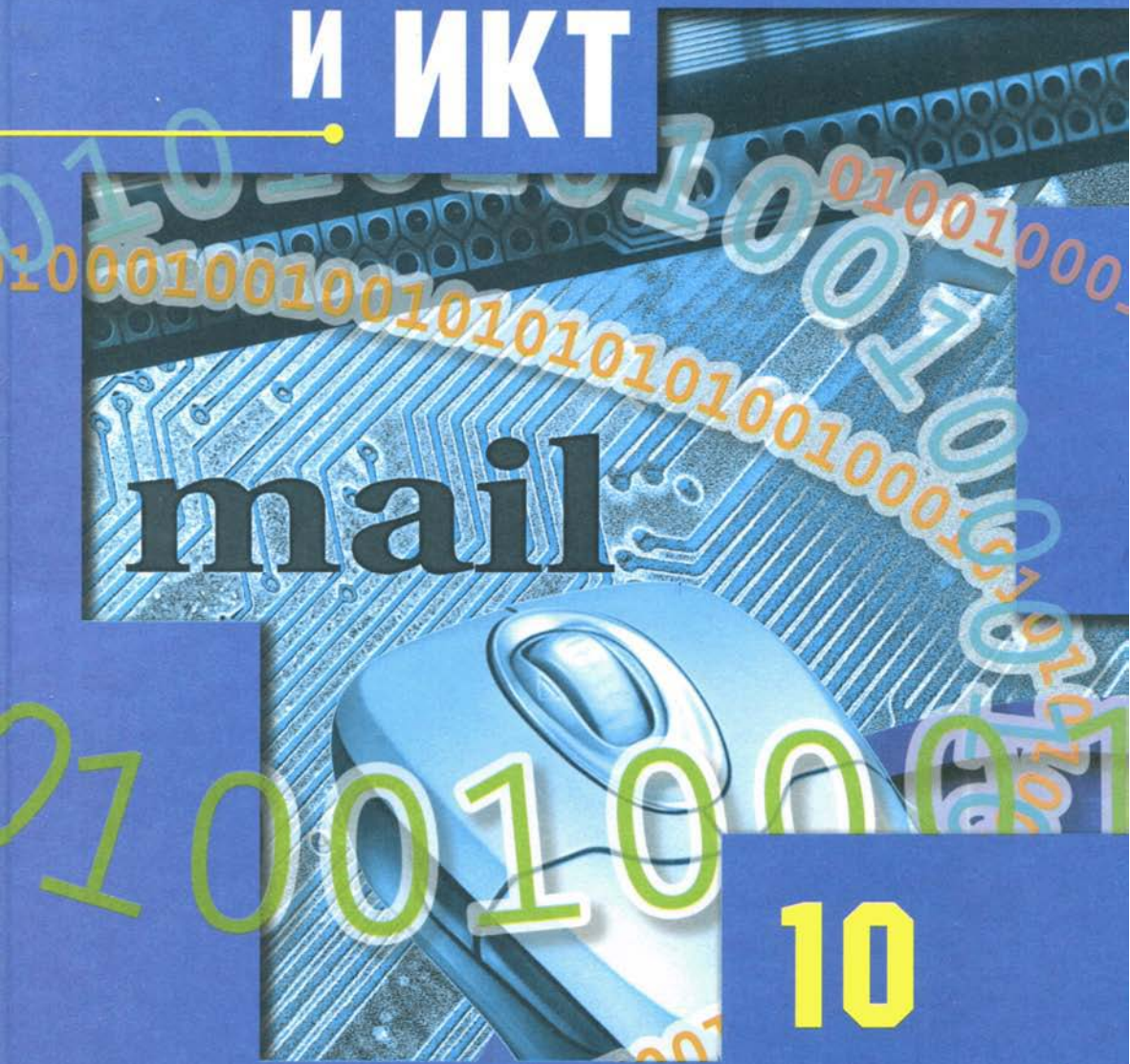
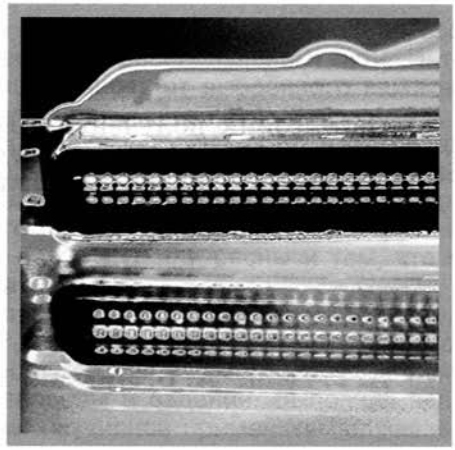
ИНФОРМАТИКА И ИКТ



10

ПРОСВЕЩЕНИЕ
ИЗДАТЕЛЬСТВО

ИНФОРМАТИКА и ИКТ



10 класс

Учебник
для общеобразовательных
учреждений

Базовый и профильный
уровни

Рекомендовано
Министерством образования и науки
Российской Федерации

2-е издание

Москва
«Просвещение»
2012

Уважаемые старшеклассники!

Вы переступили порог профильной школы и теперь смотрите на свое образование с учетом выбора дальнейшего профессионального пути. Интересы и глубина проникновения в предмет зависят от выбранного вами направления. Поэтому в учебнике материал представлен двумя уровнями — базовым и профильным. Материал профильного уровня предназначен для тех, кто выбрал математический и информационно-технологический профили или хочет изучать информатику как элективный курс в рамках иного профиля. Параграфы, относящиеся к этому уровню, обозначены рамкой [§ 12]. Впрочем, мы не исключаем, что они будут интересны и тем, кто не собирается углубленно изучать информатику.

В учебнике, кроме теоретического материала, приводятся описания лабораторных работ, которые вы будете выполнять в компьютерном классе. Невозможно обучение и без самоконтроля. Чтобы вы могли осуществлять его, учебник содержит разнообразные вопросы и задания. Выполняя задания, вы проверите, насколько хорошо вы научились применять полученные вами знания. Некоторые задания будут для вас простыми, другие окажутся сложнее. Самые трудные (разумеется, на наш взгляд) задания помечены знаком *. В конце каждой главы помещен ее конспект, который призван облегчить вам повторение пройденного, а также приведены тестовые задания, которые позволят проверить, как вами усвоен материал главы в целом.

В нашей книге, как и в любом учебнике, вам встретятся новые термины. Для удобства они напечатаны **жирным** шрифтом. Определения, свойства и правила заключены в рамку  или помечены слева знаком [i]. Заучивать свойства и правила наизусть совсем необязательно, но важно понимать их смысл и уметь применять на практике.

Для выполнения лабораторных работ вам будет нужен не только компьютер, но и комплекс программных средств — так сказать, «компьютерная поддержка» нашей книги. Описания этих программных средств приведены в учебнике. Но в вашем классе могут использоваться программы, отличающиеся от описанных в учебнике. В этом случае учитель объяснит вам, как выполнять лабораторные работы.

Желаем вам успехов в освоении самого современного учебного курса!



Информатика как наука

В школе вы изучаете различные предметы: математику, физику, химию, биологию, историю, информатику и др. Все они излагают основы соответствующих наук. А что такое наука? В толковом словаре С. И. Ожегова вы найдете следующее определение:

> **Наука** — система знаний о закономерностях в развитии природы, общества и мышления.

Система знаний излагается в виде теорий. В свою очередь, теория — это модель окружающей человека действительности, причем модель идеальная. Конечно, для построения теории и ее проверки создаются не только идеальные, но и материальные модели. Но материальные модели — это средства получения научных знаний, а собственно наука — совокупность идеальных моделей.

Одно из главных требований к системе научных знаний — их непротиворечивость друг другу, а также подтверждение научных выводов практикой, экспериментом.

Современная наука подразделяется на большое количество научных областей, каждая из которых имеет свой объект исследования. Вы знаете, что информатика изучает закономерности протекания информационных процессов в природе, обществе и технических системах. Как правило, научная область подразделяется еще на ряд научных дисциплин в зависимости от конкретного направления исследований. В информатике это исследования методов автоматической обработки информации и способов ее защиты, влияния информации и информационных технологий на общество (социальная информатика), возможностей компьютерной техники и т. д. В этой главе мы кратко охарактеризуем некоторые из указанных направлений информатики.

Фундамент любой научной дисциплины составляют понятия, с помощью которых описываются закономерности, составляющие сущность научного знания. Поэтому начинать знакомство с любой научной дисциплиной следует с освоения основных понятий, присущих данной области знания. Вот и мы начнем с определения основных понятий информатики. С некоторыми из них вы знакомы из курса информатики, который изучали в 8 и 9 классах.

§1 Информация

Представьте себе, что вы ничего не видите, ничего не слышите, не ощущаете ни запахов, ни холода, ни тепла, а вдобавок вся пища абсолютно безвкусная. Жизнь сразу бы потеряла для вас свою привлекательность. Ни музыки, ни книг, ни цветочных ароматов... И все потому, что вас лишили постоянного притока **информации**, без которого немыслимо нормальное существование человека в окружающем мире.

Попробуем разобраться подробнее с тем, что такое информация. Информация — понятие весьма не простое. Вы можете посмотреть один и тот же мультфильм с видеокассеты, с DVD-диска, с CD-диска, просто по телевизору в конце концов. Информация, полученная вами при просмотре фильма, будет одна и та же, но находится она на разных **носителях**. И в принципе нам все равно, на каком носителе находится информация, лишь бы было качественное ее воспроизведение (в данном случае нужного фильма).

Из курса физики вы знаете, что предметы вокруг нас характеризуются наличием в них вещества и энергии. Но, кроме всего прочего, любой предмет несет в себе еще и информацию. Даже самый маленький камешек, валяющийся на дороге, можно бесконечно изучать и изучать, получая информацию о его химическом составе, физических свойствах породы, особенностях его образования в земной коре и т. п.

И все же, говоря об информации, мы в первую очередь имеем в виду те сведения, которые предоставляют нам телевидение, радио, печатные издания. Информация аккумулирована в тысячах научных трудов, составляющих основу нашей цивилизации. Информация составляет основу школьных учебников, без которых трудно стать образованным человеком.

Информация — это сведения, знания об окружающем человека мире и о самом себе.

Так обычно разъясняется понятие информации в гуманитарных науках, где центральным объектом является человек как существо социальное.

Но есть и другая точка зрения на то, что такое информация. Понаблюдайте за поведением любого живого организма: муравья, пчелы, птицы, собаки. Вы убедитесь, что они тоже воспринимают информацию об окружающем мире и в соответствии с этой информацией выстраивают свое поведение. Более того, они обмениваются информацией друг с другом: муравьи — с помощью усиков; пчелы — исполнением определенных движений (называемых обычно танцем); птицы — пением, свистом, карканьем; собаки — лаем. Да и вы, отдавая приказы домашним животным, уверены, что они воспринимают передаваемую вами информацию.

Живые организмы от неживой природы отличает именно наличие у них целенаправленной деятельности, реализация которой без использования информации невозможна. Мы не предполагаем при этом, что это обязательно осознаваемая деятельность. Она может осуществляться и инстинктивно.

Информация нужна различным техническим устройствам для того, чтобы точно и правильно выполнять предписанную человеком работу. Важно, что такие устройства могут выполнять работу **автоматически**, т. е. без вмешательства человека. При этом они должны реагировать на внешние воздействия. Если, например, это станок автоматической обработки деталей, то он должен контролировать, поступила к нему очередная заготовка или нет, соответствует она требуемым исходным параметрам или нет и т. д. Если это современный фотоаппарат, то он способен по уровню освещенности автоматически установить выдержку и диафрагму, т. е. продолжительность воздействия света на фотопленку и диаметр открытого отверстия в объективе. Какое бы автоматическое устройство мы ни взяли, оно так или иначе использует информацию, получаемую от человека и окружающей среды.

Сформулируем теперь еще одну точку зрения на понятие информации.

Информация — это то, что позволяет живым организмам, их сообществам или техническим системам реагировать на окружающую среду, обеспечивая их целенаправленную деятельность.

Заметим, что цели техническим системам определяют люди, которые эти системы создали или используют.

Теперь представьте себя посередине пустыни. Вокруг вас совершенно ровная поверхность, не на чем глазу остановиться. Солнце в зените, даже тень расположилась прямо под вами. Прошло три часа. Ничего не изменилось, кроме тени. Она вытянулась в определенную сторону, и вы уже способны определить стороны света. Что же привело к появлению информации? Появление разнообразия в окружающей обстановке.

Информация — это отражение разнообразия в существующем мире.

Отсутствие разнообразия, когда неотличимы никакие два объекта, явления или процесса, — это и есть отсутствие какой бы то ни было информации. Такой весьма общий взгляд на понятие «информация» присущ многим естественным и философским наукам. В этом учебнике большей частью будет рассматриваться информация, которая выступает как человеческое знание.

Оторвитесь на минуту от учебника и выгляните в окно. Вы увидите дома и людей, солнце и небо, деревья и траву и автоматически переключитесь на анализ информации совсем другого сорта.

Она очень сильно отличается от однообразных черных строчек с буквами, какие вы видите в книге.

Информация, получаемая из сообщений, записанных с помощью букв, цифр и знаков, называется **символьной**; информация, получаемая с помощью зрительных образов окружающего мира, называется **видеоинформацией** (или **визуальной** информацией).

Человек создан природой как мощный «аппарат» по переработке видеоинформации. Вспомните пословицу «Лучше один раз увидеть, чем сто раз услышать», точно отражающую отличие в восприятии звуковой информации от видеоинформации. Однако современная цивилизация окружила нас бездонными морями символьной информации, способными поглотить любого в своих пучинах. Чтобы покорить океаны символьной информации, нужны «корабли», способные преодолеть океанские просторы. Ими и стали **компьютеры**, или, как их называли раньше, **электронно-вычислительные машины (ЭВМ)**.

Обратите внимание, что информация обязательно должна быть как-то зафиксирована. Результаты исследований облекаются в формулы, таблицы, графики. Генетическая информация фиксируется в генетическом коде, представляющем собой последовательность аминокислот. Обнаружение медоносов передается рисунком танца пчел. Информация о ландшафте местности фиксируется на планах и картах. Информацию, зафиксированную каким-либо способом, будем называть **информационным объектом**.

Конечно, кроме символьной, визуальной и звуковой информации, есть и другие виды представления информации. Информацию несет запах; осязание доставляет нам тактильную информацию. Через вкусовые рецепторы мы получаем информацию о качестве пищи и готовности ее к употреблению. Но чтобы сохранить информацию и сделать ее удобной для передачи другим людям, человек использует прежде всего символьную, визуальную и звуковую формы представления информации.

ВОПРОСЫ И ЗАДАНИЯ

1. Приведите примеры употребления термина «информация».
2. Норберт Винер, один из основоположников информатики как науки, говорил об информации так: «Информация — это обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему наших органов чувств». Сравните это определение с теми, которые приведены в тексте параграфа. Что у них общего и чем они различаются?
3. Может ли, на ваш взгляд, одна и та же информация содержаться в разных информационных объектах?
4. Какую информацию называют символьной? Что такое видеоинформация?



Норберт Винер

§2 Информационные процессы

Непрочитанная книга, конечно, содержит информацию, но она ничего не изменит ни в вас, ни в окружающем мире. Задача, которую никто не решает, тоже содержит информацию, но и она ни на что не повлияет. Другое дело, если книга прочитана, задача решена. Значит, не информация сама по себе, а те процессы, в которых она участвует и преобразуется, представляют интерес в первую очередь. Такие процессы называют **информационными**.

В предыдущем параграфе мы обсудили, что информация может использоваться только тогда, когда она представлена как информационный объект. Например, информация о тепловом состоянии какого-либо механизма может отображаться термометром в числовом виде (т. е. символьно), а может изображаться самописцем в графическом виде. Нередко приходится преобразовывать одну форму представления информации в другую. Процесс, в котором преобразуется форма представления информации, тоже относят к информационным.

Информационный процесс — это процесс, в ходе которого изменяется содержание информации или форма ее представления.

Как вы знаете, к основным видам информационных процессов относятся получение, передача, хранение и обработка информации.

Получение информации — это реализация способности к отражению различных свойств объектов, явлений и процессов окружающего мира. У высших животных и человека для получения информации имеются органы чувств, специализированные для восприятия информации определенного вида. Человек создал приборы — микроскоп и телескоп, термометры и тахометры и др., позволяющие извлекать ту информацию, которая недоступна ему в непосредственных ощущениях.

Получать информацию человек и другие живые организмы могут, не только наблюдая окружающий мир, но и общаясь друг с другом, а также из тех или иных источников информации. Такое получение информации происходит в результате информационного процесса другого вида — процесса передачи информации.

Передача информации всегда осуществляется по некоторому каналу связи от источника информации к ее приемнику (рис. 1.1). Канал связи можно уподобить транспортному средству, осуществляющему доставку информации от источника к приемнику. В истории человечества именно механическое движение долгое время было непременным участником процесса передачи информации. Почтовые кареты, почтовые вагоны поездов, авиапочта — все это каналы передачи информации, использующие механическое движение.

Однако для передачи информации люди издавна применяли и другие физические процессы. Звук и свет давно поставлены чело-

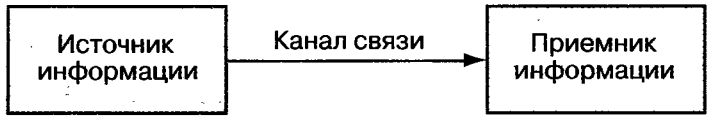


Рис. 1.1 Схема передачи информации

веком на информационную службу. Некоторые африканские племена до сих пор используют специальные барабаны (тамтамы), обмениваясь сообщениями со скоростью звука. На Руси, где леса гасят звук, для передачи срочных сообщений использовался дым костров. Ведь ни один гонец с пограничной заставы не успел бы вовремя предупредить горожан о набеге кочевников, чьи кони были не менее быстрыми.

Но с открытием радиоволн, а самое главное с изобретением устройств, способных их генерировать и улавливать, в деле передачи информации произошли революционные изменения. Теперь стало возможным в считанные секунды передавать информацию на сотни километров.

Хранение информации всегда предполагает наличие физического носителя. Камень, на котором первобытным человеком высекались рисунки, глиняные таблички Вавилона, бумага, магнитная лента, оптический диск — чем глубже проникал человек в тайны материи, тем больше информации ему удавалось сохранять в единице объема физического тела. Впрочем, борьба шла не только за увеличение емкости носителя информации, но и за надежность ее сохранения. Методы защиты информации возникли в глубокой древности, а сейчас это бурно развивающаяся область информатики.

Обработка (преобразование) информации тоже достаточно широко понимаемый информационный процесс. Под обработкой информации понимают получение новой информации из уже имеющейся или преобразование формы представления информации.

Детектив собрал улики и указал, кто совершил преступление. Математик сопоставил известные ему утверждения и доказал новую теорему. Д. И. Менделеев на основе информации о химических свойствах элементов сформулировал периодический закон. Во всех этих примерах в результате обработки информации появилась новая информация.

Обработкой информации следует признать и вычисление суммы двух чисел: ведь из двух известных данных получается новое, до того неизвестное. Обработкой информации является, например, перевод предложения с русского языка на иностранный.

Между процессами обработки информации, указанными в двух предыдущих абзацах, существует разница. Главное отличие в том, что для розыска преступника или доказательства новой теоремы нет и не может быть указано жестких правил, как должна обрабатываться исходная информация. Как говорят, человек в этих случаях действует **эвристически**. Складывая два числа, мы уже руко-

водствуемся жестко указанными правилами. Такую работу можно поручить техническому устройству, которое способно понимать и исполнять предписанную ему инструкцию. Устройства, управляемые с помощью инструкций и выполняющие свою работу автоматически, называют программируемыми и говорят, что свою работу они исполняют **формально**. В частности, можно говорить и о формальной обработке информации. Исполнитель, производящий такую обработку, не должен вникать в смысл выполняемых им действий; поэтому формальная обработка информации, как правило, касается изменения формы ее представления, а не содержания.

Однако, каким бы ни был способ обработки информации — формальным или эвристическим, существует нечто или некто, выполняющие эту обработку. Его обычно называют **исполнителем**.

В информатике из всех формальных исполнителей особое внимание уделяется компьютеру. Это техническое устройство было изобретено, чтобы в первую очередь облегчить человеку выполнение вычислительных работ. Развитие информатики и совершенствование технической базы привело к пониманию, что компьютер вопреки своему названию — вычислитель — является универсальным инструментом обработки информации.

ВОПРОСЫ И ЗАДАНИЯ

1. Что называют информационным процессом? Назовите основные виды информационных процессов.

2. Ниже перечислено несколько процессов:
   а) производство бензина из нефти;
   б) измерение температуры воздуха;
   в) перевод единиц длины из метров в сантиметры;
   г) движение Земли вокруг Солнца;
   д) фотографирование обратной стороны Луны;
   е) выплавка стали;
   ж) перевод текста с английского языка на русский;
   з) запись решения математической задачи в тетрадь;
   и) исправление ошибок в сочинении;
   к) приготовление обеда;
   л) увеличение размеров тела при нагревании;
   м) составление астропрогноза;
   н) фотосинтез.

   Выделите из перечисленных процессов информационные и укажите, к какому виду они относятся. Решение оформите в виде таблицы.

Информационный процесс				Неинформационный процесс
Получение информации	Передача информации	Хранение информации	Обработка информации	

3 Приведите примеры: получения, хранения, передачи и обработки информации.

4 Приведите примеры информационных процессов, в которых изменяется:
а) содержание информации;
б) форма представления информации.

5 Человек изобрел много различных способов хранения информации. Какие из них вы могли бы назвать?

6* Опираясь на известные вам сведения из биологии, укажите, какие формы представления информации используют животные для ее сохранения и передачи другим животным. Приведите примеры тех форм, которые не встречаются в человеческой практике.

7 а) В объяснительном тексте упоминается несколько приборов, изобретенных человеком для расширения своих возможностей по получению информации. Приведите примеры еще каких-нибудь приборов, созданных для той же цели.
б) Приведите примеры извлечения живыми существами информации об окружающей среде, которые невозможны для человеческих органов чувств.
в) Человек для расширения своих возможностей по получению информации использует не только приборы, но и способности к этому животных. Приведите примеры такого использования.

8 Существует легенда о том, как древнегреческий ученый Архимед, открыв закон о выталкивающей силе жидкости (он знаком вам из курса физики), воскликнул: «Эврика!» От этого слова произошли выражения «эвристическое решение» и «эвристическая обработка информации». Внезапное озарение, интуитивное понимание сути проблемы и ее решение — вот широкое толкование слов «эвристическое решение». Более узкое их значение — отыскание решения без осуществления полного перебора всех возможных вариантов. Для конкретной ситуации далеко не всегда можно точно сказать, производится в данной ситуации формальная или эвристическая обработка информации. Это зависит, в частности, от знаний того, кто ведет обработку информации. Гроссмейстер, например, в целом ряде шахматных ситуаций может действовать формально, опираясь на знание шахматной теории, в то время как шахматному новичку почти непрерывно приходится руководствоваться интуицией.
а) В приведенных ниже примерах укажите преимущественный, на ваш взгляд, вид обработки информации — формальный или эвристический:
- решение математической задачи;
- измерение температуры у больного;
- определение очередного хода в середине шахматной партии;
- нахождение значения функции $f(x) = x^2 - \sin 2x + \ln(x + 1)$ в точке $x = 0$.

б) Вспомните свою обычную ежедневную деятельность по обработке информации. С каким видом обработки — формальным или эвристическим — вам больше приходится иметь дело? Рассмотрите с той же точки зрения вашу учебную деятельность на уроках по разным предметам.

9 При передаче информации канал связи может быть подвержен воздействиям, в результате которых передаваемая информация будет искажена. Такие воздействия называются **помехами** и могут возникать в силу технических причин, а иногда создаваться специально с целью дезинформации.
а) Приведите примеры помех, которые могут возникать в силу физических особенностей того или иного канала связи.
б) В «Сказке о царе Салтане...» А. С. Пушкин описывает процесс передачи информации от жены царю и обратно отправкой гонца с соответствующими сообщениями. Какие помехи этого канала связи описаны поэтом?

§3 Язык как средство сохранения и передачи информации

В § 1 мы отметили, что одним из способов, позволяющих фиксировать информацию, является ее представление в символьном виде. Само появление символьной информации связано с формированием человеческой речи. Первобытный человек каждому новому предмету или явлению придумывал имя. Чем глубже человек познавал окружающий мир, тем больше имен ему требовалось. Но человек способен произносить и различать на слух не так уж много звуков-фонем. Чтобы получить необходимое разнообразие имен, он начал комбинировать звуки друг с другом, получая слова. Так появилась членораздельная речь, а впоследствии — письменность. В их основе лежит гениальная идея конечного алфавита, т. е. некоторого фиксированного упорядоченного набора знаков, из которого можно составлять как угодно много слов. В информатике **словом** принято называть любую последовательность символов заданного алфавита. Совокупность слов, используемых для записи информации в рамках какой-либо предметной области, и называют **языком**.



Рис. 1.2 Знаки дорожного движения

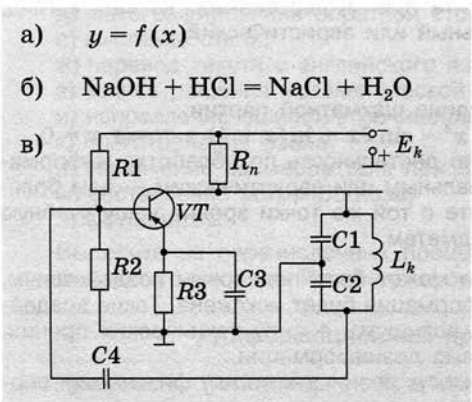


Рис. 1.3 Примеры фраз языков математики (*а*), химии (*б*), электроники (*в*)

В своей практике человек использует самые разнообразные языки. Прежде всего это языки устной и письменной речи. Довольно часто используются языки различных указателей. К таким относятся, например, знаки дорожного движения (рис. 1.2). Кроме того, человек использует ряд языков профессионального назначения. Сюда относятся языки математических и химических формул, обозначения, принятые в электронике (рис. 1.3) и т. п.

В естественных языках человеческого общения такие языки называют **коммуникативными** — смысл слова нередко зависит от его окружения, или, как говорят, от **контекста**. Сравните смысл слова «сопротивление» в двух следующих предложениях: «Сопротивление наступающей армии усилилось» и «Электрическая цепь имеет сопротивление величиной 1 Ом». В первом случае сопротивление — это весьма общее и, быть может, весьма раз-

народное воздействие на противника, во втором случае сопротивление — это указание на физическое свойство электрической цепи.

Именно неоднозначность смысла слова, его зависимость от контекста позволяет писателям создавать художественные шедевры. Она порождает ассоциации, воздействует на эмоции читателя, позволяет контекстными связями создавать целостную картину из мозаики отдельных слов.

Но такая неоднозначность смысла не годится, когда необходимо описать какое-либо явление в научных целях. В этом случае за каждым словом закрепляется одно смысловое значение. Процесс закрепления за словом одного смысла называется **формализацией**. А язык, в котором каждое слово имеет единственное значение, называется **формализованным**. Язык любой науки является формализованным.

Вопросы соотнесения слову его значения и смысла составляют **семантику** языка. Однако, кроме проблемы соотнесения слову того или иного значения, есть еще проблема образования самих слов — **синтаксис** языка.

Со временем люди обнаружили, что бывают множества, содержащие бесконечно много элементов, например множество натуральных чисел. Каждому элементу нужно сопоставить обозначающее его слово. Но бесконечный список слов не составишь, нужно придумать что-то иное. Впрочем, и для конечного, но достаточно большого множества именование каждого объекта может оказаться непростой задачей (к примеру, образование названий органических соединений в химии). Решение возникающей здесь проблемы было найдено людьми давно: можно задать правила образования слов. В явном виде эту мысль первым высказал, по-видимому, Архимед в своем трактате «Исчисление песка». В нем он изложил систему именования каких угодно больших натуральных чисел.

Решение проблемы именования элементов бесконечного множества мы как раз и проиллюстрируем на хорошо знакомом вам множестве натуральных чисел. Десять цифр 0, 1, 2, ..., 9 — это алфавит, и каждое натуральное число обозначается словом, записанным знаками этого алфавита. Запись первых девяти натуральных чисел — это сами знаки данного алфавита. Запись всех последующих чисел получается по единому (хорошо известному) правилу. Вот оно.

Пусть для числа n уже имеется запись указанными цифрами. Если последняя цифра отлична от 9, то для получения записи следующего числа $n + 1$ нужно эту последнюю цифру заменить следующей по алфавиту, а все другие цифры оставить без изменения. Если же последней цифрой числа n является 9, то она заменяется на 0, и если в старших разрядах цифры отсутствуют, то перед этим 0 пишется 1, в противном случае по уже описанному правилу меняются цифры в более старших разрядах.

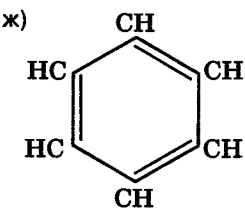
Правила, согласно которым из знаков алфавита образуются слова данного языка, называются **грамматикой** этого языка. Так

что приведенное выше правило — это грамматика десятичной нумерации. Язык десятичной нумерации — это множество записей всех натуральных чисел. Чтобы записывать числовую информацию, выражаемую дробями, алфавит приходится расширить, добавив в него знак дробной черты или десятичной запятой (иногда вместо запятой используется точка). Соответствующим образом модифицируется и грамматика языка.

Возникновение некоторых алфавитов было продиктовано необходимостью привлечения технических средств для передачи информации. Одним из примеров такого алфавита является азбука Морзе, изобретенная для передачи телеграфных сообщений. В ней каждый знак обычного алфавита кодируется набором точек и тире (что соответствует передаче коротких и длинных электромагнитных импульсов). Например, в русской версии азбуки Морзе буква «А» кодируется сочетанием • —, буква «С» — сочетанием • • • и т. д.

Использование алфавита, состоящего из двух символов, оказалось с технической точки зрения наиболее удобным. Наличие тока или его отсутствие, наличие намагниченности или ее отсутствие, наличие света или его отсутствие — два хорошо различимых состояния кодируют два символа алфавита. Мы в дальнейшем обычно будем считать, что двухсимвольный алфавит состоит из 0 и 1. В противном случае состав двухсимвольного алфавита будет оговариваться специально.

ВОПРОСЫ И ЗАДАНИЯ

1. В чем заключается процесс формализации языка?
2. Что называют грамматикой языка?
3. Даны слова: картофель, кабачок, морковь, карандаш, капуста. Укажите, какое из этих слов лишнее, если руководствоваться:
   а) семантическим подходом; б) синтаксическим подходом.
4. Для каждой записи из приведенных ниже укажите, на каком языке она сделана — коммуникативном или формализованном. Попытайтесь определить область применения каждого из использованных языков:
   а) Выхожу один я на дорогу.
   б) To be or not to be.
   в) $2KOH + H_2SO_4 = K_2SO_4 + 2H_2O$.
   г) Сяпала калуша по напушке и увазила бутявку,
   и волит: «Калушата, калушаточки! Бутявка!»
   д) $(a + b)^2 = a^2 + 2ab + b^2$.
   е) $F = ma$.
   ж)
   

   з) PRINT X.
   и)* Количество вещества, выделяющегося на электродах, прямо пропорционально количеству электричества, проходящего через электролит.
   к)* Если x_1 и x_2 — корни уравнения $x^2 + px + q = 0$, то $x_1 + x_2 = -p$ и $x_1x_2 = q$.

5* В большинстве языков слова — последовательности знаков (слева направо, справа налево, сверху вниз). И в информатике слово — это тоже последовательность символов алфавита, записанная по европейской традиции слева направо. Однако в некоторых языках используется планарное или даже пространственное расположение алфавитных знаков. Всем известный пример планарного расположения знаков — запись мелодии нотами: сам нотный знак несет информацию только о длительности звучания ноты, а его расположение на нотном стане — информацию о высоте звука. Приведите еще примеры планарных языков, используемых человеком.

§4 Универсальность двоичного кодирования

В предыдущем параграфе мы отметили, что с технической точки зрения удобно иметь дело с информацией, закодированной символами двоичного алфавита. Компьютер как техническое устройство обрабатывает информацию, тоже записанную в двоичном алфавите. Обсудим, как же основные виды информации — символьная, визуальная и звуковая — могут быть записаны всего лишь двумя символами.

Проще всего дело обстоит с символьной информацией. Она уже представлена записью в некотором алфавите. Поэтому достаточно каждый символ этого алфавита закодировать некоторым словом в двухсимвольном алфавите. Слов, состоящих из одного символа такого алфавита, только 2. Двухсимвольных слов уже 4 — это 00, 01, 10 и 11. Если рассматривать последовательность из четырех символов, то этого достаточно для записи 16 слов. В латинском алфавите 26 букв, а пятисимвольных слов из 0 и 1 — 32, т. е. достаточно, чтобы каждой букве латинского алфавита присвоить свой пятизначный код.

Но ведь буквы бывают не только строчные, но и прописные. Чтобы их различать, требуется дополнительный символ в коде. А ведь есть еще цифры, знаки препинания и арифметических действий, специальные знаки (например, %, №) и, конечно, особо популярный во всех текстах символ «пробел».

Слов из семи символов хватает для того, чтобы закодировать сообщения не только на английском, но и на хорошем русском языке. Именно таков отечественный код КОИ-7. Сокращение КОИ родилось из первых букв словосочетания «код обмена информацией».

Чтобы не употреблять длинный оборот «последовательность из стольких-то знаков, каждый из которых нуль или единица», люди договорились появление одного такого знака в последовательности называть **битом** (от английского BI*nary digi*T — двоичная цифра). Теперь можно сказать, что КОИ-7 — это *семибитное* кодирование русскоязычных сообщений.

В большинстве первых компьютеров использовался семибитный код. Однако с развитием техники это стало неудобно. Новый код был уже восьмибитным и основывался на американском стан-

дартном коде информационного обмена (ASCII — American Standard Code for Information Interchange).

Последовательность восьми битов договорились называть словом **байт**. Но и один байт окажется мал, если требуется оценить, сколько места в памяти компьютера занимает, скажем, десяток страниц текста. Поэтому были введены более крупные единицы — **килобайт** (обозначение Кбайт), **мегабайт** (Мбайт), **гигабайт** (Гбайт), **терабайт** (Тбайт) и т. д. Соотношения между ними таковы:

1 Кбайт = 1024 байт;
1 Мбайт = 1024 Кбайт;
1 Гбайт = 1024 Мбайт;
1 Тбайт = 1024 Гбайт.

Конечно, в современном мире, опутанном компьютерными сетями, даже восьмибитного кодирования недостаточно: есть же арабский алфавит, два японских, хинди, математическая символика и т. д. Поэтому был введен новый стандарт символьного кодирования — UNICODE, где каждый символ кодируется уже двумя байтами.

Из рассказанного ясно, что компьютер оказался хорошо приспособленным для обработки символьной информации, при этом неважно, текстовая она или числовая. Однако посмотрим на процесс получения и восприятия информации с другой точки зрения. В тексте человек может не заметить неправильно употребленную букву (кто из вас не делал подобных ошибок!), но, как правило, реагирует на мельчайшие изменения видимой ему картинки. Не случайно на некоторых особо опасных производствах для операторов выводят на экран не только показания датчиков, но и изображение человеческого лица, которое так или иначе отображает, в какой степени нормально протекает производственный процесс. Оператор намного быстрее реагирует на изменение лица, чем на безликие показания датчиков.

Современный компьютер пока остается устройством, обрабатывающим лишь информацию, закодированную двоичным кодом. Поэтому и видеоинформацию надо представить в двоично закодированном виде. Поступают так.

Пусть речь идет о черно-белом рисунке. Представим его изображенным на прямоугольном листе бумаги (рис. 1.4, *а*). Разобьем этот лист на квадратики с достаточно маленькой стороной (рис. 1.4, *б*). А теперь про каждый квадратик мы можем сказать, есть в нем какая-то часть рисунка или нет. В первом случае присваиваем квадратику код 1, во втором — код 0. Пробегая по всем квадратикам, например, слева направо и затем переходя по строкам сверху вниз, мы получаем двоичный код рисунка (рис. 1.4, *в*). Восстановление рисунка, т. е. декодирование, показано на рисунке 1.4, *г*.

Конечно, результат декодирования почти наверняка не совпадет с исходным рисунком. Но для того мы и выбирали размеры квадра-

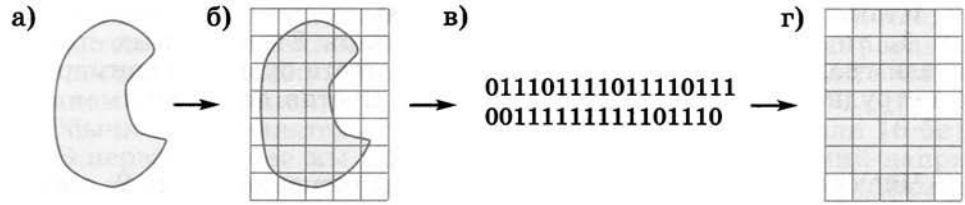


Рис. 1.4 Схема кодирования рисунка для записи его в памяти компьютера

тиков достаточно маленькими, чтобы искажения были незначительными. Кроме того, как вы знаете из биологии и физики, и человеческий глаз, и любой другой оптический прибор имеет ограниченную разрешающую способность в восприятии изображения, т. е. перестает различать соседние участки как различные, если их размеры меньше некоторой величины. В этом случае отдельные квадратики в восприятии человека опять сливаются в непрерывное целое.

А как быть, если требуется закодировать цветной рисунок? Ведь там для описания цвета квадратика двоичным кодированием не обойтись. И здесь природа снова подсказывает решение.

Хорошо известно, что любой цвет может быть получен как смесь трех основных цветов — красного, синего и зеленого. Из таблицы 1.1 вы легко определите, какие получаются цвета при таком смешивании; в ней 1 означает наличие данного цвета, 0 — его отсутствие.

Таблица 1.1

Красный	Синий	Зеленый	Цвет
0	0	0	Черный
0	0	1	Зеленый
0	1	0	Синий
1	0	0	Красный
0	1	1	Бирюзовый
1	0	1	Желтый
1	1	0	Фиолетовый
1	1	1	Белый

Здесь мы имеем только две градации яркости — 1 или 0. Но цвет может иметь разную степень яркости; тогда появится и соответствующий оттенок получаемого цвета.

Итак, если у нас только две градации яркости, кодируемые 0 и 1, то мы можем получить всего лишь 8 различных цветов. А если градаций для каждого основного цвета будет, например, 64, то нетрудно подсчитать, что различных цветов получится

$$64^3 = 262\,144.$$

Число 64 удобно тем, что оно является степенью числа 2, а именно шестой степенью, поэтому указанное число градаций кодируется шестибитовыми последовательностями, и, значит, для кодирования 262 144 цветов потребуется 18 бит.

В отличие от графической информации, кодирование звука оказалось довольно-таки простым. С XIX века известны устройства, которые улавливают звуковые волны и преобразуют их в электрический ток. Устройства эти называются микрофонами. Точно такой же непрерывный, по-другому **аналоговый**, сигнал подается и в звуковые усилители, а затем на динамики — для воспроизведения звука. Задача состоит в том, чтобы придумать способ кодирования с помощью чисел аналогового сигнала, например изображенного на рисунке 1.5.

Для этого достаточно несколько раз (несколько десятков, сотен, тысяч раз) в секунду измерять величину аналогового сигнала и кодировать ее. Фактически мы разбиваем плоскость вертикальными и горизонтальными линиями и приближенно считаем, что график проходит в точности через узлы получившейся сетки, заменяя плавную линию ломаной (рис. 1.6).

Разбиение вертикальными линиями называется **дискретизацией** (по времени) аналогового сигнала и характеризуется **частотой дискретизации**. Для реального процесса это означает, как часто мы измеряем величину аналогового сигнала.

Обычно для кодирования звука выбирают одну из трех частот — 44,1 КГц, 22,05 КГц или 11,025 КГц. Первая из этих частот соответствует музыке, записываемой обычно на компакт-диск. Величина аналогового сигнала измеряется в этом случае 44 100 раз в секунду. Естественно, что в случае стереозвучания отдельно кодируется левый и правый канал. Возвращаясь к рисунку 1.6, можно сказать, что расстояние между вертикальными линиями соответствует $\frac{1}{44\,100}$ доле секунды.

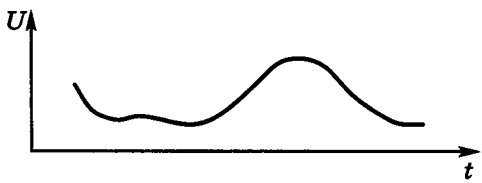


Рис. 1.5 Изображение аналогового сигнала в виде графика

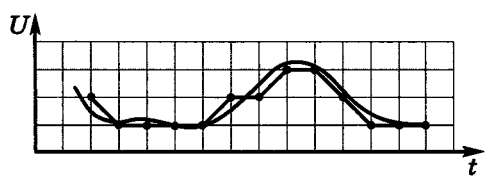


Рис. 1.6 Оцифровка аналогового сигнала

Важно и разбиение горизонтальными линиями. Чем мельче сетка, тем качественнее мы приблизим результат кодирования к аналоговому сигналу. Горизонтальная разбивка называется **квантованием** (по уровню).

Обычно применяется всего два его типа — 8-битное или 16-битное. В первом случае мы получаем 256 различных значений напряжения, во втором — уже известные вам 65 536 значений. Иными словами, интервал от нулевого до максимального напряжения разбивается либо на 256, либо на 65 536 уровней.

8-битное квантование часто применяется для оцифровки обычной речи (в частности, телефонного разговора) и радиопередач на коротких волнах; 16-битное — для оцифровки музыки и УКВ-радиопередач.

Заметим, что преобразование аналогового сигнала в цифровой формат всегда сопряжено с некоторым искажением исходного сигнала. Именно поэтому некоторые ценители музыки до сих пор предпочитают компакт-дискам добрые старые пластинки.

Подсчитаем, сколько будет занимать в килобайтах одна минута записи на музыкальном компакт-диске. При 16-битном кодировании сигнала потребуется 2 байта на запись значения напряжения. Поскольку сигнал меряется 44 100 раз в секунду, потребуется $2 \times 44\,100 = 88\,200$ байт каждую секунду или примерно 86,1 Кбайта. За минуту это составит $86{,}1 \times 60 = 5166$ Кбайт или примерно 5,05 Мбайт. Поскольку сигнал стереофонический, это значение нужно умножить на 2. Проведя еще несколько аналогичных вычислений, получим таблицу 1.2, в которой указано, какой объем будет занимать закодированная таким образом одна минута звуковой информации.

Таблица 1.2

Тип сигнала / Частота дискретизации	16 бит, стерео	16 бит, моно	8 бит, моно
44,1 КГц	**10,1 Мбайт**	5,05 Мбайт	2,52 Мбайт
22,05 КГц	5,05 Мбайт	2,52 Мбайт	1,26 Мбайт
11,025 КГц	2,52 Мбайт	1,26 Мбайт	*630 Кбайт*

Объем, обеспечивающий самое высокое качество, соответствующее записи на музыкальном компакт-диске, выделен жирным шрифтом. Объем, достаточный для записи телефонного разговора, выделен курсивом. В частности, именно такой объем памяти требуется автоответчику, программно установленному на вашем компьютере, для записи входящих звонков прямо в файл.

Огромный размер получаемых файлов достаточно долгое время был серьезным препятствием для пересылки музыки по электрон-

ной почте и размещения ее в Интернете. Поэтому было предложено несколько способов их уменьшения. Так, большинство людей не слышат очень высокие звуки, и информацией о таких звуках вполне можно пожертвовать. Кроме того, человек воспринимает звуки с эффектом «послезвучания», т. е. даже после прекращения громкого звука он как бы слышит его. Стало быть, начало тихих звуковых фрагментов, следующих за громкими, тоже можно исключить. И таких ситуаций довольно много.

Следовательно, зная законы восприятия звука человеком, можно *сжать* информацию, уменьшив размер файла с музыкальной информацией в несколько раз. Конечно, качество музыки после такой операции ухудшится, но, во-первых, для большинства людей это будет практически незаметно, во-вторых, зачастую это единственный способ передавать такие файлы по компьютерным сетям, а в-третьих, вовсе не все равно, сколько музыки можно разместить в памяти музыкального плеера — на один час или на десять! Ради такого можно согласиться и на некоторое ухудшение качества. Наиболее популярными форматами сжатых музыкальных файлов сейчас являются форматы MP3 и WMA. Кстати, большинство цифровых телефонных автоответчиков (о которых мы говорили выше) и цифровых диктофонов как раз обрабатывают звук с целью сжатия информации в памяти.

ВОПРОСЫ И ЗАДАНИЯ

1. Каков коэффициент пересчета байтов в килобайты; килобайтов в мегабайты? А коэффициент пересчета битов в байты? Сколько байтов в одном мегабайте?
2. Посчитайте, сколько примерно байт содержит одна страница вашего учебника. Выразите полученное число в килобайтах.
3. Пусть стрелка ↑ означает перемещение на одну клетку вверх, стрелка ↓ означает перемещение на одну клетку вниз, стрелка ← — перемещение на одну клетку влево, стрелка → — перемещение на одну клетку вправо.

   а) Закодируйте последовательностью стрелок кратчайший маршрут из клетки *A* в клетку *B* на клетчатом поле с перегородками, изображенном на рисунке 1.7, *а*.

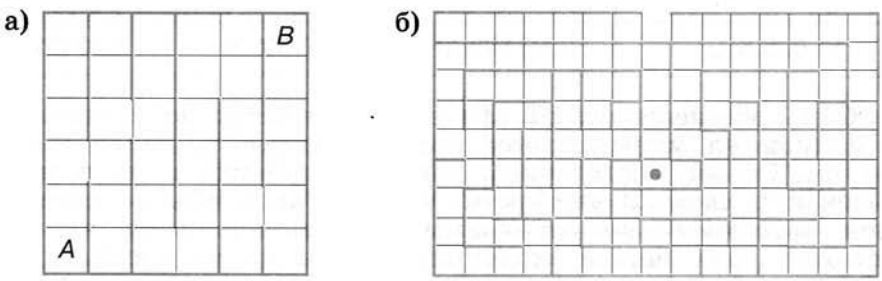


Рис. 1.7

(За один ход можно переместиться ровно на одну клетку, при этом запрещается проходить «сквозь» перегородки.)

б) Закодируйте последовательностью стрелок путь из центра до выхода в лабиринте, изображенном на рисунке 1.7, *б*.

в) Сколько бит содержит сообщение о маршруте из клетки *A* в клетку *B*, о котором шла речь в задании *а*? А в задании *б*?

4. Как получается ярко-белый цвет на экране цветного монитора?

5. Что такое дискретизация аналогового сигнала?

6. По таблице 1.2 определите, во сколько раз объем файла со звуковой записью высшего качества больше объема файла с записью на диктофон.

7. На чем основывается сжатие звуковых файлов?

§5 Информационное моделирование

Академик А. П. Ершов, которому школьная информатика обязана своим существованием, писал: «Как самостоятельная наука информатика вступает в права тогда, когда в рамках соответствующей частной науки строится информационная модель того или иного фрагмента действительности... Поэтому информационная модель выступает как средство сопряжения информатики с этими науками».

Само слово «модель» пришло из Франции, где первоначально означало образец, дающий наглядное представление о каком-либо объекте или явлении. И сегодня многие модели, используемые человеком, нацелены на демонстрацию того, что, быть может, не так легко увидеть в природе. К примеру, глобус дает нам наглядное представление о Земле, а макет кристаллической решетки — о строении вещества. Изучая курс информатики в предыдущих классах, вы познакомились с современным понятием модели.

Моделированием называют замену одного объекта, процесса или явления другим объектом или процессом, сохраняющим существенные свойства исходного объекта, процесса или явления, с целью изучения этих свойств, прогнозирования возможных изменений и на этой основе использования их в деятельности человека. Сам заменяющий объект или процесс называют **моделью**.

Какие именно свойства должны присутствовать в модели, определяется **целями моделирования**, т. е. тем, для решения каких задач предполагается использовать данную модель. В таблице 1.3 приведены некоторые модели, о которых вы наверняка уже имеете представление.

Таблица 1.3

№ п/п	Моделируемый объект или процесс	Задача, для решения которой осуществляется моделирование	Модель
1	Земная поверхность	Выбор маршрута движения	Карта
2	Механизм	Изготовление	Чертеж
3	Человек	Демонстрация одежды	Манекен
4	Человек	Идентификация как гражданина страны	Паспорт
5	Человек	Установление профессиональной квалификации	Аттестат, диплом или свидетельство
6	Человек	Представление истории жизни	Биография
7	Движение тела, брошенного вверх	Изучение зависимости высоты полета тела от времени полета	Формула $h = h_0 + v_0 t - \frac{gt^2}{2}$
8	Радиоактивный распад	Изучение зависимости массы вещества от времени	Формула $m = m_0 \cdot 2^{-\frac{t}{t_0}}$
9	Работа железнодорожной станции	Выбор времени и поезда для поездки	Расписание движения поездов
10	Работа светофора	Управление дорожным движением	Алгоритм переключения цветовых сигналов
11	Вещество	Изучение химического состава вещества	Формула, показывающая, сколько и каких атомов содержится в молекуле вещества, например H_2O

Если мы рассмотрим приведенный список моделей, то увидим, что они весьма разнообразны. Некоторые из них являются физическими копиями моделируемых объектов (например, манекен), другие — словесным описанием (например, биография). Но особую роль играют модели, в которых моделируемый объект описывается с помощью системы параметров и связей между ними. Сами параметры

и связи между ними описываются на подходящем формализованном языке, нередко с применением математической символики.

Модель, представляющая объект, процесс или явление формализованным описанием параметров и связей между ними, называется **информационной моделью**.

Информационная модель, в которой параметры и связи между ними выражены в математической форме, называется **математической моделью**.

Информационная модель, в которой параметры и связи между ними выражены комплексом программ, реализованных на компьютере, называется **компьютерной моделью**.

Построение информационной модели проходит несколько этапов:
- выделение факторов, существенных для достижения той цели, для которой производится построение модели;
- определение средств реализации модели, т. е. языка и исполнителя;
- описание факторов с помощью параметров (формализация);
- установление связей между параметрами;
- проверка полученной модели на адекватность.

Адекватность модели означает, что цель моделирования достигнута и данная модель при решении с ее помощью задач определенного класса дает удовлетворительный ответ. В частности, адекватность модели означает, что при ее построении действительно были учтены все существенные факторы. Высказывание: «Гладко было на бумаге, да забыли про овраги, а по ним ходить» — уже давно превратилось из утверждения о неадекватности построенной модели для выбора маршрута в пословицу о неправильно выделенных факторах моделирования в целом. Неправильно проведенная формализация или неверно установленные связи между параметрами тоже сделают модель непригодной для использования, но в этом случае обычно говорят об ошибочности внутри самой модели. К примеру, при разработке компьютерной модели вполне возможны ошибки алгоритмизации или программирования. Они устраняются с помощью тестирования.

Адекватность модели можно проверить только ее согласованностью с общими теоретическими положениями и экспериментальными данными. Поэтому на заключительной стадии построения модели всегда проводят эксперимент, подтверждающий или опровергающий адекватность созданной модели. Этот эксперимент может быть натурным, т. е. воспроизводятся реальные условия и сравниваются результаты, полученные физическим путем, с результата-

ми, полученными с помощью модели. Конечно, если речь идет о построении модели колебаний маятника или даже полета снаряда, то можно провести натурный эксперимент. Но человек начинает располагать средствами такой разрушительной силы и ставить перед собой такие задачи глобального вмешательства в природу, что натурные эксперименты могут привести к катастрофическим последствиям. Ныне на смену натурным экспериментам пришли компьютерные, которые позволяют предсказать опасные последствия тех или иных действий человека. Конечно, надо все равно уметь проверять, дают ли используемые в этих экспериментах компьютерные модели адекватную картину. Разработкой методов проверки адекватности занимается современная информатика.

В последующих главах вы будете иметь дело с разнообразными информационными моделями. Некоторые модели будете конструировать сами и самостоятельно проверять их адекватность.

ВОПРОСЫ И ЗАДАНИЯ

1. Что означает термин «моделирование»?
2. Что такое модель объекта, процесса или явления?
3. Что такое информационная модель? Чем информационная модель отличается от других видов моделей?
4. Какая модель называется математической? Укажите, какие из моделей, перечисленных в таблице 1.3, являются математическими.
5. Что такое компьютерная модель?
6. Перечислите этапы построения информационной модели.
7. В чем состоит свойство адекватности модели?
8. Какие факторы учтены в модели вертикального движения тела (см. табл. 1.3)? Какими параметрами эти факторы описаны?
9. Какие факторы учтены в паспорте как модели (см. табл. 1.3)? Какими параметрами эти факторы формализованы?

§6 Системный подход в моделировании

Представьте, что у вас дома появился новый книжный шкаф. Еще недавно он был просто набором досок, ручек, дверных петель, стекол, винтов. А теперь стоит, готовый принять книжные сокровища. Набор всех частей остался прежним, но теперь каждая деталь нашла свое место в соответствии с теми связями, которые предписаны этим деталям дизайнером. Благодаря этим связям набор элементов стал обладать новым свойством. Появление нового свойства благодаря реализации связей между элементами называется **системным эффектом** или, по-другому, **эмерджентностью.**

Не только книжный шкаф, но и многие реальные объекты можно представить как совокупность элементов, связанных между собой так, что имеет место системный эффект: никакая часть по отдельности не обладает тем свойством, которым обладает целостный объект. В таком случае говорят, что объект представляет собой **систему**. Впрочем, системой может быть не только объект, но и процесс, если мы рассматриваем его как соединение частей, в результате которого возникает свойство, не присущее никакой его части. К примеру, процесс доставки писем от отправителя к адресату представляет собой организацию пунктов приема писем, сортировку писем по узлам связи, транспортировку писем от узлов связи отправителей на узлы связи адресатов и, наконец, доставку писем адресатам.

Рассмотрение объекта или процесса как системы оказалось весьма продуктивным методом исследования и конструирования. Сформулированный в 1950 году американским биологом Л. фон Берталанфи этот метод получил название **системного подхода** и затем развивался в самых различных направлениях. До Берталанфи биологи изучали виды живых организмов по отдельности, а выяснилось, что живой мир — это не просто набор различных видов, а весьма тонкое и сложное их взаимодействие. Дальнейшее развитие биологии привело к появлению понятия экологической системы, которое позволяет ставить и решать вопросы об устойчивом существовании природных сообществ.

Государственное устройство любого общества также представляет собой систему. Для демократического общества характерны такие его составные части, как законодательная власть, исполнительная власть и судебная власть. Каждая из этих ветвей власти сама представляет сложную систему политических органов, взаимодействие которых определяется конституцией государства.

Одной из сложнейших систем является человек. В нем сочетаются физические, интеллектуальные, нравственные и духовные качества. Ни одно из этих качеств, взятое в отдельности, не может полноценно охарактеризовать человека. И наука пока не в состоянии дать формализованное описание связей между ними. Лишь художественной литературе под силу создать целостный образ.

Возьмите любое предложение русского языка. «Выхожу один я на дорогу», — написал М. Ю. Лермонтов. Каждое слово этого предложения — элемент системы, грамматически связанный с другими словами. Правда, системный эффект здесь обеспечивается не только грамматикой.

А теперь представьте себе, что вы собираетесь наметить маршрут будущего похода. Перед вами две модели местности — карта и набор фотографий. Скорее всего, вы предпочтете воспользоваться картой, хотя и небесполезно знать, как выглядят места, которые вам предстоит пройти. В чем различие между этими моделями? На карте отчетливо выделены основные элементы местности — реки, дороги, просеки, овраги — и показано, как они между собой связа-



Рис. 1.8 Схема московского метрополитена

ны. Разглядывая фотографию, вам самим приходится выделять эти элементы и связи между ними. Карта — это системная модель, а фотография нет.

При моделировании люди обычно предпочитают иметь дело с моделями, которые являются системами.

Для описания систем часто используется их представление в виде графа. **Граф** — это совокупность точек, называемых **вершинами**, и линий, соединяющих некоторые из вершин. Эти линии указывают на выявленные связи между элементами системы, которые изображены вершинами. Если на линиях, соединяющих вершины, указано направление, то граф называется **ориентированным** (сокращенно **орграфом**), а линии называются его **дугами**. В обычном графе, т. е. неориентированном, линии, соединяющие вершины, называются **ребрами**.

Что же мы делаем, когда изображаем систему в виде графа? Ясно, что мы тем самым строим информационную модель исходного объекта, выделяя в нем в качестве существенной информации свойство быть системой. Типичная информационная модель, выполненная в виде графа, — схема метрополитена (рис. 1.8).

Нередко около каждого ребра графа или каждой дуги орграфа проставлено число, характеризующее связь между соответствующими вершинами. Это может быть расстояние, если граф изображает какую-либо транспортную сеть, или мощность потока воды, если граф изображает какую-либо систему водоснабжения. Граф с помеченными ребрами называется **нагруженным**. Орграф с помеченными ребрами называется **сетью**. Решение задачи, моделируемой нагруженным графом или сетью, сводится, как правило, к нахождению оптимального в том или ином смысле маршрута, ведущего от одной вершины к другой.

Впрочем, ребра или дуги могут быть помечены не числами, а какой-либо другой информацией о связи между вершинами. На рисунке 1.9 в виде сети представлено предложение «Выхожу один я на дорогу». Сеть такого типа называют **семантической** — она отражает смысловые связи между словами.

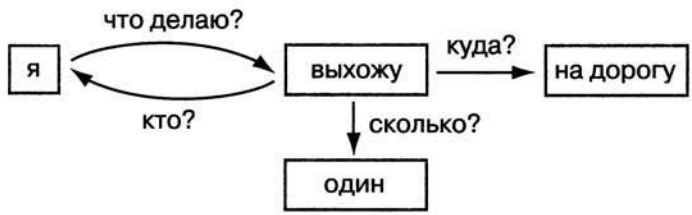


Рис. 1.9 Семантическая сеть предложения

ВОПРОСЫ И ЗАДАНИЯ

1. Что такое система? Что называют системным подходом?
2. В чем состоит системный эффект? Приведите примеры проявления системного эффекта.
3. Представьте нижеперечисленные объекты как системы:
   а) авторучка; б) дом; в) часы; г) велосипед.
   Укажите для каждой системы, что в ней является элементами и каковы связи между элементами. В чем проявляется системный эффект для каждой из систем?
4. Система может быть материальной (книжный шкаф), абстрактной (государственное устройство) или смешанной, т. е. состоящей частично из материальных объектов, частично из абстрактных (школа — школьное здание, расписание уроков и т. д.). Приведите другие примеры систем:
   а) материальных; б) абстрактных; в) смешанных.
5. Представьте школьную библиотеку как смешанную систему. Каковы в ней материальные и нематериальные объекты? За счет чего достигается системный эффект?
6. Общепризнанным достижением науки является периодическая система элементов Д. И. Менделеева. В чем проявился здесь системный эффект?
7. Для каждого графа, изображенного на рисунке 1.10, назовите систему, которая данным графом представлена.

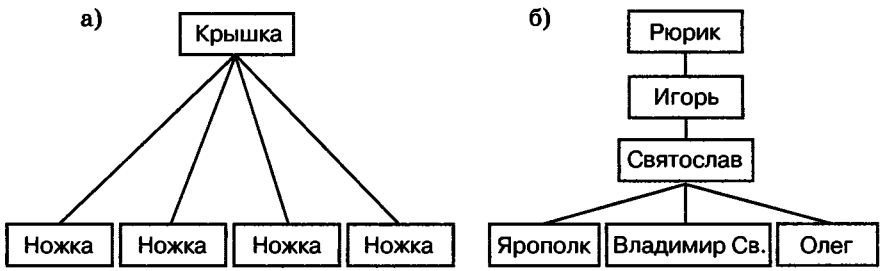


Рис. 1.10

8. Приведите примеры объектов или процессов, для которых информационную модель удобно конструировать в виде графа.
9. Экспедитору нужно развезти товары с оптовой базы по пяти магазинам. Схема путей возможного проезда к магазинам с указанием времени, затрачиваемого на дорогу, приведена в виде нагруженного графа на рисунке 1.11. (Буквами ОБ обозначена оптовая база.) Найдите наиболее быстрый путь, позволяющий экспедитору побывать в каждом магазине:
   а) если экспедитор после этого не должен возвращаться на базу;
   б) если требуется, чтобы экспедитор вернулся на базу.

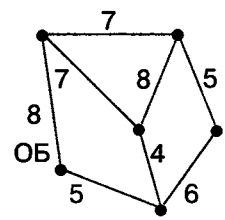


Рис. 1.11

⑩ Изобразите семантической сетью смысловые связи слов в следующих предложениях:

а) Отговорила роща золотая березовым, веселым языком.

б) Не обещайте деве юной любови вечной на земле.

в) В тот год осенняя погода стояла долго на дворе.

г) Изяществу зябко в этом суетном свете.

д) Мряка друсит пусики и на друську одного пусика тратит полдолготика.

е) Глокая куздра штепо кудланула бокра и кудлачит мохрястенького бокренка.

ж) Выстребаны обстряхнуться и дутой чернушенькой объятно хлюпнут по моргазам.

§7 Алгоритмы и их свойства

Для того чтобы формальный исполнитель мог выполнить ту или иную работу, ему нужна инструкция. В этой инструкции указывается, какие действия и в каком порядке требуется исполнить, чтобы преобразовать исходные данные в результат. Мы уже обсуждали, что формальный исполнитель не вникает в смысл исполняемых им действий, не задается вопросами, зачем он это делает и нельзя ли сделать лучше. Поэтому каждое действие должно быть сообщено исполнителю так, чтобы он понял его однозначно и мог исполнить. В инструкции должны указываться только допустимые действия данного исполнителя, а сама инструкция пишется на формализованном языке (см. § 3). В некоторых случаях требуется, чтобы не только смысл каждого слова был определен однозначно, но и сами слова должны строиться по строгим формальным правилам, т. е. грамматика языка тоже была бы формализована. Такой язык называют **формальным**. К формальным языкам относятся языки программирования и, вообще, все языки управления автоматическими устройствами. Но и человек в своей деятельности использует формальные языки, например язык записи натуральных чисел в десятичной системе счисления.

Инструкция, указывающая исполнителю, какие действия и в каком порядке он должен выполнить, и записанная на формальном языке этого исполнителя, называется **программой**. Однако написанию программы обычно предшествует разработка алгоритма.

Алгоритмом называется последовательность действий, допустимых для некоторого исполнителя, которая приводит к определенному результату.

Различие между понятиями алгоритма и программы весьма тонкое. Для поиска ошибок в программе есть специальный термин **отладка**. Отладка в качестве составной части обязательно включает в себя поиск ошибок, допущенных именно в *записи* команд. Такие ошибки принято называть **синтаксическими**, поскольку они вызваны нарушением грамматики языка, понимаемого исполни-

телем. Ошибки другого вида могут вызвать реакцию исполнителя «Не могу исполнить»; такие ошибки называются **семантическими** — исполнитель понимает смысл отданной ему команды, но исполнение ее по каким-то причинам невозможно. Например, вы предлагаете исполнителю разделить одно число на другое, а делитель оказался нулем.

Обычно алгоритмы пишутся так, чтобы они были понятны человеку. И мы будем записывать алгоритмы на обычном русском языке. При этом каждый из вас может применять какие-либо сокращения слов и вообще заменять одни слова другими. Важно только, чтобы за этими словами стояли действия, допустимые для данного исполнителя. Самого исполнителя удобно представлять себе как устройство управления, соединенное с набором инструментов. Устройство управления воспринимает и анализирует алгоритм, а затем организует его выполнение, командуя соответствующими инструментами. От исполнителя требуется лишь четкое выполнение каждого действия, входящего в алгоритм. Мы не должны объяснять ему, для каких целей предназначается алгоритм.

Хотя набор допустимых действий у каждого исполнителя свой, способы организации действий для всех исполнителей одинаковы. Для наглядного представления форм организации действий полезны так называемые **схемы алгоритмов (блок-схемы)**. В схеме алгоритмов каждая команда на выполнение какого-либо действия помещается в прямоугольник, вопрос о том, выполняется ли некоторое условие, — в ромб, а указание на ввод и вывод данных — в параллелепипед. Блоки соединяют отрезками или отрезками со стрелками, показывая очередность выполнения действий. Если на таком отрезке стрелка отсутствует, то блок, расположенный ниже, должен выполняться позже. В таблице 1.4 представлены формы организации действий в алгоритмах, способы их записи в алгоритмах и изображения в виде схемы алгоритма.

В круглых скобках со звездочками записаны комментарии, которые помогают человеку, читающему алгоритм, понять, как этот алгоритм работает.

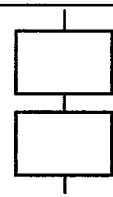
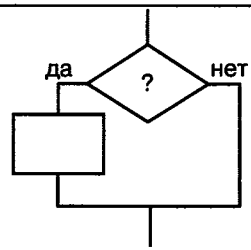
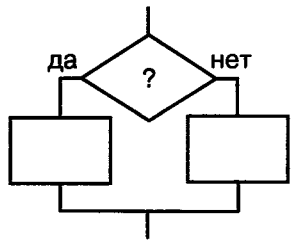
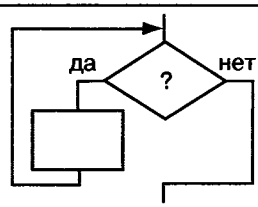
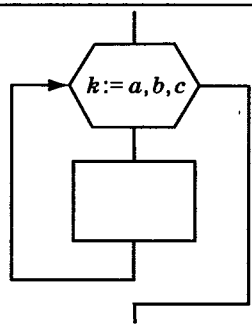
Напомним, что для хранения информации, обрабатываемой с помощью того или иного алгоритма, используются **переменные**. У каждой переменной обязательно есть **имя**. Имена позволяют одни переменные отличать от других и указывать исполнителю, из какой переменной ему нужно взять информацию для обработки и в какую поместить результат.

Какие имена можно употреблять, зависит от исполнителя. Большинство компьютерных исполнителей понимает имена, состоящие из букв и цифр или только из букв (так что никаких фамилий и отчеств у переменных нет).

Информация, которая хранится в переменной, называется ее **значением**.

Самое важное действие над переменной состоит в том, что ей можно присвоить то или иное значение. Чтобы это сделать, исполь-

Таблица 1.4

Способ организации действий	Запись в алгоритме	Схема алгоритма
Линейный	{ оператор; оператор; … }	
Ветвление (в неполной форме)	**Если** (условие), **то** { оператор; оператор; … }	
Ветвление (в полной форме)	**Если** (условие), **то** { оператор; оператор; … } **иначе** { оператор; оператор; … }	
Цикл в форме «пока»	**Делать пока** (условие) { оператор; оператор; … } (* *конец цикла* *)	
Цикл со счетчиком	**Делать от** $k:=a$ до b **с шагом** c { оператор; оператор; … } (* *конец цикла* *)	

зуют **оператор присваивания**. Его мы будем обозначать символом « := ». Например, если S — имя некоторой переменной, то действие присваивания S значения 1 записывается так:

$S := 1;$.

Слева от знака « := » всегда указывается имя переменной, которой будет присвоено некоторое значение, а справа пишется выражение, состоящее из констант и имен переменных, связанных знаками операций, допустимых для данного исполнителя. В выражении могут присутствовать скобки, указывающие на изменение естественного порядка действий.

Символ операции присваивания будем использовать и при записи заголовка цикла со счетчиком (см. табл. 1.4). Слева от него записана переменная, которую называют **параметром цикла**. Именно по его значению исполнитель определяет, надо ли еще раз выполнять тело цикла: если значение параметра больше того числа, которое стоит после слова **до**, то тело исполняться не будет, а будет исполняться команда, следующая за циклом. После каждого исполнения тела цикла к текущему значению параметра прибавляется число, которое указано как **шаг**. Если шаг отсутствует, то автоматически считается, что он равен 1.

Алгоритмы для выполнения действий над числами зависят от того, какие это числа. Например, десятичные дроби складываются совсем не так, как обыкновенные. Поэтому формальный исполнитель должен знать, с числами какого вида он будет иметь дело.

Правда, в алгоритме редко присутствуют сами числа. Обычно в нем используются переменные, которые, словно пустая тара, всегда готовы принять в себя числа на временное хранение. Но никто не хранит подсолнечное масло в бумажных пакетах или сахар в сетках. Каждому виду продукта — свой тип тары. Также и с переменными: каждому виду чисел — свой **тип переменных**. Прежде всего используются **целые** и **вещественные** типы. Для обработки текстовой информации используются **символьные** типы, а для выполнения логических операций — **логические**, или **булевы**.

Приступая к написанию программы, мы обязаны сообщить исполнителю, каков тип используемых в ней переменных. Эту информацию записывают с помощью специальной конструкции, называемой **объявлением переменных**.

В таблице 1.5 приведены обычно используемые обозначения операций. Основные способы описания переменных в языках QBasic и Pascal можно найти на форзацах учебника.

