые форматы (табл. 3.1). По умолчанию используется числовой формат с двумя десятичными знаками после запятой. Для записи чисел, содержащих большое количество разрядов, не уместяющихся в ячейке, применяется экспоненциальный (научный) формат. Числовые данные, введённые в ячейки таблицы, являются исходными данными для проведения вычислений. Изменить числовые данные можно путём их редактирования. По умолчанию числа выравниваются в ячейке по правому краю, что обеспечивает выравнивание всех чисел столбца по разрядам (единицы размещаются под единицами, десятки — под десятками и т. д.).

Таблица 3.1

Некоторые числовые форматы

Числовой формат	Пример
Числовой	1234,01
Процентный	57%
Экспоненциальный (научный)	1,234E+03
Дробный	1/8
Денежный	1234 р.
Дата	23.12.2010
Время	08:30:00

Целая и дробная части вещественного числа разделяются в электронных таблицах запятой. При употреблении в записи числа точки (в качестве разделителя его целой и дробной частей) число интерпретируется как дата. Например, 9.05 воспринимается как 9 мая, а 5.25 — как май 2025 года.

Формула — это выражение (арифметическое, логическое), задающее некоторую последовательность действий по преобразованию данных. Формула всегда начинается со знака равенства (=) и может включать в себя ссылки (имена ячеек), знаки операций (табл. 3.2), функции и числа.

Таблица 3.2

Арифметические операции, применяемые в формулах

Арифметическая операция	Знак операции
Сложение	+
Вычитание	-
Умножение	*
Деление	/
Возведение в степень	^

При записи формул действуют правила, аналогичные тем, что приняты в языках программирования. Примеры формул:

$$=0,5*(A1+B1)$$

$$=C3^2$$

Для ввода в формулу имени ячейки достаточно поместить табличный курсор в соответствующую ячейку.

В процессе ввода формулы она отображается как в самой ячейке, так и в строке ввода. После завершения ввода (нажатие клавиши Enter) в ячейке отображается результат вычислений по этой формуле (рис. 3.2). Для просмотра и редактирования конкретной формулы достаточно выделить соответствующую ячейку и провести её редактирование в строке ввода.

	A	B	C	D	E
1	5	4	1		
2	2	10	6	14	
3					

Рис. 3.2. Вычисления по формуле

При изменении исходных данных в ячейках, имена которых входят в формулу, значение выражения автоматически пересчитывается, полученный результат отображается в ячейке с этой формулой.

3.1.3. Основные режимы работы электронных таблиц

Можно выделить следующие режимы работы электронных таблиц:

- режимы формирования таблицы;
- режимы отображения таблицы;
- режимы выполнения вычислений.

Режимы формирования электронной таблицы. При работе с табличными процессорами создаются документы, которые можно просматривать, изменять, записывать на носители внешней памяти для хранения, распечатывать на принтере.

Формирование электронных таблиц предполагает заполнение и редактирование документа. При этом используются команды, изменяющие содержимое ячеек (очистить, редактировать, копировать), и команды, изменяющие структуру таблицы (удалить, вставить, переместить).

Содержимое ячеек может быть оформлено с помощью стандартных средств оформления текстов: изменения рисунка шрифта, его размеров, начертания и выравнивания относительно ячейки, направления написания. Помимо этого, пользователю доступны сред-

ства оформления самой таблицы: объединение ячеек, различные способы прорисовки границ между ячейками для печати.

Данные, формат данных и параметры оформления ячеек (шрифт, цвет заливки, тип границы и пр.) можно копировать из одних ячеек (диапазонов ячеек) в другие ячейки (диапазоны ячеек) электронной таблицы.

Режимы отображения таблицы. Для электронной таблицы может быть установлен режим отображения формул или режим отображения значений. По умолчанию включён режим отображения значений, и на экране отображаются значения, вычисленные на основании содержимого ячеек. Можно специально задать режим отображения формул, при котором в ячейках вместо результатов вычислений будут отображены сами формулы (рис. 3.3).

	A	B	C
1	3	1	=A2-B2
2	=2+A1	=(A2+B1)/2	=C1*3

Рис. 3.3. Фрагмент таблицы в режиме отображения формул

Чтобы в OpenOffice.org Calc установить режим отображения формул, следует:

- 1) выполнить команду **Сервис—Параметры—OpenOffice.org Calc—Вид**;
- 2) в области *Показать* установить флажок **Формулы** и нажать кнопку **ОК**.



