

Л. Л. Босова
А. Ю. Босова

ИНФОРМАТИКА

9

класс

БАЗОВЫЙ УРОВЕНЬ

Л. Л. Босова, А. Ю. Босова

ИНФОРМАТИКА

9 класс

БАЗОВЫЙ УРОВЕНЬ

Учебник

Допущено
Министерством просвещения
Российской Федерации

5-е издание, переработанное

Москва
«Просвещение»
2023

УДК 373.167.1:004+004(075.3)
ББК 32.81я721
Б85

Учебник допущен к использованию при реализации имеющих государственную аккредитацию образовательных программ начального общего, основного общего, среднего общего образования организациями, осуществляющими образовательную деятельность, в соответствии с Приказом Министерства просвещения Российской Федерации № 858 от 21.09.2022 г.

Босова, Людмила Леонидовна.
Б85 Информатика : 9-й класс : базовый уровень : учебник /
Л. Л. Босова, А. Ю. Босова. — 5-е изд., перераб. —
Москва : Просвещение, 2023. — 272 с. : ил.

ISBN 978-5-09-102544-6.

Учебник входит в состав УМК по информатике для 7–9 классов, включающего авторскую программу, учебники, электронные приложения, методическое пособие, рабочие тетради, сборники задач и другие компоненты. УМК может использоваться после вводного курса информатики в 5–6 классах или полностью самостоятельно обеспечивать освоение обязательного курса информатики в 7–9 классах, поддерживая базовую (1 ч/нед.) модель изучения предмета на уровне основного общего образования. Содержание учебника структурировано по темам «Разработка алгоритмов и программ», «Управление» (тематический раздел «Алгоритмы и программирование»), «Моделирование как метод познания» (тематический раздел «Теоретические основы информатики»), «Электронные таблицы» (тематический раздел «Информационные технологии»), «Информационно-коммуникационные технологии в современном обществе» (тематический раздел «Цифровая грамотность»). Теоретический материал поддержан развёрнутым аппаратом организации усвоения изучаемого материала, направленным на достижение обучающимися личностных, метапредметных и предметных образовательных результатов, обеспечивающим подготовку школьников к государственной итоговой аттестации по информатике в форме основного государственного экзамена (ОГЭ). Выдержан принцип инвариантности к конкретным моделям компьютеров и версиям программного обеспечения. Соответствует ФГОС ООО и Примерной рабочей программе по информатике для основного общего образования.

УДК 373.167.1:004+004(075.3)
ББК 32.81я721

ISBN 978-5-09-102544-6

© АО «Издательство «Просвещение», 2020, 2023
© Художественное оформление.
АО «Издательство «Просвещение», 2020, 2023
Все права защищены

ВВЕДЕНИЕ

Уважаемые девятиклассники!

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

В течение этого года вам предстоит:

- расширить свои знания, умения и навыки в области алгоритмизации и программирования, в том числе научиться разбивать сложные задачи на подзадачи; сравнивать новые задачи с задачами, решёнными ранее; определять шаги для достижения результата и т. д.
- развить представления о грамотной постановке задач, возникающих в практической деятельности, для их решения с помощью средств информационных и коммуникационных технологий (ИКТ);
- систематизировать представления об информационном моделировании как основном методе приобретения знаний;
- расширить знания и укрепить навыки использования средств ИКТ для сбора, хранения, преобразования и передачи информации различных видов;
- 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 информационных технологий.












Как и в предыдущих учебниках, на страницах учебника для 9 класса подробно рассмотрены решения типовых задач по каждой изучаемой теме. В конце каждой главы учебника при-

ведены тестовые задания, которые помогут вам оценить, хорошо ли вы освоили теоретический материал и можете ли применять свои знания для решения задач.

Электронное приложение к учебнику, содержащее видеоролики, мультимедийные презентации, интерактивные тесты, ссылки на ресурсы Интернета и файлы-заготовки для практических работ, размещено в авторской мастерской Л. Л. Босовой (<https://bosova.ru/>).

Изучая теоретический материал, работая с дополнительными материалами, отвечая на вопросы, решая задачи и выполняя практические задания на компьютере, вы сможете подготовиться к государственной итоговой аттестации по информатике в форме основного государственного экзамена (ОГЭ), требования к которому размещены на сайте Федерального института педагогических измерений (<https://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;
-  — вопросы в тексте параграфа, вопросы и задания для самоконтроля;
-  — задания для подготовки к итоговой аттестации;
-  — домашний проект или исследование;
-  — задания для выполнения на компьютере;
-  — групповая работа;
-  — межпредметные связи.

Желаем успехов в изучении информатики!

Глава 1

АЛГОРИТМЫ И ПРОГРАММИРОВАНИЕ

§ 1.1

Конструирование алгоритмов

Ключевые слова:

- пошаговая детализация
- вспомогательный алгоритм
- рекурсивный алгоритм
- формальные параметры
- фактические параметры

1.1.1. Методы построения алгоритмов

В 8 классе вы познакомились с базовыми алгоритмическими конструкциями (следованием, ветвлением, циклом), научились записывать несложные алгоритмы для формальных исполнителей и на одном из языков программирования.

При конструировании (разработке) алгоритма решения сложной задачи её принято разбивать на более простые подзадачи. При этом, как правило, используется один из следующих методов:

- 1) метод разработки «сверху вниз»;
- 2) метод разработки «снизу вверх».

Метод разработки «сверху вниз» иначе называется методом последовательного уточнения алгоритма или методом **пошаговой детализации**.

Процесс последовательного уточнения 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. 1.1).



Рис. 1.1. Линейный алгоритм, являющийся результатом первого шага последовательного уточнения алгоритма

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

- 1) задачу разбивают на несколько частей, каждая из которых проще всей задачи;
- 2) решение каждой части задачи формулируют в отдельной команде, которая также может выходить за рамки системы команд исполнителя;
- 3) 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.

При использовании метода разработки «снизу вверх» сначала разрабатывается библиотека — множество отдельных небольших алгоритмов, которые могут понадобиться для решения некоторого класса задач. После этого разрабатывается основной алгоритм, содержащий обращения к алгоритмам из библиотеки.

1.1.2. Разработка алгоритма методом пошаговой детализации 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
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 <команда 1> иначе <команда 2> все	Организация ветвления: если <условие> истинно, то выполняется <команда 1>; если <условие> ложно, то выполняется <команда 2>
нц пока <условие> <последовательность команд> кц	Организация цикла: пока <условие> истинно, выполняется <последовательность команд>

В одном условии можно использовать несколько команд, применяя логические операции **и**, **или**, **не**.

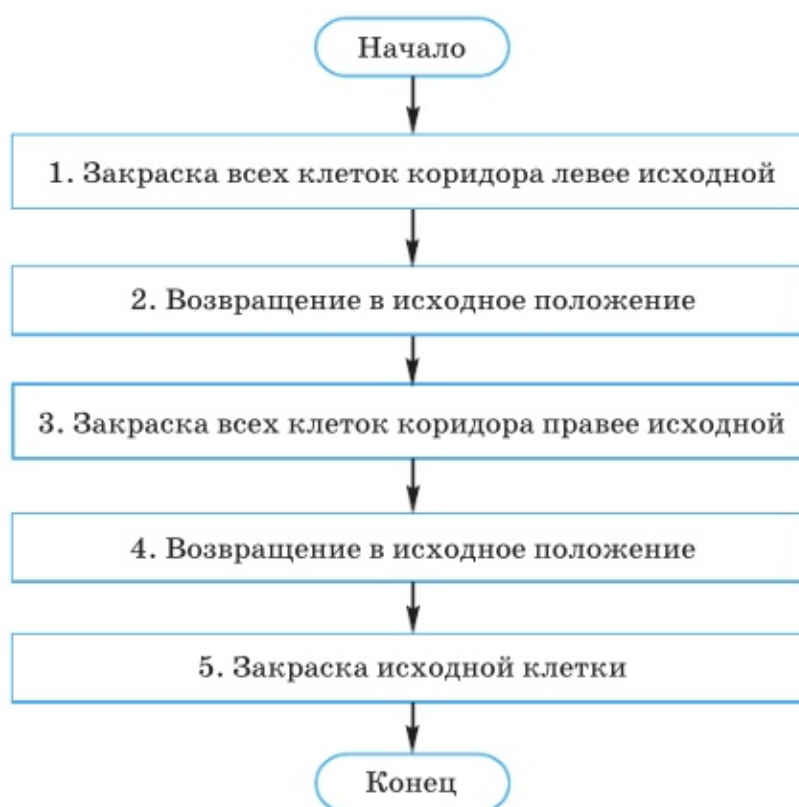
Известно, что Робот находится где-то в горизонтальном коридоре. Ни одна из клеток коридора не закрашена.



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 в клетке левее коридора:



2. Командой `вправо` вернём Робота в коридор. Вернём Робота в исходную клетку. Эта клетка — первая незакрашенная клетка, находящаяся правее Робота. Поэтому пока занимаемая Роботом клетка оказывается закрасенной, будем перемещать его вправо.

```
вправо
нц пока клетка закрасена
    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 закрасивать 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 клетку обеспечивается алгоритмом:

```
влево
нц пока клетка закрасена
    влево
кц
```



5. По команде закрасить Робот закрашивает исходную клетку.

Полностью программа управления Роботом выглядит так:

```
использовать Робот
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
```



Выполните приведённый выше алгоритм в среде *КуМир*, предварительно создав в ней стартовую обстановку — горизонтальный коридор.

Разработайте алгоритм решения исходной задачи при условии, что в системе команд Робота отсутствуют следующие команды:

```
сверху стена
снизу стена
слева стена
справа стена
```

Модифицируйте алгоритм таким образом, чтобы с его помощью можно было бы решить аналогичную задачу в вертикальном коридоре. Убедитесь в работоспособности разработанного алгоритма.

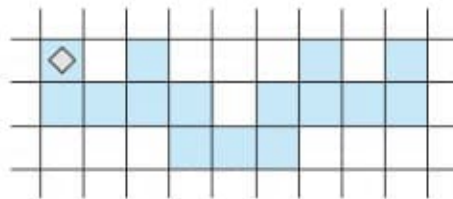
1.1.3. Вспомогательные алгоритмы

Во многих случаях при разработке алгоритмов решения сложных задач используют вспомогательные алгоритмы. Например, это удобно, когда в разных местах алгоритма необходимо выполнение одной и той же последовательности действий. Такую последовательность действий оформляют отдельным алгоритмом, который вызывают в нужных местах основного алгоритма. Один и тот же вспомогательный алгоритм может использоваться в разных программах. Вспомогательные алгоритмы делают более понятной структуру основного алгоритма.

Вспомогательный алгоритм — алгоритм, целиком используемый в составе другого алгоритма для решения некоторой подзадачи основной задачи.

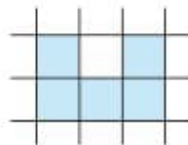
Пример 1

В среде *КуМир* составим алгоритм для исполнителя Робот, под управлением которого будет нарисован узор:



Начальное положение Робота отмечено ромбом; конечное положение значения не имеет.

Можно написать одну длинную программу, выполнив которую Робот закрасит все клетки, образующие этот узор. Но можно поступить иначе, обратив внимание на то, что в узоре трижды повторяется изображение одной и той же фигуры:



Разработав вспомогательный алгоритм для изображения этой фигуры, его можно вызвать нужное количество раз для построения требуемого узора.

```

использовать Робот
алг узор
нач
    фигура
    вправо; вниз
    фигура
    вправо; вверх
    фигура
кон
алг фигура
нач
    закрасить; вниз
    закрасить; вправо; закрасить; вправо
    закрасить; вверх; закрасить
кон

```



Выполните приведённый выше алгоритм в среде *КуМир*. Модифицируйте алгоритм таким образом, чтобы с его помощью можно было бы продолжить узор, дополнив его по горизонтали несколькими повторяющимися фигурами. Убедитесь в работоспособности разработанного алгоритма.

Каждый вспомогательный алгоритм получает уникальное имя, после которого указываются параметры алгоритма (при наличии) — имена переменных, обозначающих исходные данные (аргументы) и результаты вспомогательного алгоритма.

При представлении алгоритмов с помощью блок-схем для обозначения команды вызова вспомогательного алгоритма используется блок «предопределённый процесс» (рис. 1.2), внутри которого записывается название (имя) вспомогательного алгоритма, после которого в скобках перечисляются параметры — аргументы и результаты.

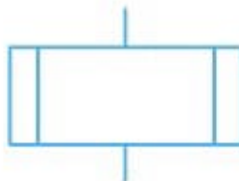


Рис. 1.2. Блок «предопределённый процесс»

Имя рассмотренного выше вспомогательного алгоритма — *фигура*; это вспомогательный алгоритм без параметров.

Пример 2

Исполнитель Черепаха перемещается на экране компьютера в квадратном окне со стороной 500 пикселей и может рисовать кончиком хвоста, оставляя след в виде линии. Исходное положение: Черепаха находится в центре окна, хвост опущен, направление — строго вверх.

Система команд исполнителя Черепаха:

Команда	Описание команды
поднять хвост	Последующие перемещения не оставляют следа в окне на экране
опустить хвост	Последующие перемещения оставляют след в окне на экране
вперед (n) назад (n)	Перемещение Черепахи на n шагов в направлении движения (в противоположном направлении); n — вещественное число
вправо (m) влево (m)	Изменение направления движения Черепахи на m градусов по часовой стрелке (против часовой стрелки); m — вещественное число
нц k раз <последовательность команд> кц	Черепаха повторяет k раз последовательность команд в скобках

Выясним, что получится в результате исполнения Черепахой следующего алгоритма:

```

нач
  нц 8 раз
    вперед (50)
    нц 3 раз
      вперед (50)
      вправо (120)
    кц
    назад (50)
    вправо (45)
  кц
кон
    
```

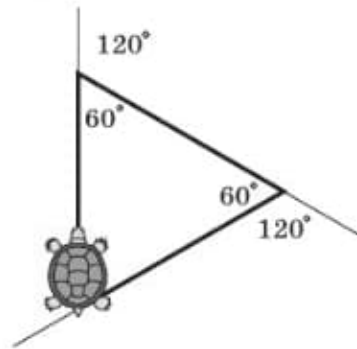
В этой программе два цикла, один из которых находится внутри другого. Вспомните, как называется такая конструкция.



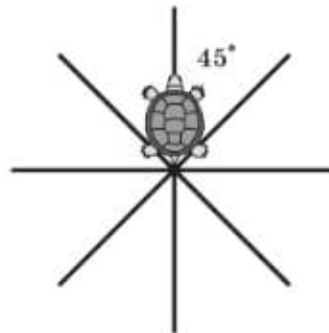


Мысленно разобьём задачу на две подзадачи, решаемые внутренним и внешним циклами соответственно.

Рассмотрим внутренний цикл. Результат его выполнения — равносторонний треугольник, нарисовав который Черепаха возвращается в исходное положение.



Результат работы внешнего цикла без учёта внутреннего цикла имеет вид:



Совместите мысленно результаты выполнения внешнего и внутреннего циклов. Опишите получившийся объект и дайте ему название.

Проверьте своё предположение, выполнив рассмотренный алгоритм в среде *КуМир*.

Оформите внутренний цикл в виде вспомогательного алгоритма без параметров **треугольник**. Убедитесь, что всё работает верно.

Можно модифицировать вспомогательный алгоритм **треугольник** во вспомогательный алгоритм с параметрами **многоугольник** (k, n), где k — количество углов, а n — длина стороны правильного многоугольника:

```

алг многоугольник (арг цел  $k, n$ )
нач
    нц  $k$  раз
        вперед ( $n$ )
        вправо ( $360/k$ )
    кц
кон
  
```

Здесь k, n — **формальные параметры**, они используются при описании вспомогательного алгоритма. При конкретном обращении к вспомогательному алгоритму (вызове вспомогательного алгоритма) 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.

Основной алгоритм, вызывающий вспомогательный алгоритм многоугольник(k, n), может выглядеть так:

```

алг
нач
  нц 8 раз
    вперед (50)
    многоугольник (5, 15)
    назад (50)
    вправо (45)
  кц
кон

```

Пример 3

Как известно, натуральные числа, бóльшие единицы, в зависимости от количества их делителей подразделяются на простые и составные:

- простое число — это натуральное число, бóльшее единицы, которое делится только на единицу и само на себя;
- составное число — это натуральное число, бóльшее единицы, которое делится не только на единицу и само на себя, но и ещё хотя бы на одно натуральное число.

На уроках математики вы неоднократно выполняли разложение натурального числа a на простые множители, скорее всего действуя так: находили p — первый простой делитель числа a (самое маленькое простое число, на которое данное число a делится без остатка) и делили на него число a ; находили первый простой делитель для полученного частного и делили частное на этот простой делитель и т. д., пока частное не становилось равным 1.

Нахождение первого простого делителя для некоторого числа — самостоятельная задача, которая может быть решена разными способами. Например, её можно решить подбором, воспользовавшись таблицей простых чисел и применив знание



таблицы умножения. В процессе разложения натурального числа на простые множители вы решали эту задачу неоднократно. Поэтому есть смысл представить её решение в виде самостоятельного алгоритма (рис. 1.3).

Здесь n — натуральное число, большее 1. Его первый простой делитель d находится перебором всех возможных делителей, начиная с 2 — наименьшего простого числа. Если само n окажется простым числом, то d будет присвоено значение n .

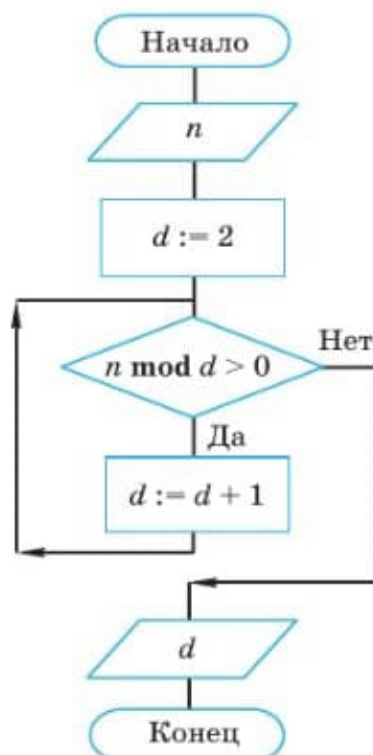


Рис. 1.3. Алгоритм нахождения первого простого делителя для некоторого числа

Попытайтесь самостоятельно записать алгоритм нахождения первого простого делителя, представленный на рис. 1.3, на известном вам языке программирования. Протестируйте свою программу на числах 121, 135 и 847. Если программа написана правильно, то результатами её выполнения будут числа 11, 5 и 7 соответственно.

Алгоритм нахождения первого простого делителя (см. рис. 1.3) можно рассматривать как вспомогательный при решении задачи разложения натурального числа на простые множители (рис. 1.4).

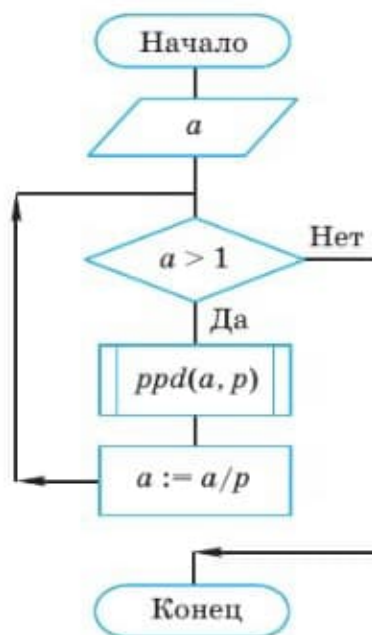


Рис. 1.4. Алгоритм разложения натурального числа на простые множители

Вспомогательный алгоритм нахождения первого простого делителя для некоторого числа мы назвали *ppd*. Его формальными параметрами являются переменные n, d ; фактическими — a, p .

В общем случае команда вызова вспомогательного алгоритма выполняется следующим образом (рис. 1.5):

- 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.

Имена и типы переменных, которые заявлены в качестве формальных параметров, объявляются внутри вспомогательного алгоритма по тем же правилам, что и для основного алгоритма. Такие переменные называются **локальными**. При необходимости работать с переменными и в основном алгоритме, и во вспомогательном их объявляют как **глобальные**.



Рис. 1.5. 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 4

Алгоритм вычисления степени с натуральным показателем n для любого вещественного числа a можно представить в виде рекурсивного (рис. 1.6).

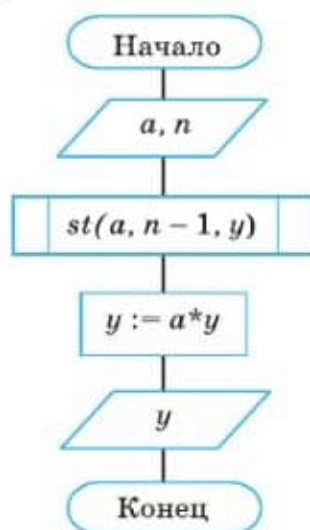


Рис. 1.6. Рекурсивный алгоритм возведения в степень

Число a в степени n (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 5

Рекурсивный алгоритм положен в основу эффективного решения головоломки «Ханойская башня» (рис. 1.7).

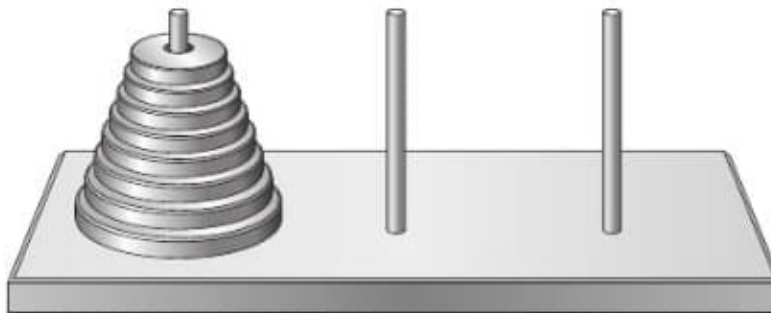


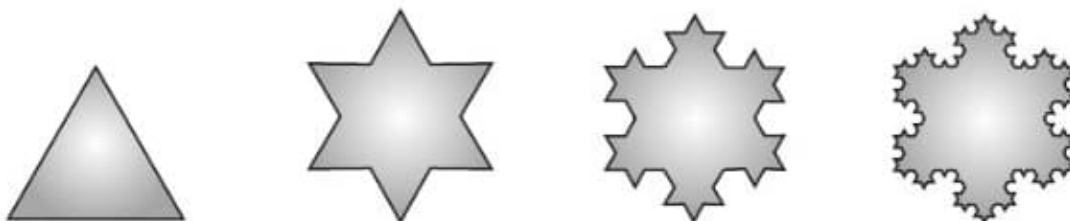
Рис. 1.7. Головоломка «Ханойская башня»

В сети Интернет размещено много разных онлайн-вариантов игры «Ханойская башня». Подберите какой-нибудь из них по своему вкусу и попытайтесь решить головоломку для восьми колец за минимально возможное количество ходов. При возникновении трудностей загляните в главу 3: подсказка содержится в задании 6 для практических работ.

www

Пример 6

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 рис. 1.8.



Начальное состояние Первый шаг Второй шаг Третий шаг

Рис. 1.8. Шаги алгоритма построения снежинки Коха

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



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

САМОЕ ГЛАВНОЕ

Один из основных методов конструирования алгоритмов решения сложных задач — метод пошаговой детализации. Его суть состоит в том, что исходная задача разбивается на несколько частей, каждая из которых проще всей задачи, и решение каждой части формулируется в отдельном предписании; если получаются предписания, 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 для решения некоторой подзадачи основной задачи.

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



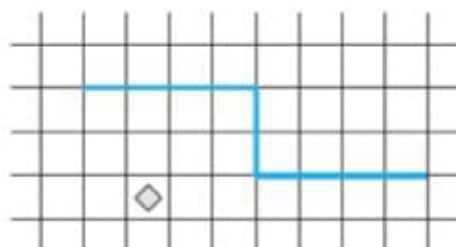
Вопросы и задания



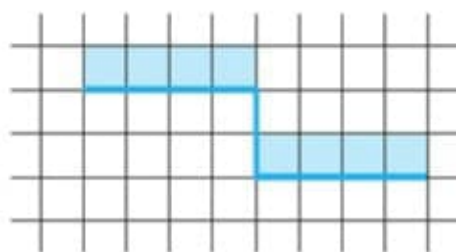
1. Почему при решении сложной задачи затруднительно сразу конкретизировать все необходимые действия? Обсудите этот вопрос в группе.
2. В чём заключается метод пошаговой детализации при построении алгоритма?
3. Какая связь между методом разработки алгоритма «сверху вниз» и такими процессами, как написание сочинения или подготовка к многодневному туристическому походу?
4. Известен рост каждого из n учеников 9 «А» класса и m учеников 9 «Б» класса. Опишите укрупнёнными блоками алгоритм сравнения среднего роста учеников этих классов.

5. На бесконечном поле имеется стена, состоящая из двух горизонтальных и одного вертикального участков, примерная форма которой представлена на рисунке. Длины участков неизвестны. Робот находится в клетке под левой горизонтальной частью стены.

На рисунке указан один из возможных вариантов расположения стены и Робота.



Задача Робота — закрасить клетки, примыкающие сверху к горизонтальным участкам стены. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки:



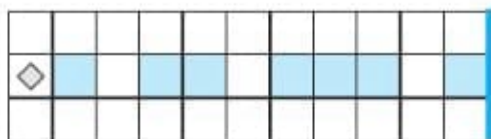
Конечное расположение Робота может быть произвольным. Задача должна быть решена для стены, горизонтальные и вертикальный участки которой могут иметь любые размеры.

Есть план действий Робота:

- 1) двигаться вверх до горизонтального участка стены;
- 2) двигаться влево под горизонтальным участком стены и сместиться на 1 клетку левее стены;
- 3) занять положение над горизонтальным участком стены;
- 4) закрасить клетки слева направо над горизонтальным участком стены;
- 5) спуститься вниз вдоль вертикального участка стены;
- 6) закрасить клетки слева направо над горизонтальным участком стены.

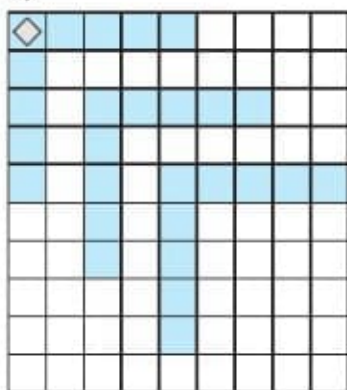
Реализуйте этот план, составив алгоритм для Робота в среде *КуМир*.

6. В ряду из десяти клеток правее Робота некоторые клетки закрашены. Последняя закрашенная клетка может примыкать к стене. Составьте алгоритм, который закрашивает клетки, примыкающие сверху и снизу к каждой закрашенной клетке. Проверьте работу алгоритма в следующих случаях:

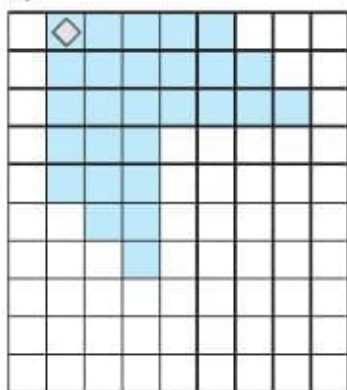


7. Для чего нужны вспомогательные алгоритмы? Опишите процесс выполнения команды вызова вспомогательного алгоритма в основном алгоритме.
8. Составьте алгоритмы, под управлением которых Робот закрашит указанные клетки. Используйте вспомогательный алгоритм.

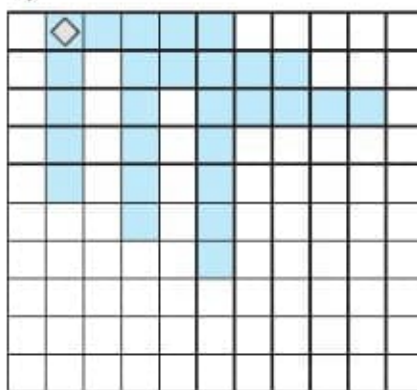
а)



б)

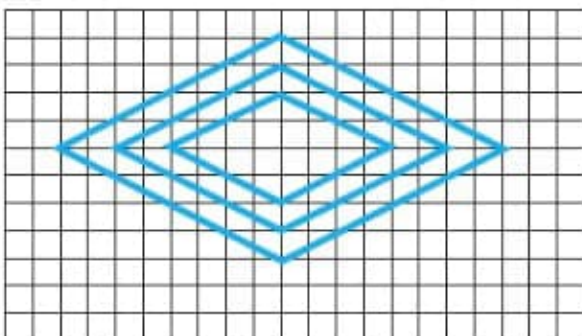


в)

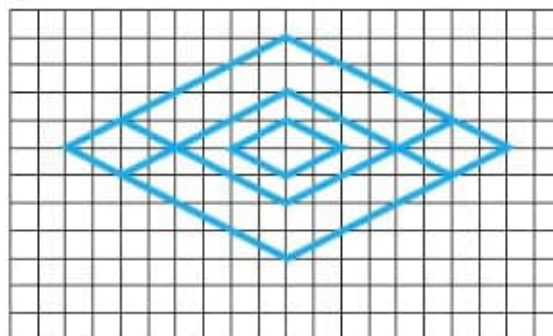


9. Сталкивались ли вы с идеей формальных и фактических параметров при изучении математики и физики? Приведите пример.
10. Для исполнителя Чертёжник в среде *КуМир* составьте алгоритмы рисования следующих фигур.

а)



б)



Воспользуйтесь вспомогательным алгоритмом с параметром ромб:

```

алг ромб (арг вещ  $x, y, a$ )
нач
    сместиться в точку  $(x, y)$ 
    опустить перо
    сместиться на вектор  $(2*a, a)$ 
    сместиться на вектор  $(-2*a, a)$ 
    сместиться на вектор  $(-2*a, -a)$ 
    сместиться на вектор  $(2*a, -a)$ 
    поднять перо
кон
    
```

Придумайте собственные рисунки на основе вспомогательного алгоритма с параметром ромб.

11. Какие алгоритмы называют рекурсивными? Приведите пример рекурсии из жизни.
12. В среде *КуМир* исследуйте алгоритм для Черепахи, содержащий рекурсию:

```

использовать Черепаха
алг чертеж (цел  $a$ )
нач
    если  $a < -150$ 
        то стоп
    все
    нц 4 раз
        вперед  $(a)$ ; вправо  $(90)$ 
    кц
    чертеж  $(a-5)$ 
кон
    
```

Проследите, что получится при изменении значений параметра a . Добавьте параметр k — количество углов основной фигуры. Предложите свои варианты рисунков для Черепахи, которые можно получить с помощью рекурсии.

§ 1.2

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 основной программы. Если подпрограмм несколько, то их описания располагаются в произвольном порядке одно за другим.

Структура описания подпрограммы аналогична структуре основной программы. Описание подпрограммы начинается с заголовка и заканчивается оператором **end**.

В Паскале различают два вида подпрограмм: процедуры и функции.

1.2.1. 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 типы. 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

Хорошо известный вам алгоритм Евклида для нахождения наибольшего общего делителя (НОД) двух чисел в виде процедуры можно описать так:

```

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;
    
```

Задача поиска НОД для последовательности натуральных чисел, количество членов которой *k* вводится с клавиатуры, может быть решена с помощью следующей программы:

<pre> var i, k, x, y, z: integer; </pre>	Переменные основной программы
<pre> 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; </pre>	Подпрограмма
<pre> begin read(k); read(x); for i := 2 to k do begin read(y); nod(x, y, z); x := z; end; writeln('НОД=', x) end. </pre>	Основная программа





Выполните приведённую выше программу в среде программирования PascalABC.NET. Протестируйте программу на последовательности из пяти чисел: 16, 32, 40, 80 и 128. Верно ли работает программа?

Пример 2



Рассмотрим программу, которая выводит все простые множители произвольного натурального числа $a > 1$ (если число простое, то выводится оно само), разработанную на основе алгоритма, блок-схема которого представлена на рис. 1.4. Блок-схема вспомогательного алгоритма представлена на рис. 1.3.



<code>var a, p: integer;</code>	Переменные основной программы
<code>..... procedure ppd(n: integer; var d: integer); begin d := 2; while n mod d > 0 do d := d + 1; writeln(d); end;</code>	Подпрограмма
<code>..... begin write('a='); readln(a); while a > 1 do begin ppd(a, p); a := a div p; end; end.</code>	Основная программа



Выполните приведённую выше программу в среде программирования PascalABC.NET. Протестируйте программу на числах 121, 135 и 847. В результате её выполнения вы должны получить наборы чисел, записанных в столбик: 11, 11; 3, 3, 3, 5; 7, 11, 11.

1.2.2. Функции



Функция — подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем 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 типы. Там же указывается тип самой функции, т. е. тип результата.

Так как единственный результат функции записывается в ячейку памяти, имя которой совпадает с именем функции, то 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 3

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

<pre>var a, b, c, d: integer;</pre>	Переменные основной программы
<pre>function max(x, y: integer): integer; begin if x > y then max := x else max := y end;</pre>	Подпрограмма
<pre>begin readln(a, b, c, d); writeln('MAX=', max(max(a,b), max(c,d))) end.</pre>	Основная программа



Выполните приведённую выше программу в среде программирования PascalABC.NET.

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

- максимальное из чисел a, b, c ;
- максимальное из чисел b, c, d ;
- минимальное из четырёх чисел;
- разность максимального и минимального из четырёх чисел.

Пример 4

В январе Саше 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 давала первый приплод (пару кроликов) через два месяца, а затем — по паре кроликов каждый месяц. Сколько пар кроликов будет у Саши в декабре?



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:

$$f(n) = f(n - 1) + f(n - 2), n \geq 3.$$



Выпишем в ряд несколько первых значений $f(n)$. 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 — **рекурсивная**; в ней реализован способ вычисления очередного значения функции через вычисление её предшествующих значений.



В среде программирования PascalABC.NET напишите и отладьте программу, вычисляющую и выводящую на экран 12 первых членов последовательности Фибоначчи.

САМОЕ ГЛАВНОЕ

Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью подпрограмм. 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. Для чего используются подпрограммы?
2. В чём основное различие процедур и функций?
3. Напишите функцию `kdn`, которая вычисляет количество цифр вводимого целого числа.
4. Напишите функцию `kbdn`, которая вычисляет количество цифр в двоичной записи вводимого десятичного числа.
5. Как известно, наименьшее общее кратное (НОК) и наибольший общий делитель двух чисел связаны соотношением:
$$\text{НОК}(a, b) = (a \cdot b) / \text{НОД}(a, b).$$
Напишите программу вычисления НОК следующих четырёх чисел: 36, 54, 18 и 15. Используйте процедуру вычисления НОД двух чисел.
6. Напишите программу обмена 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

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. Напишите программу, вычисляющую периметр треугольника по координатам его вершин с помощью этой функции.

10. Напишите функцию, вычисляющую площадь треугольника по координатам его вершин. Напишите программу вычисления площади четырёхугольника по координатам его вершин с помощью этой функции.

§ 1.3

Запись вспомогательных алгоритмов на языке Python

Ключевые слова:

- подпрограмма
- процедура
- функция
- рекурсивная функция

Запись вспомогательных алгоритмов в языках программирования осуществляется с помощью **подпрограмм**.

В Python различают два вида подпрограмм: процедуры и функции.

1.3.1. Процедуры

Процедура — подпрограмма, выполняющая некоторые действия; она может иметь произвольное количество входных параметров.

Описание процедуры имеет вид

```
def <имя процедуры>():  
    <операторы>
```

Процедура начинается со служебного слова **def** (от англ. *define* — определить). После этого записывается имя процедуры, скобки и двоеточие.

Операторы, которые входят в тело процедуры, записываются с отступом. Так мы показываем, какие команды входят в процедуру.

Для того чтобы процедура заработала, её необходимо вызвать по имени; причём таких вызовов может быть сколько угодно.

Процедура должна быть определена к моменту её вызова, т. е. должна быть выполнена команда **def**, создающая объект-процедуру в памяти. Если процедура вызывается из основной программы, то нужно поместить определение процедуры раньше точки вызова.

Пример 1

Предположим, требуется вывести четыре строки, каждая из которых состоит из семи единиц. Для этого создадим процедуру, которая выполняет вывод одной строки из семи единиц, и вызовем её четыре раза.

```
def digit():  
    print(1111111)  
print('Четыре строки из семи единиц')  
digit()  
digit()  
digit()  
digit()
```

Предположим, требуется вывести четыре строки, состоящие из семи, восьми, девяти и десяти единиц соответственно. Для этого создадим процедуру с параметром, определяющим длину строки.

Чтобы вывести единицу n раз, воспользуемся командой

```
print('1'*n)
```

Чтобы использовать эту команду в процедуре, укажем в скобках переменную (параметр), значение которой и будет определять длину строки.

```
def digit(n):
    print('1'*n)
```

Чтобы вывести строку требуемой длины, нужно вызвать процедуру, указав в скобках значение параметра n , т. е. количество символов '1'.

```
def digit (n):
    print('1'*n)
digit(7) #выводит 7 единиц: 1111111
digit(8) #выводит 8 единиц: 11111111
digit(9) #выводит 9 единиц: 111111111
digit(10) #выводит 10 единиц: 1111111111
```

Процедура может зависеть от нескольких параметров.

Давайте немного улучшим процедуру `digit`: сделаем так, чтобы можно было изменять не только длину строки, но и цифры, из которых эта строка строится. Для этого добавим в процедуру ещё один параметр, который назовём d :

```
def digit(d, n):
    print(d*n)
```

Обозначение переменных, значения которых мы вводим с клавиатуры, не обязательно должно совпадать с обозначением параметров процедуры. Мы могли бы назвать их любыми другими именами, например:

```
def digit(d, n):
    print(d*n)
x = input('Введите цифру:')
y = int(input('Введите длину строки:'))
digit(x, y)
```

Переменные `d` и `n` — это **локальные** переменные; они введены и используются внутри процедуры `digit`. Обращаться к ним вне этой процедуры нельзя. Как только работа процедуры будет закончена, все локальные переменные удалятся из памяти.

В тех случаях, когда значение переменной, полученное в подпрограмме, должно быть использовано в основной программе, эту переменную следует объявить как **глобальную**.

Выполните приведённую выше программу в любой доступной вам среде программирования на языке Python 3. Убедитесь, что всё работает верно.

Пример 2

Хорошо известный вам алгоритм Евклида для нахождения наибольшего общего делителя (НОД) двух чисел в виде процедуры можно описать так:

```
def nod(a, b):
    global x
    while a != b:
        if a > b:
            a = a - b
        else:
            b = b - a
    x = a
```

Задача поиска НОД для последовательности натуральных чисел, количество членов которой `k` вводится с клавиатуры, может быть решена с помощью следующей программы:

```
def nod(a, b):
    global x
    while a != b:
        if a > b:
            a = a - b
        else:
            b = b - a
    x = a
```

```
k = int(input('k='))
x = int(input())
for i in range(1, k):
    y = int(input())
    nod(x, y)
print('НОД=', x)
```

Подпрограмма

Основная программа



Выполните приведённую выше программу в любой доступной вам среде программирования на языке Python 3. Протестируйте программу на последовательности из пяти чисел: 16, 32, 40, 80 и 128. Верно ли работает программа?

Пример 3



Рассмотрим программу, которая выводит все простые множители произвольного натурального числа $a > 1$ (если число простое, то выводится оно само), разработанную на основе алгоритма, блок-схема которого представлена на рис. 1.4. Блок-схема вспомогательного алгоритма представлена на рис. 1.3.



<pre>def ppd(n): global d d = 2 while n % d > 0: d = d + 1 print(d)</pre>	Подпрограмма
<pre>a = int(input('a=')) while a > 1: ppd(a) a = a / d</pre>	Основная программа



Выполните приведённую выше программу в любой доступной вам среде программирования на языке Python 3. Протестируйте программу на числах 121, 135 и 847. В результате её выполнения вы должны получить соответствующие наборы чисел, записанных в столбик: 11, 11; 3, 3, 3, 7, 11, 11.

1.3.2. Функции



Функция — подпрограмма, имеющая единственный результат, записываемый в ячейку памяти.

В отличие от процедуры, функция не только выполняет какие-то команды, но и возвращает результат в виде числа, символьной строки или др.

Описание функции имеет вид

```
def <имя функции>():
    <операторы>
    return <результат>
```

Функция начинается со служебного слова **def**. После этого записывается имя функции, скобки и двоеточие. Операторы, которые входят в тело функции, записываются с отступом. После оператора **return** записывается результат, который возвращает функция.

В языке Python есть встроенная функция `max`, вычисляющая максимальное значение. Давайте посмотрим, как она работает.

Пример 4

Запишем функцию, которая возвращает значение наибольшего из двух целых чисел.

```
def max(a, b):  
    if a > b:  
        m = a  
    else:  
        m = b  
    return m
```

Результат функции можно сразу вывести на экран:

```
print(max(6, 8))
```

Также мы можем присвоить результат работы функции любой глобальной переменной:

```
x = max(6, 8)
```

Одна функция может вызывать другую.

Пример 5

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

```
def max(a, b):  
    if a > b:  
        m = a  
    else:  
        m = b  
    return m  
a, b, c, d = map(int, input().split())  
f = max(max(a, b), max(c, d))  
print('f=', f)
```



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

- максимальное из чисел a, b, c ;
- максимальное из чисел b, c, d ;
- минимальное из четырёх чисел;
- разность максимального и минимального из четырёх чисел.

Пример 6

В январе Саше 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?



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:

$$f(n) = f(n - 1) + f(n - 2), n \geq 3.$$



Выпишем в ряд несколько первых значений $f(n)$. 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.