Алгоритмы могут обладать некоторыми свойствами. Перечислим основные из них.

Прежде всего назовем **дискретность**. Дискретность алгоритма означает, что он исполняется по шагам: каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.

Таблица 1.5

Операция	Выражение	Типы операндов	Тип результата
Сложение	$A + B$	Оба целые	Целый
		Хотя бы один вещественный	Вещественный
Умножение	$A * B$	Оба целые	Целый
		Хотя бы один вещественный	Вещественный
Вычитание	$A - B$	Оба целые	Целый
		Хотя бы один вещественный	Вещественный
Деление	A / B	Целый или вещественный	Вещественный
Целое деление	A div B	Оба целые	Целый
Остаток при целом делении	A mod B	Оба целые	Целый
Целая часть	INT(A)	Целый или вещественный	Целый
Корень квадратный	sqrt(A)	Целый или вещественный	Вещественный

Другое свойство принято называть **детерминированностью**. Оно означает, что каждое действие должно пониматься исполнителем в строго определенном смысле и однозначно определять преобразование данных, полученных на предшествующих шагах алгоритма. При многократном применении алгоритма к одним и тем же начальным данным должен всегда получаться один и тот же результат.

Третье свойство — **результативность** алгоритма. Оно означает, что выполнение алгоритма обязательно должно приводить к его завершению. Иными словами, после применения алгоритма мы получаем либо определенный результат, либо сообщение о невозможности решения.

Свойство результативности алгоритма в целом обычно одновременно подразумевает и **конечность** алгоритма, т. е. завершение его работы за конечное число шагов (при этом количество шагов может быть заранее неизвестным и различным для разных начальных ситуаций).

Необходимым признаком алгоритма является использование в нем только допустимых действий исполнителя. Это свойство нередко называют **понятностью** алгоритма. Термин «понятность»

довольно естественен, поскольку реакция исполнителя на записанное в алгоритме действие, не являющееся допустимым, всегда одна: «Не понимаю».

Алгоритмы в жизни людей появились намного раньше компьютеров. Некоторые из них навсегда ушли из практики человека. Никто, например, сейчас не производит действия с дробями по тем алгоритмам, которые применялись древними египтянами. Другие, столь же давно изобретенные алгоритмы и сегодня играют важную роль. Вот один из таких алгоритмов, который называется алгоритмом Евклида. Он предназначен для вычисления наибольшего общего делителя двух натуральных чисел.

```
Алгоритм НОД
цел: m, n;
  { Запросить m;
    Запросить n;
      Делать пока (не (m = n))
          { Если (m > n) то { m := m − n; }
            иначе { n := n − m; }
          }
    Сообщить m;
  }
```

Этот алгоритм позволяет решить не одну конкретную задачу, например найти НОД чисел 178 357 и 1 049 254, а любую задачу такого типа. Свойство алгоритма, состоящее в возможности его применения к решению различных однотипных задач, называется **массовостью.**

Надо сказать, что вовсе не каждый алгоритм обязан обладать всеми указанными свойствами. Представим, что любой алгоритм должен быть конечным, т. е. завершать работу за конечное число шагов. А если это алгоритм управления космической станцией или термоядерным реактором? Момент прекращения работы алгоритма — это момент катастрофы.

Впрочем, не надо обращаться к столь далеким от вашей практики примерам. Каждая операционная система на вашем компьютере — это бесконечно исполняемый алгоритм. В тот момент, когда операционная система прекратит работу, ваш компьютер станет бесполезен. Можно сказать, что все алгоритмы управления объектами в режиме реального времени могут быть бесконечными.

Не все алгоритмы обладают и свойством массовости. Иногда бывает нужен алгоритм для решения одной, но очень трудоемкой задачи. К примеру, нужно вычислить первые 100 000 знаков после запятой для числа π. Без компьютера здесь не обойтись, а значит, нужен и алгоритм. Такой алгоритм был составлен, задача была решена. Вряд ли этот алгоритм может пригодиться для какой-нибудь другой задачи.

ВОПРОСЫ И ЗАДАНИЯ

1. Какой язык называется формальным?
2. Что такое алгоритм? Что такое программа?
3. Любые ли действия могут присутствовать в алгоритме, предназначенном для данного исполнителя?
4. Что в истории человечества появилось, по вашему мнению, раньше: письменность или алгоритмы?
5. Нужны ли комментарии формальному исполнителю?
6. Какие общие свойства алгоритмов вы можете назвать?
7. Как записывается ветвление в полной форме? А в неполной?
8. Когда целесообразно применять оператор цикла?
9. Что такое тело цикла?
10. Как вы думаете, для чего служат операторные скобки в операторах **Делать пока** и **Если** ... **то** ...? Почему они одинаковые для обоих операторов?
11. Для чего нужно имя переменной? Что называют значением переменной?
12. С помощью какой операции можно придать или изменить значение переменной?
13. Для чего нужен тип переменной? Как исполнитель узнает, с переменными какого типа ему придется иметь дело при исполнении данного алгоритма?
14. Перечислите известные вам типы переменных.
15. Что такое отладка программы?
16. а) Какую ошибку называют синтаксической, а какую — семантической? Какие из этих видов ошибок относятся к программе, а какие — к алгоритмам?

    б) Могут ли быть в алгоритме или программе другие ошибки, нежели синтаксические и семантические? Если да, то как бы вы их назвали?
17. С какими алгоритмами вы познакомились, занимаясь:
    а) русским языком;
    б) иностранным языком;
    в) математикой;
    г) литературой;
    д) химией;
    е) физкультурой;
    ж) информатикой?
18. Исполните алгоритм, изображенный схемой алгоритма на рисунке 1.12, при разных значениях a, b, c. Определите, для чего предназначен этот алгоритм.
19. Пусть дан отрезок AB. Определите, для решения какой задачи предназначен следующий алгоритм:

    Поставить ножку циркуля в точку A.
    Установить раствор циркуля равным длине отрезка AB.
    Провести окружность.
    Поставить ножку циркуля в точку B.
    Провести окружность.
    Провести прямую через точки пересечения окружностей.

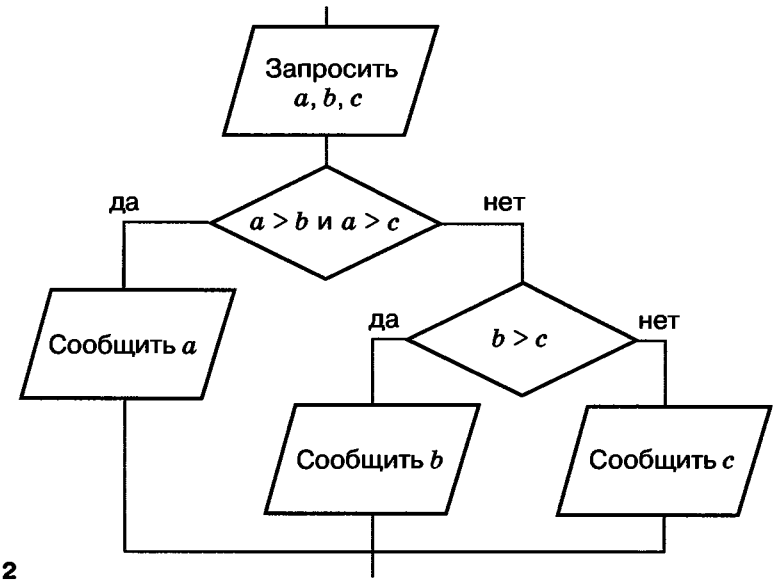


Рис. 1.12

20 Чему будут равны a и b после выполнения каждого из следующих алгоритмов? Сколько раз будет выполняться тело цикла в каждом из них?

а)
```
Алгоритм № 1
цел: a, b;
{ a := −5;
  b := −2;
Делать пока a + b < a*b
   { a := 2*a;
     b := b + 1;
   }                (* конец цикла *)
}                (* конец алгоритма *)
```

б)
```
Алгоритм № 2
цел: a, b;
{ a := 25;
  b := 1;
Делать пока a − b > a*b
   { a := a + 10;
     b := −3*b;
   }                (* конец цикла *)
}                (* конец алгоритма *)
```

в)
```
Алгоритм № 3
цел: a, b;
{ a := −3;
  b := −1;
Делать пока a < b
  { Если b < 3 то
      { a := a − b;
        b := b + 2;
      }
    иначе
      { a := a + 2;
        b := b − a;
      }
  }                (* конец цикла *)
}
```

21 Определите, сколько раз исполняется тело цикла в каждом из указанных ниже случаев:

а)
```
Делать от k := −1 до 9
с шагом 2
   { действие;
     ...
   }
```

б)
```
Делать от k := −5 до −8
с шагом −1
   { действие;
     ...
   }
```

в) **Делать от** $k := -1$ **до** 7
с шагом 2
{ действие;
...
}

г) **Делать от** $k := 1$ **до** 3
с шагом 3
{ действие;
...
}

а) В роли формального исполнителя выполните алгоритм Евклида (с. 33) для пар чисел 6 и 4, 9 и 12, 8 и 24.
б)* Нам вовсе не хочется, чтобы вы ощущали себя только формальными исполнителями. Поэтому постарайтесь обосновать, что алгоритм Евклида действительно вычисляет НОД. (С о в е т. Полезно воспользоваться таким свойством для натуральных чисел a и b: если $a > b$, то НОД(a, b) = НОД$(a - b, b)$. Конечно, это свойство тоже нуждается в обосновании.)
в)* Постарайтесь объяснить, почему алгоритм Евклида конечен при любых допустимых начальных значениях аргументов.

Исполните следующий алгоритм при разных значениях v (в записи этого алгоритма LEN(v) обозначает оператор, вычисляющий количество символов в значении символьной переменной v; Часть (a, b, c) — оператор, который в значении символьной переменной a выделяет часть, начинающуюся с символа, занимающего b-е место, и содержащую c символов).

Алгоритм № 4
сим: v, w;
цел: a, b;
{ **Запросить** v;
$a :=$ LEN(v);
$w :=$ " ";
Делать от $b := 1$ **до** a
{ $w := w +$ Часть $(v, a + 1 - b, 1)$;
} (* *конец цикла* *)
Сообщить w;
} (* *конец алгоритма* *)

§ 8 Формальный исполнитель: автомат

Человек весьма преуспел в создании разнообразных устройств, выполняющих ту или иную работу в соответствии с четкой инструкцией. При этом он уже не обязан постоянно находиться около этого устройства, чтобы управлять им. В этом случае говорят, что устройство выполняет работу автоматически, а само такое устройство называют **автоматом**.

В реальной жизни автоматы бывают очень разными. Это может быть автомат по продаже билетов на поезда пригородного сообщения, или автомат по расфасовке готовой продукции, или автомат по изготовлению каких-либо деталей. Автоматы последнего вида нередко называют **промышленными роботами**.

С точки зрения информатики совершенно все равно, из чего сделан автомат. Важно лишь то, что он воспринимает некоторые сигналы как команды и по каждой команде выполняет некоторое действие, переходя из одного состояния в другое. Поэтому можно считать, что каждый автомат описывается набором возможных со-

стояний, списком допустимых команд и перечислением того, из какого состояния в какое переходит автомат под воздействием каждой команды. Например, если команд только две, то их можно обозначить буквами (например, a и b) или цифрами, состояния автомата — буквами q_1, q_2, ..., q_m, а перечислить варианты перехода из одного состояния в другое можно с помощью таблицы (например, такой, как таблица 1.6).

Таблица 1.6

Состояние / Команда	q_1	q_2	...	q_m
a	q_2	q_{m-1}	...	q_m
b	q_1	q_m	...	q_2

Таблица 1.7

Состояние / Команда	q_1	q_2	q_3	q_4
a	q_2	q_4	q_4	q_4
b	q_1	q_3	q_1	q_3

В клетке, стоящей на пересечении строки и столбца, указывается то состояние, в которое переходит автомат, находившийся в состоянии, которое указано в заголовке того же столбца, и получивший команду, указанную в заголовке той же строки. Каждому ясно, что такая таблица представляет собой информационную модель реального автомата.

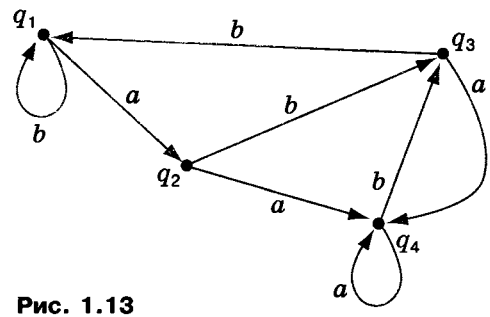


Рис. 1.13

Автомат можно описать и другой информационной моделью — орграфом. Вершинами орграфа являются состояния автомата, дугами — переходы из одного состояния в другое. На каждой дуге имеется отметка, показывающая, по какой команде осуществляется переход. Тогда автомат, описанный таблицей 1.7, изобразится так, как показано на рисунке 1.13.

Одно из состояний называется начальным — именно в нем находится автомат до начала работы. Договоримся начальное состояние всегда обозначать q_1. Некоторые из состояний являются заключительными — приведение автомата в это состояние является целью управления автоматом с помощью той или иной последовательности команд. Например, если это автомат по продаже железнодорожных билетов, то в начальном состоянии автомат ожидает, что в монетоприемник начнут поступать монеты. Конечных состояний два: выдача билета и возврат денег. Кроме того, имеются промежуточные состояния — подсчет суммы денег, поданных в автомат к этому моменту. Команды, переводящие автомат из одного состояния в другое, — это опускание монеты в монетоприемник, нажатие кнопки выдачи билета или нажатие кнопки возврата денег.

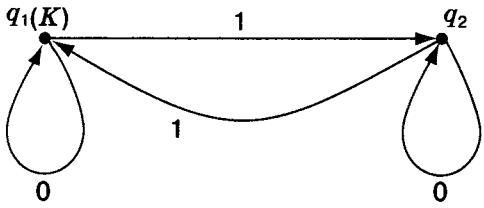


Рис. 1.14

Обозначать тот факт, что данное состояние конечное, будем буквой K в скобках около обозначения данного состояния. Например, $q_2(K)$.

Целью управления автоматом является выдача ему такой последовательности команд, которая переводит его из начального состояния в какое-либо конечное. Поскольку каждая команда обозначена буквой, то набор команд, понимаемых данным автоматом, можно считать некоторым алфавитом A. Тогда последовательность команд будет записываться как слово в этом алфавите. Например, слово *aba* переводит автомат, описанный таблицей 1.7, из начального состояния q_1 в состояние q_4.

Множество всех тех слов, которые переводят автомат из начального состояния в одно из конечных состояний, образует некий формальный язык. Этот язык называется **языком, распознаваемым данным автоматом**. Если для некоторого языка существует хотя бы один автомат, который этот язык распознает, то такой язык называют **распознаваемым**.

На рисунке 1.14 изображен автомат, имеющий два состояния — $q_1(K)$ и q_2 и понимающий две команды, которые обозначены цифрами 0 и 1. Легко сообразить, что распознаваемый этим автоматом язык состоит из тех и только тех слов, которые содержат четное количество единиц и любое количество нулей. Иными словами, этот автомат определяет, четна ли сумма цифр в числе, записанном в двоичной системе счисления.

Язык, распознаваемый каким-либо автоматом, конечно, является формализованным: у него точно определено значение каждого слова — перевод автомата из начального состояния в заданное конечное. Специалисты по автоматам установили, что для каждого распознаваемого языка существует формальная грамматика, задающая этот язык. Тем самым каждый распознаваемый язык является формальным. Эти грамматики легли в основу всех ныне существующих языков программирования.

ВОПРОСЫ И ЗАДАНИЯ

1. Какими двумя информационными моделями может быть представлен автомат?
2. Что такое язык, распознаваемый данным автоматом?
3. Какой язык называется распознаваемым?
4. Для автомата, изображенного на рисунке 1.13, определите, в каком состоянии он будет находиться после исполнения последовательности команд:
   а) *abba*; б) *ababbabbb*; в) *babaabaaa*;

г)* $a^n b^n$, где запись a^n означает, что буква a повторена n раз.

5. Для автомата, изображенного на рисунке 1.14, составьте описание в форме таблицы.

6. Постройте в виде графа информационную модель автомата, заданного таблицей 1.8.

Таблица 1.8

Состояние / Команда	q_1	q_2	$q_3(K)$
a	q_2	q_3	q_3
b	q_1	q_2	q_3

7. Какой язык над двухсимвольным алфавитом {0, 1} распознается автоматом, изображенным на рисунке 1.15?

8. Какой язык над двухсимвольным алфавитом $\{a, b\}$ распознается автоматом, заданным таблицей 1.8?

9. Изобразите в виде графа информационную модель автомата, который бы распознавал язык над алфавитом {0, 1}, состоящий из всех слов, содержащих ровно 5 подряд идущих единиц.

10. Изобразите в виде графа информационную модель автомата, который бы распознавал язык над алфавитом {0, 1}, состоящий из всех слов, содержащих ровно 5 единиц.

11. Изобразите в виде графа информационную модель автомата, который бы распознавал язык над алфавитом {0, 1}, состоящий из всех слов, начинающихся с единицы (т. е. этот автомат отличает натуральные числа, записанные в двоичной системе счисления, от произвольных последовательностей символов 0 и 1).

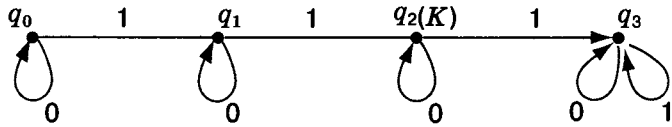


Рис. 1.15

§ 9 Универсальный исполнитель

Наверно, каждый, кто имел дело с компьютером, видел, а возможно, и сам играл в какую-либо компьютерную игру. На экране одни объекты в виде живых существ или технических устройств подчиняются командам играющего, другие — управляются компьютером. Все эти объекты — формальные исполнители, каждый со своим набором допустимых действий. Только объекты эти не реальные, а имитируемые компьютером. Получается, что с помощью одного формального исполнителя имитируются другие.

Попытаемся сформулировать, что значит один исполнитель имитируется с помощью другого. Говорят, что формальный исполнитель A имитирует формального исполнителя B, если:

- каждому объекту, над которым выполняет действия исполнитель B, однозначно соответствует объект, над которым выполняет действия исполнитель A (имитация среды исполнителя);
- каждому допустимому действию исполнителя B над тем или иным объектом среды однозначно сопоставлено допустимое

действие исполнителя A над соответствующим объектом его среды (имитация действий);

- каждая инструкция, составленная для исполнителя B и приводящая при ее исполнении к определенному результату (т. е. к определенному состоянию среды исполнителя и его самого), может быть преобразована имитацией допустимых действий в инструкцию для исполнителя A, исполнение которой приводит к соответствующему результату в среде исполнителя A.

Впрочем, предположение, что у исполнителей A и B разная среда, в которой они существуют, не принципиально с информационной точки зрения. Например, исполнитель A имеет дело с числами, а исполнитель B преобразует графические изображения. Но вы уже знаете, что фактически в каждом из этих случаев речь идет о преобразовании подходящим образом закодированной информации. Более того, можно считать, что используется один и тот же двоичный код, т. е. среда исполнителей просто одинакова — информация, представленная в форме закодированного сообщения.

Один из важнейших вопросов теоретической информатики звучит так: существует ли формальный исполнитель, с помощью которого можно имитировать любого формального исполнителя? Такого исполнителя естественно назвать **универсальным**. Легко понять, что как физическое устройство универсальный исполнитель не существует: ведь информация может кодироваться как угодно длинными сообщениями, а любой физический носитель конечен. Если же вести речь об универсальном исполнителе как идеальном объекте, то оказывается, что ответ на заданный вопрос положителен. И получен он был почти одновременно и независимо двумя выдающимися учеными — А. Тьюрингом (в 1936 году) и Э. Постом (в 1937 году). Предложенные ими исполнители отличались друг от друга, но оказалось, что они могут имитировать друг друга, а главное — имитировать вообще любого формального исполнителя.



А. Тьюринг

Универсального исполнителя принято называть машиной и давать им имена их изобретателей. Так что универсального исполнителя, придуманного А. Тьюрингом, называют машиной Тьюринга, исполнителя, описанного Э. Постом, — машиной Поста. Позже появились и другие универсальные исполнители, например машина Минского.

Итак, можно считать, что у нас имеется сообщение, записанное в некотором алфавите, и его требуется преобразовать в другое сообщение. Написать инструкцию формальному исполнителю для обработки конкретной пары сообщений —

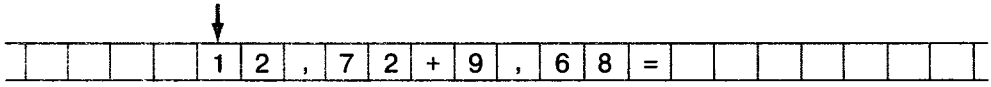


Рис. 1.16

дело нехитрое. Но вы уже знаете, что реальный интерес представляют инструкции (т. е. алгоритмы), позволяющие решать целый класс однотипных задач — так называемое свойство массовости алгоритма. Например, такая задача: к любому сообщению приписать справа еще один заранее заданный символ. Если последовательность одинаковых символов выступает кодом натурального числа (количество символов в последовательности и есть кодируемое натуральное число), то фактически речь идет о создании алгоритма увеличения числа на 1.

Считают, что сообщение записано на ленту. Более того, удобно представлять себе эту ленту разделенной на одинаковые клетки, в каждой из которых записан ровно один символ сообщения. Поскольку сообщения могут быть как угодно длинными, то ленту договоримся представлять себе бесконечной. Пустые клетки будем считать заполненными символом «пробел». Тем самым мы объявили, что любой алфавит, который будет использоваться нами для записи сообщений на этой ленте, обязательно содержит пробел. Договоримся обозначать его a_0. Остальные символы алфавита, используемого для записи сообщений на ленту, будем обозначать $a_1, a_2, \ldots, a_n$. Если нам требуется записать задачу вычисления суммы двух чисел, то алфавит можно взять таким: a_0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 0; ,; +; =. Запись на ленте может выглядеть, например, так, как показано на рисунке 1.16.

Мы не будем на ленте в пустых клетках писать символ a_0, подразумевая его там. Кроме того, договоримся, что первый символ сообщения, отличный от пробела, всегда стоит в одной и той же клетке; на рисунке 1.16 она отмечена стрелкой. Эту клетку будем называть начальной.

Сообщение, записанное на ленте, обрабатывается неким устройством, и результат снова записывается на ленту. Поскольку исполнитель формальный, он не вникает в смысл сообщения, а по составленной для него инструкции заменяет одни символы на другие. Нас не интересует, как физически осуществляется такая замена, поэтому можно представить, что вдоль ленты движется некий автомат, который считывает символ из клетки на ленте, обрабатывает полученную информацию, печатает в клетке другой символ (если это предписано инструкцией) и сдвигается к соседней клетке вправо или влево.

Вы уже знаете, что каждый автомат описывается набором состояний. Принято обозначать состояния указанного считывающе-печатающего автомата буквами q_0, q_1, q_2, ..., q_m. При этом одно состояние особое — это прекращение работы, так называемое состоя-

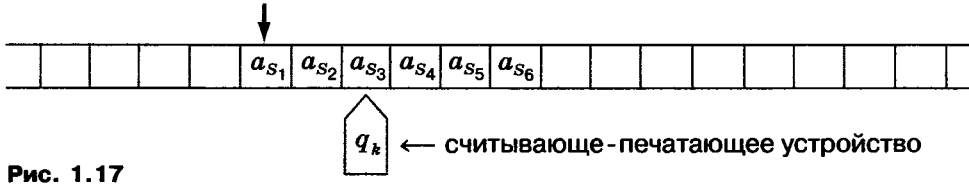


Рис. 1.17

ние останова. Его обозначают q_0. Кроме того, при включении автомата, т. е. в начале работы, он всегда находится в одном и том же состоянии, которое мы будем обозначать q_1, и располагается напротив начальной клетки.

Таким образом, машина Тьюринга формально описывается набором двух алфавитов: $A = \{a_1, a_2, \ldots, a_n\}$ и $Q = \{q_0, q_1, q_2, \ldots, q_m\}$. Алфавит A называется **внешним** и служит для записи исходных сообщений, алфавит Q называется **внутренним** и описывает набор состояний считывающе-печатающего устройства. Изображать машину Тьюринга будем так, как на рисунке 1.17. На этом рисунке изображен момент работы, когда считывающе-печатающее устройство обозревает третью клетку от начальной (в ней к этому моменту оказался символ a_{s_3}) и находится в состоянии q_k.

Итак, допустимые действия машины Тьюринга таковы:

- записать какой-либо символ внешнего алфавита в секцию ленты (символ, бывший там до того, затирается);
- сместиться в соседнюю секцию;
- сменить состояние на одно из обозначенных символом внутреннего алфавита.

В этом перечислении в каждой строке указано не одно действие, а группа действий одного вида — действий «записать символ внешнего алфавита» столько, сколько этих символов, сместиться в соседнюю секцию можно вправо, а можно влево, действий по изменению состояния столько, сколько этих состояний (т. е. сколько символов внутреннего алфавита).

Теперь надо сказать, как записываются команды на выполнение указанных действий. Каждая команда машины содержит не более одного действия из каждой группы действий и имеет вид

указание о записи символа	указание о перемещении машины	указание о смене состояния

Фактически команды выглядят так:

$a_i П q_j$ — в обозреваемую секцию записать a_i, сместиться вправо (к следующей секции) и перейти в состояние q_j;

$a_i Л q_j$ — в обозреваемую секцию записать a_i, сместиться влево и перейти в состояние q_j;

$a_i Н q_j$ — в обозреваемую секцию записать a_i, остаться на месте и перейти в состояние q_j.

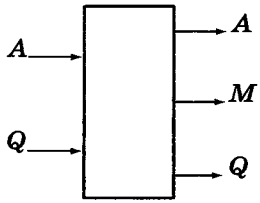


Рис. 1.18
Операционно-логический блок машины Тьюринга

Таблица 1.9

Состояние машины Q / Внешний алфавит A	q_1	…	q_j	…	q_n
a_0					
a_1					
…					
a_i			$a_s \Pi q_t$		
…					
a_m					

Для осуществления действий машина Тьюринга имеет операционно-логический блок. У этого блока есть два входа: через один из них поступает информация о том, какой символ стоит в обозреваемой ячейке, через другой — информация о том, в каком состоянии находится машина на данном шаге своей работы. Эта информация однозначно определяет, какую именно команду следует выполнить машине. Поскольку команда может содержать указание на выполнение трех действий — записи символа на ленту, смещения и смены состояния, операционно-логический блок имеет три выхода: запись символа A, смещение M и смена состояния Q (рис. 1.18).

Поскольку у этого блока всего лишь два входа, его реакцию на символы, подаваемые на них, удобно представить в виде прямоугольной таблицы, в которой строки и столбцы помечены символами внешнего и внутреннего алфавитов соответственно (табл. 1.9). В клетки таблицы записываются команды. Если машина находится напротив клетки, где написано a_i, а ее состояние при этом есть q_j, то выполняется команда, стоящая на пересечении строки, отмеченной символом a_i, и столбца, отмеченного символом q_j.

Эту таблицу называют **функциональной схемой** данной машины; она-то и выполняет роль инструкции (программы) для машины Тьюринга. Из нее, в частности, видно, каковы внешний и внутренний алфавиты машины.

Пусть, к примеру, на ленте записана последовательность из некоторого количества одного и того же символа *. Тогда функциональная схема, приведенная в таблице 1.10, заставляет машину Тьюринга увеличить на одну количество звездочек в этой последовательности.

Таблица 1.10

A / Q	q_1
a_0	*Нq_0
*	*Лq_1

Доказать, что машина Тьюринга является универсальным исполнителем, нельзя. Так же, например, как нельзя доказать закон сохранения энергии в физике. Но практика составления алгоритмов показывает, что любую задачу, кото-

рую сегодня умеет решать человек, можно запрограммировать на машине Тьюринга. Этот экспериментальный факт, называемый тезисом Тьюринга, формулируют так: для задачи существует алгоритм решения тогда и только тогда, когда имеется подходящая машина Тьюринга, посредством которой можно решить эту задачу.

ВОПРОСЫ И ЗАДАНИЯ

1. Какой формальный исполнитель называется универсальным?
2. Что такое машина Тьюринга?
3. Чем одна машина Тьюринга отличается от другой?
4. Что называют функциональной схемой машины Тьюринга?
5. Верно ли, что машина Тьюринга с написанной для нее функциональной схемой является конечным автоматом?
6. Изобразите в виде последовательности рисунков, как изменяется информация на ленте при работе машины Тьюринга, описанной таблицей 1.10.
7. Следуя функциональной схеме, описанной таблицей 1.10, машина Тьюринга приписывает к заданной последовательности звездочек еще одну звездочку слева. Составьте функциональную схему, согласно которой звездочка будет приписываться справа от заданной последовательности.
8. Пусть внешний алфавит состоит из символа a_0 и цифр 0, 1, 2, ..., 9. На ленту записывается натуральное число. Придумайте машину Тьюринга и составьте для нее функциональную схему, согласно которой данное число будет увеличено на 1.

§10 Основные направления информатики

Информатику определяют как науку, изучающую информационные процессы, а также инструменты, применяемые для получения, хранения, передачи и обработки информации. На начальном этапе развития информатики основные объемы хранимой и обрабатываемой информации были представлены текстами, зафиксированными на бумажных носителях. В это время информатика служила теоретической основой библиотечного дела — увеличивающиеся темпы роста научно-технической и социальной информации требовали разработки научно обоснованных методов систематизации информации, позволяющих создавать средства ее эффективного поиска и обработки.

Поиском информации об окружающем мире и ее обработкой занимаются все науки. Это, однако, не означает, что информатика включается в эти науки как их составная часть или, наоборот, что другие науки той или иной своей стороной входят в информатику.

Роль информатики в системе наук близка к роли математики — обеспечивать методами исследования и обработки информации. Но до изобретения ЭВМ, ныне именуемых компьютерами, эта роль была не столь значительна.

Вы, конечно, знаете, что первые ЭВМ, созданные в 50-е годы прошлого столетия, предназначались для расчетов в ядерной физике и космической технике. Однако сфера применения компьютеров быстро расширялась. Последовательное и активное внедрение ЭВМ в область управления и экономику привело к созданию новой отрасли — средств и методов автоматической обработки информации. При этом оказалось, что здесь, как и в математике, создаваемые средства и методы подчиняются некоторым общим закономерностям, не зависящим от конкретной природы задач, для решения которых они используются. Конечно, осознание этой общности происходило не на пустом месте — оно было подготовлено развитием кибернетики, математической логики, теорий формальных языков и алгоритмов. Именно в кибернетике было сформулировано положение о единстве законов протекания информационных процессов в живой природе, обществе и технике. Сама кибернетика как самостоятельная наука об общих законах управления в живых, технических и социальных системах возникла в середине XX века благодаря работе группы ученых разных направлений (математики, лингвисты, нейрофизиологи и др.) под руководством Н. Винера. В 30-е годы прошлого столетия трудами Л. Черча, А. Тьюринга, Э. Поста были заложены основы теории алгоритмов, показавшей, что независимо от происхождения информации основы ее автоматизированной обработки едины. Проблемы передачи информации по техническим системам связи без искажений, вызванных помехами, а также задачи защиты информации от несанкционированного проникновения и использования привели к глубоким исследованиям в теории кодов и методов шифрования. Эти исследования, в частности, показали, что к вопросу о количестве информации надо подходить не только с содержательной, но и с формально-технологической точки зрения.

В свою очередь, чрезвычайно важной оказывается содержательная точка зрения, которую часто называют логико-семантической. В этом случае информация рассматривается как знание, причем такое, которое используется для ориентировки в окружающей действительности, принятия решений, управления и самоуправления. Вместе с компьютерной техникой это направление информатики часто выступает как информационное моделирование, о котором шла речь в § 5. Среди разнообразных информационных моделей особое место занимают так называемые **системы искусственного интеллекта**. Такие системы на основе имеющейся информации моделируют принятие человеком решений. Одним из примеров систем искусственного интеллекта являются **экспертные системы**. Например, к экспертным системам относятся компьютерные программы медицинской диагностики.

Бурная информатизация общества сопровождается тем, что информационные процессы все больше и больше протекают в электронном виде. Информационные ресурсы глобальных компьютерных сетей, личная переписка по электронной почте, электронные банковские операции, дистанционное компьютерное управление промышленными объектами — все это требует обеспечения безопасности. Разработка средств защиты информации, представленной в электронном виде, — одно из приоритетных направлений современной информатики. Напомним, что согласно статье 20 Федерального закона об информации, информатизации и защите информации, принятого Государственной Думой Российской Федерации, «целями защиты являются:

— предотвращение утечки, хищения, утраты, искажения, подделки информации;
— предотвращение несанкционированных действий по уничтожению, модификации, искажению, копированию, блокированию информации; предотвращение других форм незаконного вмешательства в информационные ресурсы и информационные системы...».

Такое широкое понимание термина «защита информации» ставит перед разработчиками весьма разнообразные задачи — от создания эффективных программ-архиваторов, позволяющих создавать компактные резервные версии документов и программного обеспечения, до коллективных, в том числе сетевых, антивирусных средств. Напомним, что компьютерным вирусом называется программа, способная к самовоспроизведению (возможно, в измененном виде) и производящая несанкционированные пользователем изменения в информации, хранящейся в компьютере.

Важную роль в предотвращении несанкционированного доступа и искажения информации играют современные методы компьютерной криптографии — электронная подпись (подтверждающая авторство документа), электронные водяные знаки (подтверждающие подлинность графических файлов), одностороннее шифрование (подтверждающее подлинность текстового электронного документа), шифрование с открытым ключом (обеспечивающее конфиденциальность общения в сетях массовых коммуникаций) и т. п.

Подводя итог, назовем основные направления информатики как науки:

- разработка моделей процессов и явлений реального мира для получения информации о закономерностях их возникновения и развития;
- разработка алгоритмов и программ для компьютерной реализации моделей, организация компьютерного эксперимента с моделями;
- организация пользовательского интерфейса в компьютерной программной среде;
- организация сетевых структур передачи, хранения и поиска информации;

- организация хранения и поиска информации в базах данных;
- разработка систем искусственного интеллекта и создание на их основе информационных технологий;
- разработка средств защиты информации, представленной в электронном виде;
- разработка компьютерной техники и технологий, обеспечивающих развитие указанных выше направлений.

ВОПРОСЫ И ЗАДАНИЯ

1. Что изучает информатика как наука?
2. Приведите примеры известных вам информационных моделей, применяемых:
   а) для познания природы;
   б) для предсказания природных явлений;
   в) для управления производством;
   г) для оценки политической обстановки в стране.
3. Приведите примеры известных вам сетевых ресурсов поиска информации.
4. Какие направления развития аппаратных средств информатизации вы можете назвать?
5. Каковы основные цели защиты информации?
6. Что такое компьютерный вирус? Какие антивирусные программные средства вам известны?
7. Перечислите основные информационные технологии, удовлетворяющие информационным потребностям человека и общества.

ИТОГИ ГЛАВЫ 1

Информация — это то, что позволяет живым организмам, их сообществам или техническим системам реагировать на воздействие окружающей среды, обеспечивая их целенаправленную деятельность. Более общий взгляд на понятие информации состоит в том, что информация — это отражение разнообразия в существующем мире. В областях науки, изучающих в основном человека и общество, под информацией принято понимать знания об окружающем человека мире и о самом себе.

Информация существует только будучи зафиксированной на том или ином носителе. Зафиксированную каким-либо способом информацию называют информационным объектом.

Информация проявляет себя в тех или иных информационных процессах. К основным информационным процессам относятся получение информации, ее хранение, передача и обработка. Наука, изучающая информационные процессы, а также инструменты, при-

меняемые для получения, хранения, передачи и обработки информации, называется информатикой.

Основными формами представления информации, которая может в последующем обрабатываться техническими средствами, являются символьная, визуальная и звуковая. Символьная информация представляется в виде сообщений, записанных на некотором языке.

Язык — это совокупность слов, составленных из символов некоторого алфавита. Совокупность правил составления слов из символов алфавита называется грамматикой языка. Язык называется формализованным, если за каждым словом закреплено ровно одно значение. Формализованный язык, в котором грамматика также формализована, называется формальным.

Для автоматизации информационных процессов информация кодируется. Из-за технического удобства кодирование обычно осуществляется в двухсимвольном алфавите — двоичное кодирование. Для двоичного кодирования визуальной и звуковой информации применяется процесс дискретизации, или оцифровки.

Для получения, обработки, сохранения и передачи знания человек применяет моделирование, т. е. заменяет один объект, процесс или явление другим, но сохраняющим свойства исходного объекта, процесса или явления, существенные для решения поставленной задачи. При этом заменяющий объект или процесс называется моделью исходного.

Особый класс моделей составляют информационные модели. Информационной называется модель, представляющая объект, процесс или явление набором параметров и связей между ними.

Нередко объект или процесс представляет собой систему, т. е. совокупность связанных между собой элементов, причем реализация этих связей обеспечивает наличие нового качества, которого нет при рассмотрении элементов по отдельности. Появление нового качества называют системным эффектом или эмерджентностью.

Удобным способом представления системы является ее изображение в виде графа или орграфа.

Обработка, или преобразование, информации — это изменение содержания или формы информационного объекта. Обработка информации может осуществляться эвристически или формально. В первом случае исполнитель, производящий обработку, вникает в смысл обрабатываемой информации и самостоятельно принимает решения о целях и способах ее преобразования. Во втором случае исполнитель следует предписаниям составленной для него инструкции, не вникая в смысл выполняемых им действий. Формальная обработка информации выполняется автоматически и потому может быть поручена техническому устройству.

Инструкция, выполненная в форме указания последовательности действий, которые надлежит исполнить формальному исполнителю, называется алгоритмом. В алгоритме должны присутствовать только те действия, которые допустимы для данного исполнителя. Это свойство алгоритма называется понятностью.

Алгоритмы обычно обладают и рядом других свойств: детерминированностью, конечностью, результативностью, массовостью.

При составлении алгоритма используются конструкции ветвления (в полной или неполной форме), цикла (в форме «пока» или со счетчиком), обращения к вспомогательному алгоритму или подпрограмме-функции.

Чтобы один и тот же алгоритм можно было использовать для различных исходных данных, а также чтобы в процессе исполнения алгоритма передавать информацию от одной операции к другой, используются переменные. У каждой переменной есть имя, позволяющее одну переменную отличать от другой и указывать в алгоритме, из какой переменной информацию нужно взять на обработку и в какую переменную поместить результат. Вторая характеристика переменной — ее тип. Он указывает исполнителю, как должна выполняться та или иная операция над данными, хранящимися в переменной. Сама информация, которая хранится в переменной, называется ее значением. В языках программирования обычно используются числовые типы целый и вещественный; логический тип (булевый); символьный тип (строковый).

Основное действие, которое совершается над переменной, — это присваивание ей значения. Она обозначается при помощи символа «:=».

Алгоритм, записанный на языке исполнителя, называется **программой**. Поиск ошибок в программе называется ее отладкой. Ошибка может быть синтаксической, т. е. вызванной неправильным употреблением слов и конструкций языка программирования, и семантической — на нее формальный исполнитель реагирует отказом выполнить предписанное ему действие.

ПРОВЕРЬ СЕБЯ

Часть 1. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и номер правильного ответа. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

А1. Информационный объем сообщения равен 256 битам. В байтах объем того же сообщения равен:

1) 8; 2) 5; 3) 32; 4) 64.

А2. В сообщении «Информатика — мой любимый предмет» каждый символ кодируется в системе UNICODE. Информационный объем этого сообщения:

1) 528 бит; 2) 66 бит; 3) 528 килобайт; 4) 66 килобайт.

А3. Для пяти букв латинского алфавита заданы их двоичные коды:

a — 001, b — 10, c — 100, d — 101, e — 01.

Двоичной строкой 1010011010101 закодирована следующая последовательность из пяти букв:

1) *bbecd*; 2) *dbabd*; 3) *dabde*; 4) *bcdde*.

А4. Из пяти букв *a, b, c, d, e* латинского алфавита известны двоичные коды трех букв:

$$a — 001, \quad d — 101, \quad e — 01.$$

Кроме того, известно, что двоичной строкой 01001011101001010 закодирована последовательность *becdab*. Тогда последовательность *abcde* кодируется как:

1) 0011010010101; 2) 00101001110101;
3) 01010110101101; 4) 00110101001101.

А5. Для обработки звуковой информации компьютером эта информация:

1) моделируется; 2) формализуется; 3) дискретизируется;
4) в пунктах 1—3 нет правильной формулировки.

А6. Свойство модели соответствовать теоретическим воззрениям и наблюдениям практики называется:

1) системностью; 2) информативностью; 3) адекватностью;
4) среди вариантов, указанных в пунктах 1—3, нет правильной формулировки.

А7. Допустимость действия для исполнителя — это:

1) право исполнителя выполнить это действие;
2) умение исполнителя выполнить это действие;
3) необходимость для исполнителя выполнить это действие;
4) в пунктах 1—3 нет правильной формулировки.

А8. Любой алгоритм состоит из:

1) правил, по которым можно действовать исполнителю;
2) алгоритмических конструкций, понятных исполнителю;
3) допустимых действий исполнителя;
4) в пунктах 1—3 нет правильной формулировки.

А9. Конечность алгоритма означает, что:

1) каждый шаг алгоритма совершается за конечное время;
2) исполнение алгоритма завершается после конечного числа шагов;
3) исполнение алгоритма приводит к получению необходимого результата;
4) в пунктах 1—3 нет правильной формулировки.

А10. Цикл — это алгоритмическая конструкция, в которой предусматривается:

1) выбор для исполнения одного из двух блоков действий;
2) выбор исполнять или не исполнять некоторый блок действий;
3) повторение исполнения некоторого блока действий;
4) в пунктах 1—3 нет правильной формулировки.

А11. Ветвление в полной форме обеспечивает:
1) выбор для исполнения одного из двух блоков действий;
2) выбор исполнять или не исполнять некоторый блок действий;
3) повторение исполнения некоторого блока действий;
4) в пунктах 1—3 нет правильной формулировки.

А12. Формальные параметры вспомогательного алгоритма — это:
1) переменные, используемые во вспомогательном алгоритме;
2) аргументы вспомогательного алгоритма;
3) переменные, указанные в заголовке вспомогательного алгоритма;
4) количество переменных в заголовке вспомогательного алгоритма.

А13. Имеются две кучки камней, в одной из которых 13 штук, а в другой 17. Исполнитель имеет два допустимых действия:
а) взять из первой кучки 2 камня и переложить их во вторую;
б) взять из второй кучки 5 камней и переложить их в первую
и умеет проверять, остались ли в кучке камни.
Исполнитель выполняет алгоритм:

Делать пока (в первой кучке есть камни) **и** (во второй кучке есть камни)
{Взять из первой кучки 2 камня и переложить их во вторую;
Взять из второй кучки 5 камней и переложить их в первую;}

Через несколько шагов исполнитель прекратил работу, потому что:
1) кончились камни в первой кучке;
2) кончились камни во второй кучке;
3) не может исполнить допустимое действие *а*;
4) не может исполнить допустимое действие *б*.

А14. Значения переменных y и z после исполнения алгоритма, изображенного схемой на рисунке 1.19, таковы:
1) $y = 2,\ z = 1$; 2) $y = -8,\ z = 1$;
3) $y = 2,\ z = 2$; 4) $y = -8,\ z = 6$.

А15. На рисунке 1.20 изображена блок-схема алгоритма.
Значения переменных a и b после исполнения этого алгоритма:
1) $a = -144,\ b = 112$; 2) $a = 1,\ b = 5$; 3) $a = -29,\ b = 11$;
4) не совпадают ни с одной парой значений, указанных в пунктах 1—3.

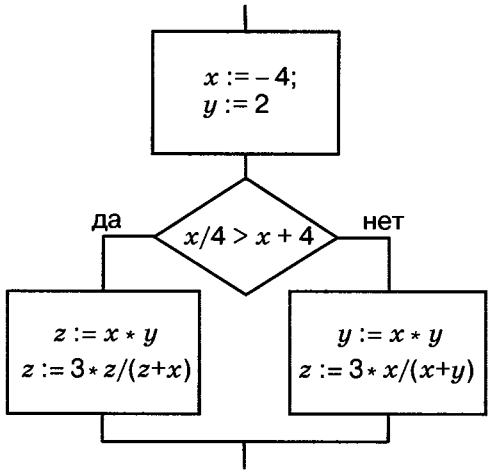


Рис. 1.19

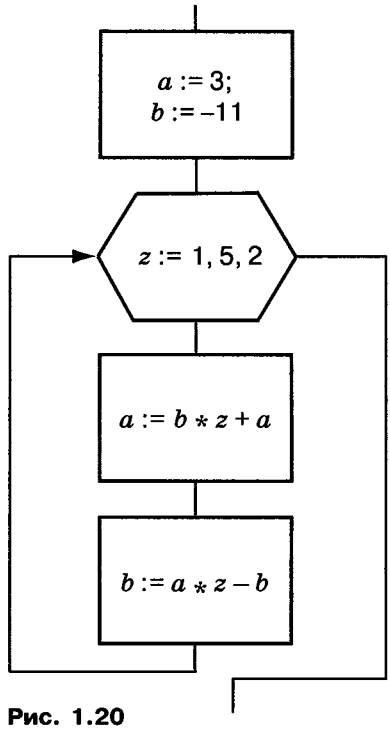


Рис. 1.20

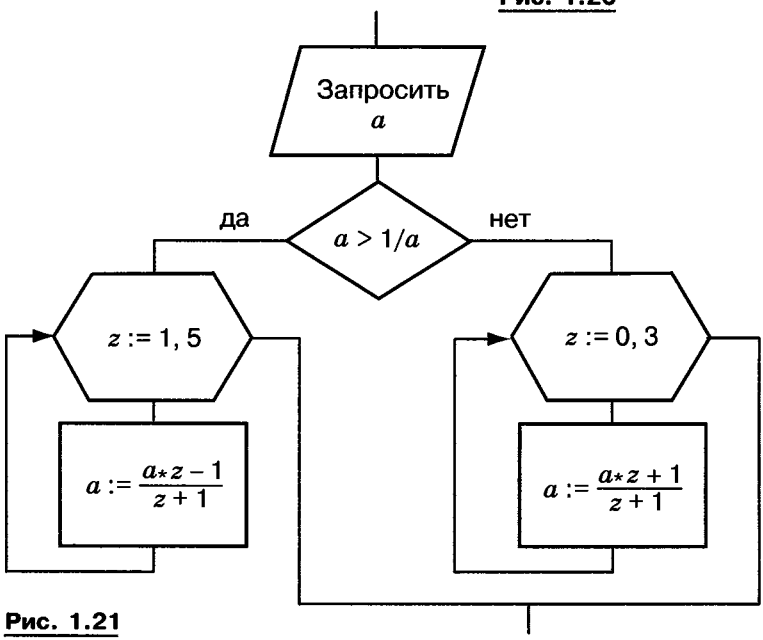


Рис. 1.21

А16. После исполнения алгоритма, изображенного схемой на рисунке 1.21, переменная a получила значение 1.
Значение, присвоенное переменной a в результате исполнения первого оператора алгоритма:

1) обязательно было равно 1; 2) могло быть равно −1;
3) обязательно было равно 2; 4) нельзя указать однозначно.

А17. Дан фрагмент алгоритма:

```
a := 1;
b := −1;
Делать пока a + b > a * b
{ a := b + 1;
  b := 2 * a;
}
```

Тело цикла будет исполнено:

1) 0 раз; 2) 1 раз; 3) 3 раза; 4) более 3 раз.

А18. Дан алгоритм:

```
Алгоритм
сим: A; цел: X, K, M
{ Запросить A;
  X := 0;
    Если (Часть (A, 2, 1) = Часть (A, 6, 1)) то
    { Делать от K := 1 до LEN(A) − 2
          { Делать от M := K + 1 до LEN(A) − 1
                { Если (Часть (A, K, 2) > Часть (A, M, 2))
                      то {X := X + 1; }
                }
          }
    }
}
```

После его исполнения переменная X имеет значение 4. Значение, присвоенное переменной A после исполнения первого оператора в теле алгоритма, могло быть:

1) ВАЛЕНТИНА; 2) ГАЛИНА; 3) ИРИНА; 4) МАРИНА.

А19. Дан алгоритм:

```
Алгоритм
цел: M;
{ Запросить M;
  Если (M < 0) то { M := −M; }
  Делать пока (M − 1) * (M − 8) > 0
  { M := M + 1;
  }
}
```

Количество тех целочисленных значений M, при которых данный алгоритм конечен, равно:

1) 1; 2) 9; 3) 17;

4) другому, нежели в пунктах 1—3, числу.

А20. Дан алгоритм:

```
Алгоритм
цел: K, M;
{ Запросить K;
  Запросить M;
  Делать пока (K mod 2 = 0 или M mod 2 = 0)
  { Если (K mod 2 = 0) то { M := M + K/2; } }
  Если (M mod 2 = 0) то { K := K + M/2; }
}
```

При введенных значениях K и M исполнение алгоритма завершилось через конечное число шагов. Тогда K и M:

1) могли быть равны 12 и 6 соответственно;

2) могли быть равны 5 и 10 соответственно;

3) могли быть равны 9 и 10 соответственно;

4) должны иметь иные значения, чем указанные в пунктах 1—3.

Часть 2. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и ответ в виде последовательности символов. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

В1. Целенаправленная замена одного объекта другим, сохраняющим существенные свойства исходного объекта, называется _____.

В2. Закончите предложение: «Для организации повторения блока действий используется алгоритмическая конструкция, которая называется _____».

В3. Дан алгоритм:

```
Алгоритм
цел: X;
{ Запросить X;
  Если ((X < 150) и (X > 75)) то
    { Делать пока X mod 2 = 0
        { X := X/2; }
    }
    иначе
    { X := X + 2;
      Если (X < 10 ) то { X := X - 3;}
```

```
    }
  Сообщить X;
}
```

После исполнения этого алгоритма было сообщено число 9. Какое число было присвоено переменной X по команде **Запросить** X?

B4. Дан алгоритм:

```
Алгоритм
сим: A; цел: X, K, M;
{ Запросить A;
  M := LEN(A);
  Делать от K := 1 до M − 1
  { A := A + Часть (A, M − K, 1);
  }
}
```

Ниже приведены четыре утверждения:
После выполнения этого алгоритма в переменной A окажется:

1) слово в 2 раза большей длины, чем исходное;

2) слово, одинаково читаемое слева направо и справа налево;

3) слово, в котором каждый символ обязательно встречается не менее двух раз;

4) слово, в котором хотя бы один символ встречается нечетное число раз.

Запишите в порядке возрастания без пробелов номера верных утверждений.

B5. Закончите предложение: «Чтобы изменить значение переменной, нужно использовать операцию _____».

B6. Укажите слово, которое надо поставить вместо пропуска в следующем предложении: «Для хранения и обработки текстовой информации используется _____ тип переменной».



2 ГЛАВА

Информационная деятельность человека и использование в ней компьютерных технологий

Обсуждая в § 1 понятия информации и информационных процессов, мы смотрели на них как бы со стороны. На самом деле человек едва ли не главная фигура в информационной картине мира. Именно человек сделал информацию глобальным ресурсом. Любой другой живой организм, любое животное сообщество воспринимает, аккумулирует и потребляет информацию, относящуюся к тому узкому мирку, который непосредственно его окружает. Лишь человечество в ходе своего развития непрерывно расширяет и сферу своего вмешательства, и сферу своих информационных интересов. Оно пытается заглянуть в микромир и в космические дали, узнать свое далекое прошлое и предсказать будущее. Добываемая человеком информация превращается в свой особый вид — в знание, которое позволило людям создать машины, взявшие на себя тяжесть физического труда, выращивать урожаи, способные прокормить непрерывно растущее по численности население Земли, создать искусственную среду обитания — теплые дома и одежду.

Более того, человечество придумало, как сохранять и передавать знания от поколения к поколению. Перекладывая на плечи машин физический труд, все больше и больше людей стало профессионально заниматься исключительно деятельностью, связанной с обработкой информации. И появление компьютеров, взявших на себя многие информационные функции, не только не уменьшило армию работников «информационного фронта», но, скорее, наоборот, лишь приумножило ее. Достаточно сказать, что в развитых странах число людей, занятых в информационной сфере, составляет в настоящее время более 60% трудоспособного населения (для сравнения скажем, что в сельском хозяйстве занято не более 4%, а в промышленности — около 18%). Постоянно растет «удельный вес» информационной составляющей в профессиональной деятельности и тех людей, которые работают не в информационной сфере. О том, какие информационные задачи обычно приходится решать людям в повседневной и профессиональной деятельности и как для этого используется компьютер, рассказывается в этой главе.

§11 Информационные задачи и этапы их решения

Люди ежедневно решают много разнообразных задач: что подарить другу на день рождения и как вывести страну из экономического кризиса? Чем лучше кормить рыбок в домашнем аквариуме и как уберечь планету от экологической катастрофы? Как организовать веселый школьный вечер и что надо сделать, чтобы улучшить жизнь в своем городе, поселке, деревне? Эти задачи совсем не похожи на задачи, содержащиеся в учебниках по математике, физике, химии и другим предметам.

Разнообразие задач, возникающих перед человеком и обществом, огромно. Как же человек решает такие задачи? Он отбирает нужную информацию, а затем, используя имеющиеся у него знания, преобразует исходную информацию в ответ к задаче. Вот тутто и оказываются нужными знания, которые накоплены человечеством за тысячелетия своего развития и частичку которых вы получаете, изучая школьные дисциплины. И хотя сказанное на первый взгляд кажется общими словами, можно сделать важный вывод: *прежде чем оперировать с реальными объектами с целью решить реальную задачу, человек оперирует с информацией об этих объектах, решая последовательно целый ряд информационных задач.*

Первая информационная задача на указанном пути состоит в том, чтобы определить, какую информацию об исходных объектах необходимо иметь и как ее получить. Сопровождающий решение этой задачи информационный процесс — получение информации.

Рассмотрим такую задачу: на мебельной фабрике нужно покрыть лаком столешницы изготовленных столов. Конечно, можно начать решать эту задачу, не подумав, — берешь баночку с лаком и лакируешь. Но тогда нет никакой уверенности, что лак не закончится раньше, чем будет решена производственная задача.

Поэтому сначала решают другую задачу — определить, сколько потребуется лака, чтобы покрыть им все изготовленные столешницы. И решают эту задачу уже не те рабочие, которые будут реальным лаком покрывать реальные столешницы, а работники планового отдела — здесь требуется не сам лак, а информация о его расходе. Вот так задача стала информационной.

Каждому ясно, как решить такую информационную задачу. Для этого надо знать площадь каждой столешницы, расход лака на единицу площади и количество изготовленных столешниц. Перемножив эти три числа, мы и будем иметь искомую величину. Правда, теперь возникает вопрос, как узнать, какую площадь имеет столешница. Если она прямоугольная, то надо измерить ее длину и ширину, а затем воспользоваться формулой площади прямоугольника. Если она круглая, то следует измерить ее диаметр и воспользоваться формулой площади круга.

Уже на этом простом примере видно, что после решения первой информационной задачи — получить информацию об исходных

объектах — приходится решать следующую информационную задачу — используя знания (в данном случае знание формулы для площади прямоугольника или для площади круга), преобразовывать имеющуюся информацию в информацию об ожидаемых результатах решения реальной задачи. Знания — это тоже, конечно, информация; она хранится в нашей памяти, в справочниках, в инструкциях по применению и т. д. и в нужный момент извлекается (т. е. передается) из своего хранилища.

Но возможно, что столешница имеет более причудливую форму, нежели прямоугольник или круг. Никакой готовой формулы для вычисления ее площади может просто не быть. Тогда поступим так: нальем лак в мерный стакан и затем лаком из этого стакана покроем столешницу. Разность между начальным и конечным уровнями лака в стакане укажет нам расход лака. Это и будет информация об исходном объекте.

Что же мы сделали в этом случае, чтобы получить исходную информацию? Мы провели эксперимент. Впрочем, если спросить, как были получены длина и ширина прямоугольной столешницы, то ответ будет такой же: экспериментально.

Рассмотренная нами задача очень проста. Нередко, однако, встречаются такие задачи, само решение которых представляет собой весьма важную информацию. Тогда эта информация переходит в **знания**, т. е. сохраняется в некотором обобщенном и систематизированном виде, готовая к применению в схожих ситуациях. Мы в последующих главах будем подробно обсуждать, как представлены знания в человеческом сообществе.

Итак, с какой бы реальной задачей ни приходилось иметь дело человеку, ее решение сопровождается информационной деятельностью, состоящей из трех этапов:

- на первом этапе определяется, какую исходную информацию необходимо иметь для решения задачи и как ее получить;
- на втором этапе привлекаются знания, позволяющие связать исходную информацию с результатами и определить, как именно получить требуемый результат;
- на третьем этапе исходная информация извлекается и преобразуется в результирующую информацию.

На первых двух этапах, как правило, осуществляется **анализ** информации, а на третьем — **синтез** новой информации.

Обратим внимание еще на одну особенность информации, с которой оперирует человек. Вся такая информация может быть разделена на два вида. Информация одного вида описывает объекты, явления, процессы, указывая на их свойства и отношения между ними. Информация другого вида определяет действия, которые должны быть произведены для преобразования одних объектов в другие, для изменения хода процесса, для возникновения или, наоборот, предотвращения того или иного явления. Информацию первого вида называют **декларативной**, информацию второго вида — **процедурной**. Информация вида «Земля вращается вокруг

Солнца» и «Вода замерзает при температуре 0 °С» — декларативная, а информация вида «Чтобы добыть огонь, надо быстро тереть друг о друга две плотно прижатые сухие деревянные палочки» и «Чтобы поймать рыбку, надо на крючок насадить приманку, забросить удочку в воду, подождать, пока рыбка клюнет, вытащить зацепившуюся за крючок рыбку» — процедурная. Иногда декларативную информацию легко превратить в процедурную и, наоборот, процедурную превратить в декларативную. Скажем, декларативную информацию «Площадь прямоугольника равна произведению его длины на ширину» каждый из вас способен почти мгновенно преобразовать в процедурную: «Чтобы найти площадь прямоугольника, надо измерить его длину и ширину, а затем полученные величины перемножить». Но иногда это сделать не так-то просто; вы убедитесь в этом, выполняя задания к данному параграфу. А иногда такое сделать просто невозможно в силу самого характера декларативной информации. Например, декларативную информацию «Период обращения Земли вокруг Солнца составляет примерно 365 с четвертью суток» в процедурную информацию преобразовать трудно.

ВОПРОСЫ И ЗАДАНИЯ

1. а) Приведите примеры, показывающие, как решение реальной задачи сопровождается соответствующими информационными процессами. Укажите, как в этих примерах реализуется три выделенных информационных этапа решения задачи.

   б) Для каждого из информационных этапов приведите примеры применения знаний на данном этапе решения задачи.

2. а) На каком из информационных этапов решения задачи о реальном объекте необходима экспериментальная деятельность?

   б) На всех ли информационных этапах решения задачи может потребоваться экспериментальная деятельность? Для этапов, где такая деятельность, по вашему мнению, может оказаться необходимой, подтвердите свою точку зрения примерами; для тех этапов, где, на ваш взгляд, эксперимент никогда не требуется, постарайтесь обосновать свою точку зрения.

3. Какая информация называется декларативной, а какая — процедурной?

4. Для каждого из приведенных ниже утверждений укажите, информацию какого вида — декларативного или процедурного — оно представляет:

   а) В грамм-молекуле любого вещества содержится $6{,}022 \cdot 10^{23}$ молекул (*число Авогадро*).

   б) Чтобы сложить две обыкновенные дроби, нужно их привести к общему знаменателю, который надо записать знаменателем результата, затем сложить получившиеся числители и записать получившуюся сумму в числитель результата.

   в) Закон Ома: сила тока, текущего по проводнику, прямо пропорциональна падению напряжения на концах проводника и обратно пропорциональна его сопротивлению.

   г) Кислота и основание реагируют с образованием соли и воды.

д) Если на тело не действует сила, то оно сохраняет состояние покоя или равномерного прямолинейного движения.

е) Чтобы найти длину гипотенузы прямоугольного треугольника, надо извлечь квадратный корень из суммы квадратов катетов.

ж) Угол падения луча света на поверхность равен углу отражения.

5 а) Рассмотрим еще раз примеры утверждений, записанных в объяснительном тексте: «Земля вращается вокруг Солнца», «Вода замерзает при температуре 0 °С», «Чтобы добыть огонь, надо быстро тереть друг о друга две плотно прижатые сухие деревянные палочки», «Чтобы поймать рыбку, надо на крючок насадить приманку, забросить удочку в воду, подождать, пока рыбка клюнет, вытащить зацепившуюся за крючок рыбку».

Для утверждений декларативного вида укажите, какие из них допускают преобразование к процедурному, а для утверждений процедурного вида укажите допускающие преобразование к декларативному и выполните эти преобразования.

б) Для утверждений из задания 4, которые имеют декларативный характер, выделите те, которые можно преобразовать в процедурную информацию, и запишите преобразованные утверждения.

в) Для утверждений из задания 4, которые имеют процедурный характер, выделите те, которые можно преобразовать в декларативную информацию, и запишите преобразованные утверждения.

г) Приведите свои примеры декларативной и процедурной информации, постарайтесь, чтобы среди примеров были утверждения как преобразуемые из одного вида в другой, так и не преобразуемые.

§12 Применение компьютера для решения простейших информационных задач

Впервые компьютер (называвшийся тогда еще ЭВМ) проявил себя не как инструмент быстрых вычислений, а как инструмент для обработки больших объемов информации в связи с созданием компьютерных баз данных и систем управления ими, хотя само понятие «база данных» в широком смысле не связано с понятием «компьютер». Ведь нередко **базу данных** определяют как **централизованное хранилище данных, в котором обеспечено хранение, первичная обработка и поиск информации**. В этом смысле любая картотека или даже каталог на бумажном носителе тоже представляют собой базы данных. Но именно создание компьютерных хранилищ систематизированной информации при обязательном наличии программ, позволяющих манипулировать с этой информацией, привело к появлению термина «база данных». Слова «база данных» будут нередко появляться в последующем тексте учебника, поэтому мы будем пользоваться общепринятым для них сокращением БД.

Каково же назначение базы данных? Решение каких информационных задач призвана обслуживать эта информационная технология? Ответ, конечно, ясен: хранение и поиск информации. Такой поиск **система управления базой данных** (сокращенно СУБД) про-

изводит по **запросу**, который пользователь адресует имеющейся в его распоряжении базе данных. Можно сказать, что основная функция базы данных — выдавать ответы на поступающие запросы. Если система управления базой данных не умеет отвечать на запросы, то все ее остальные функции бессмысленны.

Что такое запрос, мы обсудим несколько позже, а сейчас рассмотрим, какие бывают базы данных и каковы их возможности.

Каждая база данных предназначена для решения определенного класса задач. Для каждого класса задач характерен свой набор объектов и их **признаков**. Поэтому и соответствующие БД будут различными — одна БД нужна, чтобы разыскать нужную книгу, и совсем другая для облегчения работы директора школы. Даже если в разных базах объекты и одинаковы, наборы нужных признаков могут оказаться разными. Например, овощевода интересуют урожайность, сроки посева и уборки овощей, повара — калорийность овощей и рецепты овощных блюд, а директора овощехранилища — условия хранения (температура, влажность, предельная длительность хранения и т. п.). Для историка важны исторические сведения о странах, для географа — географические, для экономиста — экономические.

Как же представлены в БД сведения об объектах? Обычно эти сведения являются значениями тех признаков, которые, как мы видели, объявлены существенными для всех объектов той базы данных, с которой имеет дело пользователь. В теории баз данных признаки объектов называют **атрибутами**. Для книг набором таких атрибутов являются название книги, имя автора, год издания, издательство и т. д. Для работников некоторого предприятия атрибутами будут фамилия, имя, должность, заработная плата, домашний адрес, домашний телефон и т. д. Но для тех же людей в базе данных поликлиники часть атрибутов, конечно, сохранится, но появятся и новые, например какими болезнями болел данный человек.

Это небольшое обсуждение того, что такое база данных, показывает, что любая БД представляет собой информационную модель, в которой параметрами являются атрибуты объектов. Такие модели принято называть **фактографическими**, а сами базы данных — **информационными системами**.

Сейчас в мире созданы сотни тысяч баз данных. Они используются в библиотеках и больницах, в гидрометеоцентрах и на заводах, в магазинах и планирующих организациях, в банках и частных фирмах. Тематические базы данных для узких специалистов называют **банками данных**. Это может быть банк данных по микропроцессорам, банк по лекарственным средствам, банк публикаций в области ядерной физики и т. п.

В нашей стране действуют десятки банков данных. Пожалуй, самый большой из них — банк данных Российского института научной и технической информации. В нем содержится более 6 млн библиографических сведений о книгах и статьях практически по всем отраслям знаний. Тот, кто покупает авиабилеты, пользуется услугами другого крупного банка данных нашей страны — системой

«Сирена». С помощью компьютера кассир связывается с центральной большой ЭВМ, находящейся за тысячи километров от него, получает сведения о наличии мест на данный рейс и печатает билет.

Интернет также содержит большое количество самых разнообразных баз данных. Это и общедоступные базы данных, и базы данных сугубо специального назначения. Поиск в Интернете осуществляется с помощью поисковых систем. Среди русскоязычных поисковых систем лидерами можно назвать

Яndex (http://yandex.ru),
Google (http://google.ru),
Rambler (http://www.rambler.ru).

Ясно, что для эффективной обработки информации, представленной в БД, хранящиеся в ней данные должны быть сгруппированы в соответствии с той или иной структурой атрибутов. В зависимости от способа структурирования различают несколько видов баз данных: **иерархические, сетевые и реляционные.**

В иерархических базах данных атрибуты соподчинены между собой так, что данные образуют **древовидную** структуру. Эта структура заимствована из бескомпьютерных аналогов — различных каталогов, справочников и т. п. Скажем, авторский каталог книг. В нем главный атрибут — фамилия автора, следующий атрибут по значимости — название книги, следующий, к примеру, издательство, далее — год издания и т. д. Тем самым записи в таком каталоге будут упорядочены сначала в алфавитном порядке авторов, для одного автора — в алфавитном порядке названий, затем для одного названия — в алфавитном порядке издательств, наконец,

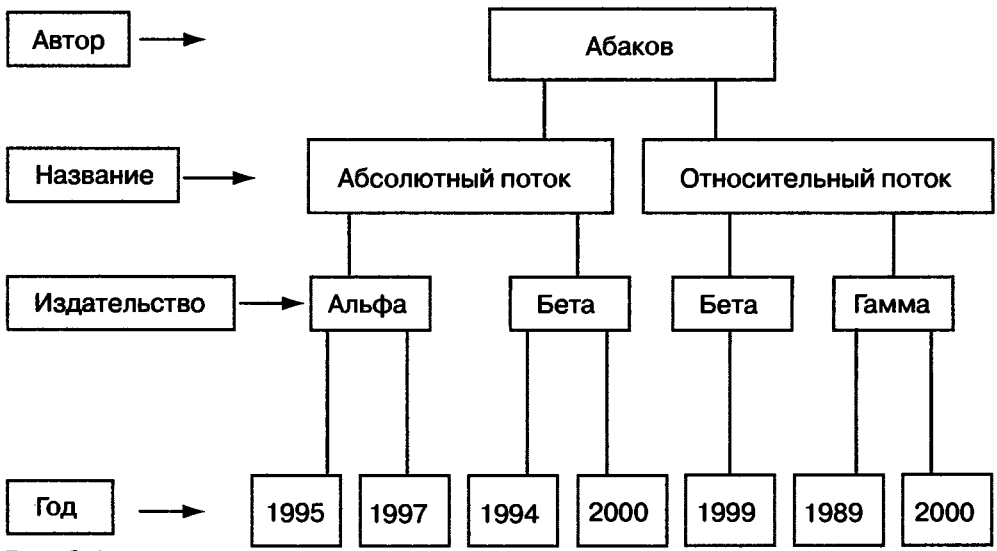


Рис. 2.1

для одного издательства — по годам издания. На рисунке 2.1 показано, как выглядит соответствующая структура записей.

Конечно, если требуется найти все произведения Абакова или любого другого писателя, то это сделать несложно. Однако найти в такой структуре, например, все издания, выпущенные издательством «Альфа», будет уже не так легко. Для подобных целей составляют другой каталог. Однако хранить много различных каталогов, т. е. иерархически организованных списков, накладно даже для компьютеров с большой памятью.

Сетевая база данных представляет собой более общую структуру, нежели иерархическая база: в ней элементы связаны произвольным образом. Каждый элемент может быть связан с произвольным количеством других элементов из разных уровней. Сетевую базу данных удобно проектировать, но трудно реализовать в ней управление так, чтобы эффективно осуществлялся поиск нужной информации.

Реляционные базы данных на сегодняшний день — один из наиболее распространенных видов компьютерных баз данных. В такой базе каждая запись представляет собой упорядоченный набор значений атрибутов. Например, в том же примере с книгами одна запись выглядит так:

Абаков, Абсолютный поток, Бета, 1994.