Самостоятельно выясните, как устанавливается режим отображения формул в табличном процессоре, имеющемся в вашем распоряжении.

Режимы выполнения вычислений. Все вычисления начинаются с ячейки, расположенной на пересечении первой строки и первого столбца электронной таблицы. Вычисления проводятся в естественном порядке; если в очередной ячейке находится формула, включающая адрес ещё не вычисленной ячейки, то вычисления по этой формуле откладываются до тех пор, пока значение в ячейке, от которого зависит формула, не будет определено.

При каждом вводе нового значения в ячейку документ пересчитывается заново — выполняется автоматический пересчёт тех формул, в которые входят новые данные. В большинстве табличных процессоров существует возможность установки ручного пересчёта: таблица пересчитывается заново только при подаче специальной команды.

В OpenOffice.org Calc выбор режима вычислений осуществляется с помощью команды Сервис—Содержимое ячеек—Пересчитать—Вычислить автоматически.

Самостоятельно выясните, как устанавливается режим вычислений в табличном процессоре, имеющемся в вашем распоряжении.



САМОЕ ГЛАВНОЕ

Электронные таблицы (табличный процессор) — прикладная программа, предназначенная для организации табличных вычислений на компьютере.

Ячейка — наименьшая структурная единица электронной таблицы, образуемая на пересечении столбца и строки. Содержимым ячейки может быть текст, число, формула.

Тексты (надписи, заголовки, пояснения) нужны для оформления таблицы. Числовые данные, введённые в ячейки таблицы, являются исходными данными для проведения вычислений. В ячейках с формулами отображаются результаты вычислений.

Формирование электронных таблиц предполагает заполнение, редактирование и форматирование документа.

При вводе в ячейку нового значения пересчёт документа осуществляется автоматически, но может быть установлен и режим ручного пересчёта.

Для электронной таблицы может быть установлен режим отображения формул или режим отображения значений.



Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Используйте эти материалы при подготовке ответов на вопросы и выполнении заданий.
2. Кем и когда были созданы первые электронные таблицы? (Для ответа на вопрос используйте дополнительные источники информации.)
3. Как известно, первые компьютеры специально создавались для выполнения всевозможных вычислений. Начало же широкого использования компьютеров в повседневной жизни не было связано с расчётами, которые представители многих профес-



сий до конца прошлого века выполняли вручную. Как вы можете объяснить это противоречие?

4. Что понимают под электронными таблицами?
5. Назовите основные элементы окна табличного процессора. Перечислите общие элементы окна табличного процессора и окна текстового процессора.
6. Какой табличный процессор установлен на вашем компьютере? Сколько строк и столбцов могут иметь создаваемые в нём электронные таблицы?
7. Как именуются ячейки таблицы? Почему имя ячейки иначе называют её координатами?
8. Какие данные могут храниться в ячейках таблицы?
9. Сравните операции ввода, редактирования и форматирования текстовой информации в текстовом процессоре и в электронных таблицах.
10. Сравните возможности ввода чисел в таблицы в текстовом процессоре и в электронных таблицах.
11. В одной из ячеек электронной таблицы записано арифметическое выражение $50 + 25 / (4 \cdot 10 - 2) \cdot 8$. Какое математическое выражение ему соответствует?
 - а) $50 + \frac{25}{4} \cdot 10 - 2 \cdot 8$
 - б) $\frac{50 + 25}{4 \cdot 10 - 2} \cdot 8$
 - в) $\frac{50 + 25}{(4 \cdot 10 - 2) \cdot 8}$
 - г) $50 + \frac{25}{4 \cdot 10 - 2} \cdot 8$
12. Сформулируйте правила ввода формул в электронных таблицах.
13. Почему электронные таблицы часто называют динамическими?
14. Сравните электронные таблицы и таблицы реляционной базы данных: что в них общее? В чём основное различие?
15. Дайте краткую характеристику режимов формирования электронных таблиц.
16. На рисунке 3.3 дан фрагмент таблицы в режиме отображения формул. Как будет выглядеть этот фрагмент в режиме отображения значений? Убедитесь в правильности своего ответа, воспользовавшись имеющимся в вашем распоряжении табличным процессором.