```
def f(n):
    if n == 1 or n == 2:
        rez = 1
    else:
        rez = f(n - 1) + f(n - 2)
    return rez
```

Полученная функция — **рекурсивная**; в ней реализован способ вычисления очередного значения функции через вычисление её предшествующих значений.



Напишите программу, вычисляющую и выводящую 12 первых членов последовательности Фибоначчи.

САМОЕ ГЛАВНОЕ

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

Процедура — подпрограмма, выполняющая некоторые действия, она может иметь произвольное количество входных параметров. Процедура не возвращает результат, который можно присвоить переменной, а только обрабатывает входные данные. В процедуру могут быть переданы глобальные переменные, которые она обрабатывает, изменяет.

В отличие от процедуры, функция не только выполняет какие-то команды, но и возвращает результат в виде числа, символьной строки или др. Вызов функции можно использовать в арифметических выражениях и условиях так же, как и переменную такого же типа, который возвращает функция. В теле функции можно вызывать другие функции и процедуры.

Рекурсивная функция — это функция, которая вызывает сама себя, напрямую или через другие процедуры и функции.

Вопросы и задания

1. Для чего используются подпрограммы?
2. Напишите процедуру с параметрами n (целое число) и a (символ), выводящую на экран n строк, каждая из которых содержит n символов a .
3. Напишите процедуру с параметрами w (ширина), h (высота), a (символ), выводящую на экран «прямоугольник» из символов a , ширина которого равна w , а высота — h .
4. В чём основное различие процедур и функций?
5. Напишите функцию $kdn()$, которая вычисляет количество цифр вводимого целого числа.
6. Напишите функцию $kbdn()$, которая вычисляет количество цифр в двоичной записи вводимого десятичного числа.
7. Как известно, наименьшее общее кратное (НОК) и наибольший общий делитель двух чисел связаны соотношением:
$$\text{НОК}(a, b) = (a \cdot b) / \text{НОД}(a, b).$$



Напишите программу вычисления НОК следующих четырёх чисел: 36, 54, 18 и 15. Используйте процедуру вычисления НОД двух чисел.

8. Напишите программу обмена значений переменных a , b , c в порядке возрастания, т. е. так, чтобы $a < b < c$. Используйте функцию `swap()`.

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

9. 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.

10. Напишите программу вычисления выражения

$$s = x^3 + x^5 + x^n,$$

где x и n вводятся с клавиатуры. Используйте подпрограмму вычисления степени.



11. Напишите функцию, вычисляющую длину отрезка по координатам его концов. Напишите программу, вычисляющую периметр треугольника по координатам его вершин с помощью этой функции.



12. Напишите функцию, вычисляющую площадь треугольника по координатам его вершин. Напишите программу вычисления площади четырёхугольника по координатам его вершин с помощью этой функции.

§ 1.4

Обработка одномерных массивов целых чисел на языке Паскаль

Ключевые слова:

- массив
- элемент массива
- индекс элемента
- значение элемента
- описание массива
- заполнение массива
- вывод массива
- обработка массива
- последовательный поиск
- сортировка

До сих пор мы создавали программы с небольшим количеством переменных, каждая из которых получала своё уникальное имя. Но при решении многих практических задач возникает потребность в обработке таких больших объёмов однотипных данных, что дать имя каждой из используемых величин становится затруднительно. В таких случаях данные принято определённым образом организовывать в **структуры данных**. Одним из примеров структуры данных является массив.

Массив — это набор элементов одного типа, которым присвоено общее имя. Каждый элемент массива имеет свой номер (**индекс**).

С подобным подходом к именованию объектов — числовых последовательностей — вы уже встречались на уроках математики. Например, члены арифметической прогрессии обозначались так: $a_1, a_2, a_3, \dots, a_n$.

Выбор имён для массивов осуществляется по тем же правилам, что и для переменных.

Размерность массива — это количество индексов, необходимое для однозначного доступа к элементу массива. Массивы с одним индексом называют одномерными, с двумя — двумерными и т. д. Мы будем рассматривать одномерные массивы (рис. 1.9).



Рис. 1.9. Одномерный массив A из восьми элементов

Здесь:

A — имя массива;

$A[1] = -15$ (читается: «значение элемента массива A с индексом 1 равно -15 », « A первое равно -15 »), $A[2] = -20$, $A[5] = -2$.

Индексом может быть не только целое число, но и целое значение переменной или арифметического выражения. Так, в нашем примере при $i = 1$ $A[4*i-2] = -20$.

Индексом может быть даже значение элемента массива.

В нашем примере $A[A[8]] = -15$.

Длина массива — это количество его элементов. 8 — длина массива, представленного на рис. 1.9. Как правило, длину массива хранят в отдельной переменной: $N = 8$.

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.

Именно при работе с массивами становятся отчётливо видны возможности компьютера по быстрому выполнению различных операций над большими объёмами однотипных данных с помощью сравнительно коротких программ.

1.4.1. 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 [<нач_знач_индекса>..
    <кон_знач_индекса>] of <тип_элементов>;
```

На рис. 1.10 приведено описание массива A из 8 целочисленных значений. При выполнении этого оператора в памяти компьютера будет выделено место для хранения восьми целочисленных переменных.

```
var A: array [1..8] of integer;
```

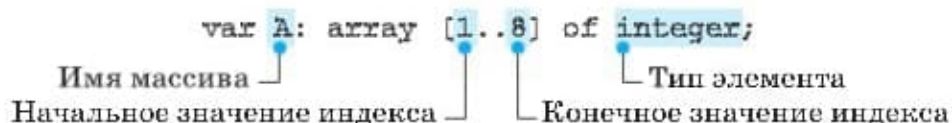


Рис. 1.10. Описание одномерного массива

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.

1.4.2. Заполнение массива

Рассмотрим несколько способов заполнения массива.

Для ввода с клавиатуры значений элементов описанного выше массива *A* используется следующий цикл с параметром:

```
for i := 1 to 8 do read A[i];
```

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

```
for i := 1 to 7 do A[i] := 2 * i;
```

Какие значения будут присвоены элементам массива *A*?



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

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

1.4.3. 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 элементов массива — в столбик и с пояснениями:

```
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>var</code> <code>i: integer;</code> <code>A: array [1..10] of integer;</code>	Блок описания переменных
<code>begin</code>	Основная программа
<code>for i := 1 to 10 do</code> <code>A[i] := random(100);</code>	Заполнение массива
<code>for i := 1 to 10 do</code> <code>writeln('A[' , i , ']= ' , A[i])</code>	Вывод массива
<code>end.</code>	



Выполните приведённую выше программу на компьютере.

1.4.4. Вычисление суммы элементов массива

Пример 1

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



Исходные данные (число жильцов) представим с помощью одномерного массива A , содержащего N элементов: $A[1]$ — число жильцов дома 1, $A[2]$ — число жильцов дома 2, ..., $A[N]$ — число жильцов дома N . В общем случае $A[i]$ — число жильцов дома i , где i принимает целочисленные значения от 1 до N (кратко обозначим это в блок-схеме как $i = 1, N$ (рис. 1.11)).

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:

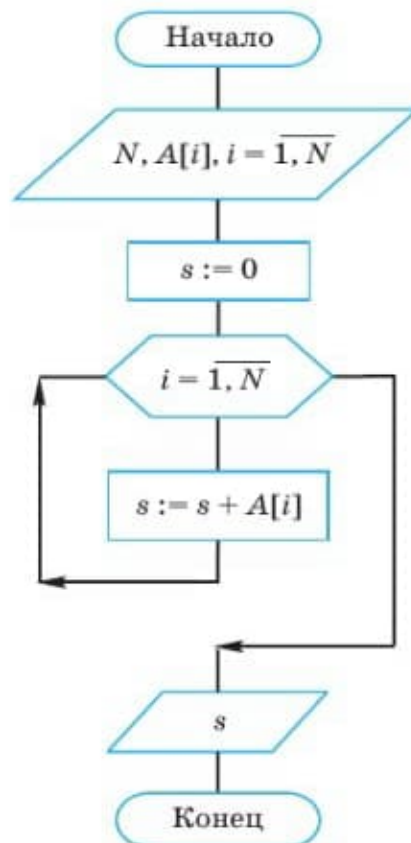


Рис. 1.11. Блок-схема алгоритма подсчёта числа жителей

- 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) N раз текущее значение переменной s складывается со значением текущего элемента массива $A[i]$; полученный результат присваивается переменной s .

Описанный процесс наглядно можно изобразить так:

Значение i	Операция присваивания	Текущее значение s
	$s := 0$	0
1	$s := s + A[1]$	$0 + A[1]$
2	$s := s + A[2]$	$0 + A[1] + A[2]$
3	$s := s + A[3]$	$0 + A[1] + A[2] + A[3]$
...
N	$s := s + A[N]$	$0 + A[1] + A[2] + A[3] + \dots + A[N]$

Запишем соответствующую программу на языке Паскаль.

<code>const</code>	Блок описания констант
<code> N = 20;</code>	
<code>var</code>	Блок описания переменных
<code> i, s: integer;</code>	
<code> A: array [1..N] of integer;</code>	
<code>begin</code>	Основная программа
<code> for i := 1 to N do</code>	Заполнение и вывод массива
<code> begin</code>	
<code> A[i] := random(100) + 50;</code>	
<code> writeln('A[' , i, ']=', A[i])</code>	
<code> end;</code>	
<code> s := 0;</code>	Вычисление суммы элементов массива
<code> for i := 1 to N do</code>	
<code> s := s + A[i];</code>	
<code> writeln('s=', s)</code>	Вывод результатов
<code>end.</code>	

Рассмотрим работу этого алгоритм для массива

$A = (100, 120, 130, 80, 70)$.

Значение i	Операция присваивания	Текущее значение s
	$s := 0$	0
$i = 1$	$s := s + A[1]$	100
$i = 2$	$s := s + A[2]$	220
$i = 3$	$s := s + A[3]$	350
$i = 4$	$s := s + A[4]$	430
$i = 5$	$s := s + A[5]$	500

Выполните программу на компьютере, считая число жильцов дома случайным числом из диапазона от 50 до 200 человек, а число домов $N = 30$.

Вы уже составляли программу, получавшую на вход количество чисел N в последовательности, а затем сами эти числа и вычислявшую сумму введённых чисел. Воспроизведите эту программу. Сравните её с программой суммирования элементов массива с точки зрения процессов, происходящих в памяти. Что у них общее? В чём отличие? Обсудите этот вопрос в группе.

1.4.5. 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;
- 3) найти количество (сумму) элементов массива, удовлетворяющих заданному условию.

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.

В программировании при обосновании корректности циклических алгоритмов используется понятие инварианта цикла.

Инвариант цикла — логическое выражение (условие), зависящее от переменных, изменяющихся в теле цикла; оно истинно непосредственно перед началом выполнения цикла и после каждого прохода тела цикла.



Условие «записанное на доске число — самое большое из всех просмотренных до сих пор» является инвариантом цикла для рассмотренного алгоритма.

Так как доступ к значению элемента массива осуществляется по его индексу, при организации поиска наибольшего элемента в одномерном массиве можно искать его индекс. Обозначим искомый индекс i_{\max} . Тогда описанный выше алгоритм поиска наибольшего элемента в массиве A на языке Паскаль можно записать так:

const N = 10;	Блок описания данных
var i, imax: integer; A: array [1..N] of integer;	
begin	Основная программа
for i := 1 to N do begin A[i] := random(100); writeln('A[' , i, ']=' , A[i]) end;	Заполнение и вывод массива
imax := 1; for i := 2 to N do if A[i] > A[imax] then imax := i;	Поиск индекса наибольшего элемента массива
writeln('Наибольший элемент: ', A[imax])	Вывод результата
end.	

Рассмотрим работу этого алгоритма для массива

$A = (100, 120, 130, 80, 70)$.

i_{\max}	i	$A[i] > A[i_{\max}]$
1	2	120 > 100 (Да)
2	3	130 > 120 (Да)
3	4	80 > 130 (Нет)
3	5	70 > 130 (Нет)
Наибольший элемент: 130		

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 она находила последний из максимальных элементов. Как следует преобразовать программу, чтобы с её помощью можно было найти минимальный элемент массива?

Проверьте свои предположения на компьютере.

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

- k — индекс элемента массива такой, что $A[k] = x$, 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>const</code>	Блок описания данных
<code> N = 10;</code>	
<code>var</code>	Основная программа
<code> i, k, x: integer;</code>	
<code> A: array [1..N] of integer;</code>	Заполнение и вывод массива
<code>begin</code>	
<code> for i := 1 to N do</code>	Ввод значения x
<code> begin</code>	
<code> A[i] := random(100);</code>	Поиск элемента с заданным свойством
<code> writeln('A[' , i , ']=', A[i])</code>	
<code> end;</code>	Вывод результата
<code> write('x=');</code>	
<code> read(x);</code>	Вывод результата
<code> k := 0;</code>	
<code> for i := 1 to N do</code>	Вывод результата
<code> if A[i] = x then k := i;</code>	
<code> if k = 0</code>	Вывод результата
<code> then writeln('Нет')</code>	
<code> else writeln('k=', k)</code>	Вывод результата
<code>end.</code>	

В этой программе последовательно просматриваются все элементы массива. Номер найденного элемента сохраняется в переменной k ; начальное значение, присваиваемое этой переменной, заведомо не совпадает ни с одним из значений индексов элементов массива. Если значение переменной k не изменилось в ходе выполнения цикла и осталось равным 0, то это означает, что в массиве нет элемента, равного x .

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 следующий фрагмент программы:

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

Выполнение этого алгоритма будет завершено в одном из двух случаев:

- 1) в массиве найден первый из элементов, равный заданному;
- 2) все элементы массива просмотрены.



Запишите полные тексты двух вариантов программы. Выполните программы 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.



Количество каких элементов массива подсчитывается с помощью следующего фрагмента программы?

```
k := 0;
for i := 1 to N do
  if A[i] mod 2 = 0 then k := k + 1;
writeln('k=', k)
```

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

Какому условию удовлетворяют элементы массива, значения которых суммируются с помощью следующего фрагмента программы?



```
s := 0;
for i := 1 to N do
  if (A[i] mod 2 = 0) and (A[i] mod 10 = 4) then s := s + A[i];
writeln('s=', s)
```

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



1.4.6. Сортировка массива

Под **сортировкой (упорядочением)** массива понимают перераспределение 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.

Разработано множество алгоритмов сортировки. Рассмотрим один из них¹ — **сортировку выбором**.

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

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

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

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
Итого:	9	6	5	4	3	2	1	0	



В этом массиве из 8 элементов операцию выбора максимального элемента мы проводили 7 раз. В массиве из N элементов такая операция будет проводиться $N - 1$ раз. Объясните почему.

Приведём фрагмент программы, реализующий описанный алгоритм:

```

for i := 1 to N 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 .

Разработайте подпрограммы `max` и `swap` для поиска максимального элемента и обмена значениями двух элементов. Модифицируйте программу сортировки выбором, используя в ней одну или обе из названных подпрограмм.

1.4.7. Массивы и последовательности целых чисел

Если требуется обработать некоторое множество однотипных целочисленных данных, то в зависимости от решаемой задачи можно пойти одним из двух путей:

- 1) сохранить все элементы множества в памяти компьютера как массив и организовать обработку массива;
- 2) завести одну переменную, в которую последовательно считывать каждый отдельный элемент множества, и сразу же производить его обработку.

В первом случае (обработка массива) в памяти компьютера будут храниться все элементы рассматриваемого множества; при необходимости к любому из них можно обратиться из разных мест программы.

Во втором случае (обработка последовательности) в памяти компьютера будет храниться единственный элемент рассматриваемого множества — тот, который был введён последним; значения ранее введённых элементов множества будут утеряны безвозвратно. Такой вариант экономичен с точки зрения использования памяти, но он не применим для решения ряда задач.

Пример 2

Камера наблюдения, установленная в населённом пункте, регистрирует в автоматическом режиме скорость проезжающих мимо неё автомобилей, округляя полученные значения до целых чисел. Программа, которую вам необходимо составить, получает на вход значения скорости проехавших $N \leq 100$ автомобилей. Программа должна анализировать скорость каждого автомобиля и один раз в сутки выводить на экран следующую информацию:

- 1) количество автомобилей, проехавших со скоростью, превышающей 60 км/ч;
- 2) максимальную скорость проехавших автомобилей;
- 3) количество автомобилей, проехавших с максимальной скоростью;
- 4) среднюю скорость проехавших автомобилей;
- 5) количество автомобилей, проехавших со скоростью, ниже средней;
- 6) для каждого проехавшего автомобиля его скорость и комментарий:

- «Нарушитель», если скорость превышает 60 км/ч, но не превышает 250 км/ч;
- «Дисциплинированный водитель», если скорость от 30 до 60 км/ч;
- «Тише едешь — дальше будешь!», если скорость меньше 30 км/ч;
- «Ошибка», если значение скорости меньше 1 или больше 250.



Для получения каких из требуемых результатов можно организовать обработку последовательности? Какие результаты можно получить, организовав обработку массива?



Напишите программу, решающую поставленную задачу.

1.4.8. Другие структуры данных

Многие современные приложения (диалоговые, сетевые, инструментальные системы, операционные системы и др.) работают с данными, объём которых заранее не может быть ограничен определённой величиной. Предположим, разрабатывается большой программный комплекс, при работе которого в оперативной памяти будет храниться большое количество различных данных, представленных в форме массивов. Область памяти, отводимая для каждого массива, непрерывна; границы области во время выполнения программы строго фиксированы. Так как объём данных заранее неизвестен, программистам придётся указывать максимально возможные размеры используемых массивов. В результате этого для хранения всех возможных данных может оказаться недостаточно доступной памяти. При этом на практике крайне редко будут встречаться ситуации, когда каждый массив будет полностью заполнен: во многих из них часть зарезервированной памяти будет оставаться свободной. Жёсткие границы не позволяют перераспределять пустое пространство одних массивов в пользу других. Как результат — неэффективное использование оперативной памяти. Выходом в этой ситуации является замена при проектировании программы некоторых массивов на списки, которые занимают именно столько памяти, сколько действительно нужно в данный момент, и не создают никаких запасов.

Список представляет собой множество элементов, которые могут быть разбросаны операционной системой по оперативной памяти как угодно. Связь элементов списка осуществляется за счёт того, что каждый элемент списка содержит, кроме данных, адрес элемента, следующего за ним в списке.

Линейный односвязный список — последовательность линейно связанных элементов, для которых разрешены операции добавления элемента в произвольное место списка и удаления любого элемента.

В линейном списке для каждого элемента, кроме первого, есть предыдущий элемент; для каждого элемента, кроме последнего, есть следующий элемент. Таким образом, все элементы списка упорядочены.

В линейном списке можно обойти все элементы, только двигаясь последовательно от текущего элемента к следующему, начиная с первого. Прямой доступ к i -му элементу невозможен; список — структура с последовательным доступом. В отличие от списка массив является структурой с произвольным доступом.

САМОЕ ГЛАВНОЕ

Для работы с большим количеством однотипных данных используются массивы.

Массив — это набор элементов одного типа, которым присвоено общее имя. Каждый элемент массива имеет свой номер (индекс).

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

```
var <имя_массива>: array [<нач_знач_индекса>..  
    <кон_знач_индекса>] of <тип_элементов>;
```

Заполнять массив можно разными способами: вводя значение каждого элемента с клавиатуры, 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.

Для вычисления суммы элементов массива используется переменная, в которой накапливается сумма. Начальное значение этой переменной равно нулю.

При подсчёте элементов, удовлетворяющих условию, используется переменная, которая увеличивается на 1 каждый раз, когда найден новый подходящий элемент. Начальное значение этой переменной равно нулю.



Вопросы и задания








1. Объясните разницу между понятиями «индекс элемента массива» и «значение элемента массива».
2. Может ли массив одновременно содержать целые и вещественные значения?
3. Для чего предназначены массивы?
4. Для чего необходимо описание массива?
5. Что вы можете сказать о массиве, сформированном следующим образом?
 - а) `for i := 1 to N do A[i] := random(101) - 50;`
 - б) `for i := 1 to N do A[i] := i;`
 - в) `for i := 1 to N do A[i] := 2 * i - 1;`
6. Какие преимущества обеспечивает хранение длины массива в отдельной переменной?
7. Напишите два варианта программы, вычисляющей среднюю за неделю температуру воздуха, организовав:
 - а) обработку последовательности;
 - б) обработку массива.
 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



8. Дан массив из десяти целых чисел. Напишите программу подсчёта:
 - а) количества чётных элементов массива;
 - б) суммы нечётных элементов массива;
 - в) количества элементов массива, имеющих максимальное значение.

9. В классе 20 учеников писали диктант по русскому языку. Напишите программу, подсчитывающую количество двоек, троек, четвёрок и пятёрок, полученных за диктант. 
10. 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. Напишите программу, считая рост претендента в команду случайным числом из диапазона от 150 до 200 см, а число претендентов $N = 50$. 
11. 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). Напишите программу, которая по имеющимся данным определит треугольник с наибольшей площадью и выведет его номер, длины катетов и площадь. Предусмотрите случай, когда таких треугольников несколько. 

12. Занесите информацию о десяти странах мира в массивы $NAME$ (название страны), K (численность населения), S (площадь страны). Напишите программу, выводящую названия стран в порядке возрастания плотности их населения. 

13. Найдите информацию о таких частных случаях списка, как стек и очередь. Подготовьте короткое сообщение. 

§ 1.5

Обработка одномерных массивов целых чисел на языке Python

Ключевые слова:

- массив
- элемент массива
- индекс элемента
- значение элемента
- заполнение массива
- вывод массива
- обработка массива
- последовательный поиск
- сортировка

До сих пор мы создавали программы с небольшим количеством переменных, каждая из которых получала своё уникальное имя. Но при решении многих практических задач возникает потребность в обработке таких больших объёмов однотипных данных, что дать имя каждой из используемых величин становится затруднительно. В таких случаях данные принято определённым образом организовывать в **структуры данных**. Одним из примеров структуры данных является массив.



Массив — это набор элементов одного типа, которым присвоено общее имя. Каждый элемент массива имеет свой номер (**индекс**).



С подобным подходом к именованию объектов — числовых последовательностей — вы уже встречались на уроках математики. Например, члены арифметической прогрессии обозначались так: $a_1, a_2, a_3, \dots, a_n$.

Выбор имён для массивов осуществляется по тем же правилам, что и для переменных.

Размерность массива — это количество индексов, необходимое для однозначного доступа к элементу массива. Массивы с одним индексом называют одномерными, с двумя — двумерными и т. д. Мы будем рассматривать одномерные массивы (рис. 1.12).

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.

Именно при работе с массивами становятся отчётливо видны возможности компьютера по быстрому выполнению различных операций над большими объёмами однотипных данных с помощью сравнительно коротких программ.



В языке Python нет такой структуры данных, как массив; для хранения группы однотипных объектов используют **списки** — объекты типа `list`. Далее, говоря о списках, мы будем использовать слово «массив».



В отличие от массивов список — это динамическая структура: количество элементов списка можно изменять во время выполнения программы, удаляя и добавляя элементы. Кроме того, в отличие от массива в списке можно хранить данные разных типов.

1.5.1. Обращение к элементу массива

Для того чтобы обратиться к элементу массива, записывают имя массива, а после него в квадратных скобках указывают индекс (номер) нужного элемента. Нумерация элементов массива в Python всегда начинается с нуля.

Рассмотрим пример одномерного массива (см. рис. 1.12).



Рис. 1.12. Одномерный массив A из восьми элементов

Здесь:

A — имя массива;

$A[0] = -15$ (читается: «значение элемента массива A с индексом 0 равно -15», «A нулевое равно -15»), $A[1] = -20$, $A[2] = -12$, $A[4] = -2$.

Индексом может быть не только целое число, но и целое значение переменной или арифметического выражения. Так, в нашем примере при $i = 1$ $A[4*i-2] = -12$.

Индексом может быть даже значение элемента массива.

В нашем примере $A[A[6]] = -15$.

Длина массива (количество элементов массива) определяется с помощью функции `len`: $N = \text{len}(A)$. 8 — длина массива, представленного на рис. 1.12.

Как правило, длину массива хранят в отдельной переменной: $N = 8$.

1.5.2. Заполнение массива

Перед использованием в программе массив необходимо создать, в противном случае обращение к несуществующему элементу вызовет ошибку и аварийное завершение программы.

Создать массив можно перечислением элементов через запятую в квадратных скобках, например так:

```
A = [1, 2, -3, 5, 7].
```

С помощью записи `A = [0]*5` будет создан массив из пяти элементов, каждый из которых равен 0.

Небольшие массивы можно вводить с клавиатуры. Для этого можно использовать цикл с параметром, выполняющий оператор ввода отдельно для каждого элемента массива:

```
N = 5
A = [0]*N
for i in range(N):
    A[i] = int(input())
```

При каждом повторении цикла строка, введённая пользователем, преобразуется в целое число с помощью функции `int`, и это число добавляется к массиву. Пользователь сам должен следить за тем, значение какого именно элемента он вводит в тот или иной момент.

Чтобы перед вводом значения очередного элемента на экране появлялась подсказка с индексом этого элемента, можно использовать следующий цикл:

```
for i in range(N):
    print("A[{}]=".format(i), end="\n")
    A[i] = int(input())
```

В этом случае, например, при вводе значения элемента с индексом 2 на экран будет выведено `"A[2]="` и справа от знака `=` будет мигать курсор — приглашение к вводу.

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

```
for i in range(N):
    A[i] = 2 * i
```



Какие значения будут присвоены элементам массива `A`?

Для работы со случайными числами вначале нужно подключить функцию `randint` модуля `random`, генерирующую случайное целое число в заданном диапазоне:

```
from random import randint
```

Далее можно использовать цикл с параметром:

```
for i in range(N):
    A[i] = randint(10, 110)
```

Массив `A` из `N` элементов будет заполнен случайными числами, принадлежащими отрезку `[10, 110]`.

1.5.3. 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.

Самый простой способ — вывести массив как один объект:

```
print(A)
```

В этом случае весь массив выводится в квадратных скобках, а его элементы разделяются запятыми.

Можно вывести элементы массива на экран в строку, используя цикл:

```
for i in range(len(A)):
    print(A[i], end=" ")
```

Параметр `end` определяет, что после вывода каждого элемента добавляется пробел, а не символ перехода на новую строку.

Значения элементов массива можно вывести в столбик:

```
for i in range(len(A)):
    print(A[i])
```

Более наглядным является следующий вариант вывода с комментариями:

```
for i in range(N):
    print('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>N = 10</code>	⋮	Создание массива
<code>A = [0]*N</code>	⋮	
<code>from random import randint</code>	⋮	Подключение генератора случайных чисел
<code>for i in range(N):</code>	⋮	Заполнение массива
<code>A[i] = randint(0, 99)</code>	⋮	
<code>for i in range(N):</code>	⋮	Вывод массива
<code>print('A[' + i + ']=' + A[i])</code>	⋮	

Выполните приведённую выше программу на компьютере.



1.5.4. Вычисление суммы элементов массива

Пример 1

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

Исходные данные (число жильцов) представим с помощью одномерного массива A , содержащего N элементов: $A[0]$ — число жильцов дома 1; $A[1]$ — число жильцов дома 2; $A[N-1]$ — число жильцов дома N .

В общем случае $A[i]$ — число жильцов дома $i+1$, где i принимает целочисленные значения от 0 до $N-1$ (кратко обозначим это в блок-схеме как $i = \overline{0, N-1}$ (рис. 1.13)).

Результат работы алгоритма обозначен через s .

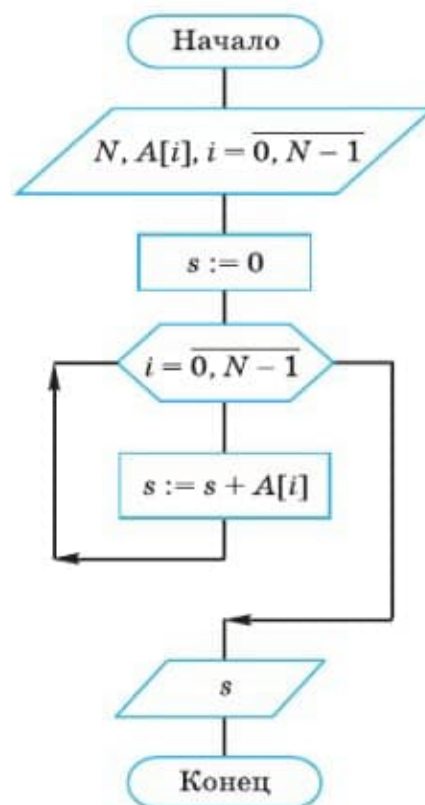


Рис. 1.13. Блок-схема алгоритма подсчёта числа жителей

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) N раз s складывается со значением текущего элемента массива $A[i]$; полученный результат присваивается переменной s .

Описанный процесс наглядно можно изобразить так:

Значение i	Операция присваивания	Текущее значение s
	$s := 0$	0
0	$s := s + A[0]$	$0 + A[0]$
1	$s := s + A[1]$	$0 + A[0] + A[1]$
2	$s := s + A[2]$	$0 + A[0] + A[1] + A[2]$
...
$N-1$	$s := s + A[N-1]$	$0 + A[0] + A[1] + A[2] + \dots + A[N-1]$

Запишем соответствующую программу на языке Python.

<code>N = 10</code> <code>A = [0]*N</code>	: Создание массива
<code>from random import randint</code>	: Подключение генератора случайных чисел
<code>for i in range(N):</code> <code> A[i] = randint(50, 200)</code>	: Заполнение массива
<code>for i in range(N):</code> <code> print('A[' + i + ']=', A[i])</code>	: Вывод массива
<code>s = 0</code> <code>for i in range(N):</code> <code> s += A[i]</code>	: Вычисление суммы элементов массива
<code>print('s=', s)</code>	: Вывод результата

Рассмотрим работу этого алгоритм для массива

$A = [100, 120, 130, 80, 70]$.

Значение i	Операция присваивания	Текущее значение s
	$s = 0$	0
$i = 0$	$s += A[0]$	100
$i = 1$	$s += A[1]$	220
$i = 2$	$s += A[2]$	350
$i = 3$	$s += A[3]$	430
$i = 4$	$s += A[4]$	500



Выполните программу на компьютере, считая число жильцов дома случайным числом из диапазона от 50 до 200 человек, а число домов $N = 30$.



Вы уже составляли программу, получавшую на вход количество чисел N в последовательности, а затем сами эти числа и вычислявшую сумму введённых чисел. Воспроизведите эту программу. Сравните её с программой суммирования элементов массива с точки зрения процессов, происходящих в памяти. Что у них общее? В чём отличие? Обсудите этот вопрос в группе.



1.5.5. 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;
- 3) найти количество (сумму) элементов массива, удовлетворяющих заданному условию.

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 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.

В программировании при обосновании корректности циклических алгоритмов используется понятие инварианта цикла.

Инвариант цикла — логическое выражение (условие), зависящее от переменных, изменяющихся в теле цикла; оно истинно непосредственно перед началом выполнения цикла и после каждого прохода тела цикла.

Условие «записанное на доске число — самое большое из всех просмотренных до сих пор» является инвариантом цикла для рассмотренного алгоритма.

Так как доступ к значению элемента массива осуществляется по его индексу, то при организации поиска наибольшего элемента в одномерном массиве можно запоминать (хранить) его индекс. Обозначим искомый индекс i_{\max} . Тогда описанный выше алгоритм поиска наибольшего элемента в одномерном массиве A на языке Python можно записать так:

<code>N = 10</code> <code>A = [0]*N</code>	Создание массива
<code>from random import randint</code>	Подключение генератора случайных чисел
<code>for i in range(N):</code> <code>A[i] = randint(0,99)</code> <code>print('A['+i+']= ' ,A[i])</code>	Заполнение и вывод массива
<code>imax = 0</code> <code>for i in range(1,N):</code> <code>if A[i] > A[imax]: imax = i</code>	Поиск индекса наибольшего элемента
<code>print('Наибольший элемент:', A[imax])</code>	Вывод результата

Рассмотрим работу этого алгоритм для массива

`A = [100, 120, 130, 80, 70]`.

<code>imax</code>	<code>i</code>	<code>A[i] > A[imax]</code>
0	1	120 > 100 (Да)
1	2	130 > 120 (Да)
2	3	80 > 130 (Нет)
2	4	70 > 130 (Нет)
Наибольший элемент: 130		



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 она находила последний из максимальных элементов. Как следует преобразовать программу, чтобы с её помощью можно было найти минимальный элемент массива? Проверьте свои предположения на компьютере.

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

- k — индекс элемента массива такой, что $A[k] = x$, 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>N = 10</code> <code>A = [0]*N</code>	Создание массива
<code>from random import randint</code>	Подключение генератора случайных чисел
<code>for i in range(N):</code> <code>A[i] = randint(0,99)</code> <code>print('A['+i+']= ', A[i])</code>	Заполнение и вывод массива
<code>x = int(input('x='))</code>	Ввод значения x
<code>nx = -1</code> <code>for i in range(0,N):</code> <code>if A[i] == x: nx = i</code>	Поиск индекса элемента, равного x
<code>if nx == -1:</code> <code>print('Нет')</code> <code>else:</code> <code>print('nx=', nx)</code>	Вывод результата

В этой программе последовательно просматриваются все элементы массива. Номер найденного элемента сохраняется в переменной `nx`; начальное значение, присваиваемое этой переменной, заведомо не совпадает ни с одним из значений индексов элементов массива. Если значение переменной `nx` не изменилось в ходе выполнения цикла и осталось равным `-1`, то это означает, что в массиве нет элемента, равного x .

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 следующий фрагмент программы:

```
nx = -1
for i in range(N):
    if A[i] == x:
        nx = i
        break
if nx >= 0:
    print('A[{}]={}'.format(nx,x))
else:
    print('Элемент не найден')
```

Используя цикл с параметром, мы начали последовательный перебор элементов массива и завершим его досрочно, как только будет найдено требуемое значение. Для выхода из цикла используется оператор **break**.

Здесь выполнение алгоритма будет завершено (прервано) в одном из двух случаев:

- 1) в массиве найден первый из элементов, равный заданному;
- 2) все элементы массива просмотрены.



В чём ещё отличие первого варианта программы от второго?



Запишите полные тексты двух вариантов программы. Выполните программы 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.



Количество каких элементов подсчитывается с помощью следующего фрагмента программы?

```
k = 0
for i in range(10):
    if A[i]%2 == 0: k += 1
print('k=', k)
```

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



Какому условию удовлетворяют элементы массива, значения которых суммируются с помощью следующего фрагмента программы?

```
s = 0
for i in range(10):
    if A[i]%2 == 0 and A[i]%10 == 4: s += A[i]
print('s=', s)
```



Запишите полные тексты двух программ и выполните их на компьютере.

1.5.6. Сортировка массива

Под сортировкой (упорядочением) массива понимают перераспределение 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.

Разработано множество алгоритмов сортировки. Рассмотрим один из них¹ — **сортировку выбором**.

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

- 1) в массиве выбирается максимальный элемент;
- 2) максимальный и первый элементы меняются местами, после чего первый элемент считается отсортированным;
- 3) в неотсортированной части массива снова выбирается максимальный элемент; он и первый неотсортированный элемент массива меняются местами;
- 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	0	1	2	3	4	5	6	7	
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	

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



В этом массиве из 8 элементов операцию выбора максимального элемента мы проводили 7 раз. В массиве из N элементов такая операция будет проводиться $N - 1$ раз. Объясните почему.

Приведём фрагмент программы, реализующий описанный алгоритм:

```
for i in range(N-1):
    imax = i
    for j in range(i+1, N):
        if A[j] > A[imax]: imax = j
    A[i], A[imax] = A[imax], A[i]
```

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 . Разработайте подпрограмму `max` для поиска максимального из двух элементов массива. Модифицируйте программу сортировки выбором, используя в ней эту подпрограмму.

1.5.7. Массивы и последовательности целых чисел

Если требуется обработать некоторое множество однотипных целочисленных данных, то в зависимости от решаемой задачи можно пойти одним из двух путей:

- 1) сохранить все элементы множества в памяти компьютера как массив и организовать обработку массива;
- 2) завести одну переменную, в которую последовательно считывать каждый отдельный элемент множества, и сразу же производить его обработку.

В первом случае (обработка массива) в памяти компьютера будут храниться все элементы рассматриваемого множества; при необходимости к любому из них можно обратиться из разных мест программы.

Во втором случае (обработка последовательности) в памяти компьютера будет храниться единственный элемент рассматриваемого множества — тот, который был введён последним; значения ранее введённых элементов множества будут утеряны безвозврат-

но. Такой вариант экономичен с точки зрения использования памяти, но он не применим для решения ряда задач.

Пример 2

Камера наблюдения, установленная в населённом пункте, регистрирует в автоматическом режиме скорость проезжающих мимо неё автомобилей, округляя полученные значения до целых чисел. Программа, которую вам необходимо составить, получает на вход скорости проехавших $N \leq 100$ автомобилей. Программа должна анализировать скорость каждого автомобиля и один раз в сутки выводить на экран следующую информацию:

- 1) количество автомобилей, проехавших со скоростью, превышающей 60 км/ч;
- 2) максимальную скорость проехавших автомобилей;
- 3) количество автомобилей, проехавших с максимальной скоростью;
- 4) среднюю скорость проехавших автомобилей;
- 5) количество автомобилей, проехавших со скоростью ниже средней;
- 6) для каждого проехавшего автомобиля его скорость и комментарий:
 - «Нарушитель», если скорость превышает 60 км/ч, но не превышает 250 км/ч;
 - «Дисциплинированный водитель», если скорость от 30 до 60 км/ч;
 - «Тише едешь — дальше будешь!», если скорость меньше 30 км/ч;
 - «Ошибка», если значение скорости меньше 1 или больше 250.

Для получения каких из требуемых результатов можно организовать обработку последовательности? Какие результаты можно получить, организовав обработку массива?

Напишите программу, решающую поставленную задачу.



САМОЕ ГЛАВНОЕ

Для работы с большим количеством однотипных данных используются массивы.

Массив — это набор элементов одного типа, которым присвоено общее имя. Каждый элемент массива имеет свой номер (индекс).

Нумерация элементов массива в Python начинается с нуля.

При обращении к элементу массива индекс записывают в квадратных скобках. Это может быть число, имя переменной целого типа или арифметическое выражение, результат которого — целое число.

Перед использованием в программе массив необходимо создать, в противном случае обращение к несуществующему элементу вызовет ошибку и аварийное завершение программы.

Заполнять массив можно, либо вводя значение каждого элемента с клавиатуры, либо присваивая элементам некоторые значения в программе. При заполнении массива и выводе его элементов на экран используется цикл с параметром, который изменяется от начального до конечного значения индекса.

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

Для вычисления суммы элементов массива используется переменная, в которой накапливается сумма. Начальное значение этой переменной равно нулю.

При подсчёте элементов, удовлетворяющих условию, используется переменная, которая увеличивается на 1 каждый раз, когда найден новый подходящий элемент. Начальное значение этой переменной равно нулю.



Вопросы и задания

1. Объясните разницу между понятиями «индекс элемента массива» и «значение элемента массива».
2. Может ли массив одновременно содержать целые и вещественные значения?
3. Для чего предназначены массивы?
4. Что вы можете сказать о массиве, сформированном следующим образом?

а) `for i in range(10): A[i] = random.randint(-50, 50)`

б) `for i in range(20): A[i] = i`

в) `for i in range(0, 5): A[i] = 2 * i - 1`

5. Какие преимущества обеспечивает хранение длины массива в отдельной переменной?
6. Напишите два варианта программы, вычисляющей среднюю за неделю температуру воздуха, организовав:
 - а) обработку последовательности;
 - б) обработку массива.

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. Напишите программу подсчёта:
 - а) количества чётных элементов массива;
 - б) суммы нечётных элементов массива;
 - в) количества элементов массива, имеющих 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. Напишите программу, считая 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). Напишите программу, которая по имеющимся данным определит треугольник с наибольшей площадью и выведет его номер, длины катетов и площадь. Предусмотрите случай, когда таких треугольников несколько.
11. Занесите информацию о десяти странах мира в массивы NAME (название страны), K (численность населения), S (площадь страны). Напишите программу, выводящую названия стран в порядке возрастания плотности их населения.
12. Найдите информацию о таких частных случаях списка, как стек и очередь. Подготовьте короткое сообщение.

§ 1.6 Алгоритмы управления

Ключевые слова:

- управление
- алгоритм управления
- обратная связь
- робот
- робототехника

1.6.1. 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.

1.6.2. 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 управляющему объекту.



Обратная связь позволяет корректировать управляющие воздействия управляющего объекта на объект управления в зависимости от состояния объекта управления (рис. 1.14). Обратная связь предусмотрена в ряде бытовых приборов (например, утюг с терморегулятором, холодильник, кастрюля-скороварка), в живых организмах, в обществе.



Рис. 1.14. Кибернетическая модель управления

В настоящее время в самых разных ситуациях всё чаще роль управляющего объекта 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.

1.6.3. Системы с программным управлением. Робототехника

Системы, в которых роль управляющего объекта поручается компьютеру, называются **автоматическими системами с программным управлением**. Наиболее впечатляющими примерами таких систем являются роботы.

Робот — автоматическое устройство, выполняющее определённые механические операции в соответствии с заложенной в него программой.

Робот — сложная система, включающая:

- 1) механическую часть — множество деталей, определённым образом скреплённых между собой;

- 2) подсистему питания — моторы, преобразующие электрическую энергию в механическую;
- 3) подсистему управления, состоящую из микроконтроллера — однокристалльного компьютера, программного обеспечения и множества датчиков.

Информацию о внешнем мире робот получает от датчиков — касания, расстояния, света, звука и др. Цифровые датчики измеряют соответствующие физические величины, преобразуют результаты измерений в двоичный код и передают полученный цифровой сигнал микроконтроллеру на обработку. Аналоговые датчики выдают информацию об измеряемой физической величине в виде электрических сигналов, которые преобразуются в двоичный код специальным блоком микроконтроллера.

Сигналы, поступающие от датчиков, — это входные данные для управляющей программы, заложенной в микроконтроллер. В зависимости от поступающих сигналов микропроцессор в соответствии с программой управляет устройствами вывода звука, лампочками, моторами и т. д., выполняя в итоге достаточно сложные операции, которые ещё десятилетие тому назад были под силу только человеку.

Сегодня в мире используются миллионы роботов. Они не только выполняют однообразные рутинные операции, которые не нравятся человеку, но и решают задачи, которые нам не под силу. Роботы управляют самолётами и поездами, помогают в строительстве космической станции, в сборке автомобилей и производстве микрочипов, охраняют здания, используются военными для разведки и разминирования, помогают спасателям искать людей под завалами. Нет такой области, в которой человек не попытался бы создать себе автоматического помощника.

По назначению можно выделить следующие основные виды роботов:

- промышленные роботы — автоматически управляемые, перепрограммируемые, многоцелевые манипуляторы, широко применяемые на производстве для выполнения ряда основных и вспомогательных технологических операций;
- медицинские роботы, используемые в том числе для проведения хирургических операций и приготовления лекарственных препаратов;

- бытовые роботы, являющиеся основой «умного дома» — роботизированной системы всего жизнеобеспечения, а также андройды — социальные роботы, предназначенные для помощи людям и общения с ними на определённые темы;
- роботы, обеспечивающие безопасность при ликвидации последствий чрезвычайных ситуаций;
- исследовательские роботы, используемые для сбора данных в экстремальных условиях;
- боевые роботы, в том числе беспилотные летательные аппараты;
- роботы, используемые в сфере образования.



Подготовьте небольшое сообщение об одном из видов роботов, приведите конкретные примеры их применения.

При большом разнообразии внешнего вида роботов и выполняемых ими функций есть нечто, присущее всем роботам; оно же отличает роботов от всех других автоматических устройств. Действительно, с помощью сенсоров робот получает информацию о состоянии среды, в которой он функционирует. На основании этой информации робот, разумеется по заложенной в него программе, принимает решение, выбирает модель поведения и воздействует на окружающий мир тем или иным доступным ему способом. Совокупность этих трёх качеств (способность воспринимать окружающий мир, способность понимать окружающий мир, способность воздействовать на окружающий мир) — отличительная черта современных роботов.

Робототехника — наука, занимающаяся разработкой автоматизированных технических систем, опирающаяся на информатику, а также на такие дисциплины, как:

- механика (движение материальных тел и взаимодействие между ними);
- телемеханика (контроль и управление объектами на расстоянии);
- электроника (методы создания электронных приборов и устройств для приёма, передачи, обработки и хранения информации);
- радиотехника (методы генерации, усиления, преобразования, излучения и приёма радиосигналов, проектирование и изготовление радиоаппаратуры).



Теоретические основы современной робототехники были заложены более 50 лет тому назад. Уже в то время нашими учёными и инженерами впервые в мире был создан планетоход — дистанционно управляемый самоходный аппарат «Луноход», около года проработавший на поверхности Луны, собирая и передавая на Землю данные об особенностях лунной поверхности, химическом составе грунта, космическом излучении и др.

В истории робототехники принято выделять этапы, связанные с созданием трёх поколений роботов:

- 1-е поколение — роботы без обратной связи, работавшие по программе, не реагирующей на изменения внешней среды, способные многократно выполнять одну и ту же последовательность действий;
- 2-е поколение — роботы с обратной связью, работающие по гибкой программе, способной выбирать оптимальный режим работы с учётом изменений внешней среды;
- 3-е поколение — интеллектуальные роботы, способные к самообучению и распознаванию образов.

На уроках информатики вы получили опыт разработки алгоритмов для управления формальными исполнителями Роботом, Чертёжником, Черепахой и др. Это тоже своего рода роботы, только виртуальные.

Для получения опыта работы с реальными роботами необходимы специальные конструкторы — наборы деталей, датчиков и управляющих устройств, снабжённые подробными инструкциями по сборке роботов. Возможно, на уроках технологии вы уже успели познакомиться с одним из таких конструкторов и собрать того или иного робота. Знаний, полученных на уроках информатики, достаточно для того, чтобы разобраться в системе команд этого робота и научить робота перемещаться, обходя препятствия.

Первые шаги в программировании роботов можно сделать даже в том случае, если в вашем распоряжении ещё нет конструктора. Для этого можно воспользоваться отечественной разработкой — бесплатной средой программирования роботов TRIK Studio (<https://gotourl.ru/15572>). В ней предусмотрен специальный режим работы, когда команды передаются не реальному роботу, а его виртуальной модели, отображающейся на экране компьютера.



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.

Робот — автоматическое устройство, действующее по заранее заложенной программе. Получая информацию о внешнем мире от датчиков (касания, расстояния, света, звука и др.), робот осуществляет производственные и иные операции, ранее выполнявшиеся человеком.



Вопросы и задания

1. Что такое управление? Приведите примеры управляющих систем и управляемых ими объектов.
2. Что такое алгоритм управления? Приведите примеры ситуаций, в которых имеют место линейные, разветвляющиеся и циклические алгоритмы управления.
3. Что изучает наука кибернетика?
4. Какая информация нужна для управления? Приведите пример.
5. Что такое обратная связь?
6. Опишите кибернетическую модель управления.
7. Чем роботы отличаются от других автоматических устройств и чем похожи на человека? В чём принципиальное отличие робота от человека?
8. Найдите информацию и подготовьте сообщение на одну из следующих тем:
 - а) «Прообразы современных роботов»;
 - б) «Откуда взяло начало слово «робот»»;
 - в) «Сферы применения робототехники».



Тестовые задания для самоконтроля

- При разработке алгоритмов зачастую используется метод пошаговой детализации. Как иначе называется этот метод?
 - Метод разработки «сверху вниз»
 - Метод разработки «снизу вверх»
- Алгоритм, целиком используемый в составе другого алгоритма для решения некоторой подзадачи основной задачи, называется:
 - рекурсивным
 - вспомогательным
 - основным
 - дополнительным
- Между формальными и фактическими параметрами при вызове вспомогательного алгоритма следует соблюдать соответствие:
 - по типу параметров
 - по количеству параметров
 - по порядку следования параметров
 - по всему перечисленному в п. а–в
- Для записи вспомогательных алгоритмов в языке Паскаль (Python) используются:
 - массивы
 - составные операторы
 - процедуры и функции
 - операторы и операнды
- Подпрограмма, имеющая единственный результат, называется:
 - процедурой
 - функцией
 - вспомогательным алгоритмом
- Дан текст процедуры на двух языках программирования:

Паскаль	Python
<pre> procedure f(n: integer); begin write('*'); if n > 1 then f(n div 2) end; </pre>	<pre> def f(n): print('*') if n > 1: f(n//2) </pre>

Сколько звёздочек будет выведено на экран в результате вызова $\text{z}(5)$?

- а) 1
 - б) 2
 - в) 3
 - г) 5
7. Набор элементов одного типа с общим именем, каждый из которых имеет свой номер, называется:
- а) множеством
 - б) последовательностью
 - в) индексом
 - г) массивом
8. Количество индексов, необходимое для однозначного доступа к элементу массива, называется:
- а) длиной
 - б) размерностью
 - в) индексом
 - г) мощностью
9. В массиве `Dat` хранятся данные измерений среднесуточной температуры за неделю в градусах. Определите, что будет вычислено в результате работы программы, фрагмент которой приведён на двух языках программирования.

Паскаль	Python
<pre> m := 0; n := 0; for k := 1 to 7 do if Dat[k] >= m then begin m := Dat[k]; n := k end; writeln(n) </pre>	<pre> m = n = -1 for k in range (7): if Dat[k] >= m: m = Dat[k] n = k print(n) </pre>

- а) Самая высокая температура за неделю
- б) Самая низкая температура за неделю
- в) Номер дня недели с самой высокой температурой
- г) Номер дня недели с самой низкой температурой

10. Кто считается основоположником кибернетики?
- а) Норберт Винер
 - б) Джон фон Нейман
 - в) Платон
 - г) И. П. Павлов
11. 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
12. Какое качество нельзя считать присущим всем современным роботам?
- а) Воспроизведение внешнего вида человека или животного
 - б) Способность воспринимать окружающий мир
 - в) Способность понимать окружающий мир
 - г) Способность воздействовать на окружающий мир

Глава 2

МОДЕЛИРОВАНИЕ

§ 2.1

Моделирование как метод познания

Ключевые слова:

- модель
- информационная модель
- моделирование
- формализация
- цель моделирования
- адекватность модели оригиналу
- материальная модель
- классификация моделей

2.1.1. 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 называть **моделью**, а исходный объект — **прототипом**, **объектом-оригиналом**, **объектом моделирования**. Примеры моделей приведены на рис. 2.1.

К созданию моделей прибегают, когда исследуемый объект слишком велик (Солнечная система) или слишком мал (атом), когда процесс протекает очень быстро (переработка топлива

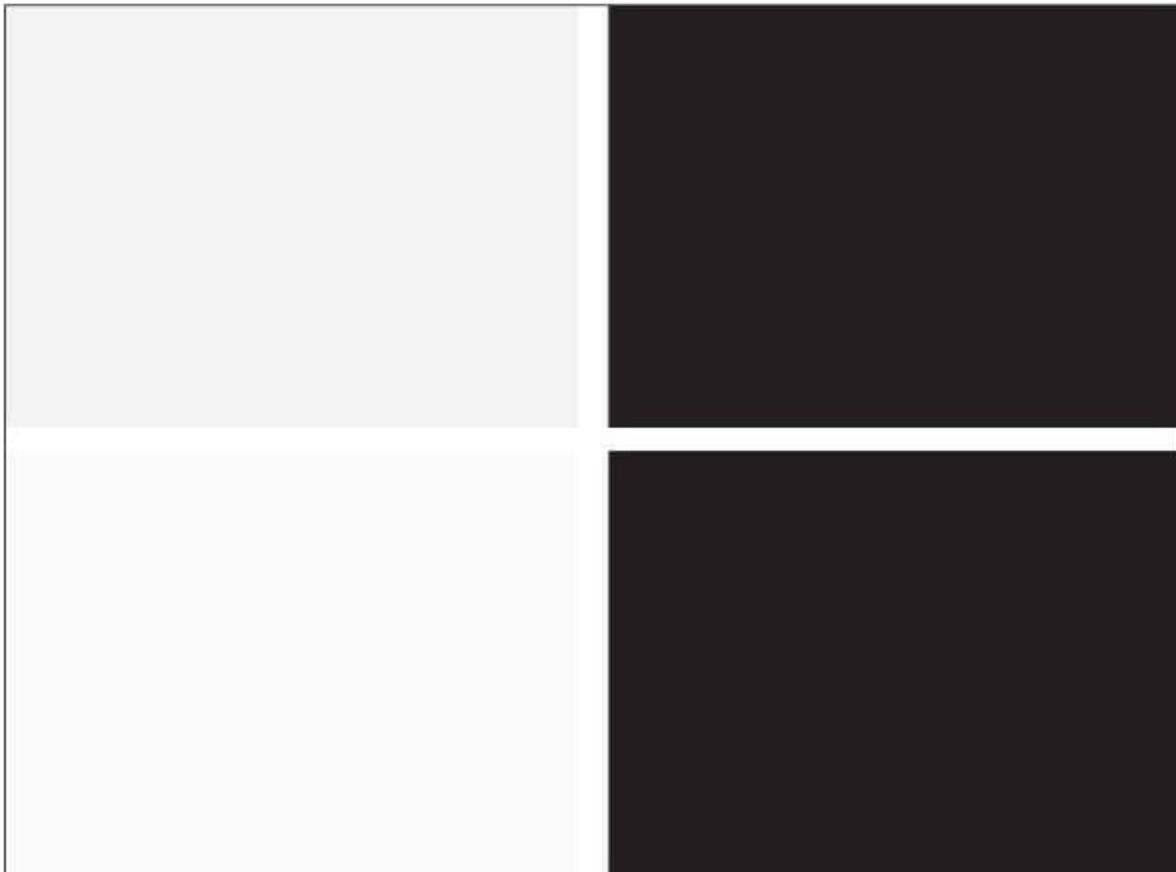


Рис. 2.1. Примеры моделей

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), 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:

- строительной компании, занимающейся возведением здания театра;
- режиссёра, готовящего постановку нового спектакля;
- кассира, продающего билеты;
- 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 (рис. 2.2).



Рис. 2.2. Разнообразие моделей

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 модели-

руемого 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.



2.1.2. Этапы построения информационной модели

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

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 (рис. 2.3). 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. 2.3. 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 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 постоянного числа ко времени, требуемому для заучивания предыдущей строки, надо сложить числа, образующие последовательность: 5, 7, 9, 11 и т. д. Заметим, что разность между соседними числами этой последовательности одна и та же.

В математике есть формула для вычисления суммы такой последовательности:

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

Здесь n — количество строк; a_1 — первый член последовательности; d — разность между соседними числами последовательности.



Формула суммы арифметической прогрессии и является искомой информационной моделью. С её помощью самостоятельно вычислите время, необходимое ученику для заучивания стихотворения.

Второй способ решения этой задачи — программирование. Разработайте циклический алгоритм для вычисления очередного члена последовательности и прибавления его к сумме. Запишите алгоритм на изучаемом языке программирования. Отладьте и выполните программу.



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.

При моделировании мы заменяем оригинал некоторой его моделью, исследуем эту модель и на основании полученных результатов делаем выводы об объекте моделирования. Но можно ли быть уверенным в том, что решение, найденное для модели, будет решением и для оригинала?

Адекватность модели объекту моделирования — это совпадение существенных для решения поставленной задачи признаков модели и соответствующих признаков оригинала.



Модель считается адекватной объекту моделирования, если результаты, полученные в процессе её исследования, близки к результатам, наблюдаемым в экспериментах с оригиналом.

Например, каждый новый разрабатываемый автомобиль подвергается серии краш-тестов (аварийных испытаний): автомобиль на скорости врезается в большое неподвижное препятствие; при столкновении кузов и другие части автомобиля могут так серьёзно деформироваться, что восстановить их будет невозможно. Ясно, что проведение большого количества краш-тестов потребует серьёзных финансовых затрат. Чтобы этого избежать, создают и исследуют компьютерную модель краш-теста. Она позволяет совершать многократные испытания, причём после каждого испытания возвращается в своё исходное состояние. Проведя серию экспериментов с моделью, проводят реальный краш-тест автомобиля. Если результаты экспериментов с моделью и реального эксперимента совпадают, то модель считают адекватной объекту моделирования и её исследование продолжают.

2.1.3. Классификация информационных моделей

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

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

По характеристике объекта моделирования различают модели **внешнего вида** (фотографии павильонов станций метро), модели **структуры** (схема метрополитена), модели **поведения** (таблица с указанием количества станций метро по годам).

В зависимости от учёта фактора времени выделяют **статические** и **динамические** модели.

В статических моделях объект моделирования представляется неизменным во времени. Такие модели используются, если надо представить внешний вид объекта моделирования, его состав и структуру. Примеры статических моделей: карта местности, фотография памятника архитектуры, химическая формула вещества.

Динамические модели описывают изменения, происходящие с объектом моделирования с течением времени. Примеры динамических моделей: график изменения среднесуточной температуры в течение месяца в некоторой географической точке, биография человека.

Динамические модели могут быть **непрерывными** и **дискретными**. Непрерывные модели описывают непрерывные процессы и позволяют проследить за изменениями их свойств в любой момент времени из некоторого диапазона.

Формула перемещения при свободном падении тела $h = v_0 t + \frac{gt^2}{2}$ — пример непрерывной динамической модели.

Для решения ряда задач нет смысла в том, чтобы получать данные об изменении некоторого процесса непрерывно. Например: фиксируется среднесуточная температура; определяется среднегодовой прирост карпов в пруду; как правило, 1 раз в 10 лет проводится перепись населения. Таблицы, составленные по этим данным, — примеры дискретных динамических моделей. Дискретные динамические модели позволяют получить информацию об объекте моделирования в определённые промежутки времени.

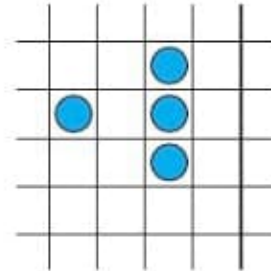


В 1970 году английский математик Джон Конвей опубликовал правила придуманной им игры «Жизнь». Действие этой игры разворачивается в мире из квадратов — размеченной на клетки поверхности. Каждая клетка на этой поверхности может находиться в одном из двух состояний: быть «живой» (заполненной) или «мёртвой» (пустой). У каждой клетки есть окружение из восьми других клеток, примыкающих к ней по горизонтали, вертикали и диагонали.

В начале игры есть некоторое распределение живых клеток по поверхности; оно называется первым поколением. Каждое следующее поколение «появляется» на основе предыдущего по приведённым ниже правилам.

Для пустых клеток:

ЕСЛИ в окружении ровно 3 живые клетки
ТО клетка рождается
ИНАЧЕ клетка остается пустой



Для заполненных клеток:

ЕСЛИ в окружении 2 или 3 живые клетки
ТО клетка остается живой
ИНАЧЕ клетка умирает

Попробуйте сыграть по этим правилам, выбрав произвольное положение для первого поколения клеток на листке бумаги и неукоснительно следуя правилам игры.

Вы без труда сможете найти множество компьютерных реализаций этой игры и понаблюдать за тем, как развивается жизнь в более крупных масштабах.

Игра «Жизнь» — пример дискретной модели, аналогии которой находят применение в математике, физике, астрономии, биологии, химии, физиологии и других областях науки.

www

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.

Модель считается адекватной объекту моделирования, если результаты, полученные в процессе её исследования, близки к результатам, наблюдаемым в экспериментах с оригиналом.

Информационные модели могут быть классифицированы по разным основаниям. По форме представления различают образные, знаковые и смешанные (образно-знаковые) информационные модели.



Вопросы и задания

1. «Модель» — слово, имеющее несколько значений. В каких смыслах оно употребляется в жизни? Приведите примеры.
2. Что такое модель в науке? 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 материальных 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. На основании чего можно утверждать, что свойства, выявленные при исследовании модели некоторого объекта, присущи и самому оригиналу?

9. Следующие модели разделите на две группы — статические и динамические. Среди динамических моделей укажите непрерывные и дискретные.

Схема метро, фотография, медицинская карта пациента, семейный фотоальбом, формула прямолинейного равноускоренного движения, уравнение химической реакции, видеозапись извержения вулкана, модель полёта артиллерийского снаряда, формула вычисления дискриминанта квадратного уравнения, таблица с данными о погоде в течение недели в некотором населённом пункте, схема персонального компьютера, ежечасные показания температуры больного с 6:00 до 21:00 в течение дня.

10. Перечислите виды информационных моделей в зависимости от формы представления информации об объекте моделирования. Приведите примеры информационных моделей каждого вида.



11. Работая в группе, приведите примеры информационных моделей, с которыми вы имели дело при изучении школьных предметов. Каждый пример оформите на отдельном слайде презентации, указав моделируемый объект (предмет, процесс, явление), цель моделирования и вид модели.

§ 2.2

Знаковые модели

Ключевые слова:

- словесные модели
- математические модели
- компьютерные модели
- этапы компьютерного моделирования

2.2.1. 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»).

2.2.2. 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

$$x = x_0 + v_0 t + \frac{a_x t^2}{2}.$$

С помощью языка алгебры логики строятся **логические модели** — формализуются (записываются в виде логических выражений) простые и составные 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. 2.4). 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 хотя бы один из переключателей.

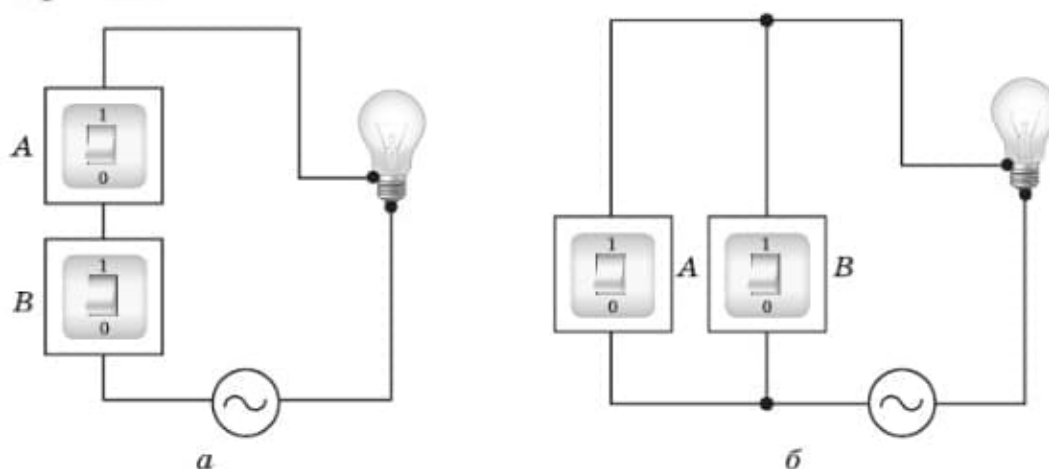


Рис. 2.4. Электрические схемы: *a* — последовательного и *б* — параллельного соединения переключателей

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:

$$\begin{aligned} F(A, B, C) &= A \& B \& (C \vee 1) \vee A \& C = A \& B \& 1 \vee A \& C = \\ &= A \& B \vee A \& C = A \& (B \vee C). \end{aligned}$$

Мы получили логическую модель, позволяющую построить схему проектируемой электрической цепи (рис. 2.5).

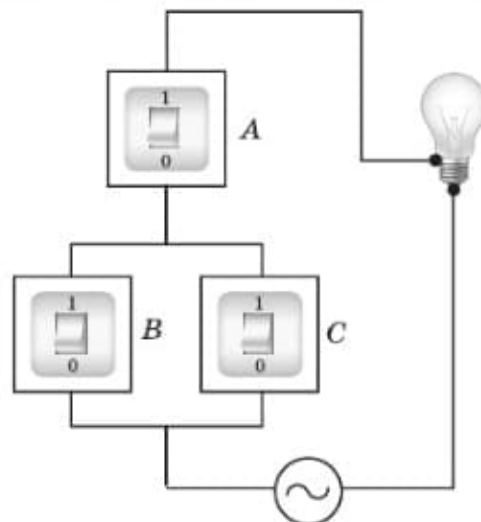


Рис. 2.5. Схема электрической цепи

2.2.3. 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;

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

- 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**.

Различают аналитические, графические и имитационные компьютерные математические модели (рис. 2.6).



Рис. 2.6. Виды компьютерных математических моделей

Аналитические модели представляют собой уравнения или системы уравнений, решение которых происходит в результате осуществления определённых последовательностей вычислительных действий. Соответствующие расчёты бывают достаточно трудоёмкими; для их выполнения требуются мощные вычислительные системы. Такие модели используют для решения многих научных и производственных задач: при составлении прогнозов погоды и предсказании глобальных климатических изменений, при разработке авиационных двигателей и новых конструкций автомобилей, в сейсморазведке, нефте- и газодобыче, для решения экологических задач и т. д.



Всем известно о разрушительной силе ядерного оружия. В настоящее время ядерные державы для совершенствования своего ядерного арсенала, как правило, используют мощнейшие суперкомпьютеры, на которых проводят расчёты, в полном объёме моделирующие ядерные взрывы и их последствия.

Существуют такие объекты, для которых в силу ряда причин аналитические модели (уравнения, неравенства) не разработаны либо их разработка невозможна. К ним относятся 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 системы. Многим

знакома ситуация, когда, придя в кассу, магазин, парикмахерскую, мы застаём там очередь. Приходится либо встать 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, но с помощью имитационных моделей можно смоделировать состояние таких систем при разных значениях этих величин.

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



Имитационное моделирование представляет собой многократный «прогон» модели с наборами случайных исходных данных, генерируемых компьютером. Те исходные данные, при которых получены самые удачные экспериментальные результаты, могут быть использованы для улучшения реального объекта. Такие наборы данных можно получить и при достаточно продолжительных наблюдениях за функционированием реальной системы, но при компьютерном моделировании это получается значительно быстрее и дешевле.

С помощью средств компьютерной графики можно визуализировать результаты расчётов, получаемых в процессе работы с компьютерными моделями. Получаемые **графические модели** позволяют «увидеть» недоступные в условиях физического эксперимента места исследуемого объекта. Например, результаты визуализированного компьютерного краш-теста автомобиля могут наглядно показать не только степень разрушения его частей, но и возможное воздействие моделируемой аварии на разные части тела и органы водителя.

2.2.4. Этапы компьютерного математического моделирования

Основными этапами компьютерного моделирования являются: постановка задачи, построение математической модели и её программная реализация, проведение компьютерного эксперимента, анализ его результатов и уточнение модели (при необходимости) (рис. 2.7).

На *первом этапе* осуществляется постановка задачи моделирования: на основании того, какие исходные данные известны и что требуется найти в решаемой задаче, определяются объект и цель моделирования; в соответствии с целью моделирования выделяются существенные признаки объекта моделирования; условие исходной задачи формализуется — создаётся её информационная (например, словесная) модель.

На *втором этапе* описательная информационная модель преобразуется в модель математическую: записываются формулы, в которых фиксируются связи между исходными данными и результатами; строится алгоритм — последовательность действий, направленная на получение результата; выбирается технология (среда программирования или прикладное программное средство) реализации алгоритма; с помощью выбранных средств создаётся компьютерная модель.

Например, компьютерная модель может быть создана в среде программирования на языке Паскаль или Python. В таком случае следует провести отладку программы — проверку её работоспособности и исправление обнаруженных при этом ошибок. Ошибки могут быть связаны с нарушением правил записи программы на конкретном языке программирования. Их программисту помогает найти используемая система программирования, выводящая на экран сообщения о выявленных ошибках.

На *третьем этапе* проводится компьютерный эксперимент, в ходе которого прежде всего проверяется адекватность компьютерной модели объекту моделирования. Делается это с помощью тестов. **Тест** — это конкретный вариант значений исходных данных, для которого известен ожидаемый результат. Тесты строятся на основании данных, полученных в процессе экспериментов с объектом-оригиналом. Если результаты компьютерного эксперимента не совпадут с результатами теста, то потребуются уточнение компьютерной модели: более полный учёт особенностей объекта моделирования, доработка математической модели, её программирование и отладка, повторное тестирование уточнённой компьютерной модели. Так продолжается до тех пор, пока результаты компьютерного эксперимента не совпадут с результатами теста. Далее проводятся необходимые вычисления, результаты которых должны дать ответы на поставленные в задаче вопросы.

На *четвёртом этапе* анализируются результаты, полученные в ходе компьютерного эксперимента, и делается вывод о том, удалось ли достигнуть цели моделирования. Если цель моделирова-

ния достигнута (получены ответы на все вопросы, интересовавшие исследователей), то принимается решение об окончании исследования. Если цель моделирования не достигнута, то принимается решение о необходимости корректировки модели и продолжении исследования.

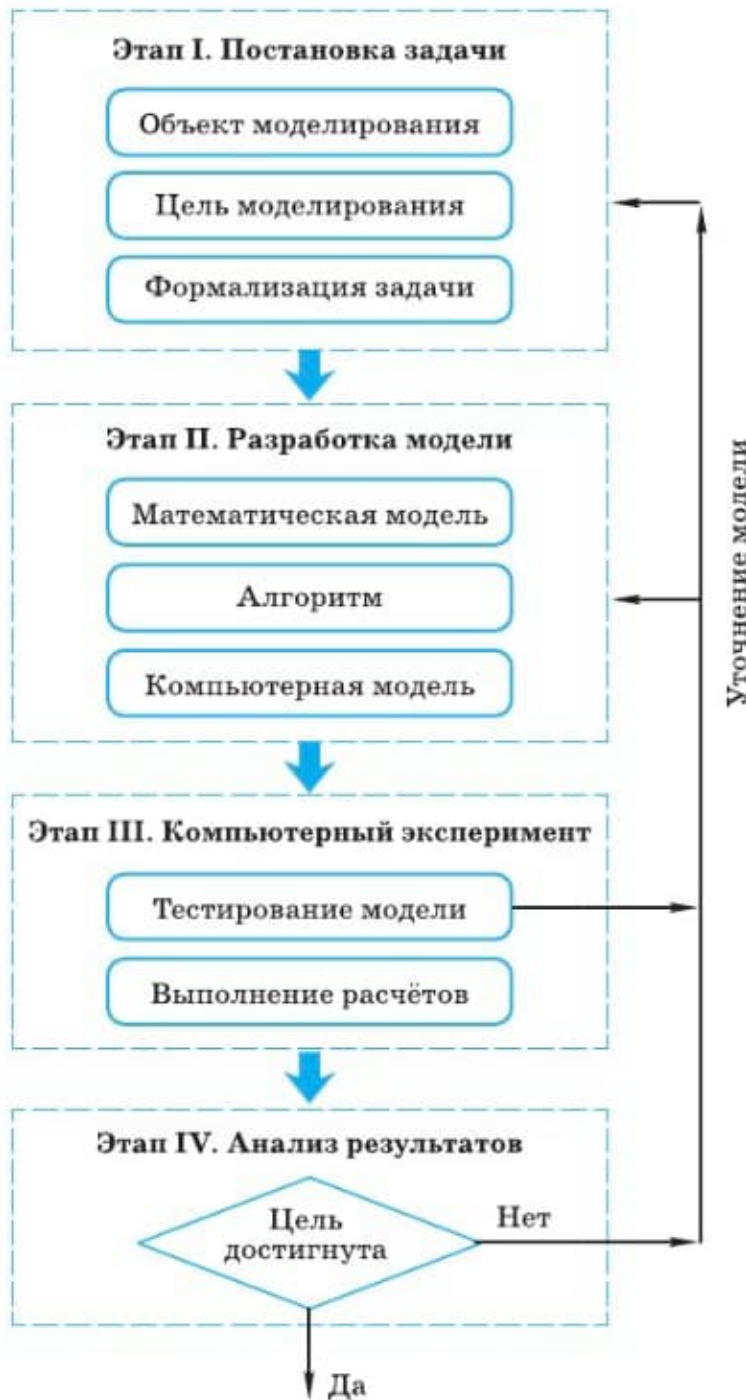


Рис. 2.7. Этапы компьютерного моделирования



Рассмотренные этапы во многом аналогичны этапам разработки современного программного обеспечения, которые при этом называются несколько иначе (рис. 2.8).



Рис. 2.8. Этапы разработки программного обеспечения

На этапе подготовки разработчик уточняет у заказчика требования к программному продукту, осуществляет предварительное планирование этапов работ, сроков, ресурсов и стоимости разработки.

На этапе проектирования составляются требования к программе, определяются её технические характеристики, выбираются алгоритмы реализации программы.

На этапе создания происходит разработка интерфейса программы (кнопки, иконки, расположение и т. д.); создаётся программный код — пишется программа, реализующая ранее выбранный алгоритм; осуществляются отладка и тестирование программы. На этом же этапе создаётся пакет документации, включающий различные описания, инструкции и руководства. Наличие подробной документации обеспечивает в том числе возможность передачи накопленных знаний другим разработчикам. На этапе поддержки происходит установка разработанного программного обеспечения (внедрение), исправление выявленных ошибок и поддержка пользователей (сопровождение).

2.2.5. Задача о пути торможения автомобиля

Рассмотрим этапы компьютерного моделирования (см. рис. 2.7) 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/с²).

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

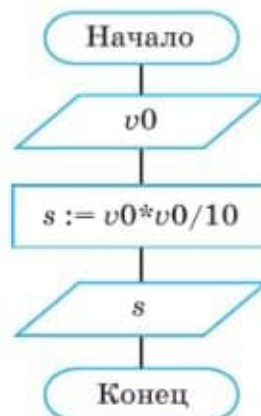
Второй этап. В данной ситуации мы имеем дело с прямолинейным равноускоренным движением тела. Формула для перемещения при этом имеет вид

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

Упростим эту формулу с учётом того, что конечная скорость должна быть равна нулю: $s_x = \frac{-v_{0x}^2}{2a_x}$.

При $a_x = -5$ м/с² получим: $s_x = \frac{v_{0x}^2}{10}$ (при условии задания скорости в метрах в секунду и вычислении пути в метрах).

Алгоритм решения задачи можно представить в виде блок-схемы:



Запишем данный алгоритм на языке программирования.

Язык программирования Паскаль

```
var v0, s: real;
begin
  writeln ('Длина пути торможения автомобиля');
  write ('Скорость движения автомобиля (км/ч)>>');
  readln (v0);
  v0 := v0 * 1000 / 60 / 60;
  s := v0 * v0 / 10;
  writeln ('s=', s)
end.
```

Язык программирования Python