Все такие записи удобно представлять подписанными одна под другой в виде таблицы. Каждый столбец в такой таблице — это некоторый атрибут, а в соответствующую клетку занесено значение этого атрибута. Например, в таблице 2.1 представлена информация о книгах.

Реляционная база данных — это совокупность подобных таблиц (о том, почему ее называют реляционной, мы поговорим в § 34). Важно при этом, что между таблицами могут быть установлены связи, указывающие на совпадение множества значений для разных атрибутов. Это позволяет при поиске нужной информации переходить от одной таблицы к другой, получая ответы на такие запросы, которые нельзя получить с использованием только одной таблицы.

Чтобы получить из такой базы нужную информацию, составляют запрос, в котором указывают известные пользователю значения атрибутов и отмечают, какую информацию требуется получить, т. е. значения каких атрибутов вывести на экран компьютера.

Например, если нужно узнать, какие произведения печатал в издательстве «Альфа» писатель Абаков, то для атрибутов ***Автор*** и ***Издательство*** задаются значения Абаков и Альфа соответственно (можно считать, что значения этих атрибутов соединены союзом *и*), а в качестве запрашиваемого атрибута надо отметить ***Название***.

Допустим, вы разыскиваете произведение, про которое лишь помните, что то ли автор у него Абаков, то ли издавалось оно в издательстве «Альфа». Тогда составляется запрос, в котором значения

Таблица 2.1

Автор	Название	Издательство	Год
Абаков	Абсолютный поток	Альфа	1995
Абаков	Абсолютный поток	Альфа	1997
Абаков	Абсолютный поток	Бета	1994
Абаков	Абсолютный поток	Бета	2000
Абаков	Относительный поток	Бета	1999
Абаков	Относительный поток	Гамма	1989
Абаков	Относительный поток	Гамма	2000
...	...	...	...
Баков	Вечный полет	Тау	1904
...	...	...	...

атрибутов ***Автор*** и ***Издательство*** соединяются союзом ***или***. Некоторые из запрашиваемых атрибутов могут быть объявлены **обязательно присутствующими**. Например, любая книга всегда имеет какое-нибудь название.

Над данными, хранящимися в БД, можно выполнять некоторые операции. Одной из таких операций является **сортировка**. Можно дать указание СУБД расположить строки в таблице по возрастанию или убыванию значений заданного числового атрибута либо в алфавитном порядке, если атрибут имеет символьный тип.

Правила составления запросов и выполнения операций над данными у каждой базы данных, разумеется, свои; обычно они описываются в инструкции пользователю. С одной из СУБД вы познакомитесь, выполняя лабораторную работу 4.

ВОПРОСЫ И ЗАДАНИЯ

1. Что такое база данных? Что такое СУБД?
2. Как вы думаете, для чего предназначена операция сортировки в БД?
3. В пунктах этого задания указаны названия баз данных, которые предполагается создать. Для каждой из проектируемых баз назовите атрибуты, которые вы предложили бы иметь в этой БД. Укажите также для каждой из БД, какую структуру — иерархическую, реляционную или сетевую — уместнее всего, по вашему мнению, для нее использовать:

а) БД «Почтовые марки»;

б) БД «Металлы»;

в) БД «Страны Европы»;

г) БД «Органические кислоты»;

д) БД «Водоемы ...» (вместо многоточия поставьте название вашей области, края или республики).

4 Предложите какую-либо полезную, на ваш взгляд, базу данных и укажите входящие в ее состав атрибуты.

5 Пусть проектируется БД «Кулинарные рецепты». Какую структуру — иерархическую, реляционную или сетевую — уместнее всего, по вашему мнению, использовать для такой БД? Приведите аргументы в поддержку своей точки зрения.

6 В БД «Озера» хранятся следующие сведения об озерах мира:

- название озера — атрибут ***Название***;
- площадь, кв. км — атрибут ***Площадь***;
- максимальная глубина, м — атрибут ***Глубина***;
- континент, на котором расположено данное озеро — атрибут ***Континент***;
- высота над уровнем моря, м — атрибут ***Высота***;
- происхождение (тектоническое, лагуна, вулканическое, запрудное, ледниковое, карстовое) — атрибут ***Происхождение***;
- наличие или отсутствие стока — атрибут ***Сток***.

В запросе к этой БД указываются название атрибута и значения, по которым требуется провести поиск. Например, чтобы найти все озера с глубиной больше 100 м, составляется запрос

Глубина > 100,

а для нахождения африканских озер, не имеющих стока, потребуется запрос

Континент = Африка ***и Сток*** = Нет.

Составьте запросы, позволяющие найти:

а) все озера-карлики (площадь меньше 100 км^2);

б) все озера с площадью более 10 000 км^2 и глубиной более 500 м;

в) все озера, расположенные выше 500 м, кроме африканских;

г) все американские озера.

7 Какой запрос БД «Озера» (см. задание 6) надо сформулировать, чтобы выяснить:

а) все ли озера ледникового происхождения мельче 100 м;

б) все ли озера мельче 100 м имеют ледниковое происхождение?

8 Однажды школьник решил воспользоваться БД «Озера» (см. задание 6), чтобы найти все африканские и австралийские озера. Тут подошел злоумышленник и быстро составил следующий запрос:

Континент = Африка ***и Континент*** = Австралия.

В БД не оказалось озер, удовлетворяющих этому запросу. Объясните почему. Какой запрос надо составить, чтобы получить нужную школьнику информацию?

9 Спроектируйте фактографическую модель вашего класса. Для этого надо определить атрибуты, по значениям которых будут разыскиваться ответы на различные запросы — о домашнем адресе, о родителях, об успеваемости, об интересах и т. п. Составьте подходящий, на ваш взгляд, список атрибутов.

⑩ Выполнив задание 9, вы указали существенные параметры информационной модели вашего класса. Они станут атрибутами той базы данных «Класс», которую вы будете создавать, выполняя лабораторную работу 4. Ниже сформулированы вопросы, на которые должна уметь отвечать проектируемая вами база данных. Проверьте, достаточно ли для этого тех атрибутов, которые вы уже выделили, и, если нет, добавьте недостающие:

а) У кого день рождения в мае?

б) Кто живет на той же улице, где расположена школа?

в) Кто самый младший ученик в классе?

г) У кого дома есть собака?

д) Кто из мальчиков класса самый старший?

е) Кто посещает спортивные секции?

ж) У кого первого в текущем учебном году был день рождения?

з) Кто из мальчиков занимается в первой подгруппе иностранного языка?

и) Имеются ли в классе однофамильцы?

к) Имеются ли в классе братья и сестры?

л) У кого в классе имеются братья или сестры?

§ 13 Эксперимент как способ познания. Компьютерная обработка результатов эксперимента

Наблюдение по праву считается одним из краеугольных камней в процессе познания окружающего мира. Наблюдая за эффектами, повторяющимися в одних и тех же условиях, человек обнаруживал закономерные связи между явлениями. Иногда эти закономерности не выходят за рамки народных примет (например, ярко-красный закат — к ветру на следующий день). Но и научные открытия нередко основаны именно на наблюдениях. Многолетние астрономические наблюдения Тихо Браге легли в основу сформулированных И. Кеплером законов движения планет. Случайно проявленная фотопластинка привела к открытию рентгеновских лучей. А вот как описывает Майкл Фарадей свое открытие явления электромагнитной индукции: «Я взял цилиндрический магнитный брусок и ввел один его конец в просвет спирали из медной проволоки, соединенной с гальванометром. Потом я быстрым движением втолкнул магнит внутрь спирали на всю его длину, и стрелка гальванометра качнулась. Затем я так же быстро вытащил магнит из спирали, и стрелка опять качнулась, но в противоположную сторону. Качания стрелки повторялись всякий раз, как магнит вталкивался или выталкивался. Это значит, что электрическая волна возникает только при движении магнита, а не в силу свойств, присущих ему в покое». Почему спираль, почему движение магнита — это удел гения предугадывать, при каких условиях удастся обнаружить искомый эффект. Хотя нелишне напомнить, что от момента, когда ученый поставил перед собой задачу «превратить магнетизм в электричество», до открытия им электромагнитной индукции прошло 10 лет.

В современной науке эксперимент занимает важное место, для его проведения нередко требуется сложная аппаратура, его ход контролируется с помощью компьютера, а для обработки полученных в эксперименте результатов используются компьютерные технологии. Вот о некоторых методах обработки результатов эксперимента и пойдет речь в данном параграфе. Рассказывать о них мы будем на примерах опытов, которые вы проводили на уроках физики.

Наш первый опыт будет относиться к закону Ома. Сам Г. Ом пришел к его формулировке из следующей аналогии. Электрический ток — это направленное движение (ток) заряженных частиц. Г. Ом уподобил поток этих частиц воде в реке, расход которой в единицу времени тем больше, чем больше разность уровней, определяющая само существование течения, и чем меньше преград на ее пути. Формулировка закона всем известна. Если обозначить через I силу тока, через U падение напряжения на участке цепи, а через R сопротивление, то закон Ома запишется формулой $I = \frac{U}{R}$.

Представьте себя на месте Г. Ома, только что сформулировавшего закон. Перед нами стоит задача проверить его экспериментально. Как решить эту задачу? Вспомним (см. § 5), что на первом этапе нужно определить, что будет служить исходной информацией. Мы можем непосредственно измерить напряжение и силу тока, тогда значения величин U и I дадут нам исходную информацию. Гипотеза — именно она выступает здесь в роли знания, применяемого на втором этапе решения задачи, — утверждает, что отношение этих величин должно быть постоянным для каждого фиксированного образца проводника электрического тока. Теперь ясно, как проводить эксперимент (вот мы и перешли к третьему этапу решения задачи): берется несколько различных проводников, на их концы подается различное напряжение и замеряется сила электрического тока, текущего по данному проводнику. Если отношение напряжения к силе тока меняется лишь при замене одного проводника другим, то гипотезу можно считать подтвержденной, а значение отношения как раз и принять за величину сопротивления данного проводника.

Пусть для некоторого проводника получили экспериментальные данные, представленные в таблице 2.2.

Таблица 2.2

U, В	3	4,5	6	7,5	9	10,5	12	13,5	15
I, А	0,031	0,045	0,055	0,074	0,092	0,106	0,119	0,138	0,148
$\frac{U}{I}$	0,0103	0,0100	0,0091	0,0099	0,0102	0,0101	0,0099	0,0102	0,0099

В третьей строке таблицы мы подсчитали искомое отношение. Видно, что оно действительно практически постоянно. Отклонения

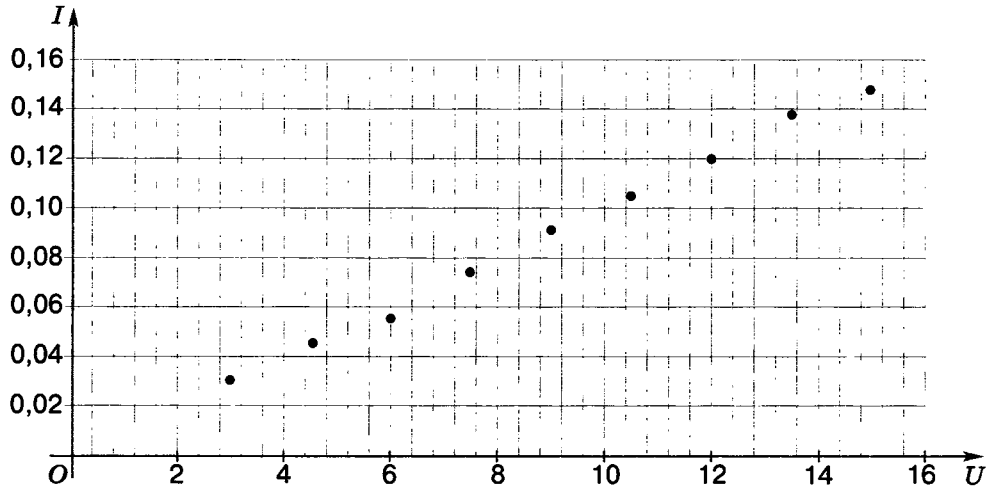


Рис. 2.2

легко объяснимы — ни в каком эксперименте нельзя совершенно точно произвести измерения, да и свойства проводника подвержены влиянию различных факторов: немного изменилась температура или качество контактов и т. п.

Но какой же принять величину измеряемого сопротивления? Довольно часто берут среднее арифметическое полученных результатов. Однако теория ошибок говорит, что более правильно поступить иначе. Нанесем наши экспериментальные данные на график (рис. 2.2).

На этом графике видно, что полученные в эксперименте значения практически точно ложатся на некоторую прямую. Ее угловой коэффициент k равен $\frac{I}{U}$, т. е. согласно закону Ома $k = \frac{1}{R}$. Тем самым перед нами стоит задача найти прямую, которая наилучшим образом проходит через экспериментальные точки. Такая задача может возникнуть не только при измерении сопротивления проводника, поэтому уделим ей отдельное внимание, отложив на время обсуждение физической стороны проблемы.

Итак, пусть при значениях переменной x, указанных в первой строке таблицы 2.3, получены соответствующие значения переменной y, и мы предполагаем, что эти переменные связаны зависимостью $y = kx$.

Таблица 2.3

x	x_1	x_2	...	x_{n-1}	x_n
y	y_1	y_2	...	y_{n-1}	y_n

Искомая прямая выбирается так, чтобы точки, соответствующие экспериментальным данным, находились как можно ближе к прямой, изображающей зависимость $y = kx$. Вряд ли можно

ожидать, что прямую удастся расположить так, чтобы все точки оказались одинаково близкими к искомой прямой. Поэтому в качестве критерия близости выбирают сумму квадратов отклонений экспериментальных точек от прямой (на рисунке 2.3 каждое такое отклонение изображено вертикальным отрезком).

Запишем указанную величину формулой

$$(y_1 - kx_1)^2 + (y_2 - kx_2)^2 + \ldots + (y_{n-1} - kx_{n-1})^2 + (y_n - kx_n)^2 = \sum_{i=1}^{n} (y_i - kx_i)^2.$$

В ней единственной неизвестной величиной является k, так что можно считать, что мы имеем дело с функцией, аргументом которой является k. Нам требуется найти минимум этой функции. Раскроем скобки и перепишем выражение так, чтобы ясно было, какой вид имеет данная функция:

$$F(k) = (\sum_{i=1}^{n} x_i^2)k^2 - 2(\sum_{i=1}^{n} x_i y_i)k + (\sum_{i=1}^{n} y_i^2).$$

Теперь видно, что $F(k)$ — квадратичная функция вида

$ak^2 + bk + c$, где

$$a = \sum_{i=1}^{n} x_i^2,\ b = -2\sum_{i=1}^{n} x_i y_i,\ c = \sum_{i=1}^{n} y_i^2.$$

Заметим, что старший коэффициент получившегося квадратного трехчлена всегда положителен. Такая функция, как вам известно из курса математики, принимает наименьшее значение при

$$k = -\frac{b}{2a} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}.$$

Это значение k и принимается за искомое. Метод, посредством которого было найдено значение коэффициента пропорциональности, называют методом наименьших квадратов.

Вернувшись к задаче о вычислении сопротивления, мы можем написать, что

$$\frac{1}{R} = k = \frac{I_1U_1 + I_2U_2 + \ldots + I_{10}U_{10}}{U_1^2 + U_2^2 + \ldots + U_{10}^2}.$$

Электронную таблицу для вычисления сопротивления по этой

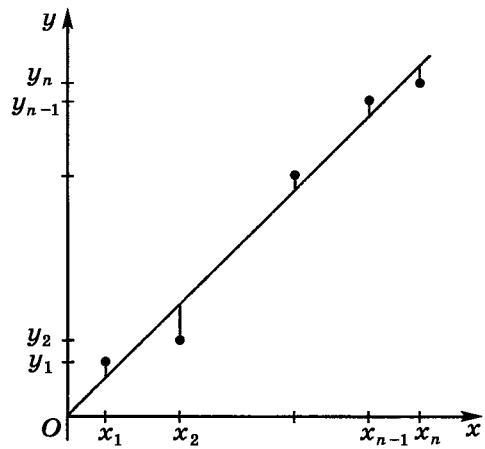


Рис. 2.3

Таблица 2.4

	A	B	C	D	E
1	Напряжение, U	Сила тока, I	U^2	I*U	SUM(C2:C11)/SUM(D2:D11)
2	$<U_1>$	$<I_1>$	A2^2	A2*B2	
3	$<U_2>$	$<I_2>$	A3^2	A3*B3	
...	...	...	...	...	
11	$<U_{10}>$	$<I_{10}>$	A11^2	A11*B11	

формуле можно заполнить так, как показано в таблице 2.4; в ячейке E1 вычисляется сопротивление R.

Вычисления с помощью электронной таблицы вы проведете, выполняя лабораторную работу 6. А сейчас обсудим один из возможных вариантов проведения эксперимента по измерению сопротивления.

Нередко для того, чтобы точнее установить результат эксперимента, его повторяют при одних и тех же условиях. Пусть и у нас проведено n опытов при одном и том же значении напряжения U. Показания амперметра при этом все равно могли получиться различными. Тогда для k мы получаем следующую формулу:

$$k = \frac{I_1U + I_2U + \ldots + I_nU}{nU^2} = \frac{1}{n}\left(\frac{I_1}{U} + \frac{I_2}{U} + \ldots + \frac{I_n}{U}\right) = \frac{k_1 + k_2 + \ldots + k_n}{n},$$

где через k_m обозначено отношение силы тока к падению напряжения, получаемое в опыте с номером m. Оказалось, что в этом случае коэффициент k равен среднему арифметическому коэффициентов, вычисленных по результатам каждого из опытов.

Общий же вывод можно сформулировать так: *если в эксперименте не изменяются условия его проведения, то для измеряемой величины в качестве результата нужно взять среднее арифметическое результатов всех проведенных опытов.*

По методу наименьших квадратов можно разыскивать и другие линии, которые будут достаточно удовлетворительно описывать экспериментально выявленные зависимости. Если речь идет о линейной, но не прямо пропорциональной зависимости, т. е. зависимости вида $ax + b$, где $b \neq 0$, то коэффициенты a и b будут вычисляться как решение системы уравнений:

$$\begin{cases} a\sum\limits_{i=1}^{n} x_i^2 + b\sum\limits_{i=1}^{n} x_i = \sum\limits_{i=1}^{n} x_i y_i, \\ a\sum\limits_{i=1}^{n} x_i + nb = \sum\limits_{i=1}^{n} y_i. \end{cases}$$

Если речь идет о квадратичной зависимости $ax^2 + bx + c$, то коэффициенты a, b и c получаются как решение следующей системы уравнений:

$$\begin{cases} a\sum\limits_{i=1}^{n} x_i^4 + b\sum\limits_{i=1}^{n} x_i^3 + c\sum\limits_{i=1}^{n} x_i^2 = \sum\limits_{i=1}^{n} x_i^2 y_i, \\ a\sum\limits_{i=1}^{n} x_i^3 + b\sum\limits_{i=1}^{n} x_i^2 + c\sum\limits_{i=1}^{n} x_i = \sum\limits_{i=1}^{n} x_i y_i, \\ a\sum\limits_{i=1}^{n} x_i^2 + b\sum\limits_{i=1}^{n} x_i + nc = \sum\limits_{i=1}^{n} y_i. \end{cases}$$

Аналогично можно вычислять коэффициенты для представления экспериментальных данных многочленами более высоких степеней.

Преимущество метода наименьших квадратов перед другими методами состоит в том, что имеется простой вычислительный алгоритм для нахождения коэффициентов функции, приближающей таблично заданную функцию, — нужно решить легко составляемую систему уравнений. Есть и другая причина того, что метод наименьших квадратов активно используется при обработке экспериментальных данных. Она состоит в том, что ошибки измерений, как правило, подчинены так называемому закону больших чисел: ошибки с большими значениями встречаются намного реже ошибок с маленькими значениями. Можно математически доказать, что в этом случае именно метод наименьших квадратов дает ту теоретическую кривую, которая должна была бы получиться, если бы измерения производились абсолютно точно.

ВОПРОСЫ И ЗАДАНИЯ

1. В чем состоит метод наименьших квадратов? В каких случаях целесообразно его применение?
2. Как должна выглядеть формула для нахождения сопротивления некоторого фиксированного проводника, если бы в эксперименте задавали силу тока, а соответствующее падение напряжения на концах проводника измеряли?
3. Во время подъема на воздушном шаре производились измерения температуры с целью установления зависимости температуры от высоты над уровнем моря. Вот данные этих измерений:

Высота, км	1,2	1,8	2,2	2,5	3,0	3,3	3,8
Температура, °С	15	11	8	6	3	1	−2

Предполагая, что зависимость задается линейной функцией $ax + b$, подготовьте заполнение электронной таблицы для нахождения коэффициентов a и b по методу наименьших квадратов.

4* Попытайтесь обосновать метод наименьших квадратов для нахождения коэффициентов в случае гипотезы о линейной зависимости между измеряемыми величинами.

§ 14 Алгоритм как форма организации процедурной информации

В § 11 мы обсудили, что информация, которой располагает человек, может быть декларативной или процедурной. Напомним, что процедурная информация указывает, как именно получить требуемый результат. Наиболее простой способ представления такой информации — это указание тому, кто будет решать задачу, последовательности действий, приводящих к достижению цели. В § 7 мы вам напомнили, что организованную последовательность допустимых для исполнителя действий, приводящую к определенному результату, называют алгоритмом.

Вот пример представления процедурной информации, списанный нами с упаковки чая:

Как приготовить хороший чай:

- Вскипятите свежую воду.
- Ополосните заварочный чайник крутым кипятком.
- Положите чай в заварочный чайник из расчета одна чайная ложка на чашку и сразу же залейте кипятком.
- Через 3 минуты размешайте.
- Добавьте сахар по вкусу.

Посмотрев на пачку чая с другой стороны, мы обнаружили тот же алгоритм, записанный на английском языке:

How to make a good tea:

- Bring fresh water to boil.
- Warm teapot by rinsing out with hot water.
- Put one teaspoonful of tea per cup into the teapot and pour immediately the boiling water into the tea.
- Stir the tea after 3 minutes.
- Add sugar to taste.

Ни у кого не возникает сомнений, что это один и тот же алгоритм. Только одна его запись сделана для исполнителя, понимающего английский язык, а другая — для русскоговорящего испол-

нителя. Ясно, что каждый из исполнителей будет выполнять одни и те же действия, и поэтому, наблюдая за их работой, мы не сможем отличить одного от другого. Более того, можно снять на видеокассету все манипуляции при исполнении алгоритма, а затем показать любому жителю Земли, и ему будет ясно, как заваривать чай. И совершенно не важно, какой язык понимает этот житель! Для нас же важен вывод, что один и тот же алгоритм может быть записан разными способами: в виде текста, схемой алгоритма, функциональной схемой для машины Тьюринга и т. п. Запись алгоритма на языке, понятном формальному исполнителю, называется **программой**.

Ясно, что алгоритм, записанный на пачке чая, предназначен для человека. Человеку вообще нередко приходится исполнять различные алгоритмы. Но гораздо приятнее поручить такую работу какому-нибудь автоматическому устройству, т. е. формальному исполнителю. Более того, нередко формальный исполнитель справится с алгоритмом гораздо лучше человека — человек может отвлечься и неточно выполнить предписанное действие, может вообще задуматься, стоит ли исполнять данный алгоритм, а может начать его модифицировать, и тогда есть риск, что исполнение алгоритма вообще никогда не закончится. А все потому, что человек, напомним, является не формальным, а эвристическим исполнителем.

Зато в умении составлять алгоритмы на сегодняшний день никакой формальный исполнитель не может превзойти человека. Существуют алгоритмы, позволяющие для некоторых классов задач автоматически строить алгоритмы решения, но эти классы задач весьма узкие. Люди придумали методы, облегчающие создание алгоритмов. Одним из таких методов является **метод пошаговой детализации**. Продемонстрируем этот метод на следующей задаче.

Еще с 5 класса вы знаете, что любое натуральное число, большее 1, можно представить в виде произведения степеней различных простых чисел. Скажем, $720 = 2^4 \cdot 3^2 \cdot 5^1$, $32 = 2^5$, $71 = 71^1$. Требуется составить алгоритм, который для каждого натурального $n > 1$ указывал бы все простые множители, входящие в такое разложение, и показатели степеней, с которыми они присутствуют в этом разложении. Первым шагом должно быть определение хотя бы одного простого множителя, делящего данное число n. Дальше путь раздваивается: можно попытаться определить показатель степени, в которой уже найденный простой множитель входит в разложение числа n, или попытаться найти следующий простой множитель, а только потом, когда будут найдены все простые множители, для каждого из них найти показатель степени, с которым он входит в разложение числа n. На рисунке 2.4 в виде схемы изображены два способа решения этой задачи.

Теперь проведем детализацию полученных алгоритмов. Для решения поставленной задачи нужно научиться решать две другие:

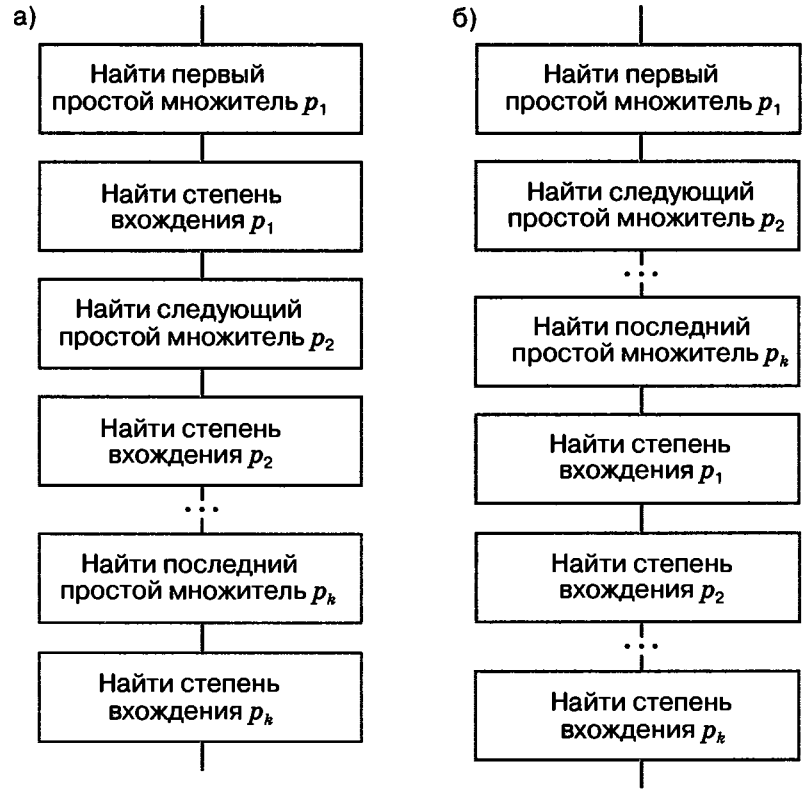


Рис. 2.4 Поиск разложения натурального числа на произведение простых множителей: а) 1-й вариант пошаговой детализации; б) 2-й вариант пошаговой детализации

находить простой множитель заданного натурального числа n и вычислять показатель степени, с которой он входит в разложение числа n. А поскольку количество различных простых множителей заранее неизвестно, то придется воспользоваться конструкцией цикла в форме **Делать пока**. Будем далее рассматривать первый способ решения. Преобразуем уже созданный алгоритм (рис. 2.4, *а*) в схему, изображенную на рисунке 2.5. В этой схеме алгоритма дважды употреблен блок, изображенный на рисунке 2.6.

Из курса информатики, который вы изучали в предшествующих классах, вам известно, что так обозначается обращение к вспомогательному алгоритму или подпрограмме. И в нашем случае для того, чтобы данная схема алгоритма приобрела полноценный вид, в каждый такой блок надо записать имя вспомогательного алгоритма и указать его формальные параметры: аргументы и результаты. Поиск наименьшего простого множителя натурального числа n можно записать следующим вспомогательным алгоритмом:

```
Алгоритм Простой_множитель
(арг: n; рез: p)
  цел: n, p;
  { p := 2;
    Делать пока n mod p ≠ 0
      { p := p + 1;
      }
  }
```

Чтобы обратиться к вспомогательному алгоритму, надо в команде вызова указать имена фактических переменных (или значения), с которыми будет исполняться вспомогательный алгоритм. Например, по команде

```
Вызвать Простой_множитель
(15, m);
```

переменной m будет присвоено значение 3.

Нередко вспомогательный алгоритм, имеющий в качестве результата только один формальный параметр, удобно оформить в виде подпрограммы-функции. Для написанного выше вспомогательного алгоритма это будет выглядеть так:

```
Функция Простой_множитель
(арг: n): цел
цел: p;
  { p := 2;
    Делать пока n mod p ≠ 0
      { p := p + 1;
      }
    знач := p;
  }
```

Здесь **знач** — имя внутренней переменной, предназначенной для хранения значения функции. А чтобы воспользоваться подпрограммой-функцией, достаточно написать

m := Простой_множитель (15); .

Чтобы завершить составление алгоритма, нужно еще составить алгоритм нахождения показателя степени, в которой данный множи-



Рис. 2.5

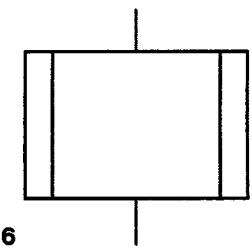


Рис. 2.6

тель входит в разложение числа n. Вы составите этот алгоритм, выполняя задание 3.

Умение решать многие задачи эвристически, т. е. не имея никакого алгоритма, — чрезвычайно важная способность человека. Например, задачу определения, кто изображен на фотографии — собака или кошка, человек решает с высоким процентом правильных ответов. Однако никакого алгоритма, позволяющего решить такую задачу, на сегодняшний день нет. Слишком слабы наши возможности формализации этой задачи, и построение адекватной информационной модели пока невозможно.

Может показаться, что проблема создания алгоритма для решения задачи упирается именно в неумение до поры до времени формализовать стоящую перед нами задачу. Речь, конечно, идет о так называемой массовой задаче, т. е. о целом множестве однотипных задач, для решения которых применяется один и тот же алгоритм. Оказывается, что это не так.

Нетрудно доказать, что не существует алгоритма, пригодного для решения любой задачи. Доказательство проведем в два этапа.

Первый этап. Докажем, что для любого исполнителя существует задача, которую он решить не может. Предположим, что это не так: пусть есть исполнитель, который может посредством своих допустимых действий решить любую задачу. Поставим перед ним задачу создать камень, который он не может поднять. А теперь, когда он этот камень создал, поставим перед ним другую задачу — поднять этот камень. Налицо противоречие. Значит, наше предположение неверно, и, следовательно, для любого исполнителя (в том числе формального) существует задача, которую он решить не может.

Приведенное рассуждение — перефразированный аргумент великого немецкого философа И. Канта, жившего в г. Кенигсберге с 1724 по 1804 год, который он привел в доказательство того, что Бог не может быть всемогущим.

Второй этап. Предположим, что существует алгоритм, пригодный для решения любой задачи. Тогда этот алгоритм предназначен для некоторого формального исполнителя. Но исполнитель, вооруженный данным алгоритмом, способен решить любую задачу, что невозможно. Полученное противоречие показывает, что сделанное предположение неверно.

Итак, не существует алгоритма, решающего любые задачи. Но это еще не дает ответа на сформулированный выше вопрос: не в том ли причина отсутствия алгоритма для решения той или иной массовой задачи, что мы не можем ее формализовать? Отрицательный ответ мы уже анонсировали. Теперь мы можем привести пример другой задачи.

Рассмотрим множество алгоритмов, предназначенных для обработки символьной переменной x (т. е. в качестве входного данного для алгоритма выступает некоторый текст). Представим все эти алгоритмы записанными на каком-либо языке програм-

мирования. Задача состоит в том, чтобы по тексту алгоритма и значению x узнать, закончит ли алгоритм работу за конечное число шагов.

Оказывается, что для этой задачи не существует алгоритма ее решения. Докажем это, рассуждая и на этот раз от противного.

Пусть существует алгоритм решения этой задачи. Назовем его алгоритмом A. Через B обозначим произвольный алгоритм из указанного множества алгоритмов. Входными данными для алгоритма A служат текст алгоритма B и значение символьной переменной x. Договоримся, что в результате своей работы алгоритм A присваивает некоторой переменной z значение 1, если алгоритм B заканчивает работу на тексте x за конечное число шагов, и значение 0 в противном случае.

Иными словами, алгоритм A выглядит так:

```
Алгоритм A (арг: B, x; рез: z)
  сим: B, x;
  цел: z;
  { Запросить B;
  Запросить x;
  Если (алгоритм B заканчивает работу на тексте x за конечное число
        шагов) то
    { z := 1; }
  иначе
    { z := 0; }
  }
```

Рассмотрим множество алгоритмов, которые в качестве допустимого значения x берут свое описание на указанном языке программирования. Построим алгоритм C, принадлежащий этому классу:

```
Алгоритм C
  сим: B, x;
  цел: z;
  { Запросить B;
    Запросить x;
    z := 1;
    Делать пока z = 1
       { Вызвать A(B, x, z);
       }
  }
```

Запустим составленный нами алгоритм C на исполнение. На запрос значений B и x введем текст алгоритма C. На следующем шаге переменной z будет присвоено значение 1, поэтому затем начнется исполнение тела цикла. Что же произойдет после обращения к вспомогательному алгоритму A?

Допустим, что алгоритм C конечен. Тогда после исполнения алгоритма A значение $z = 1$. Поэтому начнет исполняться тело цик-

ла. Однако в результате исполнения в теле цикла того же вспомогательного алгоритма значение переменной z не изменится. Поэтому тело цикла будет исполняться бесконечное число раз, и, следовательно, алгоритм C конечным не является. Значит, такая ситуация невозможна.

Допустим теперь, что алгоритм C конечным не является. Тогда после исполнения алгоритма A значение $z = 0$. В этом случае тело цикла не будет исполняться ни разу, и, следовательно, алгоритм C прекратит работу за конечное число шагов. Тем самым и в этом случае получилось противоречие. Значит, предположение, что существует алгоритм A, было ложным.

Задачи, для которых не существует алгоритма решения, называются **алгоритмически неразрешимыми**. Сегодня известно уже много алгоритмически неразрешимых задач. И как правило, доказательство их алгоритмической неразрешимости отнюдь не такое простое, как для рассмотренного нами примера. Одна из самых знаменитых задач была сформулирована в 1900 году Давидом Гильбертом на Всемирном математическом конгрессе. Всего он сформулировал 23 проблемы, решение которых, как он считал, определит лик математики XX века. И действительно, решение, даже частичное, любой из его проблем рассматривается как высшее математическое достижение. Проблема под номером 10 звучит так: «Дано произвольное уравнение $P(x_1, x_2, ..., x_n) = 0$, где $P(x_1, x_2, ..., x_n)$ — многочлен с целыми коэффициентами от указанных переменных. Существует ли алгоритм, позволяющий выяснить, имеет ли это уравнение решение в целых числах?» Лишь спустя 70 лет наш соотечественник Ю. В. Матиясевич доказал, что такого алгоритма не существует.

Пришла пора немного отвлечься и попить чайку. Например, того, что приготовили по написанным выше алгоритмам какой-нибудь русский школьник Ваня и его английский приятель Джонни. К нашему величайшему изумлению, чаек оказался разным: от Ваниного чая язык прилипает к губам, а чай Джонни почти несладкий. Почему один и тот же алгоритм в разном исполнении привел к столь различным результатам?

Все дело в последнем действии: добавлении сахара по вкусу. Это, конечно, допустимое действие для каждого исполнителя. Просто у каждого свой вкус. А мы можем сделать важный вывод:

Результат исполнения алгоритма зависит от того, какой исполнитель этот алгоритм исполняет.

ВОПРОСЫ И ЗАДАНИЯ

1. а) Что такое вспомогательный алгоритм?

   б) Когда можно вспомогательный алгоритм оформить в виде подпрограммы-функции?

2 а)* Проанализируйте алгоритм Простой_множитель и попытайтесь объяснить, почему в качестве результата будет получено простое число.
б) Объясните, почему этот алгоритм конечен.

3 а) Составьте вспомогательный алгоритм для нахождения показателя степени, в которой данный множитель входит в разложение числа n. Оформите его в виде подпрограммы-функции с двумя формальными параметрами.
б) Используя составленные вспомогательные алгоритмы, разработайте алгоритм решения основной задачи: построение разложения натурального числа n в произведение различных простых множителей.

4 а) В § 7 записан алгоритм Евклида вычисления наибольшего общего делителя двух натуральных чисел. Оформите его в виде подпрограммы-функции.
б) Функцией Эйлера $\varphi(n)$ называется количество натуральных чисел, меньших n и взаимно простых с n. Используя подпрограмму-функцию вычисления НОД, составьте подпрограмму-функцию для вычисления функции Эйлера.
в) Обозначим через $\psi(n)$ функцию, которая каждому натуральному числу n сопоставляет сумму всех натуральных чисел, меньших n и взаимно простых с n. Используя подпрограмму-функцию вычисления НОД, составьте подпрограмму-функцию для вычисления $\psi(n)$.

5 а) Составьте подпрограмму-функцию, которая по заданному натуральному числу n вычислит сумму его цифр.
б) Составьте алгоритм, подсчитывающий количество n-значных натуральных чисел, у каждого из которых сумма цифр равна заданному натуральному числу k.

6 а) Дано слово W. Составьте подпрограмму-функцию, которая преобразует слово W в слово V, содержащее те же буквы (и в том же количестве), что и слово W, но расположенные в алфавитном порядке.
б) Даны слова V и W. Составьте алгоритм, позволяющий установить, можно ли слово W получить из слова V перестановкой букв.

7 а) На координатной плоскости заданы своими координатами три точки. Составьте алгоритм, позволяющий определить, являются ли эти точки вершинами треугольника. Все координаты являются целыми числами.
б)* На координатной плоскости заданы своими координатами три точки A, B и C. Составьте алгоритм, позволяющий для четвертой точки определить, лежит ли эта точка внутри $\triangle ABC$. Все координаты являются целыми числами. (Совет: Для разработки алгоритма воспользуйтесь методом пошаговой детализации.)

8 На ферме установлены следующие расценки: бык стоит 25 000 р., корова — 16 000 р., теленок — 2000 р. Составьте алгоритм, позволяющий найти все варианты покупки на этой ферме 100 голов скота на сумму 500 000 р.

9* Имеется стержень длины n, где n — натуральное число. Этот стержень разламывают в двух точках так, что получаются отрезки целочисленной длины. Составьте алгоритм, позволяющий узнать, сколько различных треугольников может получиться, если по-разному выбирать точки разлома.

10 а) Петя и Коля в роли формальных исполнителей выполняют с помощью микрокалькулятора следующий алгоритм:

Сложить 83,2438 и 57,6847
Полученный результат умножить на 10

У Пети на табло калькулятора помещается 6 цифр, а у Коли — 8. Будут ли у них одинаковые результаты выполнения этого алгоритма?
б) Приведите другие примеры алгоритмов, результаты исполнения которых зависят от их исполнителей.

Ⓗ * Имеется чан объемом 1000 л. Автоматическое устройство, обслуживающее этот чан, имеет два допустимых действия: долить в чан еще 1 л воды; долить в чан столько воды, сколько в нем уже имеется. Составьте алгоритм, который для каждого натурального числа $n \leqslant 1000$ будет составлять алгоритм для этого автомата, в соответствии с которым автомат за наименьшее число действий нальет в пустой чан n л воды.

§ 15 Рекуррентные соотношения и рекурсивные алгоритмы

Рассмотрим задачу.

Петя и Коля прочитали в одной книжке по математике, что сумма

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \ldots + \frac{1}{N}$$

может стать больше любого числа, если только выбрать N достаточно большим.

Ребята очень удивились этому — ведь каждый раз добавляется все меньшее и меньшее число. И они решили попытаться найти такое N, чтобы указанная сумма была не меньше 20. Для этого Петя и Коля составили следующий алгоритм, который затем каждый из них попытался исполнить на своем калькуляторе:

```
Алгоритм Сумма
цел: S, N;
{
  S := 1;
  N := 1;
  Делать пока (S < 20)
    { N := N + 1;
      S := S + 1/N;
    }                              (* конец цикла *)
  Сообщить N;
}
```

Убедитесь, что этот алгоритм действительно решает поставленную задачу.

Сначала разберемся, что же делается в приведенном алгоритме. Нам требуется найти сумму N чисел. Но доступный нам исполнитель алгоритмов (калькулятор) умеет складывать только два числа. Поэтому нахождение суммы большего числа слагаемых мы заменяем на несколько последовательных сложений двух чисел, постепенно накапливая сумму в переменной S. Если через S_k обозначить сумму первых k слагаемых, то можно записать такое равенство:

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \ldots + \frac{1}{n-1} + \frac{1}{n}.$$

Из него нетрудно вывести другое равенство:

$$S_n = S_{n-1} + \frac{1}{n}.$$

Посмотрите внимательно: полученная формула показывает, что n-е значение переменной S получается из $(n-1)$-го значения прибавлением $\frac{1}{n}$.

Когда существует правило, позволяющее находить n-е значение переменной, используя ее предыдущие значения, говорят, что **переменная задана рекуррентно.** Само правило при этом называют **рекуррентным соотношением.**

Рекуррентные соотношения очень часто применяют для вычисления тех или иных «длинных» сумм. Как мы видим, Петя и Коля тоже воспользовались этим приемом, даже о нем не подозревая.

Раз уж мы начали рассказывать истории про задачи, вот еще одна.

«Древняя легенда гласит, что при сотворении мира в одном из буддийских храмов в г. Ханое была положена бронзовая плита с тремя стержнями и на один из них были нанизаны 64 золотых диска так, как показано на рисунке 2.7. Монахи этого храма должны перенести все диски с одного стержня на другой, соблюдая следующие правила:

1) за один раз можно переносить только один диск;
2) нельзя класть больший диск на меньший;
3) переносить диск можно с любого стержня на любой.

Согласно легенде, как только все диски соберутся на другом стержне, наступит конец света.

Составить алгоритм сразу для случая 64 дисков кажется весьма непростым делом. Даже глядя на 8 дисков, изображенных на рисунке, трудно сказать, что и в какой последовательности делать. Ясно, что первым действием переносится на другой стержень самый маленький диск — ведь ничего другого сделать нельзя. Вторым действием надо на другой стержень перенести второй по размеру диск. А вот третий диск уже и переносить некуда.

Давайте подойдем к решению этой задачи с другой стороны. Пусть на стержне № 1 имеется n дисков. Как решать задачу при $n = 1$, совершенно ясно — переложил этот диск на второй стержень и все. И при $n = 2$ решение тоже понятно: верхний диск переносим на стержень № 2, второй диск переносим на стержень № 3 и, наконец, меньший диск переносим

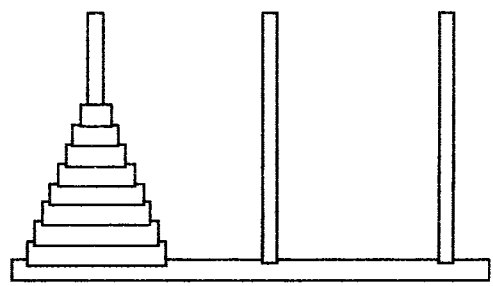


Рис. 2.7 Макет для задачи о ханойской башне

на стержень № 3. Нетрудно понять, как модифицировать последовательность действий, если мы хотим собрать диски на стержне № 2.

А теперь вспомним опыт решения предыдущей задачи: подсчитав сумму из $n - 1$ слагаемых, легко вычислить сумму n слагаемых. Пусть мы умеем (т. е. у нас есть алгоритм!) переносить башню из $n - 1$ дисков со стержня № 1 на любой другой стержень в полном соответствии с предписанными правилами. Берем теперь башню из n дисков. Верхние $n - 1$ дисков переносим на другой стержень. Оставшийся диск перекладываем на свободный стержень и теперь на него, пользуясь тем же алгоритмом, перекладываем $n - 1$ дисков. Задача решена.

Решение, как вы видите, построено на том, что при исполнении алгоритма для n дисков мы обращаемся к тому же алгоритму для $n - 1$ дисков. Ситуация, в которой какой-то алгоритм сам или через другие алгоритмы вызывает себя в качестве вспомогательного, называется **рекурсией**. Сам алгоритм при этом называется **рекурсивным**.

Из определения рекурсивного алгоритма следует, что он всегда должен быть оформлен как вспомогательный алгоритм. В частности, он может быть представлен подпрограммой-функцией.

Запишем рекурсивный алгоритм решения задачи о ханойской башне.

```
Алгоритм Перенос_башни (арг: n, a, b; рез: башня дисков на стержне с указанным номером)
        (* n — количество дисков; a — номер стержня, на котором
           находятся диски до переносов; b — номер стержня,
           на котором должны находиться диски после переносов *)
{
Если n = 1 то
  { Перенести диск со стержня с номером a на стержень с номером b;}
иначе
  { Вызвать Перенос_башни (n − 1, a, c);
    Перенести диск со стержня с номером a на стержень с номером b;
    Вызвать Перенос_башни (n − 1, c, b);
  }
}
```

Тогда алгоритм, решающий исходную задачу, выглядит совсем просто.

```
Алгоритм Ханой
цел: n;
{ Запросить n;
  Вызвать Перенос_башни (n, 1, 2);
}
```

Если на запрос алгоритма ввести число 64, то этот алгоритм совершит требуемое перемещение дисков.

ВОПРОСЫ И ЗАДАНИЯ

1. Что называют рекуррентным соотношением?

2. Что такое рекурсия? Какой алгоритм называется рекурсивным?

3. Что общего между рекуррентным соотношением и рекурсивным алгоритмом и чем они различаются?

4. а) Составьте алгоритм вычисления суммы чисел, обратных квадратам первых ста натуральных чисел.

   б) Составьте алгоритм, согласно которому будет запрошено несколько натуральных чисел и сообщено их среднее арифметическое (количество запрашиваемых чисел тоже запрашивается).

   в) Составьте алгоритм вычисления следующей алгебраической суммы:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \frac{1}{7} - \frac{1}{8} + \ldots + \frac{(-1)^{N-1}}{N}.$$

5. Рассмотрите следующий алгоритм, использующий подпрограмму-функцию Оборот.

```
Алгоритм
сим: A; C;
{ Запросить A;
  C := Оборот (A);
  Сообщить C;
}

Алгоритм Оборот (сим: B): сим
{ Если Len (B) = 1 то { Оборот := B}
  иначе { Оборот := Часть (B, Len (B), 1) + Оборот (Часть (B, 1, Len (B) – 1));}
}
```

   Что сообщит компьютер, исполнивший этот алгоритм применительно к словам: а) марс; б) топор; в) огород; г) шалаш?

6. Рассмотрите следующий алгоритм, использующий подпрограмму-функцию НОД(x, y), созданную вами при выполнении задания 4а из § 14, и подпрограмму-функцию ФОК(x).

```
Алгоритм
цел: x;
{ Запросить x;
  x := ФОК(x);
  Сообщить x;
}

Алгоритм ФОК (цел: x): цел
{ Если x = 1 то { ФОК := 1; }
  иначе { ФОК := ФОК (x – 1)*x / НОД (ФОК (x – 1), x) }
}
```

   а) Что сообщит компьютер, исполнивший этот алгоритм для чисел 1, 4, 8?

   б) Напишите нерекурсивный алгоритм, вычисляющий для натурального числа x тот же результат.

7. Составьте рекурсивный алгоритм, подсчитывающий для запрашиваемого натурального числа n сумму квадратов его цифр.

8. Модифицируйте алгоритм, решающий задачу о ханойской башне, так, чтобы он сообщал последовательность номеров переносимых дисков с указанием, с какого стержня на какой надо осуществлять перенос.

9* Составьте алгоритм, выполнив который компьютер запросит натуральное число n и сообщит все перестановки, которые можно составить из чисел 1, 2, ..., n.

10* Две ханойские башни. Имеется три стержня. На два из них нанизаны диски так, что меньший диск лежит на большем; третий стержень свободен. Все диски имеют разный диаметр. Составьте алгоритм, позволяющий собрать все диски на одном стержне с соблюдением правил задачи о ханойской башне.

§16 От переменной к массиву

Переменная — это простейшая, иногда говорят атомарная, структура данных. Работая с электронной таблицей, вы не раз убеждались, насколько бывает удобно, чтобы информация была представлена в табличной форме. Вы знаете из курса информатики, изучавшегося в 8 и 9 классах, что каждая ячейка таблицы — это фактически отдельная переменная. В любом языке программирования предусмотрена возможность табличного описания обрабатываемой информации, при этом клетки таблицы представлены переменными одного типа. Такие таблицы называются **массивами**. Номер строки и номер столбца в таблице называют **индексами** того элемента, который стоит в клетке на пересечении данной строки и данного столбца.

Обычно таблица имеет несколько строк и несколько столбцов. Массивы, соответствующие таким таблицам, называют **двумерными**.

Если массив представляет таблицу, состоящую из одной строки или одного столбца, то такой массив называют **одномерным** или, по-другому, **линейным**. В его описании будет использоваться только одно множество индексов.

Впрочем, массивы могут быть и трехмерными, и четырехмерными, и еще большей размерности — все зависит от задачи, которую приходится решать.

Итак, **массив** — это набор однотипных данных, снабженных системой из одного или нескольких индексов, каждый из которых принимает последовательные целые значения. Количество используемых индексов называют **размерностью** массива, тип входящих в него элементов — **типом** данного массива.

Чтобы задать массив, надо указать его **имя**, **тип**, **диапазон** изменения каждого индекса. Например, **вещ:** М[1:14, 3:25, 5:70]. Элемент массива также имеет имя: оно состоит из имени массива и значений индексов, определяющих данный элемент. Договоримся значения индексов указывать после имени массива в круглых скобках через запятую.

Каждый элемент массива удобно представлять себе как отдельную переменную, тем более что к нему можно обратиться и присвоить то или иное значение точно так же, как и для «самостоятельной» переменной.

В таблице 2.5 показано, как располагаются элементы в двумерном массиве $A[1:3, 1:4]$.

Таблица 2.5

$A(1, 1)$	$A(1, 2)$	$A(1, 3)$	$A(1, 4)$
$A(2, 1)$	$A(2, 2)$	$A(2, 3)$	$A(2, 4)$
$A(3, 1)$	$A(3, 2)$	$A(3, 3)$	$A(3, 4)$

ВОПРОСЫ И ЗАДАНИЯ

1. Что такое массив? Что называют типом массива? Что такое размерность массива?

2. Как связаны имена элементов массива с именем самого массива?

3. Какие значения переменных A и B сообщит компьютер, выполнив следующий алгоритм для двумерного массива M, изображенного таблицами *а*—*в*?

```
Алгоритм
  цел: A, B, I, J, M[1:5, 1:4];
  { Запросить M[1:5, 1:4];
    A := 0;
    B := 0;
    Делать от I := 1 до 5
      { Делать от J := 1 до 4
        { Если M(I, J) = 1 то { A := A + 1; }
          Если M(I, J) = -1 то { B := B + 1; }
        }                                    (* конец цикла *)
      }                                      (* конец цикла *)
    Если (A > B) то { B := A + B; }
    Если (A < B) то { B := A + B; }
    Сообщить A;
    Сообщить B;
  }
```

а)

0	1	0	−1
1	0	−1	1
2	−1	2	1
−1	0	−1	2
0	1	1	1

б)

−1	0	2	1
1	−1	0	2
0	1	−1	2
2	0	1	−1
−1	0	3	1

в)

1	0	1	2
3	−1	−1	−1
1	−1	1	−1
0	1	2	−1
−1	0	1	2

4. Требуется заполнить массив $A[1:6, 1:6]$ из целых чисел по правилу $a(k, n) = k^n - n^k$. Изобразите схему алгоритма, позволяющего заполнить массив нужным образом.

5. Дан трехмерный числовой массив $A[1:10, 1:10, 1:10]$.
а) Составьте алгоритм поиска максимального элемента в этом массиве.

б) Составьте алгоритм, сообщающий индексы всех наименьших элементов данного массива.

6 Дан трехмерный массив $M[1:100, 1:20, 1:75]$ вещественного типа.
а) Составьте алгоритм вычисления суммы всех положительных элементов массива.
б) Составьте алгоритм, позволяющий выяснить, каких элементов — положительных или отрицательных — в массиве больше и на сколько.

7 Даны 100 точек на плоскости. Будем считать, что они занумерованы числами от 1 до 100, а их координаты размещены в двумерном массиве $M[1:2, 1:100]$, где $M(1, j)$ — абсцисса j-й точки, а $M(2, j)$ — ордината той же точки. Составьте алгоритм, после исполнения которого будут сообщены номера точек, наиболее удаленных друг от друга.

8 Прямоугольная таблица заполнена числами –1, 0 и 1.
а) Требуется изменить на противоположное то ненулевое число, которое находится в клетке, наиболее удаленной от левой верхней угловой клетки таблицы. Составьте соответствующий алгоритм. (Расстояние между клетками — это расстояние между центрами этих клеток.)
б) Требуется изменить на противоположные числа в тех парах ненулевых чисел, для которых расстояние между клетками, в которых стоит эта пара, наибольшее. Составьте соответствующий алгоритм.

9 Дан одномерный массив, заполненный целыми числами. Требуется построить массив, содержащий только четные числа из исходного массива, или сообщить, что все числа в исходном массиве нечетны. Составьте алгоритм, решающий такую задачу.

10 Представьте, что одномерный массив «склеен» в кольцо, т. е. после последнего элемента снова идет первый. Требуется в таком «кольцевом» массиве каждый элемент заменить полусуммой его двух соседей. Составьте алгоритм, решающий эту задачу.

§17 Решение уравнений методом половинного деления

Рассмотрим один из методов приближенного решения уравнений. В курсе математики вы изучаете различные типы уравнений. Для некоторых имеются формулы, по которым можно вычислять значения корней. Однако в реальных задачах гораздо чаще встречаются уравнения, для которых таких формул указать нельзя. Вот одна из таких задач.

На поверхности воды плавает цилиндрический деревянный брусок длины l и радиуса r (рис. 2.8, *а*). Какова высота его надводной части?

Поскольку брусок плавает, сила тяжести уравновешена выталкивающей силой. Сила тяжести равна, как известно, произведению объема на удельный вес. Если обозначить удельный вес через d (для дерева можно принять $d \approx 0{,}8$), то вес бруска будет равен $\pi r^2 l d$. Выталкивающая сила по закону Архимеда равна весу вытесненной жидкости. Чтобы подсчитать объем вытесненной жидкости, рассмотрим рисунок 2.8, *б* и обозначим через x величину угла AOB в радианах. Тогда интересующий нас объем равен разности объ-

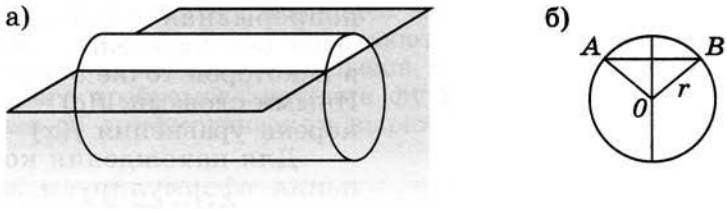


Рис. 2.8

ема всего бруска и той части, которая выступает над водой. Легко сообразить, что объем надводной части равен произведению длины l на площадь сегмента. Площадь сегмента, в свою очередь, равна разности площади сектора с центральным углом AOB и площади $\triangle AOB$. Напомним, что площадь сектора равна $\frac{1}{2}xr^2$, а площадь треугольника $\frac{1}{2}r^2\sin x$. Тем самым площадь сегмента равна $\frac{1}{2}r^2(x - \sin x)$. Поскольку удельный вес воды равен 1, получаем уравнение для определения x:

$$\pi r^2 l - \frac{1}{2}lr^2(x - \sin x) = \pi r^2 ld.$$

Преобразуем его:

$$x - \sin x - 2\pi(1 - d) = 0.$$

Если мы решим это уравнение, то без труда найдем глубину погружения бруска: $r + r\cos\frac{x}{2}$.

Однако никакой формулы, выражающей x через π и d, не существует. Возникает задача научиться находить приближенное значение x для каждого заданного значения d.

Прежде всего отметим, что при $x = 0$ левая часть данного уравнения отрицательна, а при $x = \pi$ положительна, поскольку $d > 0{,}5$. Поэтому если мы представим себе график функции $y = x - \sin x - 2\pi(1 - d)$, то при изменении аргумента от 0 до π этот график переходит из нижней координатной плоскости в верхнюю. Следовательно, в некоторой точке этот график пересечет ось абсцисс. Эта точка и даст нам корень рассматриваемого уравнения.

Любители математики могут даже доказать, что указанная функция возрастающая, поэтому рассматриваемое уравнение имеет ровно один корень.

Обсудим теперь ситуацию в общем виде, а не только применительно к той функции, которая возникла в этой задаче.

Нам дана некоторая функция $f(x)$ и отрезок $[a; b]$, причем на концах этого отрезка функция принимает значения противоположных знаков (рис. 2.9). Если функция непрерывна, т. е. ее график —

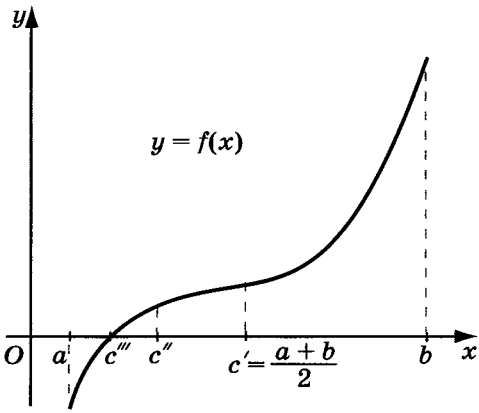


Рис. 2.9 Поиск корня методом деления пополам

непрерывная линия, то график функции пересекает ось абсцисс в некоторой точке c отрезка $[a; b]$. Иными словами, $f(c) = 0$, т. е. c — корень уравнения $f(x) = 0$.

Для нахождения корня уравнения сформулируем *метод половинного деления для приближенного решения уравнения* $f(x) = 0$.

Рассмотрим середину отрезка $[a; b]$, т. е. точку на оси Ox с координатой $(a + b)/2$. В этой точке вычисляем значение функции $f(x)$. Если это значение 0, то корень найден; если нет, то оно имеет тот же знак, что и значение на одном из концов отрезка $[a; b]$. Тогда выбираем новый отрезок так, чтобы один из его концов был точкой $(a + b)/2$, а функция $f(x)$ на его концах снова имела разные знаки. Новый отрезок тоже содержит корень уравнения $f(x) = 0$. Однако этот отрезок в 2 раза короче предыдущего. И самое главное — *с ним можно поступить точно так же*. Опять поделим отрезок пополам и тем самым еще раз уменьшим его вдвое. Таким образом мы можем получить отрезок как угодно малой длины, внутри которого содержится корень уравнения $f(x) = 0$. Например, если исходный отрезок был [3; 4], т. е. имел длину 1, то через десять шагов мы получим отрезок длиной $\frac{1}{2^{10}} = \frac{1}{1024} < 0{,}001$. Это означает, что концы отрезка дают нам приближенное значение корня с точностью, равной длине отрезка: левый конец отрезка — приближенное значение корня с недостатком, правый конец — приближенное значение корня с избытком.

Запишем теперь алгоритм решения уравнения $f(x) = 0$ методом половинного деления.

```
Алгоритм Решение_уравнения
  вещ: a, b, x, y, z, e;
  { Запросить a;                          (* левый конец промежутка *)
    Запросить b;                          (* правый конец промежутка *)
    Запросить e;                                    (* точность *)
    Делать пока b − a > e
    { z := (a + b)/2;
      Если f(z) * f(a) > 0 то { a := z; }
      иначе { b := z; }                             (* конец ветвления *)
    }                                               (* конец цикла *)
    Сообщить z;
  }                                                 (* конец алгоритма *)
```

Надо только сделать следующее замечание. Вместо $f(x)$ в алгоритме должно быть либо записано соответствующее арифметическое выражение, либо организован вызов вспомогательного алгоритма, вычисляющего требуемое значение функции. В нашем случае речь идет именно об арифметическом выражении $x - \sin x - 2\pi(1 - d)$.

ВОПРОСЫ И ЗАДАНИЯ

1. В чем состоит метод половинного деления при нахождении нужного значения переменной?

2. Уравнение $x^3 + 3x - 1 = 0$ на отрезке $[0; 1]$ имеет корень (попытайтесь объяснить почему). Для его вычисления Коля составил следующий алгоритм:

```
Алгоритм
вещ: X;
{ X := 0;
Делать пока (X^3 + 3*X - 1 < 0)
  { X := X + 0.01;
  }                                   (* конец цикла *)
Сообщить "Корень равен", X;
}
```

а) Определите, с какой точностью вычисляется корень по Колиному алгоритму.

б) Петя, списывая у Коли этот алгоритм, по невнимательности переставил местами две строки. Тем не менее и по Петиному алгоритму компьютер нашел корень. Какие две строки оказались переставленными?

в) Запишите алгоритм вычисления всех корней уравнения $x^3 + 3x - 1 = 0$ с точностью 0,01.

§18 Измерение количества информации

Название параграфа, возможно, вызовет у вас недоумение — ведь в § 4 уже говорилось о единицах измерения информационного объема, и вы, без сомнения, помните, что такое бит, байт, килобайт и т. п.

Все это верно. Но нужно понимать, что введение указанных единиц преследовало чисто технологические цели, никак не отражая смысловое содержание информационного объекта. К примеру, если на диск дважды записано одно и то же сообщение, то места на нем занято в 2 раза больше, чем в том случае, когда на диск сообщение записано однократно. Значит, и в байтах мы получим число, в 2 раза большее, чем для одного сообщения. Однако из повторного сообщения мы никакой новой (по смыслу) информации не имеем, так что количество информации остается тем же, что и после получения первого сообщения. И вообще, если в сообщении говорится о том, что уже известно получателю информации, то естественно считать, что количество информации у получателя не изменилось. Впрочем, если сообщение не может быть понято получа-

телем информации (не важно, в силу недостатка знаний у получателя или искажений в ходе передачи информации), то оно тоже не несет для него информацию.

Чтобы разобраться в связи между техническим подходом к измерению количества информации и семантическим (смысловым), рассмотрим игру «Угадайка». Правила игры таковы.

Один игрок задумывает целое число из заранее определенного диапазона, например от 1 до 16. Второй игрок, задавая такие вопросы, на которые первый может отвечать только «да» или «нет», должен выяснить, какое число было задумано. К примеру, можно спросить: «Верно ли, что задуманное число равно 7?» или «Больше ли задуманное число 3?».

Поиграйте с соседом по парте в эту игру и попытайтесь понять, какое наименьшее число вопросов гарантирует вам отгадывание числа.

Теперь порассуждаем. Конечно, число можно выяснить, задавая последовательно вопросы: «Верно ли, что задумано число 1?», «Верно ли, что задумано число 2?», «Верно ли, что задумано число 3?» и т. д. На какой-то из вопросов ответ будет «да». Но не исключено, что это будет шестнадцатый по счету вопрос.

Как же гарантированно получить ответ за меньшее число вопросов? Вспомним метод половинного деления, о котором мы рассказывали в § 17.

Разделим наши числа от 1 до 16 на две равные группы от 1 до 8 и от 9 до 16. Вопросом «Верно ли, что задуманное число больше 8?» мы заставим игрока сказать нам, в каком из этих двух интервалов находится задуманное им число. Вторым вопросом мы найденный нами интервал разделим еще раз пополам: если ответ игрока на первый вопрос был утвердительным, то спросим: «Верно ли, что задуманное число больше 13?» Теперь уже задуманное число находится в интервале, состоящем всего лишь из четырех чисел. Следующий, третий по счету вопрос разделит этот интервал еще раз пополам. Из оставшихся двух чисел узнать задуманное можно, задав всего лишь один вопрос. Итак, четырех вопросов здесь оказалось достаточно, чтобы определить задуманное число.

Нетрудно понять, что если исходный диапазон чисел был бы от 1 до 32, то хватило бы пяти вопросов для гарантированного угадывания задуманного числа. Если исходный диапазон от 1 до 64, то хватило бы шести вопросов. И т. д.

Ситуацию в игре «Угадайка» можно описать так. В начале игры неопределенность в выборе задуманного числа составляла 16 условных единиц — любое число из 16 могло оказаться задуманным. После первого вопроса неопределенность уменьшилась вдвое — задуманным теперь могло оказаться любое из 8 чисел. После второго вопроса неопределенность уменьшилась еще вдвое и т. д. Иными словами, ответ на каждый вопрос давал нам информацию, вдвое уменьшающую неопределенность.

Поэтому в информатике договорились **принять количество информации, уменьшающей вдвое неопределенность исходной ситуации, равным одному биту.**

Теперь ясно, что количество информации, уменьшающей неопределенность в 4 раза, равно двум битам, уменьшающей в 8 раз — трем битам и т. д. Вообще, если полученная информация уменьшает неопределенность в 2^k раз, то количество этой информации равно k. Возможно, что при этом число k окажется не натуральным числом — ведь неопределенность может уменьшиться, например, в 3 раза или в 5 раз, т. е. в некоторое число раз, не являющееся степенью числа 2 с натуральным показателем. Довольно скоро в курсе математики вам будет показано (а кто-то, может быть, уже знает сейчас), что для любого положительного числа n найдется такое число k, для которого $n = 2^k$. Число k в этом случае называют логарифмом числа n по основанию 2 и обозначают $\log_2 n$.

Если полученная информация уменьшает неопределенность в n раз, то количество этой информации равно $\log_2 n$. Иными словами, если необходимо выделить один объект из множества, содержащего n равноправных объектов, мы должны располагать $\log_2 n$ информации об этом объекте.

Почему же и при техническом подходе к измерению объема информации, и при семантическом подходе используются одни и те же единицы — биты, байты, килобайты?.. Попробуем в этом разобраться.

Напомним, что всю информацию, циркулирующую в компьютере и хранящуюся в его памяти, можно представлять себе закодированной в двухсимвольном алфавите, например, в виде последовательностей из 0 и 1. То же самое можно сказать и об информации, передаваемой по каналам связи. Сама схема передачи информации еще раз представлена на рисунке 2.10.

Под **приемником** информации мы здесь понимаем человека или автоматическое устройство, которое просто фиксирует сигналы, поступающие к нему по каналу связи. Он при этом не обязан вникать в смысл передаваемой информации и может просто не знать код, которым закодировано исходное сообщение. Тем самым перед получением очередного сигнала приемник информации находится в состоянии неопределенности — придет сигнал, соответствующий символу 0, или сигнал, соответствующий символу 1. Когда

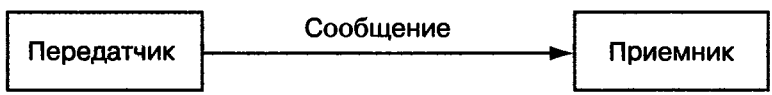


Рис. 2.10

сигнал приходит, неопределенность уменьшается вдвое (или полностью исчезает), т. е. приемник информации получает, как мы выше договорились, 1 бит. Теперь уже ясно, что, получив последовательность из n символов двоичного алфавита, приемник информации получил n бит. Вот мы и получили тот же вывод, который ранее предлагался вам в качестве определения количества информации при чисто техническом подходе.

А можно ли, наоборот, исходя из кодирования, прийти к смысловому пониманию количества информации? Чтобы ответить на этот вопрос, проведем следующий эксперимент. Рассмотрим числа от 0 до $2^k - 1$. Чтобы эксперимент был не слишком утомительным, ограничимся числами от 0 до 15 ($15 = 2^4 - 1$). Пусть задумано какое-нибудь число из этого интервала (к примеру, число 5). А теперь, задавая вопросы оптимальным образом, попытаемся угадать это число. Договоримся только, что все вопросы будут иметь форму «Верно ли, что задуманное число больше ...?». При этом, получив ответ «да», будем писать 1, а получив ответ «нет», писать 0. Вот эти вопросы и ответы для нашего числа:

1. Верно ли, что задуманное число больше 7?	Нет	0
2. Верно ли, что задуманное число больше 3?	Да	01
3. Верно ли, что задуманное число больше 5?	Нет	010
4. Верно ли, что задуманное число больше 4?	Да	0101

После этих четырех вопросов число угадано — 5. Но самое главное — в правом столбце в тот момент, когда число угадано, появляется его запись в двоичной системе счисления! Иными словами, последовательность ответов на вопросы — это двоичное кодирование информации. Сами же вопросы (точнее, метод их генерирования; в нашем случае — оговоренная заранее форма вопроса) определяют способ кодирования. Верно и то, что, если способ кодирования задан, следующий вопрос можно задавать независимо от ответа на предыдущий. К примеру, в данной игре можно сформулировать вопросы так:

1. Верно ли, что в двоичной записи задуманного числа первая цифра 1?
2. Верно ли, что в двоичной записи задуманного числа вторая цифра 1?
3. Верно ли, что в двоичной записи задуманного числа третья цифра 1?
4. Верно ли, что в двоичной записи задуманного числа четвертая цифра 1?