Организация вычислений в электронных таблицах

Ключевые слова:

- относительная ссылка
- абсолютная ссылка
- смешанная ссылка
- встроенная функция
- логическая функция
- условная функция

Основным назначением электронных таблиц является организация всевозможных вычислений. Вы уже знаете, что:

- вычисление — это процесс расчёта по формулам;
- формула начинается со знака равенства и может включать в себя знаки операций, числа, ссылки и встроенные функции.

Рассмотрим вначале вопросы, касающиеся организации вычислений в электронных таблицах.

3.2.1. Относительные, абсолютные и смешанные ссылки

Ссылка указывает на ячейку или диапазон ячеек, содержащих данные, которые требуется использовать в формуле. Ссылки позволяют:

- использовать в одной формуле данные, находящиеся в разных частях электронной таблицы;
- использовать в нескольких формулах значение одной ячейки.

Различают два основных типа ссылок:

- 1) относительные — зависящие от положения формулы;
- 2) абсолютные — не зависящие от положения формулы.

Различие между относительными и абсолютными ссылками проявляется при копировании формулы из текущей ячейки в другие ячейки.

Относительные ссылки. Присутствующая в формуле относительная ссылка определяет расположение ячейки с данными относительно ячейки, в которой записана формула. При изменении позиции ячейки, содержащей формулу, изменяется и ссылка.

Рассмотрим формулу $=A1^2$, записанную в ячейке A2. Она содержит относительную ссылку A1, которая воспринимается табличным процессором следующим образом: содержимое ячейки, находящееся на одну строку выше той, в которой находится формула, следует возвести в квадрат.

При копировании формулы вдоль столбца и вдоль строки относительная ссылка автоматически корректируется так:

- смещение на один столбец приводит к изменению в ссылке одной буквы в имени столбца;
- смещение на одну строку приводит к изменению в ссылке номера строки на единицу.

Например, при копировании формулы из ячейки A2 в ячейки B2, C2 и D2 относительная ссылка автоматически изменяется и рассмотренная выше формула приобретает вид: $=B1^2$, $=C1^2$, $=D1^2$. При копировании этой же формулы в ячейки A3 и A4 получим соответственно $=A2^2$, $=A3^2$ (рис. 3.4).

	A	B	C	D
1	2	3	4	5
2	$=A1^2$	$=B1^2$	$=C1^2$	$=D1^2$
3	$=A2^2$			
4	$=A3^2$			

Рис. 3.4. Скопированная формула с относительной ссылкой



Пример 1. В 8 классе мы рассматривали задачу о численности населения некоторого города, ежегодно увеличивающейся на 5%. Проведём в электронных таблицах расчёт предполагаемой численности населения города в ближайшие 5 лет, если в текущем году она составляет 40 000 человек.

Внесём в таблицу исходные данные, в ячейку B3 введём формулу $= B2+0,05*B2$ с относительными ссылками; скопируем формулу из ячейки B3 в диапазон ячеек B4:B7 (рис. 3.5).

	А	В		А	В	
1	Год	Численность населения		1	Год	Численность населения
2	Текущий	40 000		2	Текущий	40 000
3	1	=B2+0,05*B2		3	1	42 000
4	2	=B3+0,05*B3		4	2	44 100
5	3	=B4+0,05*B4		5	3	46 305
6	4	=B5+0,05*B5		6	4	48 620
7	5	=B6+0,05*B6		7	5	51 051

Рис. 3.5. Вид таблицы расчёта численности населения в режиме отображения формул и режиме отображения значений

Ежегодный расчёт численности населения мы (согласно условию задачи) осуществляли по одной и той же формуле, исходные данные для которой всегда находились в ячейке, расположенной в том же столбце, но на одну строку выше, чем расчётная формула. При копировании формулы, содержащей относительные ссылки, нужные нам изменения осуществлялись автоматически.

Абсолютные ссылки. Абсолютная ссылка в формуле всегда ссылается на ячейку, расположенную в определённом (фиксированном) месте. В абсолютной ссылке перед каждой буквой и цифрой помещается знак \$, например \$A\$1. При изменении позиции ячейки, содержащей формулу, абсолютная ссылка не изменяется. При копировании формулы вдоль строк и вдоль столбцов абсолютная ссылка не корректируется (рис. 3.6).