```
print ('Длина пути торможения автомобиля')
v0 = float(input('Скорость движения автомобиля (км/ч)>>'))
v0 = v0 * 1000 / 60 / 60
s = v0 * v0 / 10
print ('s=', s)
```

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

Четвёртый этап. 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.



По данным компьютерного эксперимента заполните в тетради следующую таблицу:

v_0	40	50	60	70	80	90
s						

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.

Различают аналитические, графические и имитационные компьютерные математические модели.

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

Основными этапами компьютерного моделирования являются: постановка задачи, построение математической модели и её программная реализация, проведение компьютерного эксперимента, анализ его результатов и уточнение модели (при необходимости).

Вопросы и задания

1. Приведите 2–3 собственных примера словесных моделей, рассматриваемых на уроках истории, географии, биологии.
2. Вспомните басни И. А. Крылова: «Волк и Ягненок», «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? Обсудите эти вопросы в группе.
3. 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.
4. 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.
5. Решите, составив логическую модель, следующую задачу. На международных соревнованиях по прыжкам в воду первые пять мест заняли спортсмены из Германии, Италии, Китая, России и Белоруссии. Ещё до начала соревнований эксперты высказали свои предположения об их итогах.

- 1) Первое место займёт спортсмен из Китая, а спортсмен из Белоруссии будет третьим.
- 2) Белоруссия 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?

6. 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?
7. Каковы основные преимущества использования компьютерных моделей перед натурными моделями? Приведите примеры использования компьютерных моделей. Найдите соответствующую информацию в сети Интернет.
8. Перечислите основные этапы компьютерного моделирования. Какой этап решения задачи на компьютере, по вашему мнению, является наиболее трудоёмким?
9. Что происходит на этапе постановки задачи? Что является результатом этого этапа?
10. Что происходит на этапе разработки модели? Что является результатом этого этапа?
11. Что происходит на этапе компьютерного эксперимента? Что является результатом этого этапа?
12. Как вы считаете, по силам ли одному специалисту реализация всех этапов компьютерного моделирования для решения сложной практической задачи? Обсудите этот вопрос в группе.
13. Может ли пригодиться в жизни представление об этапах компьютерного моделирования? Обоснуйте свою точку зрения.

14. 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. Требуется вычислить сумму денег, полученную авиакомпанией от продажи билетов на этот рейс, если известно, что остались нераспроданными a билетов бизнес-класса и b билетов экономического класса. Разработайте компьютерную модель для решения этой задачи. Осуществите компьютерный эксперимент.
15. Уличный продавец газет получает a рублей с продажи каждой из первых 50 газет. С продажи каждой из последующих газет он получает на 20% больше. Требуется вычислить заработок продавца, если он продаст за день n газет. Разработайте компьютерную модель для решения этой задачи. Осуществите компьютерный эксперимент.
16. Проведите аналогию между этапами компьютерного моделирования и этапами разработки программного обеспечения. Обсудите этот вопрос в группе.



§ 2.3

Графические информационные модели

Ключевые слова:

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

2.3.1. 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. Схемами 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. 2.9.

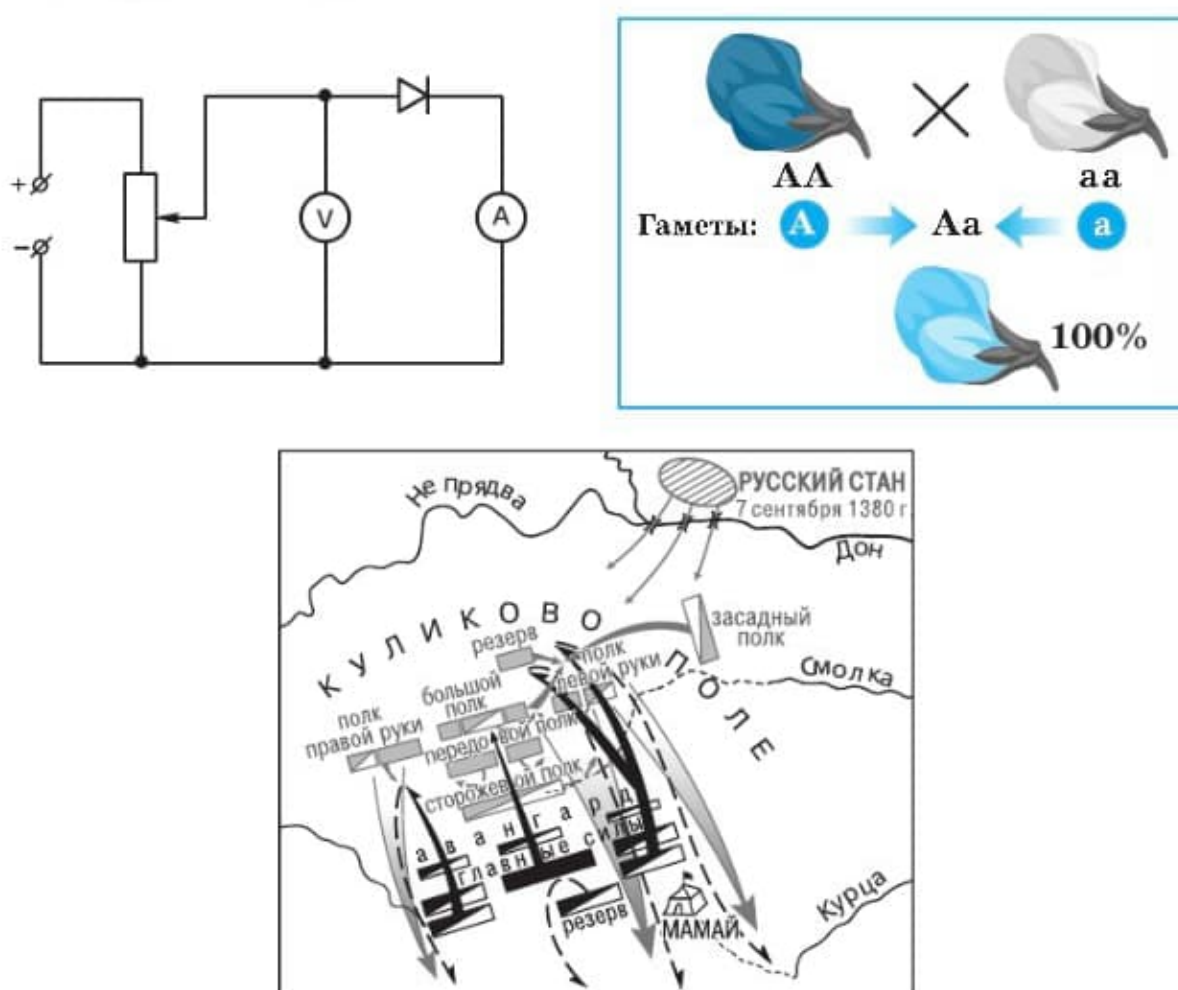


Рис. 2.9. Примеры схем, используемых на уроках физики, биологии, истории

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.

2.3.2. Графы

Если некоторые объекты изобразить **вершинами**, а связи между ними — **линиями**, то мы получим информационную модель в форме **графа**. Вершины графа могут изображаться кругами, овалами, точками, прямоугольниками и т. д. Ненаправленная (без стрелки) линия, соединяющая вершины графа, называется **ребром**. Линия направленная (со стрелкой) называется **дугой**; при этом вершина, из которой дуга исходит, называется **начальной**, а вершина, куда дуга входит, — **конечной**.

Граф называется **неориентированным**, если его вершины соединены рёбрами (рис. 2.10, *а*). Вершины **ориентированного** графа соединены дугами (рис. 2.10, *б*).

Путь — это последовательность рёбер (дуг), по которым можно перейти из одной вершины в другую.

Граф называется **связным**, если от любой его вершины можно по рёбрам перейти к любой другой вершине.

Граф называется **взвешенным**, если его вершины или рёбра характеризуются некоторой дополнительной информацией — **весами** вершин или рёбер. На рис. 2.10, *в* с помощью взвешенного неориентированного графа изображены дороги между пятью населёнными пунктами *A, B, C, D, E*; веса рёбер — протяжённость дорог в километрах.

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**. Направленный (ориентированный) **ациклический** граф — ориентированный граф, не имеющий циклов.

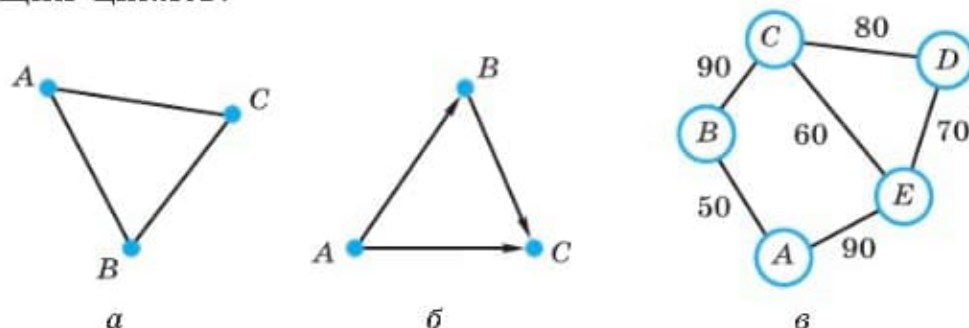


Рис. 2.10. Графы

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

Всегда ли семантическая сеть является сетью — графом с циклом?

Река Преголя в центре старинного Кёнигсберга (ныне Калининград) разделяется на два русла. В некоторых местах эти русла соединены протоками. Благодаря одной такой протоке центр города оказался разбит на четыре части, которые со временем соединили мостами. Известна старинная математическая задача, в которой спрашивалось: можно ли пройти по всем семи мостам центра старого Кёнигсберга, не проходя ни по одному из них дважды? Перед вами схема центра Кёнигсберга (рис. 2.11). Можете попробовать решить эту задачу.

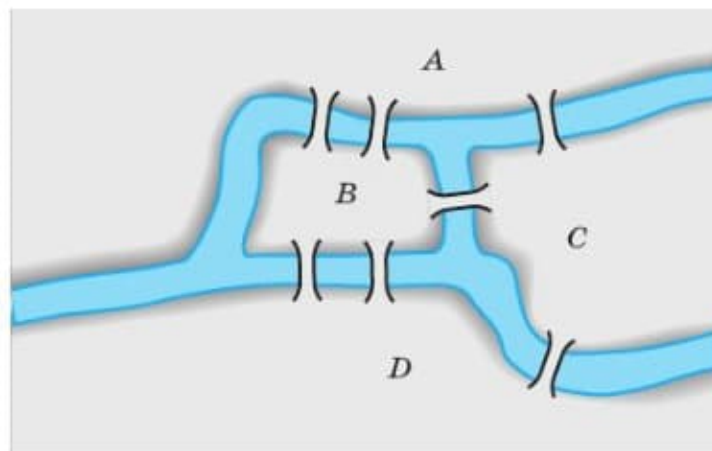


Рис. 2.11. Мосты Кёнигсберга

Впервые эта задача была решена в 1736 году математиком Леонардом Эйлером, обозначившим части города точками, а соединяющие их мосты — линиями (рис. 2.12). Благодаря этому исходная задача свелась к следующей: можно ли обойти данный граф, не отрывая карандаш от бумаги, так, чтобы каждое его ребро было пройдено ровно один раз?

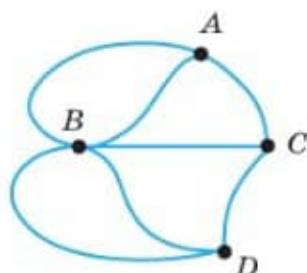


Рис. 2.12. Граф мостов Кёнигсберга

Исследовав множество аналогичных графов, Эйлер установил:

- если у графа все вершины чётные, т. е. из каждой вершины выходит чётное число рёбер, то такой граф можно обойти, не отрывая карандаш от бумаги; начинать можно в любой вершине;
- если у графа две нечётные вершины, то его можно обойти, не отрывая карандаш от бумаги; начинать надо в одной из нечётных вершин, а закончить — в другой;
- граф с более чем двумя нечётными вершинами обойти, не отрывая карандаш от бумаги, невозможно.

1736 год считается годом рождения теории графов, а Леонард Эйлер — её основоположником.

Определив чётность вершин A , B , C и D , вы можете самостоятельно завершить решение задачи о мостах Кёнигсберга.



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.

Пример 1

На рис. 2.13 изображена схема дорог, связывающих торговые точки A , B , C , D , E . По каждой дороге можно двигаться только в направлении, указанном стрелкой. Сколько существует различных путей от точки A до точки E ?

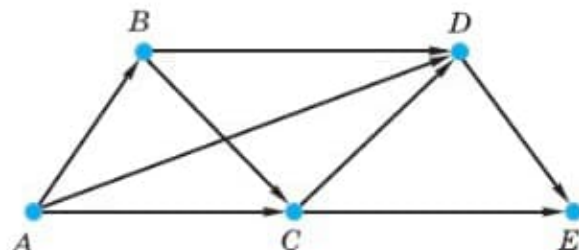


Рис. 2.13. Схема дорог, представленная ориентированным графом

Граф, представленный на рис. 2.13, — направленный ациклический граф. Обратите внимание на вершину A : это единственная вершина графа, в которую не входит ни одна дуга, а все дуги из неё выходят. A — начальная вершина (источник) рассматриваемого графа. Что касается вершины E , то есть входящие в неё дуги, но нет ни одной дуги, выходящей из неё. E — конечная вершина (сток) рассматриваемого графа.

Предложенная вам задача — пример задачи на вычисление количества путей в направленном ациклическом графе.

Решение

В вершину E можно попасть только из вершин C и D . Если мы будем знать число путей из вершины A в вершину C и из вершины A в вершину D , то, сложив их, получим искомое число путей из A в E . Действительно, для того чтобы попасть из вершины A в вершину E , мы просто все пути из вершины A в вершину C дополним дугой CE , а пути из вершины A в вершину D дополним дугой DE . Число путей при этом не изменится. Итак, число путей из вершины A в вершину E равно сумме путей из A в C и из A в D .

Можно сказать, что наша задача распалась на две более простые задачи. Решим каждую из них в отдельности.

В вершину C можно попасть непосредственно из вершины A и из вершины B . В свою очередь, существует единственный путь из вершины A в вершину B . Таким образом, из вершины A в вершину C можно попасть двумя путями: 1 (напрямую из A) $+ 1$ (через B) $= 2$.

Попробуйте доказать, что путь из вершины A в вершину B — единственный.



Что касается вершины D , она является конечной вершиной для трёх дуг: BD , AD и CD . Следовательно, в неё можно попасть из вершин A , B и C :

1 (напрямую из A) + 1 (через B) + 2 (через C) = 4. Итак, существуют четыре пути из вершины A в вершину D .

Теперь выполним подсчёт путей из A в E :

2 (через C) + 4 (через D) = 6.

Решение задачи будет гораздо проще, если двигаться от вершины A (начало маршрута) к вершине E и проставлять веса вершин — число путей из A в текущую вершину (рис. 2.14). При этом вес вершины A можно принять за 1. Действительно, существует единственный способ попасть из A в A — оставаться на месте.

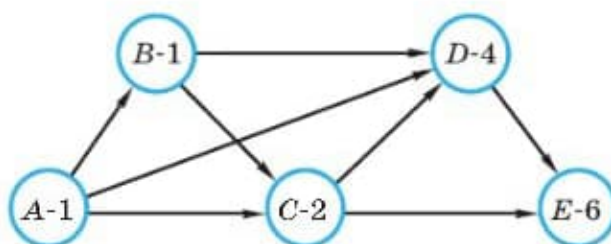


Рис. 2.14. Схема дорог, представленная взвешенным ориентированным графом

С помощью графов можно решать и разные занимательные задачи.

Пример 2

Рассмотрим несколько видоизменённую классическую задачу о переправе.

На берегу реки стоит крестьянин (K) с лодкой, а рядом с ним — собака (C), лиса (L) и гусь (G). 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 гусём 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 и результирующее размещение персонажей



на берегах 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. 2.15).

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.

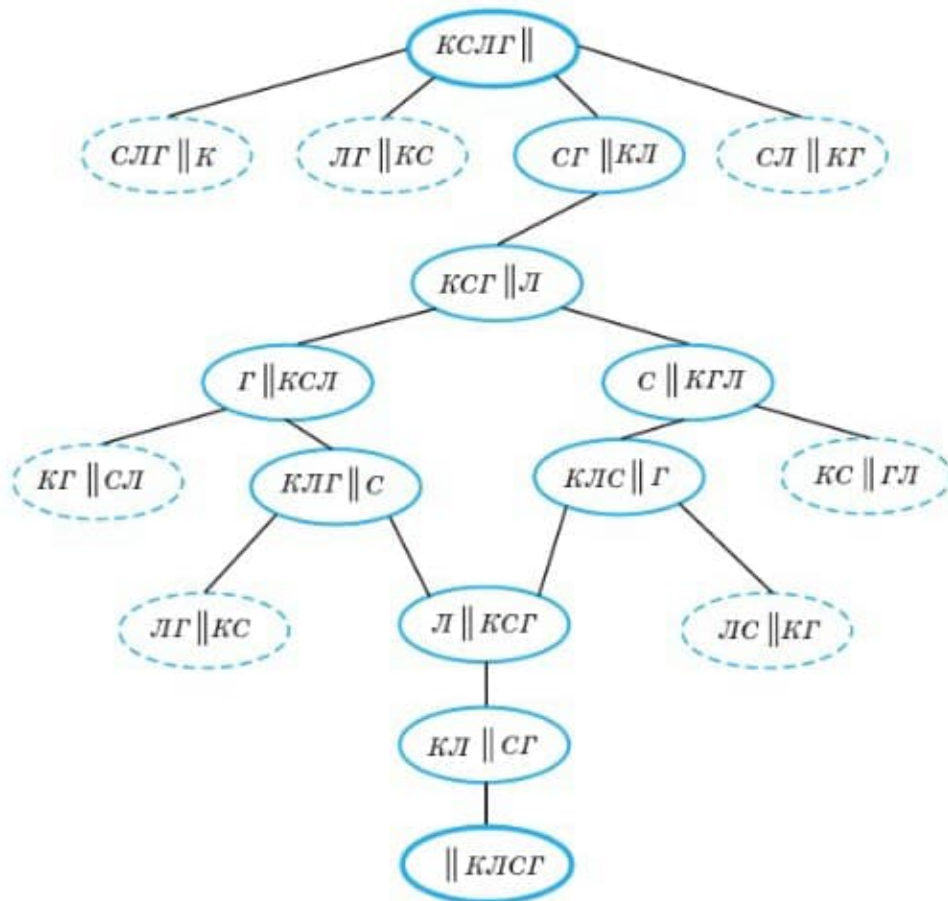


Рис. 2.15. Граф переправы

2.3.3. Деревья

Дерево — это связный 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.

У дерева выделяется одна главная вершина, называемая его **корнем** (вершина *A* на рис. 2.16). Каждая вершина (кроме корня) имеет единственного **предка**; корень предка не имеет. Для вершины *E* предком является вершина *B*; эта же вершина *B* является предком для вершины *D*. Любая вершина дерева может породить несколько¹ **потомков**. Так, потомками вершины *D* являются вершины *H*, *I*, *J*; потомками вершины *C* — вершины *F*, *G*. Вершины, не имеющие порождённых вершин (потомков), называют **листьями** (вершины *M*, *N*).

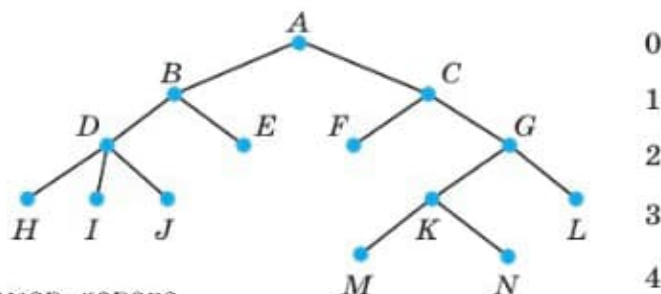


Рис. 2.16. Пример дерева

Любая вершина дерева вместе со всеми своими вершинами-потомками является **поддеревом**; в частности, любой лист образует одноэлементное поддерево. **Уровень вершины** дерева — длина пути от корня до вершины; уровень корня равен нулю. **Высота дерева** — максимальный уровень вершин, образующих дерево (наибольшая длина пути от корня к листу).

Внимательно рассмотрите дерево, представленное на рис. 2.16. Сколько у этого дерева вершин? Какова высота этого дерева? Приведите пример поддерева в этом дереве. Приведите примеры вершин 1-го, 2-го, 3-го и 4-го уровня.

Всякая иерархическая система может быть представлена с помощью дерева.

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

¹ Такой тип связи называется «один — ко многим».



Пример 3

Для того чтобы записать все трёхзначные числа, состоящие из цифр 1 и 2, можно построить **дерево вариантов** (рис. 2.17).

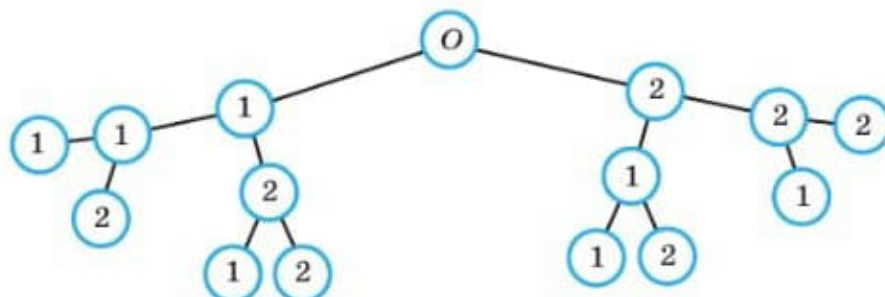


Рис. 2.17. Дерево вариантов для решения задачи о записи трёхзначных чисел

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 4

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 игрока I останутся 3 спички, игрок II может выиграть, взяв две спички и оставив одну.

Если после хода 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 выиграет первый игрок. Для этого своим первым ходом он должен взять одну спичку.

На рис. 2.18 представлен граф, называемый деревом игры; на нём отражены все возможные варианты, в том числе ошибочные (проигрышные) ходы игроков.

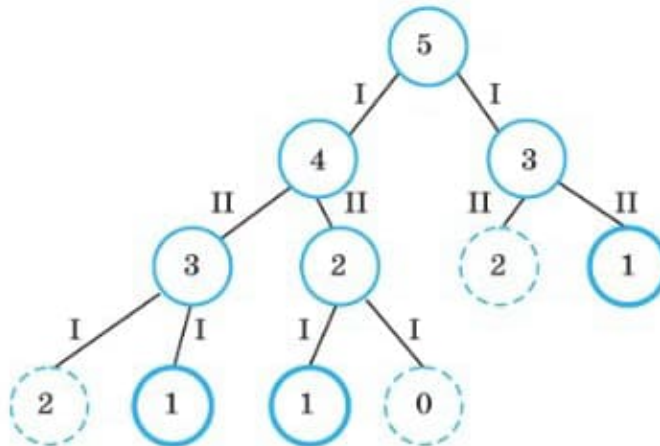


Рис. 2.18. Дерево игры

Дерево игры — один из вариантов **игровых моделей**, используемых при моделировании действий противоборствующих сторон в ходе соревнований, конкуренции в бизнесе и даже военных действий. Игровые модели позволяют разработать план действий, обеспечивающий самый лучший результат за счёт учёта возможных действий противника.

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.

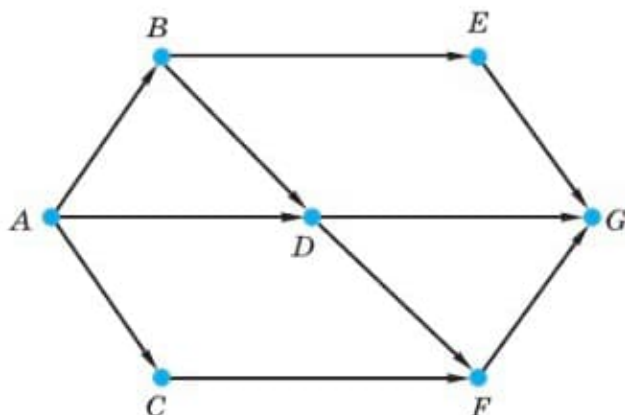
Граф состоит из вершин, связанных линиями — рёбрами или дугами. Граф называется взвешенным, если его вершины или рёбра (дуги) характеризуются некоторой дополнительной информацией — весами вершин (рёбер, дуг).

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



Вопросы и задания

1. Какие информационные модели относят к графическим?
2. Приведите примеры графических информационных моделей, с которыми вы имеете дело:
 - а) при изучении других предметов;
 - б) в повседневной жизни.
3. Что такое граф? Что является вершинами и рёбрами графа на рис. 2.10, в? 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.
4. Приведите пример системы, модель которой можно представить в форме графа. Изобразите соответствующий граф.
5. 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, 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.
6. На рисунке изображена схема дорог, связывающих торговые точки A , B , C , D , E , F , G . По каждой дороге можно двигаться только в направлении, указанном стрелкой. Сколько существует различных путей от точки A до точки G ?





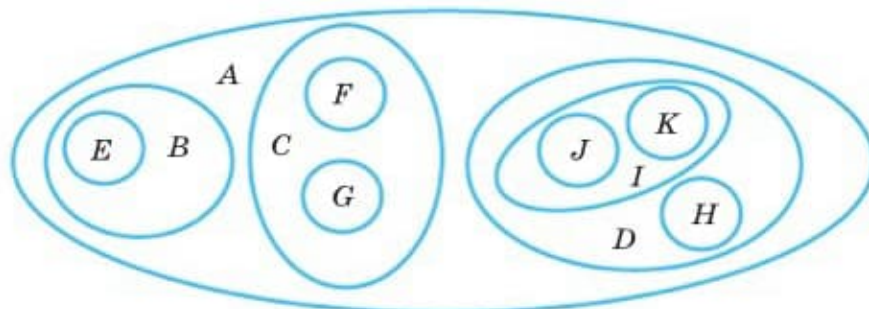
7. Работая в группе, составьте семантическую сеть по одной из русских народных сказок: «Колобок», «Курочка Ряба», «Репка».
8. Что такое дерево? Моделями каких систем могут служить деревья? Приведите пример такой системы.
9. Сколько трёхзначных чисел можно записать с помощью цифр 2, 4, 6 и 8 при условии, 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. У исполнителя Вычислитель две команды, которым присвоены номера:
 - 1 — прибавь 1;
 - 2 — умножь на 3.

Выполняя первую из них, Вычислитель прибавляет к числу на экране 1, а выполняя вторую, утраивает это число.

Какое максимальное количество разных программ, состоящих из пяти команд, можно составить для этого исполнителя?

Пусть 0 — начальное значение. Какие числа будут получены в результате выполнения всех программ для исполнителя Вычислитель, состоящих не более чем из четырёх команд? Решение оформите в виде дерева, корневой вершиной которого является начальное значение (число 0), а рёбрам соответствуют команды исполнителя; в вершинах дерева запишите результаты выполнения соответствующих программ. Какое наибольшее число будет записано в вершинах третьего уровня?

13. Представьте отношения между множествами в форме дерева.



Сколько у этого дерева вершин? Какова высота этого дерева? Приведите пример поддеревя в этом дереве. Приведите примеры листьев. Вершинами какого уровня они являются?

14. 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, построив дерево игры.

§ 2.4

Табличные информационные модели

Ключевые слова:

- таблица
- таблица «объект — объект»
- таблица «объект — свойство»
- весовая матрица

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

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

Информация в таблице наглядна, компактна и легкообозрима.

2.4.1. 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 ниже (табл. 2.1).

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 2.1

Регионы Российской Федерации

Название	Дата образования (ДД.ММ.ГГ)	Площадь (тыс. км ²)	Население ¹ (тыс. чел.)
Астраханская область	27.12.1943	44,1	1010,0
Белгородская область	06.01.1954	27,1	1532,5
Владимирская область	14.08.1944	29,0	1443,7
Вологодская область	23.09.1937	145,7	1202,4
Воронежская область	13.06.1934	52,4	2335,4
Калужская область	05.07.1947	29,9	1010,9

Располагаете ли вы аналогичной информацией о своём регионе? Уточните информацию о современной численности населения вашего региона.



В таблицах типа «**объект — объект**» отражается взаимосвязь между объектами одного или нескольких классов. Например, в школьных электронных журналах есть таблица «Успеваемость»; ниже представлен её фрагмент (табл. 2.2).

¹ Численность населения приведена по результатам переписи населения России 2010 г.

Таблица 2.2

Успеваемость

Ученик	Предмет				
	Алгебра	Биология	География	Геометрия	Информатика
Акуленко Иван	5	5	5	5	5
Баранова Мила	4	4	4	4	4
Варнаков Руслан	4	5	5	4	5
Егоров Тимур	4	5	4	3	5
Машкова Карина	3	4	4	3	4

В этой таблице отражена связь «успеваемость» между объектами класса¹ «Учащиеся» и объектами класса «Предмет».

В таблице «Расстояния между городами» (табл. 2.3) представлены расстояния между парами объектов, принадлежащих одному классу «Город».

Таблица 2.3

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	



Создайте эту таблицу в текстовом процессоре; добавьте в неё строку и столбец, куда занесите информацию о городе — административном центре своего региона.

В форме таблицы «объект — объект» можно представить информацию о наличии границ (сухопутной, морской, озёрной, речной) России с другими странами; её фрагмент представлен ниже (табл. 2.4).

¹ Класс — множество объектов, обладающих общими признаками.

Таблица 2.4

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 (наличие/отсутствие связи между объектами) свойства.

2.4.2. Использование таблиц при решении задач

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

Пример 1

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.

Решение

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 из первой и второй таблиц 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.



Пример 2

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?



Ранее мы рассмотрели способ записи решения аналогичных задач с помощью дерева. Сейчас оформим решение в виде таблицы (табл. 2.5).

Таблица 2.5

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✓				
		9, 6, 15				
		10, 2, 12				
		9, 3, 12				
	3, 6, 9	9, 6, 15				
		3, 18, 21✓				
		4, 6, 10				
		3, 7, 10				
	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, 30✓		
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 все возможные варианты ходов второго игрока, выигрышные ходы отмечены галочкой. При безошибочной игре первого игрока такие ситуации возникнуть не должны. Другими словами, первый игрок, чтобы не допустить выигрыша своего соперника, не должен создавать ситуации (9, 2, 11) и (3, 6, 9). В этом же столбце расписаны все возможные ходы второго игрока из положений (4, 2, 6) и (3, 3, 6); здесь у второго игрока нет выигрышного хода. Если получены одинаковые варианты, то все из них, кроме одного, исключаем из дальнейшего рассмотрения.

В четвёртом столбце рассмотрены 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 (4, 3, 7), 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.

Попытайтесь самостоятельно изобразить выигрышную стратегию второго игрока в виде графа.



Пример 3

Человеку удобно работать с информацией, представленной наглядно. Например, взглянув на взвешенный граф, представляющий схему дорог, соединяющих населённые пункты *A*, *B*, *C*, *D*, *E* (рис. 2.19), вы сразу же можете сказать, между какими населёнными пунктами есть дороги и какова их протяжённость. Для того чтобы эти же данные мог обработать компьютер, они должны быть представлены в табличной форме.



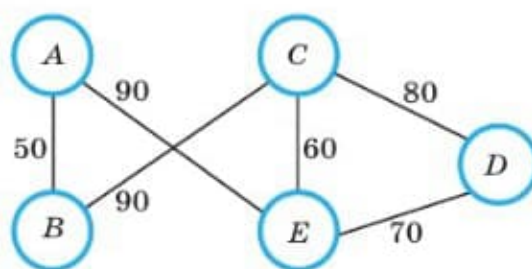


Рис. 2.19. Граф, представляющий схему дорог

Построим таблицу, соответствующую графу, представленному на рисунке 2.19.

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** (рис. 2.20).

	<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. 2.20. Весовая матрица



Этой же матрице соответствует ещё один граф, представленный на страницах учебника. Найдите его.

Для решения некоторых задач бывает удобно по имеющейся таблице (весовой матрице) строить граф. При этом одной и той же таблице могут соответствовать графы, внешне не похожие друг на друга.

Пример 4

Между населёнными пунктами *A*, *B*, *C*, *D*, *E* построены дороги, протяжённость которых (в километрах) приведена в таблице.

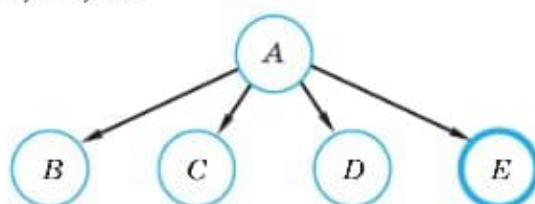
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>		1	4	3	7
<i>B</i>	1		2	5	
<i>C</i>	4	2		3	
<i>D</i>	3	5	3		2
<i>E</i>	7			2	



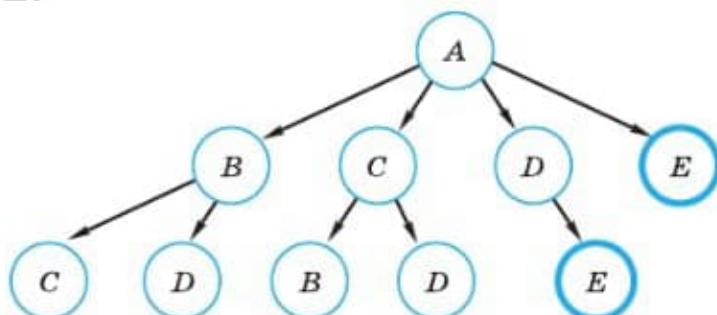
Требуется определить длину кратчайшего пути между пунктами *A* и *E*. Передвигаться можно только по дорогам, протяжённость которых указана в таблице. Каждый пункт можно посетить только один раз.

Решение

Для решения этой задачи построим дерево по информации, представленной в табличной форме. В нашем случае *A* — это корень дерева. Из таблицы видно, что вершину *A* надо соединить с вершинами *B*, *C*, *D*, *E*.

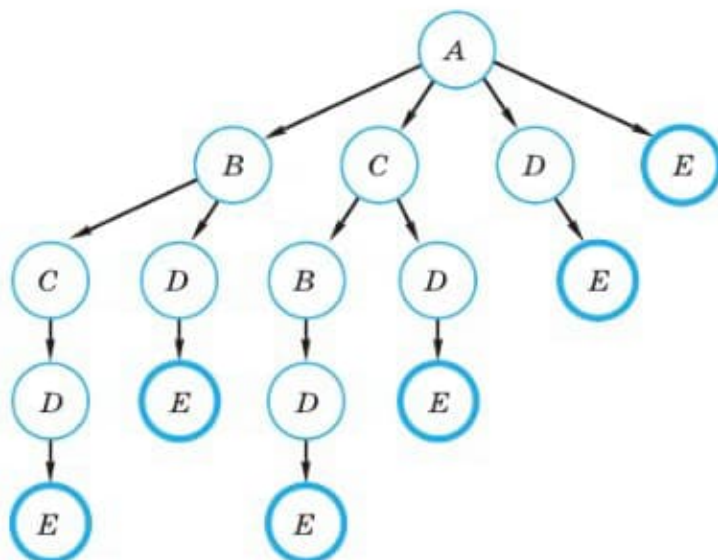


Вершину *B* соединим с вершинами *C*, *D* (с вершиной *A* она уже соединена); вершину *C* — с вершинами *B*, *D*; вершину *D* — с вершиной *E*.



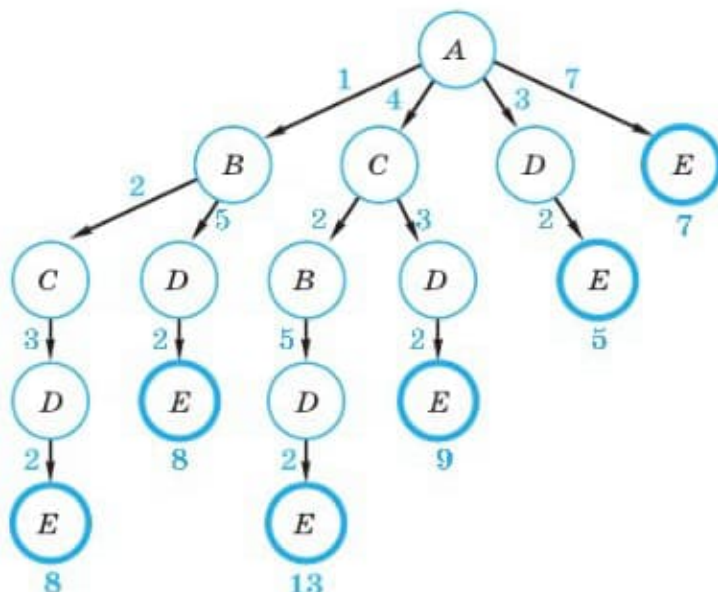
Вершину *D* можно было бы соединить с вершинами *B* и *C*. Но мы не стали этого делать, исходя из информации в последней строке таблицы: так как в вершину *E* можно попасть или напрямую из *A*, или через вершину *D*, то переходить из *D* в вершину, отличную от *E*, не имеет смысла.

Продолжим построение дерева, принимая во внимание то, что ни одну из вершин нельзя посещать дважды.



Мы получили шесть путей из A в E . Остается вычислить длину каждого пути, чтобы выяснить, какой из них минимальный.

Выбирая информацию из таблицы, расставим веса соответствующих дуг.



Сложив веса соответствующих дуг, получим длину каждого из шести путей. Видно, что минимальная длина пути из пункта A в пункт E равна пяти.



Найдите длину кратчайшего пути между пунктами A и E , проходящего через пункт C .

САМОЕ ГЛАВНОЕ

В табличных информационных моделях информация об объекте представляется в виде прямоугольной таблицы, состоящей из строк и столбцов. 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. Какие преимущества обеспечивают табличные информационные модели по сравнению со словесными описаниями? Приведите пример.
2. Приведите примеры табличных информационных моделей, с которыми вы имеете дело:
 - а) на уроках в школе;
 - б) в повседневной жизни.
3. 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.
4. 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?

5. Два игрока играют в следующую игру. Перед ними лежат две кучи 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.
6. 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 перевозок из A в B ?

1)

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

§ 2.5

База данных как модель предметной области

Ключевые слова:

- информационная система
- база данных
- реляционная база данных
- запись
- поле
- ключ
- СУБД
- таблица
- форма
- запрос
- условие выбора
- отчёт

2.5.1. 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 той или иной предметной области.

2.5.2. Реляционные базы данных

Данные в базах данных могут быть организованы разными способами, один из них основан на представлении данных в виде



таблиц. Базы данных, использующие такой способ представления данных, называются **реляционными базами данных (РБД)**.

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

Строка таблицы РБД называется **записью**, столбец — **полем** (рис. 2.21).

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

Запись
Поле

Рис. 2.21. 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, такой как: название товара; стоимость товара; количество имеющихся в наличии товаров; название книги; автор книги; год издания; фамилия, имя, отчество сотрудника; дата рождения; специальность 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.

2.5.3. Система управления базами данных

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.

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 Base*. При запуске любой из них на экран выводится окно, имеющее строку заголовка, строку меню, панели инструментов, рабочую область и строку состояния (рис. 2.22).

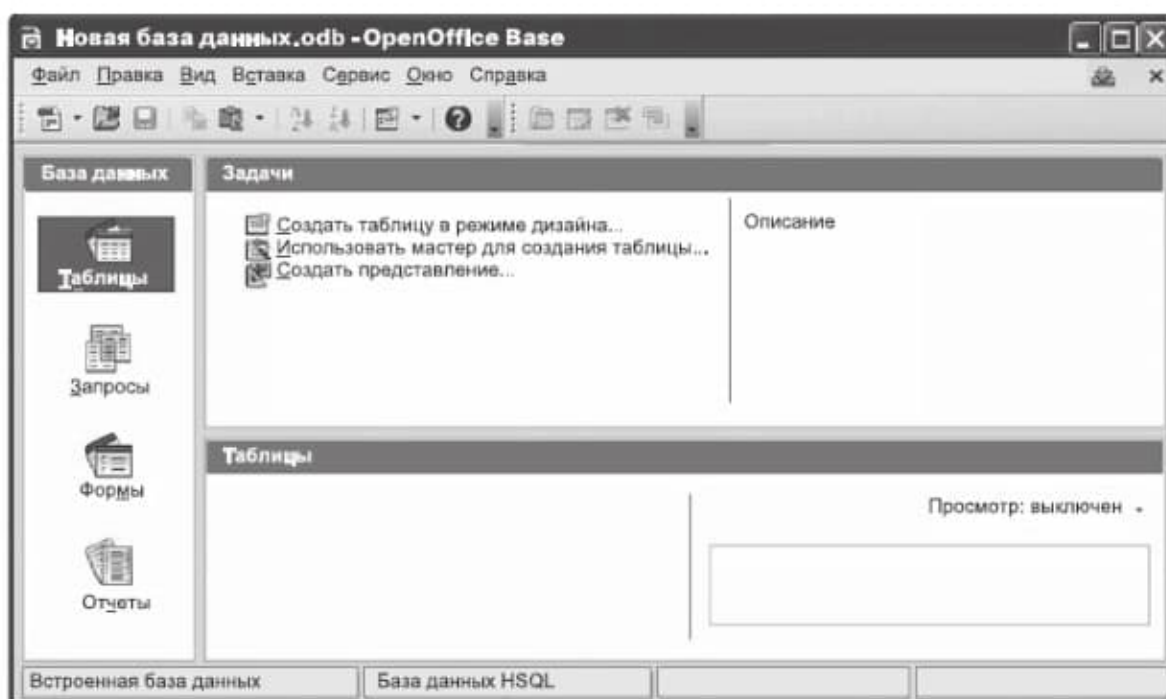


Рис. 2.22. Среда *OpenOffice Base*

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 и сортировки данных.

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

2.5.4. 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. 2.23).

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. 2.23. 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 СУБД 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. 2.24), а можно создать для этого специальный шаблон — форму (рис. 2.25).

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

Рис. 2.24. Таблица для ввода данных

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

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

Рис. 2.25. Формы для ввода данных

После выполнения всех перечисленных выше действий будет получен следующий результат (рис. 2.26).

Код	Фамилия	Имя	Дата рождения	Пол	Рост	Адрес	Увлечение	Наличие ПК
1	Гриднев	Михаил	23.05.07	м	152,0	Первомайская 16-8	футбол	<input checked="" type="checkbox"/>
2	Дементьева	Анастасия	08.04.07	ж	154,0	Школьная 5-2	танцы	<input checked="" type="checkbox"/>
3	Жихорева	Алексей	12.12.07	м	160,0	Садовая 10-14	футбол	<input checked="" type="checkbox"/>
4	Кочергина	Ольга	01.11.06	ж	164,0	Садовая 10-56	плавание	<input type="checkbox"/>
5	Новиков	Михаил	12.08.07	м	158,0	Школьная 12-24	футбол	<input type="checkbox"/>
6	Патрина	Ирина	09.05.07	ж	154,0	Первомайская 2-10	музыка	<input checked="" type="checkbox"/>
7	Патрина	Ольга	09.05.07	ж	156,0	Первомайская 2-10	танцы	<input checked="" type="checkbox"/>
8	Торопчин	Сергей	14.04.07	м	162,0	Первомайская 12-3	плавание	<input checked="" type="checkbox"/>
9	Шашков	Иван	13.11.06	м	156,0	Школьная 4-4	танцы	<input type="checkbox"/>
10	Юсуфова	Диана	01.09.06	ж	168,0	Школьная 3-15	музыка	<input checked="" type="checkbox"/>

Рис. 2.26. 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).

2.5.5. Запросы на выборку данных

После того как база данных будет 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 указаны 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. 2.6).

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; с их помощью вы записывали условия при программировании разветвляющихся алгоритмов.

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.

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 и он «меньше» любой буквы.

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

01.11.07<02.11.07;

29.11.07<02.12.08;

29.11.07<01.11.08.

Таблица 2.6

Условия выбора — составные 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'	1	Истина
			2	Ложь
5	Ученик родился в 2008 году или позже	ДАТА>#31.12.07#	8	Истина
			10	Ложь
6	Ученик имеет персональный компьютер	НАЛИЧИЕ ПК=1	7	Истина
			9	Ложь

Условия выбора могут задаваться не только простыми, но и составными логическими выражениями, содержащими логические операции (табл. 2.7). С основными логическими операциями И, ИЛИ, НЕ вы познакомились в 8 классе.

Таблица 2.7

Условия выбора — составные 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 09.05.08	ИМЯ='Ольга' И ДАТА<>#09.05.08#	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

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.

Реляционные базы данных (РБД) основаны 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.

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. Что такое информационная система? Приведите пример информационной системы, которой пользовались вы или члены вашей семьи.



2. 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? Обсудите эти вопросы в группе.
3. Какие базы данных называются реляционными?
4. Что такое запись? Какую информацию она содержит?
5. Что такое поле? Какую информацию оно содержит?
6. Перечислите основные типы полей РБД.
7. Для однотабличной базы данных
КОЛЛЕКЦИЯ (КОД, НАЗВАНИЕ ЭКСПОНАТА, АВТОР,
МЕСТО ИЗГОТОВЛЕНИЯ, ГОД ИЗГОТОВЛЕНИЯ,
ФИО ПРЕДЫДУЩЕГО ВЛАДЕЛЬЦА, ДАТА ПРИОБРЕТЕНИЯ,
СТОИМОСТЬ ЭКСПОНАТА, УПОМИНАНИЕ В КАТАЛОГАХ
(да/нет))

укажите тип каждого поля.
8. Что такое ключ таблицы базы данных? Что может служить ключом в базе данных КОЛЛЕКЦИЯ из задания 7?
9. Продумайте состав, типы полей и ключ однотабличной базы данных:
 - а) ТУРАГЕНТСТВО;
 - б) ВИДЕОТЕКА;
 - в) АВТОСАЛОН;
 - г) РЕГИОНЫ РФ.
10. Что такое СУБД?
11. Какая СУБД установлена на компьютерах в вашем классе?
12. С чего начинается создание БД?
13. Перечислите основные объекты СУБД. Какие функции они выполняют?

14. Ниже в табличной форме представлены характеристики ноутбуков, имеющих в продаже в компьютерном салоне.

Производитель	Накопитель (Гб)	Оперативная память (Гб)
ASUS	128	4
HP	256	8
Acer	512	4
Lenovo	512	8
HUAWEI	512	16
Apple	4096	16

- а) Какой по счёту будет запись, содержащая сведения о ноутбуке ASUS, после сортировки данных по возрастанию значений поля ПРОИЗВОДИТЕЛЬ?
- б) Какой по счёту будет запись, содержащая сведения о ноутбуке Acer, после сортировки данных по возрастанию значений поля НАКОПИТЕЛЬ?
- в) Какой по счёту будет запись, содержащая сведения о ноутбуке Apple, после сортировки данных сначала по возрастанию значений поля ПРОИЗВОДИТЕЛЬ, затем по возрастанию значений поля ОПЕРАТИВНАЯ ПАМЯТЬ?
15. Как будет выглядеть список (фамилия, имя) учеников после сортировки данных по возрастанию значений поля ДАТА РОЖДЕНИЯ базы данных «Наш класс» (см. рис. 2.26)?
16. Укажите все записи базы данных «Наш класс» (см. рис. 2.26), для которых будут истинными простые логические выражения 1–6 из табл. 2.6.
17. Укажите все записи базы данных «Наш класс» (рис. 2.26), для которых будут истинными составные логические выражения 1–3 из табл. 2.7.
18. Какова цель запроса на выборку?

19. 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)

20. 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

21. База данных «Спортивная секция» наряду с другими полями имеет поля с названиями ПОЛ и КЛАСС. В базе данных находятся записи о спортсменах трёх классов: '9 кл.' — девятиклассники, '10 кл.' — десятиклассники, '11 кл.' — одиннадцатиклассники. Количество записей N , удовлетворяющих различным запросам, приведено в таблице.

Запрос	N
ПОЛ='м' ИЛИ КЛАСС='11 кл.' ИЛИ КЛАСС='9 кл.'	55
КЛАСС='10 кл.'	20
ПОЛ<>'ж' И КЛАСС='10 кл.'	15

Чему равно количество записей в базе данных?

Тестовые задания для самоконтроля

1. Выберите истинное высказывание.
 - 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 ложное высказывание.
 - 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) формализация
 - в) моделирование
 - 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

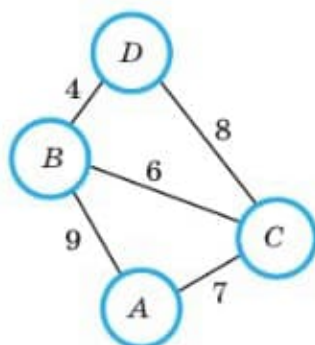
8. Выберите смешанную модель:
 - а) фотография
 - б) схема
 - в) текст
 - г) формула

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 *A*, *B*, *C*, *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.

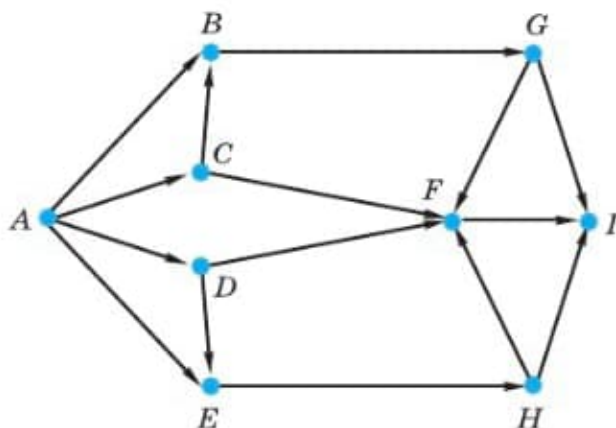
- а) $ABCD$
 б) $ACBD$
 в) $ADCB$
 г) $ABDC$
17. Между населёнными пунктами A , B , C , D , E , F построены дороги, протяжённость которых (в километрах) приведена в таблице.

	A	B	C	D	E	F
A		9	10		3	15
B	9		2			8
C	10	2		3		
D			3		3	2
E	3			3		
F	15	2		2		

Определите длину кратчайшего пути между пунктами B и E . Передвигаться можно только по дорогам, указанным в таблице.

- а) 8
 б) 12
 в) 13
 г) 15

18. На рисунке представлена схема дорог, связывающих города $A, B, C, D, E, F, G, H, I$. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города A в город I ?



- а) 3
 б) 4
 в) 10
 г) 12
19. В школе учатся четыре ученика — Андреев, Иванов, Петров, Сидоров, имеющие разные увлечения. Один из них увлекается теннисом, другой — бальными танцами, третий — живописью, четвертый — 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
20. 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

21. Строка таблицы реляционной базы данных, содержащая информацию об одном конкретном объекте, — это:
- поле
 - запись
 - отчёт
 - форма
22. Столбец таблицы реляционной базы данных, содержащий определённую характеристику объекта, — это:
- поле
 - запись
 - отчёт
 - ключ
23. Системы управления базами данных используются (выберите наиболее полный ответ):
- для создания баз данных, хранения и поиска в них необходимой информации
 - для сортировки данных
 - для организации доступа к информации в компьютерной сети
 - для создания баз данных
24. Какое из слов НЕ является названием СУБД?
- Microsoft Access*
 - OpenOffice Base*
 - OpenOffice Writer*
 - 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	12

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

- а) 2
- б) 3
- в) 5
- 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

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 заносив полученные результаты в соответствующие графы таблиц. Эта работа требовала больших временных затрат; на исправление незначительной ошибки, допущенной расчётчиком, уходило 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 называют и таблицы, созданные с помощью одноимённых 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

Широкое распространение получили табличные процессоры *Microsoft Excel*, *LibreOffice Calc* и *OpenOffice 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).

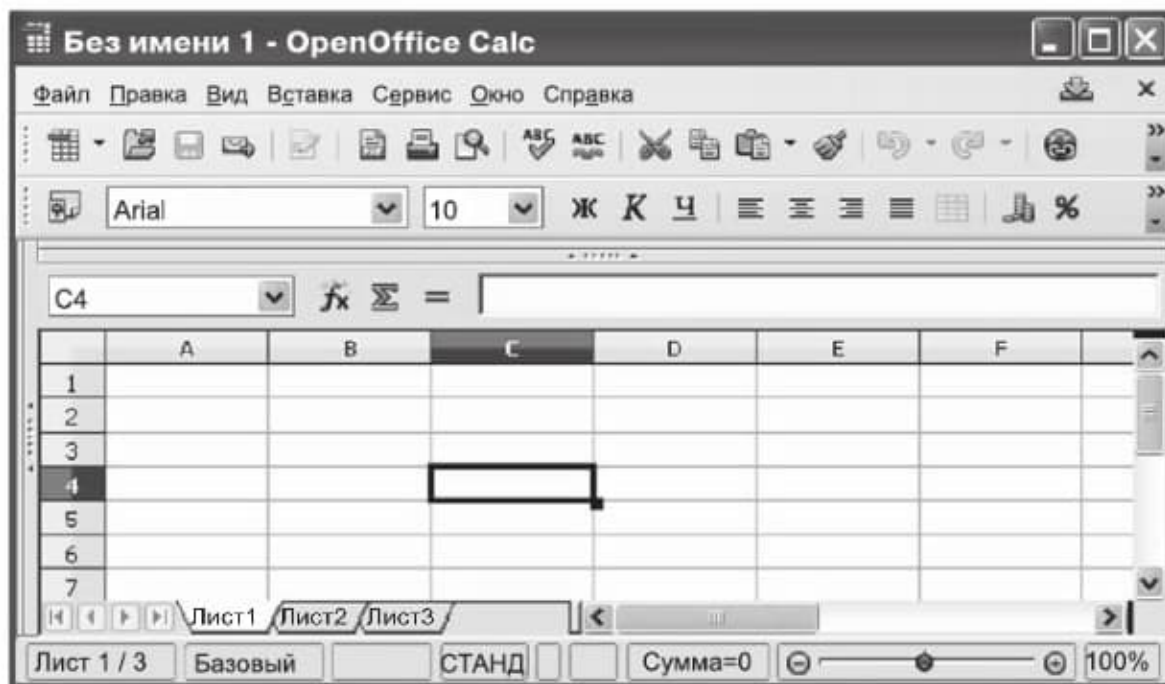


Рис. 3.1. Интерфейс табличного процессора *OpenOffice Calc*

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.

Табличный курсор — прямоугольная рамка, которую можно поместить в любую ячейку. Ячейка таблицы, которую в данный момент занимает курсор, называется **активной ячейкой**. Вводить или редактировать данные можно только в активной ячейке. На рис. 3.1 активной является ячейка С4.

В нижнем правом углу активной ячейки можно видеть небольшой квадратик — **маркер автозаполнения**.

Адрес активной ячейки и вводимые в неё данные отражаются в строке ввода. В строке ввода можно редактировать данные, хранящиеся в активной 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 представляет вся таблица, наименьший — одна ячейка. Примеры диапазонов: А1:А10, В2:С2, В2:Д10.

¹ В современных версиях *Microsoft Excel* положение ячейки может обозначаться буквой R (от англ. *row*), за которой следует номер строки, и буквой C (от англ. *column*), за которой следует номер столбца, например R1C1.

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, листы книги табличного процессора можно перелистывать, щёлкая на ярлычках, расположенных внизу окна. Каждому листу книги пользователь может дать имя, исходя из содержимого этого листа.

Лист — рабочая область электронных таблиц, состоящая из ячеек.
Книга — документ электронных таблиц, 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 (подписи, заголовки, пояснения) пужны 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	23.12.2021
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 с электронными таблицами

Можно выделить следующие режимы работы с электронными таблицами:

- 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 размера, начертания и выравнивания относительно ячейки, направления написания. Помимо этого, пользователю доступны средства 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 Calc* установить режим отображения формул, следует:

- 1) выполнить команду **Сервис** → **Параметры** → **OpenOffice 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 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. Кем и когда были созданы первые электронные таблицы? (Для ответа на вопрос используйте дополнительные источники информации.)



2. 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 профессий до конца прошлого века выполняли вручную. Как вы можете объяснить это противоречие? Обсудите этот вопрос в группе.

3. Что понимают под электронными таблицами?

4. Назовите основные элементы окна табличного процессора. Перечислите общие элементы окна табличного процессора и окна текстового процессора.

5. Какой табличный процессор установлен на вашем компьютере? Сколько строк и столбцов могут иметь создаваемые в нём электронные таблицы?

6. Как именуются ячейки таблицы? Почему имя ячейки иначе называют её координатами?

7. Какие данные могут храниться в ячейках таблицы?



8. Сравните операции ввода, редактирования и форматирования текстовой информации в текстовом процессоре и в электронных таблицах.



9. Сравните возможности ввода чисел в таблицы в текстовом процессоре и в электронных таблицах.

10. В одной из ячеек электронной таблицы записана формула $= 50 + 25 / (4 * 10 - 2) * 8$. Какое арифметическое выражение ей соответствует?



а) $50 + \frac{25}{4} \cdot 10 - 2 \cdot 8$

б) $\frac{50 + 25}{4 \cdot 10 - 2} \cdot 8$

$$в) \frac{50+25}{(4 \cdot 10 - 2) \cdot 8}$$

$$г) 50 + \frac{25}{4 \cdot 10 - 2} \cdot 8$$

11. Сформулируйте правила ввода формул в электронных таблицах.
12. Почему электронные таблицы часто называют динамическими?
13. Сравните электронные таблицы и таблицы реляционной базы данных: что в них общее? в чём основное различие?
14. Дайте краткую характеристику режимов формирования электронных таблиц.
15. На рис. 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.
16. Среди данных, содержащихся в электронной таблице, можно выделить исходные данные (они вводятся пользователем) и результаты (они рассчитываются по формулам). Назовите диапазоны ячеек с исходными данными и с рассчитываемыми данными в таблице из практического задания 1 (раздел «Задания для практических работ» к главе 3).
17. Известно ли вам, что даже в текстовых процессорах предусмотрена возможность выполнения вычислений? Исследуйте такую возможность в имеющемся в вашем распоряжении текстовом процессоре. Подготовьте небольшое сообщение о вычислительных возможностях текстового процессора.

§ 3.2

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.

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 (рис. 3.4). Она содержит относительную ссылку A1, которая воспринимается табличным процессором следующим образом: содержимое ячейки, находящееся в том же столбце и на одну строку выше той, где 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 (см. 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).



	A	B		A	B	
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 выполнялись 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).

	A	B	C	D
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).

	A	B	C
1	Начальная сумма вклада:		10 000
2	Месяц	Сумма	Приращение
3	1	=C1+C1*0,012	=B3-\$C\$1
4	2	=B3+B3*0,012	=B4-\$C\$1
5	3	=B4+B4*0,012	=B5-\$C\$1
6	4	=B5+B5*0,012	=B6-\$C\$1
7	5	=B6+B6*0,012	=B7-\$C\$1
8	6	=B7+B7*0,012	=B8-\$C\$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 только относительная часть ссылки автоматически корректируется (рис. 3.8).

	A	B	C	D
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 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 А 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 В2 формулу $=\$A2+B\1 и скопируйте её на весь диапазон В2: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, выступающими в качестве аргументов. Использование функций позволяет упростить формулы и сделать процесс вычислений более понятным.

В электронных таблицах реализовано несколько сотен встроенных функций, подразделяющихся на математические, статистические, логические, текстовые, финансовые и др.

Каждая функция имеет уникальное имя, которое используется для её вызова. Имя, как правило, представляет собой сокращённое название функции на естественном языке. При выполнении табличных расчётов достаточно часто используются функции:

- СУММ (SUM) — суммирование аргументов;
- МИН (MIN) — определение наименьшего значения из списка аргументов;
- МАКС (MAX) — определение наибольшего значения из списка аргументов;

COUNT (СЧЁТ) — определение количества числовых ячеек диапазона (пустые и текстовые ячейки не учитываются).

Диалоговое окно **Мастер функций** позволяет упростить создание формул и свести к минимуму количество опечаток и синтаксических ошибок. При вводе функции в формулу диалоговое окно **Мастер функций** отображает имя функции, все её аргументы, описание функции и каждого из аргументов, текущий результат функции и всей формулы.

Пример 4

Правила судейства в международных соревнованиях по одному из видов спорта таковы:

- 1) выступление каждого спортсмена оценивают N судей;
- 2) максимальная и минимальная оценки (по одной, если их несколько) каждого спортсмена отбрасываются;
- 3) в зачёт спортсмену идёт среднее арифметическое оставшихся оценок.

Информация о соревнованиях представлена в электронной таблице.

	A	B	C	D	E	F
1	Протокол соревнований					
2		Спортсмен 1	Спортсмен 2	Спортсмен 3	Спортсмен 4	Спортсмен 5
3	Судья 1	5,9	9,8	7,8	9,1	6,9
4	Судья 2	6,3	9,7	8,0	9,3	7,8
5	Судья 3	5,4	8,9	8,2	8,8	8,1
6	Судья 4	6,6	9,9	7,9	9,2	7,8
7	Судья 5	5,8	9,2	6,4	9,9	8,2
8	Судья 6	6,2	9,5	8,9	9,4	8,9

Требуется подсчитать оценки всех участников соревнований и определить оценку победителя. Для этого:

- 1) в ячейки A10, A11, A12 и A14 заносим тексты «Максимальная оценка», «Минимальная оценка», «Итоговая оценка», «Оценка победителя»;
- 2) в ячейку B10 заносим формулу =МАКС(B3:B8); копируем содержимое ячейки B10 в ячейки C10:F10;
- 3) в ячейку B11 заносим формулу =МИН(B3:B8); копируем содержимое ячейки B11 в ячейки C11:F11;
- 4) в ячейку B12 заносим формулу =(СУММ(B3:B8)-B10-B11)/4; копируем содержимое ячейки B12 в ячейки C12:F12;
- 5) в ячейку B14 заносим формулу =МАКС(B12:F12).



Результат решения задачи:

	A	B	C	D	E	F
1	Протокол соревнований					
2		Спортсмен 1	Спортсмен 2	Спортсмен 3	Спортсмен 4	Спортсмен 5
10	Максимальная оценка	6,6	9,9	8,9	9,9	8,9
11	Минимальная оценка	5,4	8,9	6,4	8,8	6,9
12	Итоговая оценка	6,05	9,55	7,975	9,25	7,975
13						
14	Оценка победителя	9,55				

3.2.3. Логические функции

При изучении предшествующего материала вы неоднократно встречались с логическими операциями НЕ, И, ИЛИ (**not**, **and**, **or**). Построенные с их помощью логические выражения вы использовали при организации поиска в базах данных, при программировании различных вычислительных процессов.

Реализованы логические операции и в электронных таблицах, но здесь они представлены как функции: сначала записывается имя логической операции, а затем в круглых скобках перечисляются логические операнды.

Например, логическое выражение, соответствующее двойному неравенству $0 < A1 < 10$, в электронных таблицах будет записано как

$\text{И}(A1>0;A1<10)$

(в англоязычной версии программы $\text{AND}(A1>0;A1<10)$).



Вспомните, как аналогичное логическое выражение мы записывали при знакомстве с базами данных и языком программирования Паскаль (Python).

Пример 5

Вычислим в электронных таблицах значения логического выражения НЕ А И НЕ В при всех возможных значениях входящих в него логических переменных.



	A	B	C	D	E
1	Таблица истинности НЕ А И НЕ В				
2	A	B	НЕ А	НЕ В	НЕ А И НЕ В
3	ЛОЖЬ	ЛОЖЬ	=НЕ(A3)	=НЕ(B3)	=И(C3;D3)
4	ЛОЖЬ	ИСТИНА	=НЕ(A4)	=НЕ(B4)	=И(C4;D4)
5	ИСТИНА	ЛОЖЬ	=НЕ(A5)	=НЕ(B5)	=И(C5;D5)
6	ИСТИНА	ИСТИНА	=НЕ(A6)	=НЕ(B6)	=И(C6;D6)

При решении этой задачи мы следовали известному вам алгоритму построения таблицы истинности для логического выражения. Вычисления в диапазонах ячеек C3:C6, D3:D6, E3:E6 проводятся компьютером по заданным нами формулам.

Для проверки условий при выполнении расчётов в электронных таблицах реализована логическая функция ЕСЛИ (IF), называемая **условной функцией**.

Условная функция имеет следующую структуру:

ЕСЛИ (<условие>;<действие 1>;<действие 2>)

Здесь <условие> — логическое выражение, т. е. любое выражение, построенное с помощью операций отношения и логических операций, принимающее значение ИСТИНА или ЛОЖЬ.

Если логическое выражение истинно, то значение ячейки, в которую записана условная функция, определяет <действие 1>, если ложно — <действие 2>¹.

Что вам напоминает структура условной функции?

Пример 6

Рассмотрим задачу о приёме в школьную баскетбольную команду: ученик может быть принят в эту команду, если его рост не менее 170 см.

Данные о претендентах (фамилия, рост) представлены в электронной таблице.

	А	В	С
1	Баскетбольная команда		
2	Ученик	Рост, см	Решение
3	Васечкин	169	=ЕСЛИ(В3>=170;"принят";"не принят")
4	Дроздов	182	=ЕСЛИ(В4>=170;"принят";"не принят")
5	Иванов	173	=ЕСЛИ(В5>=170;"принят";"не принят")
6	Куликов	158	=ЕСЛИ(В6>=170;"принят";"не принят")
7	Петров	190	=ЕСЛИ(В7>=170;"принят";"не принят")
8	Сидоров	170	=ЕСЛИ(В8>=170;"принят";"не принят")
9			=СЧЁТЕСЛИ(С3:С8;"принят")

Использование условной функции в диапазоне ячеек C3:C8 позволяет вынести решение (принят/не принят) по каждому претенденту.

¹ Действием может быть вычисление формулы, ввод числа или текста в ячейку.

Функция СЧЁТЕСЛИ (COUNTIF) позволяет подсчитать количество ячеек в диапазоне, удовлетворяющих заданному условию. С помощью этой функции в ячейке С9 подсчитывается число претендентов, прошедших отбор в команду.

Функция СУММЕСЛИ (SUMMIF) суммирует значения ячеек диапазона, удовлетворяющие заданному условию.

Функция СРЗНАЧЕСЛИ (AVERAGEIF) вычисляет среднее значение ячеек диапазона, удовлетворяющих заданному условию.

3.2.4. Обработка больших наборов данных

Все примеры вычислений в электронных таблицах, которые были рассмотрены до сих пор, вы могли выполнить вручную или с помощью калькулятора. Но приведённые выше функции можно применять и для обработки больших наборов данных, содержащих тысячи строк. Вряд ли кто-то из вас возьмётся за обработку такой таблицы вручную!

Пример 7

В электронную таблицу занесли данные о тестировании учеников по выбранным ими предметам.

	А	В	С	Д	Е
1	Ученик	Район	Математика	Информатика	Физика
2	Код 1	Майский	100	58	74
3	Код 2	Майский	100	80	81
4	Код 3	Майский	99	76	63
5	Код 4	Центральный	98	86	51
6	Код 5	Подгорный	97	66	47
7	Код 6	Заречный	96	98	63

В столбце А записан код фамилии и имени ученика, в столбце В — название района, в котором учится ученик; в столбцах С, Д, Е — баллы, набранные учеником на тестировании по математике, информатике и физике соответственно.

Всего в электронную таблицу были занесены данные 1000 учеников. На основании данных, содержащихся в этой таблице, требуется ответить на ряд вопросов.

Вопрос 1. Сколько всего учащихся Майского района приняли участие в тестировании? Ответ на этот вопрос надо записать в ячейку G1002 таблицы.

Вариант 1 решения

Заполним ячейки диапазона F2:F1001 вспомогательной информацией: запишем в ячейку 1, если в соответствующей ей строке указано название района «Майский», и 0 — в противном случае.

Для этого в ячейку F2 запишем формулу