Фактически именно эти вопросы и задавались в игре, но, поскольку о двоичном кодировании договоренности не было, пришлось в неявном виде это кодирование проводить.

В «Угадайку» можно играть не только с числами. Можно расшифровывать названия загаданных животных или фамилии знаменитых писателей. И вообще, задавая вопросы, на которые даются лишь ответы «да» и «нет», можно узнать любые сведения,

т. е. *смысловое содержание информации*. Если при этом на каждое «да» мы будем писать 1, а на каждое «нет» — 0, то мы получим двоичный код «добываемой» информации.

Тот факт, что с помощью вопросов, допускающих лишь ответы «да» и «нет», можно получить любую интересующую нас информацию, люди знали, по-видимому, очень давно. Однако только в середине 20-го столетия было осознано, что такой способ получения информации эквивалентен двоичному кодированию и может быть использован для вычисления количества информации.

Последовательность вопросов задает кодирование, и если она фиксирована, то по двоичному коду можно однозначно восстановить все ответы. Заметим, что, задавая лишь конечное число вопросов, мы способны выявить не всю информацию об объекте, а только какую-то ее часть. К примеру, играя в «Угадайку», мы пытались определить задуманное животное и уже узнали, что это кит. Далее мы можем выяснять, к какому виду принадлежит этот кит, каков его возраст и т. д. Каждый объект описывается бесконечным числом свойств, мы же всегда ограничиваемся конечным набором тех из них, которые нам представляются существенными. Иными словами, процесс кодирования, сопровождающий получение информации, мы всегда обрываем на конечном шаге, удовлетворяясь подходящим для нас приближением.

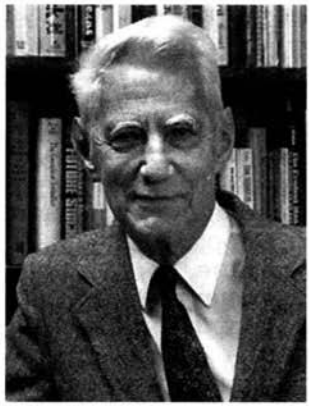
Здесь уместна такая аналогия. Вам хорошо известно, что каждое иррациональное число записывается, т. е. кодируется, в десятисимвольном алфавите бесконечной непериодической дробью. Человек не может воспринимать и обрабатывать такие дроби. Поэтому он в своей практической деятельности ограничивается подходящими приближениями, обрывая эту запись на том или ином разряде. Можно сказать, что такие приближения моделируют данное иррациональное число. Использование приближений основано на уверенности (и математически доказанном факте), что такая замена числа его приближением не приведет к существенным искажениям.

В теории информации доказана аналогичная теорема об универсальности двоичного кодирования:

всегда существует двоичное кодирование, как угодно полно отражающее информацию об объекте.

Первым понятие количества информации ввел американский ученый Клод Шеннон. Занимаясь проблемой передачи информации по каналам связи, он заложил основы теории информации, которая теперь является одним из краеугольных камней информатики. Правда, подход к измерению количества информации в сообщении у него был иной, нежели тот, который мы изложили в этом параграфе. О его воззрениях на проблему измерения количества информации мы расскажем в главе 3. Там же будет приведена более общая формула для количества информации.

Как видно из нашего обсуждения, во всех рассмотренных нами способах определения количества информации важную роль игра-



Клод Шеннон

ет способ кодирования этой информации. А нельзя ли так определить количество информации, чтобы избежать этой зависимости от кодирования? Оказывается, можно. Идею, как это можно сделать, предложил выдающийся математик академик А. Н. Колмогоров. Вот в чем состоит его идея.

Представим, что некое сообщение (текстовое или графическое) создано в соответствующем редакторе и сохранено на диске. Оно имеет вполне определенный информационный объем. Применим какой-нибудь архиватор. Информационный объем сообщения уменьшится, причем для разных архиваторов результат вполне может оказаться различным. Каков же на самом деле информационный объем сообщения?

Заметим, что применение архиватора можно рассматривать как простое изменение способа кодирования информации. На первый взгляд за количество информации можно взять минимальный информационный объем, который получают, используя разные способы кодирования. Но тогда, во-первых, можно считать, что все сообщение закодировано каким-нибудь одним символом (и тогда информационный объем просто всегда равен 1), во-вторых, чтобы получатель информации понял, что ему сообщают, придется дополнительно посылать расшифровку этого символа. А тогда мы только увеличиваем информационный объем — посылаем символ и к нему его расшифровку в виде того же сообщения.

Разумеется, можно послать не расшифровку, а алгоритм декодирования или, что еще лучше, встроить алгоритм декодирования в само сообщение (наподобие самораспаковывающихся архивов). Поэтому естественно при подсчете количества информации в сообщении учитывать и информацию, необходимую для кодирования и декодирования. То, что мы не делаем этого при других подходах к измерению количества информации, объясняется тем, что способ кодирования считается известным как источнику информации, так и его получателю. Тогда сообщение о способе кодирования уже не несет для получателя информации и на количестве передаваемой информации не сказывается.

Таким образом, определение количества информации, по Колмогорову, отличается от других определений тем, что учитывает процессы кодирования и декодирования информации.

Приведем пример. Пусть нам требуется передать в сообщении все натуральные числа от 262 144 до 823 543. Если составить сообщение из всех этих чисел, не забыв пробелы между ними, то в таком сообщении будет 3 929 799 символов, т. е. информационный объем сообщения составит более 3837 Кбайт.

Но вместо этого можно в виде сообщения передать следующий алгоритм:

```
Алгоритм Пересылка
цел: n;
{ Делать от n := 262 144 до 823 543
  { Сообщить n;
  }                                    (* конец цикла *)
}                                      (* конец алгоритма *)
```

В записи этого алгоритма использовано меньше 100 символов, т. е. его информационный объем меньше 100 байт. Тем не менее информация этим сообщением будет передана та же самая.

Отметим, что это еще не самая короткая запись алгоритма; можно, например, исключить комментарии. Тем не менее ясно, что теоретически существует самое короткое самодекодирующееся сообщение, содержащее данную информацию. Вот его информационный объем и берется за количество информации в данном сообщении.

На практике пользоваться определением количества информации, по Колмогорову, весьма затруднительно, в частности потому, что нет способов нахождения оптимального алгоритма кодирования для любого наперед заданного сообщения. Поэтому, как правило, используются другие подходы. Однако для теории информации это определение играет важную роль.

ВОПРОСЫ И ЗАДАНИЯ

1. Как связано уменьшение неопределенности с количеством полученной информации?

2. Какое количество информации мы должны получить, чтобы угадать одно из 64 первых натуральных чисел? одно из 128 первых натуральных чисел? одно из 100 первых натуральных чисел? Сколько вопросов в игре «Угадайка» придется задать, чтобы гарантированно получить ответ в каждом из этих трех случаев?

3. а) Задумывается нечетное число от 1 до 63. Сколько вопросов в игре «Угадайка» придется задать, чтобы гарантированно угадать задуманное число?

   б) Известно, что задуманное число меньше 1000 и является квадратом некоторого целого числа. Сколько вопросов в игре «Угадайка» придется задать, чтобы гарантированно угадать задуманное число? Сколько бит информации вы получили дополнительно, располагая сведениями, что задуманное число является не просто целым неотрицательным числом, а квадратом некоторого целого числа?

4. Объясните, почему в игре «Угадайка» с первыми 16 натуральными числами нельзя гарантировать угадывание числа за 3 вопроса.

5* Два игрока играют в «Угадайку» (для химиков): один загадывает название химического элемента, а другой, задавая вопросы о его свойствах, должен определить загаданный элемент. Какое наименьшее число вопросов требуется задать, чтобы гарантированно угадать элемент?

6 а) Заметим, что $262\,144 = 2^{18}$, а $823\,543 = 7^7$. Используя эту информацию, составьте еще более короткий, нежели в объяснительном тексте, алгоритм, позволяющий передать сообщение, состоящее из всех чисел в диапазоне от 262 144 до 823 543.

б)* При выполнении задания *а* вам удалось сократить информационный объем передаваемого сообщения за счет использования дополнительной информации о содержании сообщения. Обдумайте это наблюдение и попытайтесь привести другие подобные примеры. Какие выводы можно сделать из этого наблюдения?

ИТОГИ ГЛАВЫ 2

Повторим кратко то основное, что вы узнали, изучив главу 2.

Фактографическими называют модели, в которых информация о некой совокупности объектов хранится в виде записей, сопоставляющих каждому объекту его свойства (признаки, атрибуты). Фактографические модели обычно реализуются в виде баз данных. Для работы с базой данных используется программная оболочка — система управления базой данных (сокращенно СУБД). Пользователю базой данных нередко предоставляется ограниченный доступ к функциям, реализуемым СУБД, а именно: обычно разрешается осуществлять лишь поиск в базе данных без права размещения в ней собственной информации и изменения уже имеющейся. База данных с такими ограниченными возможностями называется информационно-поисковой системой (ИПС). По способу организации связей между атрибутами различают три вида баз данных: иерархические, реляционные и сетевые.

Моделью деятельности человека или работы автоматического устройства нередко выступает алгоритм. При разработке алгоритмов для решения сложных задач полезно использовать вспомогательные алгоритмы. Такой алгоритм имеет заголовок, в котором указывается имя вспомогательного алгоритма, а также список имен переменных, служащих для передачи информации, подлежащей обработке вспомогательным алгоритмом, и результатов его работы. Эти переменные называются формальными параметрами. Для обращения к вспомогательному алгоритму используется оператор **Вызвать**, в котором указываются значения переменных, передаваемые во вспомогательный алгоритм. Вспомогательные алгоритмы, записанные на языке программирования, называются подпрограммами.

Вспомогательные алгоритмы часто возникают, когда для решения задачи применяется метод пошаговой детализации. Он состоит в том, что исходная задача разбивается на несколько задач, решение которых либо уже известно, либо представляется, что решить их будет легче, чем исходную задачу.

Последовательность чисел называется заданной рекуррентно, если, начиная с некоторого номера, каждый ее член определен как значение функции от предшествующих членов.

Рекурсивным называется алгоритм, который обращается сам к себе как к вспомогательному алгоритму (непосредственно или через другие вспомогательные алгоритмы).

Для обработки большого числа данных одинаковой природы используются массивы. Массив — это набор однотипных переменных, снабженных системой из одного или нескольких индексов, каждый из которых принимает последовательные целые значения. Количество используемых индексов называют размерностью массива, тип входящих в него элементов — типом данного массива. Чтобы задать массив, надо указать его имя, тип, диапазон изменения каждого индекса. Элемент массива также имеет имя, состоящее из имени массива и значений индексов, определяющих данный элемент. Каждый элемент массива представляет собой отдельную переменную: к нему можно обратиться и присвоить значение точно так же, как и для «самостоятельной» переменной.

Одним из эффективных методов решения уравнений является так называемый метод деления пополам. Если функция $f(x)$ непрерывна (т. е. ее графиком является непрерывная линия) и на концах некоторого отрезка принимает значения разных знаков, то, во-первых, на этом отрезке обязательно есть корень уравнения $f(x) = 0$ и, во-вторых, методом деления пополам можно вычислить корень этого уравнения с любой наперед заданной точностью.

В информатике принято количество информации, уменьшающей вдвое неопределенность заданной ситуации, считать равным одному биту.

ПРОВЕРЬ СЕБЯ

Часть 1. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и номер правильного ответа. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

А1. Формальные параметры вспомогательного алгоритма — это:

1) переменные, используемые во вспомогательном алгоритме;
2) аргументы вспомогательного алгоритма;
3) переменные, указанные в заголовке вспомогательного алгоритма;
4) количество переменных в заголовке вспомогательного алгоритма.

А2. Массив — это:

1) последовательность переменных;
2) набор однотипных переменных, снабженных системой целочисленных индексов;
3) расположение элементов в виде одной или нескольких строк;
4) в пунктах 1—3 нет правильной формулировки.

А3. База данных, в которой информация представлена в виде записей в таблицах, является:

1) иерархической; 2) реляционной; 3) сетевой;

4) любой из указанных в пунктах 1—3 видов.

А4. В таблице дан фрагмент однотабличной базы данных «Озера» (в первой строке указаны имена атрибутов).

Название	Глубина, м	Площадь, кв. км	Континент	Высота над уровнем моря, м
Аральское	68	33 640	Азия	53
Байкал	1741	31 500	Азия	455
Гурон	228	59 700	Америка	177
Маракайбо	250	13 512	Америка	0
Виктория	80	69 485	Африка	1134
Чад	4	11 000	Африка	240
Ладожское	225	17 700	Европа	5
Севан	99	1400	Европа	1914

Количество записей, удовлетворяющих запросу
«Площадь > 11 000 *и* Высота < 200 *или* Глубина < 250»,
равно:

1) 1; 2) 3; 3) 5; 4) другому, чем в пунктах 1—3, числу.

А5. В результате эксперимента получена информация, которая уменьшила неопределенность в 8 раз. Количество этой информации равно:

1) 3 бита; 2) 1 байт; 3) 3 байта; 4) другому числу байтов.

А6. Размерность массива — это:

1) количество индексов, используемых для описания массива;

2) наибольшее значение какого-либо индекса массива;

3) количество элементов в массиве;

4) в пунктах 1—3 нет правильной формулировки.

А7. Дан алгоритм:

```
Алгоритм
вещ: X, Y, M[1:10]; цел: K;
{ X := (M(1) + M(10) + ABS(M(1) – M(10)))/2;
  Y := (M(1) + M(10) – ABS(M(1) – M(10)))/2;
  Делать от K := 1 до 10
```

```
  { Если (Y < M(K) < X) то
    { Сообщить M(K);
    }                                   (* конец ветвления *)
  }                                     (* конец цикла *)
}
```

Этот алгоритм был исполнен для массива M, описанного следующей таблицей:

−1,2	−0,5	3,6	−1,2	1	2,1	1,9	0	−1,5	3

После исполнения сообщено:

1) 1 число; **2)** 3 числа; **3)** 5 чисел; **4)** 6 чисел.

А8. Дан алгоритм:

```
Алгоритм
вещ: X, Y, M[1:10]; цел: K;
{ X := M(1);
  Y := M(1);
  Делать от K := 2 до 10
  { Если (M(K) < X) то
    { X := M(K);
    }                                   (* конец ветвления *)
    Если (M(K) > Y) то
    { Y := M(K);
    }                                   (* конец ветвления *)
  }                                     (* конец цикла *)
  Сообщить Y − X;
}
```

Этот алгоритм был исполнен для массива M, описанного следующей таблицей:

1,2	−0,5	3,6	0,8	−1	2,1	1,9	0	−1,5	3

После исполнения сообщено число:

1) 2,1; **2)** 5,1; **3)** −1,5;

4) другое, чем указанное в пунктах 1—3.

А9. Некто задумал натуральное число, не превосходящее 1000, а затем сообщил: «Задуманное мною число является кубом целого числа». Количество бит информации, содержащейся в этом сообщении, равно:

1) $\log_2 1000$; **2)** 2; **3)** $\log_2 10$; **4)** другому числу.

А10. Дан алгоритм:

```
Алгоритм
цел: K, M, N; вещ: L;
{ Запросить K;
  L := 0;
```

```
  Делать от M := 1 до K
  { Делать от N := 1 до K
    { L := (L + (M + N) / (M*N))*(L − (M/N + N/M));
    }                                          (* конец цикла *)
  }                                            (* конец цикла *)
  Сообщить L;
}
```

На некотором компьютере при $K = 10\,000$ этот алгоритм исполняется в течение 0,5 с. На том же компьютере этот алгоритм при $K = 100\,000$ будет исполняться:

1) ≈ 5 с; 2) ≈ 25 с; 3) ≈ 50 с; 4) ≈ 2 мин.

А11. Дан алгоритм, в работе которого используются две подпрограммы-функции:

```
Алгоритм
цел: K, M, N;
{ Запросить M;
  Запросить N;
  K := A (M, N);
  Сообщить M*N/K;
}

Подпрограмма-функция A (цел: M, N): цел
{ Если M < 3 то { знач := M; }
  иначе { знач := Int (M/3) + B (M − 1, N); }
}

Подпрограмма-функция B (цел: M, N): цел
{ Если N < 3 то { знач := N; }
  иначе { знач := 3 * N − A (M, N − 2); }
}
```

Результат работы алгоритма при $M = N = 10$ обозначим через C, а результат работы алгоритма при $M = N = 9$ обозначим через D. Тогда

1) $C < D$; 2) $C = D$; 3) $C > D$;

4) при одном из указанных наборов значений M и N алгоритм не заканчивает работу.

Часть 2. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и ответ в виде последовательности символов. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

В1. В квадратной таблице размером 8×8 одна клетка закрашена черным цветом, остальные клетки белые. Требуется установить, где она располагается (т. е. узнать номер ряда и номер столбца, на пересечении которых находится клетка). Получена информация, что клетка стоит

на диагонали, начинающейся в верхней левой клетке. Количество полученной информации равно _____ бит.

B2. В базе данных отдела кадров небольшой фирмы имеется 50 записей о рабочих трех категорий: токарях, слесарях и разнорабочих. Среди атрибутов в этих записях, кроме специальности, фигурирует атрибут «пол» со значением «м» или «ж». Количество записей N, удовлетворяющих различным запросам, приведено в таблице:

Запрос	N
пол = м *и* специальность ≠ разнорабочий	28
специальность ≠ слесарь	35
пол = ж *и* специальность ≠ токарь	12

Количество записей, удовлетворяющих запросу «пол = м *и* специальность = токарь *или* пол = ж *и* специальность = разнорабочий», равно _____.

B3. Дан алгоритм и подпрограмма-функция:

```
Алгоритм
цел: K, M;
{ Запросить M;
  K := ФОКУС (M);
  Сообщить M/K;
}

Подпрограмма-функция ФОКУС (цел: M): цел
{ Если M < 10 то { знач := M; }
  иначе { знач := (M mod 10) * ФОКУС (INT(M/10); }
}
```

Если на запрос алгоритма будет введено число 128, то по окончании работы алгоритма будет сообщено число _____.

3 ГЛАВА
Моделирование процессов живой и неживой природы

В этой главе речь пойдет о моделировании процессов, изучаемых различными научными дисциплинами. А поскольку речь идет о моделировании реальных процессов, то главное внимание будет уделено проблеме адекватности построенных моделей.

§ 19 Моделирование физических процессов

Современная физика — это необъятная область человеческого знания. Начнем с раздела, научный стаж которого в физике насчитывает более пяти веков. Этот раздел — механика.

В школьном курсе физики подробно обсуждаются два вида механического движения — равномерное и равноускоренное. Если тело движется равномерно и прямолинейно, то это значит, что никакие силы на него не действуют. Равноускоренное движение возникает в том случае, когда сила, вызывающая движение, постоянна.

Обычным примером движения под воздействием постоянной силы является движение твердого тела в поле земного тяготения. Действительно, если тело находится на достаточно близком расстоянии от Земли, то можно считать, что тело данной массы притягивается с постоянной силой. Потому модель равноускоренного движения рассматривается на уроках физики, когда говорят о движении тела под углом к горизонту. Математические формулы, описывающие такую модель, известны каждому.

Выйдем на улицу и посмотрим, так ли уж хорошо эта модель согласуется с различными природными явлениями. Например, действие силы земного тяготения заставляет падать на землю капли дождя. Каждый из вас попадал под теплый веселый летний дождь или под осенний моросящий. Какую скорость согласно данной модели должна иметь около поверхности Земли капля, сорвавшаяся с высоты 8 км?

На уроках физики вы получили формулу для скорости тела при его движении в поле силы тяжести, если начальная скорость была нулевая:

$$v = \sqrt{2gh},$$

где h — высота, с которой тело начинает движение к поверхности Земли. Поскольку нам не нужны точные расчеты, будем считать, что $g \approx 10$ м/с^2. В этом случае скорость $v \approx 400$ м/с. Но капля, летящая с такой скоростью, подобна пуле, удар которой пробил бы насквозь оконное стекло.

Этого, однако, не происходит. Значит, модель, которой мы воспользовались, не вполне адекватна, в ней не учтены какие-то существенные факторы.

Один из таких факторов — сопротивление воздуха. При малой скорости движения этим сопротивлением можно пренебречь — вряд ли мы замечаем сопротивление воздуха, если перемещаемся прогулочным шагом. Но стоит побежать — и сопротивление станет сразу весьма ощутимым. Оказывается, что сила сопротивления пропорциональна квадрату скорости движущегося тела. Коэффициент пропорциональности зависит от геометрической конфигурации движущегося тела и плотности среды — каждый знает, что в воде двигаться намного труднее, чем по суше.

Итак, речь пойдет о движении тела в поле силы земного тяготения в среде с сопротивлением. Модель этого процесса в удобном для компьютерного исследования виде мы и будем строить. А самой задаче (для дальнейших ссылок) дадим имя: **«Движение тела в среде с сопротивлением»**.

Чтобы построить нужную модель, мы должны определить не только, что существенно для конструируемой модели, но и те теоретические положения физики, которые позволят нам без проведения натурных экспериментов определить зависимости между параметрами строящейся информационной модели. Еще раз обратим внимание, что всякая модель должна быть согласованной не только с существующей теорией, но и с практикой. Будем считать, что выполнены законы механики, сформулированные Ньютоном. Прежде всего речь идет о втором законе, связывающем движущую силу и ускорение для тела постоянной массы:

$$\vec{F} = m\vec{a}.$$

Из ранее сказанного ясно, что мы имеем дело с математической моделью, в которой участвуют три параметра: сила, масса и ускорение. Отметим, что сила и ускорение — векторные величины, а масса — величина скалярная.

Что касается условия «сопротивление воздуха пропорционально квадрату скорости», будем считать это экспериментально установленным фактом и тоже включим его в список исходных предположений. Можно считать, что сила сопротивления противоположна скорости движущегося тела. Значение коэффициента пропорциональности устанавливается экспериментальным путем. Это значение нам понадобится лишь тогда, когда будут проводиться расчеты, поэтому пока мы просто должны учесть, что одним из параметров, описывающих сопротивление среды, является коэффициент пропорциональности k.

Для простоты предположим невероятное: пока тело движется, совсем не дует ветер. Это предположение обеспечивает нам возможность рассматривать движение тела в одной плоскости.

Чтобы модель не получилась слишком сложной, сделаем еще и «геометрическое» предположение: будем считать, что поверхность Земли плоская. Будем также считать постоянным ускорение свободного падения g (хотя по мере удаления от поверхности Земли сила притяжения убывает согласно закону всемирного тяготения, сформулированному Ньютоном, но мы уже договорились не рассматривать движение на больших высотах).

Важный принцип ньютоновской механики состоит в том, что всякое движение в вертикальной плоскости можно рассматривать как сумму двух независимых «одномерных» движений — горизонтального и вертикального. Выберем систему координат так, чтобы в момент начала движения тело располагалось на оси Oy и вектор его скорости был направлен в первый квадрант (рис. 3.1).

Пусть теперь через некоторое время t тело оказалось в точке с координатами x и y и имеет в это время скорость $\vec{v}$. Результирующая сила $\vec{F}$, определяющая движение тела, равна векторной сумме сил сопротивления и тяжести. Сила сопротивления $\vec{F}_{\text{сопр}}$ согласно сделанному предположению равна $-km|\vec{v}|\vec{v}$, где через $|\vec{v}|$ обозначена длина вектора $\vec{v}$. Чтобы рассматривать движение тела как сумму двух движений, мы должны рассмотреть проекции сил и скоростей на оси. Тогда $F_x = -km|\vec{v}|v_x$, а $F_y = -mg - km|\vec{v}|v_y$, где индексы x и y указывают, на какую ось осуществляется проецирование. Второй закон Ньютона дает нам систему соотношений:

$$\begin{cases} ma_x = -km|\vec{v}|v_x, \\ ma_y = -mg - km|\vec{v}|v_y. \end{cases}$$

Фактически мы уже построили информационную модель процесса. Перепишем ее в более удобном виде, выразив длину вектора $\vec{v}$ через его координаты:

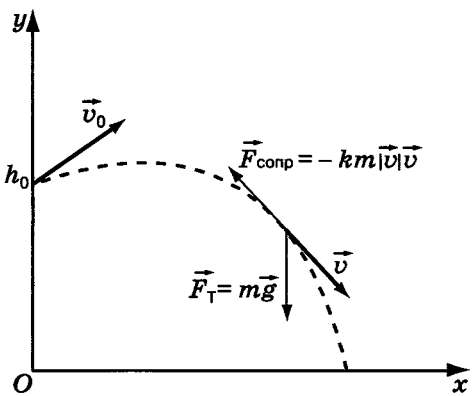


Рис. 3.1

$$\begin{cases} a_x = -k\sqrt{(v_x)^2 + (v_y)^2}\,v_x, \\ a_y = -g - k\sqrt{(v_x)^2 + (v_y)^2}\,v_y. \end{cases}$$

Высота $h(t)$, на которой находится тело в момент времени t, совпадает с ординатой y; так что фактически нам надо научиться выражать изменение y со временем.

Чтобы процесс движения был определен, надо задать, на какой высоте в начальный момент времени находилось тело и какова

была его скорость в этот момент. Поскольку скорость представляет собой вектор, она задается двумя своими координатами. Правда, вместо координат можно задать величину скорости и угол, который вектор скорости составляет с горизонтальной осью (т. е. осью абсцисс). Построение модели данного процесса закончено. Полное описание модели вы сделаете сами, выполнив задание 2.

ВОПРОСЫ И ЗАДАНИЯ

1. Почему возникает необходимость учитывать сопротивление среды?

2. Оформите описание модели движения в среде с сопротивлением, выделив предположения, при которых строится данная модель. Определите на их основе существенные факторы и параметры, описывающие эти факторы. Запишите связь между параметрами. Какого вида оказалась построенная информационная модель?

3. Разведчикам удалось добыть артиллерийские таблицы, составленные на основе испытаний противником новой пушки.

Угол наклона ствола, градус	10	20	30	40	50	60	70	80
Дальность полета снаряда, м	9282	10 889	11 222	10 770	9683	8026	5830	3121

По этим данным нужно найти начальную скорость снаряда. На основе модели движения в среде с сопротивлением предложите способ получения этой информации. Коэффициент сопротивления среды k известен и равен 0,0002 $м^{-1}$.

4.* При движении тела в поле земного притяжения без сопротивления работа силы притяжения переходит в кинетическую энергию движущегося тела.

а) В какой еще вид энергии превращается работа силы сопротивления при движении в среде с сопротивлением?

б) Постройте математическую модель, позволяющую вычислить повышение температуры тела при его движении в среде с сопротивлением. (Потерями энергии за счет теплопередачи в окружающую среду пренебречь.)

5. Что такое дискретизация непрерывного процесса? (С о в е т. Если вы затрудняетесь дать ответ на вопрос, перечитайте § 4.)

§ 20 Компьютерное исследование модели движения в среде с сопротивлением

Как же теперь применить компьютер для исследования процесса с помощью модели, полученной в § 19? Для этого надо выбрать какую-либо компьютерную технологию и реализовать модель ее средствами.

Выбор компьютерной технологии неоднозначен. Выполняя задание 2 из § 19, вы установили, что нами построена математическая

модель. Поэтому и компьютерная технология должна быть ориентирована на численные расчеты. Вам известны две такие технологии: одна из них опирается на применение электронной таблицы, другая — на составление вычислительного алгоритма. Но сначала обсудим, как приспособить нашу модель для применения этих технологий. Дело в том, что численные методы расчетов на компьютере (как и алгоритмы) обладают свойством дискретности. Поэтому построенную в § 19 непрерывную модель нужно заменить дискретной моделью, т. е., как говорят, провести ее **дискретизацию**. Это означает, что процесс, непрерывно развертывающийся во времени, разбивается на отдельные стадии, каждая следующая стадия отстоит от предыдущей на небольшой промежуток времени. Этот малый промежуток обычно обозначают Δt. Естественно предполагать, что в течение этого промежутка времени состояние объектов, участвующих в процессе, остается постоянным, а затем скачком меняется на новое. В нашем случае состояние объекта, т. е. движущегося тела, описывается:

1) координатами $x(t)$ и $y(t)$ точки в момент времени t;
2) вектором скорости $\vec{v}(t)$ в момент времени t;
3) вектором ускорения $\vec{a}(t)$ в момент времени t.

В следующий момент $t + \Delta t$ состояние движущегося тела описывается значениями $x(t + \Delta t)$ и $y(t + \Delta t)$, $\vec{v}(t + \Delta t)$ и $\vec{a}(t + \Delta t)$. Запишем теперь соотношения, показывающие, как связаны изменения названных величин по истечении интервала времени Δt:

$$\frac{x(t+\Delta t)-x(t)}{\Delta t} = v_x(t); \qquad \frac{y(t+\Delta t)-y(t)}{\Delta t} = v_y(t);$$

$$\frac{v_x(t+\Delta t)-v_x(t)}{\Delta t} = a_x = -k\sqrt{v_x(t)^2+v_y(t)^2}\,v_x(t);$$

$$\frac{v_y(t+\Delta t)-v_y(t)}{\Delta t} = a_y = -(g+k\sqrt{v_x(t)^2+v_y(t)^2}\,v_x(t)).$$

Преобразуем эти равенства к следующему виду:

$$x(t + \Delta t) = x(t) + v_x(t) \cdot \Delta t;$$
$$y(t + \Delta t) = y(t) + v_y(t) \cdot \Delta t;$$

$$v_x(t+\Delta t) = v_x(t) - k\sqrt{v_x(t)^2+v_y(t)^2}\,v_x(t)\cdot\Delta t;$$

$$v_y(t+\Delta t) = v_y(t) - (g+k\sqrt{v_x(t)^2+v_y(t)^2}\,v_y(t))\cdot\Delta t.$$

Теперь мы по значениям переменных $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$ в момент времени t можем вычислить их значения в момент времени $t + \Delta t$. Можно сказать, что эти формулы описывают процесс не только *декларативно*, но и *процедурно*, показывая, *как* осуществить нахождение искомых величин. Мы еще усилим акцент на процедурное представление нашей модели, записав алгоритм расчета

траектории движения тела. Но сначала напомним, что служит исходными данными и что — результатами.

Исходные данные:

- высота h, на которой по отношению к горизонту располагается тело в начальный момент времени, м;
- угол A, под которым по отношению к линии горизонта начинает движение тело, рад;
- величина скорости v, с которой тело начинает движение, м/с;
- коэффициент сопротивления среды k, $м^{-1}$;
- величина ускорения свободного падения g, $м/с^2$;
- интервал Δt, на величину которого будет осуществляться переход от одного момента времени к следующему, с.

Результаты:

- координаты положения тела в моменты времени $n\Delta t$, где n — натуральное число, изменяющееся от 1 до такого значения N, при котором ордината тела становится отрицательной.

Запишем теперь алгоритм, реализующий вычисление результатов по исходным данным.

```
Алгоритм Модель_движения
вещ: h, A, v, k, g, Δt, x, y, vx, vy, t, R
{ Запросить h;                          (* начальная высота *)
  Запросить A;          (* начальный угол наклона к горизонту *)
  Запросить v;                          (* начальная скорость *)
  Запросить k;                  (* коэффициент сопротивления *)
  Запросить Δt;                 (* интервал изменения времени *)
  g := 9.8;
  x := 0;                                  (* x — абсцисса тела *)
  y := h;                                  (* y — ордината тела *)
  t := 0;                                  (* счетчик времени *)
  vx := v*COS(A);           (* vx — абсцисса вектора скорости *)
  vy := v*SIN(A);           (* vy — ордината вектора скорости *)
  Делать пока vy > 0
    { x := x + vx*Δt;
      y := y + vy*Δt;
      R := 1 - k*√(vx*vx+vy*vy)*Δt;
      vx := vx*R;
      vy := vy*R - g*Δt;
      t := t + Δt;
  Сообщить "В момент времени", t, "абсцисса", x, "ордината", y;
    }                                      (* конец цикла *)
}                                          (* конец алгоритма *)
```

Другая компьютерная технология, позволяющая вести расчеты, — это электронная таблица. В таблице 3.1 приведем вариант заполнения электронной таблицы для моделирования движения в среде с сопротивлением; записи в угловых скобках показывают, где располагаются начальные данные.

	A	B	C
1	Время	Высота	Дальность
2	0	<h>	0
3	A2+G2	B2+D2*G2	C2+E2*G2
4	A3+G2	B3+D3*G2	C3+E3*G2
5	A4+G2	B4+D4*G2	C4+E4*G2
6	A5+G2	B5+D5*G2	C5+E5*G2
	...	...	...
200	A199+G2	B199+D199*G2	C199+E199*G2
	...	...	

В этом параграфе мы перестраивали математическую модель, созданную в предыдущем параграфе, в модель, к которой применимы компьютерные технологии. Информационная модель, реализованная посредством той или иной компьютерной технологии, называется **компьютерной моделью**. Мы располагаем двумя компьютерными моделями данного процесса.

Компьютерный эксперимент с одной из построенных компьютерных моделей вы будете проводить на практикуме в соответствии с его описанием в лабораторной работе 6.

ВОПРОСЫ И ЗАДАНИЯ

1. Какую модель называют компьютерной?
2. Составьте алгоритм для вычисления координат тела при движении в среде с сопротивлением.
3. В задании 3 из § 19, вы вычисляли начальную скорость по данным таблицы.
   а) Преобразуйте приведенный в объяснительном тексте алгоритм «Модель_движения» в подпрограмму-функцию, которая по стартовым значениям скорости, угла и высоты вычисляла бы дальность полета. Используя эту подпрограмму-функцию, составьте алгоритм вычисления скорости по данным таблицы.
   б) Модифицируйте заполнение электронной таблицы (табл. 3.1) так чтобы с ее помощью можно было находить начальную скорость снаряда.
4. * Используя ваше решение задачи 4б из § 19, дополните заполнение электронной таблицы так, чтобы с ее помощью можно было находить изменение температуры движущегося тела.

Таблица 3.1

D	E	F	G
Скорость по вертик.	Скорость по гориз.	Скорость* Коэф.	
G5*Sin(G6)	G5*Cos(G6)		<Δt>
D2–(G4+F3*D2)*G2	E2–F3*E2*G2	$G3*\sqrt{D2^2+E2^2}$	<k>
D3–(G4+F4*D3)*G2	E3–F4*E3*G2	$G3*\sqrt{D3^2+E3^2}$	<g>
D4–(G4+F5*D4)*G2	E4–F5*E4*G2	$G3*\sqrt{D4^2+E4^2}$	<v>
D5–(G4+F6*D5)*G2	E5–F6*E5*G2	$G3*\sqrt{D5^2+E5^2}$	<A>
…	…	…	
D199–(G3+F200*D199)*G2	E199–F200*E199*G2	…	
…	…	…	

5 Между учениками возник спор. Одни утверждали, что для исследования процессов и явлений вполне достаточно компьютерных моделей и экспериментов с ними, поэтому нет необходимости тратить силы на разработку сложных математических методов исследования моделей. Другие им возражали. Какова ваша точка зрения по данному вопросу? Аргументируйте свою точку зрения.

§21 Моделирование процессов в биологии

В этом параграфе мы приступим к обсуждению биолого-экологических моделей.

В 1937 году на остров Протекшен завезли 8 фазанов. Никто на этих фазанов не охотился (ни люди, ни звери), корма для них было вдоволь, и через год фазанов стало 26. Прошел еще год, фазанов снова пересчитали — на этот раз их было 83. Интересно, сколько будет фазанов через n лет?

Мы намерены использовать компьютер для исследования того, как растет число фазанов на острове, значит, надо строить информационную модель. А построение любой модели начинается с выделения существенных факторов.

Факторов, влияющих на жизнь фазанов, много. Это и климатические условия, и даже конкретная погода в то или иное время года — скажем, холодная зима или засушливое лето отрицательно скажутся на росте популяции фазанов. Появление или быстрый рост вида животных, имеющих ту же кормовую базу, что и фаза-

ны, тоже не будет способствовать процветанию фазаньего сообщества. Хорошо еще, что на них никто не охотится.

Мы в принципе не можем учесть все существенные факторы, влияющие на жизнь фазаньего племени. Но естественно предположить, что прирост числа фазанов за единицу времени пропорционален их уже имеющемуся количеству. Такое предположение согласуется с обычными представлениями о размножении: чем больше живых организмов, тем больше у них потомков. А окружающая среда выступает как регулятор прироста количества фазанов.

Высказанные предположения показывают, что для нашей модели существенна следующая информация: начальное количество (обозначим его $M(0)$) и коэффициент прироста за один год (обозначим его k).

Пусть по истечении n лет количество фазанов достигло, как мы договорились, величины $M(n)$. Тогда прирост за один год составит $M(n + 1) - M(n)$. Значит, высказанные нами предположения можно записать равенством

$$M(n + 1) - M(n) = kM(n).$$

Преобразуя это равенство, получаем

$$M(n + 1) = M(n) + kM(n),$$

или

$$M(n + 1) = (1 + k)M(n).$$

По этой формуле, зная начальное количество $M(0)$ и коэффициент прироста k, можно сначала найти $M(1)$, т. е. число фазанов, которое будет через год, затем найти $M(2)$, т. е. количество фазанов через два года, затем $M(3)$ и т. д.

Те из вас, кто дружен с математикой, распознают в последовательности $M(n)$ геометрическую прогрессию. Обнаруженную закономерность можно сформулировать так: если действие окружающей среды сказывается лишь на скорости прироста, то живые организмы размножаются в геометрической прогрессии. И совершенно неважно, идет ли речь о фазанах или китах, о водорослях или гигантских секвойях. Мы построили модель некоторого биологического процесса, и при этом не играет роли, какие именно живые организмы участвуют в этом процессе. Построенную модель называют моделью **неограниченного роста**.

Если спросить, какого вида модель нами построена, каждый, наверное, сразу ответит — *математическая*. Действительно, все параметры в ней числовые, а связи между параметрами выражаются функцией

$$M(n) = (1 + k)^n M(0).$$

Мы надеемся, что все узнали в ней формулу общего члена геометрической прогрессии.

Задумаемся, адекватно ли наша модель отражает биологические процессы на Земле?

Ответ, по существу, очевиден: безудержный рост массы живых организмов, предсказанный данной моделью, привел бы к тому, что в какой-то момент она превысила бы массу Земли, что невозможно ни при каких, даже самых благоприятных, условиях. Масса живых организмов не может превысить массу планеты — это нарушило бы фундаментальный закон сохранения вещества. Модель неограниченного роста, которую мы построили, оказалась пригодной, как и следовало ожидать, лишь пока выполнено главное предположение — действие всех факторов выступает ограничителем только скорости прироста массы. Видимо, в нашей модели, описывающей действие природных факторов, надо что-то подправить.

Поступим так. По-прежнему не вдаваясь в подробности, как именно факторы окружающей среды влияют на жизнь организмов, выдвинем предположение, что имеется некоторое предельное значение массы растений, «проживающих» на той или иной территории. Так, ученые показали, что запас массы растений не может превосходить 20 т на гектар в полярной зоне, 350 т на гектар в лесной зоне, 440 т на гектар в тропиках. А на всей Земле масса растений не может превысить $5 \cdot 10^{12}$ т.

Еще одно наше предположение будет таким: чем ближе масса живых организмов к своей максимально возможной, тем меньшим становится коэффициент прироста k. В качестве обоснования для такого предположения вы могли бы использовать свои собственные наблюдения (на пришкольном участке, в саду): вначале растения быстро набирают массу, затем их рост становится все медленнее. Иными словами, коэффициент k не является неизменной величиной, а зависит от разности $L - M(n)$, где L — предельное значение массы растений на данной территории. Самая простая функция, как вы знаете, — это прямая пропорциональность. Будем считать, что коэффициент прироста меняется по формуле

$$k(n) = a(L - M(n))$$

для $n = 0, 1, 2, 3, \ldots$.

Подводя итог проделанной работе по уточнению модели, опишем получившуюся модель роста массы живых организмов.

Предположения:

- прирост массы живых организмов за единицу времени пропорционален уже имеющейся массе;
- существует некоторое предельное значение массы живых организмов;
- коэффициент прироста живых организмов за единицу времени пропорционален разности между максимально возможным значением массы и массой, имеющейся к данному моменту времени.

Параметры модели:

- начальная масса живых организмов $M(0)$;
- предельное значение массы живых организмов L;

- коэффициент пропорциональности a в формуле для коэффициента прироста;
- время n.

Связь между параметрами модели задается соотношением

$$M(n+1) = M(n) + aM(n)(L - M(n)),$$

где через $M(n)$ по-прежнему обозначена масса живых организмов по истечении n лет.

Эту модель принято называть **моделью ограниченного роста.** Однако получить и в этом случае явную формулу зависимости $M(n)$ от n, увы, невозможно. Остается надеяться на компьютерный эксперимент, который вы проведете, выполняя лабораторную работу 12.

ВОПРОСЫ И ЗАДАНИЯ

1. Какие предположения были сформулированы для модели неограниченного роста?
2. Как растет масса живых организмов, если действие окружающей среды сказывается только на скорости прироста?
3. Какие факторы ограничивают массу живых организмов? Приведите примеры.
4. Какие предположения были сформулированы для модели ограниченного роста?
5. Составьте алгоритм для вычисления массы живых организмов через заданное число лет, руководствуясь моделью ограниченного роста. Запишите его на подходящем для вас языке программирования.
6. а) Назовите параметры модели неограниченного роста. Какие существенные факторы описываются этими параметрами?
   б) Выполните задание *а* для модели ограниченного роста.

§22 Границы адекватности модели

В § 5 был сформулирован важный принцип: никакая модель не эквивалентна исходному объекту, процессу или явлению. Ведь в любой модели можно учесть только часть информации об объекте, процессе или явлении, самую существенную с точки зрения того, кто строит модель, но все равно не всю.

Но тот же принцип можно сформулировать иначе: *всякая модель имеет ограниченную область адекватности* и за пределами этой области она перестает удовлетворительно отражать свойства моделируемого объекта. Поэтому и применять модель для решения той или иной жизненной задачи допустимо только тогда, когда мы убедились, что не вышли за границы области адекватности.

Как же проверить, что выбранная нами модель применима? Прежде всего надо убедиться, что все факторы, существенные для данной задачи, присутствуют в модели. Затем надо проверить, что в исходных данных задачи значения параметров, описывающих действие факторов, не выходят за границы адекватности модели.

К сожалению, часто эти требования игнорируются. Мы нередко слышим предложения использовать в России экономические модели, прекрасно работающие в западных странах. Как правило, таким предложениям не предшествует анализ базовых предпосылок, на которых основываются предлагаемые модели. Ведь даже внешняя схожесть ситуаций еще не означает ни совпадения факторов, ни попадания в область адекватности.

О выделении существенных факторов мы уже говорили неоднократно. Теперь обсудим, как находить границы адекватности.

Как вы помните из § 5, адекватность модели определяется ее согласованностью с практикой и общетеоретическими положениями. Предложение провести вам какой-либо натурный эксперимент для проверки адекватности той или иной модели, конечно, выходит за рамки курса информатики. Поэтому наш эксперимент будет компьютерным и основанным на общетеоретических положениях.

В § 21 мы упомянули, что модель неограниченного роста остается адекватной, пока масса живых организмов достаточно мала по сравнению с предельно допустимой массой этих организмов в данных природных условиях. Попытаемся определить, насколько мала должна быть исходная масса живых организмов по отношению к предельной массе, чтобы модель неограниченного роста оставалась адекватной в течение нескольких лет (напомним, что существование предельного значения массы — следствие общетеоретических положений). Иными словами, мы хотим найти границы адекватности модели неограниченного роста.

Вспомним параметры и связи между этими параметрами, участвующие в моделях ограниченного и неограниченного роста.

Модель неограниченного роста: начальная масса $M(0)$, коэффициент прироста k, число лет n, масса живых организмов через n лет $M(n)$, связь между параметрами

$$M(n+1) = (1+k)M(n).$$

Модель ограниченного роста: начальная масса $M_O(0)$, коэффициент прироста k, предельное значение массы L, число лет n, масса живых организмов через n лет $M_O(n)$ (буква «O» внизу показывает, что вычисление массы идет в модели ограниченного роста), связь между параметрами, найденная в лабораторной работе 12:

$$M_O(n+1) = \left(1 + k\frac{L - M_O(n)}{L - M(0)}\right) M_O(n).$$

Поскольку $M_O(0) = M(0)$, то нетрудно подсчитать, что $M_O(1) = M(1)$, т. е. через год масса живых организмов, подсчитанная по обеим моделям, будет еще одинаковой. Но вот $M_O(2)$ будет уже меньше, чем $M(2)$. И чем дальше, тем больше будет различие между значениями M_O и M. Значит, надо договориться, какое расхождение между M_O и M мы будем считать еще допустимым. Пусть, к примеру, мы считаем модель неограниченного роста адекватной, если разница $M - M_O$ составляет не более 10% от M_O.

Чтобы найти границы адекватности, мы должны установить, в каких пределах и как по отношению друг к другу могут меняться параметры модели, чтобы она оставалась адекватной. Ответы на эти вопросы вы получите, выполнив очередную лабораторную работу. А сейчас подготовим для этой лабораторной работы макет электронной таблицы. Можно ее заполнить, например, так, как показано в таблице 3.2.

Таблица 3.2

	A	B	C	D
1	Коэффициент прироста *k*	<k>	Предельное значение массы *L*	<L>
2	Год	Неограниченный рост	Ограниченный рост	Отклонение, %
3	0	1	1	(B3–C3)/C3*100
4	A3+1	(1+B1)*B3	(1+B1*(D1–C3)/(D1–C3))*C3	(B4–C4)/C4*100
5	A4+1	(1+B1)*B4	(1+B1*(D1–C4)/(D1–C3))*C4	(B5–C5)/C5*100
6	A5+1			
7				

В ячейку B1 заносится коэффициент прироста *k*, а в клетку D1 — предельное значение массы живых организмов. В ячейки B3 и C3 мы занесли значение начальной массы живых организмов, взяв ее равной 1. Заполнение остальных ячеек, скорее всего, ясно из написанных в них формул.

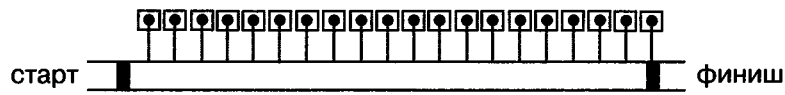
И снова немного истории. Первым модель неограниченного роста рассмотрел Т. Мальтус (1798 г.). Опираясь на эту модель, он пытался обосновать неизбежность войн и других кризисных явлений социально-политической жизни человеческого общества. Он ввел понятие демографического давления как показателя превышения численности населения, проживающего на данной территории, над возможностью данной территории обеспечить это население продовольствием. Из его заключений делался вывод о необходимости постоянного расширения жизненного пространства, и этот вывод использовался для построения и оправдания расовых теорий, националистических теорий борьбы за жизненное пространство и т. п.

Проанализируем эту ситуацию с позиций модельного подхода. В модели неограниченного роста в качестве существенных принимались только биологические факторы. Но если живые организмы образуют сообщества (стада, стаи и т. п.), то вступают в силу иные факторы, характерные для данного сообщества. Их можно было бы назвать социальными, хотя термин «социальные» обычно применя-

ют к сообществам людей. Что касается человека, то для него одним из важнейших социальных факторов является развитие науки и производства. За счет применения удобрений с той же, что и раньше, сельскохозяйственной территории стали снимать в несколько раз больший урожай, а значит, эта территория способна прокормить большее население, чем прежде. Поэтому с точки зрения информатики несостоятельность применения модели неограниченного роста к человеческому обществу состоит уже в том, что учтены не все существенные факторы, а сама модель применяется не в той области, где она является адекватной, — в области социально-политической, а не чисто биологической.

ВОПРОСЫ И ЗАДАНИЯ

1. Сформулируйте принцип адекватности модели. Как его можно обосновать?
2. Выполнение каких условий надо проверить, прежде чем использовать какую-либо модель для решения жизненной задачи?
3. Что значит найти границы адекватности данной модели?
4. Для изучения графика бега на дистанции 100 м были установлены на одинаковом расстоянии 20 фотодатчиков, фиксирующих время, которое прошло от старта до пересечения спортсменом соответствующего светового луча.



В таблице 3.3 приведены данные датчиков, округленные до десятых долей секунды.

Таблица 3.3

Номер датчика	1	2	3	4	5	6	7	8	9	10
Время, с	1,2	1,7	2,1	2,5	2,9	3,3	3,7	4,1	4,5	4,9

Номер датчика	11	12	13	14	15	16	17	18	19	20
Время, с	5,3	5,7	6,1	6,5	6,9	7,3	7,8	8,4	9,0	9,8

Попытайтесь по этим данным определить, на каких участках дистанции бег спортсмена адекватно описывается моделью равномерного движения, а на каких — равноускоренного.

5. В объяснительном тексте подготовлено заполнение электронной таблицы для определения границ адекватности сравнением моделей ограниченного и неограниченного роста. Составьте алгоритм, позволяющий выполнить ту же работу, и запишите его на изученном вами языке программирования.

§ 23 Моделирование эпидемии гриппа

Модель ограниченного роста может послужить основой для создания моделей других процессов, в том числе и важных с социальной точки зрения. Например, для принятия эффективных мер борьбы с эпидемией гриппа очень важно знать, как такая эпидемия может развиваться, чтобы в нужный момент усилить местную систему здравоохранения: в аптеки завезти больше противогриппозных медикаментов, развернуть дополнительные места в больницах, наметить карантинные меры и т. д. Вот моделированием развития эпидемии гриппа мы сейчас и займемся.

Пусть в некотором населенном пункте с численностью населения L началась эпидемия гриппа. Прогнозировать в нашей модели мы будем количество людей, болеющих гриппом на n-й день после начала эпидемии. Обозначим это количество $B(n)$. На первый взгляд величина $B(n)$ удовлетворяет предположениям модели ограниченного роста. Действительно, прирост числа больных пропорционален числу уже заболевших, но, разумеется, не может превосходить числа L. Отметим, что отсчет начинается с $n = 1$, так что начальное значение — это $B(1)$. В качестве $B(0)$ возьмем значение 0 и для $B(n)$ напишем уже известную нам формулу:

$$B(n+1) = \left(1 + k\frac{L - B(n)}{L}\right)B(n).$$

Однако согласно графику функции $B(n)$ (вы построили график для $M(n)$, выполняя лабораторную работу 12) приходим к странному выводу — количество больных достигает некоторого уровня и дальше не меняется. Эпидемия продолжается вечно!

Так, конечно, не бывает. Значит, построенная модель неадекватна. И надо поискать неучтенные существенные факторы модели. Впрочем, сделать это нетрудно — в модели не учитывается, что люди выздоравливают, у них вырабатывается иммунитет и больше они не заболевают. Для простоты будем считать, что болезнь длится 7 дней. Обозначим через $V(n)$ количество людей, уже переболевших гриппом к n-му дню с момента начала эпидемии. Тогда ясно, что $V(1) = V(2) = V(3) = \ldots = V(7) = 0$, а $V(8) = B(1)$, $V(9) = V(8) + B(2)$ и, вообще, $V(n + 1) = V(n) + B(n)$ при $n \geqslant 8$. Запишем получившуюся модель:

$$V(1) = V(2) = \ldots = V(7) = 0;$$

$$V(8) = B(1);$$

$$V(n + 1) = V(n) + B(n) \text{ при } n > 7;$$

$$B(n+1) = \left(1 + k\frac{L - B(n) - V(n)}{L}\right)B(n) \text{ при } n > 0.$$

ВОПРОСЫ И ЗАДАНИЯ

1. Какие предположения положены в основу модели эпидемии гриппа? Каковы параметры этой модели?
2. Продумайте и предложите заполнение электронной таблицы для проведения компьютерного эксперимента с моделью эпидемии гриппа.
3. Составьте алгоритм для вычисления числа больных гриппом через заданное число дней, руководствуясь моделью эпидемии гриппа. Запишите его на подходящем для вас языке программирования.
4. * Для некоторых штаммов гриппа иммунитет оказывается нестойким и длится несколько дней. Постройте модель эпидемии гриппа для такой ситуации.

§24 Вероятностные модели

Со случайностью мы сталкиваемся на каждом шагу. Можно случайно встретить на улице своего приятеля, можно случайно разбить чашку, можно, наконец, случайно выиграть в лотерею.

Но иногда случайные числа нужно уметь получать искусственным образом. Например, определить случайным образом порядок выступления фигуристов на соревновании или номера лотерейных билетов, на которые падает выигрыш. Даже при проведении научных экспериментов случайность изменения условий эксперимента тоже имеет значение. Как в этих случаях организуется случайная последовательность чисел?

Казалось бы, это совсем разные вещи: природная случайность и случайность, специально организуемая человеком. Однако это не совсем так.

Сначала человек научился использовать природную случайность. Чтобы определить, кому начинать игру, можно подбрасывать монетку. Чтобы случайно выбрать одно число из шести, можно подбрасывать кубик. Для того чтобы строить более разнообразные последовательности случайных чисел, можно регистрировать время прихода космических частиц (например, с помощью счетчика Гейгера). Но как *изучать* процессы, в которых самой природой заложена случайность?

Мы надеемся, что все, изложенное ранее, сделало для вас очевидным важный тезис: изучить — значит построить модель, отражающую существенные стороны изучаемого объекта, процесса или явления. Изучать же процессы, в которых случайность — неотъемлемая черта, человеку приходится весьма часто. Это могут быть обычные телефонные разговоры по линии связи, начинающиеся в случайное время и непонятно сколько продолжающиеся, или приход покупателя в магазин и продолжительность его пребывания там, или в конце концов непредсказуемое поведение элементарных частиц в ядерном реакторе.

Все непонятно, неясно, непредсказуемо, однако линия связи должна иметь достаточную пропускную способность, в универсаме в нужное время должно быть нужное количество кассиров, а реакторы должны надежно работать.

Для успешного решения всех этих задач необходимо уметь делать следующее:

1. Получать искусственную последовательность случайных чисел, успешно заменяющую реальную, определяемую случайными событиями.

2. С помощью этой последовательности моделировать случайные события и находить параметры, необходимые для проектирования линии связи (ядерного реактора или графика работы продавцов магазина).

Отметим прежде всего, что для обсуждения случайных событий надо иметь дело с многократно повторяющимися опытами, при которых могут происходить данные события. Каждое повторение опыта обычно называют **испытанием**, а множество многократно проведенных испытаний в одном опыте — **серией** испытаний. **Частотой** события A в какой-то серии испытаний называется число, показывающее, сколько раз произошло событие A в этой серии.

Важной числовой характеристикой случайного события является **вероятность** его наступления. Связаны ли вероятность и частота? Да: чем чаще происходит событие, тем больше вероятность его наступления. Тем не менее сказать, что вероятность — это частота наступления события, очевидно, нельзя: в серии с бóльшим числом испытаний одно и то же событие будет иметь большую частоту, нежели в серии с меньшим числом испытаний. Мы провели два опыта по подбрасыванию обыкновенного игрального кубика. В первом опыте было проведено 20 испытаний, во втором — 40 испытаний. Исходом в каждом испытании было число очков, выпавших на кубике. Результаты опытов представлены в таблице 3.4.

Таблица 3.4

Количество испытаний в серии	Исходы	1	2	3	4	5	6
20	Частота	3	3	3	2	4	5
	Относительная частота	0,15	0,15	0,15	0,1	0,2	0,25
40	Частота	6	6	7	7	6	8
	Относительная частота	0,15	0,15	0,175	0,175	0,15	0,2

Во втором столбце вы прочитали новые слова: «Относительная частота». Это число, равное отношению частоты к числу испытаний в данной серии. Она, как мы видим, мало меняется при увеличении числа испытаний в серии.

Конечно, 20 и даже 40 испытаний в серии — это совсем немного. Но для большинства случайных событий выполняется свойство: *чем больше увеличивать число испытаний в сериях, тем меньше будут отличаться друг от друга относительные частоты данного события в этих сериях.* Иными словами, *при большом числе испытаний относительные частоты одного и того же события в различных сериях группируются вокруг одного и того же числа.* Это число и называют *вероятностью данного события.*

В ряде ситуаций определить вероятность события можно и не прибегая к длинным сериям испытаний. Ясно, например, что если игральный кубик не имеет дефектов, то каждое число очков от 1 до 6 должно выпадать одинаково часто. А это означает, что относительная частота каждого из исходов должна группироваться около числа $\frac{1}{6}$. Следовательно, вероятность выпадения на кубике данного количества очков равна $\frac{1}{6}$.

Далеко не всегда посредством умозаключений можно установить вероятность исследуемого события. Тогда без проведения испытаний не обойтись. Вот здесь-то и может пригодиться компьютерное моделирование: если построить модель случайного процесса, то наблюдать его на экране компьютера гораздо дешевле, а иногда и безопаснее, чем проводить натурные эксперименты, многократно повторяя данный процесс.

Модели, в которых используются вероятности моделируемых событий, называются **вероятностными моделями.**

ВОПРОСЫ И ЗАДАНИЯ

1. Приведите примеры, когда необходимо получать последовательности случайных чисел. Как такие последовательности в этих случаях создавались?
2. Вспомните, в каких жизненных ситуациях вы пользовались случайными числами. Как вы такие числа получали?
3. Что такое частота и относительная частота случайного события? Что такое вероятность случайного события?
4. Какие модели называют вероятностными?
5. Для указанных ниже опытов и их исходов определите, в каких случаях вероятность исходов может быть получена умозрительными заключениями, а в каких обязательно требуется провести серии испытаний.

   а) Опыт — подбрасывание кнопки; исходы: кнопка упала острием вверх, кнопка упала на острие.

   б) Опыт — вытаскивание бочонка для игры в лото; исходы: вынут бочонок с числом 1, вынут бочонок с числом 2, ..., вынут бочонок с числом 90.

   в) Опыт — подбрасывание бутерброда; исходы: бутерброд упал маслом вверх, бутерброд упал маслом вниз.

   г) Опыт — удар по воротам в футбольном матче; исходы: гол забит, гол не забит.

§25 Датчики случайных чисел и псевдослучайные последовательности

В начале предыдущего параграфа были сформулированы два условия, необходимые для моделирования случайных процессов:

- наличие способа получать последовательность случайных чисел, успешно заменяющих реальную и определяемую случайными событиями;
- умение с помощью этой последовательности моделировать случайные процессы.

Одно случайное число получить легко. Посмотрите, например, на часы в этот самый момент и засеките, сколько секунд они показывают. Вот вам и случайное число.

Можно еще раз взглянуть и получить второе случайное число. В чем же заключается случайность этих чисел? В случайности выбора одного числа из некоторого множества чисел, в данном случае — из множества целых чисел от 1 до 60.

Еще один способ получения случайного числа уже обсуждался в предыдущем параграфе: будем бросать игральный кубик и смотреть, какое число выпадет на верхней грани. Здесь множество чисел, которые могут появиться, содержит всего 6 элементов: числа 1, 2, 3, 4, 5, 6. Для хорошего кубика появление любого из этих чисел равновероятно (т. е. вероятности выпадения каждого из чисел равны).

Рассмотрим третий способ получения случайного числа. Будем стрелять по мишени из лука. Как бы хорошо мы ни прицеливались, колебания воздуха (в частности, порывы ветра), непредсказуемость силы, с которой натягивается тетива, индивидуальность аэродинамических свойств каждой конкретной стрелы и т. п. вносят случайность в ее полет. Поэтому расстояние от центра мишени до точки, куда попала стрела, тоже случайное число. Здесь перечислить множество всех возможных значений не удастся — их бесконечно много. Фактически таким значением может быть любое действительное неотрицательное число. Конечно, если мы прицеливаемся хорошо, то здесь появление тех или иных чисел неравновероятно: маленькие числа должны появляться чаще больших.

Обычно нужна не любая последовательность случайных чисел, а удовлетворяющая некоторым условиям. В случае с кубиком вероятности всех исходов одинаковы, поэтому получающаяся последовательность случайных чисел называется **равномерно распределенной**. Если же вероятность появления очередного числа неодинакова для разных чисел, то это неравномерно распределенная последовательность.

Потребность в случайных числах возникла довольно давно. И способов их получения разработано достаточно. Ученые раскладывали карты, бросали кости, вытаскивали черные и белые шары из коробки, которую предварительно как следует трясли.

В 1927 году Л. Триппетт опубликовал таблицы, содержащие свыше 40 000 случайных цифр, произвольно взятых из отчетов о переписи. Позже были сконструированы специальные машины, механически вырабатывающие случайные числа (первую такую машину использовали М. Дж. Кендалл и Б. Бэбингтон-Смит при создании таблиц из 100 000 случайных цифр).

Практически сразу же после создания ЭВМ начались поиски эффективных методов получения случайных чисел для использования в компьютерных программах. Можно было пользоваться и таблицами, но они требовали для своего размещения большой объем оперативной памяти. Поначалу к ЭВМ подключали **датчики случайных чисел** (ДСЧ), основанные на различных физических эффектах, в основном на так называемом эффекте шума электронных ламп и излучении радиоактивных веществ.

Но датчики были слишком медленными, дорогими и небезопасными. Кроме того, при отладке программ, использующих ДСЧ, крайне важно, чтобы при одинаковых начальных данных получались одинаковые конечные результаты, чего датчики на основе физических эффектов не могли обеспечить в принципе: потоку космических частиц не прикажешь прилететь еще раз с той же частотой.

Несовершенство этих методов пробудило интерес к получению случайных чисел с помощью арифметических операций самого компьютера. Первым такой подход предложил в 1946 году Джон фон Нейман, известный вам как разработчик архитектуры первых ЭВМ. Его идея состоит в том, что нужно только самое первое случайное число. А дальше применяется следующее рекуррентное правило: число возводится в квадрат и из результата берется середина. Например, мы хотим получить случайную последовательность, состоящую из десятизначных чисел. Пусть первое десятизначное число нашей последовательности 5772156649 (мы просто набрали подряд 10 цифр, вы можете написать другое число). Возводим его в квадрат и получаем двадцатизначное число 33317792380594909201. Отбрасываем пять первых и пять последних цифр. Получаем новое десятизначное число 7923805949. Это второй член нашей последовательности. Возводим его в квадрат и получаем 62786700717407790601. Снова отбрасываем пять первых и пять последних цифр и получаем третий член нашей последовательности: 7007174077. Так же получаем четвертое число и т. д.

Фактически строится рекуррентно заданная последовательность чисел. А где же здесь случайность? Конечно, никакой случайности здесь нет. Но важно, что получаемые таким способом числа ведут себя как случайные: частота появления любого числа в этой последовательности примерно одинакова. Математики доказали, что если рассмотреть бесконечную последовательность таких чисел (т. е. как будто мы проводим бесконечную серию опытов), то для каждого числа вероятность его появления одна и та же. А в моделировании экспериментов со случайными числами только это и надо!

В итоге вместо последовательности случайных чисел мы имеем ее модель, сохраняющую самое главное свойство: равномерное распределение вероятностей появления членов этой последовательности. Такие последовательности, моделирующие последовательности случайных чисел, стали называть **псевдослучайными**. Зато алгоритмы получения псевдослучайных последовательностей попрежнему называют датчиками случайных чисел.

В течение почти десяти лет после изобретения Дж. фон Неймана математики придумывали различные датчики случайных чисел. Проблема в том, что все псевдослучайные последовательности являются периодическими, т. е., начиная с некоторого места, числа в последовательности появляются в том же порядке. Для последовательности, предложенной Дж. фон Нейманом, этот эффект осознать совсем легко: если еще раз появится число 5772156649, то следующее за ним будет 7923805949, потом 700717407 и т. д. И тогда уже о моделировании случайности говорить не приходится. Правда, произойдет это только через 10^{10} операций. Для человека провести столько опытов в лучшем случае дело нескольких лет, даже если просто подбрасывать монетку, а для компьютера в худшем случае дело нескольких часов (если моделируется что-то сложное). Но для получения достоверных результатов моделирования, например, работы ядерного реактора нередко требуется намного больше случайных чисел. Вот за создание быстрых и надежных ДСЧ с большим периодом (порядка 10^{100}) и развернулась борьба математиков.

Сегодня в любом языке программирования есть ДСЧ. Обычно он выполнен в виде подпрограммы-функции и для пользователя выглядит как оператор, присваивающий переменной случайное значение из промежутка [0; 1). Есть такой оператор и в электронной таблице Microsoft Excel. В алгоритмах мы этот оператор будем записывать сокращением ДСЧ. Например, по команде

x := ДСЧ;

переменной x будет присвоено случайное значение из промежутка [0; 1). Если требуется присвоить равномерно распределенное случайное значение из промежутка $[a; b)$, то используется команда

$x := a + (b - a)$*ДСЧ.

Иногда требуется получить целое случайное число x, удовлетворяющее неравенству $m \leqslant x \leqslant n$. Тогда используется команда

$x := m$ + INT(($n + 1 - m$)*ДСЧ).

ВОПРОСЫ И ЗАДАНИЯ

1. Что такое случайное число?
2. Что значит последовательность случайных чисел равномерно распределена?

3. Приведите примеры процессов, в которых встречаются равномерно и неравномерно распределенные последовательности случайных чисел.

4. Предположим, что вы хотите получить последовательность случайных чисел от 1 до 99. Какой из перечисленных ниже методов вы предпочтете?
а) Выбрать из взятых наугад страниц телефонного справочника две последние цифры наугад выбранных номеров телефонов.
б) Воспользоваться двумя младшими разрядами номеров страниц того же справочника.
в) Позвонить нескольким знакомым и попросить их назвать какое-нибудь натуральное число, меньшее 100.
г) Взять номера, выигравшие в «Русском лото» за последние несколько розыгрышей.

5. Проверьте метод фон Неймана для двузначных чисел, а именно установите, как ведет себя последовательность в зависимости от выбора первого числа последовательности.

6* а) Объясните, почему в последовательности десятизначных чисел, построенной по методу фон Неймана, некоторое число обязательно повторится (следовательно, последовательность наверняка периодична).
б) Предложите какой-нибудь способ построения непериодических последовательностей n-значных чисел.

7. Подумайте и попытайтесь сформулировать ответ на следующий вопрос: в каких задачах при компьютерном моделировании вам кажется более уместным использовать ДСЧ языка программирования, а в каких — электронной таблицы?

§26 Моделирование случайных процессов

Применим полученные знания для построения вероятностных моделей. Речь пойдет о моделировании так называемых систем массового обслуживания. Представление о таких системах дают следующие примеры:

- На телефонную станцию поступают телефонные звонки. Для их обслуживания требуются свободные каналы связи. Сколько таких каналов должно быть, чтобы не возникало простоев оборудования и слишком больших очередей абонентов, дожидающихся связи?
- На станцию «Скорой помощи» поступают вызовы от населения. Сколько врачебных бригад должно дежурить в то или иное время суток, чтобы бесперебойно оказывать помощь и при этом чтобы не оставались без работы бригады?
- В магазине самообслуживания покупатели рассчитываются через кассы на выходе. Сколько надо установить кассовых аппаратов и каков должен быть режим работы кассиров, чтобы не создавалось слишком длинных очередей и не простаивали без дела кассиры?

Во всех этих примерах есть две черты, их объединяющие:

1) требуется обрабатывать некоторые объекты, поступающие постоянно, но через случайные промежутки времени;

2) время на обработку каждого объекта тоже случайная величина.

Системы с такими особенностями и называют **системами массового обслуживания.**

Для моделирования систем массового обслуживания надо знать характер тех «случайностей», которые описаны в пунктах 1 и 2. Для этого их статистически изучают, проводя соответствующие наблюдения и измерения. Поэтому для разных систем массового обслуживания получаются, вообще говоря, разные модели, их описывающие. Мы разберем самую простую — модель кассового обслуживания в магазине.

В разное время суток интенсивность посещения магазина покупателями различна. Мы выберем некоторый интервал времени, когда интенсивность не сильно меняется и можно считать, что количество покупателей, приходящих в магазин в течение минуты, — это равномерно распределенная случайная величина, принимающая целые значения в интервале от 0 до N_0.

Точно так же равномерно распределенной случайной величиной мы будем считать время, которое требуется кассиру на обслуживание покупателя. Пусть данная случайная величина изменяется в пределах от T_0 до T_1.

Кроме того, прежде чем подойти к кассе, покупатель проводит некоторое время в торговом зале, выбирая товары. Это тоже случайная величина, и пусть она равномерно распределена в интервале от V_0 до V_1.

Набрав корзину с товарами, покупатель идет к той кассе, где меньше очередь. Поэтому можно считать, что длина очереди во всех кассах одинакова. Мы хотим найти такое число k — количество касс, чтобы очередь в каждую из них не превышала m человек.

Таким образом, для моделирования процесса обслуживания покупателей нам трижды требуется случайное число: количество покупателей в магазине, время обслуживания одного покупателя кассиром и время пребывания покупателя в торговом зале для выбора товаров.

Выше мы фактически перечислили основные предположения модели и определили параметры, которые ее описывают. Теперь между этими параметрами надо установить связи.