	А	В	С	Д
1	2	3	4	5
2	= \$A\$1^2	= \$A\$1^2	= \$A\$1^2	= \$A\$1^2
3	= \$A\$1^2			
4	= \$A\$1^2			

Рис. 3.6. Скопированная формула с абсолютной ссылкой

Пример 2. Некий гражданин открывает в банке счёт на сумму 10 000 рублей. Ему сообщили, что каждый месяц сумма вклада будет увеличиваться на 1,2%. Для того чтобы узнать возможную сумму и приращение суммы вклада через 1, 2, ..., 6 месяцев, гражданин провёл следующие расчёты (рис. 3.7).



	А	В	С
1	Начальная сумма вклада:		10 000
2	Месяц	Сумма	Приращение
3	1	=С1+С1*0,012	=В3-С\$1
4	2	=В3+В3*0,012	=В4-С\$1
5	3	=В4+В4*0,012	=В5-С\$1
6	4	=В5+В5*0,012	=В6-С\$1
7	5	=В6+В6*0,012	=В7-С\$1
8	6	=В7+В7*0,012	=В8-С\$1

Рис. 3.7. Расчёт приращения суммы вклада



Прокомментируйте формулы в таблице на рис. 3.7.



Выполните аналогичные расчёты для начального вклада, равного 15 000 рублям.

Смешанные ссылки. Смешанная ссылка содержит либо абсолютно адресуемый столбец и относительно адресуемую строку (\$A1), либо относительно адресуемый столбец и абсолютно адресуемую строку (A\$1). При изменении позиции ячейки, содержащей формулу, относительная часть адреса изменяется, а абсолютная часть адреса не изменяется.

При копировании или заполнении формулы вдоль строк и вдоль столбцов относительная ссылка автоматически корректируется, а абсолютная ссылка не корректируется (рис. 3.8).

	А	В	С	Д
1	2	3	4	5
2	=A\$1^2	=B\$1^2	=C\$1^2	=D\$1^2
3	=A\$1^2			
4	=A\$1^2			

Рис. 3.8. Скопированная формула со смешанной ссылкой



Чтобы преобразовать ссылку из относительной в абсолютную и наоборот, можно выделить её в строке ввода и нажать клавишу F4 (Microsoft Office Excel) или комбинацию клавиш Shift+F4 (OpenOffice.org Calc). Если выделить относительную ссылку, такую как A1, то при первом нажатии этой клавиши (комбинации клавиш) и для строки, и для столбца установятся абсолютные ссылки (\$A\$1). При втором нажатии абсолютную ссылку получит только строка (A\$1). При третьем нажатии абсолютную ссылку получит только столбец (\$A1). Если нажать клавишу F4 (комбинацию клавиш Shift+F4) ещё раз, то для столбца и строки снова установятся относительные ссылки (A1).

Пример 3. Требуется составить таблицу сложения чисел первого десятка, т. е. заполнить таблицу следующего вида:

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1									
3	2									
4	3									
5	4									
6	5									
7	6									
8	7									
9	8									
10	9									

При заполнении любой ячейки этой таблицы складываются соответствующие ей значения ячеек столбца A и строки 1. Иначе говоря, у первого слагаемого неизменным остаётся имя столбца (на него следует дать абсолютную ссылку), но изменяется номер строки (на неё следует дать относительную ссылку); у второго слагаемого изменяется номер столбца (относительная ссылка), но остаётся неизменным номер строки (абсолютная ссылка).

Внесите в ячейку B2 формулу $=\$A2+B\1 и скопируйте её на весь диапазон B2:J10. У вас должна получиться таблица сложения, знакомая каждому первокласснику.

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	2	3	4	5	6	7	8	9	10
3	2	3	4	5	6	7	8	9	10	11
4	3	4	5	6	7	8	9	10	11	12
5	4	5	6	7	8	9	10	11	12	13
6	5	6	7	8	9	10	11	12	13	14
7	6	7	8	9	10	11	12	13	14	15
8	7	8	9	10	11	12	13	14	15	16
9	8	9	10	11	12	13	14	15	16	17
10	9	10	11	12	13	14	15	16	17	18

3.2.2. Встроенные функции

При обработке данных в электронных таблицах можно использовать встроенные функции — заранее определённые формулы. Функция возвращает результат выполнения действий над значениями,