```
=ЕСЛИ(B2="Майский";1;0)  
(=IF(B2="Майский";1;0))
```

Скопируем эту формулу во все ячейки диапазона F3:F1001.

Обратите внимание! Для того чтобы быстро скопировать формулу из ячейки F2 во все следующие ячейки столбца F, достаточно выполнить двойной щелчок мышью на маркере автозаполнения ячейки с формулой.

В ячейку G1002 запишем формулу

```
=СУММ(F2:F1001)  
(=SUM(F2:F1001))
```

Вариант 2 решения

Можно воспользоваться функцией СЧЁТЕСЛИ (COUNTIF), подсчитав с её помощью количество ячеек диапазона B2:B1001, содержащих название района «Майский». Для этого достаточно в ячейку G1002 записать формулу

```
=СЧЁТЕСЛИ(B2:B1001;"=Майский")  
(=COUNTIF(B2:B1001;"=Майский"))
```

Вопрос 2. Сколько учащихся Кировского района набрали больше 75 баллов по математике? Ответ на этот вопрос надо записать в ячейку G1003 таблицы.

Вариант 1 решения

Заполним ячейки диапазона G2:G1001 вспомогательной информацией: запишем в ячейку 1, если в соответствующей ей строке указано название района «Кировский» и есть информация о том, что количество баллов по математике больше, чем 75; в противном случае укажем 0.

Для этого в ячейку G2 запишем формулу

```
=ЕСЛИ(И(B2="Кировский";C2>75);1;0)  
(=IF(AND(И(B2="Кировский";C2>75);1;0))
```

Скопируем эту формулу во все ячейки диапазона G3:G1001.

В ячейку G1003 запишем формулу

```
=СУММ(G2:G1001)  
(=SUM(G2:G1001))
```



Вариант 2 решения

Функция СЧЁТЕСЛИ (COUNTIF) не работает с составными условиями. Для этого предназначена функция СЧЁТЕСЛИМН (COUNTIFS). Вот формула, которую для ответа на вопрос достаточно ввести в ячейку G1003:

```
=СЧЁТЕСЛИМН(B2:B102;"Кировский";C2:C102;">75")
(=COUNTIFS(B2:B102;"Кировский";C2:C102;">75"))
```

Вопрос 3. Каков средний балл по информатике учеников из Центрального района? Ответ на этот вопрос надо вычислить и записать в ячейку G1004 таблицы с точностью не менее двух знаков после запятой.



Вариант 1 решения

Заполним ячейки диапазона H2:H1001 вспомогательной информацией: если в строке указано название района «Центральный», то в соответствующей ячейке укажем количество баллов, набранных учеником по информатике; в противном случае укажем 0. В ячейку H2 запишем формулу

```
=ЕСЛИ(B2="Центральный";D2;0)
(=IF(B2="Центральный";D2;0))
```

Определим число учеников Центрального района, сдававших информатику; оно равно количеству ненулевых значений в столбце H. В ячейку I2 запишем формулу

```
=СЧЁТЕСЛИ(H2:H1001;">0")
(=COUNTIF(H2:H1001;">0"))
```

Подсчитаем сумму баллов, набранных учениками Центрального района по информатике; она равна сумме значений ячеек столбца H. В ячейку I3 запишем формулу

```
=СУММ(H2:H1001)
(=SUM(H2:H1001))
```

В ячейку G1004 запишем формулу

```
=I3/I2
```

Установим требуемую точность ответа.



Вариант 2 решения

Для подсчёта числа учеников Центрального района в ячейку I2 введём формулу