Пусть по истечении t минут в торговом зале магазина находятся (но не стоят в кассу) a_t покупателей. За следующую минуту в магазин зайдет еще $\mathrm{INT}((N_0 + 1) \cdot \text{ДСЧ})$ покупателей — эта формула приведена в конце предыдущего параграфа. Иными словами,

$$a_{t+1} = a_t + \mathrm{INT}((N_0 + 1) \cdot \text{ДСЧ}) - b_t,$$

где b_t — количество покупателей, оплачивающих в кассах покупки в течение t-й минуты.

Как определить число b_t? Будем рассуждать так. Поскольку максимальная продолжительность выбора покупок V_1, то все a_t

покупателей зашли в торговый зал не более как V_1 минут назад. Но некоторые из них находятся в зале менее V_0 минут, т. е. они уж точно покупки еще не выбрали. По условию объявлено равномерное распределение покупателей, поэтому на интервал времени $V_1 - V_0$ приходится в среднем $\frac{V_1 - V_0}{V_1} a_t$ покупателей. Но не все эти покупатели отправились в кассу. Поэтому и данное число мы должны умножить на очередное случайное число. Итак,

$$b_t = \mathrm{INT}\left(\left(\frac{V_1 - V_0}{V_1} a_t + 1\right) \cdot \text{ДСЧ}\right).$$

(Хотя здесь и написано снова ДСЧ, этот оператор выработает уже другое число, нежели то, которое фигурировало при вычислении a_t.)

Теперь подсчитаем, сколько человек стоит в очереди. Обозначим через c_t количество покупателей, стоящих в очередях в кассу. Тогда $c_{t+1} = c_t + b_t - d_t$, где d_t — количество покупателей, обслуженных кассами в течение t-й минуты. Время, которое уходит на обслуживание одного покупателя, — случайная величина, равномерно распределенная на промежутке $[T_0; T_1)$. С помощью датчика случайных чисел это время моделируется величиной

$$(T_1 - T_0) \cdot \text{ДСЧ} + T_0.$$

Но тогда за 1 мин будет выполнена

$$\frac{1}{(T_1 - T_0) \cdot \text{ДСЧ} + T_0}$$

часть работы по обслуживанию одного покупателя. В k кассах эта величина будет соответственно в k раз больше. Следовательно,

$$d_t = \frac{k}{(T_1 - T_0) \cdot \text{ДСЧ} + T_0}.$$

Осталось сформулировать критерий «большой очереди». Напомним, что мы договорились считать очередь большой, если в каждую кассу стоит больше чем m человек, т. е. когда $c_t > mk$.

Чтобы полностью определить величины a_t, b_t, c_t и d_t, надо еще задать начальные значения этих величин: a_0, b_0, c_0 и d_0. Последовательно вычисляя данные величины, будем подсчитывать количество минут, когда очередь оказывалась большой. Если число таких минут в течение часа оказывается большим, значит, касс работает мало и надо открыть дополнительные. Если же таких минут не оказалось вообще, то можно попробовать уменьшить число работающих касс.

Как обычно, для реализации этой модели на компьютере требуется выбрать подходящую технологию. Мы выберем алгоритми-

зацию. Вот как может выглядеть алгоритм, позволяющий вести компьютерный эксперимент с этой моделью.

```
Алгоритм
  вещ: N0, T0, T1, V0, V1, r;
                (* N0 — максимально возможное число покупателей,
                заходящих в магазин в течение минуты,
                [T0; T1) — промежуток времени, необходимый для
                обслуживания одного покупателя кассиром,
                [V0; V1) — промежуток времени, необходимый покупателю для выбора товаров,
                k — количество работающих касс,
                m — максимально допустимая длина очереди в одну кассу,
                r — доля тех минут от общей продолжительности времени работы магазина, когда очередь в кассы превышала допустимую длину *)
  вещ: a, b, c, d, T, s;
  цел: k, m;
  { Сообщить "Введите максимально возможное число посетителей,
              входящих в магазин в течение минуты";
    Запросить N0;
    Сообщить "Введите два числа, задающие промежуток времени,
              необходимый для обслуживания одного покупателя";
    Запросить T0, T1;
    Сообщить "Введите два числа, задающие промежуток времени,
              необходимый покупателю для отбора товаров";
    Запросить V0, V1;
    Сообщить "Введите максимально допустимую длину очереди";
    Запросить m;
    Сообщить "Введите предлагаемое вами количество работающих
              касс";
    Запросить k;
    Сообщить "Введите предлагаемое вами общее время, в течение
              которого моделируется работа магазина";
    Запросить T;
    a := 0; b := 0; c := 0; d := 0;
    s := 0;                          (* количество минут, когда очередь
                                        превышала допустимую длину *)
    Делать от I := 1 до T
      { a:= a + INT((N0 + 1) * rand (1)) - b;
                    (* оператор получения случайного числа, равномерно распределенного на промежутке [0; v),
                                        будем обозначать rand(v) *)
        Если a < 0 то { a := 0; }    (* в магазине не может находиться отрицательное количество человек! *)
        c := c + b - d;
```

```
            Если c < 0 то { c := 0; }      (* в очереди не может стоять
                                              отрицательное количество человек! *)
            Если c > m*k то { s := s + 1; }
              b := INT((a * (V1 - V0)/V1 + 1) * rand (1));
              d := k/((T1 - T0) * rand (1) + T0);
          }
        r := s/T;
        Сообщить "Количество работающих касс", k;
        Сообщить "Доля плохих минут составляет", r;
    }
```

Компьютерный эксперимент с этой моделью вы проведете, выполняя лабораторную работу 16. Мы тоже проводили компьютерный эксперимент. Его результаты для различных наборов начальных данных представлены в таблицах 3.5—3.7.

Таблица 3.5

$N_0 = 4;\ T_0 = 0{,}5;\ T_1 = 2;\ V_0 = 3{,}5;\ V_1 = 6;\ m = 3.$

Количество работающих касс	1	2	3	4	5
Доля плохих минут	0,99994	0,99951	0,00492	0	0

Таблица 3.6

$N_0 = 10;\ T_0 = 1;\ T_1 = 3{,}5;\ V_0 = 4;\ V_1 = 10;\ m = 3.$

Количество работающих касс	7	8	9	10	11
Доля плохих минут	0,99989	0,99988	0,99932	0,93705	0,07146

Таблица 3.7

$N_0 = 6;\ T_0 = 1;\ T_1 = 3{,}5;\ V_0 = 4;\ V_1 = 8;\ m = 3.$

Количество работающих касс	3	4	5	6	7	8
Доля плохих минут	0,99989	0,99989	0,99982	0,93601	0,01193	0,0001

У вас численные результаты могли получиться иные — датчик случайных чисел мог дать совсем иные значения, нежели у нас. Тем не менее мы убеждены, что общий характер процессов одинаков. Обратим внимание на то, что с увеличением числа касс ситуация из очень плохой скачком меняется на очень хорошую. Ведь доля 0,93 означает, что очереди в кассах имеются на протяжении 55 мин каждого часа, а доля 0,07 — что такие очереди будут в среднем всего лишь 4 мин за час. В последнем случае у руководства

магазина возникает соблазн закрыть хотя бы одну кассу. Но стоит это сделать, как сразу возникнут большие очереди, и покупатель уйдет в соседний магазин, где его обслужат без очереди. У опытного менеджера очередей в магазине не будет не потому, что там слишком много продавцов или кассиров, а потому, что их там столько, сколько нужно.

Вывод о скачкообразном изменении ситуации мы сделали на основе нескольких компьютерных экспериментов с моделью. Но имеется математическая теория систем массового обслуживания, которая этот вывод строго обосновывает. Правда, эта теория не всегда способна указать, сколько именно надо открыть касс, проложить телефонных линий и т. п. Для получения таких конкретных рекомендаций как раз и требуется компьютерное моделирование. Здесь проявляются непростые взаимоотношения между теорией и компьютерной практикой.

Вычислительные методы, использующие датчик случайных чисел, получили название **методов Монте-Карло** (по названию города, где расположена знаменитая рулетка, которую можно рассматривать как «генератор» случайных чисел).

ВОПРОСЫ И ЗАДАНИЯ

1. Какие системы называют системами массового обслуживания? Назовите характерные черты таких систем.

2. Каковы преимущества компьютерного моделирования систем массового обслуживания?

3. Запишите алгоритм, предложенный в объяснительном тексте параграфа, на изученном вами языке программирования.

4. Всем хорошо известно такое физическое явление, как броуновское движение частицы, плавающей по поверхности жидкости. Оно вызывается хаотическим движением молекул жидкости.

   а) Можно считать, что частица за некоторую единицу времени (весьма малую, разумеется) перемещается на одно и то же расстояние в случайном направлении. Смоделируйте броуновское движение, рассматривая направление движения (т. е. угол по отношению, скажем, к оси абсцисс) как равномерно распределенную случайную величину на промежутке $[0; 2\pi)$. Можно также считать, что время t в этом процессе принимает последовательно значения 0, 1, 2, 3 и т. д. Предусмотрите в этой модели, чтобы для каждого момента времени t подсчитывалось расстояние s от точки начального положения частицы до точки ее нахождения в момент t. Выберите подходящую, на ваш взгляд, информационную технологию компьютерной реализации построенной вами модели и создайте соответствующую компьютерную модель.

   б) Постройте модель броуновского движения частицы, предполагая, что не только направление движения является случайной величиной, но и длина «пробега» тоже равномерно распределенная случайная величина на промежутке $[0; 1)$. И в этом случае выберите подходящую, на ваш взгляд, информационную технологию компьютерной реализации построенной модели и создайте соответствующую компьютерную модель.

§ 27 Метод Монте-Карло



Рис. 3.2

В предыдущих параграфах этой главы мы построили немало математических моделей для описания различных явлений. И у вас могло сложиться впечатление, что математические модели нужно строить только для решения задач, весьма далеких от математики. На самом деле и в математике для решения той или иной задачи может потребоваться построение математической модели. Одна из таких задач — вычисление площади геометрической фигуры. Конечно, для простейших фигур (прямоугольников, треугольников, кругов) вычисление площади не составляет труда: надо в известные формулы подставить исходные данные. А как быть, если фигура имеет сложную форму (рис. 3.2) и требуется вычислить ее площадь?

Можно предложить разные модели, позволяющие решить эту задачу. Например, в 6 классе вас учили использовать палетку: на фигуру накладывается клетчатая прозрачная бумага (палетка) и подсчитывается количество квадратиков, попавших в фигуру. В этой модели молчаливо предполагается, что, чем мельче клетки, тем точнее будет результат независимо от того, каким образом наложить палетку на фигуру.

Можно придумать и физическую модель: скопировать фигуру на картон, аккуратно вырезать ее, взвесить и поделить на вес единичного квадрата из этого же картона.

В 11 классе вы познакомитесь еще с одним способом нахождения площадей фигур — с помощью интегралов.

Как видите, и для решения математических задач можно составлять различные математические модели. Математическая модель объекта, процесса или явления, изучаемого какой-либо научной дисциплиной, — это «мост» между этой дисциплиной и математикой. Математическая модель, применяемая для решения математической задачи, — это «мост» между различными разделами математики. Например, метод координат позволяет создавать алгебраические модели геометрических объектов.

Предложенные модели вычисления площади имеют, однако, один недостаток — для них довольно трудно составить алгоритмы для расчетов с помощью компьютера, т. е. трудно преобразовать эти модели в компьютерные.

Математическая модель, которую мы сейчас построим, может показаться вам неожиданной, но она позволяет очень эффективно применять компьютер для решения задач на нахождение площадей, объемов и т. п.

Фигуру, площадь которой нужно найти, поместим в квадрат. Будем наугад (как говорят математики, случайным образом) бросать точки в этот квадрат. Естественно предположить, что, чем большую

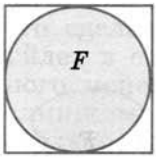


Рис. 3.3

площадь в квадрате занимает фигура, тем чаще в нее будут попадать точки. Представьте себе, например, квадратный дворик и в нем детскую площадку. Во время снегопада количество снежинок, попавших на детскую площадку, пропорционально ее площади. Таким образом, можно сделать допущение: при большом числе точек, наугад выбранных внутри квадрата, доля точек, содержащихся в данной фигуре, приближенно равна отношению площади этой фигуры к площади квадрата.

Определим параметры нашей модели. Обозначим для удобства дальнейших рассуждений данную фигуру буквой F.

Одним из параметров являются сторона a квадрата, содержащего фигуру F, другим — количество точек N, которые мы будем случайным образом выбирать внутри квадрата, третьим параметром является искомая площадь S фигуры F.

Теперь определим связи между параметрами модели. Если через M обозначить число тех наугад выбранных точек, которые содержатся в фигуре F, то площадь фигуры приближенно равна

$$S \approx a^2 \frac{M}{N}.$$

К связям между параметрами модели следует отнести и математические соотношения, позволяющие определить, попала ли выбранная точка в фигуру F.

Эти соотношения будут отличать модели, построенные для разных фигур. Значит, фактически мы описали не одну модель, а, скорее, некоторый способ получения моделей. Для каждой конкретной фигуры будет получаться своя модель.

Давайте рассмотрим математическую модель для приближенного нахождения площади круга радиуса R. Формула площади круга вам, конечно, известна: $S = \pi R^2$. Однако, если вы помните, эту формулу вам сообщили фактически без доказательства. Вот подходящий случай проверить ее с помощью компьютера.

Пусть для определенности $R = 1$. На рисунке 3.3 изображен круг радиуса 1, заключенный в квадрат со стороной $a = 2$. Таким образом, в качестве фигуры F выступает круг единичного радиуса. Выбрать точку — это значит задать ее координаты: числа x и y. Точка принадлежит квадрату, если $-1 \leqslant x \leqslant 1$ и $-1 \leqslant y \leqslant 1$. Если $x^2 + y^2 \leqslant 1$, то точка попадает в круг F, иначе она вне круга. Это и есть математическое соотношение, позволяющее для каждой точки определить, лежит ли она в F.

Математическая модель вычисления площади круга построена, и можно приступать к составлению алгоритма.

Алгоритм Площадь круга
вещ: x, y;
цел: N, M;
{ **Запросить** N;

```
    M := 0;
    Делать от I := 1 до N
      { x := 2*rand(1) - 1;
        y := 2*rand(1) - 1;
        Если x^2 + y^2 ⩽ 1 то { M := M + 1; }
      }                                    (* конец цикла *)
    S := 4*M/N;
    Сообщить "Число выбранных точек:", N, "площадь:", S;
}
```

Вычислительный эксперимент с этим алгоритмом вы проведете, выполняя лабораторную работу 17.

ВОПРОСЫ И ЗАДАНИЯ

1 Составьте алгоритм нахождения с помощью метода Монте-Карло площадей фигур, изображенных на рисунке 3.4.

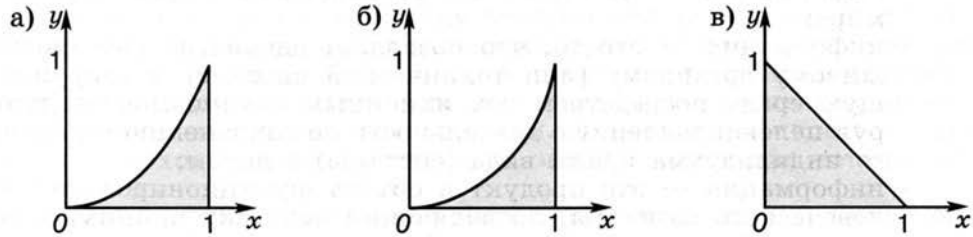


Рис. 3.4 а) кривая задана уравнением $y = x^2$; б) кривая задана уравнением $y = x^3$; в) треугольник

2* (Задача с детективным сюжетом.) Каждый день резидент приходит на встречу со своим агентом в случайный момент времени с 11 до 13 ч и ждет его 15 мин. В свою очередь агент приходит на встречу с резидентом каждый день также в случайный момент времени от 11 до 13 ч и тоже ждет 15 мин. Как часто в течение года встречаются резидент и агент? Составьте математическую модель и алгоритм решения задачи.

3* Ученик пишет квадратные уравнения вида $x^2 + px + q = 0$, выбирая коэффициенты p и q случайным образом из отрезка $[-1; 1]$. С какой частотой он будет писать уравнения, имеющие действительные корни? Составьте математическую модель и алгоритм решения этой задачи.

4 Завод сбрасывает в реку ежедневно случайным образом из-за неисправности очистных сооружений от 0 до 30 кг вредных веществ. За каждый килограмм сверх 15 завод обязан заплатить штраф в размере 10 000 р. Прибыль завода от реализации его продукции 70 000 р. в день. Как часто в течение года штраф будет превышать прибыль? Рентабелен ли такой завод? Составьте математическую модель и алгоритм решения этой задачи.

5 (К решению этой задачи дети до 14 лет не допускаются.) Постройте математическую модель игры в рулетку. Составьте алгоритм, имитирующий эту игру с помощью компьютера.

6 Учитель хочет, чтобы компьютер помог ему проверить знание учащимися таблицы умножения, задавая сомножители случайным образом. Составьте математическую модель и алгоритм опроса учащихся.

7 Перечитайте § 1, 4, 12. Выпишите трактовки понятия информации, которые приведены в этих параграфах.

§ 28 Еще раз об измерении количества информации*

В § 12 мы уже обсуждали понятие количества информации. Что же заставляет нас еще раз вернуться к этому вопросу? Разумеется, то, что понятие информации — одно из самых сложных в современной науке и оно может по-разному трактоваться в разных ее разделах. Напомним точки зрения на трактовку понятия информации, которые были приведены в § 1 (вы их выписали, выполняя задание 7 к предыдущему параграфу, и теперь можете сравнить с тем, что указано нами):

- информация — это отражение разнообразия в существующем мире;
- информация — это то, что позволяет адекватно реагировать живому организму (или технической системе) на окружающую среду посредством тех или иных механизмов, индуцируя целенаправленную деятельность по сохранению отдельного индивидуума и/или вида (системы) в целом;
- информация — это продукт и объект функционирования человеческого сознания, позволяющий человеку принимать решения, обеспечивающие его эффективную деятельность;
- информация — это последовательность сигналов или символов некоторого алфавита, кодирующая сообщение без учета смыслового содержания этого сообщения.

Самой общей точкой зрения на понятие информации является, очевидно, первая из перечисленных; две последние наиболее узкие и, можно сказать, противоположные друг другу: в одной из них смысл сообщения не учитывается вообще, а в другой, наоборот, главным объявляется именно смысл сообщения. Тем не менее оба эти подхода приводят, как показано в § 12, к одной и той же формуле для вычисления количества информации.

А как обстоит дело в случае первого понимания термина «информация»? Хочется надеяться, что и в этом случае мера количества информации останется той же. Чтобы разобраться в этом, уясним, как измерить разнообразие и как связать эту меру с количеством информации.

Начнем с рассмотрения нескольких мысленных экспериментов. Представьте, что вы идете по улице и считаете ворон. Каждая следующая ворона точно такой же окраски, как и предыдущая. Вы

* Данный параграф рекомендуется изучать только учащимся, знакомым с темой «Логарифмы».

уже насчитали N ворон, и все N одного цвета — никакого разнообразия! Хотя и говорят, что белые вороны существуют.

А теперь давайте считать, сколько мужчин и сколько женщин встретилось вам, пока вы идете, считая ворон. Если вы идете в расположении воинской части, то на разнообразие опять рассчитывать не приходится. Зато если вы идете по людной улице большого города, то, скорее всего, вам одинаково часто будут попадаться навстречу мужчины и женщины. Итак, в одном случае — отсутствие разнообразия и потому полная определенность очередного события, в другом — два равновероятных исхода и соответственно полная неопределенность исхода: в очередной раз встретится мужчина или встретится женщина.

В общем виде эту ситуацию можно описать так. Пусть имеется k исходов некоторого опыта. Если какой-то исход намного вероятнее других, то неопределенность того, что произойдет при очередном испытании, мала. Если же все исходы равновероятны, то у нас полная неопределенность, что же случится в очередном испытании.

Обозначим через H величину неопределенности в опыте с k исходами. Хотя мы еще не научились измерять H, уже ясно, что при $k = 1$ неопределенность равна 0. А с ростом k неопределенность возрастает. Если в опыте всего два исхода, то неопределенность меньше, чем если таких исходов три. А неопределенность в случае трех исходов меньше, чем для опыта, в котором четыре исхода, и т. д.

Пусть теперь наш опыт состоит в бросании двух кубиков одновременно. Для одного кубика было 6 равновероятных исходов: выпадение любого числа очков от 1 до 6. Нетрудно подсчитать, что для двух кубиков получается 36 различных равновероятных комбинаций выпадения очков. Все они представлены в следующей таблице (первое число в паре указывает количество очков, выпавших на первом кубике, второе число — количество очков, выпавших на втором):

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)
(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)

Число исходов увеличилось в 6 раз. Следует ли при этом считать, что неопределенность возросла в 6 раз?

Если мы будем подбрасывать монету, то число исходов равно 2. Если подбрасывать две монеты, число исходов становится равным 4. Схема опыта одна и та же: вместо одного предмета подбрасываются два.

Вообще считают, что величина неопределенности удваивается независимо от того, сколько исходов имеет опыт с одним предме-

том. И вообще, если проводится два независимых опыта и в одном из них неопределенность исхода равна H_1, а в другом — H_2, то договариваются считать, что *при одновременном проведении этих двух опытов неопределенность равна сумме неопределенностей в каждом из них*. Легко подсчитать, что если в одном опыте было k равновероятных исходов, а в другом — m, то число исходов при их одновременном проведении равно km. Все вышесказанное означает, что

$$H(km) = H(k) + H(m).$$

Хорошо известно, что таким свойством обладает логарифмическая функция. Оказывается (это умеют доказывать математики), что другой функции здесь и быть не может. Поэтому $H(k) = \log(k)$. Основание логарифма принципиальной роли не играет. Обычно его выбирают равным 2, поскольку за единицу измерения неопределенности принимают неопределенность, имеющуюся при наступлении одного из двух равновероятных событий. Если за единицу неопределенности взять неопределенность, присутствующую при проведении опыта с десятью равновероятными исходами, то получится десятичный логарифм (чтобы $\log(10) = 1$). На самом деле выбор основания логарифма принципиального значения не имеет, поскольку переход к другому основанию, как все помнят из школьного курса математики, приводит всего-навсего к умножению на константу. Это эквивалентно переходу к другой единице измерения, совершенно аналогично тому, как умножение на соответствующую константу происходит при переходе от метров к сантиметрам или от килограммов к граммам.

Итак, мерой неопределенности опыта, имеющего k равновероятных исходов, договорились считать величину

$$H(k) = \log_2 k.$$

Назвали эту величину **энтропией**.

Приведенная нами формула была в 1928 году предложена американским инженером-связистом Хартли и поэтому носит его имя. Удивительное дело — она снова совпала с той формулой для количества информации, которая была приведена в § 18.

Попробуем теперь разобраться, как получается значение энтропии, если исходы опыта неравновероятны. Можно рассуждать так. Если все исходы равновероятны, то каждый исход вносит в неопределенность одну и ту же лепту. Поскольку общая неопределенность равна $\log_2 k$, то вклад в нее одного исхода составляет

$$\frac{1}{k}\log_2 k = -\frac{1}{k}\log_2 \frac{1}{k}.$$

Число $\frac{1}{k}$ — это вероятность каждого исхода, если все они равновероятны. Естественным поэтому представляется, что для неравновероятных исходов вклад одного исхода в общую неопределен-

ность опыта составляет $-p\log_2(p)$, где p — вероятность данного исхода. Если теперь опыт имеет n исходов, а их вероятности равны p_1, p_2, ..., p_n, то мера неопределенности для данного опыта в целом равна

$$H(p_1, p_2, ..., p_n) = -p_1\log_2 p_1 - p_2\log_2 p_2 - ... - p_n\log_2 p_n.$$

Полученную формулу для энтропии опыта с неравновероятными исходами называют формулой Шеннона в честь основоположника теории информации Клода Шеннона, обосновавшего эту формулу в своих первых трудах, посвященных теории передачи информации по каналам связи.

Можно проверить, что, когда p стремится к 0, значение функции $-p\log_2 p$ также стремится к 0. Поэтому можно считать, что в нуле данная функция обращается в 0. Это полностью соответствует представлению, что для невозможного события (а вероятность такого события как раз и равна 0) никакой неопределенности нет, так что и энтропия должна быть нулевой.

Точно так же и для достоверного события, т. е. события, которое всегда происходит, вероятность равна 1, неопределенности никакой нет и энтропия, вычисленная по данной формуле, тоже равна 0.

Кроме того, величина $H(p_1, p_2, ..., p_n)$ достигает своего максимума, только когда все вероятности p_1, p_2, ..., p_n равны между собой. Это тоже отвечает нашим представлениям, что именно в случае равновероятных исходов неопределенность опыта максимальна.

Получив возможность измерять неопределенность того или иного опыта, можно объективно судить о том, в каком случае эта неопределенность выше. Вот один пример.

Пусть из многолетних наблюдений известно, что в данной местности 15 ноября вероятность дождя равна 0,65, вероятность снега — 0,15 и вероятность отсутствия осадков — 0,2. Для 15 июня в той же местности вероятность дождя равна 0,4, вероятность отсутствия осадков — 0,6, снег не наблюдался ни разу. В какой день состояние погоды следует считать более неопределенным?

Подсчитаем энтропию для каждого из этих дней:

$$H_{\text{15 ноября}} = -0{,}65\log_2 0{,}65 - 0{,}15\log_2 0{,}15 - 0{,}2\log_2 0{,}2 \approx 1{,}279;$$
$$H_{\text{15 июня}} = -0{,}4\log_2 0{,}4 - 0{,}6\log_2 0{,}6 \approx 0{,}971.$$

Значит, в ноябре неопределенность состояния погоды выше.

Мы довольно значительное место отвели введению меры неопределенности. Но насколько при этом мы приблизились к главному вопросу: как измерить количество информации? Ясно, что ликвидировать неопределенность — это и значит получить информацию. Следовательно, приведенная выше формула Шеннона показывает, какое максимальное количество информации можно получить для той или иной конкретной системы, если полностью избавиться от

неопределенности. Речь идет именно о максимальном количестве информации, ибо неопределенность может оказаться снятой не полностью, и тогда количество информации окажется меньшим. А можно сказать иначе: эта формула показывает, каким количеством информации нужно располагать, чтобы полностью снять неопределенность.

Итак, если информация понимается как отражение разнообразия, то мерой для ее количества выступает мера неопределенности, которой обладает рассматриваемая в этот момент ситуация. Описывая неопределенность на языке вероятностей, мы приходим к формуле Шеннона. Она-то и является тем обобщением формулы для подсчета количества информации, которое было обещано нами в § 18.

Более того, формула Шеннона объясняет, как может появиться дробное количество информации. Представим себе снова игру «Угадайка», только теперь мы хотим узнать одно число от 1 до 48. Количество вопросов, необходимое для гарантированного отгадывания числа, т. е. полного снятия неопределенности, мы и взяли за количество информации. Что же будет происходить в данном случае? Ясно, что 5 вопросов здесь, вообще говоря, недостаточно, а 6 оптимально заданных вопросов гарантируют угадывание. Но не остановимся на констатации этого факта, а продолжим рассуждения. Пусть при угадывании мы, как и прежде, придерживаемся стратегии деления пополам. Для первых четырех вопросов все пройдет хорошо, а перед пятым вопросом нам будет «противостоять» трехэлементное множество. Поэтому либо мы угадаем число за один вопрос (с вероятностью $\frac{1}{3}$), либо нам потребуется еще два вопроса (с вероятностью $\frac{2}{3}$). Итак, если числа задумываются с равной вероятностью, то при таком первом вопросе для гарантированного угадывания нам с вероятностью $\frac{2}{3}$ потребуется 6 вопросов, а с вероятностью $\frac{1}{3}$ потребуется только 5 вопросов, т. е. при такой стратегии в среднем требуется $5\frac{2}{3}$ вопросов для гарантированного ответа. Однако нет никакой уверенности, что выбранная нами стратегия оптимальна, т. е. что она дает минимальное среднее количество вопросов для гарантированного угадывания одного числа из 96. И она действительно неоптимальна: можно доказать, что при оптимальной стратегии число вопросов в среднем будет равно $\log_2 48 = 5{,}58496\ldots$ Естественно это число и взять за количество информации, которое требуется получить, чтобы определить задуманное число.

Если множество содержит n элементов, то количество информации, необходимое для определения одного элемента, выделенного тем или иным образом, равно $\log_2 n$.

А это совпадает с результатом, подсчитанным по формуле Шеннона в случае, когда все вероятности равны между собой и, значит, они равны $\frac{1}{n}$.

На самом деле предположение о равной вероятности загадывания чисел малоправдоподобно, если разрешить загадывать не любые числа, а числа из ограниченного диапазона. Вероятность загадывания числа больше миллиона во много раз меньше, чем вероятность загадывания числа из первой сотни. Да и для чисел из первой сотни эти вероятности тоже неодинаковы. Если известны вероятности, с которыми загадываются числа, то среднее количество вопросов, которое потребуется для угадывания числа при следовании оптимальной стратегии, как раз определяется формулой Шеннона.

Иногда интересуются тем, сколько информации несет ответ на тот или иной вопрос. Мы уже знаем: если ответ уменьшает неопределенность вдвое, то он несет 1 бит информации. Эта ситуация имеет место в том случае, если вариантов ответа только 2 и они равновероятны. Рассмотрим общую ситуацию. Пусть задумано 10 чисел. Зададим вопрос задумавшему число: «Верно ли, что задумано число 6?» Вероятность ответа «Да», как нетрудно подсчитать, равна 0,1 (мы предполагаем, что у задумавшего нет предпочтения к каким-либо числам, так что все исходы равновероятны). Но если дан ответ «Да», то неопределенность снята полностью, и, значит, нами получена вся информация. Как было сказано, количество этой информации равно $\log_2 10 = -\log_2 0{,}1 \approx \approx 3{,}322$.

Если вероятность некоторого события равна p, то количество информации, которое получено, если это событие произошло, равно $-\log_2 p$.

Например, ответ «Нет» в той же ситуации имеет вероятность 0,9, поэтому он несет всего лишь $-\log_2 0{,}9 \approx 0{,}152$ бит информации. Такое малое количество информации вполне естественно, ведь этот ответ никак нельзя признать неожиданным.

ВОПРОСЫ И ЗАДАНИЯ

1. Какому пониманию термина «информация» соответствует формула Шеннона для измерения количества информации?
2. В приведенном примере вычисления неопределенности состояния погоды мы учитывали вид осадков. Если же интересоваться только тем, есть осадки или нет, то для 15 ноября значение неопределенности будет другим. Найдите его. В какой из этих двух дней состояние погоды с этой точки зрения имеет более высокую неопределенность?

3 В игре «Угадайка» отгадываются числа от 1 до 20, при этом оказалось, что каждое из чисел от 1 до 10 загадывается с вероятностью $\frac{2}{30}$, а каждое из чисел от 11 до 20 — с вероятностью $\frac{1}{30}$.

а) Какое в среднем число вопросов потребуется для угадывания числа при оптимальной стратегии?

б)* Попытайтесь описать оптимальную стратегию в этой игре.

4 Имеется 1000 монет, из которых одна фальшивая (легче других). Придумайте способ нахождения фальшивой монеты за 7 взвешиваний на чашечных весах без гирь. Докажите, что нельзя придумать способ, который гарантирует нахождение фальшивой монеты за 6 взвешиваний.

5 а) Задумано двузначное число. Вам сообщили, что оно больше 54. Сколько информации вы получили?

б) Задумано трехзначное число. Вам сообщили, что оно меньше 325. Сколько информации вы получили?

6 В одном из рассказов К. Чапека описывается случай, как поэт стал свидетелем дорожно-транспортного происшествия, виновник которого умчался с места аварии. Поэт, потрясенный увиденным, написал стихотворение, в котором была такая строка: «О, шея лебедя! О, грудь! О, барабан!»
Полиции он сказал, что эти образы навеяны цифрами номера умчавшейся машины: двойкой, тройкой и нулем. Считая, что номера машин в то время были трехзначными, определите, сколько информации сообщил поэт полиции.

7 К экзамену надо выучить 20 вопросов. Ученик Иванов считает, что 5 вопросов он знает на «отлично», 8 вопросов — на «хорошо», 6 вопросов — на «удовлетворительно» и 1 вопрос не знает совсем. Ученик Петров считает, что он 3 вопроса знает на «отлично», 10 вопросов — на «хорошо», 5 вопросов — на «удовлетворительно» и 2 вопроса не знает совсем. На экзамене придется отвечать на один вопрос.

а) У кого неопределенность с оценкой больше?

б) У кого из этих учеников больше неопределенность того, сдаст ли он экзамен?

ИТОГИ ГЛАВЫ 3

Повторим кратко то основное, что вы узнали, изучив главу 3.

Для изучения процессов, протекающих в живой и неживой природе, обычно используются математические модели, устанавливающие зависимость параметров этих процессов от времени. При преобразовании этих моделей в компьютерные, как правило, используется язык программирования или электронная таблица.

Адекватность модели означает, что цель моделирования достигнута и при решении с ее помощью задач определенного класса получается удовлетворительный ответ. В частности, адекватность модели означает, что при ее построении действительно бы-

ли учтены все существенные факторы. Адекватность модели можно проверить только ее согласованностью с общими теоретическими положениями и экспериментальными данными. Поэтому на заключительной стадии построения модели всегда проводят эксперимент, подтверждающий или опровергающий адекватность созданной модели. Этот эксперимент может быть натурным или компьютерным. В таком эксперименте обычно определяются границы адекватности модели, т. е. та область изменения параметров модели, в пределах которой модель можно считать адекватной.

Среди процессов особую группу составляют процессы, для которых существенным является фактор случайности. Случайное событие, происходящее в таком процессе, обычно характеризуют параметром, который называется вероятностью данного события. Вероятность события — это число, к которому стремится значение относительной частоты наступления события в серии опытов, когда количество опытов неограниченно возрастает. Для построения моделей случайных процессов — такие модели называются вероятностными — используют датчики случайных чисел.

Датчики случайных чисел, реализованные в виде алгоритма для компьютера, генерируют псевдослучайную последовательность чисел. Она не является случайной в строгом смысле этого слова, поскольку закономерность образования членов этой последовательности изначально задана. Однако для достаточно большого числа своих членов такая последовательность обладает необходимыми (с точки зрения обеспечения адекватности вероятностной модели) свойствами последовательности случайных чисел. Во многих процессах встречаются последовательности равномерно распределенных случайных величин — это означает, что вероятности появления у всех членов последовательности одинаковы.

В языках программирования датчик случайных чисел, как правило, реализован в виде одного оператора, присваивающего переменной случайное значение. Для инициализации работы датчика случайных чисел требуется начальное значение, которое задается с помощью специального оператора.

Вероятностные модели могут использоваться и в математике, например, для вычисления площади геометрической фигуры или объема геометрического тела. Такие методы вычислений получили название методов Монте-Карло.

Количество информации, которое можно получить в опыте, имеющем n исходов с вероятностями p_1, p_2, ..., p_n, вычисляется по формуле Шеннона:

$$H(p_1, p_2, ..., p_n) = -p_1\log_2 p_1 - p_2\log_2 p_2 - ... - p_n\log_2 p_n.$$

Если при этом вероятности всех n исходов одинаковы (равномерное распределение), то количество информации равно $\log_2 n$ (формула Хартли).

ПРОВЕРЬ СЕБЯ

Часть 1. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и номер правильного ответа. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

А1. Свойство модели соответствовать теоретическим воззрениям и наблюдениям практики называется:

1) информативностью; 2) адекватностью; 3) системностью;

4) среди вариантов, указанных в пунктах 1—3, нет правильного.

А2. Модель, при построении которой в качестве существенного учитывается фактор случайности, называется:

1) адекватной; 2) вероятностной; 3) системной;

4) среди вариантов, указанных в пунктах 1—3, нет правильного.

А3. Некто задумал натуральное число, не превосходящее 1000, а затем сообщил: «Задуманное мною число является степенью числа 2 с целым показателем». Количество бит информации, содержащейся в этом сообщении, равно:

1) $\log_2 1000$; 2) 9; 3) $\log_2 100$; 4) $\log_2 10$.

А4. Студент Иванов сдает экзамен по аппаратным средствам информатизации, а студент Петров — экзамен по информационным технологиям. Иванову надо выучить 20 вопросов, а Петрову — 25. Иванов считает, что он 4 вопроса знает на «отлично», 9 вопросов — на «хорошо», 5 вопросов — на «удовлетворительно» и 2 вопроса не знает совсем. Петров считает, что он 5 вопросов знает на «отлично», 10 вопросов — на «хорошо», 7 вопросов — на «удовлетворительно» и 3 вопроса не знает совсем. На экзамене придется отвечать на один вопрос. Неопределенность с оценкой:

1) у Петрова больше, чем у Иванова;

2) у Петрова меньше, чем у Иванова;

3) одна и та же у Петрова и у Иванова;

4) у Петрова и у Иванова нельзя сравнивать, поскольку они сдают разные экзамены.

А5. Следующий алгоритм предназначен для нахождения площади фигуры методом Монте-Карло, в нем через rand(1) обозначен датчик случайных чисел, вырабатывающий равномерно распределенную последовательность чисел из промежутка [0; 1).

Алгоритм
цел: M, N; **вещ:** X, Y;
{ $N := 0$;
Делать от $M := 1$ **до** 100000

```
{ X := 2*rand(1) − 1;
  Y := rand(1);
  Если (|X| + Y ⩽ 1) то { N := N + 1; }
}                                        (* конец цикла *)
Сообщить N / 50000;
}
```

Площадь этой фигуры равна:

1) 0,5; 2) 1; 3) 1,5;

4) другому, нежели в пунктах 1—3, числу.

А6. Следующий алгоритм предназначен для нахождения площади фигуры методом Монте-Карло, в нем через rand(1) обозначен датчик случайных чисел, вырабатывающий равномерно распределенную последовательность чисел из промежутка [0; 1).

```
Алгоритм
цел: M, N; вещ: X, Y;
{ N := 0;
  Делать от M := 1 до 100000
  { X := rand(1);
    Y := 2*rand(1) − 1;
    Если (X^2 + Y^2 ⩽ 1) то { N := N + 1; }
  }                                      (* конец цикла *)
  Сообщить N/50000;
}
```

Этой фигурой является:

1) треугольник; 2) круг; 3) полукруг; 4) ромб.

А7. Вами изучались следующие инструменты информационных технологий:

а) текстовый редактор;
б) графический редактор;
в) электронная таблица;
г) СУБД;
д) язык программирования.

При построении компьютерной модели на основе преобразования математической модели используются инструменты:

1) *а* и *в*; 2) *в* и *г*; 3) *в* и *д*; 4) *б* и *д*.

Часть 2. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и ответ в виде последовательности символов. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

В1. Выберите из списка необходимые действия, установите правильную последовательность их выполнения и запишите номера этих действий в порядке полученной последовательности.

Для построения информационной модели надо:

1) определить свойства объекта, существенные для решения задачи;
2) выбрать материал, из которого будет изготовляться модель;
3) сформулировать цель моделирования;
4) описать выделенные свойства с помощью параметров;
5) определить, какая технология будет использоваться для реализации модели;
6) указать связи между параметрами;
7) убедиться, что модель адекватна.

В2. Выберите номера правильных ответов и запишите их в порядке возрастания без пробелов и знаков препинания между ними.
Для алгоритмической модели, представленной компьютерной программой, параметры модели могут быть представлены:
1) текстом; 2) числами; 3) переменными;
4) ничем из перечисленного в пунктах 1—3.

В3. В следующем предложении требуется вместо <процесса> вставить словосочетание из левого столбца таблицы 3.8, а вместо <список параметров> — нужные слова из правого столбца той же таблицы.
Для информационной модели <процесса> параметрами являются <список параметров>.
Ответ к этому заданию записывается в виде 1АБВ, 2ВК и т. д.

Таблица 3.8

Процесс	Параметры
1) равномерное движение; 2) движение под действием силы (второй закон Ньютона); 3) упругая деформация (закон Гука); 4) работа магазина; 5) работа театра; 6) равноускоренное движение	А) скорость; Б) сила; В) ускорение; Г) путь; Д) время; Е) длина; Ж) модуль упругости; З) список товаров; И) количество товара; К) репертуар; Л) список цен; М) список работников; Н) масса

В4. Выберите номера правильных ответов и запишите их в порядке возрастания без пробелов и знаков препинания между ними.
Для описания нижеперечисленных физических явлений используется математическая модель в виде прямо пропорциональной зависимости между параметрами:

1) равномерное прямолинейное движение (параметры: пройденный путь, время);

2) работа рычага (параметры: длина рычага, сила);

3) деформация пружины под действием силы (параметры: величина силы, изменение линейных размеров);

4) равномерное движение по окружности (параметры: центробежная сила, угловая скорость);

5) изменение размеров тела при нагревании (параметры: изменение температуры, изменение линейных размеров);

6) среди вариантов, указанных в пунктах 1—5, нет правильного.

B5. Последовательность чисел, вырабатываемая датчиком случайных чисел, называется _____.

B6. Для преобразования непрерывной математической модели в компьютерную модель производится ее _____.



Логико-математические модели

Уже достаточно сказано о возможностях компьютеров и применяемых информационных технологий, чтобы каждый понял, какую неоценимую помощь может оказывать компьютер в предоставлении и обработке информации. Однако за человеком остается право (и обязанность) делать выводы на основе предоставляемой информации и принимать соответствующие решения. Иногда мы говорим, что решение принимает компьютер. Например, если компьютер, управляя каким-либо автоматом, получает от датчиков этого автомата информацию об опасности его разрушения в тех условиях, в которых он оказался, то компьютер может «принять решение» об эвакуации автомата из этой обстановки. Но каждому ясно, что на самом деле это решение было заранее запрограммировано человеком и компьютер безоговорочно следует инструкции.

Принятие решения — это всегда прерогатива человека. Во-первых, потому, что любая информация лишь модельно отражает реальную действительность, и, следовательно, принимаемое компьютером решение может оказаться неадекватным сложившейся ситуации. В вышерассмотренном примере вполне вероятна ситуация, что автомат должен погибнуть, чтобы предотвратить большее несчастье. Во-вторых, передать компьютеру право принимать решения — значит вверить судьбу человечества электронике. И хотя сегодня бунт роботов — это все еще удел фантастических произведений, в их основе лежит именно предположение о предоставлении компьютеру права принимать решения наравне с человеком.

Тем не менее помощь компьютера в решении человеком информационных задач намного бы выросла, если бы компьютер «научился» предлагать человеку на выбор обоснованные варианты возможных решений. Но для этого надо научить компьютер делать выводы из имеющейся информации. Иными словами, надо в компьютере смоделировать процесс рассуждений, которые проводит человек. О тех подходах к решению данной проблемы, которые сегодня разрабатываются в информатике, и пойдет речь в этой главе.

§29 Понятие моделей искусственного интеллекта

Первым, кто предпринял удачную попытку построить модель человеческих рассуждений, был, по мнению историков науки, древнегреческий ученый Аристотель. Именно он сформулировал первые законы рассуждений, заложив основы новой науки, названной им **логикой**. В дальнейшем вклад в эту науку вносили психологи и философы, лингвисты и математики. У этой науки появились разные направления исследований, она разделилась на ряд областей, одна из которых называется **формальной** или **математической логикой**. Более того, не только специализировались исследования внутри логики, но и расширялись сферы исследований на всю интеллектуальную деятельность человека. Как человек распознает предметы и явления, как происходит обучение и самообучение, как вырабатываются стратегии в реальной жизни или играх, каковы механизмы стихосложения и сочинения музыки — все это стало предметом исследований ученых. А модели, создаваемые в результате таких исследований, стали называть моделями **искусственного интеллекта**.

Для описания моделей искусственного интеллекта используются разнообразные, иногда довольно сложные средства. В основе многих из них лежат графы, о которых мы кратко вспомнили в § 3. Тем не менее любая модель искусственного интеллекта как одна из разновидностей информационных моделей обязательно содержит набор параметров, описывающих существенные факторы. Поскольку модель искусственного интеллекта направлена на принятие того или иного решения, то в качестве связей между параметрами, описывающими исходные данные, и результатом выступают так называемые **правила вывода**. Их суть в том, что они указывают, какое решение должно быть рекомендовано в рамках данной модели при тех или иных значениях исходных параметров.

К примеру, модель, обеспечивающая медицинскую диагностику, учитывает такие параметры, как температура пациента, кашель, сыпь, жалобы на боль в тех или иных частях тела и т. п. Правило вывода по значениям этих исходных параметров должно предоставить название наиболее вероятного заболевания (или список таковых, расположенных, например, в порядке убывания вероятности). Кроме того, правило вывода может предусматривать выдачу рекомендации провести еще какие-либо анализы или задать дополнительные вопросы пациенту для уточнения диагноза.

Модель распознавания графических образов, например рукописных букв и цифр (а сейчас такие системы создаются для автоматизации ввода в компьютер рукописных текстов), содержит уже иные параметры и иные правила вывода о том, что именно представляет собой данный графический образ.

Модель, описывающая процессы восприятия человеком музыки, оперирует с параметрами, характеризующими метроритмичес-

кую, ладогармоническую и другие структуры музыкального произведения, а также с параметрами психофизиологических реакций на эти структуры. Правила вывода здесь могут указывать на то, в какой мере и как данная музыка воспринимается тем или иным человеком.

Правила вывода, применяемые в конкретной системе искусственного интеллекта, могут иметь разную форму, и нередко сами такие системы классифицируются именно по тому признаку, какую форму имеют используемые правила вывода. Рассматривать подробно различные системы искусственного интеллекта мы не будем — слишком широка и разнообразна ныне эта область информатики. Но с одной разновидностью, а именно с **экспертными системами**, которые все активнее применяются в научных исследованиях и технологических процессах, мы вас познакомим и даже предоставим возможность самостоятельно сконструировать несложную экспертную систему. Речь об этом пойдет в § 38, а пока вам предстоит познакомиться с теми средствами, которые для этого будут использоваться.

ВОПРОСЫ И ЗАДАНИЯ

1. Какой признак выделяет модели искусственного интеллекта из множества информационных моделей в целом? Попытайтесь привести примеры моделей искусственного интеллекта, помимо тех, которые указаны в объяснительном тексте параграфа.
2. Почему право принятия решения должно быть оставлено за человеком? Приведите свои аргументы в пользу этого тезиса.

§ 30 Элементы логики высказываний

Психологи давно установили, что мыслительная деятельность человека всегда осуществляется посредством какого-либо языка. Использование языка придает мыслительной деятельности форму рассуждений. Что такое рассуждение? Рассуждение — это последовательность вопросов, задаваемых самому себе или собеседнику, и ответов на них.

Вопросы могут быть разного типа. Можно спросить: «Сколько существует простых чисел?» Ответом служит утверждение: «Простых чисел бесконечно много». Можно спросить: «Как найти тысячное простое число?» Ответом будет алгоритм, позволяющий вычислить требуемое простое число. Напомним (см. § 11), что информацию, содержащуюся в ответе на первый вопрос, мы относим к декларативному типу, а информацию, содержащуюся в ответе на второй вопрос, — к процедурному.

Свойства алгоритмов вы изучили в базовом курсе информатики, а мы еще раз их обсудили в § 7. Среди них есть, например, ре-

зультативность и правильность. А вот информацию, представленную декларативно, можно, пожалуй, оценить только с одной позиции — истинна она или ложна.

Повествовательное предложение, в отношении которого имеет смысл говорить о его истинности или ложности, называется **высказыванием**. Такое определение высказыванию дал Аристотель. Рассмотрим следующие предложения.

1. Число $\sqrt{2}$ иррационально.
2. Число $\sqrt{2}$ рационально.
3. Любое натуральное число x рационально.
4. Любое число x рационально.
5. Существует число x, которое иррационально.
6. Число x рационально.
7. Если число x иррационально, то число $x + 1$ тоже иррационально.
8. Число называется рациональным, если оно равно отношению целого числа к натуральному.
9. Верно ли, что число $\sqrt{2}$ иррационально?
10. Докажите, что число $\sqrt{2}$ иррационально.

Первые пять предложений, очевидно, являются высказываниями, причем 1, 3 и 5-е истинны, а 2-е и 4-е ложны. Два последних предложения не являются высказываниями, поскольку они вовсе не повествовательные: одно из них вопросительное, а другое побудительное. Предложение 8 не является высказыванием, поскольку о его истинности говорить бессмысленно: оно лишь определяет новое понятие через ранее введенные.

В повествовательном предложении 6 истинность заключенной в нем информации зависит от значения переменной x: при одних ее значениях утверждение окажется истинным, при других — ложным. Так что это предложение нельзя отнести к высказываниям. Мы рассмотрим подобные утверждения в § 36. Точно так же переменную x содержит предложение 7, поэтому и его не следует относить к высказываниям. Но если задуматься о смысле этого предложения, то станет ясно, что оно истинно при любом значении переменной x.

Пример предложения 7 показывает, что данное Аристотелем объяснение, что такое высказывание, не является, строго говоря, определением. Более того, оно фактически перекладывает на плечи других научных дисциплин — математики, физики, химии, биологии, истории (список вы легко продолжите сами) — саму проблему получения ответа на вопрос, будет ли истинным то или иное утверждение. И вот если ответ на такой вопрос будет получен, то это утверждение будет называться высказыванием.

Впрочем, для некоторых утверждений ответ об их истинности или ложности нельзя получить никакими средствами. Вот одно из

таких утверждений: «Это предложение ложно». Если предположить, что само это утверждение истинно, то оно обязано быть ложным. Если же считать, что утверждение ложно, то оно обязано быть истинным. Выявление и недопущение в рассуждениях подобных внутренне противоречивых утверждений тоже одна из задач логики.

Оставляя другим наукам право отвечать на вопрос об истинности конкретного утверждения, логика интересуется, как из набора истинных утверждений можно получать новые истинные утверждения. Это означает, что логика изучает такие операции над высказываниями, в результате применения которых снова получается высказывание. При этом вовсе не требуется вникать в смысл высказываний, над которыми производятся операции. Математическая логика предлагает формализованный язык для описания этих операций, что позволяет построить формальную модель человеческих рассуждений.

С некоторыми логическими операциями вы наверняка знакомы еще по базовому курсу информатики. Это прежде всего операция конъюнкции (по-другому, операция ***и***), дизъюнкции (по-другому, операция ***или***) и операция отрицания (по-другому, операция ***не***). Но есть и другие операции, они перечислены в таблице 4.1.

Таблица 4.1

Логическая операция	**Обозначение**	**Смысл в обычном языке**
Конъюнкция, логическое умножение	&, ·, ∧	Союзы *и*, *а*, *но*
Дизъюнкция, логическое сложение	∨, +	Союз *или*
Отрицание, инверсия	¯, ¬	Частица *не*
Разделительная дизъюнкция, исключающее ***или***, сложение по модулю 2	⊕, Δ	Оборот *или только..., или только...*
Импликация, следование	→, ⇒	Оборот *если..., то...*
Равносильность, равнозначность, эквиваленция	↔, ⇔, ≡, ~	Обороты *тогда и только тогда, необходимо и достаточно*

В названиях и обозначениях логических операций, приведенных в таблице 4.1, на первом месте указаны те, которые будут использоваться в нашем учебнике. В другой литературе по математической логике употребляются и другие из указанных названий и обозначений.

Во всех логических операциях, кроме операции отрицания, участвуют два аргумента. Поэтому применение, например, конъюнкции к высказываниям X и Y обычно записывают как X & Y, а при-

менение импликации к тем же высказыванием как $X \to Y$. Отрицание высказывания X записывают в виде $\overline{X}$.

Значения логических операций задаются, как вы знаете, с помощью **таблиц истинности**. В них для всевозможных комбинаций значений аргументов записывается результат применения операции. Для всех операций одновременно эти таблицы собраны в таблице 4.2; в ней значения аргументов и результат применения операции обозначены буквами «И» (Истина) и «Л» (Ложь).

Таблица 4.2

X	Y	$X \& Y$	$X \vee Y$	$\overline{X}$	$X \oplus Y$	$X \to Y$	$X \leftrightarrow Y$
И	И	И	И	Л	Л	И	И
И	Л	Л	И	Л	И	Л	Л
Л	И	Л	И	И	И	И	Л
Л	Л	Л	Л	И	Л	И	И

Как видно из таблицы, истинность высказывания, полученного с применением дизъюнкции, имеет место, когда истинно либо одно высказывание, либо другое, либо оба одновременно. К примеру, истинность высказывания «Идет дождь ***или*** дует ветер» означает, что на улице имеет место одна из трех ситуаций: идет дождь и нет ветра; нет дождя, но дует ветер; одновременно идет дождь и дует ветер. Поэтому, записывая данную фразу средствами математической логики, естественно представить ее в виде $X \vee Y$, где X — это высказывание «Идет дождь», а Y — высказывание «Дует ветер».

А вот высказывание «Петя сидит на уроке физики ***или*** Петя сидит на уроке истории» ложно, когда оба высказывания, из которых оно составлено, истинны, — не может Петя одновременно быть и на уроке физики, и на уроке истории. Здесь по смыслу применена операция исключающего *или*. Поэтому, переводя эту фразу на язык математической логики, естественно воспользоваться операцией $\oplus$ и соответственно записать $X \oplus Y$, где X — это высказывание «Петя сидит на уроке физики», а Y — высказывание «Петя сидит на уроке истории».

Возможно, вас удивила таблица истинности для операции следования. Многим кажется, что утверждение «Если X, то Y» истинно в том и только в том случае, когда X и Y одновременно истинны, т. е. совпадает с конъюнкцией этих высказываний. Но давайте подумаем, когда ложны эти высказывания. Легко понять, что конъюнкция X и Y ложна тогда и только тогда, когда ложно хотя бы одно из высказываний X или Y. А ложность высказывания «Если X, то Y» означает, что, хотя высказывание X истинно, высказывание Y ложно. Отсюда и вытекает то формальное определение импликации, которое приведено в таблице 4.2. В частности, высказывание «Если $2 \times 2 = 5$, то $2 \times 2 = 4$» истинно. Как, впрочем, истинно и высказывание «Если $2 \times 2 = 5$, то $2 \times 2 = 3$». Неред-

ко отмеченное свойство импликации формулируют так: из истины следует истина, а из лжи — что угодно.

Обычно с помощью логических операций стараются свести любое высказывание к таким высказываниям, в которые уже нельзя выделить другие высказывания. Такие высказывания, никакая часть которых высказыванием не является, называются простыми.

ВОПРОСЫ И ЗАДАНИЯ

1 Из приведенных ниже предложений выберите высказывания и обоснуйте свой выбор:

а) Выхожу один я на дорогу.
б) Я спросил у ясеня: «Где моя любимая?»
в) Зачем вы, девушки, красивых любите?
г) Если у вас нет собаки, ее не отравит сосед.
д) Давайте восклицать, друг другом восхищаться.
е) Не пой, красавица, при мне ты песен Грузии печальной.
ж) Вот кто-то с горочки спустился.
з) Ничего на свете лучше нету, чем бродить друзьям по белу свету.
и) Умом Россию не понять.
к) Но кто-то камень положил ему в протянутую руку.

2 Для приведенных ниже высказываний укажите, какие из них истинны и какие ложны:

а) Число 3 является делителем любого числа, у которого сумма цифр равна 6.
б) Некоторые млекопитающие не живут на суше.
в) Никто не может объять необъятное.
г) Демосфен утверждал: «В одну реку нельзя войти дважды».
д) В каждом году есть месяц, в котором 13-е число приходится на пятницу.

3 Приведите по два примера истинных и ложных высказываний из информатики, физики, математики, биологии, истории, политической жизни.

4* Для приведенных ниже высказываний попытайтесь указать, какие из них истинны и какие ложны:

а) Натуральных чисел бесконечно много.
б) Если диагональ выпуклого четырехугольника разбивает его на два равных треугольника, то этот четырехугольник — параллелограмм.
в) Серединный перпендикуляр к отрезку — это множество точек, равноудаленных от концов данного отрезка.
г) В десятичной записи числа π встречается 100 раз подряд цифра 9.
д) В записи числа $\frac{1}{7}$ в виде бесконечной десятичной дроби не встречается цифра 9.
е) Каждое натуральное число либо простое, либо произведение простых чисел.
ж) Для всех действительных значений x выполняется неравенство $\frac{1}{1+x^2} \leqslant 1$.
з) Для всех действительных значений x выполняется неравенство $\frac{1}{1+x} \leqslant 1$.
и) Любое натуральное число, у которого сумма цифр равна 2, не делится на 7.

Для всех ли высказываний вам удалось выполнить задание? Почему тем не менее можно утверждать, что все приведенные в пунктах *а—з* предложения являются высказываниями?

5 В приведенных ниже высказываниях выделите два простых, обозначьте их буквами и запишите все высказывания с помощью этих обозначений и логических операций:
а) Петя и Коля идут гулять.
б) Петя идет гулять, а Коля гулять не идет.
в) Если Коля идет гулять, то Петя гулять не идет.

6 В приведенных ниже высказываниях выделите два простых, обозначьте их буквами и запишите все высказывания с помощью этих обозначений и логических операций:
а) число 1234 четно и делится на 3;
б) число 1234 нечетно и делится на 3;
в) число 1234 четно или не делится на 3;
г) число 1234 нечетно или не делится на 3.

7 Укажите, какую, на ваш взгляд, дизъюнкцию — обычную или разделительную — надо употребить при записи следующих высказываний на языке математической логики:
а) На деревьях распустились листья или раскрылись бутоны.
б) На улице замерз ртутный градусник или от жары пересохла речка.
в) На небе светит Солнце или видна Луна.

8 Определите, какие из приведенных ниже импликаций истинны:
а) Если число 1357 нечетно, то оно делится на 3.
б) Если число 1357 четно, то оно делится на 3.
в) Если число 1357 нечетно, то оно не делится на 3.
г) Если число 1357 делится на 3, то оно нечетно.
д) Если число 1357 не делится на 3, то оно нечетно.
е) Если число 1357 не делится на 3, то оно четно.

9 Составьте таблицы истинности для высказываний:
а) $\overline{Y} \to \overline{X}$;
б) $X \to (Y \to X)$;
в) $\overline{X \vee Y} \to (X \,\&\, Y)$;
г) $X \,\&\, Z \to (Y \vee Z \to X)$.

§31 Законы алгебры высказываний

Об алгебре как разделе математики вы узнали в 7 классе. Вы привыкли к этому слову в расписании уроков, и, возможно, многие из вас не сразу ответят на вопрос, что такое алгебра. В математике алгеброй называется дисциплина, изучающая свойства операций на множествах. Школьная алгебра изучает множество действительных чисел и операции сложения, умножения, вычитания, деления, возведения в степень, которые над этими числами можно выполнять. Некоторые свойства операций вы знаете еще по начальной школе, например переместительный закон сложения: от перемены мест слагаемых сумма не меняется. Справедлив аналогичный закон и для умножения чисел.

Одно из важных изобретений человека, которое легло в основу алгебры, — это введение буквенной символики для записи

свойств операций. К примеру, тот же переместительный закон сложения записывается с помощью букв совсем просто: $a + b = b + a$. Такие равенства, которые верны при любых значениях букв, называются, как вы помните, **тождествами**. По мере изучения алгебры тождеств появлялось все больше и больше. Достоинством тождеств является то, что, подставляя одни тождества в другие, можно получать новые тождества. Такую процедуру называют **тождественными преобразованиями**.

Математическая логика, а точнее, тот раздел, о котором мы сейчас рассказываем, имеет дело с множеством высказываний. В предыдущем параграфе мы ввели несколько операций над высказываниями. Следовательно, существует алгебра высказываний — раздел математической логики, изучающий свойства операций над высказываниями. Прежде всего нас будут интересовать те свойства, которые записываются как тождества. В нашем случае буквы будут обозначать произвольные высказывания, а знак равенства будет по-прежнему выражать тот факт, что значение левой части равенства совпадает со значением правой части равенства, какие бы высказывания мы в них ни подставляли (разумеется, вместо одинаковых букв мы должны подставлять одно и то же высказывание, хотя вовсе не обязательно, чтобы разные буквы заменялись разными высказываниями). В математической логике такие тождественно равные высказывания принято называть **равносильными**. Сами высказывания, в которых фигурируют буквы, обозначающие произвольные высказывания и соединенные знаками логических операций, называют **формулами**. Буквы, входящие в такие формулы, называют **логическими переменными**. Формулы называют **тождественно равными** или **равносильными**, если равносильны представленные ими высказывания.

В алгебре действительных чисел доказательство тождеств подчас является довольно трудным делом, требующим определенной изобретательности. В алгебре высказываний доказательство тождества — процесс несложный, хотя и может оказаться весьма трудоемким. Причина здесь в том, что действительных чисел бесконечно много и все не перепробуешь, подставляя их вместо переменных. Что касается алгебры высказываний, то, как вы знаете, значение выражения, составленного с помощью логических операций из других высказываний, зависит только от истинности и ложности входящих в его состав высказываний. Поэтому, составив таблицы истинности для двух таких высказываний, мы можем легко убедиться, равносильны они или нет. Убедимся, к примеру, в равносильности высказываний $X \to Y$ и $\overline{X} \vee Y$ (табл. 4.3).

Таблица 4.3

X	Y	$\overline{X}$	$X \to Y$	$\overline{X} \vee Y$
И	И	Л	И	И
И	Л	Л	Л	Л
Л	И	И	И	И
Л	Л	И	И	И

Всем видно, что столбцы для $X \to Y$ и $\overline{X} \vee Y$ совпали. Значит, можно записать: $X \to Y = \overline{X} \vee Y$.

Приведем список основных тождеств алгебры высказываний.

1. $X \,\&\, Y = Y \,\&\, X$; 2. $X \vee Y = Y \vee X$;	(коммутативность, или переместительные законы)
3. $(X \,\&\, Y) \,\&\, Z = X \,\&\, (Y \,\&\, Z)$; 4. $(X \vee Y) \vee Z = X \vee (Y \vee Z)$;	(ассоциативность, или сочетательные законы)
5. $(X \,\&\, Y) \vee Z = (X \vee Z) \,\&\, (Y \vee Z)$; 6. $(X \vee Y) \,\&\, Z = (X \,\&\, Z) \vee (Y \,\&\, Z)$;	(дистрибутивность, или распределительные законы)
7. $X \,\&\, X = X$; 8. $X \vee X = X$;	(идемпотентность)
9. X & Л = Л; 10. $X \vee$ И = И; 11. X & И = X; 12. $X \vee$ Л = X;	
13. $X \,\&\, \overline{X} =$ Л; 14. $X \vee \overline{X} =$ И;	(законы исключенного третьего)
15. $\overline{X \,\&\, Y} = \overline{X} \vee \overline{Y}$; 16. $\overline{X \vee Y} = \overline{X} \,\&\, \overline{Y}$;	(законы де Моргана)
17. $(X \,\&\, Y) \vee Y = Y$; 18. $(X \vee Y) \,\&\, Y = Y$;	(законы поглощения)
19. $\overline{\overline{X}} = X$;	(закон двойного отрицания)
20. $X \to Y = \overline{X} \vee Y$;	
21. $X \leftrightarrow Y = (\overline{X} \vee Y) \,\&\, (X \vee \overline{Y}) = (X \,\&\, Y) \vee (\overline{X} \,\&\, \overline{Y})$;	
22. $X \oplus Y = (X \,\&\, \overline{Y}) \vee (\overline{X} \,\&\, Y)$.	

Мы выше проверили равенство 20. Остальные равенства можно проверить тем же способом — составить таблицы истинности. Для равенств 1—6 и законов де Моргана вы проведете такую проверку, выполнив задания 3 и 4.

Приведенные выше законы алгебры логики обычно используют для преобразования одних формул в другие, им равносильные. Вот пример:

$$\overline{X \to Y} \,\&\, (X \vee \overline{Y}) = \overline{(X \,\&\, Y) \vee (\overline{X} \,\&\, \overline{Y})} \,\&\, (X \,\&\, \overline{Y}) =$$
$$= \overline{(X \,\&\, Y)} \,\&\, \overline{(\overline{X} \,\&\, \overline{Y})} \,\&\, (X \,\&\, \overline{Y}) = (\overline{X} \vee \overline{Y}) \,\&\, (\overline{\overline{X}} \vee \overline{\overline{Y}}) \,\&\, (X \vee \overline{Y}) =$$
$$= (\overline{X} \vee \overline{Y}) \,\&\, (X \vee Y) \,\&\, (X \vee \overline{Y}) = (\overline{X} \vee \overline{Y}) \,\&\, (X \,\&\, (Y \vee \overline{Y})) =$$
$$= (\overline{X} \vee \overline{Y}) \,\&\, (X \,\&\, \text{И}) = (\overline{X} \vee \overline{Y}) \,\&\, X = (\overline{X} \,\&\, X) \vee (\overline{Y} \,\&\, X) =$$
$$= \text{Л} \vee (\overline{Y} \,\&\, X) = \overline{Y} \,\&\, X.$$

Первое равенство в этой цепочке объясняется применением формулы 21, второе и третье равенства имеют место согласно законам де Моргана, четвертое получается применением формулы двойного отрицания, пятое равенство — это последовательное применение законов коммутативности и дистрибутивности, шестое равенство имеет место в силу формулы 14, седьмое — в силу формулы 11, восьмое — это опять применение дистрибутивного закона, девятое равенство верно в силу формулы 13 и, наконец, десятое равенство объясняется формулой 12.

Приведенные 22 тождества вовсе не являются независимыми друг от друга — можно одни из них получать, используя другие тождества этого же списка. Вот как, например, выводится закон поглощения (тождество 17) из тождеств 11, 1, 6, 10:

$$(X \,\&\, Y) \vee Y = (X \,\&\, Y) \vee (Y \,\&\, \text{И}) = (X \,\&\, Y) \vee (\text{И} \,\&\, Y) =$$
$$= (X \vee \text{И}) \,\&\, Y = \text{И} \,\&\, Y = Y.$$

Закон ассоциативности для операции конъюнкции позволяет не писать скобки, если эта операция применяется подряд к нескольким переменным. Например, вместо $(X \,\&\, Y) \,\&\, ((Z \,\&\, U) \,\&\, V)$ можно писать просто $X \,\&\, Y \,\&\, Z \,\&\, U \,\&\, V$. Именно так мы и будем делать. Аналогично будем записывать выражения с операцией $\vee$. Если же в записи высказывания встречаются разные операции — отрицание, конъюнкция, дизъюнкция, импликация, то договоримся, что они выполняются в указанном порядке: сначала выполняется отрицание, затем конъюнкция, а уже потом дизъюнкция. Если же порядок выполнения операций надо изменить, то применяют скобки. Поэтому выражения $(X \,\&\, Y) \vee (Z \,\&\, U)$ и $X \,\&\, Y \vee Z \,\&\, U$ совпадают, но отличаются от $X \,\&\, (Y \vee Z) \,\&\, U$. Операции $\leftrightarrow$ и $\oplus$ имеют самый низкий приоритет.

Формула называется **тождественно истинной** или **тавтологией**, если она принимает значение Истина при любых значениях входящих в нее переменных. Примером простейшей тавтологии является формула $X \to X$.

Из определения операции $\leftrightarrow$ следует, что формулы F_1 и F_2 равносильны тогда и только тогда, когда формула $F_1 \leftrightarrow F_2$ является тавтологией.

Математическая логика лишь моделирует логику человеческого рассуждения. К примеру, высказывания $X \,\&\, Y$ и $Y \,\&\, X$ математическая логика признает равносильными. И действительно, высказывания «Идет дождь, и дует ветер» и «Дует ветер, и идет дождь» в смысловом содержании идентичны друг другу. Но срав-

ните высказывания «Она вошла в зал, и заиграла музыка» и «Заиграла музыка, и она вошла в зал», и вы почувствуете тонкую, но вполне уловимую разницу, идущую от человеческого восприятия союза «и» как некой упорядоченности событий по времени.

ВОПРОСЫ И ЗАДАНИЯ

1. Какие высказывания, содержащие переменные, обозначающие высказывания, называются равносильными?
2. Составьте таблицы истинности и выясните, равносильны ли следующие высказывания:
   а) $X \to Y$ и $\overline{Y} \to \overline{X}$; б) $X \to Y$ и $X \,\&\, Z \to Y \,\&\, Z$;
   в) $(X \oplus Y) \,\&\, Z$ и $X \,\&\, Z \oplus Y \,\&\, Z$; г) $X \to (Y \to Z)$ и $(X \to Y) \to Z$.
3. Составив таблицы истинности, проверьте равенства 1—6, приведенные в объяснительном тексте параграфа.
4. Проверьте законы де Моргана, составив соответствующие таблицы истинности.
5. Упростите следующие формулы:
   а) $\overline{(X \vee Y) \,\&\, (X \,\&\, \overline{Z})}$;
   б) $X \,\&\, Y \vee \overline{X} \,\&\, Y \vee X \,\&\, \overline{Y}$;
   в) $X \,\&\, (X \to Y) \,\&\, (X \to \overline{Y})$;
   г) $(X \to (Y \to Z)) \,\&\, (Z \to (X \to Y)) \,\&\, (Y \to (Z \to X))$.
6. а) Выведите тождество 18 из тождеств с меньшими номерами.
   б) Покажите, что законы де Моргана выводятся друг из друга с помощью закона двойного отрицания.
7. Проверьте, что следующие формулы являются тавтологиями:
   а) $X \to X$; б) $X \to (Y \to X)$.
8. Выясните, какие из следующих формул являются тавтологиями:
   а) $(X \to \overline{X}) \to X$; б) $X \vee Y \to X$; в) $X \,\&\, Y \to X$.
9. Выразите высказывание X через высказывания A и B, если имеет место равенство $\overline{(X \vee A)} \vee \overline{(X \vee \overline{A})} = B$.
10. * Выясните, существует ли такая формула F, при подстановке которой в следующую формулу эта формула становится тавтологией:
    а) $X \,\&\, Y \to F \,\&\, Z$; б) $F \,\&\, X \vee \overline{F} \,\&\, \overline{X} \,\&\, \overline{Y}$.