```
=СЧЁТЕСЛИ(B2:B1001;"=Центральный")
(=COUNTIF(B2:B1001;"=Центральный"))
```

Чтобы найти суммарный балл, набранный учениками Центрального района по информатике, можно воспользоваться функцией СУММЕСЛИ(SUMIF). Для этого в ячейку I3 введём формулу

=СУММЕСЛИ(B2:B1001;"=Центральный";D2:D1001)

(=SUMIF(B2:B1001;"=Центральный";D2:D1001))

Обратите внимание на аргументы этой функции:

- первый — это диапазон, по которому выполняется проверка условия;
- второй — проверяемое условие;
- третий — диапазон, значения которого суммируются.

В ячейку G1004 запишем формулу

=I3/I2

Установим требуемую точность ответа.

Вариант 3 решения

Функция СРЗНАЧЕСЛИ(AVERAGEIF) позволяет решить задачу с помощью единственной формулы:

= СРЗНАЧЕСЛИ(B2:B1001;"=Центральный";D2:D1001)

(=AVERAGEIF(B2:B1001;"=Центральный";D2:D1001))

Откройте файл `Тестирование.xls` с данными из примера 7. Выполните на компьютере все рассмотренные выше вычисления.



САМОЕ ГЛАВНОЕ

Для организации вычислений в электронных таблицах используются формулы, которые могут включать в себя ссылки и функции.

Различают относительные, абсолютные и смешанные ссылки.

Относительная ссылка определяет расположение ячейки с данными относительно ячейки, в которой записана формула. При изменении позиции ячейки, содержащей формулу, изменяется и ссылка.

Абсолютная ссылка всегда ссылается на ячейку, расположенную в определённом месте. При изменении позиции ячейки, содержащей формулу, абсолютная ссылка не изменяется.

Смешанная ссылка содержит либо абсолютно адресуемый столбец и относительно адресуемую строку, либо относительно адресуемый столбец и абсолютно адресуемую строку. При изменении позиции ячейки, содержащей формулу, относительная часть адреса изменяется, а абсолютная часть адреса не изменяется.

Функции — это заранее определённые и встроенные в электронные таблицы формулы. Использование функций позволяет упростить формулы и сделать процесс вычислений более понятным.

Электронные таблицы удобно применять для обработки больших наборов данных, содержащих тысячи строк.



Вопросы и задания

1. Назовите основные типы ссылок.
2. Охарактеризуйте относительный тип ссылок.
3. По данным электронной таблицы определите значение в ячейке C1.

	A	B	C
1	3	=A1*2+2	=A1+B1

4. Дан фрагмент электронной таблицы:

	A	B	C
1	110	25	=A1+B1
2	45	55	
3	120	60	

Определите значения в ячейках C2 и C3 после копирования в них формулы из ячейки C1. Проверьте свои предположения на компьютере.

5. Дан фрагмент электронной таблицы:

	A	B	C	D
1	110	25	=A1+B1	
2	45	55	=A2+B2	
3	120	60	=A3+B3	

Определите значения в ячейках диапазона D1:D3 после копирования в них формулы из ячейки C3. Проверьте свои предположения на компьютере.

6. Охарактеризуйте абсолютный тип ссылок.
7. Дан фрагмент электронной таблицы:

	A	B	C
1	110	25	=\$A\$1+B1
2	45	55	
3	120	60	

Определите значения в ячейках C2 и C3 после копирования в них формулы из ячейки C1. Проверьте свои предположения на компьютере.

8. Охарактеризуйте смешанный тип ссылок.
9. Дан фрагмент электронной таблицы:

	A	B	C	D
1	110	25	=\$A1+B1	
2	45	55		
3	120	60		

Определите значения в ячейках диапазона C1:D3 после копирования в них формулы из ячейки C1. Проверьте свои предположения на компьютере.

10. Как можно изменить тип ссылки?
11. О чём идёт речь в следующем высказывании: «Знак доллара "замораживает" как весь адрес, так и его отдельную часть»? Дайте развёрнутый комментарий к высказыванию, основываясь на материале параграфа. Обсудите этот вопрос в группе.
12. Для чего нужны встроенные функции?
13. Какие категории встроенных функций реализованы в табличном процессоре, имеющемся в вашем распоряжении?

14. Дан фрагмент электронной таблицы:

	A	B	C	D
1	2	3	1	=MIN(A1:C1)
2	1	4	=B2+A2	=SUM(A2:C2)
3				=D2*D1

Определите значение в ячейке D3.

15. Какая из формул не содержит ошибок?

- а) =ЕСЛИ((C4>4) И (C5>4));"Принят!";"Не принят")
- б) =ЕСЛИ(И(D2=0;B2/4);D3-A1;D3+A1)
- в) =ЕСЛИ((A4=0 И D1<0);1;0)
- г) =ЕСЛИ(ИЛИ(A2>10;C2>10);1;"ура!")

16. В ячейке A5 электронной таблицы находится суммарная стоимость товаров, заказанных Иваном в интернет-магазине. Формула, позволяющая подсчитать полную стоимость заказа с учётом стоимости его доставки, имеет вид:

=ЕСЛИ(A5>2000;A5;A5+150).

По данной формуле постройте блок-схему. Определите, какие льготы предоставляются покупателю в случае, если суммарная стоимость заказанных им товаров превышает 2000.

17. Оплата за аренду конференц-зала вычисляется по следующим правилам: каждый из первых четырёх часов аренды стоит 1000 рублей, каждый последующий час — 750 рублей. В ячейке B8 электронной таблицы находится количество полных часов аренды зала. Какая из формул позволяет подсчитать полную стоимость аренды зала?

- а) =ЕСЛИ(B8<=4;B8*1000;4000+B8*750)
- б) =ЕСЛИ(B8<=4;B8*1000;B8*1000+(B8-4)*750)
- в) =ЕСЛИ(B8<=4;B8*1000;(B8+(B8-4)*750))
- г) =ЕСЛИ(B8<=4;B8*1000;4000+(B8-4)*750)

18. В электронную таблицу занесли результаты муниципального этапа олимпиады по программированию:

	A	B	C	D	E	F
1	Ученик	Класс	Школа	1-я задача	2-я задача	3-я задача
2	Иванов Иван	7	Ивановская СОШ	25	25	25
3	Петрова Света	9	Лучинская СОШ	20	20	20

В столбце А указаны фамилия и имя учащегося; в столбцах В и С — класс и школа, в которой он учится; в столбцах D, E, F — баллы, полученные учеником за решение каждой из олимпиадных задач. За каждую задачу можно получить от 0 до 25 баллов. Всего в электронную таблицу занесены данные 115 учащихся. Гарантируется, что есть ученики, правильно решившие все задачи. Порядок записей в таблице произвольный.

Определите, значения каких величин будут получены в результате последовательного выполнения вычислений по следующим формулам.

1) =СУММ(D2:F2)

Формула находится в ячейке G2 и копируется в диапазон G3:G116.

2) =МАКС(G2:G116)

Формула находится в ячейке G117.

3) =ЕСЛИ(G2=\$G\$117;"Победитель!";"")

Формула находится в ячейке H2 и копируется в диапазон H3:H116.

4) =СЧЁТЕСЛИ(B2:B116;"=7")

Формула находится в ячейке B117.

5) =СУММ(D2:F2)*100/75

Формула находится в ячейке I2 и копируется в диапазон I3:I116.

6) =СРЗНАЧ(F2:F116)

Формула находится в ячейке F117.

7) =ЕСЛИ(И(G2<\$G\$117;I2>80);"Призёр";"")

Формула находится в ячейке J2 и копируется в диапазон J3:J116.

8) =СЧЁТЕСЛИ(J2:J116;"Призёр")

Формула находится в ячейке J117.

9) =ЕСЛИ(C2="Ивановская СОШ";D2+E2+F2;"")

Формула находится в ячейке K2 и копируется в диапазон K3:K116.

§ 3.3

Средства анализа и визуализации данных

Ключевые слова:

- анализ данных
- сортировка
- поиск (фильтрация)
- визуализация
- диаграмма
- график
- круговая диаграмма
- гистограмма (столбчатая диаграмма)
- численное моделирование

Анализ — метод исследования, характеризующийся выделением и изучением отдельных частей объектов исследования. **Анализ данных** — процесс обработки информации после её получения и сбора, направленный на выявление закономерностей и достижение ясности в исследуемой ситуации. Результаты анализа данных позволяют осуществлять прогнозирование и принимать решения.

3.3.1. Сортировка и поиск данных

Важной частью анализа данных является их **сортировка**. С помощью сортировки данные можно расположить **по возрастанию** или **по убыванию** содержимого ячеек (табл. 3.3); также можно организовать сортировку по цвету ячеек, цвету шрифта и некоторым другим параметрам.

Таблица 3.3

Основные способы выполнения сортировки

Данные	По возрастанию	По убыванию
Текст	От А до Z, от А до Я	От Z до А, от Я до А
Числа	От наименьших к наибольшим	От наибольших к наименьшим
Дата и время	От старых к новым	От новых к старым

Сортировка позволяет группировать в одном столбце или одной строке данные с одинаковыми значениями, в группах с одинаковыми значениями осуществлять последующую сортировку другого столбца или строки.

Результатом сортировки является удобная для восприятия форма представления данных, что позволяет быстрее находить необходимую информацию и в конечном счёте принимать более обдуманные решения.

В отличие от баз данных электронные таблицы позволяют сортировать данные в отдельном столбце. Будьте внимательны: сортировка по одному столбцу диапазона может привести к нежелательным результатам.

Вспомните задачу о формировании школьной баскетбольной команды и подумайте, что получится, если отсортировать по возрастанию данные только в ячейках диапазона B3:B8.

Пример 1

В электронную таблицу занесены данные о ценах на бензин трёх марок (92, 95, 98) на заправочных станциях некоторого региона. Каждой заправке присвоен уникальный номер; всего имеются сведения о тысяче заправочных станций.

	A	B	C
1	№ станции	Марка бензина	Цена, руб.
2	1	92	44,6
3	2	98	54,1
4	3	92	45,2
5	4	95	48,4

Требуется ответить на следующий вопрос: «Сколько заправочных станций региона продают бензин марки 92 по максимальной цене?»

Для ответа на этот вопрос сначала нужно узнать максимальную цену на бензин марки 92. Сделать это можно следующим образом (рис. 3.9)¹:

- 1) отсортировать таблицу по возрастанию по марке бензина;
- 2) при равных значениях марки бензина отсортировать таблицу по убыванию цены бензина².

¹ Существуют и другие способы решения этой задачи.

² Вместо сортировки можно использовать функцию поиска максимального значения в диапазоне ячеек с ценами на бензин марки 92.

Затем можно к диапазону ячеек с ценами на бензин марки 92 применить функцию СЧЁТЕСЛИ и с её помощью определить число заправочных станций, продающих бензин по максимальной цене.

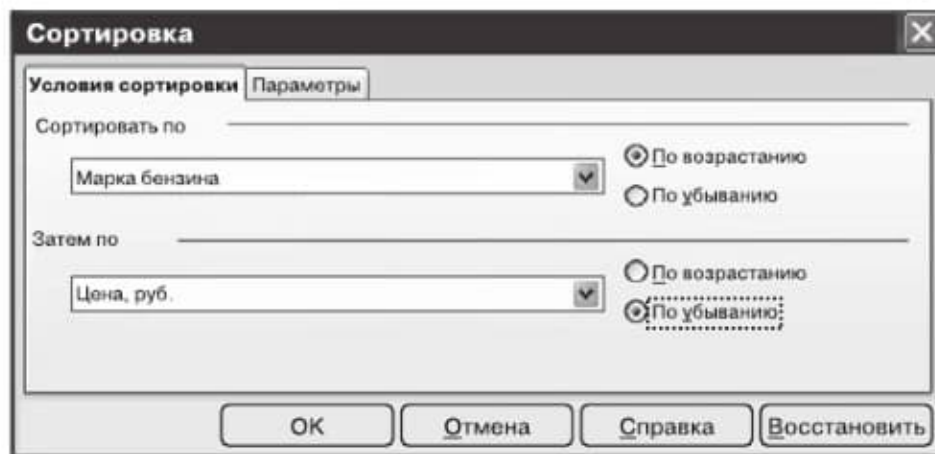


Рис. 3.9. Окно Сортировка в электронных таблицах *OpenOffice Calc*

Поиск данных в электронных таблицах осуществляется с помощью **фильтров**, которые «не пропускают» на экран записи, не удовлетворяющие условиям поиска.

Например, к электронной таблице с информацией о заправочных станциях можно применить фильтр для поля «Марка бензина», состоящий из условия «=» со значением 92 (рис. 3.10).



Рис. 3.10. Фрагмент окна Стандартный фильтр в электронных таблицах *OpenOffice Calc*

В результате мы получим информацию только о тех заправочных станциях, где продаётся бензин марки 92:

	А	В	С
1	№ станции	Марка бензина	Цена, руб.
2	1	92	44,6
3	3	92	45,2

Полученную таблицу можно подвергать дальнейшей сортировке и фильтрации.

3.3.2. Построение диаграмм

С помощью электронных таблиц можно не только быстро и качественно обрабатывать большие объёмы однотипной числовой информации, но и **визуализировать** её, представляя числовые данные в графической форме, более удобной для зрительного восприятия и анализа.

Основным средством графического представления табличных данных являются диаграммы.

Диаграмма — средство наглядного графического представления количественных данных. Диаграммы помогают анализировать данные, проводить их сравнение и выявлять закономерности, скрытые в последовательностях чисел.

Электронные таблицы позволяют создавать диаграммы нескольких типов, основными из которых являются график, круговая диаграмма и гистограмма.

Графики используются для отображения зависимости значений одной величины (функции) от другой (аргумента); графики позволяют отслеживать динамику изменения данных (рис. 3.11).

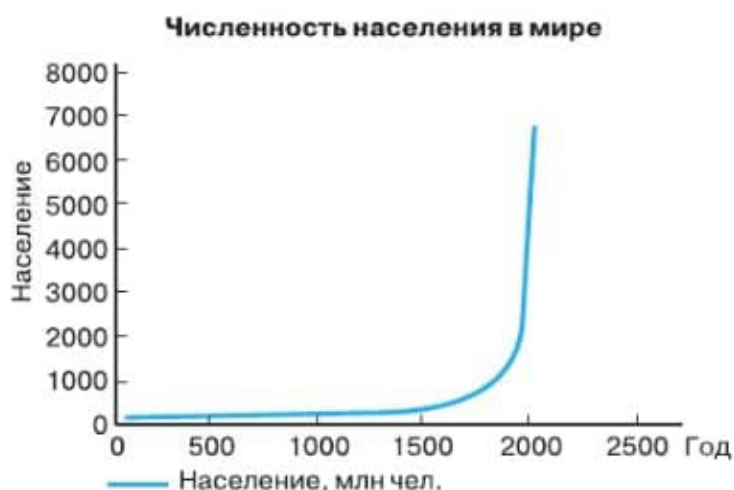


Рис. 3.11. Пример графика

Круговые диаграммы используются для отображения величин (размеров) частей некоторого целого; в них каждая часть целого представляется как сектор круга, угловой размер которого прямо пропорционален величине (размеру) части (рис. 3.12).

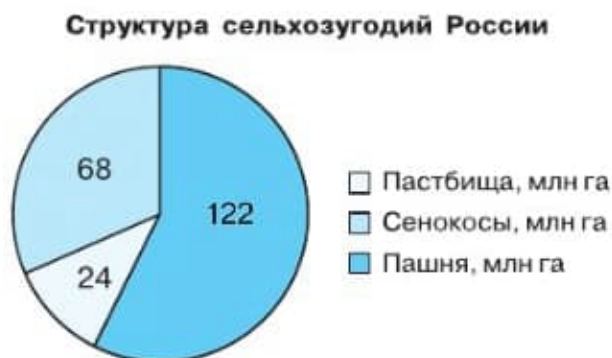


Рис. 3.12. Пример круговой диаграммы

Гистограммы (столбчатые диаграммы) используются для сравнения нескольких величин; в них величины отображаются в виде вертикальных или горизонтальных столбцов. Высоты (длины) столбцов соответствуют отображаемым значениям величин (рис. 3.13).

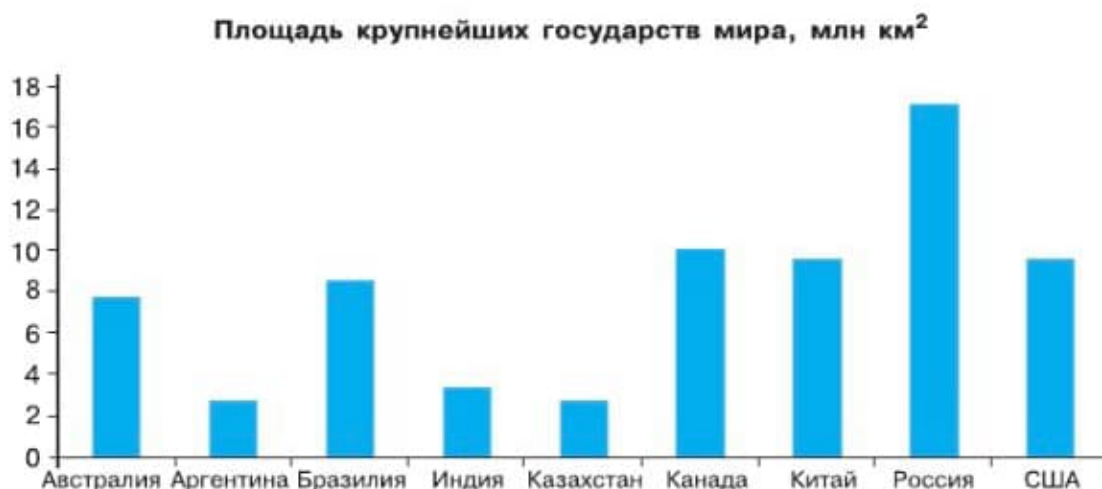


Рис. 3.13. Пример столбчатой диаграммы

Ярусные диаграммы (гистограммы с накоплением) дают представление о вкладе каждой из нескольких величин в общую сумму; в ней значения нескольких величин изображаются объединёнными в одном столбце (рис. 3.14).

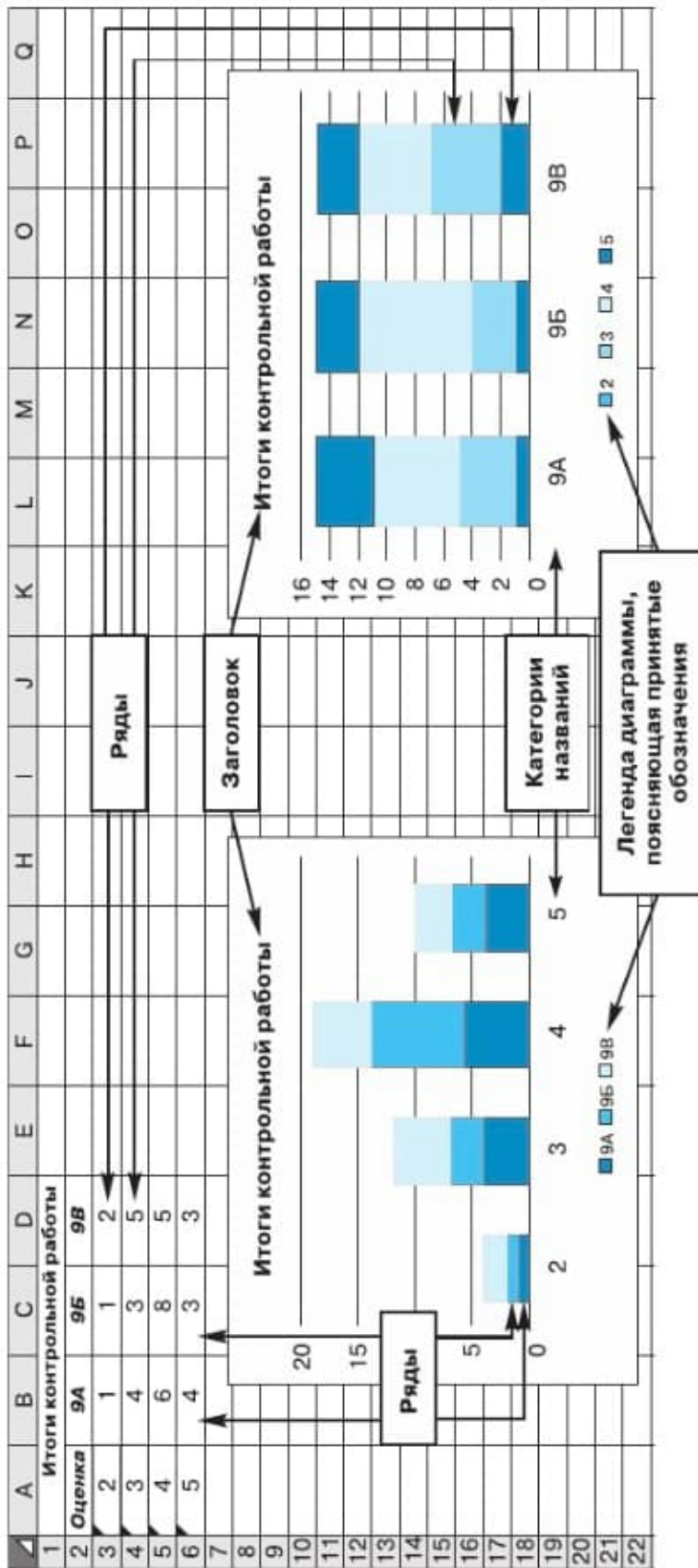


Рис. 3.14. Примеры ярусных диаграмм в электронных таблицах

Ряд данных — это множество значений, которые необходимо отобразить на диаграмме. Диаграммы позволяют визуально сопоставить значения одного или нескольких рядов данных (см. рис. 3.14). Наборы соответствующих друг другу значений из разных рядов называются **категориями**.

Большинство диаграмм строятся в прямоугольной системе координат, где вдоль оси *X* подписываются названия категорий, а по оси *Y* отмечаются значения рядов данных.

Диаграмма — это составной объект, который может содержать:

- заголовок диаграммы;
- оси категорий и значений и их названия;
- изображения данных;
- легенду, поясняющую принятые обозначения.

В электронных таблицах диаграммы строятся под управлением **Мастера диаграмм**, в котором предусмотрены следующие основные шаги:

- 1) выбор типа диаграммы;
- 2) выбор данных, на основе которых строится диаграмма;
- 3) настройка элементов оформления диаграммы.

Диаграммы в электронных таблицах сохраняют свою зависимость от данных, на основе которых они построены: при изменении данных соответствующие изменения происходят в диаграмме автоматически.

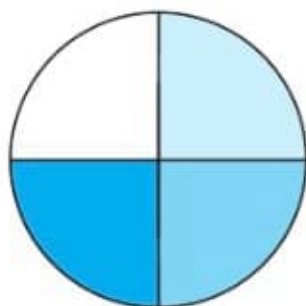
Пример 2

Дан фрагмент электронной таблицы в режиме отображения формул:

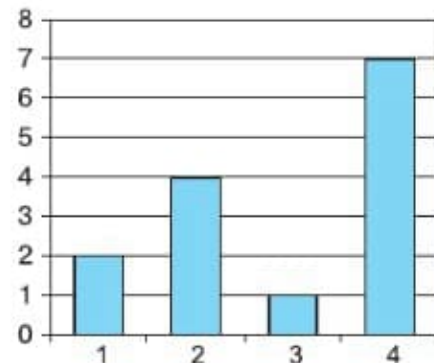
	A	B
1	1	2
2	=B1-A1	=2*B1
3	=(A2+B1+B2)/B4	1
4	=B2/B1+B3	7

После выполнения вычислений по значениям ячеек диапазона A1:A4 была построена диаграмма. Требуется найти получившуюся диаграмму среди приведённых ниже образцов.

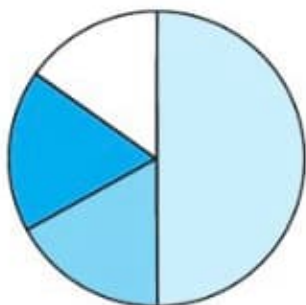
а)



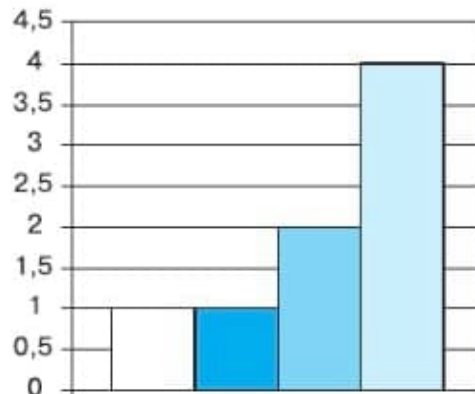
б)



в)



г)



Решение

После вычислений по формулам в ячейках таблицы будут следующие значения:

	A	B
1	1	2
2	1	4
3	1	1
4	3	7

Диапазон A1:A4 содержит три одинаковых значения — три единицы; четвёртое значение — это 3, что равно сумме трёх других значений. На диаграмме значениям диапазона A1:A4 долж-

ны соответствовать три равных по площади столбца или сектора, а также столбец или сектор, площадь которого равна сумме площадей трёх других столбцов или секторов. Таким условиям соответствует только диаграмма *в*.



Самостоятельно обоснуйте, почему значения ячеек диапазона A1:A4 не могут быть представлены диаграммами *а*, *б* или *г*. Какая диаграмма может быть построена по значениям ячеек диапазона B1:B4?

3.3.3. Численное моделирование в электронных таблицах

Как известно, математическая модель — это модель, в которой сведения об объекте (предмете, процессе, явлении) моделирования представлены в виде математических символов, выражений, формул.



Моделирование поведения объекта путём получения численного решения уравнений математической модели называется **численным моделированием**.

Рассмотрим несколько примеров численного моделирования в электронных таблицах.

Пример 3

Вернувшись домой с занятий кружка по математическому моделированию, Александр застал свою младшую сестрёнку за странным занятием: перед Ритой лежал лист картона размером 200×300 мм, на котором она что-то чертила, измеряла линейкой, вычисляла на калькуляторе, снова чертила, измеряла и вычисляла. Рита была сильно озабочена, и Александр решил выяснить, что же она делает. Оказалось, что на уроке математики Рита узнала, как вычисляется объём прямоугольного параллелепипеда. Дома Рита и её одноклассники должны были изготовить из листа картона прямоугольный параллелепипед (коробку без крышки) и вычислить его объём. При этом было важно изготовить коробку наибольшей вместимости (рис. 3.15).

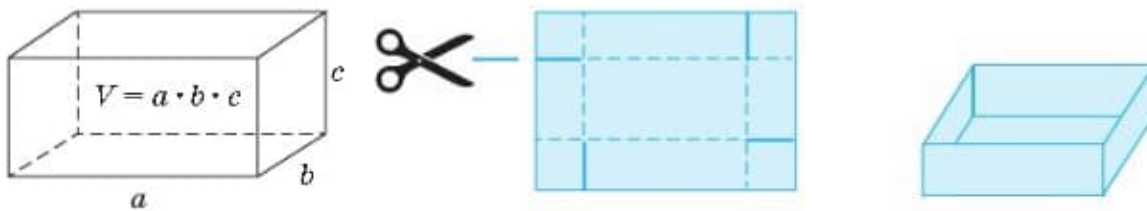


Рис. 3.15. Изготовление коробки в форме прямоугольного параллелепипеда

Александр решил помочь сестре. Он быстро набросал чертёж (рис. 3.16) и записал формулу для вычисления объёма коробки:

$$V = (300 - 2x) \cdot (200 - 2x) \cdot x.$$

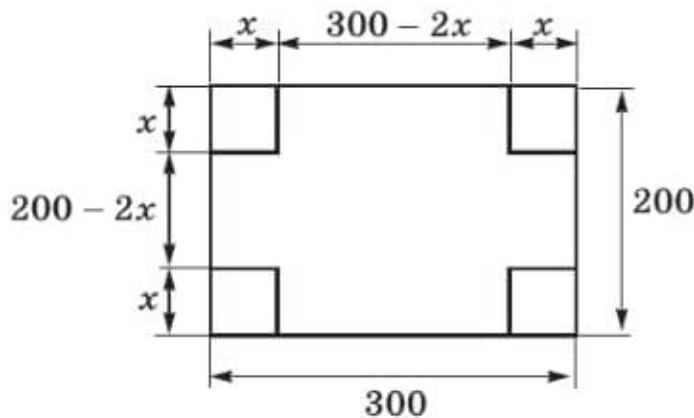


Рис. 3.16. Чертёж коробки

Величина x может принимать множество значений, для которых справедливо неравенство

$$0 < x < 100.$$

При разных значениях величины x будут получаться и разные значения величины V . Требуется выяснить, при каком значении x величина V будет иметь наибольшее значение.

С помощью табличного процессора Александр построил таблицу значений функции

$$V(x) = (300 - 2x) \cdot (200 - 2x) \cdot x$$

для значений аргумента x , изменяющихся от 0 до 100 с шагом 5 (рис. 3.17).

	A	B	C	D
1	Высота (x)	Длина (300 - 2x)	Ширина (200 - 2x)	Объём (V)
2	0	300	200	0
3	5	290	190	275500
4	10	280	180	504000
5	15	270	170	688500
6	20	260	160	832000
7	25	250	150	937500
8	30	240	140	1008000
9	35	230	130	1046500
10	40	220	120	1056000
11	45	210	110	1039500
12	50	200	100	1000000
13	55	190	90	940500
14	60	180	80	864000
15	65	170	70	773500
16	70	160	60	672000
17	75	150	50	562500
18	80	140	40	448000
19	85	130	30	331500
20	90	120	20	216000
21	95	110	10	104500
22	100	100	0	0

Рис. 3.17. Таблица значений функции $V(x)$

Можно проанализировать последний столбец полученной таблицы или же построенный по ней график (рис. 3.18). В любом случае видно, что наибольшее значение объёма V получается при $x = 40$.

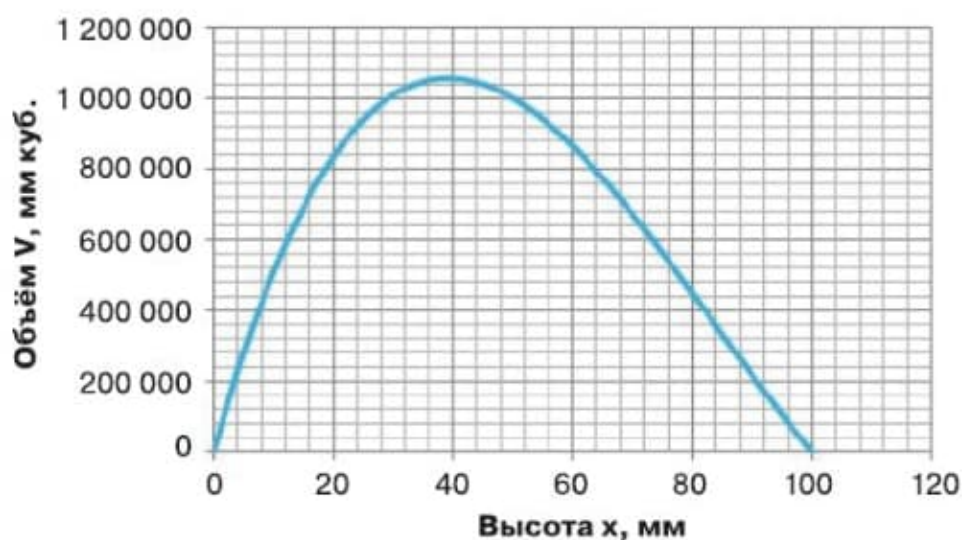


Рис. 3.18. График функции $V(x)$

В имеющемся в вашем распоряжении табличном процессоре воспроизведите заполнение таблицы значений функции

$V(x) = (300 - 2x) \cdot (200 - 2x) \cdot x$ и построение её графика.

Для того чтобы получить ещё более точное значение x , постройте график этой же функции для значений аргумента, изменяющихся от 30 до 50 с шагом 1.

Пример 4

Семья Михаила решила разводить карпов в небольшом пруду на своей даче. Отец Михаила — биолог, ему известен следующий закон: прирост какого-либо вида живых организмов за счёт рождаемости прямо пропорционален их количеству, а убыль за счёт смертности прямо пропорциональна квадрату их количества:

$$\Delta N = k \cdot N - q \cdot N^2.$$

Здесь:

ΔN — прирост;

N — исходное количество организмов;

k — коэффициент прироста;

q — коэффициент смертности.

Коэффициенты прироста и смертности зависят от вида живых организмов и условий, в которых они обитают. Отец Михаила в справочниках по рыбоводству нашёл информацию о том, что при разведении карпов в имеющихся условиях (климат, состояние пруда, наличие корма) надо учитывать следующие значения коэффициентов:

$$k = 1,2; \quad q = 0,001.$$

Таким образом, если первоначально в пруд запустить N_0 рыб, то через год их количество будет таким:

$$N_1 = N_0 + (1,2 \cdot N_0 - 0,001 \cdot N_0^2).$$

Михаил организовал по этой формуле вычисления в табличном процессоре (рис. 3.19) и начал проводить численный эксперимент: подсчитывать, сколько карпов будет в пруду через один, два, три года и т. д. при разном количестве первоначально запущенных в пруд карпов (10, 20, 100 и др.). Эти данные он вводил в ячейку B2.

	A	B	C	D
1		N	k	q
2	Год		1,2	0,001
3	1	=ЦЕЛОЕ(B2+\$C\$2*B2-\$D\$2*B2^2)		
4	2			

Рис. 3.19. Таблица для численного эксперимента

Результаты численного эксперимента несколько смутили Михаила, так как по прошествии некоторого числа лет прирост рыбы прекращался.

В имеющемся в вашем распоряжении табличном процессоре воспроизведите описанный выше численный эксперимент. Постройте график изменения количества карпов в пруду с течением времени.

Сколько карпов вы бы порекомендовали первоначально запустить в пруд? На каком году разведения карпов можно заняться их отловом, чтобы это не сказалось на приросте рыбы? Что надо учесть в модели, чтобы рассчитать возможную экономическую выгоду от разведения карпов?

САМОЕ ГЛАВНОЕ

Анализ данных — процесс обработки информации после её получения и сбора, направленный на выявление закономерностей и достижение ясности в исследуемой ситуации.

С помощью сортировки данные можно расположить по возрастанию или по убыванию содержимого ячеек.

Поиск данных в электронных таблицах осуществляется с помощью фильтров, которые «не пропускают» на экран записи, не удовлетворяющие условиям поиска.

Диаграмма — средство наглядного графического представления количественных данных. Диаграммы помогают анализировать данные, проводить их сравнение и выявлять закономерности, скрытые в последовательностях чисел.

Электронные таблицы позволяют создавать диаграммы нескольких типов, основными из которых являются: график, круговая диаграмма и гистограмма.

Моделирование поведения объекта путём получения численного решения уравнений математической модели называется численным моделированием. Численное моделирование удобно проводить в электронных таблицах.



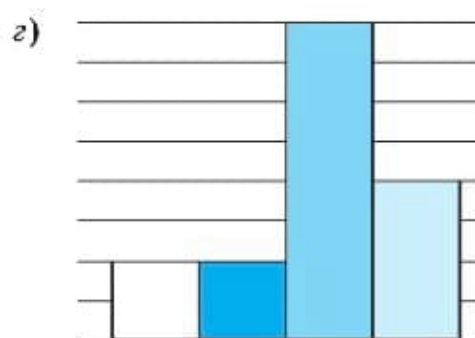
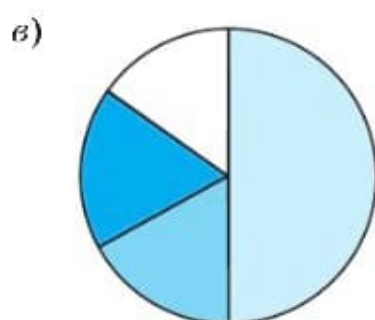
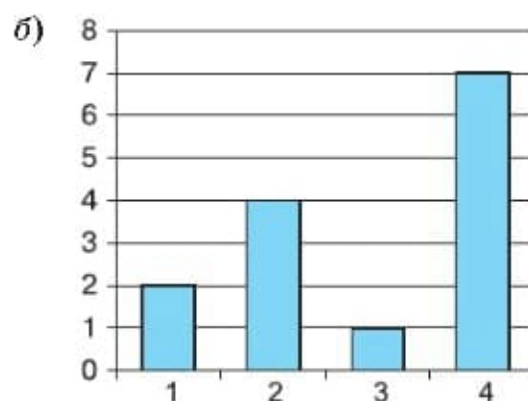
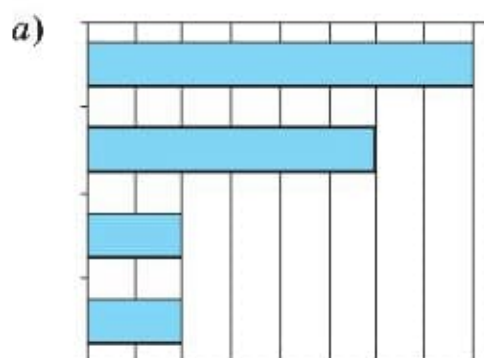
Вопросы и задания

1. Перечислите основные способы выполнения сортировки, реализованные в электронных таблицах.
2. В чём заключается различие между сортировкой данных в столбцах электронной таблицы и сортировкой данных в базе данных?
3. Как осуществляется поиск информации в электронной таблице?
4. Каким образом сортировка и поиск, реализованные в электронных таблицах, способствуют анализу данных?
5. Предложите вариант ответа на вопросы из примера 7 (пункт 3.2.4) с помощью сортировки данных и фильтрации.
6. Что такое диаграмма? Каково назначение диаграмм?
7. Назовите основные типы диаграмм.
8. Обоснуйте выбор типа диаграммы для представления:
 - а) результатов контрольной работы по алгебре в вашем классе;
 - б) результатов контрольной работы по математике в 9 «А» и 9 «Б» классах;
 - в) динамики изменения температуры воздуха в течение месяца;
 - г) площадей водной поверхности крупнейших озёр нашей страны;
 - д) доли федеральных округов Российской Федерации в общем объёме промышленного производства.
9. Как вы понимаете смысл фразы «Диаграммы в электронных таблицах сохраняют свою зависимость от данных, на основе которых они построены»?
10. Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D
1	3	=A1+1		
2	=B1-A1	=A1-A2*2	=A1+B2	=B1/2

После выполнения вычислений по значениям ячеек диапазона A2:D2 была построена диаграмма. Укажите получившуюся диаграмму.





11. В зимней спартакиаде принимают участие лыжники (Л), биатлонисты (Б), конькобежцы (К) и хоккеисты (Х). Спортсмены имеют разный уровень мастерства: каждый имеет либо II, либо I разряд или является мастером спорта (М). На диаграмме 1 представлено количество спортсменов по видам спорта, а на диаграмме 2 — соотношение спортсменов с различным уровнем мастерства.

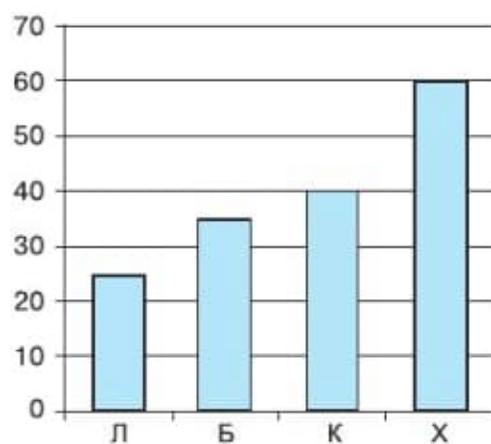


Диаграмма 1

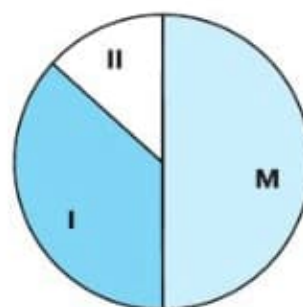


Диаграмма 2

Какое из следующих утверждений истинно?

- а) Все спортсмены, имеющие I разряд, могут являться конькобежцами.
- б) Все мастера спорта могут быть хоккеистами.
- в) Все биатлонисты могут иметь II разряд.
- г) Все спортсмены, имеющие I разряд, могут являться хоккеистами.

12. Всегда ли представление информации в виде диаграмм предпочтительнее представления информации в табличной форме? Приведите пример.
13. В чём суть численного моделирования? Каковы его преимущества по сравнению с использованием других вариантов моделей? Обсудите этот вопрос в группе.



Задания для практических работ

Внимание! Для выполнения практических заданий используйте имеющийся в вашем распоряжении табличный процессор. Все выполненные задания сохраняйте на разных листах одной книги, хранящейся в вашей личной папке.

1. Представьте в электронной таблице свои расходы за неделю; для заполнения диапазонов ячеек В7:І7, І3:І6 используйте формулы. Образец структуры и оформления таблицы:

	А	В	С	Д	Е	F	G	Н	І
1	Расходы за неделю								
2		<i>Понедельник</i>	<i>Вторник</i>	<i>Среда</i>	<i>Четверг</i>	<i>Пятница</i>	<i>Суббота</i>	<i>Воскресенье</i>	Всего:
3	<i>Проезд</i>	26,0р.	26,0р.	26,0р.	26,0р.	26,0р.	26,0р.	26,0р.	182,0р.
4	<i>Буфет</i>	20,0р.	15,0р.	25,0р.	20,0р.	30,0р.			110,0р.
5	<i>Канцтовары</i>	35,0р.		20,0р.		12,0р.			67,0р.
6	<i>Развлечения</i>						70,0р.	100,0р.	170,0р.
7	Итого:	81,0р.	41,0р.	71,0р.	46,0р.	68,0р.	96,0р.	126,0р.	529,0р.

2. Оформите лист для представления ёмкости диска в разных единицах:

	А	В
1	Ёмкость диска в битах	
2	Ёмкость диска в байтах	
3	Ёмкость диска в килобайтах	
4	Ёмкость диска в мегабайтах	
5	Ёмкость диска в гигабайтах	

Узнайте ёмкость в байтах жёсткого диска имеющегося в вашем распоряжении компьютера, запишите её в соответствующую ячейку диапазона В1:В5. Получите с помощью формул в других ячейках диапазона ёмкость жёсткого диска в единицах измерения, указанных в соответствующих ячейках столбца А.

3. Составьте таблицу умножения на число n ($1 \leq n \leq 9$). Значение n задаётся в ячейке B2.

	А	В
1	Таблица умножения	
2	на число	
3	Множитель	Произведение
4	1	
5	2	
6	3	
7	4	
8	5	
9	6	
10	7	
11	8	
12	9	

4. Составьте таблицу умножения чисел первого десятка. Используйте смешанные ссылки.
5. Подготовьте следующую таблицу, заполнив диапазон ячеек B1:B20 с помощью автозаполнения.

	А	В
1		1
2		2
3		3
4		4
5		5
6		6
		...
19		19
20		20
21	Сумма:	

Выполните следующие расчёты:

- получите сумму всех целых чисел от 1 до 20 в ячейке B21;
- получите в диапазоне ячеек C1:C20 квадраты соответствующих чисел из столбца В и сумму квадратов в ячейке C21;
- получите в диапазоне ячеек D1:D20 первые 20 чётных чисел и их сумму в ячейке D21.

Проверьте, такие ли результаты получились в ячейках диапазона B21:D21:

	А	В	С	Д
21	Сумма	210	2870	420

6. В основу эффективного решения головоломки «Ханойская башня» положен алгоритм, суть которого сводится к следующему: для перемещения башни, состоящей из n колец, с первого стержня на третий мы должны решить чуть более простую задачу — переместить на второй стержень башню, состоящую из $n - 1$ кольца. После этого нижний диск с первого стержня перемещается на третий и повторно осуществляется перемещение башни из $n - 1$ кольца, но уже со второго диска на третий. Таким образом, число ходов, необходимых для перемещения башни из n колец, равно удвоенному числу ходов, необходимых для перемещения башни из $n - 1$ кольца, и ещё одному ходу. Используйте эту закономерность для вычисления числа ходов, необходимых для перемещения башни из 64 колец. Вычислите, сколько времени займёт такое перемещение, если считать, что на один ход требуется 1 секунда.
7. Как известно, игра в шахматы была придумана в Индии. Согласно старинной легенде, индусский царь, восхищённый игрой, решил щедро одарить её изобретателя. Но тот, по мнению царя, запросил ничтожную награду: он просил выдать одно пшеничное зерно за первую клетку шахматной доски, а за каждую следующую клетку (всего их 64) — вдвое больше против предыдущей. Рассчитайте, сколько всего пшеничных зёрен должен был получить изобретатель. Какими могли бы быть размеры амбара для размещения этого зерна, если кубический метр пшеницы содержит около 15 миллионов зёрен?
8. Известно количество учеников во всех классах начальной школы:

Класс	Параллель		
	А	Б	В
1	23	19	27
2	25	26	18
3	20	24	21
4	21	18	22

Определите, на сколько число учеников в самом многочисленном классе превышает число учеников в самом малочисленном классе. Вычислите среднюю наполняемость классов.

9. Постройте таблицу истинности для логического выражения НЕ (A ИЛИ B) .
10. Известно количество учеников в каждом из классов начальной школы. Класс, в котором более 25 учеников, считается переполненным. Используя данные практического задания 8, создайте таблицу следующего вида:

	A	B	C
1	Начальная школа		
2	Класс	Число учеников	Комментарий
3	1А		
4	1Б		
5	1В		
6	2А		

В столбце С дайте комментарии («переполнен», «соответствует норме») по наполняемости каждого класса. Подсчитайте, сколько классов переполнено и сколько имеет наполняемость, соответствующую норме.

11. На основании данных, содержащихся в файле `Тестирование.xls` (см. пример 7 из пункта 3.2.4), найдите ответы на следующие вопросы.
- 1) Сколько всего учащихся набрали больше 84 баллов по математике?
 - 2) Сколько учащихся набрали более 50 баллов по каждому из трёх предметов?
 - 3) Сколько учащихся Центрального района набрали более 50 баллов по каждому из предметов?
 - 4) Сколько учащихся Центрального района набрали более 210 баллов в сумме по трём предметам?
 - 5) Чему равна разница между максимальным и минимальным баллами по информатике среди всех учащихся?
 - 6) Чему равна наименьшая сумма баллов по предметам у учеников Подгорного района?

12. Постройте столбчатую и круговую диаграммы своих расходов за неделю. Воспользуйтесь таблицей, полученной при выполнении практического задания 1.

Образец выполнения задания:



Столбчатая диаграмма «Расходы по статьям» строится по данным несмежных диапазонов ячеек A3:A6, I3:I6. Круговая диаграмма «Расходы по дням недели» строится по данным несмежных диапазонов ячеек B2:H2, B7:H7.

Внимание! Для выделения несмежных диапазонов ячеек выделите первый диапазон ячеек и, удерживая нажатой клавишу *Ctrl*, выделите второй диапазон ячеек.

13. По данным, приведённым в файле `Тестирование.xls` (см. пример 7 из пункта 3.2.4), постройте круговую диаграмму, отображающую соотношение числа участников тестирования из Майского, Кировского, Центрального, Подгорного и Заречного районов.
14. Постройте графики (для *Microsoft Office* тип диаграммы — **Точечная**, для *OpenOffice.org* — **Линии**) следующих функций:
- $y = |x|$ для значений аргумента, изменяющихся от -10 до 10 с шагом 1 ;
 - $y = 2x^2 + 5x - 10$ для значений аргумента, изменяющихся от -5 до $2,5$ с шагом $0,5$;
 - $y = x^2 - 2|x| - 3$ для значений аргумента, изменяющихся от $-3,5$ до $3,5$ с шагом $0,5$.

Внимание! Для построения графика функции предварительно следует создать таблицу значений функции, в которую занести значения аргумента функции и значения функции при заданных значениях аргумента.

Тестовые задания для самоконтроля

1. Рабочая книга табличного процессора состоит из:
 - а) ячеек
 - б) строк
 - в) столбцов
 - г) листов
2. Обозначением строки в электронной таблице является:
 - а) 18D
 - б) K13
 - в) 34
 - г) AB
3. Строка формул используется в табличном процессоре для отображения:
 - а) только адреса текущей строки
 - б) только адреса текущей ячейки
 - в) только содержимого текущей ячейки
 - г) адреса и содержимого текущей ячейки
4. Ввод формул в таблицу начинается со знака:
 - а) \$
 - б) f
 - в) =
 - г) @
5. Ровно 20 ячеек электронной таблицы содержатся в диапазоне:
 - а) E2:F12
 - б) C2:D11
 - в) C3:F8
 - г) A10:D15
6. В электронной таблице выделили группу из четырёх соседних ячеек. Это может быть диапазон:
 - а) A1:B4
 - б) A1:C2
 - в) A1:B2
 - г) B2:C2

7. Среди приведённых записей формулой для электронной таблицы является:
- а) $A2+D4B3$
 - б) $=A2+D4*B3$
 - в) $A1=A2+D4*B3$
 - г) $A2+D4*B3$
8. В ячейки A3, A4, B3, B4 введены, соответственно, числа 7, 4, 6, 3. Какое число будет находиться в ячейке C1 после введения в эту ячейку формулы $=СУММ(A3:B4)$?
- а) 4
 - б) 20
 - в) 14
 - г) 15
9. В электронной таблице при перемещении или копировании формул абсолютные ссылки:
- а) преобразуются независимо от нового положения формулы
 - б) преобразуются в зависимости от нового положения формулы
 - в) преобразуются в зависимости от наличия конкретных функций в формулах
 - г) не изменяются
10. Укажите ячейку, адрес которой является относительным:
- а) D30
 - б) E\$5
 - в) \$A\$2
 - г) \$C4
11. Укажите ячейку, в адресе которой не допускается изменение только имени строки:
- а) E\$1
 - б) H5
 - в) \$B\$6
 - г) AG14

12. Дан фрагмент электронной таблицы, содержащий числа и формулы:

	C	D	E
1	110	25	=C1+D1
2	45	55	
3	120	60	

Значение в ячейке E3 после копирования в неё формулы из ячейки E1 будет равно:

- а) 60
 - б) 180
 - в) 170
 - г) 135
13. Дан фрагмент электронной таблицы, содержащий числа и формулы:

	C	D	E
1	23	18	=C1+\$D\$1
2	45	24	

Значение в ячейке E2 после копирования в неё формулы из ячейки E1 будет равно:

- а) 63
 - б) 180
 - в) 170
 - г) 135
14. В ячейку E4 введена формула =C2*D2. Содержимое ячейки E4 скопировали в ячейку F7. Какая формула будет записана в ячейке F7?
- а) =D5*E5
 - б) =D7*E7
 - в) =C5*E5
 - г) =C7*E7
15. В ячейку B7 записана формула =\$A4+D\$3. Формулу скопировали в ячейку D7. Какая формула будет записана в ячейке D7?
- а) = \$C4+F\$3
 - б) =\$A4+F\$3
 - в) =\$A4+D\$3
 - г) =\$B4+D\$3

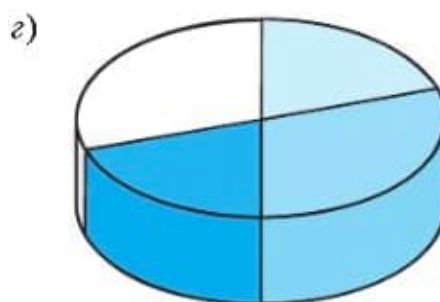
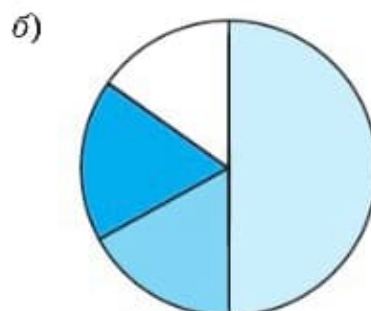
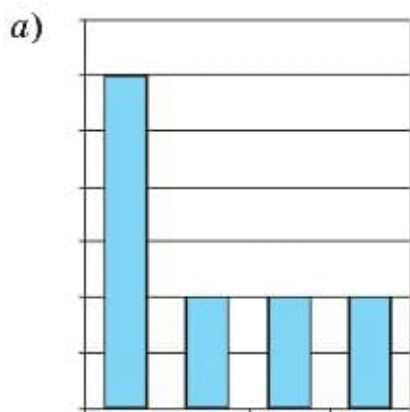
16. Уличный продавец газет получает 3 рубля за продажу каждой из первых 50 газет. За продажу каждой из последующих газет он получает 5 рублей. В ячейке C3 находится количество газет, проданных продавцом за день. Какая из формул позволяет подсчитать заработок продавца за день?

- а) =ЕСЛИ(C3<50;C3*3;C3*5-100)
 б) =ЕСЛИ(C3<=50;C3*3;150+C3*5)
 в) =ЕСЛИ(C3<=50;C3*3;150+(C3-50)*5)
 г) =ЕСЛИ(C3=50;150;C3*5)

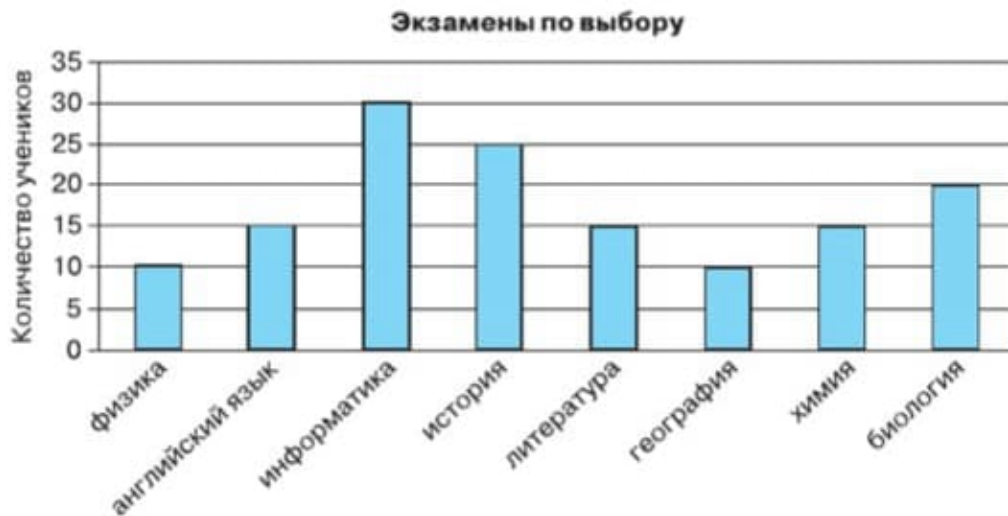
17. Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D
1	3	2	3	2
2	=(C1+A1)/2	=A1-B1	=C1-D1	=A1-2

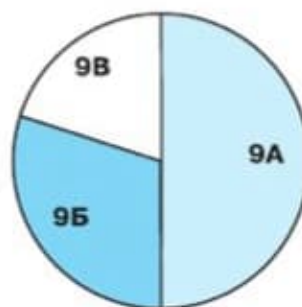
После выполнения вычислений по значениям ячеек диапазона A2:D2 было построено несколько диаграмм. Укажите диаграмму, которая не могла быть получена.



18. Для наглядного представления площадей крупнейших государств мира целесообразно использовать:
- круговую диаграмму
 - график
 - столбчатую диаграмму
 - ярусную диаграмму
19. Для наглядного представления изменения температуры воздуха в течение месяца следует использовать:
- круговую диаграмму
 - график
 - столбчатую диаграмму
 - ярусную диаграмму
20. Кроме обязательных экзаменов по русскому языку и математике, каждый из учеников 9-х классов выбрал для итоговой аттестации ещё два предмета. На диаграммах отражено количество учеников, выбравших тот или иной предмет, и соотношение численности учеников в 9-х классах.



Соотношение численности учащихся 9-х классов



Какое из следующих утверждений истинно?

- а) Все ученики 9 «А» класса могли выбрать экзамен по информатике.
- б) Все ученики 9 «Б» класса сдают по выбору только химию и биологию.
- в) Все ученики, выбравшие физику, могут учиться в 9 «В» классе.
- г) Историю могли выбрать только ученики 9 «Б» класса.

Глава 4

ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ В СОВРЕМЕННОМ ОБЩЕСТВЕ

Жизнь современного человека невозможно представить без использования компьютера, смартфона, других мобильных устройств, многочисленных и разнообразных информационных ресурсов и сервисов сети Интернет (мессенджеров, социальных сетей, электронной почты и т. д.) — всего того, что принято называть информационно-коммуникационными технологиями.



Технология — это совокупность методов и инструментов для достижения желаемого результата; **информационно-коммуникационные технологии (ИКТ)** — это технологии, осуществляемые с применением компьютеров и средств телекоммуникации для сбора, обработки, хранения, защиты и распространения информации.

§ 4.1

Локальные и глобальные компьютерные сети

Ключевые слова:

- канал связи
- компьютерная сеть
- скорость передачи данных
- локальная сеть
- глобальная сеть
- Интернет
- протокол
- IP-адрес

4.1.1. Передача информации

Мы с вами достаточно подробно рассмотрели применение компьютеров для решения ряда информационных задач; средства телекоммуникации вам также знакомы — это Интернет, мобильная связь, спутниковые системы связи, цифровое и аналоговое телевидение, телефонная связь и другие средства передачи информации на большие расстояния.

Термин «телекоммуникации» образован слиянием двух слов: греческого *tele* (далеко, вдаль) и латинского *communicatio* (общение); он означает «связь, общение на расстоянии».

Ранее мы уже говорили о том, что передача информации — один из важнейших информационных процессов. Информация передаётся от источника к приёмнику в форме некоторой последовательности сигналов, символов, знаков. Например, при непосредственном разговоре между людьми происходит передача звуковых сигналов — речи; при чтении текста человек воспринимает графические символы — буквы. Передаваемая последовательность сигналов, символов, знаков называется **сообщением**.

Телеграф, радио, телефон, телетайп, телефакс — вот технические средства передачи информации, являющиеся неотъемлемой частью истории развития нашего общества. Найдите информацию об этих средствах и подготовьте о них небольшое сообщение.

Канал связи (передачи информации) — это система технических средств и среда распространения сигналов для передачи сообщений от источника к приёмнику. При непосредственном общении людей информация передаётся с помощью звуковых волн, при разговоре по телефону — с помощью акустических и электрических сигналов, распространяемых по линиям связи, при чтении — с помощью световых волн.

Любое преобразование информации, идущей от источника, в форму, пригодную для её передачи по каналу связи, называется **кодированием**. В настоящее время широко используется цифровая связь, когда передаваемая информация преобразуется в двоичный код.





Недостаточное техническое качество каналов связи и некоторые другие причины могут приводить к искажению передаваемого сигнала и потере информации. Во избежание таких ситуаций передаваемый по линии связи код делают избыточным. За счёт этого потеря какой-то части информации при передаче может быть компенсирована. Кроме того, в современных системах цифровой связи все сообщения разбиваются на части (пакеты, блоки). Для каждого блока вычисляется контрольная сумма (сумма двоичных цифр), которая передаётся вместе с блоком. В месте приёма заново вычисляется контрольная сумма принятого блока, и если она не совпадает с первоначальной суммой, то передача блока повторяется.

На протяжении столетий для передачи писем человечество пользовалось услугами почтовой связи; во второй половине XIX века была изобретена технология передачи звука (телефон); с 30-х годов XX века для передачи изображений стал использоваться телефакс. В наши дни для передачи текстов, изображений, звука и многих других видов информации повсеместно используются **компьютерные сети** — два и более компьютеров, соединённых линиями передачи информации. С появлением компьютерных сетей стало возможным отправить письмо, которое доходит быстрее, чем телеграмма, получить ответ, узнать последние новости, поговорить с другом, сидящим у компьютера за сотни километров, так, будто он находится в соседней комнате, заказать билет на самолёт или номер в гостинице, скачать нужную программу, мелодию или фильм.

Важной характеристикой компьютерной сети является **скорость передачи данных**.



Скорость передачи данных — это объём данных, переданных по каналу связи за единицу времени (например, за 1 с).

Основная единица измерения скорости передачи данных — бит в секунду (бит/с).

Производные единицы — килобиты в секунду (Кбит/с), мегабиты в секунду (Мбит/с), гигабиты в секунду (Гбит/с). При этом приставки «кило», «мега», «гига» являются десятичными:

$$1 \text{ Кбит/с} = 1000 \text{ бит/с} = 10^3 \text{ бит/с};$$

$$1 \text{ Мбит/с} = 1000 \text{ Кбит/с} = 10^6 \text{ бит/с};$$

$$1 \text{ Гбит/с} = 1000 \text{ Мбит/с} = 10^9 \text{ бит/с}.$$

Различают локальные и глобальные компьютерные сети.



4.1.2. Что такое локальная компьютерная сеть

Локальная компьютерная сеть объединяет компьютеры, установленные в одном помещении (например, школьный компьютерный класс) или в одном здании (например, в локальную сеть могут быть объединены все компьютеры, находящиеся в здании школы). Локальная сеть позволяет пользователям получить совместный доступ к ресурсам компьютеров, а также к периферийным устройствам (принтерам, сканерам, дискам, модемам и др.), подключённым к сети.

Локальные сети бывают одноранговыми и с выделенным сервером.

В небольших локальных сетях все компьютеры равноправны, т. е. каждый из них может использовать ресурсы другого. Пользователи самостоятельно решают, какие ресурсы своего компьютера (файлы, папки, диски) сделать доступными для всей сети. Такие сети называются одноранговыми.

В сетях с большим количеством пользователей нежелательно, чтобы все они имели доступ ко всем компьютерам сети. При объединении более 10 компьютеров целесообразно выделять наиболее мощный компьютер — сервер (англ. *server* — обслуживающий). На жёстком диске сервера размещают файлы (данные и программы), к которым получают доступ другие компьютеры сети — клиенты. Кроме того, всем пользователям сети может быть доступно периферийное оборудование, подключённое к серверу (например, принтер или сканер).

Каждый компьютер, подключаемый к локальной сети, должен иметь специальную плату — сетевой адаптер. Её функция — передача и приём сигналов, распространяемых по каналам связи.

Соединение компьютеров (их сетевых плат) в локальную сеть осуществляется с помощью различных типов кабелей (витая пара, оптическое волокно — рис. 4.1) или по беспроводным каналам (типа Wi-Fi).



Рис. 4.1. Кабели:
витая пара
и оптоволокно



Витая пара представляет собой два изолированных медных провода, скрученных один относительно другого. Такое скручивание проводов снижает влияние помех на сигналы, передаваемые по этому кабелю. Соединение «витая пара» представляет собой несколько витых пар (2 или 4), покрытых пластиковой оболочкой. Скорость передачи данных — от 10 Мбит/с до 1 Гбит/с.

Оптоволоконный кабель передаёт свет по стеклянному волокну. Такой тип соединения обеспечивает очень высокую скорость передачи, протяжённость канала составляет сотни и тысячи километров, и он абсолютно не подвержен электромагнитным помехам. Скорость передачи данных — от 100 Мбит/с до 100 Гбит/с.

Беспроводное соединение Wi-Fi обеспечивает скорость передачи данных до 600 Мбит/с.

Задача

Скорость передачи данных по сети Wi-Fi равна 80 Мбит/с. Сколько времени потребуется на передачу файла размером 250 Мбайт?

Решение

$$\begin{array}{l|l|l}
 v = 80 \text{ Мбит/с} & v = 80 \text{ Мбит/с} = 80 \cdot 10^6 \text{ бит/с} & I = v \cdot t, \\
 I = 250 \text{ Мбайт} & I = 250 \text{ Мбайт} = 250 \cdot 2^{23} \text{ бит} & t = \frac{I}{v} \\
 t = ? & &
 \end{array}$$

$$t = \frac{250 \cdot 2^{23}}{80 \cdot 10^6} \approx 25 \text{ (с)}$$

Ответ: 25 с.

4.1.3. Что такое глобальная компьютерная сеть

Локальные сети, объединяя десятки компьютеров на небольшой территории, не обеспечивают совместный доступ к информации пользователям, находящимся на значительном расстоянии друг от друга (например, в различных населённых пунктах).



Глобальная компьютерная сеть — это система связанных между собой компьютеров, расположенных на сколько угодно большом удалении друг от друга (например, в разных странах и на разных континентах).

Примерами глобальных компьютерных сетей могут служить региональные и корпоративные сети. Региональные компьютерные сети обеспечивают объединение компьютеров в пределах одного региона (города, области, края, страны). Корпоративные компьютер-

ные сети создаются для обеспечения деятельности различного рода корпоративных структур, имеющих территориально удалённые подразделения (например, банков со своими филиалами).

Основой любой глобальной компьютерной сети являются компьютерные узлы и каналы связи. Узел — это мощный компьютер, постоянно подключённый к сети. К узлам компьютерной сети подключаются абоненты — персональные компьютеры пользователей или локальные сети.

Для передачи данных в глобальных сетях применяют самые разнообразные физические каналы: электрический кабель; радиосвязь через ретрансляторы и спутники связи; инфракрасные лучи (как в телевизионных пультах дистанционного управления); оптоволоконный кабель; обычную телефонную сеть.

Наиболее известной и самой обширной глобальной компьютерной сетью является Интернет.

Интернет (англ. *Internet*, от лат. *inter* — между и англ. *net* — сеть) — всемирная компьютерная сеть, объединяющая множество локальных, региональных и корпоративных сетей. Каждая входящая в Интернет сеть имеет свой собственный эксплуатационный центр, который отвечает за работу данного регионального участка Интернета. У каждой из этих сетей может быть владелец, но Интернет в целом не принадлежит никому. Координирует развитие Всемирной сети общественная организация Сообщество Интернета (**ISOC** — *Internet Society*).

Первая компьютерная сеть, которая соединила компьютеры, находившиеся на большом расстоянии друг от друга (буквально в разных концах страны), была впервые запущена в 1969 году. Она называлась ARPAnet.

Выясните, в какой стране это было. Какие задачи стояли перед разработчиками сети?

Надёжность функционирования сети Интернет обеспечивается наличием большого количества каналов передачи информации между входящими в неё локальными, региональными и корпоративными сетями.

Для того чтобы подключить к сети Интернет свой домашний компьютер, необходимо воспользоваться услугами интернет-провайдера. При каждом выходе пользователя в Интернет его компьютер соединяется с компьютерной системой провайдера.

Интернет соединяет различные модели компьютеров, с разным программным обеспечением. Это возможно благодаря реализации



в программном обеспечении особых соглашений (правил), называемых **протоколами**.

www

С помощью веб-сервиса «Яндекс.Интернетометр» (<http://gotourl.ru/15877>) узнайте скорость интернет-соединения для имеющегося в вашем распоряжении стационарного компьютера или мобильного устройства.

На то, как быстро открываются сайты и скачиваются файлы, влияет скорость входящего соединения; на быстроту передачи данных с вашего устройства в Сеть влияет скорость исходящего соединения.

?

Вычислите примерное время:

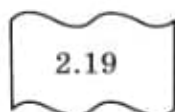
- 1) скачивания из Сети мультимедийной презентации размером 16 Мбайт;
- 2) скачивания из Сети фильма размером 1,5 Гбайт;
- 3) загрузки в облако 32 фотографий, каждая размером 5 Мбайт.

4.1.4. IP-адрес компьютера

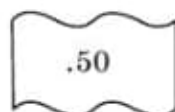
Каждый компьютер, подключённый к Интернету, получает свой уникальный 32-битовый идентификатор, называемый **IP-адресом**. Таких адресов более 4 миллиардов ($2^{32} - 1 = 4\,294\,967\,295$). Человеку, в отличие от технических систем, сложно работать с длинными цепочками из нулей и единиц. Поэтому вместо 32-битового представления мы используем запись IP-адреса в виде четырёх десятичных чисел (от 0 до 255), разделённых точками, например 204.152.190.71. Для осуществления такого перехода 32-битовая запись разбивается на четыре части (по 8 бит), каждая из которых как 8-разрядное двоичное число переводится в десятичную систему счисления.

Задача

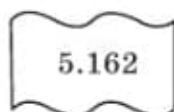
Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Требуется восстановить IP-адрес.



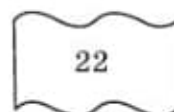
А



Б



В



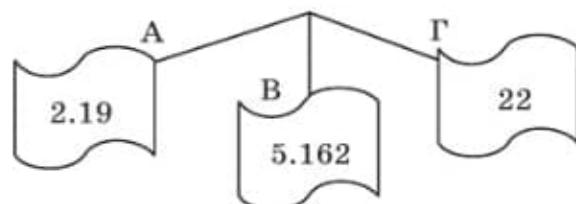
Г

Решение

Исследуем возможные комбинации фрагментов адреса с учётом того, что каждое из четырёх чисел в IP-адресе не должно превышать 255.

Так как адрес не может начинаться с точки, в качестве первого фрагмента совершенно точно нельзя использовать фрагмент Б.

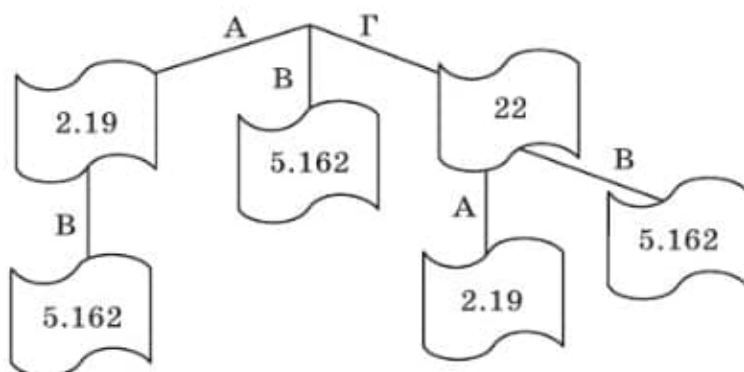
Получаем возможные варианты:



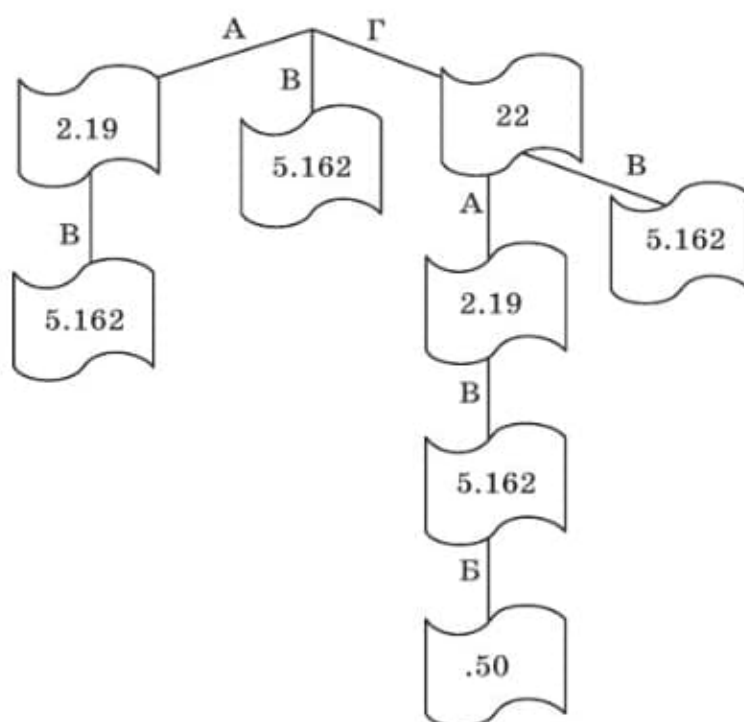
Фрагмент Б не может находиться на втором месте, так как он заканчивается на 50 и добавление к нему справа первой цифры любого из оставшихся фрагментов приведёт к образованию числа, превышающего 255.

Если в качестве первого взят фрагмент А, то после него совершенно точно не может следовать фрагмент Г (в противном случае получается число $1922 > 255$). Если в качестве первого взят фрагмент В, то после него не может следовать ни один из оставшихся фрагментов. После фрагмента Г может следовать любой из фрагментов А и В.

Получаем возможные варианты:



После фрагментов А–В мог бы следовать только фрагмент Б, но в рассматриваемом примере он не может быть третьим (по той же причине, что и вторым). По этой же причине после фрагментов Г–А может следовать только фрагмент В (фрагмент Б мы исключаем из рассмотрения). После фрагментов Г–В не могут следовать ни А, ни Б.



Таким образом, существует единственный способ соединения имеющихся фрагментов: Г–А–В–Б. Соответствующий адрес имеет вид 222.195.162.50

Как правило, при каждом выходе в Интернет ваш компьютер получает новый IP-адрес. Информация о том, когда и какие IP-адреса присваивались вашему компьютеру, сохраняется у провайдера.

www

Чтобы узнать свой текущий IP-адрес во время интернет-сеанса, достаточно воспользоваться уже известным вам веб-сервисом «Яндекс.Интернетометр».

Наряду с цифровыми IP-адресами в Интернете действует более удобная и понятная для пользователей **доменная система имён (DNS — Domain Name System)**, благодаря которой компьютеры получают уникальные символьные адреса.

Доменная система имеет иерархическую структуру. Полное доменное имя состоит из непосредственного имени домена и далее имён всех доменов, в которые он входит, разделённых точками.

Каждая страна имеет свой домен (иногда — несколько доменов) верхнего уровня. Например, нашей стране принадлежат домены ru, рф, su.

Пример

Полное имя `shop.prosv.ru` обозначает домен третьего уровня `shop`, входящий в домен второго уровня `prosv`, принадлежащий домену верхнего уровня `ru`.

Чтобы узнать IP-адрес заинтересовавшего вас веб-сайта, достаточно выполнить команды **Пуск** → **Все программы** → **Стандартные** → **Командная строка** и в появившемся окне **Командная строка** ввести команду `ping` и доменное имя сайта. Например, набрав `ping prlib.ru`, вы получите IP-адрес Президентской библиотеки им. Б. Н. Ельцина — общегосударственного электронного хранилища цифровых копий важнейших документов по истории, теории и практике российской государственности.



4.1.5. Протоколы передачи данных

Для того чтобы передача данных от одного компьютера к другому не занимала сеть надолго, файлы по сети передаются небольшими порциями — **пакетами**.

Передаваемые пакеты постепенно добираются до своего адресата, попадая с одного сервера на другой, причём на каждом сервере производится операция **маршрутизации**, т. е. определение адреса следующего сервера, наиболее близкого к получателю, на который можно переслать этот пакет (рис. 4.2). Маршрутизацию пакетов позволяет осуществлять **протокол IP**.



Рис. 4.2. Пакетная передача данных

Так как пакеты передаются независимо друг от друга, каждый пакет может дойти до адресата по своему пути. На конечном пункте все пакеты собираются в один файл. Если какого-либо пакета не хватает, компьютер-адресат посылает запрос на компьютер-отправитель с сообщением, какой пакет отсутствует.

Нужный пакет заново посылается адресату. Установление надёжной передачи сетевых пакетов между двумя компьютерами обеспечивает **протокол ТСР**.

Таким образом, все сети, которые подключаются к Интернету, используют для соединения протоколы:

- **ТСР** (*Transmission Control Protocol*) — **транспортный протокол**;
- IP** (*Internet Protocol*) — **протокол маршрутизации**.

Как правило, эти протоколы используются вместе и практически неотделимы друг от друга. Поэтому для них используется термин **протокол ТСР/IP**.

Протокол ТСР/IP и доменная система имён, появившиеся в 1980-х годах, придали Интернету вид, близкий к современному. Это, а также появление на рынке персональных компьютеров послужило началом коммерческого использования Интернета.

САМОЕ ГЛАВНОЕ

Компьютерная сеть — это два и более компьютеров, соединённых линиями передачи информации.

Локальная компьютерная сеть объединяет компьютеры, установленные в одном помещении или в одном здании, и обеспечивает пользователям возможность совместного доступа к ресурсам компьютеров, а также к периферийным устройствам, подключённым к сети. Локальные сети бывают одноранговыми и с выделенным сервером.

Глобальная компьютерная сеть — это множество связанных между собой компьютеров, расположенных на сколько угодно большом удалении друг от друга (например, в разных странах и на разных континентах).

Интернет — всемирная компьютерная сеть, соединяющая вместе множество локальных, региональных и корпоративных сетей, в состав которых могут входить разные модели компьютеров. Это возможно благодаря реализации в программном обеспечении компьютеров особых соглашений (правил), называемых протоколами.

Каждый компьютер, подключённый к Интернету, имеет свой IP-адрес — уникальный 32-битовый идентификатор.

DNS — доменная система имён; благодаря ей компьютеры получают уникальные символьные адреса.

По сети файлы передаются небольшими порциями — пакетами. Маршрутизацию пакетов позволяет осуществлять протокол IP. Установление надёжной передачи сетевых пакетов между двумя компьютерами обеспечивает протокол TCP.

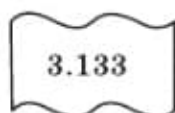
Вопросы и задания



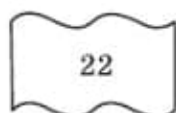
1. Что такое компьютерная сеть?
2. Что такое канал связи? Как определяется скорость передачи данных по каналу связи?
3. Как устроена одноранговая локальная сеть?
4. Как устроена локальная сеть с выделенным сервером?
5. Какого типа локальная сеть установлена в вашем компьютерном классе? Какие функции она выполняет?
6. Какие сети называются глобальными? Приведите примеры таких сетей.
7. Какие каналы связи используются для передачи данных в глобальных компьютерных сетях?
8. В локальной сети для соединения компьютеров используется кабель «витая пара», обеспечивая скорость передачи данных 100 Мбит/с. Прав ли Руслан, утверждающий, что на передачу по этой сети файла размером 500 Мбайт потребуется не более 30 секунд?
9. Что такое Интернет?
10. Благодаря чему в сети Интернет удаётся соединять различные модели компьютеров с разным программным обеспечением?
11. Для чего нужен IP-адрес?
12. Каким образом осуществляется переход от 32-битового IP-адреса к его записи в виде четырёх десятичных чисел?
13. Запишите в тетради 32-битовый IP-адрес в виде четырёх десятичных чисел, разделённых точками:
 - а) 11001100 10011000 10111110 01000111;
 - б) 11011110 11000011 10100010 00110010.



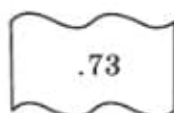
14. Запишите в тетради IP-адрес из четырёх десятичных чисел в 32-битовом виде:
- 210.171.30.128;
 - 10.55.0.225.
15. Почему следующие последовательности чисел не могут быть IP-адресами?
- 132.97.1212.14
 - 113.256.0.138
 - 123.132.321.231
16. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



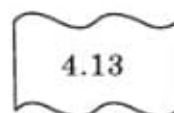
А



Б



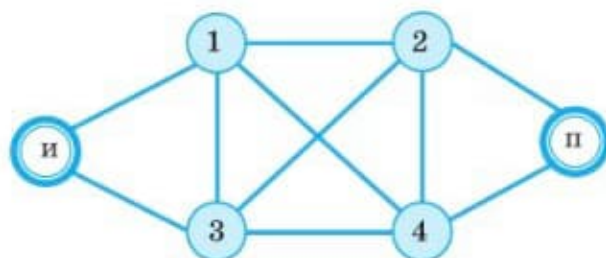
В



Г

17. Воспользуйтесь поиском информации в сети Интернет для того, чтобы:
- привести примеры доменов верхнего уровня, принадлежащих пяти европейским странам;
 - привести примеры доменов верхнего уровня, принадлежащих пяти странам, имеющим самую большую территорию;
 - привести примеры доменов верхнего уровня, принадлежащих пяти странам, имеющим самую большую численность населения;
 - пояснить, почему нашей стране принадлежит домен верхнего уровня su.
18. Опишите процесс маршрутизации и транспортировки данных по компьютерным сетям.

19. Рассмотрите все возможные маршруты доставки интернет-пакетов от сервера И (источник) к серверу П (приёмник) через серверы 1, 2, 3, 4 с учётом имеющейся архитектуры сети. Сколько таких маршрутов?



§ 4.2

Информационные ресурсы и сервисы Интернета

Ключевые слова:

- Всемирная паутина
- URL
- файловые архивы
- облачные хранилища
- электронная почта
- форум
- чат
- социальная сеть

Сеть Интернет привлекает пользователей своими **информационными ресурсами и сервисами (услугами)**, обеспечивая нам безграничные возможности для накопления и распространения информации, общения и сотрудничества.

4.2.1. Информационные ресурсы Интернета

Всемирная паутина, или **WWW**, — это сервис, с помощью которого пользователи Интернета получают доступ к информационным ресурсам, хранящимся на компьютерах в разных частях света. Основой WWW являются **веб-страницы** и **веб-сайты**, на которых информация представлена в виде гипертекстовых и гипермедийных документов.



1989 год — официальный год рождения WWW. Её создатель — выдающийся специалист в области информационных технологий Тим Бернерс-Ли.

Можно выделить несколько этапов в развитии Всемирной паутины.

Web 1.0. Преобладание статических сайтов, содержимое страниц которых было неизменным и предназначалось для просмотра и чтения; минимальное количество простых мультимедийных элементов, полное отсутствие видео.

Web 2.0. Бурное развитие динамических сайтов; их содержимое пополняется самими пользователями сайтов, которые получили возможность размещать собственные материалы, общаться между собой непосредственно на сайте. Страницы динамических сайтов не размещены на сервере в готовом виде, а «собираются» по запросу конкретного пользователя. Большое количество мультимедийной информации — графики, музыки, видео. В настоящее время такая форма представления контента в WWW является преобладающей. Серьёзный недостаток Web 2.0 — то, что пользователи размещают в Сети разную, в том числе некачественную информацию.

Web 3.0. Особое внимание уделяется качеству контента, наполнение сайта информацией контролируется профессионалами. Использование 3D-графики; анализ информации на сайтах с помощью искусственного интеллекта; Интернет вещей.

Web 4.0. Голосовое взаимодействие пользователя с WWW; пользователи взаимодействуют со Всемирной паутиной точно так же, как и между собой. Можно отследить все действия и данные пользователя. Контент с использованием виртуальной и дополненной реальности. Электронные устройства и приложения «чувствуют» потребности своих пользователей благодаря искусственному интеллекту. Перед человеком открываются новые возможности. Это будущее.

Вы уже неоднократно путешествовали по Всемирной паутине с помощью браузеров, осуществляя переходы по гиперссылкам; искали ответы на интересующие вас вопросы с помощью поисковых систем по ключевым словам.

Кроме ключевых слов, в запросах можно использовать символы логических операций. Так, символ | используется для обозначения логической операции ИЛИ (поиск по любому из слов); символ & используется для обозначения логической операции И (поиск по всем словам).

Пример

Рассмотрим следующие запросы:

- 1) Ахтуба | Ветлуга | Сура
- 2) Ахтуба & Ветлуга & Сура
- 3) (Ахтуба & Ветлуга) | Сура
- 4) (Ахтуба | Ветлуга) & Сура

Пусть A , B и C — множества страниц, которые будут найдены поисковым сервером соответственно по каждому из запросов Ахтуба, Ветлуга, Сура. Логической операции ИЛИ в запросе будет соответствовать операция объединения множеств; логической операции И — операция пересечения множеств.

Изобразим множества A , B и C графически с помощью кругов, расположенных следующим образом (рис. 4.3).

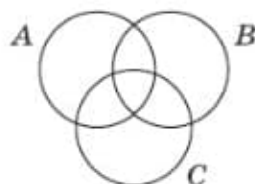


Рис. 4.3. Графическое изображение отношений между тремя множествами

На этой схеме удобно изображать множества, получаемые из множеств A , B и C (их называют базовыми) с помощью операций пересечения, объединения и дополнения.

Например, закрашенные области на рис. 4.4 — это объединение и пересечение двух базовых множеств A и B .

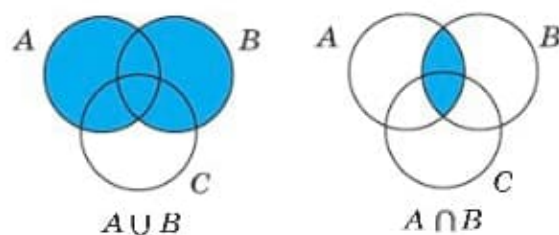


Рис. 4.4. Объединение и пересечение двух множеств

Множество $A \cup B$ — множество найденных поисковым сервером страниц, на которых присутствует хотя бы одно из ключевых слов Ахтуба, Ветлуга; множество $A \cap B$ — множество найденных поисковым сервером страниц, на которых одновременно присутствуют оба ключевых слова.

На рис. 4.5 представлены множества, полученные из трёх базовых множеств с помощью операций объединения и пересечения в соответствии с запросами 1–4.

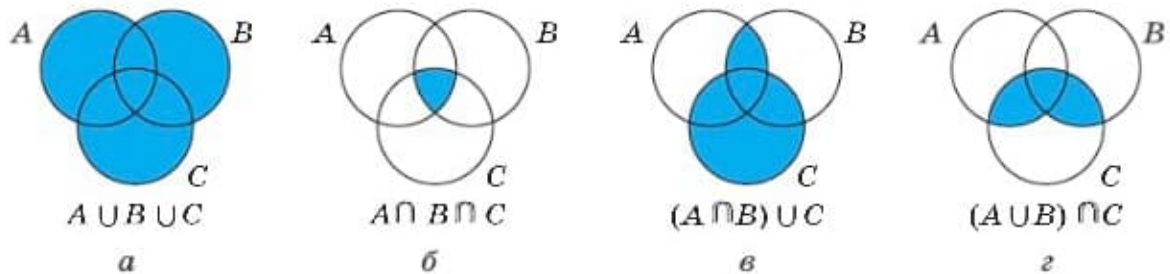


Рис. 4.5. Примеры отношений между тремя базовыми множествами

Пользуясь такими схемами (их ещё называют диаграммами Эйлера — Венна), можно дать оценку мощности (количества элементов) множеств, полученных из двух или трёх базовых множеств.

Например, мощность множества $A \cup B$ не меньше (больше или равна) мощности множества $A \cap B$ (см. рис. 4.4). Буквы *бгва* (см. рис. 4.5) задают порядок расположения выделенных множеств по неубыванию их мощности.



Расположите номера запросов в рассматриваемом примере в порядке невозрастания количества страниц, которые найдёт поисковый сервер по каждому запросу.

Использование множеств позволяет получать не только качественные (больше, меньше) но и количественные (сколько?) характеристики запросов. Так, для любых двух множеств A и B справедливо следующее утверждение.



Чтобы найти количество элементов объединения множеств A и B , надо сложить количество элементов множества A и количество элементов множества B и из суммы вычесть количество элементов пересечения множеств A и B :

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

Задача

В таблице приведены запросы к поисковому серверу и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тыс.)
Ахтуба	120
Ахтуба & Ветлуга	30
Ахтуба Ветлуга	280

Какое количество страниц (в тысячах) будет найдено по запросу Ветлуга? На каком количестве страниц рассматриваемого сегмента сети Интернет будет упоминаться Ахтуба, но не будет упоминаний о Ветлуге?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение

Пусть A и B — множества страниц, которые будут найдены поисковым сервером соответственно по запросам Ахтуба и Ветлуга.

Тогда $|A| = 120$, $|A \cap B| = 30$, $|A \cup B| = 280$.

Для ответа на первый вопрос найдём мощность множества B .

Используем соотношение $|A \cup B| = |A| + |B| - |A \cap B|$: подставим в него все известные данные и выполним простые арифметические преобразования:

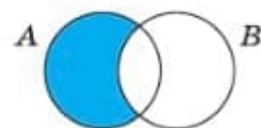
$$280 = 120 + |B| - 30,$$

$$280 = 90 + |B|,$$

$$|B| = 190.$$

Таким образом, по запросу Ветлуга будет найдено 190 тысяч страниц.

Для ответа на второй вопрос воспользуемся схематическим изображением множества страниц, где есть упоминание об Ахтубе или о Ветлуге.



Это множество состоит из трёх непересекающихся частей:

- левая часть схемы соответствует множеству страниц, где упоминается Ахтуба, но нет упоминаний о Ветлуге;
- центральная часть соответствует множеству страниц, где одновременно упоминаются Ахтуба и Ветлуга;
- правая часть схемы соответствует множеству страниц, где упоминается Ветлуга, но нет упоминаний об Ахтубе.

Нас интересует мощность левой части. Это не что иное, как \overline{B} — дополнение множества B до множества $A \cup B$.

$$|\overline{B}| = |A \cup B| - |B| = 280 - 190 = 90.$$

Таким образом, в рассматриваемом сегменте сети Интернет на 90 тысячах страниц будет упоминаться Ахтуба, но не будет упоминаний о Ветлуге.



Какова мощность дополнения множества A до множества $A \cup B$? О чём будет идти речь на страницах, образующих дополнение множества $A \cap B$ до множества $A \cup B$? Какова мощность этого множества?

Вам известна и ещё одна возможность поиска нужного документа в Интернете — это использование адреса документа (URL), состоящего из следующих частей:

- 1) название протокола со знаками `://` в конце названия;
- 2) доменное имя сервера со знаком `/` в конце имени;
- 3) полное имя файла на сервере, где файл находится.



Проанализируйте адрес документа

`https://bosova.ru/metodist/authors/informatika/3/eor9.php`, выделив в нём составные части. Вспомните, для чего предназначен протокол HTTPS.

В Интернете существует множество **файловых архивов** — своего рода библиотек, хранящих файлы с программным обеспечением, графикой, музыкой и другой информацией. Внешне файловый архив выглядит как перечень папок (каталогов) и файлов. Доступ пользователей к файлам в файловых архивах возможен по специальному протоколу передачи файлов **FTP** (*File Transfer Protocol*). FTP позволяет подключаться к серверам файловых архивов, просматривать содержимое каталогов и загружать файлы с сервера или на сервер. На тот случай, если передача файла прерывается по каким-либо причинам, протокол FTP предусматривает средства для докачки файла, что бывает очень удобно при передаче больших файлов.

Адрес файла на сервере файлового архива включает в себя протокол доступа к файлу, имя сервера и полное имя файла. Так, если доступ к файлу `htm.exe`, находящемуся на сервере `com.edu`, осуществляется по протоколу FTP, его адрес запишется следующим образом:

`ftp://com.edu/htm.exe`

Многие файлы, принимаемые по протоколу FTP, являются сжатыми, т. е. уменьшенными в размере. Сжатые файлы занимают меньше места на диске, быстрее передаются по Сети. Имена таких файлов обычно заканчиваются расширениями .zip, .arj, .rar и другими в зависимости от типа программы-архиватора, в котором они были созданы. После копирования сжатого файла его следует распаковать, т. е. восстановить его первоначальный вид.

Облачные сервисы. До появления и широкого распространения компьютерных сетей все программы, необходимые для работы, пользователи устанавливали на свои компьютеры; все данные хранились непосредственно на жёстком диске компьютера или на других переносных устройствах внешней памяти (дискетах, оптических дисках и др.). Выход из строя компьютера или переносного устройства внешней памяти мог привести к полной потере данных. Кроме того, в полном объёме доступ к необходимому программному обеспечению и данным был возможен только на рабочем месте пользователя. Эти и многие другие проблемы были решены с появлением облачных сервисов.

Облачные сервисы — это среды, обеспечивающие удобный сетевой доступ к компьютерному оборудованию, дисковой памяти, программному обеспечению и базам данных через Интернет в удалённом формате.

Каждый день мы сталкиваемся с необходимостью надёжного хранения различного контента (документов, фотографий, видео), возможность лёгкого доступа к которому — ещё одно важное требование современного ритма жизни.

Облачные хранилища информации обладают следующими *преимуществами*.

Удобство размещения и последующего поиска всей информации в одном месте. Чтобы не искать нужный материал (текстовый документ, фотографию, презентацию, видео и др.) на компьютере и многочисленных устройствах внешней памяти, все свои файлы можно разместить в едином облачном хранилище.

Надёжность. Не нужно делать резервные копии ценных файлов на внешних устройствах — облачные хранилища обеспечивают достаточно надёжное хранение информации, хотя и не гарантируют её полную защиту от несанкционированного доступа.

Если вы хотите хранить в облаке отсканированные копии важных документов, то предварительно поместите их в архив и защитите архив паролем.

Мобильность и доступность. Все файлы, сохранённые в облаке, доступны вам в любой географической точке с любого компьютерного устройства, подключённого к Интернету. Вы можете делиться своими материалами с другими пользователями, открывая им доступ по ссылке к соответствующим файлам, размещённым в облаке.

Возможность синхронизации данных. Многие облачные хранилища позволяют выполнять синхронизацию данных. Например, вы пользуетесь дома стационарным компьютером, а в школу берёте с собой ноутбук. Работая над рефератом, вы можете начать работу дома на стационарном компьютере, проконсультироваться с учителем и продолжить работу в школе на ноутбуке, вернуться к работе дома уже на стационарном компьютере и т. д. При этом каждый раз надо будет переносить файл реферата с одного устройства на другое, придумывать имена для файлов с разными вариантами реферата. Это требует дополнительного времени, к тому же можно запутаться в именах файлов и потерять какую-то часть информации. Если же вы используете облачное хранилище, то все изменения, вносимые в файл на любом устройстве, вносятся и в копию файла, хранящуюся в облаке. Как только вы захотите продолжить работу на другом устройстве, где хранится более ранняя версия реферата, имеющийся на устройстве файл обновится автоматически — вы продолжите работу с последней версией своего документа.

Воспользоваться всеми преимуществами облачных хранилищ пользователь может только в случае, если у него есть высокоскоростной канал доступа к Интернету.

Скорее всего, вы уже имеете опыт использования одного из следующих облачных хранилищ данных:

Облако Mail.ru (<https://cloud.mail.ru/>);

Яндекс.Диск (<https://disk.yandex.ru/>);

Google Диск (<https://drive.google.com/>).

Изучите доступную в сети Интернет информацию об этих облачных хранилищах.

Выясните, что у них общего и чем они различаются.

www



В облачных хранилищах размещают не только данные, но и программы. Например, *Google Документы* — это облачный пакет офисных приложений, включающий текстовый редактор, редактор презентаций и электронные таблицы. Работать с этими программами можно с любого компьютера, подключённого к Интернету. Онлайн-офис *Google Документы* — пример сервиса типа «программное обеспечение как услуга».

4.2.2. Коммуникационные сервисы Интернета

Интернет — важнейшее средство личной и социальной коммуникации. Личная коммуникация осуществляется с помощью электронной почты, а также ряда мессенджеров, обеспечивающих возможность мгновенного обмена сообщениями. Социальные сети являются основным инструментом социальной коммуникации.

Электронная почта (e-mail) — система обмена сообщениями (письмами) между абонентами компьютерных сетей. Это одна из первых служб Интернета, до сих пор не утратившая своего значения.

Любой пользователь Интернета может завести свой **почтовый ящик** — поименованную область дисковой памяти на **почтовом сервере** своего провайдера, куда будет помещаться входящая и исходящая корреспонденция. Пользователь, зарегистрировавший свой почтовый ящик, получает **адрес электронной почты**, который имеет вид

<имя_пользователя>@<имя_сервера>

Первая часть <имя_пользователя> выбирается самим пользователем; вторая часть <имя_сервера> жёстко связана с сервером, на котором пользователь зарегистрировал свой почтовый ящик.

Разделителем частей адреса служит символ @. Его официальное название «коммерческое эт», но в разговорном русском языке этот символ называют «собакой».

Есть у этого символа и другие названия: англичане, американцы, итальянцы и французы называют его «улиткой»; голландцы, поляки, немцы — «обезьяньим хвостом»; китайцы — «мышонком»; финны — «кошкой», а шведы — «булочкой с корицей».

Схема работы электронной почты представлена на рис. 4.6.





Рис. 4.6. Схема работы электронной почты

В отличие от других сервисов Интернета электронная почта использует не один, а два протокола:

- **протокол SMTP** (*Simple Mail Transfer Protocol* — простейший протокол передачи сообщений) — для отправки почты;
- **протокол POP3** (*Post Office Protocol 3* — протокол почтового отделения, версия 3) или **IMAP** (*Internet Message Access Protocol* — протокол доступа к сообщениям в Интернете) — для приёма почты.

Так как отправлять почту имеет право кто угодно и кому угодно, протокол SMTP не предполагает идентификацию отправителя. Получить письмо должен именно тот, кому оно адресовано; тут необходима идентификация получателя (по логину и паролю), которая и обеспечивается протоколом POP3 или IMAP.



Используя дополнительные источники, выясните, в чём различие между протоколами POP3 и IMAP.

Просматривать сообщения электронной почты пользователи могут либо с помощью специальных программ (почтовых клиентов), либо посредством веб-интерфейса.

У каждого из этих способов есть свои достоинства и недостатки. Преимуществом **почтовых клиентов** является то, что при их использовании нет необходимости постоянного соединения с Интернетом. Интернет нужен на непродолжительное время для загрузки сообщений с почтового сервера на компьютер пользователя. Вся корреспонденция пользователя (и полученная, и отправленная) сохраняется на этом компьютере. Недостатком этого

способа является то, что доступ к почтовому ящику осуществляется только с того компьютера, на котором установлен почтовый клиент, настроенный на конкретного пользователя.

При использовании **веб-интерфейса** работа с электронной почтой осуществляется с помощью браузера, в окно которого загружена специальная гипертекстовая страница сайта почтовой службы. При этом вся корреспонденция пользователя хранится в его почтовом ящике на сервере почтовой службы. Доступ к сообщениям осуществляется с любого компьютера, подключённого к сети, однако для чтения корреспонденции в этом случае необходимо наличие подключения к Интернету.

Примеры почтовых клиентов: *Microsoft Outlook* и *Mozilla Thunderbird* (рис. 4.7).



Рис. 4.7. Логотипы почтовых клиентов *Microsoft Outlook* и *Mozilla Thunderbird*

Примеры почтовых служб: *Mail.ru* (<https://mail.ru/>), *Яндекс.Почта* (<https://mail.yandex.ru/>), *Gmail* (<https://gmail.com/>).

Одним из ощутимых недостатков электронной почты является возможность массовой рассылки спама (англ. *spam*) — электронных посланий, рекламных писем и т. п. адресатам, не выразившим желания получать такого рода корреспонденцию.

Одной из разновидностей спама являются «нигерийские письма». Выясните, что это такое.



Пользователи Сети имеют возможность принимать участие в различных формах коллективного взаимодействия — форумах, чатах, социальных сетях.

Для организации общения посетителей веб-сайта организуются веб-форумы. **Форум** предлагает набор разделов для обсуждения. Зарегистрированные на сайте пользователи, посылая свои сообщения, могут создавать внутри разделов темы и вести обсуждения в рамках этих тем. Сообщение и все ответы на него образуют «ветку» форума. Незарегистрированные пользователи получают статус гостей, которые могут просматривать ветки форума, но

не имеют права принимать участие в обсуждениях. За соблюдением правил следят **модераторы**, имеющие право редактировать, перемещать и удалять сообщения в определённом разделе или теме.

Чат (англ. *chat* — болтать) — сервис группового общения, предполагающего обмен сообщениями по компьютерной сети в режиме реального времени. Коммуникация именно в реальном времени отличает чат от форума.

В наше время среди пользователей Интернета широкое распространение получили **социальные сети** — интерактивные многопользовательские веб-сайты, содержание (контент) которых создаётся самими участниками сети. Социальные сети позволяют общаться группам пользователей, объединённых общими интересами. Система «друзей» и «групп» — отличительная черта социальных сетей.

Ещё одной средой социального общения является **блог** — общедоступный интернет-дневник, владелец которого (блогер) желает разделить свои интересы с единомышленниками. Основным содержанием такого рода веб-сайтов являются регулярно добавляемые его владельцем записи (посты) с указанием даты их создания. Любой посетитель блога может прочесть заинтересовавший его пост и оставить после него свой комментарий.



В мире уже более 2 миллиардов блогеров, и число их постоянно растёт. Блоги и ведущие их блогеры способны оказывать влияние на взгляды как отдельных подписчиков, так и общества в целом. Блоги, имеющие большое число подписчиков, могут приносить их владельцам доход за счёт размещения рекламы. Поэтому многие молодые люди мечтают стать блогерами. Впечатление, что успешным блогером может стать каждый, обманчиво. Для успеха надо найти свою тему, научиться создавать интересный пользователям контент и качественно его представлять. Ведение блога — это серьёзная и кропотливая работа.

4.2.3. Другие интернет-сервисы

Без многочисленных информационных ресурсов и сервисов сети Интернет трудно представить жизнь современного человека. На компьютере, подключённом к Сети, регулярно обновляются антивирусные базы; пользователям предлагаются новые версии программного обеспечения. Поработав с ознакомительной версией интересующей вас программы, можно, не выходя из дома, опла-

тить и активировать её полнофункциональную версию. В Интернете можно узнать самый точный прогноз погоды (<https://gismeteo.ru>), расписание электричек, поездов и самолётов (<https://rasp.yandex.ru>), найти любой адрес и проложить маршрут к пункту назначения (<https://maps.yandex.ru>) и пр. Через специальные сайты можно купить электронный билет на самолёт или на поезд, в театр или на выставку. В интернет-магазинах можно купить книги, бытовую технику и электронику, одежду и обувь, другие товары, которые будут доставлены покупателю на дом.

Как правило, цены в интернет-магазинах оказываются ниже, чем цены на такие же товары в обычных торговых центрах. Как вы можете это объяснить? Обсудите этот вопрос в группе.

Через порталы государственных и муниципальных услуг можно записаться на приём к врачу, оплатить услуги жилищно-коммунального хозяйства, проследить за успеваемостью ученика (посмотреть оценки, домашние задания, информацию о школьных событиях), решить множество других важных вопросов.

С помощью Интернета организуются деловые встречи и совещания, семинары и конференции. В течение одного дня, не выходя из офиса, можно провести несколько встреч с партнёрами, находящимися в разных уголках страны и за рубежом, сэкономив при этом время и силы участников мероприятий, бюджеты организаций. Означает ли это, что следует отказаться от проведения очных встреч? Обсудите этот вопрос в группе.

4.2.4. Интернет как источник больших данных

Пользователи Интернета не только получают информацию из Сети, но и сами активно насыщают её Всемирную паутину: создают сайты и блоги, заводят личные страницы в социальных сетях, принимают обсуждения на форумах и в чатах. Эта информация, а также информация, хранящаяся во всевозможных архивах и базах данных различных административных, коммерческих и научных структур, наряду с данными, поступающими от всевозможных устройств (Интернет вещей) и измерительных приборов, является основным источником **больших данных (Big Data)** — колоссальных объёмов структурированной и неструктурированной информации, обработка которых специальными методами позволяет делать точные прогнозы и принимать решения.



Объём больших данных измеряется в зеттабайтах. Чтобы лучше понять, что представляет собой зеттабайт, познакомьтесь с информацией, представленной в табл. 4.1.

Таблица 4.1

Единицы измерения информации

Название единицы измерения информации	Значение единицы измерения в байтах ¹	Название соответствующего числительного	Пример величины с соответствующим количественным значением
Килобайт	10^3 байт	тысяча	Количество звёзд, видимых глазом: $5 \cdot 10^3$
Мегабайт	10^6 байт	миллион	Количество книг в мире: $130 \cdot 10^6$
Гигабайт	10^9 байт	миллиард	Население Земли: $7 \cdot 10^9$
Терабайт	10^{12} байт	триллион	Количество кирпичей, которыми можно покрыть все материки равномерным сплошным пластом высотой 10 м: 10^{12}
Петабайт	10^{15} байт	квадриллион	Количество особей в популяции муравьёв на Земле: 10^{15}
Эксабайт	10^{18} байт	квинтиллион	Количество насекомых на Земле: $2,5 \cdot 10^{18}$
Зеттабайт	10^{21} байт	секстиллион	Количество молекул в капле воды: $7 \cdot 10^{21}$

¹ Так сложилось исторически, что в зависимости от ситуации могут использоваться или двоичные, или десятичные значения единиц измерения информации. Об этом мы более подробно поговорим в старшей школе. Впрочем, вы можете провести самостоятельное исследование этого вопроса.

САМОЕ ГЛАВНОЕ

Сеть Интернет обеспечивает пользователей многочисленными сервисами для накопления и распространения информации, общения и сотрудничества.

Всемирная паутина (WWW) — это сервис, с помощью которого пользователи Интернета получают доступ к информационным ресурсам, хранящимся на компьютерах в разных частях света.

Документ, размещённый в Интернете, имеет адрес — URL. Доступ пользователей к документам, размещённым на веб-сайтах, возможен по протоколу HTTPS (HTTP).

Файловый архив — хранилище в сети Интернет файлов с программным обеспечением, графикой, музыкой и другой информацией. Доступ к файловым архивам возможен по протоколу передачи файлов FTP.

Облачные сервисы — это среды, обеспечивающие удобный сетевой доступ к компьютерному оборудованию, дисковой памяти, программному обеспечению и базам данных через Интернет в удалённом формате.

Интернет — важнейшее средство личной и социальной коммуникации. Личная коммуникация осуществляется с помощью электронной почты, а также ряда мессенджеров, обеспечивающих возможность мгновенного обмена сообщениями. Социальные сети являются основным инструментом социальной коммуникации.

Электронная почта (e-mail) — система обмена сообщениями (письмами) между абонентами компьютерных сетей. Это одна из первых служб Интернета, до сих пор не утратившая своего значения.

Пользователи Сети имеют возможность принимать участие в различных формах коллективного взаимодействия — форумах, чатах, социальных сетях.

Большие данные — колоссальные объёмы структурированной и неструктурированной информации, обработка которых специальными методами позволяет делать точные прогнозы и принимать решения. Одним из основных источников больших данных являются социальные сети, блоги, форумы и другие интернет-сервисы.



Вопросы и задания

1. Опишите организацию и назначение сервиса WWW.

2. Даны запросы к поисковому серверу:

- 1) алабай & сербернар
- 2) алабай
- 3) такса & сербернар | алабай
- 4) (сербернар | такса) & алабай
- 5) алабай | сербернар

Изобразите графически множества страниц, которые найдёт поисковый сервер по каждому запросу. Расположите номера запросов в порядке убывания количества страниц, которые найдёт поисковый сервер по каждому запросу.

3. В языке запросов поискового сервера для обозначения логической операции ИЛИ используется символ «|», а для обозначения логической операции И — символ «&». В таблице приведены запросы к поисковому серверу и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тыс.)
олимпиада	4300
конкурс	4150
олимпиада & конкурс	1750

Какое количество страниц (в тысячах) будет найдено по запросу

олимпиада | конкурс ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

4. В языке запросов поискового сервера для обозначения логической операции ИЛИ используется символ «|», а для обозначения логической операции И — символ «&». В таблице приведены запросы к поисковому серверу и количество найденных по ним страниц некоторого сегмента сети Интернет.



Запрос	Найдено страниц (в тыс.)
дневник & учебник & тетрадь	49
дневник & учебник	113
дневник & тетрадь	186

Какое количество страниц (в тысячах) будет найдено по запросу

(дневник & учебник) | (дневник & тетрадь) ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

5. Что такое URL? Какова его структура?
6. Проанализируйте адреса следующих документов:
 - 1) <https://inf.1sept.ru/2007/13/01.htm>
 - 2) <https://bosova.ru/metodist/authors/informatika/3/gia.php>
7. Ребята на каникулах ездили всем классом в Музей искусств. С разрешения экскурсовода Саша сфотографировал много интересных экспонатов на свой смартфон. Вернувшись домой, Саша создал презентацию и выложил её на школьный компьютер по адресу: D:\2022\Каникулы\9b.ppt. Презентация оказалась настолько удачной, что классный руководитель выложила её на школьном сервере. Она переместила файл в каталог 9bclass, расположенный в корневом каталоге на сайте school2.site.org, доступ к которому осуществляется по протоколу https. Имя файла не изменилось. Укажите новый адрес файла с презентацией.
8. По какому протоколу возможен доступ пользователей к информации, хранящейся в файловых архивах?
9. Как вы считаете, любую ли информацию можно хранить в облаке? Обсудите этот вопрос в группе.
10. Опишите в общих чертах схему работы электронной почты.
11. Какие существуют способы работы с сообщениями электронной почты? Проанализируйте достоинства и недостатки каждого из них. Обсудите этот вопрос в группе.





12. Что общего и в чём различие между такими формами сетевого коллективного взаимодействия, как форум и чат? Обсудите этот вопрос в группе.

13. Что вы знаете о социальных сетях? Подготовьте небольшое сообщение.



14. Перечислите известные вам интернет-сервисы. Какими из них пользуетесь вы или члены вашей семьи? Обсудите этот вопрос в группе.



15. Работая в группе, составьте перечень крупнейших интернет-магазинов в мире или в нашей стране. Подготовьте сообщение о том, как устроена их работа.

16. Человеку трудно представить, насколько велики большие данные, так как в физическом мире трудно найти объекты, обладающие соответствующими количественными характеристиками.

Вот пример, который был приведён в одном из средств массовой информации: «В 2011 году объём мировых данных был таков, что ими можно было бы заполнить 56 млрд планшетов Apple iPad ёмкостью 32 гигабайт. Такого количества планшетов достаточно, чтобы 16 раз опоясать ими экватор!»

О каком объёме мировых данных идёт речь? Выразите его числом. Проведите вычисления, подтверждающие справедливость сравнения этого числа с числом, выражающим длину экватора.

§ 4.3

Деятельность в сети Интернет

Ключевые слова:

- онлайн-офис
- образовательная платформа
- система дистанционного обучения
- видео-конференц-связь
- структура сайта
- оформление сайта
- хостинг

Современный человек «живёт» в глобальной компьютерной сети, проводя в ней в среднем около 7 часов в сутки. Можно выделить несколько основных направлений деятельности, связанных с сетью Интернет: познавательную, коммуникативную, досуговую деятельность, а также деятельность, связанную с функционированием сети Интернет.

Познавательная деятельность прежде всего связана с поиском информации в сети Интернет; она вам уже достаточно хорошо известна. Хорошо известна каждому подростку и коммуникативная деятельность — общение с родными, друзьями и знакомыми с помощью мессенджеров и в социальных сетях.

Что касается досуговой деятельности, то прежде всего это просмотр видео- и аудиоресурсов, а также разнообразные сетевые игры.

Знаете ли вы, что такое киберспорт? Подготовьте небольшое сообщение на эту тему, подобрав соответствующую информацию в сети Интернет.



4.3.1. Учебная деятельность в сети Интернет

В последние годы ресурсы и сервисы сети Интернет всё чаще используются для учебной деятельности, тесно связанной с познавательной и коммуникативной деятельностью.

Онлайн-сервисы для создания цифровых продуктов. В сети Интернет существует множество сервисов, с помощью которых могут быть созданы разнообразные учебные материалы: презентации, всевозможные графические схемы, ленты времени, викторины, тесты и опросы по темам школьной программы. Учителя могут предлагать вам ссылки на такие учебные материалы или же давать задание с помощью подходящего онлайн-сервиса подготовить тот или иной материал самостоятельно.

Особое значение для уроков информатики имеют онлайн-среды разработки программ. Возможно, вы имеете опыт работы с одной из таких сред. Подготовьте короткое сообщение об этой среде, сравнив её возможности с аналогичным десктопным приложением. Если вы не знакомы с онлайн-инструментами программирования, то найдите информацию о них в Интернете. Как вы считаете, в чём основное преимущество таких инструментов?



Особо выделим **инструменты совместной работы**, наиболее известным из которых является **онлайн-офис Google Документы**.

С Google-документом могут одновременно работать несколько человек, которым был открыт соответствующий доступ. В зависимости от того, какие права определены для пользователя, он может работать в одном из трёх режимов: редактирование, комментирование, чтение. Создание и редактирование Google-документов возможно не только на компьютерах, но и на смартфонах.

Подготовка к совместной работе с документом состоит из следующих этапов:

- 1) создание документа (это делает учитель или ученик по поручению учителя);
- 2) открытие доступа к документу с правом редактирования;
- 3) приглашение соавторов (предоставление им ссылки на документ);
- 4) создание и редактирование документа.

Для того чтобы приступить к совместной работе над документом, ученику нужно:

- 1) завести учётную запись в *Google* (в этом случае все члены группы будут видеть имена друг друга);
- 2) перейти по полученной ссылке.

Созданные в сервисе документы автоматически сохраняются на *Google Диск*, и просмотреть их можно с любого компьютера или мобильного устройства (необходимо вспомнить пароль для входа в свой аккаунт и иметь доступ в Интернет).

Специализированные образовательные платформы и системы дистанционного обучения. Скорее всего, вы уже знакомы и имеете опыт работы с такими ресурсами. Например, пропустив по какой-нибудь причине школьный урок, можно самостоятельно освоить его содержание, ознакомившись с видеороликами и выполнив интерактивные задания на портале «Российская электронная школа» (<https://resh.edu.ru/>).

Вы можете принять участие в дистанционных конкурсах (Международный конкурс по информатике «Бобёр» <http://gotourl.ru/11457>), пройти дистанционную подготовку к олимпиаде по информатике (<http://gotourl.ru/12586>), более полно изучить интересующую вас тему (онлайн-курсы Stepik <http://gotourl.ru/12827>) и подготовиться к сдаче ОГЭ (образовательный портал для подготовки к экзаменам <http://gotourl.ru/15603>). Многие люди дистанционно получают высшее образование.

Всё больше учеников дистанционно осваивают интересные их курсы в системе дополнительного образования.

Такие курсы, как правило, строятся на базе систем дистанционного обучения (СДО), которые не только предоставляют обучающемуся разнообразные учебные материалы, но и полностью отслеживают то, как успешно он осваивает этот материал. Примером одной из самых распространённых СДО является *Moodle* (<http://gotourl.ru/15592>).

В последнее время во всем мире широкое распространение получают **открытые образовательные ресурсы** — обучающие, учебные или научные ресурсы, существующие в открытом доступе или выпущенные под лицензией, разрешающей их свободное использование или переработку.