§32 Как построить логическую формулу

Каким бы ни было сложное высказывание, для него всегда можно составить таблицу истинности. А если дана некоторая таблица истинности, то всегда ли можно записать сложное высказывание, у которого была бы именно такая таблица истинности? Ответ на этот вопрос положителен. Мы покажем на примере, как строить сложное высказывание по таблице истинности, а потом сформулируем общее правило. Пусть, к примеру, таблица истинности выглядит так, как таблица 4.4.

Выберем строки, в которых для искомого высказывания стоит значение Истина. Для каждой такой строки вместо значения Истина в столбце простого высказывания напишем само высказывание, а вместо значения Ложь напишем его отрицание. Получим таблицу 4.5.

Теперь в каждой стро-ке соединим получившиеся высказывания конъюнкцией, а полученные таким способом сложные высказывания — дизъюнкцией. У нас получится сложное высказывание

$$A \,\&\, \overline{B} \,\&\, C \vee A \,\&\, B \,\&\, \overline{C} \vee \overline{A} \,\&\, B \,\&\, C \vee \overline{A} \,\&\, \overline{B} \,\&\, \overline{C}.$$

Это высказывание можно теперь преобразовывать по указанным ранее законам.

В общем случае для того, чтобы получить высказывание по таблице истинности, надо поступать точно так же:

- оставить в таблице те строки, в которых значение искомого выражения — Истина;
- в каждой клетке этих строк записать вместо значения Истина само высказывание из заголовка столбца, а вместо значения Ложь записать его отрицание;
- соединить конъюнкцией высказывания, стоящие в одной строке, а затем соединить дизъюнкцией получившиеся высказывания для всех отобранных строк.

Таблица 4.4

A	*B*	*C*	Искомое высказывание
И	И	И	Л
И	Л	И	И
И	И	Л	И
И	Л	Л	Л
Л	И	И	И
Л	Л	И	Л
Л	И	Л	Л
Л	Л	Л	И

Таблица 4.5

A	*B*	*C*	Искомое высказывание
A	$\overline{B}$	*C*	И
A	*B*	$\overline{C}$	И
$\overline{A}$	*B*	*C*	И
$\overline{A}$	$\overline{B}$	$\overline{C}$	И

Обратите внимание, что по этому алгоритму всегда получается формула, в которой используются только операции отрицания, конъюнкции и дизъюнкции, причем так, что отрицание применяется только к переменным, конъюнкцией соединены переменные или их отрицания, причем каждая переменная в таком конъюнктивном выражении фигурирует ровно один раз, и, наконец, дизъюнкции соединяют получившиеся конъюнктивные выражения. Выражение, записанное в таком виде, называется **совершенной дизъюнктивной нормальной формой** (сокращенно СДНФ). В силу договоренностей о порядке выполнения операций в СДНФ скобки не требуются.

ВОПРОСЫ И ЗАДАНИЯ

1. Сформулируйте алгоритм построения логической формулы по таблице истинности для логического выражения.
2. Что такое СДНФ?
3. Преобразуйте в выражение, не содержащее скобок, следующие высказывания:
   а) $(\overline{A} \,\&\, \overline{B} \vee \overline{A} \,\&\, B) \,\&\, A$; б) $\overline{\overline{A} \,\&\, B \,\&\, (\overline{B} \vee C) \vee C \,\&\, \overline{A}}$; в) $\overline{\overline{A} \vee B \vee B \,\&\, \overline{A}}$.
4. а) Запишите через A, B и C высказывания X, Y и Z, заданные таблицей 4.6.
   б) Запишите выражения $X \vee Y$, $Y \vee \overline{Z}$, $\overline{X \vee Z}$ и преобразуйте их в равносильные выражения, не содержащие скобок.

Таблица 4.6

A	B	C	X	Y	Z
И	И	И	Л	И	И
И	Л	И	И	Л	И
И	И	Л	Л	Л	Л
И	Л	Л	Л	Л	Л
Л	И	И	Л	И	И
Л	Л	И	И	И	И
Л	И	Л	И	Л	И
Л	Л	Л	Л	Л	Л

5. Известно, что высказывание F, зависящее от трех высказываний A, B и C, принимает значение Истина в том и только в том случае, когда ровно одно из высказываний A, B и C имеет значение Ложь. Составьте для F логическую формулу.
6. а) Известно, что высказывание F, зависящее от трех высказываний A, B и C, принимает то значение для каждого данного набора значений A, B и C, которое принимает большинство из этих трех высказываний. Функцию F поэтому называют **функцией голосования**. Составьте для F логическую формулу.
   б) Составьте логическую формулу для функции голосования от четырех переменных.

§33 Решение логических задач средствами математической логики

Цель изучения логики состоит в том, чтобы обеспечить средствами для доказательства логической обоснованности выводов одних утверждений из других. Для таких обоснований нередко полезным оказывается применение алгебры высказываний.

Проиллюстрируем применение алгебры высказываний на примере двух задач.

З а д а ч а 1. ***Кто изучал логику?*** На вопрос, кто из трех учащихся — Антон, Борис или Виктор — изучал логику, учитель, преподававший логику, ответил: «Если логику изучал Антон, то изучал и Боря, но неверно, что если изучал Витя, то изучал и Боря». Кто же изучал, а кто не изучал логику?

Р е ш е н и е. Обозначим буквой A высказывание «Антон изучал логику», буквой B высказывание «Борис изучал логику», буквой C высказывание «Виктор изучал логику». Тогда высказывание учителя можно записать так:

$$(A \to B)\ \&\ \overline{(C \to B)}.$$

Преобразуем эту запись, освободившись от скобок:

$$(A \to B)\ \&\ \overline{(C \to B)} = (\overline{A} \vee B)\ \&\ (C\ \&\ \overline{B}) = ((\overline{A}\ \&\ \overline{B}) \vee (B\ \&\ \overline{B}))\ \&\ C =$$
$$= ((\overline{A}\ \&\ \overline{B}) \vee Л)\ \&\ C = \overline{A}\ \&\ \overline{B}\ \&\ C.$$

Поскольку утверждение учителя мы, конечно, принимаем за истинное, истинными должны быть и утверждения $\overline{A}$, $\overline{B}$ и C. Следовательно, утверждение A ложно, утверждение B тоже ложно, а утверждение C истинно. Значит, логику изучал Виктор, а Антон и Борис логику не изучали.

Задача 2. ***Вердикт суда.*** Гражданин Иванов обвиняется в совершении преступления при соучастии Петрова и Сидорова. Суд присяжных в ходе слушания пришел к выводу, что следствием доказано следующее:

1) если Иванов не виновен или виновен Петров, то Сидоров виновен;

2) если Иванов не виновен, то Сидоров не виновен.

Присяжным требуется принять решение, виновен ли Иванов. Достаточно ли у них для этого оснований?

Решение. Обозначим буквой A высказывание «Иванов виновен», буквой B высказывание «Петров виновен», буквой C высказывание «Сидоров виновен». Тогда имеющуюся у присяжных информацию можно записать следующей формулой:

$$((\overline{A} \vee B) \to C)\ \&\ (\overline{A} \to \overline{C}).$$

Составим для нее таблицу истинности (табл. 4.7).

Анализу подлежат те строки, в которых значение составленной нами формулы равно И. Мы видим, что во всех этих случаях значение A равно И. Следовательно, Иванов виновен.

А вот для утверждения, что виновен Петров, оснований недостаточно. Ведь значение И наша формула может иметь и при истинном значении B, и при ложном. Впрочем, нет оснований и объявить Петрова невиновным.

Таблица 4.7

A	B	C	$((\overline{A} \vee B) \to C\ \&\ (\overline{A} \to \overline{C})$
И	И	И	И
И	Л	И	И
И	И	Л	Л
И	Л	Л	И
Л	И	И	Л
Л	Л	И	Л
Л	И	Л	Л
Л	Л	Л	Л

Если для перевода условия задачи на язык математической логики приходится использовать формулы с четырьмя, пятью или еще большим числом переменных, таблица истинности может насчитывать 16, 32 и более строк. Для составления таких таблиц уже полезно применить компьютер. Вы будете это делать при выполнении лабораторной работы 18.

ВОПРОСЫ И ЗАДАНИЯ

1. Обозначьте буквами простые высказывания и воспользуйтесь алгеброй логики, чтобы ответить на вопрос задачи.
Если Джон не встретил этой ночью Смита, то Смит был убийцей или Джон врет. Если Смит не был убийцей, то Джон не встретил Смита этой ночью и убийство произошло после полуночи. Если убийство произошло после полуночи, то Смит был убийцей или Джон лжет. Эксперты утверждают, что убийство произошло до полуночи. Можно ли утверждать, что Смит был убийцей?

2. Обозначьте буквами простые высказывания и воспользуйтесь алгеброй логики, чтобы ответить на вопрос задачи.

   Намеченная атака удастся, если захватить противника врасплох или его позиции плохо защищены. Захватить противника врасплох можно, только если он беспечен. Он не будет беспечен, если его позиции плохо защищены. Следует ли из этого, что атака не удастся?

3. Внимание Андрея, Дениса и Михаила привлек проехавший мимо них автомобиль.
— Это английская машина марки «феррари», — сказал Андрей.
— Нет, машина итальянская, марки «понтиак», — возразил Денис.
— Это «сааб», и сделан автомобиль не в Англии, — сказал Михаил.
Оказавшийся рядом знаток автомобилей сказал, что каждый из мальчиков прав только в одном из двух своих высказываний. Какой марки автомобиль и в какой стране он изготовлен?

§34 Реляционные модели

Для многих систем искусственного интеллекта важной составной частью является информация о тех объектах, процессах и явлениях, с которыми предстоит иметь дело пользователю данной системы искусственного интеллекта. Каждый объект при этом описывается значениями некоторого набора параметров. В § 9 мы обсуждали, что такое описание удобно представлять в виде таблицы. Например, такой, как таблица 4.8.

Таких таблиц для данного набора объектов может быть несколько — в каждой из них фигурирует свой набор параметров. Более того, по мере накопления знаний об изучаемых объектах могут составляться новые таблицы — совсем необязательно новую информацию записывать в уже имеющуюся таблицу, увеличивая в ней набор столбцов. Совокупность таблиц, в которых для рассматриваемой совокупности объектов приведены значения параметров, описывающих эти объекты, называется **реляционной моделью** данной системы объектов. Что означает слово «реляционная» и почему эта модель так называется, мы поясним немного позже. А сейчас обсудим, как с помощью такой модели можно получать нужную информацию. Приведенный ниже пример хотя и несколько далек от науки, но ярко показывает полезность таких моделей в разных областях человеческой деятельности.

Таблица 4.8

Параметр / Номер объекта	**Имя параметра 1**	**...**	**Имя параметра *k***
Объект 1	Значения параметра 1 для объекта 1	...	Значения параметра *k* для объекта 1
Объект 2	Значения параметра 1 для объекта 2	...	Значения параметра *k* для объекта 2
...	...	...	...
Объект ...	Значения параметра 1 для объекта ...	...	Значения параметра *k* для объекта ...

Представим себе деятельность междугородной телефонной компании, например, в Санкт-Петербурге. Она предоставляет своим абонентам связь с другими городами. Информацию о предоставленных услугах связи удобно представить таблицей.

Таблица 4.9

Номер	**Дата**	**Город**	**Продолжительность разговора**
111-22-33	01.01.02	Москва	7
111-22-33	02.01.02	Париж	9
122-33-44	01.01.02	Женева	20
122-33-44	02.01.02	Цюрих	17
122-33-44	02.01.02	Москва	7
123-34-45	01.01.02	Париж	11
...	...	...	...

Каждый новый звонок добавляет строку в эту таблицу.

Но телефонная компания хранит и информацию о владельцах телефонов, например, в такой форме, как таблица 4.10 (адреса и фамилии взяты нами, конечно, условно).

Пусть нас интересует, с какими городами и когда разговаривал Петров. Чтобы получить ответ, нужно проделать фактически то же самое, что и в предыдущем примере:

— по заданному значению параметра Владелец в таблице 4.10 разыскиваем значение параметра Номер;

Таблица 4.10

Номер	Владелец	Адрес
111-22-33	Иванов	Невский, 17
122-33-44	Петров	Фонтанка, 4
123-34-45	Сидоров	Лиговский, 7
...	...	...

Таблица 4.11

Дата	Город
01.01.02	Женева
02.01.02	Цюрих
02.01.02	Москва
03.01.02	Берлин

— по найденному значению параметра Номер в таблице 4.9 разыскиваем значения параметров Дата и Город;

— результат представляем новой таблицей 4.11, где отображены разговоры Петрова.

Таким образом, чтобы получить нужную информацию, между таблицами необходимо установить **связь**, указав, какие параметры для них являются общими. В рассмотренном примере таким параметром является Номер.

Идея реляционной модели была предложена американским ученым Е. Ф. Коддом в начале 70-х годов XX века. Само слово «реляционная» происходит от английского *relation* — отношение, связь. Иными словами, суть реляционного подхода заключается в том, что информация об объектах представляется в виде **отношений**, т. е. связанных между собой характеристик изучаемых объектов.

Впрочем, в самой идее реляционности как раз нет ничего удивительного. Описывая окружающий мир, мы всегда не просто перечисляем его объекты, а обязательно называем отношения, которыми эти объекты связаны друг с другом. Рассмотрите для примера несколько фраз:

1. Дрозд — это птица;
2. Петр — отец Павла;
3. Васе нравится Аня;
4. Прямые a, b и c пересекаются в одной точке;
5. Маша взяла у Алеши сказки «Тысяча и одна ночь».

В каждой из этих фраз фиксируется то или иное отношение между объектами. Первая фраза дает пример отношения принадлежности: дрозд *принадлежит* множеству птиц. Вторая и третья устанавливают отношения *быть отцом* между Петром и Павлом и *нравится* между Васей и Аней. Четвертая фраза указывает на отношение между тремя объектами. Наконец, пятая фраза описывает отношение между Машей, Алешей и конкретной книгой.

Разумеется, в одном и том же отношении могут находиться самые разнообразные объекты. Например, в отношении *быть отцом* находятся, конечно, не только Петр и Павел, но и многие другие пары людей. В отношении *кто-то у кого-то что-то взял* находят-

ся тоже не только Маша, Алеша и «Тысяча и одна ночь». Поэтому если сосредоточить внимание на самом отношении, то оказывается удобным считать, что отношение — это некое выражение с переменными:

объект x — это отец объекта y;
объект x взял у объекта y объект z; и т. д.

Осталось сделать еще один шаг: унифицировать запись отношений. Это можно сделать, например, так:

быть_отцом (x, y);
взять (x, y, z); и т. д.

То, что стоит перед скобками, — **имя** отношения, в скобках перечислены **аргументы** отношения, количество аргументов отношения называют его **арностью**. Так что первое из записанных в стандартной форме отношений **бинарное** (по-русски двуместное), второе — тринарное (по-русски трехместное). Часто отношения между объектами, обозначенными буквами a, b, c, d и т. д., схематично записывают так: $R(a, b, c, d, \ldots)$, где через R обозначено **имя** рассматриваемого отношения. Впрочем, если отношение связывает только два объекта a и b, то нередко пишут aRb. Например, мы пишем $a < b$ для чисел, $a \parallel b$ для прямых, $a \in A$ для элемента a из множества A и т. п. Мы тоже будем так писать для уже устоявшихся обозначений отношений (равенства, неравенства, параллельности, принадлежности и т. п.).

В предложенном варианте записи отношения часть информации может оказаться утерянной. Скажем, из записи *быть_отцом* (x, y) уже не видно, то ли x — отец для y, то ли наоборот. Выход простой: для каждого аргумента указать, что он означает; можно сказать, что мы каждому аргументу присваиваем **имя**, фактически указывающее, из какого множества будут браться объекты для данного аргумента. Например:

быть_отцом (отец: x, ребенок: y);
взять (кто_взял: x, у_кого_взял: y, что_взял: z).

Имя аргумента нередко называют **атрибутом** данного отношения (это согласуется с понятием атрибута, введенного в § 12).

Вместо аргументов нельзя подставлять любые объекты из тех множеств, которые обозначены атрибутами. Скажем, нельзя в отношении *быть_отцом* вместо x подставлять имя любого мужчины, а вместо y — имя любого ребенка. Как же тогда определять, каковы те наборы значений аргументов, для которых имеет место данное отношение? Иногда это настоящая детективная история, когда, например, следователь пытается выяснить, кто же без спроса взял у хозяина его бриллианты.

Отношения могут задаваться по-разному. Нередко бывает так, что для каждого аргумента множество его значений конечно. В этом случае отношение можно задать списком всех тех наборов

Таблица 4.12

Мама	Папа	Ребенок
Ева	Адам	Авель
Ева	Адам	Каин
...	...	...

Таблица 4.13

Начальник	Подчиненный
Иванов	Петров
Иванов	Сидоров
...	...

значений аргументов, которые находятся в данном отношении. А сам такой список удобно представлять в виде таблицы, где каждая строка — это набор значений аргументов, находящихся в данном отношении, а каждый столбец — это перечень значений соответствующего атрибута. Сами атрибуты выступают при этом в роли заголовков столбцов.

В таблицах 4.12 и 4.13 представлены отношения «родство» и «служба».

Эти отношения — человеческие. Но отношения могут быть между объектами любой природы. Ведь каждая таблица может рассматриваться как некое отношение. Железнодорожное расписание задает отношения между маршрутами, станциями и т. п. Таблица умножения задает отношение между числами.

Если внимательно приглядеться, то отношение (а следовательно, и таблицу) можно обнаружить где угодно. Например, память компьютера можно трактовать как отношение с атрибутами *Адрес* и *Содержимое*. Достаточно вспомнить, что оперативная память состоит из ячеек, каждая из которых имеет адрес, задаваемый натуральным числом, а содержимое ячейки — это последовательность нулей и единиц. И если, скажем, в 1-й ячейке записано число 1001, во 2-й — число 10101 и т. д., то память можно задать таблицей 4.14.

Таблица 4.14

Адрес	Содержимое
1	1001
2	10101
...	...

Наличие отношения между характеристиками разных объектов или даже одного и того же объекта свидетельствует о зависимости этих характеристик друг от друга. Наиболее жестко характеристики оказываются связанными, если по значению одной из них можно определить значения остальных, находящихся в данном отношении. Такие отношения мы будем рассматривать в § 35. Но следует понимать, что даже если нет такой жесткой зависимости между параметрами, все равно отношение, не совпадающее со всевозможными наборами значений, указывает на наличие некоторой связи между ними. Именно поэтому отношения стали важным инструментом построения и исследования самых разнообразных информационных моделей — ведь одним из компонентов информационной модели как раз является описание связей между ее параметрами.

ВОПРОСЫ И ЗАДАНИЯ

1. Какую роль играет понятие отношения в информационном моделировании?
2. Как можно задавать отношения?
3. Вы знаете, что графиком функции $y = f(x)$ называется множество точек координатной плоскости, имеющих координаты $(x; f(x))$. Можно ли поэтому рассматривать график функции как способ задания некоторого отношения?
4. Приведите какие-либо примеры отношений, рассматриваемых:
   а) в математике; б) в физике;
   в) в химии; г) в биологии.
   Какие из названных вами отношений удобно представлять в виде таблицы?
5. Пусть множество M состоит из чисел 1, 2, 3, 4, 5 и 6. На этом множестве заданы следующие отношения:
   а) $R_1(x, y)$: число x делится на число y;
   б) $R_2(x, y)$: числа x и y таковы, что $|x - y| < 3$;
   в) $R_3(x, y)$: число $x + y$ принадлежит множеству M.
   Запишите каждое из этих отношений в виде таблицы.
6. Даны отношения Нагрузка_*учителя* (фамилия:, класс:, предмет:) и *Расписание* (класс:, предмет:, день_недели:, номер_урока:). Выберите 12—15 классов вашей школы и составьте для них таблицы, соответствующие указанным отношениям.

§35 Функциональные отношения

Понятие функциональной зависимости — одно из величайших изобретений человеческой мысли. И хотя слово «функция» наверняка ассоциируется у вас с математикой, на самом деле с функциями человек имеет дело повсеместно. Но прежде чем привести примеры, подтверждающие это высказывание, напомним, что функцией называется сопоставление каждому элементу одного множества ровно одного элемента из другого множества. Отметим, что это «другое множество» совсем не обязано отличаться от исходного.

В математике обычно рассматриваются числовые функции, т. е. оба множества — это множества чисел. На самом деле это совсем необязательно. Вот примеры функций:

1) каждому человеку сопоставляется его фамилия (функция из множества людей в некоторое множество слов);
2) каждому человеку сопоставляется его рост (функция из множества людей в некоторое множество положительных чисел);
3) каждому городу России сопоставляется его почтовый индекс (функция из множества городов России в множество шестизначных натуральных чисел);
4) каждой точке на поверхности Земли сопоставляются ее географические координаты (функция из множества точек поверхности в некоторое множество пар, каждая компонента которой — положительное число с указанием северного или южного, восточного или западного полушарий);

5) каждому набору отпечатков пяти пальцев правой руки, имеющихся в картотеке МВД, сопоставляется человек с такими отпечатками.

Пусть нам дана некоторая функция f, сопоставляющая каждому элементу x из одного множества элемент y другого множества. Поскольку y по x определяется однозначно, то обычно пишут $y = f(x)$. Определим отношение R, объявив пару $(x; y)$ находящейся в отношении R в том и только в том случае, если $y = f(x)$.

Тем самым каждая функция может быть описана как некоторое отношение, и даже не одно. Обратное, однако, неверно — ведь не у каждого отношения можно так выбрать атрибуты, чтобы значения одного из них однозначно определяли значения другого. Рассмотрим отношение «x является родным братом для y». Конечно, если в семье только два брата, то для каждого x значение y определено однозначно. Но есть семьи с бóльшим числом братьев.

Говоря о представлении функции в виде отношения, мы имели в виду лишь бинарные отношения. Однако для функции значения аргумента, как и значения функции, вовсе не обязаны состоять из значений одного атрибута. Скажем, для функции, сопоставляющей набору отпечатков пяти пальцев имя, отчество и фамилию их владельца, аргумент содержит пять атрибутов, а значение функции — три. Записывая эту функцию как отношение, мы получим отношение с восемью атрибутами, разделенными на две группы: первая задает аргумент функции, вторая — ее значение.

Будем называть отношение **функциональным**, если его атрибуты можно разбить на две группы так, чтобы значения одной группы атрибутов однозначно определяли значения другой группы. Иными словами, первая группа атрибутов может рассматриваться как аргумент некоторой функции, а вторая группа определяет значение этой функции. Вовсе не обязательно, чтобы атрибуты одной группы были соседними в записи отношения. Если отношение функционально, то набор атрибутов, относящихся к аргументу функции, задаваемой этим отношением, будем называть **ключевым**.

Рассмотрим для примера отношение, заданное таблицей 4.15.

Атрибут *Адрес* не может быть ключевым, поскольку в одном доме оказалось несколько телефонов. Если среди владельцев имеются однофамильцы, то атрибут *Владелец* также не может быть ключевым. А вот *Номер* телефона — это ключевой атрибут. Скорее всего, ключевым будет и набор из двух атрибутов *Владелец*—*Адрес*: слишком мала вероятность проживания в доме однофамильцев или владельца двух телефонных номеров сразу. Для отношения, пред-

Таблица 4.15

Номер	Владелец	Адрес
111-22-33	Иванов	Невский, 17
122-33-44	Петров	Фонтанка, 4
123-34-45	Сидоров	Лиговский, 7
123-35-47	Алексеев	Лиговский, 7
...	...	...

ставленного таблицей, легко сформулировать критерий того, что данный набор атрибутов является ключевым. А именно

! *набор атрибутов ключевой, если в таблице нет двух строк с одинаковым набором значений этих атрибутов.*

ВОПРОСЫ И ЗАДАНИЯ

1. Какое отношение называют функциональным? Приведите примеры функциональных отношений.

2. В § 34 в качестве примера рассматривалось отношение *быть_отцом* (отец: x, ребенок: y). Является ли это отношение функциональным? Если да, то какой атрибут служит аргументом, а какой — значением функции?

3. Рассмотрим отношение *музыкальное_произведение* (название: x, автор: y, характер произведения: z). Вот примеры значений аргументов, для которых отношение имеет место:
музыкальное_произведение (название: *Лебединое озеро*, автор: *П. И. Чайковский*, характер произведения: *балет*);
музыкальное_произведение (название: *Пиковая дама*, автор: *П. И. Чайковский*, характер произведения: *опера*);
музыкальное_произведение (название: *Кармен*, автор: *Бизе*, характер произведения: *опера*);
музыкальное_произведение (название: *симфония № 8*, автор: *Л. ван Бетховен*, характер произведения: *симфония*).
Является ли это отношение функциональным? Если да, то какой набор атрибутов можно рассматривать как ключевой?

Таблица 4.16

Порядковый номер элемента	Электронный слой			
	K	*L*	*M*	*N*
19	2	8	8	1
20	2	8	8	2
21	2	8	9	2
22	2	8	10	2
23	2	8	11	2
24	2	8	13	1
25	2	8	13	2
26	2	8	14	2
27	2	8	15	2
28	2	8	16	2
29	2	8	18	1
30	2	8	18	2
31	2	8	18	3
32	2	8	18	4
33	2	8	18	5
34	2	8	18	6
35	2	8	18	7
36	2	8	18	8

4. Рассмотрим на множестве натуральных чисел отношение *сумма* (слагаемое 1: x, слагаемое 2: y, результат: z). Это отношение имеет место для чисел x, y и z в том и только в том случае, когда $z = x + y$. Является ли это отношение функциональным? Если да, то какой набор атрибутов можно рассматривать как ключевой?

5.* В таблице 4.16 для элементов четвертого периода указано строение электронных слоев.

а) Первый атрибут — порядковый номер элемента — является ключевым. Какие еще ключевые наборы атрибутов имеет эта таблица?

б) Какие химические свойства определяют наборы ключевых атрибутов, выделенные вами в задании *а*?

§36 Логические функции и логические выражения

Логической функцией или, по-другому, **предикатом** на множестве M называют такую функцию от нескольких аргументов, которая при любом наборе значений этих аргументов из множества M принимает только одно из двух значений. Обычно одно из этих значений называют Истина, другое — Ложь.

В языках программирования часто используются английские слова того же смысла True и False (реже Да и Нет или 1 и 0). Нередко предикат называют еще высказывательной формой, поскольку после подстановки вместо переменных элементов множества получается некое утверждение об этом наборе элементов, которое является либо истинным, либо ложным. Например, предикат «сумма x и y равна z» от трех аргументов x, y и z, рассматриваемый на множестве натуральных чисел, принимает значение Истина при $x = 3$, $y = 4$, $z = 7$ и значение Ложь при $x = 2$, $y = 2$, $z = 5$. По аналогии с общим обозначением в математике функции как $f(x_1, x_2, ..., x_n)$ в качестве общего обозначения предиката мы будем использовать запись $P(x_1, x_2, ..., x_n)$. Впрочем, вместо P можно использовать любую букву латинского алфавита.

В приведенном примере переменные x, y и z **свободны** в том смысле, что могут принимать любые значения из множества натуральных чисел. Поэтому данная логическая функция имеет три аргумента. Но не всегда число аргументов логической функции совпадает с числом фигурирующих в ее описании переменных. Рассмотрим, такой предикат: «существует x, для которого сумма x и y равна z». Хотя в описании фигурируют три переменные x, y и z, подставлять числа можно только вместо двух из них — y и z. Так что здесь только два аргумента: y и z. В таблице 4.17 приведены значения данной логической функции для некоторых наборов значений аргументов y и z (этот предикат мы рассматриваем на множестве натуральных чисел).

Переменная x в такой функции называется **связанной**. При этом говорят, что переменная x связана **квантором существования**. Для него есть специальное обозначение: ∃. Происхождение этого знака простое: в английском слове «Exist» — существовать — взята первая буква и симметрично отражена относительно вертикальной оси. С помощью этого символа рассматриваемый нами предикат записывается так: $\exists x\,(x + y = z)$.

Таблица 4.17

y	z	Значение функции	Комментарий
1	2	Истина	Подходит $x = 1$
2	2	Ложь	Не существует подходящего x
3	5	Истина	Подходит $x = 2$
5	3	Ложь	Не существует подходящего x

Таблица 4.18

x	z	Значение функции	Комментарий
1	1	Истина	При любом y верно $1 + y > 1$
1	2	Ложь	Не подходит $y = 1$
3	5	Ложь	Не подходит, например, $y = 2$
5	3	Истина	При любом y верно $5 + y > 3$

Впрочем, переменная может быть связанной и по-другому. Рассмотрим, для примера, на множестве натуральных чисел предикат «для любого y выполнено неравенство $x + y > z$». Здесь связанной переменной является y. Примеры значений этого предиката приведены в таблице 4.18.

В этом случае говорят, что переменная связана **квантором всеобщности**, который обозначают символом $\forall$. Его происхождение аналогично: от слова «All» (все) взята первая буква и симметрично отражена относительно горизонтальной оси. С помощью этого квантора данный предикат запишется так: $\forall y\,(x + y > z)$.

В предикате могут оказаться связанными не одна, а несколько переменных. Например, можно рассматривать предикат $\forall y\,\exists x\,(x + y = z)$ — для любого y существует x, такой, что выполняется равенство $x + y = z$. Или другой предикат: $\exists x\,\forall y\,(x + y = z)$ — существует x, такой, что для любого y выполняется равенство $x + y = z$. Каждый из них является логической функцией от одного аргумента z, но это разные функции. Например, на множестве целых чисел первая из этих функций при любом значении аргумента z принимает значение Истина, в то время как вторая функция на том же множестве при любом значении аргумента z принимает значение Ложь. Как видите, порядок, в котором употреблены кванторы, имеет принципиальное значение.

Если в предикате все переменные оказались связанными, то такой предикат является высказыванием. Например, предикат $\forall z\,\forall y\,\exists x\,(x + y = z)$ — это высказывание, утверждающее, что для любых чисел z и y существует их разность (она обозначена переменной x). Это высказывание истинно на множестве целых чисел, но ложно на множестве натуральных чисел. Поэтому, обсуждая свойства того или иного предиката, надо всегда указывать множество, на котором он рассматривается. Впрочем, и для числовых функций, которые вы уже много лет изучаете на уроках математики, самое первое, о чем идет речь, — это их область определения.

Над логическими функциями можно выполнять все те же логические операции, которые рассматривались нами для высказываний в § 30. Чтобы вычислить значение такой «составной» функции, достаточно знать логические значения функций, из которых она составлена. Например, логическая функция $\overline{P(x_1, x_2, \ldots, x_n)}$ принимает значение Истина тогда и только тогда, когда логическая функция $P(x_1, x_2, \ldots, x_n)$ принимает значение Ложь.

При словесном описании логической функции построение отрицания к какому-либо утверждению можно выполнить добавлением словосочетания «Неверно, что...», после чего следует исходное утверждение. Например, отрицание высказывания «Я пошел в кино» можно выразить так: «Неверно, что я пошел в кино». Правда, таким образом свою мысль выражают крайне редко. Обычно говорят: «Я не пошел в кино». Но заметьте, что ни одна из фраз «Не я пошел в кино» и «Я пошел не в кино» не является отрицанием исходного высказывания.

А теперь рассмотрим, как строится отрицание высказывания, полученного связыванием переменной при помощи квантора. Вот пример высказывания: «Все ученики нашего класса имеют дома компьютер». Его отрицанием является высказывание «Неверно, что все ученики нашего класса имеют дома компьютер». Но каждому ясно, что это высказывание равносильно такому: «Существует ученик нашего класса, у которого дома нет компьютера». Как видите, при построении отрицания квантор всеобщности преобразуется в квантор существования. Более точно, если через $P(x)$ обозначить предикат «ученик x имеет дома компьютер», то исходное высказывание запишется так: $\forall x\,(P(x))$. А его отрицание запишется как $\exists x\,(\overline{P(x)})$. Аналогично можно объяснить, почему при построении отрицания квантор существования заменяется квантором всеобщности.

Итак, для логических функций, имеющих вид

$$Q_1x_1\,Q_2x_2\,\ldots\,Q_kx_k(P(x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_n)),$$

где $Q_1, Q_2, \ldots, Q_k$ — символы $\forall$ или $\exists$, $x_1, x_2, \ldots, x_k$ — связанные переменные, $y_1, y_2, \ldots, y_n$ — свободные переменные предиката P, справедливо следующее правило построения отрицания. Чтобы получить отрицание логической функции, надо каждый квантор всеобщности заменить квантором существования и наоборот, а предикат P заменить его отрицанием.

ВОПРОСЫ И ЗАДАНИЯ

1. Какую функцию называют логической? Приведите примеры логических функций.

2. а) Для предиката «существует x, для которого сумма x и y равна z», рассматриваемого на множестве натуральных чисел, назовите еще одну пару значений аргументов, для которых этот предикат истинен, и одну пару, для которой он ложен.
б) Для того же предиката, рассматриваемого на множестве слов русского языка (сложение здесь понимается как операция конкатенации), определите значение этого предиката, если y = Ель, z = Газель; y = Гель, z = Газель; y = Газель, z = Газель.

3. Для каждого из предикатов, приведенных в пунктах *а*—*в*, укажите, сколько аргументов он имеет, и назовите, какие его переменные свободны, а какие связаны.
а) Существует только один металл x, который в ряду активности располагается между элементами y и z.

б) Нет такого животного x, которое в ряду эволюции располагалось бы между животными y и z.
в) Для каждого химического элемента x существует химический элемент y, с которым x образует соединение.

4 Рассмотрите предикаты, заданные на множестве натуральных чисел:
а) x — нечетное число, и для любого нечетного числа y выполнено неравенство $x \leqslant y$;
б) x — простое число, и для любого простого числа y выполнено неравенство $x \leqslant y$.
Для каждого из этих предикатов укажите все те значения аргумента x, для которых данный предикат истинен.

5 Рассмотрим предикат $P(x, y)$: Фигура x вписана в фигуру y.
а) Пусть x пробегает множество всех треугольников, расположенных на некоторой плоскости, а y — множество всех окружностей на той же плоскости. Каждое из высказываний

$$\forall x\, \forall y\, (P(x, y));\ \ \forall x\, \exists y\, (P(x, y));$$
$$\exists x\, \forall y\, (P(x, y));\ \ \exists x\, \exists y\, (P(x, y))$$

запишите предложением русского языка. Определите, какое из них истинно, а какое ложно.
б) Выполните то же задание, что и в пункте *а*, если x пробегает множество всех окружностей, расположенных на некоторой плоскости, а y — множество всех треугольников той же плоскости.

6 Рассмотрим предикат $P(x, y)$: Фигура x вписана в фигуру y.
а) Пусть x пробегает множество всех выпуклых четырехугольников, а y — множество всех окружностей. Укажите, для каких из четырех вариантов кванторных приставок $\forall x\, \forall y$, $\forall x\, \exists y$, $\exists x\, \forall y$, $\exists x\, \exists y$ после связывания ими переменных в предикате $P(x, y)$ получается истинное высказывание.
б) Выполните то же задание, что и в пункте *а*, если x пробегает множество всех окружностей, а y — множество всех выпуклых четырехугольников.

7 В объяснительном тексте параграфа было показано, что логические функции $\forall x\, \exists y\, (P(x, y, z))$ и $\exists y\, \forall x\, (P(x, y, z))$, вообще говоря, различны.

а) Различны ли функции $\forall x\, \forall y\, (P(x, y, z))$ и $\forall y\, \forall x\, (P(x, y, z))$, где $P(x, y, z)$ — произвольный предикат от трех переменных? А функции $\exists x\, \exists y\, (P(x, y, z))$ и $\exists y\, \exists x\, (P(x, y, z))$?

б)* Известно, что высказывание $\exists y\, \forall x\, (P(x, y))$ истинно. Можно ли утверждать, что высказывание $\forall x\, \exists y\, (P(x, y))$ тоже истинно?

8 На множестве натуральных чисел заданы следующие предикаты: $N(x)$ — число x нечетно; $D(x, y)$ — число x делится на число y. Используя указанные предикаты, запишите подходящей формулой каждое из следующих утверждений:
а) число x четно;
б) число x — нечетный делитель числа y;
в) числа x и y равны;
г)* число x равно 1;
д)* число z — наибольший общий делитель чисел x и y;
е)* числа x и y взаимно просты (т. е. их наибольший общий делитель равен 1);
ж)* число x простое;
з)* число x равно 2;
и)* если x — простое число, то для любого числа y либо x и y взаимно просты, либо y делится на x;
к)* любое число, большее 1, имеет простой делитель.

9 На множестве натуральных чисел задан предикат $U(x, y, z)$: Число z равно произведению чисел x и y. Используя этот предикат, запишите подходящей формулой каждое из следующих утверждений:
а) число x — делитель числа y;
б) числа x и y равны;
в) число x равно 1;
г)* число x простое;
д)* число x является степенью простого числа.

10 Ниже приведено несколько пар высказываний. Определите, в каких парах одно высказывание является отрицанием другого.
а) «На улице светит солнце» и «На улице пасмурно».
б) «На улице светит солнце» и «На улице идет дождь».
в) «Число 3 простое» и «Число 3 составное».
г) «Число 1 простое» и «Число 1 составное».
д) «В классе есть ученик, который решил все задачи контрольной работы» и «В классе каждый ученик не решил хотя бы одну задачу из контрольной работы».
е) «В классе есть ученик, который решил все задачи контрольной работы» и «В контрольной работе есть задание, которое не решили все ученики класса».
ж) «Все числа иррациональны» и «Существует иррациональное число».
з) «Для любого числа x существует число z, для которого произведение xz рационально» и «Существует число x, для которого произведение xz иррационально для любого числа z».
и)* «Все нечетные числа простые» и «Существует нечетное простое число».

11 Ниже приведено несколько пар логических функций. Определите, в каких парах одна логическая функция является отрицанием другой функции из той же пары.
а) «Число x положительно» и «Число x отрицательно».
б) «Он мой друг» и «Он мой враг».
в) «Ученик решил все задания контрольной работы» и «Есть задание контрольной работы, которое ученик не решил».

12 а) На множестве действительных чисел задан предикат

$$\forall x\,(x^2y - x(y - 3) + 1 > 0).$$

Укажите все значения переменной y, для которых этот предикат истинен.
б)* Выполните то же задание, что и в пункте *а*, если предикат задан на множестве положительных чисел.

§37 Логика СУБД Access

Логические функции, рассматривавшиеся нами в качестве примеров в предыдущем параграфе, были довольно простыми и легко записывались на естественном языке. Но компьютер, как вы знаете, понимает только формальный язык. И любой формальный исполнитель, каким, в частности, является СУБД Access, тоже понимает только формальный язык. Вот об этом языке и пойдет у нас речь. Основу его составляют логические функции, о которых мы рассказывали в предыдущем параграфе.

А теперь вспомните ваш опыт обращения с числовыми функциями на уроках математики. Изобретение буквенных обозначений переменных и языка формул для записи связей между ними стало

революционным шагом. Вот и для логических функций естественно ввести некоторый язык формул, позволяющий относительно просто и достаточно стандартно записывать их. Строятся эти формулы, как и формулы в алгебре, из некоторых «кирпичиков», соединенных знаками операций. Для алгебраических выражений «кирпичиками» выступают числа и переменные, в качестве операций выступают арифметические операции — сложение, умножение, вычитание и деление, а также некоторые стандартные функции (показательная, логарифмическая, тригонометрические и т. п.).

Что же выступает в роли «кирпичиков» логического выражения? Это некоторые стандартные предикаты, определяющие, истинно или ложно утверждение о том, что при заданных значениях переменных они находятся в заданном отношении. В качестве таких отношений обычно фигурируют отношения сравнения = (равно), <> (не равно), < (меньше), > (больше) и т. п. Более точно можно сказать так: **атомарным** называется отношение вида $A\ \theta\ B$, где A и B — *однотипные* выражения от переменных, а θ — один из символов отношения сравнения. Однотипность выражений означает, что после вычисления их значений для любого допустимого набора переменных мы получаем значения одного типа, например числового, или символьного, или еще какого-либо.

Теперь можно формально определить логические выражения. Мы будем обозначать их большими латинскими буквами.

1. Всякое атомарное отношение есть логическое выражение.
2. Истина и Ложь — логические выражения.
3. Если A — логическое выражение, то выражение ***не*** (A) тоже логическое выражение.
4. Если A и B — логические выражения, то выражения A ***и*** B, A ***или*** B и ***не*** B тоже являются логическими.

Вот примеры логических выражений:

не (x = СЛОН),
($x = 2 + 2$) ***и*** $y > 2*(x + 3)$.

Правила 1—4 определяют формальную грамматику языка логических выражений. Заметим, что в определении логического выражения мы использовали не все логические операции, но это не сказывается на возможностях данного языка, поскольку, как вы знаете, операции «следование», «равносильность», «исключающее или» выражаются через указанные три операции. Но дело тут не в экономии используемых операций, а в том, что СУБД Access воспринимает, как мы чуть позже убедимся, язык логических выражений с операциями отрицания, дизъюнкции и конъюнкции.

Логические выражения нужны для того, чтобы компьютеру на понятном для него языке сформулировать, с какой логической функцией ему надлежит иметь дело.

Пусть теперь на множестве M нам дано некоторое отношение с атрибутами x_1, x_2, ..., x_n. Рассмотрим логическую функцию от тех же переменных (не обязательно всех). Возьмем произвольный

набор значений атрибутов, который принадлежит данному отношению. Подставим эти значения в логическую функцию вместо соответствующих переменных. Тогда логическая функция примет определенное значение. Выберем из всех наборов значений атрибутов, принадлежащих данному отношению, все те, для которых данная функция принимает значение Истина. Такую операцию над отношением называют **фильтрацией**, а саму логическую функцию называют **фильтром**. Если отношение задано таблицей, то фильтрация приводит к отбору тех строк, в которых стоят значения атрибутов, дающих истинное значение заданной логической функции.

Как вы помните, таблицы — это основной способ представления информации в реляционных базах данных. Знакомясь с СУБД Access, вы рассматривали и фильтрацию, т. е. отбор строк таблицы, удовлетворяющих условиям записанным на бланк QBE (эскиз бланка представлен таблицей КП. 5). Теперь, мы надеемся, вам понятно, что на бланк QBE вы специальным образом записывали некоторый предикат (без кванторов!), в котором имена атрибутов — это имена переменных, с помощью союза ***и*** в нем соединены атомарные отношения, стоящие в одной строке, а получившиеся таким образом сложные выражения соединяются союзом ***или***. Итак, на бланке QBE записывается не что иное, как логическое выражение, представлено оно в дизъюнктивной нормальной форме. Чтобы это увидеть, достаточно «списать» его с бланка QBE по указанным правилам. Напомним (см. § 32), что любое сложное логическое выражение может быть записано через простые с помощью дизъюнктивной нормальной формы. Так что фильтр в СУБД Access — это фактически бескванторное логическое выражение, записанное в дизъюнктивной нормальной форме.

Наверное, у вас возник вопрос: почему в языке логических выражений СУБД Access отсутствуют кванторы? Ответ такой: в любой базе данных, созданной с помощью СУБД, присутствует лишь конечное число значений для любого атрибута. Представим, что у некоторого атрибута x множество всех его значений — это a_1, a_2, ..., a_n. Легко понять, что тогда для любого предиката

$$P(x, y_1, y_2, \ldots, y_n)$$

выражение

$$\forall x\,(P(x, y_1, y_2, \ldots, y_n))$$

равносильно конъюнкции

$$P(a_1, y_1, y_2, \ldots, y_k) \textit{ и } P(a_2, y_1, y_2, \ldots, y_k) \textit{ и } \ldots \textit{ и } P(a_n, y_1, y_2, \ldots, y_k),$$

а выражение

$$\exists x\,(P(x, y_1, y_2, \ldots, y_n))$$

равносильно дизъюнкции

$$P(a_1, y_1, y_2, \ldots, y_k) \textit{ или } \ldots \textit{ или } P(a_n, y_1, y_2, \ldots, y_k).$$

Так что в кванторах нужды нет.

Обратимся к нашему «телефонному» примеру. В базе данных телефонной станции имеется таблица, в которой указано, кто именно является абонентом данного телефонного номера (см. табл. 4.15). Она обновляется нечасто. Кроме того, имеется таблица совершенных телефонных звонков (см. табл. 4.9), которая посстоянно обновляется. Имеются еще таблицы, содержащие сведения об оплате телефонных разговоров их владельцами, о выставлении счетов и т. д. Иметь все эти сведения в одной таблице не только неудобно, но и неэффективно, поскольку ее объем будет огромен. Но вот наступает момент, когда надо выписать абонентам счета за текущий месяц. Теперь надо по таблице вызовов через номер телефона определить абонента и записать в счет все относящиеся к нему вызовы. На языке отношений это можно выразить так.

Пусть $S(x, y, z)$ — отношение, показывающее, что по телефону с номером x был разговор с пунктом y продолжительностью z минут, а $A(x, v)$ — отношение, показывающее, что v является владельцем телефона с номером x. Нам же нужно отношение $M(v, y, z)$, показывающее, что абонент v разговаривал с пунктом y в течение z минут. Отношение определяется так: имеет место отношение $M(v, y, z)$, если существует x, для которого $S(x, y, z)$ и $A(x, v)$. В этом случае говорят, что $M(v, y, z)$ является **произведением отношений** $S(x, y, z)$ и $A(x, v)$.

Как видите, появился квантор существования. Поскольку Access — многотабличная СУБД, то объединения таблиц приходится выполнять операцией умножения отношений. Правда, в документации к этой СУБД такая операция называется **соединением таблиц**. Поскольку Access имеет графический интерфейс, то установление соединения таблиц по нужным атрибутам производится с помощью мышки «протягиванием веревочки» между этими атрибутами. На рисунке 4.1 показана такая связь между атрибутами *а1* и *б1* двух таблиц (одна названа нами А, другая — Б).

Если об атрибутах ничего не сказано, то связь выглядит именно так. Если же какой-либо атрибут из А объявлен ключом (см. § 35), то около него, над «веревочкой» появляется цифра 1, а на другом конце появится значок ∞. Это означает, что каждая строка таблицы А может оказаться связанной с несколькими строками таблицы Б, но каждая строка таблицы Б связана ровно с одной строкой таблицы А. Такую связь таблиц называют **связью по типу один-ко-многим**. Если же оба атрибута объявлены ключевыми, то на обоих концах появляются 1, которые означают, что в обеих таблицах каждая строка связана ровно с одной строкой из другой таблицы. Такую связь называют **связью по типу один-к-одному**. Чтобы атрибут можно было объявить ключом, нужно, чтобы он удовлетворял требованиям, сформулированным в § 35.

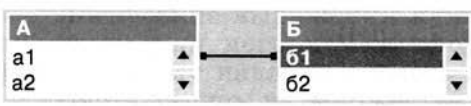


Рис. 4.1 Связь между атрибутами (полями) а1 и б1

ВОПРОСЫ И ЗАДАНИЯ

1. Объясните, почему в языке логических выражений, на котором в СУБД Access записываются запросы на фильтрацию, не нужны кванторы.
2. Для таблицы 4.16 (см. задание 5 к § 35) укажите результат фильтрации относительно логической функции: «$M > 12$ и $2 \leqslant N \leqslant 4$». По периодической системе Д. И. Менделеева определите, какие химические элементы будут отобраны этой фильтрацией.
3. Даны отношения *Нагрузка_учитель* (фамилия:, класс_и_предмет:) и *Расписание* (класс_и_предмет:, день_недели:, номер_урока:). Какую информацию несет произведение этих отношений?
4. Пусть множество M состоит из чисел 1, 2, 3, 4, 5 и 6. На этом множестве заданы следующие отношения:
   а) $R_1(x, y)$: число x делится на число y;
   б) $R_2(y, z)$: числа y и z таковы, что $|y - z| < 3$.
   Найдите произведение отношений R_1 и R_2. (Совет. Воспользуйтесь таблицами, составленными вами при выполнении задания 5 из § 34.)
5. Для каких целей используется соединение таблиц в СУБД Access?
6. Каким свойством обладает связь атрибутов разных таблиц, если оба атрибута являются ключевыми?

§ 38 Базы знаний и экспертные системы

Приходилось ли вам советоваться, прежде чем принять то или иное решение? Наверняка да, и, скорее всего, не однажды. Все мы постоянно советуемся с друзьями и родственниками по самым разным вопросам.

Но есть и профессиональные советчики, к которым мы обращаемся, потому что они являются специалистами по интересующему нас вопросу. Если человек заболел, он обращается к врачу, чтобы тот посоветовал ему, какие лекарства принимать и какие процедуры делать. Если возникают правовые проблемы, идут к юристу. Если возникли проблемы в семье, то обращаются к психологу, специализирующемуся на проблемах семейных отношений, и т. д. Таких людей мы будем называть консультантами или **экспертами**.

Но верно ли, что экспертное заключение может дать только человек в личном общении? Конечно, нет. Существуют справочники и энциклопедии по различным областям знаний, в которых тоже содержатся советы по многим волнующим человека вопросам. Более того, вряд ли всегда можно найти такого же многознающего человека, как, например, какая-нибудь энциклопедия или хорошее руководство. Отличие состоит в том, что эксперту вы задали свой вопрос (или, наоборот, он вам задал какие-то вопросы), после чего эксперт подумал и дал совет. А справочник на помощь не придет — сам ищи, где имеется нужный для тебя совет. Совместить, казалось бы, трудносовместимое — энциклопедичность знаний и актив-

ную реакцию на потребность пользователя — позволяет одна из разновидностей систем искусственного интеллекта, а именно, так называемые **экспертные системы**.

В современном мире информационных технологий экспертные системы находят все большее применение. Конструкторам они помогают подобрать материалы, из которых будут изготавливаться те или иные изделия: ведь здесь надо учесть такие факторы, как прочность, легкость, сочетаемость с другими материалами, простота обработки. Ни один материал не обладает всеми этими свойствами одновременно. Выбрать оптимальный вариант помогают материаловедческие экспертные системы.

Экспертные системы другого вида помогают врачу осуществлять диагностику. Впрочем, «лечить» бывает надо не только людей или животных, но и сложные технические системы.

Эксперту, чтобы дать обоснованный совет, нужны знания. Поэтому каждая экспертная система содержит в качестве ядра **базу знаний** той предметной области, для работы в которой предназначается экспертная система. Вы уже знакомы с понятием базы данных. Теперь же речь идет о базах знаний. Уже из сопоставления названий каждый, наверное, ощущает, что в этих понятиях должно быть что-то родственное. Чтобы в этом разобраться, вспомним (см. § 11), что знания — это информация, хранимая в некотором обобщенном виде и готовая к применению в схожих ситуациях. Можно сказать, что данные — это всегда информация о конкретных объектах, в то время как знания, как правило, выражают общие зависимости между разными признаками объектов. Конечно, данные тоже представляют собой некоторое знание, но это его весьма частный случай.

Отметим, что различие между знаниями и данными проявляется и в языковом аспекте. При записи знаний на формальном языке (а это необходимо, чтобы для работы со знаниями можно было применять компьютер) мы получаем утверждения в форме предикатов с переменными, хотя не исключается, что все переменные в этом предикате связанные. Например, предикат «для любого натурального числа x существует простое число y, такое, что $y > x$» выражает знание, являясь довольно непростой теоремой, хотя известной уже Евклиду. Данные же всегда записываются выражением, не содержащим переменных, например число 101 простое и больше числа 100.

Итак, основу базы знаний составляют описания моделей, выполненные посредством некоторого формального языка, понятного компьютеру. Между этими моделями устанавливаются логические связи, превращающие совокупность моделей, содержащихся в базе знаний, в систему. Как база данных каждая база знаний оснащена системой управления (сокращенно СУБЗ), которая позволяет заполнять и изменять базу знаний (как говорят, редактировать базу), а также отвечать на запросы, направляемые базе знаний.

Другим компонентом экспертной системы является блок, называемый обычно **решателем**. Он моделирует ход рассуждений экс-

перта на основании знаний, имеющихся в базе знаний. По-другому решатель называют **блоком логического вывода.**

Еще одним компонентом экспертной системы является блок, позволяющий пользователю получить ответы на вопросы: «Как была получена данная рекомендация?» и «Почему экспертная система приняла такое решение?» Отвечая на вопрос «Как...», экспертная система предъявляет пользователю шаг за шагом весь процесс получения решения с указанием использованных фрагментов базы знаний. При ответе на вопрос «Почему...» система демонстрирует последнее применение правила вывода, иными словами, последний шаг в процессе принятия решения.

Наконец, для каждой экспертной системы предусмотрен интерфейс, позволяющий пользователю на некотором языке записывать свой запрос к системе, а самой системе давать ответ на этот запрос.

Существуют специальные готовые оболочки экспертных систем, которые позволяют посредством редактора базы знаний заполнить ядро экспертной системы и превратить схемы правил вывода в конкретные правила. Занимается этим специалист, которого обычно называют инженером по знаниям или когнитологом. Можно сказать, что он выступает промежуточным звеном между живыми экспертами и создаваемой базой знаний.

ВОПРОСЫ И ЗАДАНИЯ

1. Что такое экспертная система?
2. Какие основные блоки включает в себя экспертная система? Каково назначение каждого из них?
3. Что составляет основу базы знаний, входящей в состав любой экспертной системы? Для чего служит СУБЗ?
4. В чем проявляются различия между понятиями «данные» и «знания»?
5. * В объяснительном тексте приведены примеры применения экспертных систем в конструкторской и управленческой деятельности, в медицинской практике. Возможно ли применение экспертных систем в научной деятельности? Приведите аргументы в пользу предлагаемого вами ответа.

§39 Реляционная модель экспертной системы

Обычный режим работы любой экспертной системы — диалог с пользователем. Пользователь сообщает в той или иной форме общую постановку задачи, т. е. что он хочет получить в качестве результата. В ответ экспертная система попросит его ввести какие-то исходные сведения об объектах, фигурирующих в задаче. Если «по размышлении» она придет к выводу, что для ответа этих данных недостаточно, то задаст соответствующий вопрос пользователю. Получив ответ, она снова поразмышляет и так шаг за шагом будет продвигаться к нужному ответу. В какой-то момент может оказать-

ся так, что у пользователя нет ответа на вопрос системы. Тогда он вправе попросить предъявить уже достигнутый уровень решения и, более того, весь ход рассуждений, который привел к создавшемуся положению. Впрочем, пользователь и в случае полного решения задачи может получить исчерпывающую информацию о том, как этот ответ был получен. В этом смысле экспертная система может выступать как средство обучения, демонстрируя пользователю то, как нужно рассуждать, чтобы принять требуемое решение.

Экспертных систем на сегодняшний день создано много. В их основу положены разные принципы организации баз данных и правил вывода. Существуют специальные языки программирования, предназначенные для разработки систем искусственного интеллекта, и в частности экспертных систем. Одним из таких языков является Пролог. О нем мы расскажем чуть позже. И вы сможете на этом языке создать несложную экспертную систему. Конечно, если транслятор с этого языка имеется на вашем компьютере. А если его нет, то, чтобы рассказ об экспертных системах не остался сказкой о чем-то, что никак нельзя «потрогать руками» обыкновенному школьнику, мы предлагаем создать небольшую и простую экспертную систему средствами, которые на вашем компьютере наверняка есть. Тогда вы сможете увидеть реализацию тех подходов, которые используются в больших экспертных системах.

Для построения модели экспертной системы прежде всего объявим, что мы будем строить эту модель на основе реляционного подхода. Во-первых, потому, что реляционный подход используется сегодня во многих реальных экспертных системах, во-вторых, мы надеемся, что вы уже хорошо его усвоили, а в-третьих, у нас в распоряжении имеется компьютерная технология реализации реляционного подхода — это Access.

Мы в § 38 говорили, что разным классам задач соответствуют разные типы экспертных систем. Система, которую намерены построить мы, относится к классу идентификационных (или диагностических) систем. Системы этого класса решают задачу определения, т. е. идентификации, объекта по его признакам. Такие системы составляют значительную часть существующих экспертных систем. А в качестве примера, как правило, рассматривается задача определения вида животного или растения. Вот и мы будем проектировать такую ботаническую экспертную систему.

Начнем с того, что сформулируем знания по указанному вопросу. Вот эти формулировки.

Если класс голосемянные и форма листа чешуеобразная, то семейство кипарисовые.

Если класс голосемянные и форма листа иглоподобная и конфигурация хаотическая, то семейство сосновые.

Если класс голосемянные и форма листа иглоподобная и конфигурация — 2 ровных ряда и серебристая полоса, то семейство сосновые.

Если класс голосемянные и форма листа иглоподобная и конфигурация — 2 ровных ряда и серебристой полосы нет, то семейство болотный кипарис.

Если тип деревья и форма листа широкая и плоская, то класс покрытосемянные.

Если тип деревья и неверно, что форма листа широкая и плоская, то класс голосемянные.

Если стебель зеленый, то тип травянистые.

Если стебель древесный и положение стелющееся, то тип лианы.

Если стебель древесный и положение прямостоящее и один основной ствол, то тип деревья.

Если стебель древесный и неверно, что положение прямостоящее и один основной ствол, то тип кустарниковые.

Мы специально выбрали систему, в которой потребуется не так уж много знаний. Но попробуйте, глядя только в выписанные формулировки, ответить на вопрос, какому семейству принадлежит растение, если в руках вы держите один лист, имеющий форму иголки. Скорее всего, за полминуты это не получится (если вы, конечно, не ботаник).

Систему логического вывода мы изобразим орграфом (рис. 4.2). Каждая вершина графа помечена уточняющим вопросом экспертной системы к пользователю или ее ответом на задачу. Для удобства все вершины перенумерованы, начиная с нуля.

Если вершина помечена вопросом экспертной системы, то из нее выходят две дуги. Одна дуга помечена одним ответом пользователя, другая — его альтернативным ответом. Вершина, соответствующая ответу экспертной системы на задачу, не имеет выходящих дуг. Будем в дальнейшем вершину орграфа называть состоянием экспертной системы.

По существу, работа экспертной системы означает «путешествие» по этому орграфу. Она состоит из последовательности однотипных шагов, на каждом из которых пользователь должен решить, по какой дуге он пойдет из очередной вершины.

Исповедуя реляционный подход, мы должны теперь описать этот орграф подходящими таблицами. Сначала составим таблицу 4.19. В ней три столбца: «Начало» и «Конец», где указаны начальная и конечная вершины дуги, а также «Ответ пользователя». Назовем эту таблицу *Орграф*.

Но одной этой таблицы мало, поскольку требуется еще информация о реакциях экспертной системы на ответы пользователя. Реакция же может быть двоякой: ответ системы или очередной вопрос пользователю. Поэтому и таблиц будет две.

Таблица 4.20 *Имена* имеет два атрибута: ***Имя*** и ***Состояние***.

Таблица 4.21 *Вопросы* имеет также два атрибута: ***Состояние*** и ***Вопрос***.

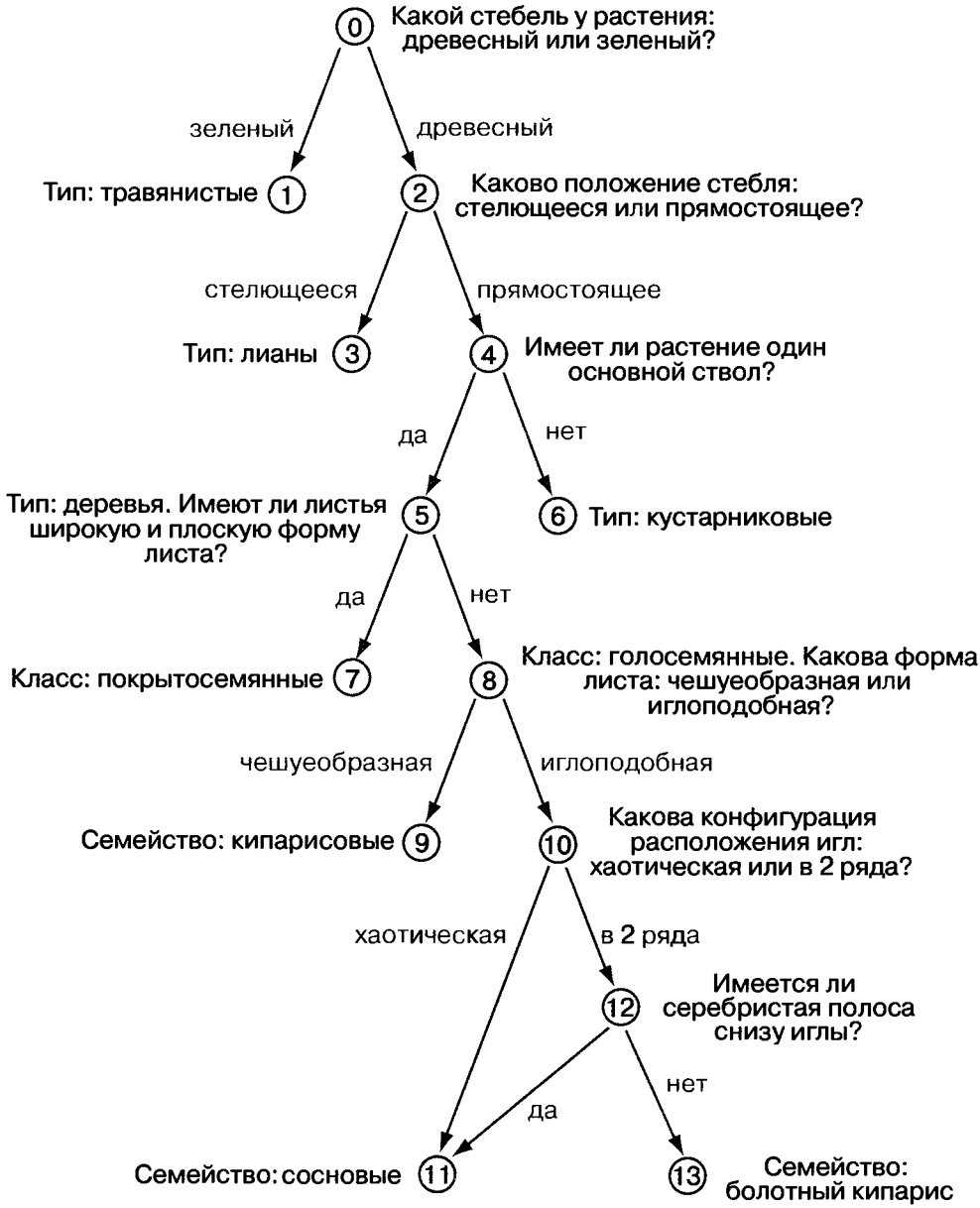


Рис. 4.2 Структура логического вывода в экспертной системе

Теперь надо определить взаимодействие этих таблиц. Для этого нам в каждый момент работы экспертной системы потребуется знать, в каком состоянии она находится. Номер этого состояния

Таблица 4.19

Начало	Конец	Ответ пользователя
0	1	стебель — зеленый
0	2	стебель — древесный
2	3	положение — стелющееся
2	4	положение — прямое
4	5	один ствол — да
4	6	один ствол — нет
5	7	форма листа широкая и плоская — да
5	8	форма листа широкая и плоская — нет
8	9	форма листа — чешуеобразная
8	10	форма листа — иглоподобная
10	11	конфигурация — хаотическая
10	12	конфигурация — 2 ряда
12	11	серебристая полоса — да
12	13	серебристая полоса — нет

Таблица 4.20

Имя	Состояние
тип — кустарниковые	6
тип — травянистые	1
тип — лианы	3
тип — деревья	5
класс — голосемянные	8
семейство — кипарисовые	9
семейство — сосновые	11
семейство — болотный кипарис	13
класс — покрытосемянные	7

будем хранить еще в одной таблице (табл. 4.22), которую назовем *Текущее состояние*.

Сейчас в эту таблицу записано начальное состояние экспертной системы.

Таблица 4.21

Состояние	Вопрос
0	Какой стебель у растения — древесный или зеленый?
2	Каково положение стебля — стелющееся или прямостоящее?
4	Имеет ли растение один основной ствол?
5	Имеют ли листья широкую и плоскую форму?
8	Какова форма листа — чешуеобразная или иглоподобная?
10	Какую конфигурацию имеет расположение игл — хаотическую или в 2 ряда?
12	Имеется ли серебристая полоса снизу иглы?

Таблица 4.22

Состояние
0

По текущему состоянию экспертная система генерирует запрос к таблице *Орграф*. При этом должно быть реализовано соединение таблиц *Текущее состояние* и *Орграф* по атрибутам ***Начало = Состояние***. В зависимости от реакции на этот запрос пользователя экспертная система переходит в следующее состояние и либо выдает ответ, либо генерирует очередной запрос. Изменение состояния экспертной системы организуется с помощью макрокоманд, допустимых той программной оболочкой, которая используется для реализации этой модели экспертной системы. Это уже вопросы технические, и мы рассмотрим их при выполнении лабораторной работы 19.

ВОПРОСЫ И ЗАДАНИЯ

1. Какие основные блоки включает в себя экспертная система? Каково назначение каждого из них?
2. Разработанная нами экспертная система в качестве вершин ответов не всегда указывает на семейство, а завершает свою работу указанием, например, класса растения. Это сделано лишь потому, что система носит чисто иллюстративный характер. Разработайте более полный вариант базы знаний данной экспертной системы.
3. * а) Попытайтесь разработать экспертную систему классификации химических веществ по их свойствам (таких систем может быть несколько в зависимости от выбираемых свойств).

   б) Попытайтесь разработать экспертную систему, советующую пользователю, какой выбрать метод для решения имеющегося у него тригонометрического уравнения.

§ 40 Знакомимся с логическим программированием

Выразительные возможности языка СУБД Access весьма ограниченны. Впрочем, они достаточны для решения того круга задач, на который ориентирована любая база данных. А если вы хотите решать другие задачи, строить другие компьютерные модели, пользуйтесь языком программирования. Хотя бы с одним таким языком вы уже знакомы. Его отличительной чертой является так называемый **процедурный стиль**. Эти слова означают, что компьютер выступает в качестве формального исполнителя, которому нужно подробно и точно описать всю последовательность действий (процедур), необходимую для получения результата.

Теоретики программирования не могли смириться с тем, что компьютеру нельзя сказать: «Подумай сам и получи результат, связанный с исходными данными и известными тебе соотношениями». Например, если тренер предложил вам построиться по росту, то он вряд ли рассказывает вам алгоритм выполнения задачи. Разве что для младших ребятишек первый раз он пояснит, что тот, кто выше, должен стоять правее (т. е. укажет в явном виде связь между исходными данными — вашим ростом — и результатом).

Представлялось соблазнительным предложить такой язык программирования, на котором достаточно было бы описать исходные данные, указать требуемый результат и связи между ними, а построение решения свести к некой стандартной процедуре, которая была бы встроена в транслятор с этого языка.

В 1972 году шотландский ученый Р. Ковальский предложил использовать в качестве такого языка модификацию языка математической логики, оперирующего с предикатами и операциями над ними. Эта идея была практически реализована и обобщена учеными Франции, Великобритании, Португалии, Советского Союза и других стран. Разработанный язык программирования получил название Пролог (от французского PROgramming et LOGicque). И он действительно явился прологом в новом направлении, получившем название **логического программирования**.

Логическое программирование основано на том, что компьютер должен решать задачу в свойственной человеку манере. Оно предполагает, что сведения о задаче и предположения, достаточные для ее решения, формулируются в виде логических аксиом. Эта совокупность представляет собой **базу знаний** задачи.

База знаний может быть использована для решения задачи, если ее постановка формализована в виде логического выражения, подлежащего выводу из аксиом и называемого **целевым утверждением**. База знаний вместе с соответствующим целевым утверждением называются **логической программой**. А собственно построение решения (которое в этом случае называется исполнением программы) сводится к определенной стандартной процедуре, разработанной специалистами-математиками.

База знаний в системе логического программирования делится на две части: описание фактов и описание правил вывода. Как это делается, разберем на небольшом примере большой семьи.

В этой семье есть:

— мама (ее имя Нина);
— папа (его имя Саша);
— сын (его имя Андрей);
— сын (его имя Павел);
— дочь (ее имя Вера);
— сестра мамы (ее имя Ирина);
— брат папы (его имя Юрий);
— брат папы (его имя Игорь);
— отец мамы (его имя Егор);
— мать мамы (ее имя Анна);
— отец папы (его имя Петр);
— мать папы (ее имя Софья);
— жена Андрея (ее имя Татьяна);
— дочь Андрея и Татьяны (ее имя Мария);
— муж Веры (его имя Антон);
— сын Веры и Антона (его имя Виктор).

Остановимся на этом. Видно, что здесь присутствуют следующие отношения:

быть_отцом (x, y),
быть_сыном (x, y),
быть_братом (x, y),
быть_мужем (x, y),
быть_матерью (x, y),
быть_дочерью (x, y),
быть_сестрой (x, y),
быть_женой (x, y).

Каждому из этих отношений соответствует предикат, который принимает значение Истина в том и только в том случае, если значения его переменных таковы, что соответствующие им объекты (в нашем случае — субъекты) находятся в данном отношении.

Например, если предикат, соответствующий отношению

быть_отцом (x, y),

обозначить как Father(x, y), то

Father(Саша, Андрей) = Истина, а
Father(Саша, Мария) = Ложь.

Введем нужные нам предикаты:

быть_отцом (x, y) — Father(x, y),
быть_матерью (x, y) — Mother(x, y),
быть_сыном (x, y) — Son(x, y),
быть_дочерью (x, y) — Daughter(x, y),
быть_братом (x, y) — Brother(x, y),
быть_сестрой (x, y) — Sister(x, y),
быть_мужем (x, y) — Husband(x, y),
быть_женой (x, y) — Wife(x, y).

Известная нам информация тогда запишется так:

Father(Саша, Андрей);
Father(Саша, Павел);
Father(Саша, Вера);

Son(Андрей, Нина);
Son(Андрей, Саша);
Son(Павел, Нина);
Son(Павел, Саша);

Father(Егор, Нина);
Father(Петр, Саша);

Husband(Антон, Вера);

Son(Виктор, Вера);
Son(Виктор, Антон);

Sister(Ирина, Нина);

Mother(Нина, Андрей);
Mother(Нина, Павел);
Mother(Нина, Вера);

Daughter(Вера, Нина);
Daughter(Вера, Саша);

Brother(Юрий, Саша);
Brother(Игорь, Саша);

Mother(Анна, Нина);
Mother(Лидия, Саша);

Wife(Татьяна, Андрей);

Daughter(Мария, Андрей);
Daughter(Мария, Татьяна).

Это факты нашей базы знаний. А теперь запишем правила. Вот первое:

$$\text{Grandfather}(x, y) := \text{Father}(x, z), \text{Father}(z, y)$$

В переводе на обычный язык это правило означает: если x — папа для некоторого z и тот же z — папа для y, то x — дедушка для y. На языке логики предикатов то же самое запишется так:

$$\forall x \, \forall y \, (\exists z \, (\text{Father}(x, z) \,\&\, \text{Father}(z, y)) \rightarrow \text{Grandfather}(x, y)).$$

Как видите, вместо знака конъюнкции пишется запятая, вместо импликации пишется знак присваивания, причем правая и левая части поменялись местами. Предикат, стоящий в левой части правила, называется **головой правила**, а предикаты, стоящие в правой части, образуют **тело правила**.

А вот с кванторами немного сложнее. В языке Пролог кванторы не пишутся, а только подразумеваются. При этом если переменные встречаются в голове правила, то к ним применяются кванторы всеобщности, а если переменные встречаются только в теле правила, то к ним применяются кванторы существования.

У многих, наверно, возник недоуменный вопрос: ведь дедушка — это не только папин папа, но и мамин папа. На языке предикатов это записывается так:

$$\forall x \, \forall y \, (\exists z \, ((\text{Father}(x, z) \,\&\, \text{Father}(z, y)) \vee (\text{Father}(x, z) \,\& \,\& \, \text{Mother}(z, y))) \rightarrow \text{Grandfather}(x, y)).$$

Иными словами, нам нужна операция дизъюнкции. А ее в правилах Пролога нет.

Давайте вспомним, как обошлись с дизъюнкцией разработчики СУБД Access. Очень просто: разнесли в разные строки запрашиваемые параметры. Также и в Прологе: предикаты, соединенные дизъюнкцией, разносятся в разные правила. Значит, мы должны просто записать еще одно правило:

Grandfather(x, y) := Father(x, z), Mother(z, y)

Теперь, когда все условности оговорены, мы будем записывать все утверждения по правилам Пролога. Можно, к примеру, определить предикат, соответствующий отношению *быть_бабушкой* или отношению *быть_внуком*. Совокупность правил, которые вы здесь напишете, и создает «интеллектуальную» начинку разрабатываемой базы знаний «Семья».

О том, как в базе знаний получать информацию, мы расскажем в следующем параграфе. А сейчас продемонстрируем только один запрос:

? Grandfather(x, Вера)

Ответом будет:

x = Петр
x = Егор

Иными словами, Пролог разыскивает те значения x, для которых данный предикат истинен. Поэтому чем больше предикатов вы определите, тем больше запросов сможет обрабатывать эта база знаний.

Правда, для описания фактов пришлось использовать много разных предикатов. Нельзя ли уменьшить их количество? Можно, только предикаты нужны другие. Возьмем, например, такие предикаты:

Man(x) — x является мужчиной;
Woman(x) — x является женщиной.

Тогда легко определить предикаты Son и Daughter:

Son(x, y) := Mother(y, x), Man(x)
Son(x, y) := Father(y, x), Man(x)

Daughter(x, y) := Mother(x, y), Woman(x)
Daughter(x, y) := Father(x, y), Woman(x)

Чуть сложнее определяются предикаты Sister и Brother:

Sister(x, y) := Father(z, x), Father(z, y), Woman(x)
Sister(x, y) := Mother(z, x), Mother(z, y), Woman(x)

Brother(x, y) := Father(z, x), Father(z, y), Man(x)
Brother(x, y) := Mother(z, x), Mother(z, y), Man(x)

Теперь совокупность фактов запишется по-другому:

Father(Саша, Андрей);	Mother(Нина, Андрей); Woman(Нина);	Man(Саша);
Father(Саша, Павел);	Mother(Нина, Павел); Woman(Вера);	Man(Андрей);
Father(Саша, Вера);	Mother(Нина, Вера); Woman(Анна);	Man(Павел);
Father(Егор, Нина);	Mother(Анна, Нина); Woman(Лидия);	Man(Егор);
Father(Петр, Саша);	Mother(Лидия, Саша); Woman(Ирина);	Man(Петр);

Father(Егор, Ирина); Mother(Анна, Ирина); Man(Виктор); Woman(Татьяна);
Father(Андрей, Мария); Mother(Татьяна, Мария); Man(Антон); Woman(Мария).
Father(Антон, Виктор); Mother(Вера, Виктор); Man(Юрий);
Father(Петр, Юрий); Mother(Лидия, Юрий); Man(Игорь);
Father(Петр, Игорь); Mother(Лидия, Игорь).

Но дело не только в простоте. Давайте напишем запрос:

? Son(Юрий, x)

В первой базе знаний ответа не будет, хотя ясно, что Юрий, будучи братом Саши, имеет тех же родителей. А во второй базе благодаря написанным нами правилам для предиката Son мы получим ответ:

x = Петр
x = Лидия

Значит, вторая база знаний нами спроектирована точнее. Как мы уже знаем из § 37, проектирование базы знаний — дело непростое, и занимаются им специалисты-когнитологи.

ВОПРОСЫ И ЗАДАНИЯ

1. В чем состоят отличия процедурного и логического стилей программирования?
2. Каковы основные части базы знаний, создаваемой на языке логического программирования?
3. Каковы составные части правила, записанного в соответствии с требованиями синтаксиса языка Пролог? Как такое правило переводится на язык логики предикатов?
4. Запишите, пользуясь предикатами второй (улучшенной) версии базы знаний «Семья», правила, определяющие предикаты Grandmother(бабушка), Uncle(дядя), Aunt(тетя), CousinM(двоюродный брат), CousinW(двоюродная сестра).

§ 41 Запросы в базе знаний на Прологе

Пусть имеется некоторая база знаний, созданная на языке Пролог, например вторая версия базы знаний «Семья», описанная в предыдущем параграфе. Как и для баз данных, ответ от базы знаний мы получаем, направив ей запрос.

Запросы к базе знаний можно разделить на два типа: запросы на подтверждение факта и запросы на перечисление значений переменных, удовлетворяющих запросу.

Запрос на подтверждение факта выглядит так:

? Son(Юра, Петр)

Ответ будет «да».

Или другой запрос:

? Son(Виктор, Саша)

Ответ будет «нет».

Теперь о запросах второго типа. В таком запросе каждую переменную Пролог воспринимает как требование найти такое значение, чтобы предикат оказался истиной. Например, на запрос

? Grandfather(x, y)

будет дан следующий ответ:

x = Петр, y = Андрей
x = Петр, y = Павел
x = Петр, y = Вера
x = Егор, y = Андрей
x = Егор, y = Павел
x = Егор, y = Вера
x = Саша, y = Виктор
x = Саша, y = Мария

Поймет Пролог и запрос, сформулированный в виде конъюнкции предикатов:

? Grandfather(x, y), Brother(x, z)

Мы в этом запросе спрашиваем, кто является одновременно дедушкой и братом и кому именно. Ответ таков:

x = Саша, y = Виктор, z = Юрий
x = Саша, y = Виктор, z = Игорь
x = Саша, y = Мария, z = Юрий
x = Саша, y = Мария, z = Игорь

Впрочем, вас могут не интересовать подробности, кому именно некто будет дедушкой и в то же время еще кому-то братом. Тогда запрос можно записать так:

? Grandfather(x, _), Brother(x, _)

Символ _ обозначает, как говорят, **пустой аргумент** и фактически исполняет роль квантора существования. Надо только помнить, что, употребленный в разных местах, он обозначает разные переменные. Если предикат, определяющий логическое выражение данного запроса (надеемся, что вы помните: каждый запрос реализуется неким логическим выражением), записать на языке математической логики, то получится так:

$\exists y\, \exists z$ Grandfather(x, y), Brother(x, z),

а не так:

$\exists y$ Grandfather(x, y), Brother(x, y).

Сам механизм поиска ответов на запросы мы рассказывать не будем. Но не потому, что это государственная тайна, просто это относится не к информатике, а к математике.

ВОПРОСЫ И ЗАДАНИЯ

1. Определите, какие ответы будут получены из базы знаний «Семья» на следующие запросы:
   а) ?Grandmother(x, Мария)
   б) ?Uncle(Игорь, Виктор)
   в) ?CousinW(Мария, x)
   г) ?Uncle(_, x), Woman(x)
2. Запишите запросы по правилам языка Пролог для получения следующей информации из базы знаний «Семья».
   а) Кто является внуками Лидии?
   б) Кто является тетей?
   в) У кого есть братья?
   г) Кто отец племянницы Юрия?
   д) У какого деда есть братья?

§ 42 Встроенные предикаты в логических языках программирования. Простейшие программы

В § 40 мы, строя базу знаний «Семья», сами определяли все нужные для этой базы предикаты: часть из них определялась совокупностью фактов, другая часть — правилами вывода. Но представьте себе, как было бы неудобно, работая с процедурными языками программирования, каждый раз писать, например, алгоритм сложения двух чисел, или алгоритм вычисления значения функции $\sqrt{x}$, или алгоритм конкатенации двух слов. От всей этой работы вы освобождены благодаря тому, что указанные операции и функции встроены в язык.

Так же и в логических языках программирования: наиболее употребительные предикаты имеются в самом языке. Например, для организации простейших арифметических операций имеется встроенный предикат, который мы назовем Linear(x, y, z, v); в реальных языках логического программирования он может называться по-другому. Этот предикат принимает значение Истина в том и только в том случае, когда его аргументы связаны соотношением $v = xy + z$. На основе этого предиката легко определить предикаты, позволяющие подсчитывать сумму и произведение двух чисел:

Sum(x, y, z) := Linear(x, 1, y, z),
Mult(x, y, z) := Linear(x, y, 0, z).

Обратите внимание, что переменные в описании предиката — это формальные параметры и вместо них, как обычно, могут быть

записаны имена любых других переменных того же типа или соответствующие константы. В нашем случае мы воспользовались числовыми константами 0 и 1.

В каждом языке логического программирования есть предикат сравнения. Будем считать, что он записывается так: *Comparison*(x, y, z), где x и y — числовые или символьные переменные, а z — логическая переменная, принимающая два значения: Истина и Ложь. Этот предикат дает значение Истина в том и только в том случае, когда $x < y$ и z имеет значение Истина (в случае символьных переменных это означает, что значение переменной x по алфавиту располагается раньше значения переменной y). На основе этого предиката легко определить предикаты, соответствующие отношению «меньше», «не меньше», «равно», «больше»:

Less(x, y) := Comparison(x, y, Истина);
Notless(x, y) := Comparison(x, y, Ложь);
Equal(x, y) := Comparison(x, y, Ложь), Comparison(y, x, Ложь);
More (x, y) := Less(y, x).

Встроенные логические предикаты (в них все переменные логические) отрицания и конъюнкции мы запишем так:

Not(x, y) истинен тогда и только тогда, когда x — Истина, а y — Ложь или x — Ложь, а y — Истина;

And(x, y, z) истинен тогда и только тогда, когда x — Истина, y — Истина и z — Истина.

Благодаря законам де Моргана предикат дизъюнкции определяется так:

Or(x, y, z) := Not(x, v), Not(y, w), And(v, w, s), Not(s, z).

А теперь попрограммируем. Тип переменных, входящих в предикаты (целый, вещественный, символьный и т. д.), объявляется, как и в процедурных языках, в начале программы. Однако мы не будем этого делать, поскольку все создаваемые нами программы носят иллюстративный характер и обычно понятно по умолчанию, переменные какого типа в них используются.

Пусть требуется найти объем прямоугольного параллелепипеда с ребрами 10, 20 и 30. Пишем предикат, определяющий объем:

Vol(a, b, c, v) := Mult(a, b, z), Mult(z, c, v).

И делаем запрос:

? Vol(10, 20, 30, v)

Легко предсказать, какой ответ даст Пролог на этот запрос. Но заметьте, что для решения другой задачи — найти высоту прямоугольного параллелепипеда длины 10, ширины 20 и объема 600 — мы обращаемся к тому же предикату, но с другим запросом:

? Vol(10, 20, c, 600)

Это и понятно — база знаний о том, что такое объем прямоугольного параллелепипеда, не меняется. А какие задачи вы будете решать — дело ваше.

Можно сказать так, что база знаний — это модель объекта, процесса или явления, записанная на языке логического программирования, а запросы к ней — это использование данной модели для решения конкретных задач.

А теперь запрограммируем нахождение наибольшего общего делителя двух натуральных чисел a и b. О НОД(a, b) мы знаем следующее:

1) НОД(a, a) = a;
2) Если $a < b$, то НОД(a, b) = НОД(a, $b - a$);
3) Если $a > b$, то НОД(a, b) = НОД($a - b$, b).

Вот и запишем это с помощью уже известных нам предикатов:

LCD(a, a, a);
LCD(a, b, c) := Less(a, b), Sum(a, y, b), LCD(a, y, c);
LCD(a, b, c) := More(a, b), Sum(x, b, a), LCD(x, b, c).

В этой базе знаний один факт и два правила вывода. Сделаем к этой базе знаний запрос:

? LCD(20, 35, x)

Поскольку $20 \neq 35$, под факт наш запрос не попадает. Не попадает он и под второе правило, поскольку значение предиката More(20, 35) — Ложь. А в первом правиле предикат Less(a, b) истинен, поэтому разыскивается такой y, чтобы был истинен предикат Sum(20, y, 35). Такое значение y только одно: 15. Поэтому надо разыскивать такое x, чтобы был истинен предикат LCD(20, 15, x). Для поиска такого значения переменной x снова обращаемся к базе знаний. Теперь придется воспользоваться вторым правилом и разыскивать x, для которого истинен предикат LCD(5, 15, x). Следующий шаг приведет к поиску x, для которого истинен предикат LCD(5, 10, x). Еще один шаг приводит к предикату LCD(5, 5, x). И тут происходит обращение к факту, и получается ответ: $x = 5$.

Разобравшись в этом процессе, легко понять, что в базе знаний нами было записано **рекурсивное правило**, т. е. правило, в котором предикат, записанный в голове, фигурирует и в теле этого правила. Впрочем, как и для алгоритмов, рекурсия может создаваться не одним правилом, а некоторой их совокупностью.

В реальном языке логического программирования есть много других встроенных предикатов, в том числе позволяющих строить графические изображения на экране компьютера. Мы, однако, не преследуем цель научить вас программированию на языке Пролог, но надеемся, что представление о таком программировании вы получили.

ВОПРОСЫ И ЗАДАНИЯ

1. Для чего нужны встроенные предикаты?
2. Какое правило называется рекурсивным?
3. а) Используя встроенные и определенные в объяснительном тексте предикаты, определите предикат Square(x, y), позволяющий возводить число x в квадрат.

   б) Требуется извлечь квадратный корень из числа. Можно ли для этого воспользоваться предикатом Square(x, y), определенным вами при выполнении задания из пункта *а*?

   в) Используя предикат Square(x, y), составьте запросы, позволяющие узнать, можно ли извлечь квадратный корень из числа 16, из числа 0, из числа –1.
4. а) Используя встроенные и определенные в объяснительном тексте предикаты, определите предикат Cube(x, y), позволяющий возводить число x в куб.

   б) Используя предикат Square(x, y) из задания 1 и предикат Cube(x, y), составьте запрос, позволяющий находить числа, меньшие 1000, которые одновременно являются квадратом и кубом некоторых целых чисел. Считайте, что в этих предикатах переменные объявлены имеющими целый тип.
5. а) Используя встроенные и определенные в объяснительном тексте предикаты, постройте базу знаний, позволяющую для каждого числа находить его абсолютную величину.

   б) Постройте базу знаний, позволяющую из двух чисел выбрать большее.

   в)* Постройте базу знаний, позволяющую из трех различных чисел выбрать то, которое расположено между двумя другими.
6. Используя встроенные и определенные в объяснительном тексте предикаты, постройте базу знаний, позволяющую для числа $a \neq 0$ и целого неотрицательного числа n вычислить a^n.
7. Последовательностью Фибоначчи называется последовательность чисел 1, 1, 2, 3, 5, 8, ..., в которой каждый член, начиная с третьего, равен сумме двух предыдущих.

   а) Постройте базу знаний для предиката Fib(n, x), позволяющего по номеру n находить n-й член x данной последовательности.

   б) Укажите, какие ответы будут получены на запросы:
   а) ?Fib(n, 55), б) ?Fib(8, 16), в) ?Fib(0, 0), г) ?Fib(–1, x), д) ?Fib(n, –1).

ИТОГИ ГЛАВЫ 4

Повторим кратко то основное, что вы узнали, изучив главу 4.

Наука, изучающая закономерности рассуждений, называется логикой. Ее раздел, изучающий формальные модели логических рассуждений, получил название математической логики. Расширение сферы исследований на всю интеллектуальную деятельность человека привело к построению моделей, которые стали называть системами искусственного интеллекта. Любая система искусственного интеллекта как одна из разновидностей информационных

моделей, обязательно содержит набор параметров, описывающих существенные факторы. Поскольку система искусственного интеллекта направлена на принятие того или иного решения, то в качестве связей между параметрами, описывающими исходные данные, и результатом выступают так называемые правила вывода. Их суть в том, что они указывают, какое решение должно быть рекомендовано в рамках данной модели при тех или иных значениях исходных параметров.

Повествовательное предложение, в отношении которого имеет смысл говорить о его истинности или ложности, называется высказыванием. Над высказываниями можно производить логические операции: *конъюнкции* (операция *и*), *дизъюнкции* (операция *или*), *отрицания* (операция *не*), *импликации* (операция *следования*), *равносильности*. Результат применения каждой из логических операций задается с помощью таблицы истинности.

Из переменных, принимающих логические значения Истина и Ложь, с помощью логических операций составляются формулы. Формулы называются тождественно равными или равносильными, если при подстановке вместо переменных всевозможных значений получаются равносильные высказывания. Это означает, что формулы равносильны в том и только в том случае, когда у них одинаковы таблицы истинности. Формула называется тождественно истинной или тавтологией, если она принимает значение Истина при любых значениях входящих в нее переменных.

По каждой таблице истинности можно написать формулу, для которой таблица истинности совпадает с данной. Формула, получающаяся применением стандартного для этой цели алгоритма, имеет вид дизъюнкции нескольких конъюнктивных выражений, в которых знаком конъюнкции соединены переменные или их отрицания, причем в каждом конъюнктивном выражении присутствует каждая переменная, и притом ровно один раз. Такой вид называется совершенной дизъюнктивной нормальной формой (СДНФ).

Современные базы данных, реализующие фактографические модели, в большинстве своем создаются на основе реляционного подхода. Суть этого подхода заключается в том, что информация об объектах представляется в виде отношений, т. е. связанных между собой характеристик изучаемых объектов. Каждое отношение имеет имя и набор аргументов, которые в теории баз данных называются атрибутами (в СУБД Access атрибуты реализуются подходящим полем). Количество атрибутов в отношении называется его арностью.

Отношение называется **функциональным**, если его атрибуты можно разбить на две группы так, чтобы значения первой группы атрибутов однозначно определяли значения второй группы. Иными словами, первая группа атрибутов может рассматриваться как аргумент некоторой функции, а вторая группа определяет значение этой функции. Если отношение функционально, то набор атрибу-

тов, относящихся к аргументу функции, задаваемой этим отношением, называется ключевым. Набор атрибутов является ключевым, если в таблице нет двух строк с одинаковым набором значений этих атрибутов.

Логической функцией или, по-другому, **предикатом** на множестве M называют такую функцию от нескольких аргументов, которая при любом наборе значений этих аргументов из множества M принимает одно из двух значений: Истина или Ложь. Переменные, которые используются при записи предиката в виде логического выражения, могут быть свободными или связанными посредством кванторов всеобщности и существования.

Над логическими функциями можно выполнять те же операции, что и над высказываниями. При построении отрицания действует правило: каждый квантор, фигурирующий в записи логического выражения, заменяется на двойственный, а бескванторная часть заменяется ее отрицанием.

Выбор из всех наборов значений атрибутов, находящихся в данном отношении, тех, для которых некоторая заданная логическая функция принимает значение Истина, называют фильтрацией, а саму эту функцию называют фильтром.

При работе с многотабличной базой данных важную роль играет операция соединения таблиц по заданным атрибутам. Тип связи между атрибутами соединяемых таблиц зависит от того, объявлены ли эти атрибуты ключевыми. Если один из связываемых атрибутов ключевой, а другой нет, то устанавливается связь типа «один-ко-многим», если же оба связываемых атрибута ключевые, то устанавливается связь типа «один-к-одному».

Одной из разновидностей систем искусственного интеллекта являются экспертные системы. Каждая экспертная система содержит в качестве ядра базу знаний той предметной области, для работы в которой она предназначена. Другой компонент экспертной системы — решатель. На основании сведений, имеющихся в базе знаний, он моделирует ход рассуждений эксперта. По-другому решатель называют блоком логического вывода.

На основе языка математической логики созданы языки логического программирования. Логическое программирование основано на том, что сведения о задаче и предположения, достаточные для ее решения, формулируются в виде логических аксиом. Эта совокупность представляет собой базу знаний задачи. Постановка самой задачи формализована в виде логического выражения, подлежащего выводу из аксиом и называемого целевым утверждением. База знаний и соответствующее целевое утверждение называется логической программой. При этом построение решения (которое в этом случае называется исполнением программы) сводится к определенной стандартной процедуре. Языки логического программирования активно используются для создания систем искусственного интеллекта. Первым языком логического программирования явился язык Пролог.

ПРОВЕРЬ СЕБЯ

Часть 1. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и номер правильного ответа. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

А3. Высказывание $(X \to Y)\ \&\ (\overline{X} \to \overline{Y})$ равносильно высказыванию:

1) $X\ \&\ Y$; 2) $X \vee Y$; 3) $X \oplus Y$; 4) $X \leftrightarrow Y$.

А2. Отрицанием к высказыванию «Для любого натурального числа, оканчивающегося цифрой 3, существует кратное ему число, оканчивающееся цифрой 1» служит высказывание:

1) Нет натурального числа, оканчивающегося цифрой 3, для которого существовало бы кратное ему число, оканчивающееся цифрой 1.

2) Существует натуральное число, оканчивающееся цифрой 3, для которого любое кратное ему число не оканчивается цифрой 1.

3) Любое натуральное число, оканчивающееся цифрой 3, не имеет кратного числа, оканчивающегося цифрой 1.

4) Отличное от указанных в пунктах 1—3.

А3. Таблицу истинности, совпадающую с таблицей 4.23, имеет формула:

1) $((X \to Y) \to X) \vee \overline{X}\ \&\ \overline{Y}$;

2) $\overline{X \to Y} \vee (\overline{X} \vee Y)\ \&\ X$;

3) $(Y \to X)\ \&\ (\overline{X} \vee Y)$;

4) отличная от указанных в пунктах 1—3.

Таблица 4.23

X	Y	$F(X, Y)$
И	И	И
И	Л	И
Л	И	Л
Л	Л	Л

А4. Отрицанием к высказыванию $\exists x\, \forall y\, (x^2 + y^3 = 1)$ служит высказывание:

1) $\forall x\, \exists y\, (x^2 + y^3 = 1)$; 2) $\forall y\, \exists x\, (x^2 + y^3 \neq 1)$; 3) $\forall x\, \exists y\, (x^2 + y^3 \neq 1)$;

4) отличное от указанных в пунктах 1—3.

А5. Формула, принимающая при всевозможных значениях логических переменных значение Истина, называется:

1) правильной;

2) совершенной дизъюнктивной нормальной формой;

3) тавтологией;

4) по-другому, нежели в пунктах 1—3.

А6. В совершенной дизъюнктивной нормальной форме представлена формула от трех переменных X, Y и Z:

1) $X \,\&\, Z \vee \overline{X} \,\&\, \overline{Y} \vee Y \,\&\, Z$;

2) $X \,\&\, \overline{Y} \,\&\, Z \vee \overline{X} \,\&\, \overline{Y} \,\&\, \overline{Z} \vee \overline{X} \,\&\, Y \,\&\, Z$;

3) $X \,\&\, \overline{Y} \,\&\, Z \,\&\, Y \vee \overline{X} \,\&\, \overline{Y} \,\&\, \overline{Z} \,\&\, X \vee \overline{X} \,\&\, Y \,\&\, Z \,\&\, \overline{Z}$;

4) среди вариантов, записанных в пунктах 1—3, нет формулы в СДНФ.

А7. Если соединение таблиц в СУБД Access осуществляется по ключевым атрибутам, то устанавливаемая связь имеет тип:

1) один-к-одному;

2) один-ко-многим;

3) многие-ко-многим;

4) между такими атрибутами нельзя установить связь.

А8. На награждение отличившихся работников компании собираются прийти президент компании, генеральный директор или ведущий менеджер. Уже известно, что:
а) если на награждение не придет президент или придет ведущий менеджер, то обязательно придет генеральный директор;
б) если придет генеральный директор, то придет и ведущий менеджер.
Наименьшее число представителей управления компанией, которые придут награждать отличившихся работников,

1) 1; 2) 2; 3) 3;

4) нельзя определить по заданным условиям.

А9. Высказывание $\exists x \, \forall y \, (x^2y^3 \leqslant 0)$ можно прочитать так:

1) Хотя бы для одного значения x найдется значение y, чтобы выполнялось неравенство $x^2y^3 \leqslant 0$.

2) При некотором значении x для каждого значения y выполняется неравенство $x^2y^3 \leqslant 0$.

3) Каким бы ни было значение y, найдется значение x, для которого $x^2y^3 \leqslant 0$.

4) При каждом значении x для любого значения y имеет место неравенство $x^2y^3 \leqslant 0$.

А10. Термин «реляционный» происходит от слова:

1) функция; 2) отношение; 3) связь;

4) другого слова, нежели в пунктах 1—3.

Таблица 4.24

а)

№ п/п	Фамилия	Имя	Класс	Школа
1	Смирнов	Николай	9	15
2	Иванов	Вадим	9	28
3	Белов	Сергей	10	135
4	Петров	Павел	11	36
5	Сидоров	Андрей	11	72
6	Смирнов	Алексей	11	135

б)

№ п/п	Школа	Район
1	135	Центральный
2	72	Юго-западный
3	36	Юго-западный
4	28	Центральный
5	15	Юго-западный
6	103	Западный

А11. В таблицах 4.24, *а* и 4.24, *б* дан фрагмент базы данных. Эти таблицы соединены по атрибуту Школа. Согласно запросу по признаку Район = Юго-западный сформирована таблица-результат со следующими атрибутами, причем записи в ней отсортированы по номерам школ в порядке возрастания.

№ п/п	Фамилия	Имя	Класс

После исполнения запроса в строке с номером 1 таблицы-результата будет стоять фамилия:

1) Иванов; 2) Петров; 3) Сидоров;

4) другого ученика.

А12. В таблице КП 4.24, *а* ключевым атрибутом является:

1) Фамилия; 2) Имя; 3) Класс; 4) Школа.

Часть 2. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и ответ в виде последовательности символов. Сверьте получившуюся у вас запись с ключами на с. 269.

В1. В таблице для некоторого атрибута нет двух строк, в которых значения этого атрибута совпадали. Этот атрибут является _____.

В2. На множестве натуральных чисел рассматривается предикат $\forall x\, \exists y\, (y < zx^2 - 3x)$. Количество значений переменной z, для которых он ложен, равно _____.

В3. На множестве слов рассматривается высказывание $\exists x\, \forall y\, (x + y = y + x)$. Определите, истинно оно или ложно, и закончите предложение: «Данное высказывание _____».

B4. Ниже приведено несколько повествовательных предложений:

1) В бесконечной десятичной дроби, представляющей число 1/7, нет двух рядом стоящих одинаковых цифр.

2) Четырехугольник является квадратом.

3) На столе лежит яблоко.

4) В некотором царстве, в некотором государстве живет царевна-лягушка.

5) Хочешь быть счастливым, будь им.

6) Кому-то на Руси жить хорошо.

Запишите номера тех предложений, которые являются высказываниями, в порядке возрастания без знаков препинания и пробелов.

5 ГЛАВА
Информационные модели в задачах управления

До сих пор мы строили модели процессов, считая, что факторы, влияющие на функционирование объекта, неизменны. Они заданы самой природой и не подвергаются воздействиям человека. На самом деле человек, познавая природу и общество, все активнее и шире вмешивается в действие этих факторов, иногда не осознавая своего влияния, но чаще сознательно, преследуя конкретную цель — заставить тот или иной объект функционировать нужным человеку образом.

Воздействие на объект или процесс, имеющее своей целью получение требуемых значений параметров этого объекта или процесса, называется **управлением**.

В этой главе мы на примерах покажем, как информационные технологии и компьютерное моделирование могут применяться для решения задач управления.

§43 Что такое управление

Выше мы уже сформулировали определение управления. Но чтобы хорошо понимать его суть и уметь самим осуществлять управление, надо знать основные принципы и способы его организации. Общие закономерности управления в системах различной природы и принципы его организации изучаются **кибернетикой.** Такое название дал этой науке один из ее основоположников — Норберт Винер.

Для простоты будем считать, что управляющее воздействие оказывает некий объект. Мы надеемся, что никого не смутит то обстоятельство, что под словом «объект» может подразумеваться человек (например, шофер), автоматическое устройство (например, светофор), организация (например, Министерство внутренних дел, устанавливающее правила дорожного движения на территории России) и т. п. Простейшая ситуация управления, когда взаимодей-

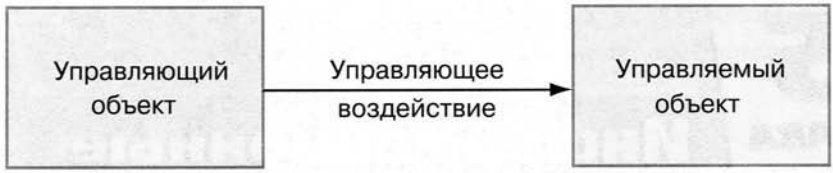


Рис. 5.1 Схема взаимодействия двух объектов в процессе управления

ствуют два объекта: управляющий и управляемый. Такое взаимодействие можно изобразить в виде схемы (рис. 5.1).

Физически управляющее воздействие может быть реализовано по-разному. Управление телевизором его владелец осуществляет нажатием кнопок и вращением ручек настройки, управление движением автотранспорта светофор осуществляет посредством цветовых сигналов с заранее оговоренным смыслом каждого из них, управление собакой хозяин осуществляет голосовыми командами. Говоря об управлении вообще, необходимо отвлечься от физической формы управляющего воздействия. Тогда становится ясно, что суть управляющего воздействия — это передача информации от управляющего объекта к объекту управления.

Управляющая информация может быть представлена инструкцией о том, что должен делать объект управления. Если объект управления — формальный исполнитель, то такая инструкция, как вы знаете, обычно представляет собой алгоритм. Алгоритмическое управление формальным исполнителем является одним из важнейших вариантов реализации управления, особенно часто встречающимся в технических системах. Но в живой природе и обществе далеко не всякое управление сводится к алгоритмическому; существуют и другие формы управления, о которых мы расскажем в последующих параграфах.

Телевизор как объект управления вместе с допустимыми для него воздействиями один, а управлять им могут разные люди, причем с совершенно различными целями — одному хочется смотреть фильм, а другому — спортивную программу; для одного звук слишком громкий, а для другого, наоборот, тихий. Та или иная социальная группа (например, учащиеся одного класса) как объект управления тоже может подвергаться воздействию разных управляющих объектов (в том числе и находящихся внутри этой группы) с весьма разными целевыми установками. Экономический регион как объект управления тоже может подвергаться управляющему воздействию со стороны совершенно различных управляющих объектов. Отвлекаясь опять-таки от физической реализации управляющего объекта (а это, как отмечалось выше, естественно при теоретическом рассмотрении понятия управления), мы видим, что каждый управляющий объект характеризуется *целевыми установками и своими возможностями воздействовать на управляемый объект*. Что же касается объекта управления, то он характеризу-



Рис. 5.2 Схема управления объектом

ется *своими функциональными возможностями* (т. е. тем, что он может сделать или в какое состояние перейти) *и допустимыми на него воздействиями.*

Допустимые воздействия для каждого объекта, как правило, предопределены самой его природой. С кибернетической точки зрения воздействие на объект управления — это восприятие им информации через специальные входы. Реакция объекта на входную информацию, т. е. изменение его состояния, тоже описывается соответствующей информацией. Можно считать, что эта информация подается на тот или иной выход: именно благодаря этой информации управляющий объект может судить о том, достигнута ли поставленная цель. Поэтому схема управления выглядит так, как показано на рисунке 5.2.

Важно иметь в виду, что говорить об управлении можно только применительно к **динамическим** системам, т. е. системам, меняющимся во времени или как-то функционирующим. Ведь если система статична, то ею нельзя управлять — что бы вы с ней ни делали, она не откликается на ваши воздействия; ее можно только сломать. Даже когда речь идет об управлении автомобилем (который сам, конечно, в результате управления не изменяется), то на самом деле рассматривается движение автомобиля по некоторой дороге, т. е. фактически управляемым объектом выступает система «автомобиль + дорога».

В следующих параграфах мы на примерах покажем, как информационные технологии и компьютерное моделирование могут применяться для решения задач управления.

ВОПРОСЫ И ЗАДАНИЯ

1. Что такое управление объектом или процессом?

2. Ниже приведено несколько ситуаций, в которых осуществляется управление. Укажите, в чем состоит процесс управления: попытайтесь определить цель управления, управляющий и управляемый объекты, допустимые воздействия на управляемый объект.
   а) Движение поезда метро.
   б) Объявление начала посадки в самолет.
   в) Ловля рыбы на удочку.
   г)* Обработка детали на фрезерном станке.
   д)* Лечение пациента.

3. Среди процессов, перечисленных ниже, укажите процессы управления.
а) Сталевар разливает металл в ковши.
б) Следователь допрашивает подозреваемого.
в) Кандидат в депутаты выступает на митинге с призывом голосовать за него.
г) Художник рисует пейзаж.
д) Режиссер снимает фильм.

4. Почему управлять можно только динамическими системами?

5. Для каждого из следующих объектов управления укажите, что для него является допустимыми воздействиями:
а) утюг; б) автомат по продаже железнодорожных билетов;
в) продавец мороженого; г)* ваш класс.

§44 Сколько можно взять у природы

Богата российская земля различными ресурсами. Ее недра хранят самые разнообразные полезные ископаемые. Ее леса служат источником для работы лесоперерабатывающей промышленности. В озерах и реках водится рыба...

Вот о ловле рыбы и пойдет у нас речь. Не на удочку, конечно, а в промышленных масштабах.

Рыба с промышленной точки зрения относится к так называемым возобновляемым ресурсам — вы ее ловите, а она снова рождается и растет в том же водоеме. В отличие, скажем, от железной руды, которую если один раз добыл, то больше ее на этом же месте не найдешь. Поэтому руды получили название невозобновляемых ресурсов. Но о них мы здесь говорить не будем.

Рыба рождается ежегодно, но растет не так уж быстро, и если сразу выловить ее слишком много, то можно остаться без рыбы. Возникает задача управления: сколько рыбы можно вылавливать ежегодно, чтобы обеспечить ее нормальное воспроизводство?

Мы знаем модель прироста растительной массы без вмешательства человека — это модель ограниченного роста (см. § 21). Кроме того, мы видели, что управляемый объект удобно изображать в виде предмета, имеющего входы и выходы. Изобразим и рассматриваемую нами рыбную популяцию в виде схемы (рис. 5.3).

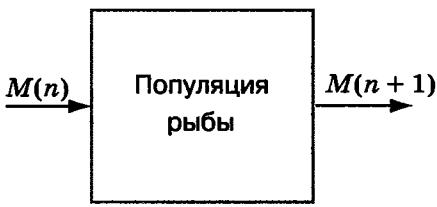


Рис. 5.3 Модель изменения возобновляемого ресурса

Входы и выходы — это масса рыбы в n-м году и соответственно спустя год. И если рыбу никто не ловит, то эта масса меняется по следующему рекуррентному закону:

$$M(n+1) = \left(1 + k\frac{L - M(n)}{L - M(0)}\right)M(n),$$

где $M(0)$ — масса рыбы в начальный момент времени; L — предель-

ное значение массы рыбы для данного водоема; k — коэффициент ежегодного прироста.

Теперь у нас появился еще один вход — воздействие человека R (рис. 5.4). Если мы считаем, что ежегодно вылавливается одна и та же масса рыбы R, то наша модель ограниченного роста изменится по внешнему виду незначительно:



Рис. 5.4 Модель потребления возобновляемого ресурса

$$M(n+1) = \left(1 + k\frac{L - M(n)}{L - M(0)}\right)M(n) - R.$$

Построенную модель назовем **моделью потребления возобновляемых ресурсов.**

ВОПРОСЫ И ЗАДАНИЯ

1. Приведите примеры управления какими-либо объектами или процессами.
2. Какие предположения положены в основу модели потребления возобновляемых ресурсов?
3. а) Подготовьте заполнение электронной таблицы для вычисления массы рыбы через заданное количество лет, используя модель потребления возобновляемых ресурсов.
   б) Составьте алгоритм для вычисления массы рыбы через заданное количество лет, используя модель потребления возобновляемых ресурсов.

§45 Задача о лесопарке

В окрестностях города Екатеринбурга по берегам озера Шарташ раскинулся замечательный лесопарк «Каменные палатки». Когда-то он был практически за городом, но теперь около него возник огромный жилой микрорайон, и с лесопарком стало происходить что-то неладное.

Причина понятна — резко возрос поток посетителей и парк уже не справляется с их нашествием. Подобная проблема характерна для любого крупного города.

Управлять потоком посетителей лесопарка невозможно. Но есть места (природные парки, заповедники и заказники), куда можно ограничить вход любителей отдохнуть на природе. Возникла управленческая задача: как организовать посещение природного парка, чтобы не нанести ему непоправимого ущерба?

Решение задачи надо начинать с построения модели. Нужно прежде всего решить, что значит «организовать посещение». Можно это понимать так: парк открыт для посещения подряд несколь-

ко дней (обозначим это число K) и в эти дни его ежедневно посещает N человек. Потом парк закрывается до практически полного его восстановления.

Теперь об ущербе, наносимом посетителями. Любая природная система способна к самовосстановлению. Ученые-экологи научились измерять ущерб в процентах к исходному состоянию и установили, что за ночь восстанавливается примерно 30% нанесенного ущерба. Если ущерб составляет 80%, то такая ситуация становится непоправимой — парк погибает. Для другого парка в других условиях обе цифры могут быть другими.

Итак, пусть за один день посетители нанесли парку некоторый ущерб. Обозначим его величину через A и будем считать, что она зависит только от числа посетителей N и не меняется в зависимости от дня посещения.

Прошел день, и прошла ночь — парк встречает новых посетителей уже с ущербом; уровень его состояния оценивается на $100\% - (100\% - 30\%)A$. Если мы не хотим все время писать значок процентов, то можем уровень состояния выражать просто в долях от 1. Формула уровня состояния парка к началу второго дня получится такая:

$$1 - (1 - 0{,}3)A.$$

Нетрудно понять, что к началу третьего дня уровень состояния будет уже

$$(1 - (1 - 0{,}3)A)^2,$$

к началу четвертого —

$$(1 - (1 - 0{,}3)A)^3$$

и т. д. К началу последнего K-го дня уровень состояния будет

$$(1 - (1 - 0{,}3)A)^{K-1}.$$

Иными словами, если мы допустим посетителей еще на один день, то ущерб станет непоправимым, т. е. уровень состояния, равный $(1 - (1 - 0{,}3)A)^K$, будет ниже чем $1 - 0{,}8$.

Итак, если нам известно A, то значение K — это наибольший показатель степени в выражении $(1 - (1 - 0{,}3)A)^K$, для которого данное выражение остается больше чем 0,2.

Теперь надо решить задачу, как величина A связана с числом посетителей, т. е. определить A как функцию от N.

Для этого надо снова провести исследования. Что ж, в управлении без науки осмысленного шага не сделаешь. Прикидки на глазок да надежда на авось до добра не доводят.

Опять-таки экологи установили, что величина A хорошо описывается функцией $b + c\sqrt{N}$. Вот только b и c для каждой местности свои. Известны также для нашего парка показатели ущерба (табл. 5.1).

Теперь надо подобрать числа b и c так, чтобы значение A, вычисленное по формуле, не сильно отличалось от значений, приве-

денных в таблице. Воспользуемся снова электронной таблицей. Пример ее заполнения показан в таблице 5.2.

Во второй клетке второго и третьего столбцов будут подбираться значения b и c, для которых наибольшее из отклонений будет минимальным. Подобным вы уже занимались, когда исследовали броуновское движение частицы.

После того как будет найдена формула для A, мы сможем находить значение K по N и, наоборот, N по K. Но этим вы займетесь на лабораторной работе.

Таблица 5.1

Число посетителей	Ущерб, %
30	7,0
50	10,8
100	18,3
120	20,7
180	27,0
250	33,0
400	43,5
600	55,0

Таблица 5.2

	A	B	C	D	E
1	N	b	c	Ущерб	Отклонение от значения, вычисленного по формуле
2	30			0,07	|B2+C2*$\sqrt{A2}$–D2|
3	50	Макс.	max(E2:E9)	0,108	|B2+C2*$\sqrt{A3}$–D3|
4	100			0,183	|B2+C2*$\sqrt{A4}$–D4|
5	120			0,207	|B2+C2*$\sqrt{A5}$–D5|
6	180			0,27	|B2+C2*$\sqrt{A6}$–D6|
7	250			0,33	|B2+C2*$\sqrt{A7}$–D7|
8	400			0,435	|B2+C2*$\sqrt{A8}$–D8|
9	600			0,55	|B2+C2*$\sqrt{A9}$–D9|

ЗАДАНИЕ

Пусть уже найдены коэффициенты b и c в формуле для A. Продумайте, как по заданному числу ежедневных посетителей N вычислить число K, показывающее, сколько дней подряд можно посещать парк. И наоборот, как по заданному числу K найти ежедневное количество посетителей парка.

§46 Учимся у природы правильной организации управления

В предисловии к этой главе мы определили управление как целенаправленное воздействие на те или иные факторы объекта или процесса. Поэтому название параграфа на первый взгляд выглядит странно: природа в своем существовании не преследует какие-либо цели, поэтому и управлять природа ничем и никем не может. Как же тогда учиться управлению у природы?

Наблюдая природные явления, мы можем «подсмотреть» у нее те механизмы, которые позволяют природе в течение тысячелетий существовать и развиваться. Ведь и человек обычно стремится к тому, чтобы его деятельность тоже могла обеспечить устойчивое существование и развитие. Это не всегда ему удается. Попробуем разобраться почему.

Вспомните свойство живой природы, которое вы обнаружили, выполняя лабораторную работу 22, — явление *саморегулирования*: природа стремится восстановить то стабильное состояние, из которого ее вывели тем или иным способом. Это возможно только в том случае, когда само *состояние, в котором находится объект, является воздействующим на него фактором.*

Если представлять себе управляемый объект как систему со входами и выходами (рис. 5.2), то в этом случае некоторые его выходы одновременно являются его входами. Именно так обстоит дело в модели ограниченного или неограниченного роста: выходной параметр $M(n + 1)$ характеризует очередное состояние системы, т. е. одновременно является входом. Вход и выход различают только моменты времени, в которых рассматриваются значения этого параметра.

Следовательно, изображая управляемый объект с элементами саморегуляции, мы должны такие выходы замкнуть на соответствующие входы (рис. 5.5).

В модели управления добычей возобновляемых ресурсов эта схема выглядит так, как показано на рисунке 5.6.

Воздействие выходных параметров динамической системы на ее же входные параметры называют **обратной связью**.

В живой природе любая система обязательно обладает механизмом обратной связи. Простейший пример — увеличение теплоотдачи живым организмом при увеличении выделения энергии (например, при интенсивной работе). Этот механизм настолько эффективен, что при здоровом состоянии организма он способен поддерживать постоянную температуру тела при воздействии самых разнообразных внешних и внутренних факторов. Здесь мы имеем дело с так называемой **отрицательной обратной связью**. Эпитет «отрицатель-

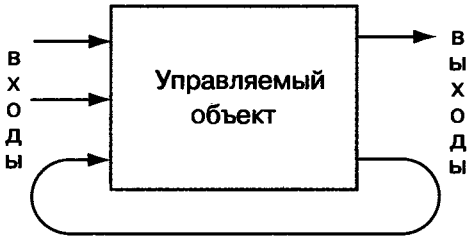


Рис. 5.5 Схема управления с обратной связью



Рис. 5.6 Модель потребления возобновляемого ресурса как системы с обратной связью

ная» означает, что она обеспечивает выработку управляющего воздействия, направленного на уменьшение рассогласования между заданным и действительным значениями параметра системы. Возникающий при этом эффект устойчивости системы к внешним воздействиям называют **гомеостазом** системы.

Нередко в динамических системах встречается и **положительная обратная связь**, усиливающая возникшее рассогласование. В природе примером положительной обратной связи является реакция пищеварительной системы на слабое раздражение пищей, вызывающее выделение пищеварительных энзимов и тем самым усиление раздражения с еще большим выделением энзимов и т. д. Это явление описывается народной приметой «Аппетит приходит во время еды». В школьных курсах физики и химии изучается другой тип явлений с положительной обратной связью — цепные реакции. К ним относится атомный взрыв, взрыв гремучего газа и т. д.

Понятие обратной связи возникло в ходе построения различных автоматических регуляторов. Одним из первых таких регуляторов был регулятор Уатта (рис. 5.7), управлявший подачей пара в цилиндры паровой машины (о нем вам, скорее всего, рассказывали на уроках физики). Позже понятие обратной связи было выделено как самостоятельное, описывающее саморегулирующиеся системы.

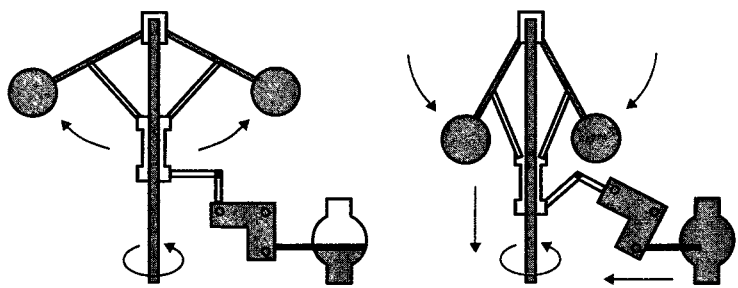


Рис. 5.7 Схема работы регулятора Уатта

Объект с саморегуляцией может быть достаточно четко структурирован; в частности, в нем могут быть выделены управляющий и управляемый подобъекты. Эти подобъекты связаны информационными каналами: по одним каналам передается управляющая информация от управляющего объекта к управляемому, по другим — информация о результатах управления. Для паровой машины с регулятором Уатта управляющий объект — сам регулятор, управляемый — паровой котел.

Главным в понятии обратной связи является то, что *изменение состояния объекта непосредственно связано с этим состоянием.*

В паровой машине, как только скорость маховика начинает превышать установленные пределы, объем подачи пара уменьшается пропорционально этой скорости и, наоборот, когда скорость падает ниже установленных пределов, увеличивается объем подачи пара. Отметим, что в этой системе регулятор не «выясняет» причин увеличения или уменьшения скорости (разнообразие этих причин не поддается описанию), он просто поддерживает заданный режим работы паровой машины.

ВОПРОСЫ И ЗАДАНИЯ

1. При каких условиях возможно явление саморегулирования? Приведите примеры саморегулирующихся динамических систем.
2. Что такое обратная связь?
3. Какую обратную связь называют отрицательной, положительной? Приведите примеры систем с отрицательной и положительной обратной связью.
4. Что такое гомеостаз системы?
5. Среди приведенных ниже примеров взаимодействия укажите те, которые относятся к понятию обратной связи. Для каждого приведенного вами случая укажите тип обратной связи, т. е. положительна она или отрицательна:
   а) изменение диаметра кровеносных сосудов организма и теплоотдачи организма;
   б) раскрытие (или закрытие) зонтика и начало (или прекращение) дождя;
   в) изменение доходности производства и объема выпускаемой продукции;
   г) изменение покупательского спроса и цены на товар;
   д) изменение яркости горения лампочки и подаваемого напряжения;
   е) изменение скорости бега и сопротивления воздуха;
   ж) изменение скорости бега и частоты пульса;
   з) восход (заход) солнца и выключение (включение) осветительных приборов;
   и) звонок будильника и пробуждение человека.
6. Выполняя тестовые задания к главам нашего учебника, вы получали информацию о том, насколько хорошо усвоен изучаемый материал. Мы надеемся, что вы использовали эту информацию для построения управления своим процессом обучения. Какой тип обратной связи при этом реализуется?

§47 Изучаем системы с обратной связью

Из предыдущего параграфа вы уже, наверно, поняли, что системами с обратной связью мы занимались еще в главе 3, изучая модели ограниченного и неограниченного роста. Но вы об этом тогда не догадывались. Да это было и не нужно — в то время вы осваивали понятия границ адекватности. Теперь же мы сосредоточим внимание именно на обратной связи. Начнем с обсуждения такой задачи.

На некотором острове живут лисы и кролики. Кролики питаются травой, которой вдоволь на острове, а лисы охотятся на кроликов. Экологи время от времени пересчитывают кроликов и лис. Вот что они установили:

- коэффициент прироста числа кроликов зависит от условий обитания (холодная или теплая зима, сухое или влажное лето, ранняя или поздняя весна и т. д.) и колеблется в пределах от 3,2 до 4,7;
- коэффициент прироста числа лис пропорционален коэффициенту прироста числа кроликов.

Требуется установить, как меняется численность кроликов и лис с течением времени.

Решение задачи, как обычно, начинаем с построения модели. Нам даны две популяции: кролики и лисы. Схематично изобразим их так, как показано на рисунке 5.8. Здесь $M(n)$ — масса кроликов через n лет, а $L(n)$ — масса лис в тот же момент времени.

Но эти популяции не существуют раздельно, каждая сама по себе, а взаимодействуют друг с другом. Это взаимодействие можно изобразить схемой, приведенной на рисунке 5.9. В этой системе легко просматриваются три контура обратных связей. Опишем их расчетными формулами.

Сначала займемся кроликами. Поскольку корма у них вдоволь, то их прирост идет в соответствии с моделью неограниченного роста. Вот только лисы их поедают. Конечно, аппетит у раз-

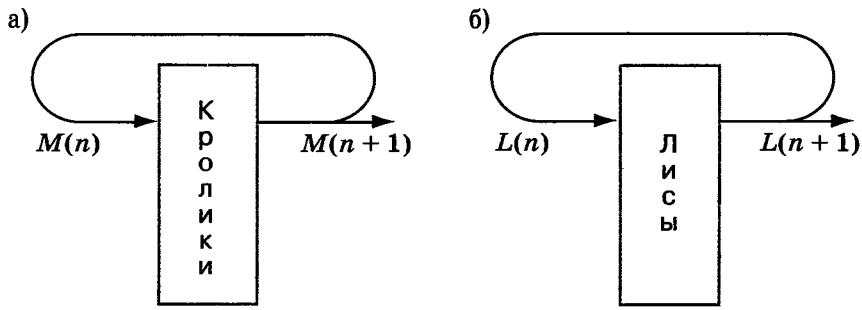


Рис. 5.8 Модели простейших экологических систем

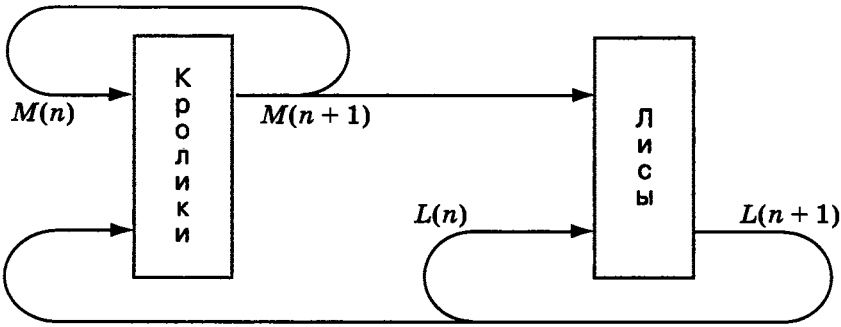


Рис. 5.9 Структура модели экологической системы «лисы и кролики»

ных лис может оказаться разным, но в среднем можно считать, что съедаемая крольчатина по лисам распределяется равномерно, т. е. масса кроликов, съеденных одной лисой, пропорциональна общему числу кроликов. Эту массу можно записать как $cM(n)$, где c — коэффициент, характеризующий, можно сказать, «неувертливость» кроликов или, наоборот, умелость лис в охоте. Значит, изменение массы кроликов за год можно записать так:

$$M(n + 1) - M(n) = kM(n) - cM(n)L(n).$$

Здесь k — коэффициент прироста кроликов за год.

То же соотношение можно переписать так:

$$M(n + 1) = (1 + k - cL(n))M(n).$$

А как обстоят дела у лисиц? Ясно, что прирост численности лис пропорционален их имеющемуся количеству $L(n)$ и приросту крольчатины $M(n + 1) - M(n)$. Если прирост нулевой, то новым лисам нечего будет есть; если же прирост отрицательный (т. е. число кроликов уменьшается), то должно уменьшаться и число лис. Обозначим коэффициент пропорциональности через s. Значит, для лис можно написать такую формулу:

$$L(n + 1) - L(n) = s(M(n + 1) - M(n))L(n).$$

Перепишем эту формулу в виде, удобном для вычисления $L(n + 1)$:

$$L(n + 1) = (1 + s(M(n + 1) - M(n)))L(n).$$

Выбрав теперь некоторый год в качестве начального (т. е. считая $n = 0$), мы, зная массу кроликов $M(0)$ и массу лис $L(0)$, можем последовательно вычислить $M(1)$ и $L(1)$, $M(2)$ и $L(2)$ и т. д. Коэффициенты k, c и s известны из наблюдений экологов.

Модель построена; можно приступать к компьютерному эксперименту с этой моделью. Вы его проведете, выполняя лабораторную работу 24.

ВОПРОСЫ И ЗАДАНИЯ

1. Как сказано в объяснительном тексте, в рассмотренной модели «лисы и кролики» имеется три контура обратных связей. Укажите их. Какие из этих связей являются положительными, а какие — отрицательными?

2. а) Предположим, что выполняется равенство $L(0) = \frac{k}{c}$. Какие значения будут иметь $M(1)$ и $L(1)$; $M(2)$ и $L(2)$, ..., $M(n)$ и $L(n)$?

   б) Пусть имеет место равенство, записанное в пункте *а*. Говорят в этом случае, что значения $M = M(n)$ и $L = L(n)$ характеризуют положение *экологического равновесия* в данной биологической системе. Как вы думаете, почему используется такое название?

3. Какую роль играют положительные обратные связи в эволюции биологических систем?

4. Подставим в формулу, полученную для $L(n + 1)$, выписанное ранее соотношение для $M(n + 1) - M(n)$. Тогда

   $$L(n + 1) = (1 + s(k - cL(n))M(n))L(n).$$

   Какая модель для популяции лис описывается этой формулой, если предположить, что популяция кроликов имеет постоянную массу $M(n)$?

§48 Управление по принципу обратной связи

Сравним функционирование двух систем — экологической системы «лисы и кролики» и паровой машины с регулятором Уатта (табл. 5.3). Сходство налицо, не правда ли? Можно сказать, что популяция лис выступает регулятором численности кроликов.

Таблица 5.3

	Изменение в системе	**Реакция системы на изменение**
Лисы и кролики	Увеличение (или уменьшение) численности кроликов	Увеличение (соответственно уменьшение) численности лис
Паровая машина с регулятором Уатта	Увеличение (или уменьшение) скорости	Увеличение (соответственно уменьшение) сброса пара в котле

Конечно, в природной системе ни о какой целенаправленности воздействия речь идти не может. Но природа подсказывает нам, что и при создании управляемой системы ее устойчивость обеспечивается наличием обратной связи между управляемым и управляющим (регулирующим) объектами системы. Управление, использующее обратную связь между управляемым и управляющим объектами, называется **управлением по принципу обратной связи**. Мы можем продемонстрировать это схемой, изображенной на рисунке 5.10.



Рис. 5.10 Управление по принципу обратной связи

Сравним управление по принципу обратной связи с работой светофора, который управляет движением на перекрестке. Здесь отсутствует всякая обратная связь, ибо каждый сигнал горит заранее заданное время вне зависимости от того, есть ли машины, движущиеся в данном направлении. Вот и получается нередко, что на зеленый свет ехать некому, хотя в то же время на красный стоит вереница машин. К сожалению, в нашей социальной жизни часто имеет место управление именно такого типа. Не случайно большинству людей управление представляется как грубое насилие, выражающееся в указах, приказах, законах, постановлениях. Подобное управление стремится предусмотреть и регламентировать все возможные случаи поведения управляемого объекта, и если вдруг обнаруживается лазейка для обхода того или иного постановления, то проводится расследование причин и принимаются меры по дальнейшему недопущению. Но лазейки-то всегда есть и будут... Напомним, что главное преимущество управления по принципу обратной связи состоит в том, что регулятору не требуется выяснять причины и обстоятельства, которые привели к изменению состояния управляемого объекта, а он просто реагирует на это изменение.

В экономической сфере принцип обратной связи проявляется в действии рыночных законов. Повышение спроса приводит к повышению цены, повышение цены делает выгодным увеличение производства данного товара, увеличение производства приводит к снижению спроса, снижение спроса влечет падение цены и т. д. На рисунке 5.11 изображена схема экономического взаимодействия.

Конечно, эта модель весьма упрощена. Здесь, к примеру, не отражено действие такого фактора, как покупательная способность населения и многое другое. Но мы обращаем ваше внимание именно на наличие обратной связи уже в данной, совсем простой модели экономического управления.

Проведенный вами компьютерный эксперимент с моделью «лисы и кролики» демонстрирует еще один важный момент в функционировании динамической системы. Вспомните, если числен-

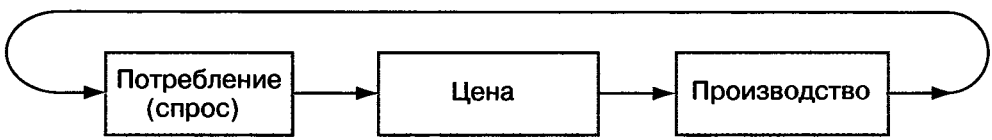


Рис. 5.11 Простейшая экономическая модель

ность лисиц намного отличалась от той, которая необходима для экологического равновесия, система не могла прийти к этому равновесию и разрушалась.

Для каждой системы существуют определенные границы устойчивости этой системы, в которых она сохраняет свою целостность; выход за эти границы приводит к разрушению системы.

Именно выходом за границы устойчивости объясняются экологические катастрофы и экономические кризисы. И чтобы система была восстановлена, необходимо вернуть ее в область саморегулирования. А для этого важно знать, каковы же границы этой области. Находить их как раз и помогает компьютерное моделирование задач управления.

ВОПРОСЫ И ЗАДАНИЯ

1. Что такое управление по принципу обратной связи?
2. В чем причины разрушения систем, способных к саморегулированию?
3. Приведите примеры управления по принципу обратной связи.
4. На рисунке 5.11 приведена простейшая схема экономического взаимодействия. Введем еще один элемент экономической системы (рис. 5.12).
Как, по вашему мнению, его следует встроить в указанную схему? Перечислите возникающие при этом контуры обратных связей и для каждой из обратных связей определите, положительна она или отрицательна.



Рис. 5.12 Дополнительный элемент экономической системы

5. В § 44 рассматривалась задача об организации посещения парка. Предложенное для этой задачи решение не предусматривает обратной связи в управлении посещениями парка. Предложите управление, регулирующее посещение парка и основанное на принципе обратной связи. (С о в е т. При построении управления используйте экономические факторы.)
6. а) Известна история о том, что правитель одного из восточных государств платил своему врачу только в то время, когда он не болел. Рассмотрите эту ситуацию как управление в системе «врач — пациент». Укажите используемый здесь тип обратной связи — положительна она или отрицательна.
б) Какой тип обратной связи используется в современном, наиболее распространенном виде платного медицинского обслуживания?
в) Какой тип управления реализуется при бесплатном медицинском обслуживании?
7. Практически все государства мира законодательно борются с мздоимством (дачей и получением взяток) своих чиновников. Эта борьба иногда бывает более успешной, зачастую менее, но не может искоренить зла, поскольку опирается только на страх перед разоблачением и последующим наказанием. Предложите какой-либо механизм борьбы со взяточничеством, основанный на принципе обратной связи.

§49 Глобальные модели

В предыдущих параграфах мы рассмотрели весьма простые модели управления. Относятся они к малым территориям или весьма малым совокупностям объектов. Это так называемые локальные модели. В реальном управлении используются намного более сложные модели, учитывающие многочисленные и разнородные факторы. Но принципы моделирования и управления остаются общими как для больших моделей, так и для маленьких. А главное, что роль маленьких моделей совсем не маленькая. Ведь мы можем соединять их между собой, словно элементы конструктора, и получать сложные модели, описывающие большие пространства и многофакторное воздействие. Такие модели, описывающие многофакторные процессы, протекающие на территориях больших регионов, стран или даже всего земного шара, получили название *глобальных моделей*. В 70-е годы в Вычислительном центре Академии наук СССР под руководством академика Н. Н. Моисеева была начата разработка первой глобальной модели биосферы Земли, способной показать, что будет происходить с Землей в результате тех или иных воздействий на нее человечества. Именно на этой модели были предсказаны последствия ядерной войны, в которой не может быть победителя, ибо последствия ее будут необратимы.

Другие глобальные модели, построенные в 80-е годы XX века, показали, что безудержный рост потребления человечеством даже возобновляемых ресурсов тоже приведет к краху. Нужна продуманная, экологически выверенная стратегия управления на всех уровнях — от района до государства и сообщества государств, чтобы человечество осталось на Земле. Такие стратегии есть, и находить их помогают информационные компьютерные технологии.

ВОПРОСЫ

1. Что такое глобальные модели?
2. В чем вы видите роль информатики и информационных технологий в жизни современного общества?

ИТОГИ ГЛАВЫ 5

Повторим кратко то основное, что вы узнали, изучив главу 5.

Наука, изучающая информационные процессы в управлении живыми организмами, социальными и техническими системами, называется кибернетикой. Управлением называется воздействие на объект или процесс, имеющее своей целью получение требуемых значений параметров этого объекта или процесса.

Одним из важнейших вариантов реализации управления, особенно часто встречающимся в технических системах, является управление на основе заранее подготовленной инструкции, определяющей программу действий объекта. Однако в живой природе преобладает другой принцип управления — управление на основе обратной связи.

Обратной связью называют воздействие выходных параметров управляемого объекта (процесса) на его же входные параметры. При наличии обратной связи объект (процесс) обладает способностью к саморегуляции. Осуществление обратной связи возможно лишь в том случае, когда состояние, в котором находится рассматриваемый объект (процесс), является фактором, воздействующим на него самого.

Обратная связь называется отрицательной, если она обеспечивает выработку управляющего воздействия, направленного на уменьшение рассогласования между заданным и действительным значениями параметра системы. Возникающий при этом эффект устойчивости системы к внешним воздействиям называют гомеостазом системы. Положительной называется обратная связь, обеспечивающая выработку управляющего воздействия, увеличивающего возникшее рассогласование. В живых природных системах все процессы подчинены принципу обратной связи.

Для каждой системы существуют определенные границы устойчивости ее функционирования, в которых она сохраняет свою целостность; выход за эти границы приводит к разрушению системы.

ПРОВЕРЬ СЕБЯ

Часть 1. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и номер правильного ответа. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

А1. Воздействие выходных параметров объекта на его входные параметры называется:

1) гомеостазом; 2) обратной связью; 3) эволюцией;

4) среди вариантов, указанных в пунктах 1—3, нет правильного.

А2. Основоположником кибернетики является:

1) Дж. фон Нейман;

2) Н. Винер;

3) К. Шеннон;

4) А. Н. Колмогоров.

А3. Для того чтобы обеспечить устойчивость системы при управлении по принципу обратной связи, обратная связь должна быть:

1) стабильной;
2) положительной;
3) отрицательной;
4) любой.

А4. Система может быть управляемой только в том случае, если она:
1) материальная;
2) информационная;
3) динамическая;
4) находится в состоянии гомеостаза.

Часть 2. При выполнении предложенных ниже заданий запишите у себя в тетради шифр задания и ответ в виде последовательности символов. По окончании работы сверьте получившуюся у вас запись с ключами на с. 269.

В1. Обратная связь, при которой возникшее в системе рассогласование увеличивается, называется _____.

В2. Целенаправленное воздействие на объект или процесс, совершаемое с целью получения требуемых значений параметров этого объекта или процесса, называется _____.

В3. Ниже приведены примеры некоторых регулирующих устройств:
1) регулятор подачи воды в бак: поплавок, соединенный с заглушкой, перекрывающей доступ воды;
2) светофор;
3) демпферный регулятор двери: чем меньше открыта дверь, тем медленнее она закрывается;
4) терморегулятор утюга;
5) электрический предохранитель;
6) солнцезащитные стекла: чем интенсивнее свет, тем они темнее.

Выберите из этого списка регуляторы, действующие по принципу обратной связи, и запишите их номера в порядке возрастания без пробелов и знаков препинания.



Компьютерный практикум

Выполнять задания лабораторных работ, помещенных в этом разделе учебника, вы будете в компьютерном классе. Поэтому напомним правила техники безопасности — ведь к каждому рабочему месту подведено опасное для жизни напряжение электричества.

1. Если вы обнаружили какую-либо неисправность, немедленно сообщите об этом преподавателю. Не работайте на неисправном оборудовании!
2. Не включайте и не выключайте компьютеры без разрешения учителя.
3. Не трогайте, а тем более не дергайте различные провода.
4. Не стучите по клавиатуре и мышке.
5. Не работайте на компьютере с грязными руками.

Для вашего здоровья важно и то, как вы сидите перед компьютером:

Позвоночник должен быть вертикально прямым; спина опирается на спинку стула.
Ступни удобно стоят на полу или специальной подставке.
Не наклоняйтесь к экрану. Расстояние от глаз до экрана должно быть не менее вытянутой руки.
Верхний край экрана располагается примерно на уровне ваших глаз.
Если вы посмотрите на центр экрана, то линия вашего взгляда должна быть перпендикулярной плоскости экрана, т. е. экран должен быть развернут немного вверх.
Плечи отведены назад и опущены, и тогда вам не составит труда удобно расположить руки над основными клавишами среднего ряда.

Хотя практикум называется компьютерным, это вовсе не означает, что, приступая к выполнению заданий лабораторной работы, нужно тут же включать компьютер. Мы советуем сначала внимательно прочитать задание и поразмышлять, как именно будет использоваться та или иная компьютерная технология для его выполнения.