Примеры открытых образовательных ресурсов:

- онлайн-курсы образовательного центра «Сириус» (<https://gotourl.ru/15593>);
- образовательная платформа *GeekBrains* (<https://gotourl.ru/15594>).

Онлайн-занятия. Скорее всего, вы уже имеете опыт участия в онлайн-занятиях, которые проходили с использованием одной из платформ для **видео-конференц-связи (ВКС)**: *Zoom*, *Microsoft Teams*, *Сферум*, *BigBlueButton*, *Discord*. Онлайн-занятия предстоят вам и в будущем. Поэтому важно знать и соблюдать основные правила поведения на занятиях такого типа.

- Подготовьтесь к занятию. Приведите в порядок рабочее место и свой внешний вид. Сообщите членам семьи время видеоконференции и договоритесь, чтобы вас не отвлекали во время занятия. Перед началом занятия закройте лишние вкладки, выйдите из чатов в соцсетях и игр.
- Не опаздывайте на занятие. Подключитесь к нему за 5 минут до начала. Проверьте микрофон, звук. При обнаружении каких-либо неполадок сообщите об этом в чат. Убедитесь в наличии достаточного уровня зарядки используемых устройств; приготовьте письменные принадлежности.
- Не скрывайте своё имя за нечитаемым псевдонимом; на учебном занятии желательно использовать свои настоящие имя и фамилию.
- При входе на онлайн-занятие необходимо поприветствовать учителя, написав в чат «Здравствуйте!» или «Добрый день!».
- При ответе на вопрос учителя нужно включить камеру; после ответа её можно выключить.



- Если вы хотите задать вопрос, то нажмите на значок «поднять руку»; задавайте вопрос только после того, как будет получено разрешение учителя. Чтобы не перебивать говорящего, вопрос можно написать в чат.
- Во время онлайн-занятия запрещено писать сообщения и подавать реплики, не относящиеся к его теме.

Зачастую для поддержки учебной коммуникации между учениками и учителем используются электронная почта и мессенджеры.



В 7 классе подробно обсуждался вопрос о том, каких правил следует придерживаться в электронной переписке. Вспомните эти правила. С помощью подходящего онлайн сервиса оформите инфографику «Правила этикета при работе с электронной почтой».

Переписка ученика с учителем по электронной почте, как правило, носит официальный характер. Использование мессенджеров (*WhatsApp*, *Telegram* и др.) носит чуть менее формальный характер, но тоже подчиняется некоторым правилам.

- В качестве аватара желательно установить своё реальное фото, а не фото домашнего питомца или любимого киноактёра.
- Не создавайте группы, включая в них людей без их согласия, или делайте это хотя бы после предупреждения.
- Соблюдая правила орфографии и пунктуации, обратите внимание на использование точки, которую в конце сообщения можно не ставить: считается, что точка свидетельствует о желании закончить разговор.
- Отправляя фото (особенно фото текста), проверьте его качество; если фото низкого качества и буквы на нём плохо читаются, сделайте фото повторно.
- Иногда в чате задаётся несколько вопросов. Выделяйте конкретный вопрос и давайте на него ответ, пользуйтесь функцией «ответить».
- Воздержитесь от использования смайлов и эмоджи в деловой переписке — они могут быть неоднозначно истолкованы.
- Не пересылайте личное сообщение, адресованное вам, другим людям или в общий чат.
- Чтобы логика вашего сообщения не была нарушена, отправляйте его целиком, а не частями.

- Голосовые сообщения отправляйте только по взаимному согласованию.
- Сообщения отправляйте в условно рабочее время — с 9:00 до 21:00.

Обсудите приведённые выше правила в группе. Поясните, чем вызвана необходимость соблюдения каждого из них. Какими ещё правилами можно дополнить этот перечень?



4.3.2. Создание веб-сайтов

Деятельность, связанная с функционированием сети Интернет, обширна и разнообразна. Мы рассмотрим её направление, относящееся к разработке веб-сайтов.

Технологии создания сайтов. Существует несколько способов создания веб-сайтов.

Во-первых, сайт можно создать, воспользовавшись **языком разметки гипертекста HTML** (*Hyper Text Markup Language*). В этом случае в текстовом редакторе (например, в *Блокноте*) текст, который хотят разместить на страницах сайта, размечают специальными метками, называемыми тегами. **Теги** содержат указания о том, как должен выглядеть текст. Чтобы отличать теги от текста, их заключают в угловые скобки (рис. 4.8).

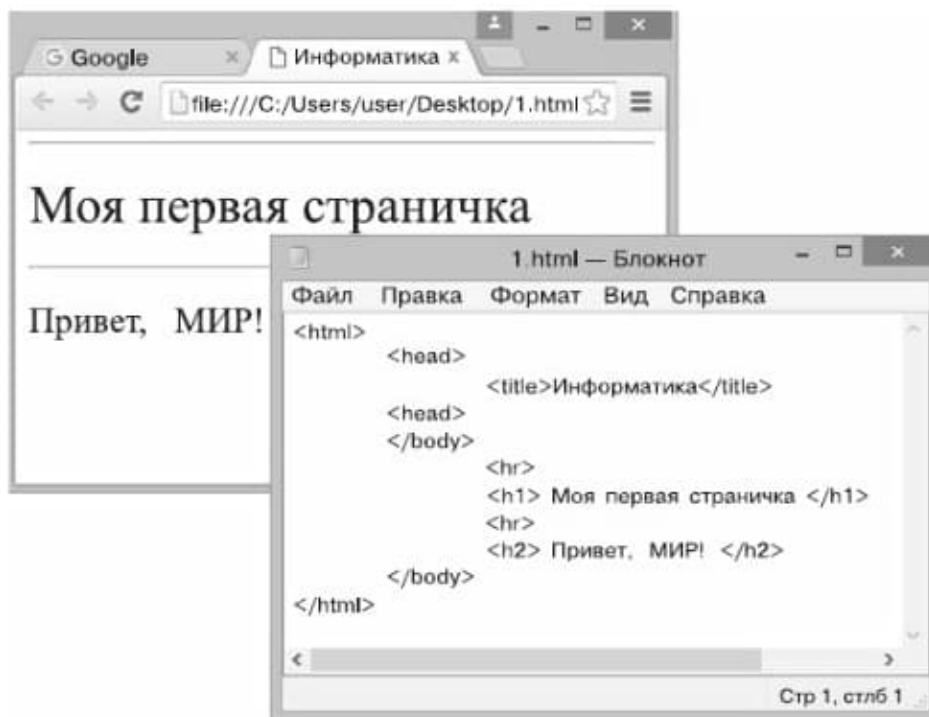


Рис. 4.8. Простейшая веб-страница и её HTML-код

Во-вторых, можно документ, подготовленный в текстовом процессоре (*Microsoft Word, OpenOffice Writer*), сохранить как **web-страницу** (в формате HTML).

В-третьих, можно воспользоваться специальным программным обеспечением — визуальными **HTML-редакторами** или **конструкторами сайтов**.

www

В сети Интернет найдите информацию о нескольких бесплатных конструкторах веб-сайтов. Сравните их по возможностям, понятности интерфейса, целевой аудитории и т. д. Какой из этих конструкторов сайтов вы хотели бы освоить?

В-четвёртых, сайт можно создать, воспользовавшись одним из языков программирования (PHP, Python, Perl, JavaScript). Эту сферу деятельности называют **веб-программированием**.

Веб-дизайнер — специалист, проектирующий структуру веб-сайта, а также подбирающий способы подачи информации и выполняющий художественное оформление веб-проекта. Веб-дизайнер должен знать современные веб-технологии; желательно, чтобы он имел художественное образование.

Содержание и структура сайта. Независимо от того, каким инструментарием вы будете пользоваться для создания сайта, первоначально следует продумать его содержание и структуру.

Предположим, вы посещаете кружок компьютерной графики и решили создать сайт этого творческого объединения. **Содержание сайта** должно быть таким, чтобы отражать творческую жизнь участников объединения, привлекать внимание единомышленников.

Теперь необходимо определить **структуру (план) сайта** — разбиение общего содержания на смысловые разделы и отдельные страницы с указанием связей между ними.

Очевидно, сайт может включать в себя общую информацию (программу и расписание занятий, место проведения занятий, фамилию, имя, отчество руководителя и т. д.), страницу новостей и объявлений, личные страницы руководителя и участников объединения, галерею работ участников объединения, полезные ссылки на другие ресурсы сети Интернет и многое другое. Изобразим структуру сайта в виде графа (рис. 4.9).



Рис. 4.9. Иерархическая структура сайта.

Представление структуры сайта в виде графа обеспечивает наглядное представление содержания сайта и помогает организовать **навигацию** — переходы с одной страницы сайта на другую.

Оформление сайта. Любой сайт загружается с **главной (домашней) страницы** (*home page*), которая прежде всего должна давать ответ на вопрос «О чём этот сайт?». Для этого на главной странице размещают название сайта, тематическое графическое изображение, короткий текст с описанием содержания сайта, а также главное меню — ссылки на основные разделы сайта. Также на главной странице могут быть размещены имя автора сайта и его контактная информация, счётчик посетителей, новости и т. д.

Каждая страница сайта, как правило, имеет несколько постоянных элементов, которые всегда находятся на одних и тех же местах. Это:

- заголовок сайта, расположенный в самом верху страницы;
- главное меню, как правило размещаемое в левой части страницы (количество пунктов меню, их названия и порядок неизменны на всех страницах сайта).

Желательно, чтобы страницы сайта были выполнены в едином стиле. Чтобы выдержать стиль, проще вначале разработать **шаблон страницы сайта**, на котором представить элементы, имеющиеся на всех страницах сайта (рис. 4.10).



Рис. 4.10. Примерный шаблон страниц сайта

При наполнении страниц сайта информационными материалами следует придерживаться следующих рекомендаций.

- Не ставьте точку в конце названий сайта и страниц.
- Названия пунктов меню делайте краткими; недопустимо растягивание названия пункта на несколько строк.
- Не выбирайте для страниц пёстрый фон, на таком фоне текст читается с трудом.
- Не размещайте на странице слишком много информационных и художественных блоков.
- Не пишите слишком длинные тексты. Текст разбивайте на небольшие абзацы, отделяя их друг от друга пустыми строками.
- В каждом абзаце используйте одни и те же приёмы форматирования.

Размещение сайта в Интернете. Перед размещением сайта в сети Интернет следует провести его тестирование, т. е. убедиться в том, что он правильно отображается разными браузерами: тексты хорошо читаются на выбранном фоне, рисунки расположены на своих местах, гиперссылки обеспечивают правильные переходы и т. д.

Хостинг — услуга по размещению сайта на сервере, постоянно находящемся в сети Интернет; хостинг может быть как платным, так и бесплатным.

Платным хостингом для размещения своих сайтов обычно пользуются крупные компании. При этом они получают удобное для прочтения и запоминания доменное имя второго уровня, которое может совпадать с названием компании.

Частные лица для своих сайтов могут воспользоваться бесплатным хостингом, возможности которого ограничены по сравнению с платными вариантами (небольшое дисковое пространство, ограничение размера отдельного файла, доменное имя только третьего уровня и т. д.), но их вполне достаточно для домашних страниц пользователей и некоммерческих сайтов. Главным недостатком бесплатных хостингов является то, что при размещении вашего сайта компания самостоятельно добавляет на его страницы коммерческую рекламу.

Список серверов, предоставляющих бесплатный хостинг, можно получить поиском по запросу «бесплатный хостинг».

САМОЕ ГЛАВНОЕ

Современный человек «живёт» в глобальной компьютерной сети, проводя в ней в среднем около 7 часов в сутки. Основными направлениями деятельности, связанными с сетью Интернет, являются: познавательная деятельность, включающая поиск информации в сети Интернет; коммуникативная деятельность — общение с родными, друзьями и знакомыми с помощью мессенджеров и в социальных сетях; досуговая деятельность (игры, кино, музыка), а также деятельность, связанная с функционированием сети Интернет.

В последние годы ресурсы и сервисы среды Интернет всё чаще используются для учебной деятельности, тесно связанной с познавательной и коммуникативной деятельностью. Основными инструментами учебной деятельности являются: онлайн-сервисы для создания цифровых продуктов; онлайн-офисы, поддерживающие возможность совместной работы; специализированные образовательные платформы и системы дистанционного обучения; платформы для видео-конференц-связи.

Деятельность, связанная с функционированием сети Интернет, обширна и разнообразна; одним из важных её направлений является разработка веб-сайтов.

Структура (план) сайта — разбиение общего содержания на смысловые разделы и отдельные страницы с указанием связей между ними.

Представление структуры сайта в виде графа обеспечивает наглядное представление содержания сайта и помогает организовать навигацию — переходы с одной страницы сайта на другую.

Шаблон страницы — это схема страницы, на которой представлены элементы, имеющиеся на страницах сайта. Использование шаблонов страниц помогает выдержать единый стиль оформления сайта.

Хостинг — услуга по размещению сайта на сервере, постоянно находящемся в сети Интернет; хостинг может быть как платным, так и бесплатным.



Вопросы и задания

1. Приведите 2–3 примера использования информационных ресурсов сети Интернет в своей учебной деятельности.
2. Приведите 2–3 онлайн-сервиса, предназначенных для создания цифровых продуктов.
3. В сети Интернет найдите несколько онлайн-сервисов, позволяющих создавать кластеры, ментальные карты, ленты времени, инфографику, интерактивные кроссворды или тесты. С их помощью создайте один из следующих цифровых продуктов: интерактивный кроссворд по теме «Моделирование»; интерактивный тест по теме «Обработка числовой информации в электронных таблицах»; кластер «Информационные модели»; ментальную карту «Алгоритмы и программирование»; ленту времени «История Интернета» (например, выстроив все события, представленные на страницах параграфов 4.1 и 4.2, в хронологической последовательности); инфографику «Алгоритмы управления». Можете предложить собственную тему цифрового продукта.
4. Работая в группе, с помощью онлайн-офиса *Google Документы* создайте презентацию «История компьютера в лицах». Представьте в ней краткие биографические справки и фотографии людей, внёсших вклад в развитие компьютерной техники и программного обеспечения.
5. В Интернете можно найти множество бесплатных онлайн-сред, позволяющих создавать графические изображения, не устанавливая себе на компьютер никаких программ. Найдите один или несколько таких инструментов. Попробуйте создать графическое изображение по собственному замыслу.



6. Найдите в сети Интернет информацию об истории появления и перспективах использования смайлов и эмодзи. Подготовьте короткое сообщение на эту тему.
7. Перечислите основные способы создания сайтов и коротко охарактеризуйте каждый из них.
8. Что такое структура сайта?
9. Продумайте и изобразите в виде графа структуру одного из следующих сайтов:
 - а) «Наш класс»;
 - б) «Моя семья»;
 - в) «Информатика».
10. Перечислите основные элементы, размещаемые на страницах сайта.
11. Какие рекомендации следует учитывать при наполнении страниц сайта информационными материалами?
12. С какой целью проводится тестирование сайта?
13. Что является основным недостатком бесплатного хостинга?
14. Назовите известный вам сервер Интернета, обеспечивающий пользователям бесплатный хостинг сайтов.
15. Любым доступным вам способом создайте одностраничный сайт-визитку, содержащий заголовок, красивую фотографию, приветствие, фамилию с именем, контакты для связи через социальную сеть или почту. Примерный вид страницы:



§ 4.4

Информационное общество: возможности и проблемы

Ключевые слова:

- информационное общество
- информационная безопасность
- информационная этика
- защита информации
- информационное право

4.4.1. Основные черты информационного общества

Большую и всё возрастающую роль в жизни современного человека играет информация. Неуклонно возрастает число людей, занятых в области информационных и коммуникационных технологий, всё шире проникающих в экономическую, политическую, социальную и духовную сферы жизни общества. Глобальное информационное пространство обеспечивает нам широкие возможности для личного и социального взаимодействия, доступа к мировым информационным ресурсам, удовлетворения многих потребностей в информационных продуктах и услугах. Во всех сферах деятельности происходит перевод различных видов информации в цифровую форму; разворачиваются процессы цифровой трансформации — преобразования методов осуществления профессиональной деятельности путём интеграции цифровых технологий и перехода к модели принятия решений, основанной на данных.

Всё это говорит о том, что мы живём в информационном обществе.



Информационное общество — современный этап развития цивилизации, когда главную роль играют информация и информационно-коммуникационные технологии.

Жизнь человека в обществе подчиняется нормам этики и права; в условиях информационного общества особое значение приобретает соблюдение человеком норм информационной этики и информационного права.

4.4.2. Информационная этика

Слово «этика» в переводе с греческого обозначает «нрав», «обычай»; его происхождение связывается с возникновением и развитием правил совместного сосуществования людей.

Этика — это система моральных и нравственных норм, принятых в обществе; её важнейшими категориями являются «добро», «зло», «ответственность», «справедливость», «долг». Продолжительное время вопросы этики затрагивали исключительно сферу «реальных» отношений в обществе.

В информационном обществе человек одновременно существует в двух мирах — мире реальном и мире виртуальном (цифровом). Виртуальный мир, обеспечивая условия для взаимодействия на основе анонимности и отсутствия цензуры, не только предоставляет пользователям возможность открыто и свободно выражать свои мысли и чувства, но и создаёт обстановку, в которой некоторые пользователи начинают ощущать себя безответственно; они пренебрегают этическими нормами, действующими в реальном мире: ощущая свою безнаказанность, размещают в Сети материалы, порочащие честь и достоинство других людей, материалы, призывающие к насилию, и т. п. Благодаря пропускной способности современных каналов связи, высокой скорости передачи и тиражирования информации такие действия могут нанести урон не только отдельной личности, но и обществу в целом. Поэтому в современном мире большое внимание уделяется информационной этике.

Информационная этика — это область этики, которая рассматривает проблемы, возникающие в связи с развитием и использованием ИКТ, в том числе глобальной сети Интернет.

Можно выделить следующие основные **нормы информационной этики**:

- 1) недопустимость несанкционированного открытия принадлежащих другим пользователям папок, файлов, почтовых ящиков, аккаунтов и просмотра их содержимого;
- 2) недопустимость несанкционированного изменения настроек рабочего стола компьютера, не являющегося собственностью пользователя;
- 3) недопустимость поиска, скачивания, создания и размещения в Сети контента, неэтичного по своему содержанию (например, содержащего жестокость, насилие, представляющего людей в неподобающем внешнем виде и т. д.);

- 4) недопустимость передачи содержания переписки по электронной почте, в мессенджерах и социальных сетях третьим лицам без предварительного согласия собеседника;
- 5) недопустимость нарушения норм сетевого общения, использования в социальных сетях, мессенджерах и электронной почте:
 - компьютерных жаргонизмов, влекущих засорение русского языка;
 - спама — ненужных адресату электронных посланий, рекламной информации и другого информационного мусора;
 - цепочечных писем («писем счастья» и «писем несчастья»), влияющих на психологическое состояние получателя;
 - оффтопика — размещения сообщений, не соответствующих общему направлению форума, явно выходящих за рамки установленной темы общения;
 - кросспостинга — сознательного размещения одной и той же информации в разных разделах форума или блога;
 - флуда — сообщений, текст которых представляет собой бессмысленный набор повторяющихся фраз, символов, букв, слов, не несущих смысловой нагрузки;
 - смайл-флуда — сообщений, изобилующих смайлами;
 - флейма — эмоциональных сообщений, текст которых не связан с рассматриваемой темой по существу, а сводится к выяснению личных отношений, может содержать ненормативную лексику и оскорбления друг друга;
 - троллинга — намеренного размещения сообщений провокационного характера с целью создания конфликтной ситуации между участниками обсуждения.

4.4.3. Информационное право

Понятие этики тесно связано с понятием **права**, так как и в том, и в другом случае речь идёт о ценностных формах сознания, служащих регуляторами поведения людей. Этика и право имеют общую цель, состоящую в том, чтобы гармонизировать интересы личности и общества, укрепить свободу и достоинство человека, сохранить общественный порядок. Основное различие между вышеназванными категориями состоит в том, что этиче-

ские установки существуют в общественном сознании, в то время как право выражает волю государства, оно директивно и не всегда предоставляет человеку возможность выбора поведения. Правовые нормы — общеобязательные, формально определённые правила поведения, зафиксированные в нормативно-правовых документах.

Информационное право — область права, регулирующая общественные отношения, связанные с созданием, хранением, обработкой, распространением, использованием информационных ресурсов.

Можно выделить следующие основные **нормы информационного права**.

На законодательном уровне **запрещено**:

- 1) распространять информацию, которая «направлена на пропаганду войны, разжигание национальной, расовой или религиозной ненависти и вражды, а также иную информацию, за распространение которой предусмотрена уголовная или административная ответственность» (Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27 июля 2006 г. № 149-ФЗ);
- 2) нарушать «неприкосновенность частной жизни, личную и семейную тайну», а также «право на тайну переписки, телефонных переговоров, почтовых, телеграфных и иных сообщений» (статья 23 Конституции РФ), используя информацию, полученную в процессе переписки по электронной почте, в мессенджерах и социальных сетях;
- 3) передавать третьим лицам без предварительного согласия сетевых пользователей их личные данные — реальные имена, адреса, телефоны, фотографии (Федеральный закон «О персональных данных» от 27 июля 2006 г. № 152-ФЗ);
- 4) нарушать авторские права и использовать нелицензионные материалы, в том числе программное обеспечение, являющиеся чужой интеллектуальной собственностью (ГК РФ);
- 5) осуществлять несанкционированный доступ к чужому контенту, размещённому на ПК, носителях информации, сетевых сервисах, путём взлома аккаунтов других пользователей (УК РФ);
- 6) разрабатывать, распространять и использовать вредоносные программы (программы генерации паролей, ключей, программы взлома систем защиты программ) и компьютерные вирусы (УК РФ).

4.4.4. Информационная безопасность



Информационная безопасность — это защищённость информации, которой обладает человек, исключение рисков её уничтожения, искажения и утечки, которые могут привести к невосполнимым потерям или ущербу для лица, обладающего этой информацией.

В условиях повсеместного распространения Интернета, высокопроизводительных персональных компьютеров и разнообразных мобильных устройств необходимо, чтобы:

- все граждане были осведомлены о необходимости обеспечивать информационную безопасность;
- все граждане несли ответственность за информационную безопасность.

Рассмотрим следующие угрозы информационной безопасности:

- **непреднамеренные**, возникающие вследствие ошибок пользователя; сбоя технических и программных средств информационных систем; природных явлений или иных событий, не направленных специально на изменение информации;
- **преднамеренные**, возникающие в результате целенаправленных действий, нарушающих установленные права и правила доступа, приводящих к утечке, искажению, подделке, уничтожению, блокированию доступа к информации, а также к утрате, уничтожению или сбою функционирования носителя информации.



Защита информации — деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

Пользователь, являющийся частным лицом, может обеспечить защиту информации на своём компьютере, если будет:



- постоянно использовать блок бесперебойного питания;
- периодически выполнять резервное копирование файлов на внешние носители;
- постоянно использовать антивирусную программу, регулярно обновлять антивирусные базы и осуществлять антивирусную проверку компьютера.

Возможности для защиты информации от случайной потери или удаления предусмотрены и в программном обеспечении компьютера. Например:

- предусмотрены предупреждения о наличии в электронном документе макроопределений, под которые могут быть замаскированы вирусы;
- имеется возможность отменять последние действия;
- предусмотрены запросы на подтверждение команд, приводящих к изменению содержания или удалению файла или группы файлов и т. д.

Одним из основных способов противодействия угрозам информационной безопасности при работе в сети Интернет является аутентификация, требуемая при доступе к таким интернет-сервисам, как электронная почта, социальные сети, интернет-магазины, платёжные системы и др.

Для использования такого рода интернет-сервисов пользователь должен пройти регистрацию в системе и завести логин и пароль. Зайдя на сайт интернет-сервиса, пользователь указывает свой логин (идентификатор), фактически заявляя, кем он является; этот этап носит название идентификации. Следующим этапом является аутентификация — пользователь вводит свой пароль, тем самым подтверждая, что предъявленный им идентификатор является подлинным. Далее происходит авторизация — проверка того, что пользователю с указанными логином и паролем разрешён доступ к запрашиваемому ресурсу.

Описанный процесс (идентификация — аутентификация — авторизация) может показаться вам излишне сложным. Но проанализируйте, что будет происходить с каждым из вас, например, при сдаче ОГЭ. Вы должны будете подойти к организатору экзамена и предъявить ему паспорт. Организатор найдёт вашу фамилию в списке и проводит вас в аудиторию, где будет проходить экзамен. Что в данном случае будет аналогично идентификации, аутентификации и авторизации?

На протяжении всех трёх лет изучения курса информатики мы неоднократно обращали внимание на необходимость придерживаться определённых правил поведения в Интернете, иначе называемых безопасными стратегиями поведения в сети Интернет.

Основная идея правил (стратегий) безопасного поведения в сети Интернет: не делайте в виртуальном мире ничего такого, что бы вы не стали делать в реальном мире.

Правила безопасного поведения в сети Интернет



Создавая свой профиль в социальных сетях, не указывайте свой адрес, дату рождения, школу, класс. Придумывая себе логин (ник, имя пользователя), вместо имени используйте псевдоним; вы можете отразить в нём свои стремления, характер, интересы. При этом личную информацию, такую как ваша фамилия или дата рождения, включать в логин не рекомендуется.



Пароли должны быть уникальными; цифры и спецсимволы значительно усложняют процесс их подбора злоумышленниками. В социальные сети, мессенджеры и почту безопаснее входить через приложения; ввода паролей в браузерах следует избегать. Не пользуйтесь сервисами, которые сохраняют пароли: онлайн-сервисы для хранения паролей ненадёжны; их часто взламывают. Никому не сообщайте свои пароли. При завершении работы с общедоступным компьютером корректно выходите из учётных записей, которые вы использовали.



Будьте особенно бдительны при совершении интернет-покупок, согласовывайте все платежи с родителями. Обратите внимание на строку браузера на странице, на которой вводятся платёжные данные; проверьте, есть ли в строке браузера изображение закрытого навесного замочка — он означает безопасную передачу данных. Когда замочек открыт, защита недостаточна. Если есть изображение восклицательного знака или перечёркнутого замка, на таком сайте не стоит вводить свои данные.



Свои персональные данные (фамилию, домашний адрес, номер телефона, название школы, номера документов и т. д.) можно вводить только на государственных сайтах или на сайтах для покупки билетов. И только в том случае, если соединение устанавливается по протоколу HTTPS. Не оставляйте свои персональные данные на других сайтах.



Старайтесь не выкладывать полную информацию (фотографии, видеозаписи) о себе и жизни своей семьи на всеобщее обозрение. Доступ к такой информации можно открывать только проверенным людям: родным, близким, друзьям и людям, с которыми вы знакомы в реальной жизни. Помните: всё, что попало в Сеть, останется там навсегда. Как правило, стереть данные из Сети невозможно. Единственный способ избежать утечки информации — не делиться ею без особой на то необходимости.



Учитесь различать оригинальные и поддельные сайты. Поддельные сайты могут иметь дизайн и адрес, напоминающие сайт-оригинал. Злоумышленники ждут, когда человек введёт логин и пароль на поддельном сайте. Узнав эти данные, они используют их для входа в настоящий профиль своей жертвы.



Особенно внимательно ведите себя в онлайн-играх: игровые объекты, членство в командах, социальные связи внутри игры — всё это может стать механизмом манипуляции для мошенников и других злоумышленников. В Интернете много сайтов, содержащих аркады, головоломки, другие игры с системой начисления очков; здесь деньги не тратятся. Сайты с азартными играми, как правило, связаны с выигрышем или проигрышем денег. Привычка к азартным играм приводит к формированию у человека психологической зависимости, представляющей серьёзную социальную и медицинскую проблему.



Поскольку каждый пользователь Интернета может опубликовать любую информацию, не всё, что вы видите в Сети, верно. Старайтесь мыслить критически, чтобы оценить достоверность материалов. При поиске информации по интересующему вас вопросу обращайтесь внимание на источник информации, отдавая предпочтение официальным сайтам. Сверяйте найденную в Сети информацию по 2–3 источникам; проверяйте, есть ли в Сети другие мнения и факты, которые противоречат ранее найденной вами информации. Не доверяйте безоговорочно сайтам с кричащими заголовками и обилием рекламы; следует насторожиться, если пользователя, щёлкнувшего на какой-либо новости, перекидывают куда-то дальше. Не посещайте сайты расистского, дискриминационного, насильственного содержания — они способны поставить под угрозу ваше психологическое и физическое здоровье.



Соблюдайте сетевой этикет. Не старайтесь привлечь к себе внимание за счёт эпатажа. Пишите грамотно. Не оскорбляйте других, не будьте навязчивы, не позволяйте своим негативным эмоциям выходить из-под контроля. Получив оскорбительное или иное сообщение, заставляющее вас чувствовать себя некомфортно, не отвечайте на него. Никогда не участвуйте в травле: буллинг в Сети ничем не отличается от реального и одинаково опасен и для жертвы, и для агрессора.



Онлайн-друг может быть совсем не тем человеком, за кого он себя выдаёт. Злоумышленники, выдавая себя за вашего сверстника, могут пытаться выведать частную информацию о вас и членах вашей семьи. Затем, в зависимости от своих целей, они могут искать личной встречи, угрожать жертве. Прежде чем вступить в диалог с незнакомцем, обратите внимание на его возраст и число друзей в Сети; следует насторожиться, если незнакомец старше вас, имеет очень мало друзей, просит вас выслать ему какие-нибудь фото или данные, — это явные приметы злоумышленника.



Фишинг — вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей. Опасайтесь интернет-мошенничества: получив сообщение о выигрыше или возможности бесплатного получения какой-то вещи, не вводите пароли, номера телефонов, банковских карт или другую личную информацию. Помните: подлинные интернет-сервисы не рассылают пользователям писем с просьбами сообщить свои учётные данные, пароль и прочее.



Многие веб-сайты предлагают пользователям для бесплатного скачивания различные приложения, игры, музыку, фильмы, документы, которые могут содержать вирусы. Избежать заражения вирусами помогают антивирусные программы.



Позаботьтесь о безопасности копий важных документов, которые вы решили хранить в облаке. Отсканированные документы поместите в архив, при создании которого выберите опцию «непрерывный архив» (*solid archive*) и установите для этого архива пароль. Не используйте один и тот же пароль для разных архивов.



Избегайте любых деструктивных и криминальных форм сетевой активности.



Помните! Защита вашей информации, ваша информационная безопасность — это прежде всего зона вашей ответственности.

4.4.5. Профессии, связанные с информатикой и информационными технологиями

Информатика и информационные технологии оказывают очень важное влияние на развитие экономики всего мира и каждой страны.

Отрасль информационных технологий (ИТ-отрасль) — стратегический сегмент экономики России. В национальной программе «Цифровая экономика Российской Федерации» определены научно-технические направления, имеющие ключевое значение для развития и процветания нашей страны. Среди них: большие данные; нейротехнологии и искусственный интеллект; промышленный Интернет; компоненты робототехники и сенсорика; технологии беспроводной связи; технологии виртуальной и дополненной реальностей и др.

Безусловно, такие сложные вопросы не изучаются в школе. Тем не менее вы уже имеете некоторые представления по каждому из этих направлений и можете думать о том, чем бы вы хотели заниматься в недалёком будущем.

Что вам известно о таких профессиях, как веб-дизайнер, программист, разработчик мобильных приложений, тестировщик, архитектор программного обеспечения, специалист по анализу данных, системный администратор? вспомните, при изучении каких тем школьного курса информатики вы получили представление об этих профессиях.

Окружающий мир стремительно изменяется. Не исключено, что профессии, востребованные сегодня, не будут столь же востребованы уже через 6–7 лет. Для того чтобы лучше представлять будущее, советуем вам заглянуть в книгу «Атлас новых профессий 3.0» (<http://gotourl.ru/15595>). В ней представлена информация о профессиях ближайшего будущего, охватывающих 27 отраслей — от добычи полезных ископаемых до медиа и развлечений. Причём все они так или иначе связаны с использованием информационных технологий.

Какую бы профессию вы ни выбрали для себя в будущем, чтобы достигнуть в ней успеха, прежде всего необходимо обладать способностью ставить цели и достигать их, способностью учиться, в том числе на своих ошибках, способностью взаимодействовать с другими и т. д. В любой профессиональной сфере необходимы



навыки чтения и письма, вычислительные навыки, а также навыки тайм-менеджмента и др. Обладая перечисленными выше способностями и навыками, вы сможете не только освоить ряд специализированных навыков в некоторой сфере профессиональной деятельности (например, в программировании), но и постоянно их обновлять (например, осваивая новые среды и новые технологии программирования).

Способность к саморазвитию — главное условие успеха в современном, стремительно изменяющемся мире!

САМОЕ ГЛАВНОЕ

Информационное общество — современный этап развития цивилизации, когда главную роль играют информация и информационно-коммуникационные технологии.

Жизнь человека в обществе подчиняется нормам этики и права. Этика — это система моральных и нравственных норм, принятых в обществе; правовые нормы — общеобязательные, формально определённые правила поведения, зафиксированные в нормативно-правовых документах.

В условиях информационного общества особое значение приобретает соблюдение человеком норм информационной этики и информационного права.

Информационная этика — это область этики, которая рассматривает проблемы, возникающие в связи с развитием и использованием ИКТ, в том числе глобальной сети Интернет.

Информационное право — область права, регулирующая общественные отношения, связанные с созданием, хранением, обработкой, распространением, использованием информационных ресурсов.

Информационная безопасность — это защищённость информации, которой обладает человек, исключение рисков её уничтожения, искажения и утечки, которые могут привести к невозможным потерям или ущербу для лица, обладающего этой информацией. Различают непреднамеренные и преднамеренные угрозы информационной безопасности.

Защита информации — деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

Помните, защита вашей информации, ваша информационная безопасность — это прежде всего зона вашей ответственности.




Информатика и информационные технологии оказывают очень важное влияние на развитие экономики всего мира и каждой страны.

Вопросы и задания



1. Какое общество называется информационным? Перечислите его основные черты.
2. Какую роль играют нормы этики и права в жизни общества?
3. С чем связано появление информационной этики и информационного права?
4. Назовите основные нормы информационной этики.
5. Назовите основные нормы информационного права.
6. Какие правовые нормы использования программного обеспечения вам известны?
7. Что такое информационная безопасность? Приведите примеры угроз информационной безопасности.
8. Что такое защита информации? Кто несёт ответственность за защиту информации пользователя?
9. Как пользователь может обеспечить защиту информации на своём компьютере?
10. Какие возможности для защиты информации от случайной потери или удаления предусмотрены в программном обеспечении компьютера?
11. Какими приёмами фишинга пользуются интернет-мошенники? Найдите соответствующие примеры в сети Интернет и подготовьте презентацию на эту тему.
12. Как вы считаете, можно ли похитить пароли, реквизиты банковских карт, списки контактов или другую конфиденциальную информацию со смартфонов? Что вам известно о вирусах и антивирусных программах для мобильных устройств?



13. Какова основная идея правил (стратегий) безопасного поведения в сети Интернет?
-  14. Распределите между учениками класса приведённые в учебнике правила безопасного поведения в сети Интернет. Совместными усилиями создайте презентацию «Безопасность в Интернете», разместив на её слайдах правила и соответствующие им визуальные образы — их вы можете подобрать в сети Интернет.
-  15. Одной из проблем информационного общества является социальная инженерия. Выясните, что это такое. Сталкивались ли вы, ваши родственники или знакомые с этим явлением?
16. Определитесь с профессиональным будущим вам могут помочь тесты на профориентацию, размещённые на сайте «Профориентатор» (Центр профориентации для школьников при МГУ им. М. В. Ломоносова, <http://gotourl.ru/15596>). По своему усмотрению выберите один из бесплатных тестов и пройдите его.
-  17. Совместными усилиями всех учеников класса создайте презентацию о профессиях, связанных с ИКТ.

Тестовые задания для самоконтроля

1. Дайте самый полный ответ. Множество компьютеров, соединённых линиями передачи информации, — это:
 - а) компьютерная сеть
 - б) локальная сеть
 - в) глобальная сеть
 - г) Интернет
2. Компьютерная сеть, действующая в пределах одного здания, — это:
 - а) локальная сеть
 - б) глобальная сеть
 - в) Интернет
 - г) одноранговая сеть
3. Локальная сеть, все компьютеры в которой равноправны, — это:
 - а) региональная сеть
 - б) сеть с выделенным сервером
 - в) Интернет
 - г) одноранговая сеть
4. Компьютер, предоставляющий свои ресурсы в пользование другим компьютерам при совместной работе, называется:
 - а) модемом
 - б) коммутатором
 - в) сервером
 - г) сетевой картой
5. Скорость передачи данных через некоторое соединение равна 128 000 бит/с. Сколько времени (в минутах) займёт передача файла объёмом 5 Мбайт по этому каналу?
 - а) 328
 - б) 41
 - в) 5,5
 - г) 40

6. Скорость передачи данных по некоторому соединению составляет 56 000 бит/с. Какое максимальное количество байт можно передать за 15 секунд по этому протоколу?
- а) 840 000
 - б) 84 000
 - в) 105 000
 - г) 105
7. Всемирная глобальная компьютерная сеть, сеть сетей — это:
- а) локальная сеть
 - б) сеть с выделенным сервером
 - в) Интернет
 - г) одноранговая сеть
8. Компьютер, подключённый к Интернету, обязательно имеет:
- а) IP-адрес
 - б) сервер
 - в) домашнюю веб-страницу
 - г) доменное имя
9. Адрес компьютера, записанный четырьмя десятичными числами, разделёнными точками, — это:
- а) URL
 - б) WWW
 - в) протокол
 - г) IP-адрес
10. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты цифрами 1, 2, 3 и 4. Восстановите IP-адрес.

.75

1

21

2

3.21

3

9.255

4

- а) 2413
- б) 3214
- в) 2341
- г) 4231

11. IP-адресу 64.129.255.32 соответствует 32-битовое представление:
- а) 10000000100000011111111100100000
 - б) 01000000100000011111111100100000
 - в) 01111111100000001111111110000000
 - г) 10000000100000011111111101000000
12. Запросы к поисковому серверу закодированы буквами А, Б, В, Г.
- А) Пушкин | Лермонтов | поэзия
 - Б) Пушкин | Лермонтов | поэзия | проза
 - В) Пушкин | Лермонтов
 - Г) Пушкин & Лермонтов & проза
- Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу.
- а) ВАБГ
 - б) ВБГА
 - в) ГВАБ
 - г) БАВГ
13. Указатель, содержащий название протокола, доменное имя сайта и адрес документа, — это:
- а) URL
 - б) WWW
 - в) протокол
 - г) IP-адрес
14. На сервере `ict.ru` находится документ `demo.html`, доступ к которому осуществляется по протоколу HTTPS. Фрагменты адреса данного файла закодированы цифрами от 1 до 7. Укажите последовательность цифр, которая кодирует адрес указанного документа в Интернете.

1	demo
2	.html
3	://
4	/
5	https
6	ict
7	.ru

- а) 5467312
 - б) 2367415
 - в) 5367412
 - г) 5312467
15. Сервис, обеспечивающий пересылку файлов между компьютерами Сети независимо от их типов, особенностей операционных систем, файловых систем и форматов файлов, — это:
- а) FTP
 - б) URL
 - в) ICQ
 - г) TCP/IP
16. Сервис, позволяющий любому пользователю Сети передавать и получать электронные сообщения, — это:
- а) FTP
 - б) e-mail
 - в) WWW
 - г) HTTPS
17. Облачным хранилищем не является:
- а) *Облако Mail.ru*
 - б) *Яндекс.Диск*
 - в) *Google Диск*
 - г) *Google Документы*
18. Система принятых в обществе моральных и нравственных норм, рассматривающая проблемы, возникающие в связи с развитием и использованием ИКТ, — это:
- а) информационная безопасность
 - б) информационное общество
 - в) информационная этика
 - г) информационное право
19. Общеобязательные, формально определённые и зафиксированные в нормативно-правовых документах правила, регулирующие общественные отношения, связанные с созданием, хранением, обработкой, распространением, использованием информационных ресурсов, — это:
- а) правила информационной безопасности
 - б) нормы информационной этики
 - в) нормы информационного права
 - г) способы защиты информации

20. Вид интернет-мошенничества, целью которого является получение доступа к логинам и паролям, номерам банковских карт пользователей, называется:
- а) фишингом
 - б) кибербуллинг
 - в) троллингом
 - г) флудом

КЛЮЧИ К ТЕСТОВЫМ ЗАДАНИЯМ ДЛЯ САМОКОНТРОЛЯ

Глава 1

Задание	1	2	3	4	5	6	7	8	9	10
Ответ	а	б	г	в	б	в	г	б	в	а
Задание	11	12								
Ответ	в	а								

Глава 2

Задание	1	2	3	4	5	6	7	8	9	10
Ответ	в	г	г	б	б	г	а	б	а	б
Задание	11	12	13	14	15	16	17	18	19	20
Ответ	г	г	а	г	в	г	а	в	б	г
Задание	21	22	23	24	25	26				
Ответ	б	а	а	в	б	в				

Глава 3

Задание	1	2	3	4	5	6	7	8	9	10
Ответ	г	в	г	в	б	в	б	б	г	а
Задание	11	12	13	14	15	16	17	18	19	20
Ответ	а	б	а	а	б	в	г	в	б	в

Глава 4

Задание	1	2	3	4	5	6	7	8	9	10
Ответ	а	а	г	в	в	в	в	а	г	в
Задание	11	12	13	14	15	16	17	18	19	20
Ответ	б	в	а	в	а	б	г	в	в	а

ОТВЕТЫ К ВОПРОСАМ И ЗАДАНИЯМ ДЛЯ САМОСТОЯТЕЛЬНОЙ ПОДГОТОВКИ

Глава 2

§ 2.2

3. 32,5 км/ч.

5. Российский спортсмен занял первое место, китайский — второе, белорусский — третье, спортсмен из Германии — четвертое, из Италии — пятое.

§ 2.3

5. 1,5 ч.

6. 6.

9. 24.

10. 648.

11. 16.

12. 32, 13.

14. Первый игрок.

§ 2.4

5. Второй игрок.

6. 3).

§ 2.5

21. 60.

Глава 3

§ 3.2

16. Если стоимость товаров превышает 2000, то доставка осуществляется покупателю бесплатно.

§ 3.3

11. г).

Глава 4

§ 4.1

- 8. Нет.
- 19. 20.

§ 4.2

- 2. 14235.
- 3. 6700 тыс.
- 4. 250 тыс.

ОГЛАВЛЕНИЕ

Введение	3
Глава 1. Алгоритмы и программирование	5
§ 1.1. Конструирование алгоритмов	5
§ 1.2. Запись вспомогательных алгоритмов на языке Паскаль	24
§ 1.3. Запись вспомогательных алгоритмов на языке Python	30
§ 1.4. Обработка одномерных массивов целых чисел на языке Паскаль	39
§ 1.5. Обработка одномерных массивов целых чисел на языке Python	55
§ 1.6. Алгоритмы управления	72
Тестовые задания для самоконтроля	79
Глава 2. Моделирование	82
§ 2.1. Моделирование как метод познания	82
§ 2.2. Знаковые модели	92
§ 2.3. Графические информационные модели	105
§ 2.4. Табличные информационные модели	118
§ 2.5. База данных как модель предметной области	131
Тестовые задания для самоконтроля	145
Глава 3. Обработка числовой информации в электронных таблицах	152
§ 3.1. Электронные таблицы	152
§ 3.2. Организация вычислений в электронных таблицах ...	162
§ 3.3. Средства анализа и визуализации данных	178
Задания для практических работ	194
Тестовые задания для самоконтроля	200

Глава 4. Информационно-коммуникационные технологии в современном обществе	206
§ 4.1. Локальные и глобальные компьютерные сети.....	206
§ 4.2. Информационные ресурсы и сервисы Интернета.....	219
§ 4.3. Деятельность в сети Интернет	236
§ 4.4. Информационное общество: возможности и проблемы	248
Тестовые задания для самоконтроля.....	261
Ключи к тестовым заданиям для самоконтроля	266
Ответы к вопросам и заданиям для самостоятельной подготовки	268



Учебное издание

**Босова Людмила Леонидовна
Босова Анна Юрьевна**

ИНФОРМАТИКА

9 класс

Базовый уровень

Учебник

Центр развития углублённого и профильного образования,
функциональной грамотности, технологии и ИКТ-компетенций

Ответственный за выпуск *О. Полежаева*
Ведущий редактор *О. Полежаева*
Художественное оформление *Е. Чайко*
Компьютерная вёрстка *Л. Катуркиной*
Технический редактор *Е. Денюкова*
Корректор *О. ШUTOва*

Подписано в печать 01.11.2022. Формат 70×100/16. Гарнитура SchoolBookCSanPin.
Усл. печ. л. 22,1. Тираж экз. Заказ

Акционерное общество «Издательство «Просвещение».
Российская Федерация, 127473, г. Москва,
ул. Краснопролетарская, д. 16, стр. 3, этаж 4, помещение I.
Адрес электронной почты «Горячей линии» — vopros@prosv.ru.