Лабораторная работа 1 (к § 5)

Обработка числовой информации с помощью электронной таблицы

Первые компьютеры были призваны облегчить человеку обработку числовой информации. Правда, они делали это далеко не в самой удобной для человека форме. Настоящий переворот в технологии обработки числовой информации произошел, когда в 1979 году Д. Бриклин и Р. Фрэнкстон создали первую электронную таблицу. Мы тоже предлагаем вам начать компьютерный практикум с обработки числовой информации с помощью электронной таблицы. В описании работы мы ориентируемся на электронную таблицу Microsoft Excel, которой, вероятнее всего, будет пользоваться большинство из вас.

Приступая к выполнению лабораторной работы, вы должны вспомнить, как заполняется электронная таблица и как с ее помощью производятся вычисления. Основные правила мы приводим ниже; если вам понадобится дополнительная информация, вы легко ее получите, воспользовавшись встроенной в Excel справкой.

При использовании электронной таблицы часть исходной числовой информации заносится прямо в ее клетки (**ячейки** электронной таблицы). Эту информацию называют **исходные данные**. Кроме того, в другие ячейки необходимо записать формулы, по которым рассчитываются те данные, которые являются интересующими вас результатами.

Каждая страница электронной таблицы разбита на *столбцы*, обозначенные буквами латинского алфавита, и на *строки*, пронумерованные целыми числами, но само это разбиение при выводе на принтер не печатается.

Благодаря разбиению на пронумерованные строки и поименованные столбцы, каждая ячейка электронной таблицы имеет свой собственный **адрес**.

Для каждой таблицы существуют специальные правила ее заполнения, указанные в инструкции пользователю. Для электронной таблицы Microsoft Excel эти правила таковы:

- Если среди символов, вводимых в ячейку, есть буквы или другие символы, которых не может быть в числе, то это текст.

Например, текстом являются следующие последовательности символов: Василий, или 10.234.245, или a1234.

- Если вводится правильное число, то это число.

Например, 234.

- Чтобы электронная таблица распознавала, что вводится текст или формула, используется заранее обусловленный знак. Формула всегда начинается со знака «=».

Электронная таблица имеет довольно значительный набор стандартных функций, облегчающих жизнь пользователю. Пусть, к примеру, надо в ячейке **C6** получить сумму чисел, записанных в ячейках **B6**, **B7**, **B8**, **B9**, **B10**, **B11**, **B12**. Для этого в **C6** можно записать формулу

=B6+B7+B8+B9+B10+B11+B12

А если надо сложить не семь, а двадцать семь или сто семь чисел? Чтобы не писать длинное выражение, можно воспользоваться стандартной функцией, которая называется ***суммирование*** содержимого ***блока ячеек***:

СУММ(B6:B12)

Напомним, что ***блоком ячеек*** в электронной таблице называется совокупность всех ячеек, заполняющих некоторый прямоугольник. Для того чтобы электронная таблица знала, с каким блоком ей иметь дело, указывают через двоеточие ***адреса*** ячеек, стоящих в левом верхнем и правом нижнем углах прямоугольника. У электронной таблицы есть много разных операций над блоками ячеек. Это поиск максимального или минимального элемента, расчет среднего значения и т. д. Блок ячеек можно скопировать или перенести в другое место таблицы. При этом надо учитывать, что в Excel основной является **относительная адресация** ячеек. Это означает, что при копировании формулы из одной ячейки в другую адреса ячеек, входящих в формулу, изменяются на те величины, на которые отличается адрес новой ячейки. Например, если указанную выше формулу из ячейки C6 скопировать в ячейку E3, то в ячейку E3 будет записана формула **СУММ(D3:D9)** — ведь буква E получается из буквы C сдвигом на две позиции вправо, поэтому B в формуле заменяется на D; а число 6 от числа 3 отличается на −3. Если необходимо, чтобы какой-то элемент адреса не менялся при копировании (так называемая **абсолютная адресация**), то перед ним ставится знак $. Например, если бы в C6 была написана формула **СУММ($B6:B$12)**, то после копирования этой формулы в E6 там была бы записана формула **СУММ($B3:D$12)**.

А теперь попробуем приспособить электронную таблицу для решения задачи, которая может возникнуть в практической деятельности.

1 Владелец автостоянки хочет автоматизировать расчеты с клиентами за услуги. Разработайте заполнение электронной таблицы, которое позволило бы это сделать.

Договоримся информацию об использовании места на стоянке записывать в строку. Чтобы таблица не была очень большой по числу строк, договоримся, что на стоянке 10 мест. Простейший вариант заполнения таблицы может выглядеть так, как показано в таблице КП.1.

Для времени приезда и отъезда выбран формат, в котором до запятой указываются часы, а после запятой — минуты. Это позволяет работать с этими данными как с числами.

Таблица КП.1

	A	B	C	D	E	F
1	№ места	Время приезда	Время отъезда	Продолжительность стоянки, мин	Цена за час	Стоимость
2	1	10,43	12,58	40*(Целое(B2)–Целое(C2))+(C2–B2)*100	10	D2*$E$2/60
3	2	8,15	11,33	40*(Целое(B3)–Целое(C3))+(C3–B3)*100		D3*$E$2/60
4	1	13,37	16,05	40*(Целое(B4)–Целое(C4))+(C4–B4)*100		D4*$E$2/60
	…	…	…	…	…	…
21	7			40*(Целое(B21)–Целое(C21))+(C21–B21)*100		D21*$E$2/60

Таблица КП.2

№ места	1	2	1	4	3	5	2	7	2	5	8	3	10	1	6	5	9	10
Время приезда	10,43	8,15	13,37	6,10	7,25	13,40	12,25	8,10	15,15	16,50	8,50	17,23	13,24	20,12	7,45	20,05	9,10	21,17
Время отъезда	12,58	11,33	16,05	18,23	14,10	16,13	13,47	17,25	17,05	18,30	23,11	18,18	20,20	22,57	16,33	23,12	20,34	21,43

▷ Требует обсуждения и формула расчета продолжительности стоянки. Перевод времени приезда в минуты выглядит так:

60*Целое(B2)+(B2–Целое(B2))*100,

где Целое(B2) — это целая часть числа, записанного в ячейку B2. Теперь уже легко понять, как получаются формулы, записанные в столбце D. Формулу можно написать только один раз, а затем скопировать ее в другие ячейки столбца D. Аналогично заполняется столбец F (не забудьте отметить некоторые элементы адресов как абсолютные). ◁

2 Заполните столбцы B и C в соответствии с таблицей КП.2.

3 А теперь в ячейке E3 подсчитайте дневной доход автостоянки. Введите в ячейку G7 формулу:

=ЕСЛИ(И(B2>11;A2<12);1;0)+ЕСЛИ(И(B3>11;A3<12);1;0)+
+ЕСЛИ(И(B4>11;A4<12);1;0)+ЕСЛИ(И(B5>11;A5<12);1;0)+
+ЕСЛИ(И(B6>11;A6<12);1;0)+ЕСЛИ(И(B7>11;A7<12);1;0)+
+ЕСЛИ(И(B8>11;A8<12);1;0)+ЕСЛИ(И(B9>11;A9<12);1;0)+
+ЕСЛИ(И(B10>11;A10<12);1;0)+ЕСЛИ(И(B11>11;A11<12);1;0)+
+ЕСЛИ(И(B12>11;A12<12);1;0)+ЕСЛИ(И(B13>11;A13<12);1;0)+
+ЕСЛИ(И(B14>11;A14<12);1;0)+ЕСЛИ(И(B15>11;A15<12);1;0)+
+ЕСЛИ(И(B16>11;A16<12);1;0)+ЕСЛИ(И(B17>11;A17<12);1;0)+
+ЕСЛИ(И(B18>11;A18<12);1;0)+ЕСЛИ(И(B19>11;A19<12);1;0)+
+ЕСЛИ(И(B20>11;A20<12);1;0)+ЕСЛИ(И(B21>11;A21<12);1;0)

▷ Напомним, что оператор И(x; y) дает значение Истина, если истинны оба аргумента x и y; оператор ЕСЛИ(x; y; z) дает значение y, если аргумент x истинен, и дает значение z в противном случае. Легко понять, что указанный оператор подсчитывает количество мест, которые были заняты между 11 и 12 ч. ◁

4 А теперь в ячейки G2—G6 запишите выражения, согласно которым будет подсчитано количество занятых стоянок между 6 и 7 ч, между 7 и 8 ч, ..., между 10 и 11 ч соответственно; в ячейки G8—G19 запишите аналогичные данные для промежутка от 12 до 24 ч. Подумайте, как можно облегчить ввод таких длинных формул. Теперь можно проанализировать, когда загрузка автостоянки была больше.

▷ Нетрудно подсчитать, что при круглосуточном полном заполнении автостоянки дневная выручка составила бы 2400 р. Сравнивая это число с полученным результатом в ячейке E3, можно увидеть, что возможности используются далеко не полностью. ◁

5 Повысить эффективность можно, если обеспечить круглосуточную работу автостоянки. В этом случае машину можно ставить в один день, а забирать в другой. Для организации учета в такой ситуации нужно ввести в таблицу еще два столбца: дату

приезда и дату отъезда — и соответствующим образом модифицировать расчет продолжительности стоянки.

6 Предположим, что владелец решил ввести дневной и ночной тарифы оплаты: с 7 до 22 ч действует дневной тариф, а с 22 до 7 ч — ночной. Модифицируйте таблицу для этой ситуации.

7 Для привлечения клиентов владелец объявил скидки в случае, если место занято более суток, а также ввел дополнительную услугу: абонирование места. Хозяин машины платит абонентскую плату за место, закрепленное за его машиной, когда автомобиля там нет (более низкую, чем за стоянку самого автомобиля). Сделайте необходимые модификации заполнения электронной таблицы.

Лабораторная работа 2 (к § 6)

Обработка текстовой и графической информации

В курсе информатики, изучавшемся вами в 8—9 классах, вы знакомились с текстовым и графическим редакторами. Выполняя эту работу, вы вспомните имеющиеся знания.

1 Представьте, что вы сотрудник небольшой полиграфической фирмы и вам поступил заказ на изготовление визитной карточки. Визитная карточка — это информационная модель ее владельца. Поэтому нужно сразу определить с заказчиком, какую существенную информацию он хочет на ней разместить. Вот один из возможных вариантов:
- фамилия, имя и отчество;
- название фирмы, в которой заказчик работает;
- занимаемая должность;
- эмблема фирмы;
- почтовый адрес;
- телефон, факс, e-mail;
- адрес сайта.

2 Теперь надо определить, как эту информацию расположить на карточке, размеры которой стандартны: 90×50 мм. Можно, например, расположить так, как показано на рисунке КП.1.

Придумайте, как будет называться фирма, как зовут клиента, его адрес и т. д. Вам необходимо также принять решение, каким шрифтом будет представлен каждый информационный элемент визитной карточки.

▶ Напомним, что каждый шрифт характеризуется **кеглем** (т. е. высотой букв), **гарнитурой** (графическим образом букв) и **начертанием** (прямой, *наклонный*, **жирный**). Отметим, что шрифты делятся на рубленые и с засечками. Вот примеры:

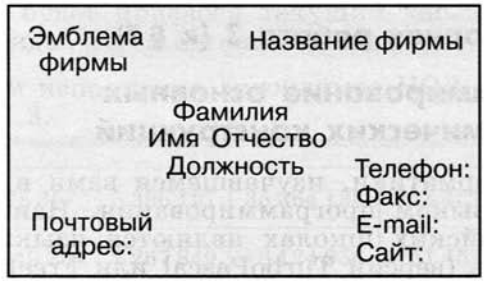


Рис. КП.1 Макет визитной карточки

Этот текст набран рубленым шрифтом гарнитуры Futuris.
А этот текст набран шрифтом с засечками гарнитуры Times.

Шрифты с засечками визуально как бы объединяют слово в одно целое, и это увеличивает скорость чтения на 10—15%. Рубленые шрифты обычно применяют для заголовков. Создавая карточку, поэкспериментируйте со шрифтами и выберите для каждого текстового элемента наиболее, по вашему мнению, удачный. Не забудьте, что шрифт может быть цветным, однако карточка не должна быть слишком пестрой. ◀

3 А теперь запустите текстовый редактор. Для того чтобы информационные элементы располагались на карточке так, как вы задумали, ее удобно создавать в виде таблицы, у которой выделена только внешняя рамка, а внутренние стороны клеток не прорисовываются. Конечно, размер клетки для каждого элемента будет свой, так что вам предстоит поэкспериментировать с перемещением границ клеток.

4 Теперь займемся эмблемой. Ее тоже можно создавать средствами графики, встроенной в текстовый редактор. Но эффективнее воспользоваться графическим редактором. Запустите имеющийся в вашем распоряжении графический редактор и создайте подходящую эмблему. Цвета, применяемые вами в эмблеме, должны хорошо сочетаться с цветами шрифтов. Возможно, после создания эмблемы вы захотите поменять шрифтовые решения. Импортируйте эмблему в визитную карточку. Оцените созданный вами продукт в целом. Если что-то не понравилось, еще не поздно изменить.

▶ Визитная карточка может отражать вовсе не профессиональный облик заказчика, а его увлечения или интересы. Возможно, он является членом какого-либо спортивного клуба (например, туристического) или клуба по интересам (например, филателистского). Разработайте соответствующую информационную модель и создайте визитную карточку для такого представления вашего заказчика. ◀

Лабораторная работа 3 (к § 7)

Программирование основных алгоритмических конструкций

В курсе информатики, изучавшемся вами в 8—9 классах, вы знакомились с языком программирования. Наиболее распространенными в российских школах являются языки Бейсик (версия QBasic) и Паскаль (версии TurboPascal или FreePascal). В таблице на форзацах показано, как на этих языках выглядят основные алгоритмические конструкции. Если же вы изучали другой язык, то описание этих конструкций вам напомнит учитель или вы найдете его в соответствующем справочнике.

А теперь выполните несколько заданий.

1 Запрограммируйте алгоритм Евклида — он приведен в объяснительном тексте § 7. Протестируйте вашу программу на небольших числах, для которых легко найти НОД и без компьютера. Убедившись, что программа работает правильно, найдите НОД(178 357, 1 049 254).

2 Модифицируем алгоритм следующим образом:

```
Алгоритм НОДм
цел: m, n, k;
{ Запросить m;
  Запросить n;
  k := m + n;
  Делать пока (не (n = 0))
  { m := n;
    n := k mod n;
    k := m;
  }
  Сообщить m;
}
```

Запрограммируйте и отладьте модифицированный алгоритм.

▷ Давайте сравним эффективность этих двух алгоритмов. Для этого в программы надо вставить операторы, показывающие время.

В языке Бейсик можно воспользоваться функцией TIMER. По команде t = TIMER переменной t будет присвоено значение, равное числу секунд, которые прошли от 0 часов текущих суток.

В языке Паскаль для вывода времени нужно записать следующую конструкцию:

```
Uses Dos;
Var h, m, s, ss: word;
begin
  gettime(h, m, s, ss);
end.
```

Переменной h будет присвоен текущий час, переменной m — минуты, s — секунды, ss — миллисекунды. ◁

3 По результатам исполнения алгоритмов НОД и НОДм заполните таблицу КП.3.

Таблица КП.3

1-е число	178 357	5 386 127	35 364 128	213 457 281	3 245 798 457
2-е число	1 049 254	7 267 849	76 124 956	751 457 358	7 254 124 751
Время работы алгоритма НОД					
Время работы алгоритма НОДм					

Для каждой пары чисел найдите отношение времени работы алгоритма НОД к времени работы алгоритма НОДм. Как, на ваш взгляд, можно охарактеризовать это отношение?

4 Запрограммируйте алгоритм, представленный схемой алгоритма в задании 18 из § 7. Запустите отлаженную программу для некоторых начальных данных. Подтвердилась ли ваша гипотеза о том, для чего предназначен этот алгоритм?

5 Запрограммируйте алгоритм, приведенный в задании 23 из § 7. Напомним, что операция LEN(W) в языке Бейсик записывается LEN(W$), а в языке Паскаль — length(W), операция Часть (S, a, b) в языке Бейсик записывается MID$(S$, a, b), а в языке Паскаль — COPY(S, a, b).



Лабораторная работа 4 (к § 12)

Фактографическая модель «Класс»

Выполнив задания 9 и 10 из § 12, вы определили параметры информационной модели «Класс». Перечислим параметры, которые фигурируют в этих заданиях:

- фамилия;
- имя;
- дата рождения;
- адрес;
- телефон;
- информация о братьях и сестрах: год рождения и имя;
- спортивные результаты;
- увлечения.

Теперь на основе этой модели создадим базу данных. Предварительно соберите нужную информацию об учениках своего класса в соответствии с параметрами информационной модели.

Мы опишем создание базы данных в СУБД Access. СУБД Access является одним из приложений ОС Windows и обычно входит в набор офисных технологий. Главными объектами СУБД Access являются таблицы с информацией, но, кроме них, данная СУБД способна создавать производные объекты — формы, запросы, отчеты, которые облегчают пользователю его работу и восприятие информации. Но сначала поговорим о создании таблиц. СУБД Access позволяет создавать не одну, а несколько связанных между собой таблиц. Но нельзя осваивать все сразу, и мы пока остановимся на однотабличном варианте.

После запуска СУБД открывается окно базы данных. Оно позволяет увидеть, какие объекты (кроме скрытых) имеются в базе, и выполнить над ними такие операции, как копирование, вставка, переименование, удаление. Основные операции, выполняемые в этом окне, представлены в контекстном меню, открываемом нажатием правой клавиши мыши. Это операции, выполняемые над объектом в целом и не требующие его открытия. Отметим, что если был выделен не объект, а его имя, то нажатие правой клавиши открывает другое контекстное меню, где собраны команды, относящиеся к имени объекта.

В меню *Файл* вы можете выбрать либо режим *Создать*, либо режим *Открыть*. В первом случае Access попросит вас ввести имя и место сохранения создаваемого файла. Имени файла будет автоматически присвоено расширение mdb.

А теперь приступим к созданию базы данных.

1 Запустите СУБД Access. В меню *Файл* выберите режим *Создать*. В диалоговом окне *Создание таблицы* щелкните на кнопке *Новая таблица*. Перед вами появится макет таблицы в режиме конструктора. Заполните в первой строке имя поля «Фамилия» и укажите тип — его можно выбрать из раскрывающегося списка. После этого в нижней части окна откроется бланк свойств данного поля. Для текстового поля надо указать размер поля — допустимую длину значения. При создании числового поля надо указать формат представления числа, например целое. Клетки в столбце «Описание» заполнять необязательно.

Второе поле «Имя» тоже текстовое.

Для поля «Дата/время» (вы же будете указывать дату рождения) можно выбрать *Краткий формат даты*, тогда компьютер будет ждать от вас введения даты в формате ДД.ММ.ГГГГ, где ДД — число, когда вы родились (например, 05, если это произошло пятого числа), ММ — номер месяца (например, 07, если это произошло в июле) и ГГГГ — четырехзначный номер года.

Приступая к описанию параметра «Адрес», подумайте, целесообразно ли отвести под него одно поле. Ведь может потребоваться отыскать всех, кто живет на какой-либо одной улице.

Так, обдумывая каждый параметр, создайте макет вашей базы данных.

2 Завершив создание макета, сохраните его (через меню *Файл–Сохранить как...*). Напомним, что расширение этому имени СУБД присвоит сама.

3 Закройте конструктор.

4 Откройте файл с созданным макетом базы данных.

5 Заполните ее подготовленной вами информацией.

6 Сохраните созданную вами БД.

▶ Теперь у вас имеется несложная база данных, которой может пользоваться даже непрофессионал. ◀

Лабораторная работа 5 (к § 12)

Поиск информации в базе данных

Как уже говорилось, главная задача любой СУБД — выдавать информацию на запросы пользователя. Ответом на запрос будет также таблица, содержащая только те атрибуты и строки, которыми мы интересуемся. Этому действию соответствует так называемый **запрос на выборку**.

Но прежде чем осваивать формирование запроса на выборку, мы научим вас создавать фильтр — как вы позже увидите, действия будут очень похожими.

Если воспринимать СУБД как некоторого формального исполнителя, а это так и есть, то фильтрация — это не что иное, как выполнение команды «Если (строка таблицы содержит нужную информацию), то ее включить в таблицу-результат». Но в условном операторе в качестве условия должно фигурировать высказывание, проверка истинности которого является допустимым действием исполнителя. Как и в большинстве языков программирования, такое высказывание записывается в виде выражения, составленного из простейших высказываний с помощью операций И, ИЛИ и НЕ. Простейшие же высказывания построены так:

— если атрибут числового типа, то пишется выражение, в котором значение атрибута сравнивается с заданным числом (при помощи операций =, <, >, $\leqslant$ и $\geqslant$);

— если атрибут текстовый, то пишется выражение, в котором значение атрибута сравнивается с заданным значением только при помощи операции =.

Например, для числового атрибута Вес можно написать выражение Вес > 3, а для текстового атрибута Имя можно написать Имя = = Мария. Как было сказано, такие простейшие выражения могут быть соединены в более сложные с помощью логических операций. Например, (НЕ (Вес > 90)) И (НЕ (Имя = Вася)). Это, как вы понимаете, означает, что разыскивается информация о человеке, вес которого не больше 90 (килограммов), причем его не зовут Вася.

Таблица КП.4

Поле	Вес	Имя		
Сортировка				
Условие отбора	> 90	Вася		
или	<=50	Мария		

Как же оформляется фильтр в Access? Для этого в Access есть несколько возможностей. Сейчас мы рассмотрим только одну из них — использование так называемого бланка QBE (от Query By Example — запрос по образцу). Для этого надо воспользоваться специальным окном *Фильтр*, вызываемым командой *Записи-Фильтр-Расширенный фильтр* (меню *Записи* появляется при открытой таблице).

Окно фильтра состоит из двух частей. В верхней части находится прямоугольник, содержащий имя таблицы и перечень атрибутов; в нижней части находится сам бланк QBE. Его вид представлен в виде таблицы КП.4.

Как видно, бланк QBE начинается строкой ПОЛЕ. В ней указываются атрибуты, по которым осуществляется либо фильтрация, либо сортировка. Ниже располагается строка СОРТИРОВКА, в которой, если нужно, указывается порядок сортировки (по убыванию или по возрастанию). Еще ниже располагается строка УСЛОВИЕ ОТБОРА и несколько строк ИЛИ (из них обычно видна только одна).

Фактически бланк QBE — это снова таблица, в клетках которой для атрибута указываются операции сравнения и значения, с которыми сравниваются значения атрибута. Каждая клетка тем самым представляет простейшее логическое выражение или его отрицание, а имя атрибута вынесено как заголовок столбца. Выражения из клеток, стоящих в одной строке, соединяются операцией И, т. е. каждая строка — это некоторое сложное логическое выражение. А вот между собой эти «строковые» выражения соединяются уже союзом ИЛИ, о чем и напоминает слева от строки стоящее слово. Таким образом, на приведенном в качестве примера бланке QBE одна из строк задает логическое выражение (Вес > 90 И Имя = Вася). Вы вправе спросить, откуда взялся знак =, связывающий атрибут Имя со значением Вася. Ответ прост: разработчики Access для облегчения жизни пользователю разрешили его не писать на этом бланке (но и не запретили его писать!). Для другой строки получается выражение (Вес $\leqslant$ 50 И Имя = Мария). Теперь эти два выражения надо соединить союзом ИЛИ.

Возникает вопрос: всякое ли условие, выражаемое каким-нибудь логическим выражением, можно записать в таком виде, чтобы отрицания относились только к простейшим выражениям, союзом И связывались только простейшие выражения и их отрицания и уже на последнем месте шла операция ИЛИ? Ответ на этот вопрос утвердительный, но мы обоснуем его в § 32.

Если вы намерены данным фильтром пользоваться неоднократно, то его можно сохранить как запрос на выборку. Однако, открыв окно запроса-выборки, вы обнаружите на бланке QBE еще одну строку — Вывод на экран. Если в этой строке в клетке соответствующего атрибута стоит перечеркнутый квадратик — так называемый **флажок**, то значения этого атрибута будут выводиться на экран после выполнения данного запроса. Это весьма удобно — на экран могут не поместиться все атрибуты, которых может оказаться гораздо больше, чем нужно для получения интересующей вас информации. В остальном работа по созданию и использованию запроса на выборку совершенно аналогична работе с фильтром.

Откройте созданную вами базу данных и выполните в ней следующие задания:

1 Попытайтесь с помощью базы данных составить список всех мальчиков вашего класса. Для этого создайте соответствующий фильтр так, как это объяснено выше.

2 Определите, у кого лучшие результаты в прыжках в длину (или в беге на 60 м). Для этого в меню выберите режим Записи, а затем режим Сортировка.

3 Подумайте, как сделать так, чтобы в таблице сначала шли записи про всех девушек, а затем про всех юношей. Потом вернитесь к алфавитному расположению фамилий в записях.

4 Попытайтесь организовать получение ответов на все вопросы, сформулированные в задании 10 из § 12.

Лабораторная работа 6 (к § 13)

Компьютерная обработка экспериментальных данных

В § 13 мы достаточно подробно обсудили, как на основе экспериментальных данных разыскивать коэффициент прямо пропорциональной зависимости k. Физический смысл коэффициента зависит от рассматриваемой задачи; в нашем случае это была величина, обратная сопротивлению проводника электрического тока.

1 Заполните электронную таблицу так, как указано в таблице 2.4, приведенной в объяснительном тексте § 13, и найдите величину сопротивления.

Вычисление по методу наименьших квадратов легко провести и без электронной таблицы, например с помощью инженерного калькулятора. Таблица удобнее, поскольку при проведении дополнительных опытов вы просто дописываете данные в последующие строки и меняете параметры суммирования в формуле, помещенной в ячейку E1.

▶ Обсудим теперь другой способ выбора наилучшего приближения к экспериментальным данным. Рассмотрим еще раз рисунок 2.3. В методе наименьших квадратов величина коэффициента k выбиралась так, чтобы минимизировать сумму квадратов отклонений. Но можно руководствоваться другим критерием наиболее подходящего значения k. При заданном значении k выберем из всех отклонений максимальное по абсолютной величине. Для каждого значения это k будет различным. А теперь выберем k так, чтобы это число оказалось наименьшим. В этом случае говорят, что мы разыскиваем **равномерное приближение**. Однако нельзя указать формулу для вычисления k. А электронные таблицы благодаря запрограммированным в них математическим методам позволяют решить эту задачу. Продемонстрируем это на том же примере поиска сопротивления проводника по экспериментальным данным. ◀

2 Заполните электронную таблицу так, как показано в таблице КП.5. Ячейка С1 отведена под разыскиваемое значение коэффициента k (сейчас там записан 0). В ячейках С2—С11 находятся абсолютные значения отклонений отношений силы тока к напряжению от коэффициента k. В ячейке С12 вычисляется максимум этих отклонений. Именно значение в этой ячейке мы хотим минимизировать за счет изменения значения ячейки С1. Функцию, значение которой требуется минимизировать или максимизировать, называют **целевой**. Так что в ячейке С12 у нас записана целевая функция.

Таблица КП.5

	A	B	C
1	Напряжение	Сила тока	0
2	$<U_1>$	$<I_1>$	ABS(B2–C1*A2)
3	$<U_2>$	$<I_2>$	ABS(B3–C1*A3)
...	...	...	...
11	$<U_{10}>$	$<I_{10}>$	ABS(B11–C1*A11)
12			МАКС(C2:C11)

▶ А теперь обратимся к специальной программе, которая называется *Поиск решения*. Эта программа является дополнительной, поэтому надо убедиться, что она была установлена при инсталляции пакета MS Office. Для этого в меню *Сервис* выберите пункт *Надстройки*. В открывшемся диалоговом окне против этой программы, если она доступна, должна стоять галочка. Если галочки нет, то поставьте ее. Обращение к программе осуществляется выбором в меню *Сервис* пункта *Поиск решения*. ◀

3 Приступим к поиску k.

Для этого нужно выполнить следующие действия:

1) Установите курсор в ячейку C12.
2) Вызовите подпрограмму *Поиск решения.*
3) В открывшемся диалоговом окне укажите адрес ячейки целевой функции C12, адрес изменяемой ячейки C1 (если изменяться будет несколько чисел, т. е. у целевой функции несколько аргументов, то в поле «Изменяя ячейки» записывается область изменяемых ячеек).
4) Щелчком мыши отметьте, что мы разыскиваем минимум.
5) Щелчком по кнопке *Выполнить* запустите программу на исполнение.

Найденное значение k появится в окне Результат поиска. Вы можете сохранить это значение в ячейке C1, щелкнув по кнопке ОК.

4 Сравните это значение коэффициента k с тем, которое получилось по методу наименьших квадратов. Какому из них отдать предпочтение, решать вам.

5 А теперь постарайтесь с помощью электронной таблицы закончить выполнение задания 3 к этому же параграфу. Коэффициенты a и b найдите двумя способами: по методу наименьших квадратов и как равномерное приближение.

Лабораторная работа 7 (к § 14)

Метод пошаговой детализации

Перед тем как приступить к выполнению лабораторной работы, вспомните, как в языках Бейсик и Паскаль оформляются подпрограммы. Основные операторы для этих языков приведены на форзацах вашего учебника.

А теперь выполните несколько заданий.

1 Используя ваше решение задачи 3 из § 14, напишите программы, реализующие решение задачи о нахождении разложения натурального числа на простые множители. Отладьте эти программы.

2 Запрограммируйте решение задачи 5 из § 14. Отладьте составленные вами программы.

3 Запрограммируйте решение задачи 6 из § 14. Отладьте составленные вами программы.

4 Запрограммируйте решение задачи 8 из § 14. Отладьте составленные вами программы.

5* Запрограммируйте решение задачи 7 из § 14. Отладьте составленные вами программы.

6* Запрограммируйте решение задачи 9 из § 14. Отладьте составленные вами программы.

Лабораторная работа 8 (к § 15)

Рекуррентные соотношения и рекурсивные алгоритмы

Вы еще мало знаете о программировании, но уже вполне достаточно, чтобы без подсказок выполнить следующие задания.

1 Запрограммируйте решения задач 4а, 4б и 4в из § 15. Отладьте составленные вами программы.

2 Напишите программы, реализующие алгоритм из задания 6 и рекурсивную подпрограмму-функцию ФОК. Отладьте их.

3 Запрограммируйте и отладьте составленный вами алгоритм, решающий ту же задачу.

4 Проведите сравнение эффективности этих двух алгоритмов по той же схеме, как в задании 3 лабораторной работы 3.

5 Запрограммируйте решение задачи 7 из § 15. Отладьте составленные вами программы.

6 Для программирования решения задачи 8 из § 15 договоримся, что диски на стержне перенумерованы числами 1, 2, 3, ..., n, начиная с верхнего диска. Запишите и отладьте программу решения задачи о ханойской башне.

7 Запрограммируйте решение задачи 9 из § 15. Отладьте составленные вами программы.

Лабораторная работа 9 (к § 16)

Программы для обработки массивов

Одной из наиболее часто используемых структур данных являются массивы. Каждый массив имеет имя, тип и размерность. Размерность массива — это количество индексов, которыми снабжен каждый элемент массива. Для каждого индекса требуется указать диапазон его изменения.

Для выполнения заданий этой лабораторной работы нужны заполненные массивы. Заполнять массивы вручную — дело долгое и малоинтересное. Поэтому мы предлагаем создавать массивы программным путем. Алгоритмы заполнения массивов включены в тексты заданий. А теперь выполните несколько заданий.

1 Запрограммируйте решения задач 7а и 7б из § 16. Отладьте составленные вами программы на массивах с небольшим числом элементов (эти отладочные массивы создайте самостоятельно).

2 Создайте массив $A[1:10, 1:10, 1:10]$, запрограммировав следующий алгоритм:

```
Алгоритм
  цел: I, J, K; вещ: A[1:10, 1:10, 1:10];
{ Делать от I := 1 до 10
  { Делать от J := 1 до 10
    { Делать от K := 1 до 10
      { A(I, J, K) := 2*COS (I + 2*J) + 7*SIN(I^2/K);
      }                                              (* конец цикла *)
    }                                                (* конец цикла *)
  }                                                  (* конец цикла *)
}
```

Решите для этого массива задачи 7а и 7б.

3 Запрограммируйте решения задач 8а и 8б из § 16. Отладьте программы на массивах с небольшим числом элементов.

4 Создайте массив $M[1:100, 1:20, 1:75]$, запрограммировав следующий алгоритм:

```
Алгоритм
цел: I, J, K; вещ: M[1:100, 1:20, 1:75];
{ Делать от I := 1 до 100
  { Делать от J := 1 до 20
    { Делать от K := 1 до 75
      { M(I, J, K) := K*COS (I + 2*J) + J*SIN(I^2/K);
      }                                              (* конец цикла *)
    }                                                (* конец цикла *)
  }                                                  (* конец цикла *)
}
```

Решите для этого массива задачи 8а и 8б.

5 Запрограммируйте решение задачи 9 из § 16. Отладьте программу на массиве с небольшим числом элементов.

6 Создайте массив $M[1:2, 1:100]$, запрограммировав следующий алгоритм:

```
Алгоритм
цел: I, J; вещ: M[1:2, 1:100];
{ Делать от I := 1 до 2
  { Делать от J := 1 до 100
    { M(I, J) := COS (I + 2*J) + J*SIN(J^2/I);
    }                                                (* конец цикла *)
  }                                                  (* конец цикла *)
}
```

Решите для этого массива задачу 9.

7 Пусть прямоугольная таблица, о которой идет речь в задании 10 из § 16, имеет 10 строк и 20 столбцов. Тогда она представляет-

ся некоторым массивом $A[1:10, 1:20]$. Запрограммируйте решение задачи для этого массива. Отладьте составленную вами программу на массиве с небольшим числом элементов.

8 Создайте массив $M[1:10, 1:20]$, запрограммировав следующий алгоритм:

```
Алгоритм
цел: I, J, A[1:10, 1:20];
{ Делать от I := 1 до 10
  { Делать от J := 1 до 20
    { A(I, J) := INT(COS (I + 2*J) + J*SIN(J^2/I));
    }                                              (* конец цикла *)
  }                                                (* конец цикла *)
}
```

Решите для этого массива задачу 10.

Лабораторная работа 10 (к § 17)

Решение уравнений

В объяснительном тексте § 17 приведен алгоритм решения уравнения методом деления пополам.

1 Запрограммируйте этот алгоритм для $f(x) = x - \sin x - 2\pi(1 - d)$. Отладьте программу и получите ответ для $d = 0{,}8$. Вычислите глубину погружения бруска, взяв радиус бревна 0,2 м.

2 Запрограммируйте алгоритм, который вы составили, выполняя задание 4в из § 16.

▶ У вас может возникнуть вопрос: неужели программирование — единственная компьютерная технология, позволяющая пользователю решать уравнения? Конечно нет. Уравнения можно эффективно решать, используя электронную таблицу.

Можно воспользоваться режимом *Поиск решения*, который вы осваивали, выполняя лабораторную работу 6. Возможно, вы обратили внимание, что в диалоговом окне режима *Поиск решения* есть пункт *Равной значению*, после выбора которого появляется окно для ввода числа (по умолчанию там всегда стоит 0). Воспользуйтесь этим режимом и решите уравнение $x - \sin x - 2\pi(1 - d) = 0$. При этом вам не требуется помнить и вводить значение числа π, достаточно написать ПИ( ).

В описании лабораторной работы 6 мы говорили, что режим *Поиск решения* — это надстройка, и потому не обязательно входить в электронную таблицу Excel. Но в любой таблице Excel есть функция *Подбор параметра*, позволяющая автоматически подобрать значение в изменяемой ячейке так, чтобы в ячейке, где записана целевая функция, оказалось требуемое ее значение. Вызов этой функции

осуществляется через меню *Сервис* и пункт *Подбор параметра*. Например, для решения уравнения $x - \sin x - 2\pi(1 - d) = 0$ электронную таблицу можно заполнить так, как показано в таблице КП.6.

Таблица КП.6

	A	B
1	<x>	
2	<d>	A1–SIN(A1)–2*ПИ()*(1–A2)

В ячейку A2 нужно ввести значение удельной плотности дерева, ячейка A1 является изменяемой, а в ячейке B2 записана целевая функция, значение которой нужно сделать равным 0. ◀

3 Найдите значение x с помощью функции *Подбор параметра*.

Лабораторная работа 11 (к § 20)

Модель движения в среде с сопротивлением

Прежде всего выберите, какую компьютерную технологию вы намерены использовать для моделирования движения в среде с сопротивлением (будет интереснее, если какая-то часть класса воспользуется программой, а какая-то — электронной таблицей).

1 Если вы предпочитаете язык программирования, то воспользуйтесь своим решением задачи 2 из § 19 и отладьте составленную вами программу.

Если вы решили применить электронную таблицу, то заполните ее так, как опять-таки сказано в § 19. Правда, у вас может возникнуть вопрос: сколько строк в этой таблице надо заполнить? В программе написан цикл в форме «пока», в электронной таблице так не сделаешь (точнее, вы еще этого делать не умеете). Ответ наш таков: заполните пока 200 строк.

2 А теперь начинаем компьютерный эксперимент.

В качестве первого опыта выберите $v = 600$, $h = 1300$ и $A = 0$. Это означает, что тело начинает свое движение на высоте 1300 м со скоростью 600 м/с и в горизонтальном направлении.

Тот, кто воспользовался программой, получил три столбика цифр. А те, кто использовал электронную таблицу, получили пять столбиков. Хватило ли строк, чтобы тело «долетело» до поверхности Земли? Если нет, скопируйте еще строки столько раз, сколько окажется необходимым.

3 Сравните, одинаковое ли получилось время «приземления». А «дальность» полета? Надеемся, что результаты у вас оказались близкими независимо от использовавшейся технологии. Если нет, то разберитесь, у кого ошибка.

▷ Разглядывая столбики цифр, вы должны были еще раз убедиться в том, что человек лучше приспособлен к восприятию визуальной информации, нежели символьной. Мы знаем теперь, сколько времени продлится «полет» и как далеко «улетит» тело, но проанализировать характер движения уже весьма непросто.

Потребуем от компьютера, чтобы он изображал получаемые данные в виде графика. Чтобы осуществить это в программном варианте, потребуются графические операторы. Некоторые из них приведены в таблице КП.7.

Операторы включения графики, очистки экрана и назначения системы координат должны стоять в самом начале программы. После них, пользуясь оператором построения отрезка, можно создать на экране систему координат. А вот построение самого графика — это фактически замена в программе оператора **Сообщить** оператором вывода на экран точки с нужными координатами.

В случае применения электронной таблицы построение графика надо поручить самой таблице. Напомним, как это делается.

Нас в первую очередь интересует траектория полета тела. Поэтому выделите столбцы, содержащие высоту и дальность, т. е. блок ячеек B2:C200 (или большее значение второго параметра при C, если потребовалось больше строк). Щелкните по кнопке *Мастер диаграмм*. Дальнейшие действия осуществляются по шагам в соответствии с указаниями *Мастера диаграмм*.

1) Выбор типа диаграммы. Выбираем тип *Точечная*. Выберите вид со сглаживанием. Щелкните по кнопке *Далее*.

2) Выбор источника данных. Фактически вы этот шаг выполнили выделением блока ячеек. Но если что-то не устраивает (например, диапазон участвующих ячеек), можно поменять, не выходя из *Мастера диаграмм*. Щелкните по кнопке *Далее*.

3) Выбор параметров. Здесь вы можете сделать все поясняющие надписи, установить или отменить линии сетки и т. п. Щелкните по кнопке *Далее*.

4) Выбор места размещения. Обычно размещают на том же листе, где и данные для графика. Отметьте этот вариант и щелкните по кнопке *Готово*. У вас на экране появится траектория движения тела. ◁

4 Если вы использовали программу, то модифицируйте ее соответствующим образом и отладьте. Для электронной таблицы постройте график, как показано выше.

5 Поэкспериментируйте с разными значениями начальной скорости и начальной высоты. Похожи ли, на ваш взгляд, полученные графики на привычную параболу?

▷ Мнения, скорее всего, разделились. Чтобы ответ стал более очевидным, попробуем «бросать» тело под ненулевым углом к горизонту. Но надо иметь в виду, что угол мы измеряем в градусах, а функции синус и косинус воспринимают аргумент в радианах.

Таблица КП.7

Назначение	Qbasic	Pascal

Включение и завершение графического режима	SCREEN 12	Uses Graph;
		Var gd,gm:integer;
		Begin
		gd:=detect;
		InitGraph(gd,gm,'BGI');
		(Вместо 'BGI' должен стоять путь к папке, где расположен графический инициализатор egavga.bgi, но обычно это папка BGI в корне Паскаля) CloseGraph; End

Очистка экрана	CLS	
Назначение новой системы координат	WINDOW(X1,Y1)–(X2,Y2) (X1 — значение абсциссы самой левой точки экрана, X2 — значение абсциссы самой правой точки экрана, Y1 — значение ординаты самой левой точки экрана, Y2 — значение ординаты самой правой точки экрана)	
Изображение точки заданного цвета	PSET(X,Y),Z (X — абсцисса точки, Y — ордината точки, Z — номер цвета; числу 0 соответствует черный цвет, 1 — синий, 4 — красный, 15 — белый)	PutPixel(X, Y, Z); (X — абсцисса точки, Y — ордината точки, Z — номер цвета; числу 0 соответствует черный цвет, 1 — синий, 4 — красный, 15 — белый)
Изображение отрезка с заданными концами	LINE(X1,Y1)–(X2,Y2),Z (X1;Y1 — координаты одного конца отрезка, X2, Y2 — координаты другого конца отрезка, Z — номер цвета)	line(X1,Y1,X2,Y2); (X1;Y1 — координаты одного конца отрезка, X2,Y2 — координаты другого конца отрезка)

Для перевода из градусов в радианы в Excel есть специальная функция РАДИАНЫ(x). Чтобы вычисления были правильными, в ячейки D2 и E2 надо ввести формулы G5*SIN(РАДИАНЫ(G6)) и G5*COS(РАДИАНЫ(G6)) соответственно. ◀

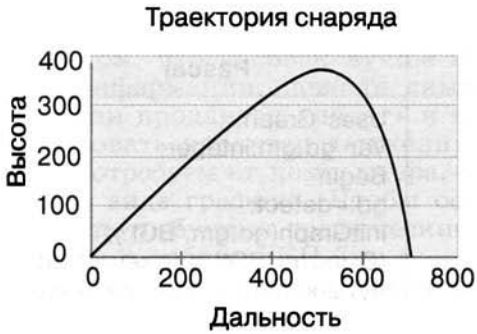


Рис. КП.2 Движение тела под углом 45° с начальной скоростью 600 м/с

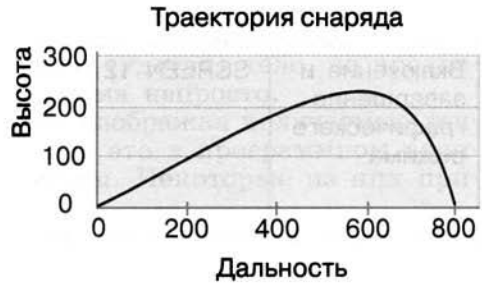


Рис. КП.3 Движение тела под углом 30° с начальной скоростью 600 м/с

6 Теперь проведите вычисления при $v = 600$, $h = 0$ и $A = 45$. Траектория получилась такой, как на рисунке КП.2. На параболу эта линия не похожа.

7 Еще один эксперимент — $A = 30$. Траектория для этого случая показана на рисунке КП.3.

► Мы уверены, что результаты, полученные вами, дают такую же картину. Обратите внимание: горизонтальная составляющая скорости с течением времени стремится к нулю, движение становится практически отвесным (т. е. тело перестает перемещаться в горизонтальном направлении), а сама скорость становится практически постоянной. Иными словами, с течением времени движение в среде с сопротивлением становится практически *вертикальным* и *равномерным*. Вот какой удивительный факт мы обнаружили в результате компьютерного моделирования данного процесса.

Равномерность движения тела в среде с сопротивлением означает, что сила тяжести уравновешивается силой сопротивления среды. Иными словами, при таком движении имеет место равенство

$$mg - mkv^2 = 0.$$

Отсюда $v = \sqrt{\frac{g}{k}}$. Таким образом, скорость тела, движущегося в среде с сопротивлением, с течением времени должна быть все ближе к величине $\sqrt{\frac{g}{k}}$. ◄

8 Проверьте, выполнено ли приближенное равенство $v_{\text{кон}} \approx \sqrt{\frac{g}{k}}$ в расчетах при достаточно долгом движении тела в среде с сопротивлением. Если оно наблюдается, то это важный аргумент в пользу адекватности построенной нами компьютерной модели.

▶ Конечно, наш компьютерный эксперимент *не доказывает*, что обнаруженное с его помощью свойство движения тела в среде с сопротивлением имеет место всегда. И сколько бы экспериментов с различными начальными условиями мы ни провели, всегда будет грызть червячок сомнения, а вдруг при каких-то начальных данных обнаруженное свойство отсутствует.

И тут на помощь физике приходит математика. Именно в математике разработаны методы исследования поведения функции. На уроках математики вы учились для функций, заданных какой-либо формулой, исследовать ее поведение и строить график. В нашем случае никакой формулы, определяющей зависимость скорости от времени, нет; есть только некоторые соотношения, связывающие скорость и ускорение. Математики разработали методы, позволяющие по таким соотношениям определять, какими свойствами обладает функция. В рассматриваемом случае можно доказать, что обнаруженные свойства справедливы при любых начальных значениях высоты, скорости и угла наклона.

В результате исследования движения в среде с сопротивлением мы пришли к выводу, что со временем это движение становится близким к равномерному движению. Не означает ли это, что можно было с самого начала считать модель равномерного движения вполне адекватной данному процессу? Подумайте над этим вопросом и постарайтесь дать аргументированный ответ.

Продолжим компьютерный эксперимент. ◀

9 Попытайтесь выяснить, при каком угле наклона дальность полета будет наибольшей. Начальная скорость и начальная высота прежние: $v = 600$, $h = 0$. Для этого можно применить метод деления пополам или воспользоваться режимом электронной таблицы Поиск решения.

У нас получилось $\approx 29°$. А какой ответ у вас?

10 А теперь попробуем выяснить, какова начальная скорость снаряда из задания 3 § 19. Для этого воспользуйтесь вашим решением задачи 3 из § 20.

11 Если вам хочется узнать, насколько нагревается тело во время полета в среде с сопротивлением, реализуйте компьютерную модель, созданную вами при выполнении задания 4 из § 19.

Лабораторная работа 12 (к § 21)

Модели неограниченного и ограниченного роста

В § 21 описаны две модели роста популяции живых организмов. Проведем с ними компьютерный эксперимент. И снова здесь можно применить две технологии: электронные таблицы и язык программирования. Чтобы было интереснее, разделитесь в классе так,

Таблица КП.8

Природная зона	Коэффициент k
Тундра	0,6
Тайга	1,8
Степь	1,2
Пустыня	0,8

чтобы одна часть из вас работала с электронной таблицей, а другая — с языком программирования. Ниже мы приводим задания, описывая их выполнение с помощью электронной таблицы. Надеемся, что выполнить те же задания, используя программу, составленную при выполнении задания 5 из § 21, для вас не составит труда.

Для проведения компьютерного эксперимента надо знать исходные данные: $M(0)$, k, a, L. В качестве $M(0)$ возьмем массу, равную 1. В конце концов для нашего эксперимента совершенно безразлично, будет ли это 1 грамм, или 1 тонна, или 1 мегатонна. Аналогично величина верхней границы L тоже не имеет принципиального значения. Пусть L будет, например, 11 000. А вот коэффициент прироста весьма существенный параметр процесса. Поэтому предусмотрительно запасемся значениями коэффициента k, экспериментально полученными учеными-биологами для растений в разных природных зонах (мы уже отмечали, что не играет роли, какие именно живые организмы рассматриваются в этой модели, пусть это будут растения). Значение k для различных природных зон приведено в таблице КП.8.

Теперь, используя эти числа, попытайтесь предсказать, через сколько лет масса растений превысит 100, если первоначально масса растений на некотором участке была всего 1. Запишите ваш прогноз, чтобы потом сравнить его с результатом, полученным на компьютере.

Осталось определить коэффициент a. Из формулы $k(n) = a(L - M(n))$ видно, что для нахождения коэффициента a достаточно разделить $k(0)$ на $L - M(0)$. В свою очередь, число $k(0)$ — это коэффициент прироста в самом начале процесса, и потому его нужно взять равным числу k из таблицы КП.8.

Приступим к компьютерному эксперименту.

1 Загрузите электронную таблицу и занесите в нее необходимые данные так, как показано в таблице КП.9.

▷ Как же узнать, через сколько лет масса растений в тундре превысит 100 т? Представим, что в таблице заполнены все строки, начиная с 7-й. Тогда достаточно в столбце C найти ячейку, когда первый раз встретится число, большее 100, и в той же строке в первом столбце мы увидим год, в котором масса растений превысит 100 т. Точно так же можно поступить со столбцами, в которых отображена информация для других природных зон.

Конечно, заполнить все строки таблицы сразу слишком трудоемкая да, наверно, и ненужная работа. Проще поступить так: последовательно копировать блок ячеек A7 : J7 в последующие строки.

Таблица КП.9

	A	B	C	D	E	F
1	год	Природная зона	Тундра		Тайга	
2		Тип модели	Неогр.	Огран.	Неогр.	Огран.
3		Коэффициент размножения k	0,6	0,6	1,8	1,8
4		Предельное значение массы L		11 000		11 000
5		Коэффициент a		D3/(D4–D6)		F3/(F4–F6)
6	0	Начальная масса $M(0)$	1	1	1	1
7	A6+1	Масса через 1 год	C6*(1+C3)	D6*(1+D5*(D4–D6))	E6*(1+E3)	F6*(1+F5*(F4–F6))
8	A7+1	Масса через 2 года	C6*(1+C3)	D7*(1+D5*(D4–D7))	E6*(1+E3)	F7*(1+F5*(F4–F7))
9	A8+1					

(Подумайте, адреса каких ячеек при таком копировании не должны меняться.) Как только во всех столбцах C, D, E, F, G, H, I, J возникнут числа, большие 100, копирование можно прекратить. ◀

2 Проделайте указанную работу с таблицей. Для каждой природной зоны запишите год, когда произошло превышение числа 100. Сравните получившиеся данные с вашим прогнозом, записанным ранее.

3 Теперь после первого опыта попробуйте угадать, когда масса растений станет равной 1000 т.

▶ Проверьте свой прогноз с помощью электронной таблицы или программы. (Интересно, улучшились ли ваши прогностические способности?)

Скорее всего, теперь ваш прогноз оказался завышенным (как и у авторов этой книги, проводивших такой же компьютерный эксперимент). Удивительный факт, не правда ли: масса увеличилась в 10 раз, а на ее «производство» потребовалось всего лишь два-три дополнительных года. Полученные результаты тоже запишите к себе в тетрадь.

На следующее «удесятерение» массы живых организмов в модели неограниченного роста понадобится тоже два-три года (убедитесь в этом сами и внесите результаты в таблицу). ◀

4 Каждая современная электронная таблица позволяет автоматически построить график зависимости между величинами, вычисленными с ее помощью. Воспользуйтесь этой возможностью и постройте графики зависимости массы растений от числа прошедших лет (для каждой из природных зон). Какие общие особенности этих графиков вы смогли обнаружить?

▶ Разница в поведении этих моделей налицо. В модели ограниченного роста время «удесятерения» растет, причем быстрее для тех растений, где коэффициент k меньше. Это неудивительно: чем медленнее рост, тем больше времени требуется для «удесятерения» массы. Этот качественный анализ различий между моделями неограниченного и ограниченного роста мы в следующей лабораторной работе дополним количественным анализом. ◀

Лабораторная работа 13 (к § 22)

Поиск границ адекватности модели

Исследовать модель неограниченного роста на адекватность мы будем уже при известных нам параметрах: значение k возьмем 1,8 (как для тайги), а L будем считать равным 11 000.

1 Заполните электронную таблицу, как показано в таблице 3.2 из § 22, и внесите исходные данные. Последовательно копируя

блок A4:D4 в последующие строки, найдите, в какой год отклонение превысит границу 10%. (При этом не забудьте указать электронной таблице, адреса каких ячеек не должны меняться при копировании.)

Теперь исследуйте значение k, равное 1,2.

2 Скопируйте блок B1:D... (здесь вы должны указать номер последней заполненной строки в таблице), начиная с клетки F1. Запишите в F1 коэффициент 1,2. В каком году отклонение превысит границу 10%? Пришлось ли вам еще копировать строки? Уже эти два компьютерных эксперимента показывают, что с уменьшением k граница n отодвигается.

3 Для подкрепления этого вывода проведите компьютерный эксперимент еще и при $k = 1$.

▶ Теперь можно исследовать, как граница адекватности зависит от величины L. Общие соображения подсказывают, что с ростом L граница n должна увеличиваться. Но каков характер этой зависимости? Давайте, к примеру, удвоим значение L. ◀

4 Введите удвоенное значение L при $k = 1$.

Посмотрите, граница отодвинулась на один год.

5 А если еще раз удвоить? Проведите этот эксперимент.

Опять граница отодвинулась на один год.

6 А если исходное L уменьшить вдвое? Проведите эксперимент при $L = 5500$ (но по-прежнему $k = 1$).

▶ Эффект тот же самый — теперь граница n уменьшилась на 1. Напрашивается гипотеза, что L образует геометрическую прогрессию относительно границы адекватности n. Если вспомнить формулу общего члена геометрической прогрессии, то получим

$$L = b \cdot 2^{n-1},$$

где b — некоторый коэффициент. Найти этот коэффициент нетрудно — достаточно разделить L на соответствующую степень 2.

Задумаемся, однако, над полученной формулой. Ведь в общей формуле для L, наверно, еще должно участвовать k. Учитывая, что при $k = 1$ выполнено соотношение $2 = 1 + k$, можно предположить, что

$$L = b(1 + k)^{n-1}.$$

Осталось найти b и убедиться, что этот коэффициент практически не зависит от k. ◀

7 Найдите b при $L = 5000$ при различных k: 1; 1,2; 1,5; 2. Убедитесь, что во всех случаях $b \approx 8$.

▶ Итак, наши компьютерные эксперименты показали, что моделью неограниченного роста можно пользоваться с уровнем по-

грешности 10% при выполнении условия $L \geqslant 8(1 + k)^{n-1}$. Тот, кто знает, как решаются показательные неравенства, легко получит явное выражение для n, показывающее, как долго можно пользоваться моделью неограниченного роста при заданных L (предельный уровень массы живых организмов) и k (коэффициент ежегодного прироста). Вот эта формула:

$$n \leqslant 1 + \frac{\lg(0{,}125L)}{\lg(1 + k)}.$$

Цель, поставленная в начале лабораторной работы, достигнута. Но мы надеемся, что вы в ходе выполнения этой работы научились большему. Вы наблюдали, как с помощью средств информационных технологий исследуется характер зависимости между различными переменными, выдвигаются, а затем проверяются в компьютерном эксперименте гипотезы о формуле для этой зависимости. Здесь и сейчас вы получили первый опыт этой интересной исследовательской деятельности.

Возможно, не всех устроил наш поиск на глазок коэффициента b, да и уверенности в выражении $1 + k$ тоже нет. Вот только характер зависимости $L = b(1 + sk)^{n-1}$ сомнений не вызывает. Поступим так. Прологарифмируем это равенство и обозначим $\lg L$ через y, $\lg b - (1 + sk)$ через c, а $1 + k$ через d. Получаем зависимость $y = c + dn$. Это линейная зависимость, для которой из экспериментальных данных требуется подобрать коэффициенты c и d. Вспомните, такую задачу мы обсуждали в § 13, а о ее компьютерной реализации рассказали в лабораторной работе 6. ◀

8 Организуйте подбор коэффициентов c и d одним из описанных ранее способов. По полученным значениям c и d найдите коэффициенты b и s. Значительно ли отличается b от 8, а s от 1?

Лабораторная работа 14 (к § 23)

Компьютерная модель эпидемии гриппа

1 Как сказано в § 23, компьютерная модель эпидемии гриппа строится на основе модели ограниченного роста. Реализуйте сначала компьютерную модель, описываемую формулой

$$B(n + 1) = \left(1 + k\frac{L - B(n)}{L}\right)B(n).$$

2 Проведите с ней эксперимент.

▶ Этот эксперимент показал, что модель неадекватна реальной ситуации: все болеют и болеют, эпидемия никак не заканчивается. Причина неадекватности понятна — не учли существенный фактор, что люди выздоравливают. ◀

3 Корректировка модели проведена в объяснительном тексте § 23. Постройте соответствующую компьютерную модель и проведите с ней эксперимент. Для наглядности постройте графики, отражающие количество больных в разные дни эпидемии.

Лабораторная работа 15 (к § 25)

Проверяем датчик случайных чисел

Любой пользователь с доверием относится к программному обеспечению компьютера, которым он пользуется. Однако, как говорится, доверяй, но проверяй. Мы вовсе не призываем вас с недоверием относиться к широко применяемым инструментам информационных технологий. Но надо помнить, что датчики случайных чисел всего лишь моделируют случайность, и потому совсем не факт, что псевдослучайная последовательность, которую вырабатывает стандартный ДСЧ, годится для решаемой вами задачи.

Между тем нетрудно проверить, хороша ли вырабатываемая ДСЧ псевдослучайная последовательность.

Есть такая игра — «Орлянка». Условия игры таковы: берут монетку и подбрасывают ее. Если выпадает орел, то выигрывает один игрок, если решка — выигрывает другой.

А теперь смоделируем эту игру с помощью ДСЧ. Поскольку датчик вырабатывает числа от 0 до 1, то договоримся так: если получается число, меньшее 0,5, то, значит, выпадает орел, а в противном случае — решка.

Тем самым алгоритм таков:

```
Алгоритм Орлянка
цел: a, k, n; вещ: x;
{ Запросить n;                    (* число подбрасываний монеты *)
  a := 0;
  Делать от k := 1 до n
    { x := rand(1);
      Если x < 0,5 то { a := a + 1; }
    }
  Сообщить "Число выпадений герба", a;
  Сообщить "Частота выпадения герба", a / n;
}
```

Для перевода этого алгоритма на язык Бейсик надо использовать два новых для вас оператора RANDOMIZE и RND. Первый оператор инициализирует датчик случайных чисел. Он пишется один раз в самом начале тела программы, если в ней предполагается использовать ДСЧ. Но если вы просто напишете оператор RND, то, выполняя его, компьютер запросит у вас первое случайное число. Чтобы этого не происходило, вы можете сразу после оператора через пробел написать какое-нибудь число. Но тогда RND

будет всегда генерировать одну и ту же случайную последовательность. Это удобно для отладки программы (помните, мы обсуждали это в § 24), но совершенно не годится при моделировании случайных процессов. Поэтому обычно пишут так:

RANDOMIZE TIMER

В этом случае каждый раз, выполняя программу, компьютер в качестве начального значения будет брать текущее время. Ясно, что оно всегда разное, так что и псевдослучайные последовательности будут получаться тоже разными.

Что касается оператора RND, то его надо писать вместо каждого употребления оператора rand(1) в наших алгоритмах.

В языке Паскаль используются аналогичные операторы: Randomize и Random(a). Первый из них инициализирует датчик случайных чисел и задает первое число, определяемое внутренним временем компьютера. Второй оператор задает целое случайное число в промежутке [0; a).

А теперь заполните таблицу по образцу таблицы КП.10. У нас она получилась такой (использовался стандартный датчик фирмы Microsoft, имеющийся в языке QBasic, входящем в систему MS DOS версии 5.0).

Таблица КП.10

Количество бросаний, n	**Количество выпадений герба, a**	**Частота выпадений герба, a/n**
10	5	0,5
50	22	0,44
100	48	0,48
200	85	0,425
500	254	0,508
1000	498	0,498
2000	1000	0,5
5000	2470	0,494
10000	5031	0,5031
20000	10102	0,5051
50000	24790	0,4958
100000	50216	0,50216

Наверное, у вас получилась другая таблица. Но мы уверены, что в третьем столбце и у вас записаны числа, близкие к 0,5. Для вас это не должно быть удивительным (если вы выполнили задание 5 б из § 24) — есть два равновозможных исхода, так что вероятность выпадения герба равна 0,5.

Чем же может помочь нам эта таблица? Ясно, что совсем плохой датчик будет все время давать частоту 0,5 — это полное отсут-

ствие случайности: через раз выпадает герб, через раз — решка. Идеальный датчик должен выдавать отклонения от этой частоты. Оказывается, известно, каким должно быть это отклонение для идеального датчика. Степень удовлетворительности того или иного датчика определяется тем, с какой вероятностью отклонение, получившееся у проверяемого датчика, может быть у идеального. Иными словами, при оценке качества мы имеем дело с функцией, связывающей три параметра:

n — число бросаний;

d — отклонение среднего числа выпадения герба от 0,5;

p — вероятность того, что для идеального ДСЧ отклонение будет не меньше чем d.

Нужную нам формулу еще в начале XIX века получил французский выдающийся математик Пьер Симон Лаплас:

$$p = 2(1 - \Phi(2d\sqrt{n})).$$

В этой формуле фигурирует функция $\Phi(x)$, известная математикам еще до Лапласа, которая называется нормальной функцией распределения. Формула для этой функции использует математические символы, которые вам еще неизвестны. Но сама формула нужна лишь в теоретических исследованиях, а нам достаточно таблицы значений, которая есть в любом справочнике по математике. А вы можете получить ее значение, например, записав в ячейке электронной таблицы Excel формулу =НОРМСТРАСП(A1), если в ячейку A1 записан аргумент функции $\Phi(x)$. Для данных, приведенных в таблице КП.11, получаем следующие результаты:

Таблица КП.11

Количество бросаний, n	Частота выпадений герба, a/n	Отклонение, d	$x = 2d\sqrt{n}$	$p = 2(1 - \Phi(2d\sqrt{n}))$
10	0,5	0	0	1
50	0,44	0,06	0,85	0,396
100	0,48	0,02	0,4	0,689
200	0,425	0,075	2,12	0,034
500	0,508	0,008	0,36	0,721
1000	0,498	0,002	0,13	0,899
2000	0,5	0	0	1
5000	0,494	0,006	0,85	0,396
10000	0,5031	0,0031	0,62	0,535
20000	0,5051	0,0051	1,44	0,149
50000	0,4958	0,0042	1,88	0,060
100000	0,50216	0,00216	1,378	0,172

Какой вывод можно сделать? Хорош ли наш датчик?

Давайте рассуждать. Если, скажем, p получилось равным 0,89, то это значит, что в 89 случаях из 100 идеальный датчик будет давать отклонение большее, чем наш. Иными словами, наш датчик недостаточно случаен.

А если $p = 0{,}15$, то наш датчик опять плох — такое и большее отклонение идеальный датчик дает всего лишь в 15 случаях из 100.

Значит, для хорошего датчика значение p должно быть близким к 0,5, скажем, попадать в интервал от 0,3 до 0,7. У нас в таблице КП.11 таких случаев всего лишь 4 из 12. Более того, с ростом n ситуация не улучшается. Видно, наш датчик хоть и фирменный, но не слишком хороший. Возможно, вам повезло больше (мы специально взяли довольно старую версию программного обеспечения и в следующих лабораторных работах пользовались более надежными ДСЧ).

Лабораторная работа 16 (к § 26)

Компьютерная модель системы массового обслуживания

В объяснительном тексте § 26 построена модель системы массового обслуживания покупателей в магазине. Более того, выполнив задание 7, вы написали программу, позволяющую проводить компьютерный эксперимент.

1 Отладьте эту программу.

2 Проведите компьютерный эксперимент для предложенных ниже вариантов исходных данных.

1) $N_0 = 4$; $T_0 = 0{,}5$; $T_1 = 2$; $V_0 = 3{,}5$; $V_1 = 6$; $m = 3$;
2) $N_0 = 10$; $T_0 = 1$; $T_1 = 3{,}5$; $V_0 = 4$; $V_1 = 10$; $m = 3$;
3) $N_0 = 6$; $T_0 = 1$; $T_1 = 3{,}5$; $V_0 = 4$; $V_1 = 8$; $m = 3$;
4) какие-нибудь свои данные.

3 Запишите результаты в таблицы, подобные таблицам 3.6—3.8 из объяснительного текста § 26.

Лабораторная работа 17 (к § 26)

Моделирование броуновского движения

Приступим к компьютерному моделированию броуновского движения частицы. Выполнив задание 8 из § 26, вы построили две модели броуновского движения: в пункте а — с постоянной скоростью движения и в пункте б — со случайной скоростью движе-

ния. Сначала рассмотрим первую модель. Как и раньше, компьютерное моделирование может быть выполнено либо средствами языка программирования, либо с помощью электронной таблицы.

Если вы предполагаете воспользоваться языком программирования, то программу можете составить в соответствии с алгоритмом, который приведен ниже.

```
Алгоритм
вещ: v, x, y, f; цел: n, I;
{ Запросить v;                          (* скорость частицы *)
  Запросить n;                          (* количество толчков *)
  x := 0;                               (* начальное значение абсциссы *)
  y := 0;                               (* начальное значение ординаты *)
  Делать от I = 1 до n
    { f := 2*3.14*rand(1);              (* вычисление случайного угла,
                                           определяющего очередное направление движения *)
      x := x + v*cos(f);                (* вычисление нового значения абсциссы *)
      y := y + v*sin(f);                (* вычисление нового значения ординаты *)
      Сообщить "Абсцисса частицы", x, ", ордината частицы", y;
    }                                   (* конец цикла *)
}
```

Если же вы намерены использовать электронную таблицу, то прежде всего уточните, как в ней реализуется получение последовательности случайных чисел. В Excel она получается с помощью оператора СЛЧИС(). Подробную информацию о том, как работает этот оператор, вы можете получить, заглянув в Справку. Вот как можно заполнить таблицу, чтобы реализовать данную модель:

Таблица КП.12

	A	B	C	D
1	Время	Абсцисса	Ордината	
2	0	0	0	<v>
3	A2+1	B2+D2*COS(D3)	B2+D2*SIN(D3)	2*ПИ()*СЛЧИС()
4	A3+1	B3+D2*COS(D3)	B3+D2*SIN(D3)	
5	A4+1	B4+D2*COS(D3)	B4+D2*SIN(D3)	
...	...	...	...	

1 Проведите компьютерный эксперимент с моделью броуновского движения с постоянной скоростью для $n = 100$. Используя средства изображения зависимостей, вычертите траекторию частицы. Далеко ли ушла частица от своего начального положения?

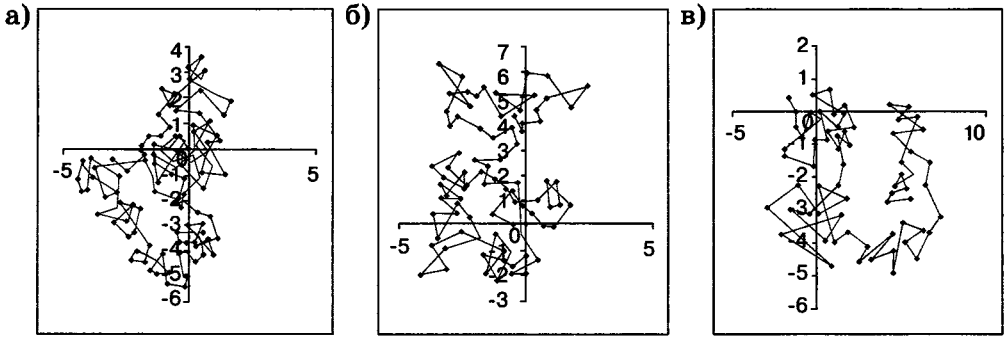


Рис. КП.4 Траектория частицы при $n = 100$

2 Повторите компьютерный эксперимент с тем же значением n.

3 А теперь проведите эксперимент для $n = 200, 300, 400, 500, 600, 700, 800, 900$.

4 Постройте график зависимости расстояния l, на которое ушла частица от начального положения, от n. Похож ли получившийся у вас график на график какой-либо знакомой вам функции?

▶ В наших экспериментах зависимость получилась похожей на функцию $l = a\sqrt{n}$. Коэффициент a мы не сообщаем специально. Если и у вас график оказался похожим на график такой функции, найдите этот коэффициент (методом наименьших квадратов или методом наименьших отклонений).

Перейдем к исследованию второй модели броуновского движения — случайной величиной является не только направление движения, но и скорость. Внесите нужные изменения в алгоритм и/или в заполнение электронной таблицы. ◀

5 Проведите компьютерный эксперимент с этой моделью для $n = 100$. На рисунке КП.4 приведены три такие траектории при указанном значении n. У вас траектории получились иные.

6 Проведите компьютерный эксперимент для $n = 200, 300, 400, 500, 600, 700, 800, 900$. Найдите и для этой модели зависимость расстояния l от n.

▶ Еще две траектории, полученные нами при $n = 500$, изображены на рисунке КП.5.

Но наблюдение за траекториями еще не дает значительной информации, хотя можно заметить, что каждая из траекторий на рисунке КП.4 умещается в квадрате со стороной 10. А для траекторий на рисунке КП.5 получается аналогичный квадрат со стороной 20. Но поскольку все направления равноправны, то более естественно говорить не о квадрате, а о круге, внутри которого располагается

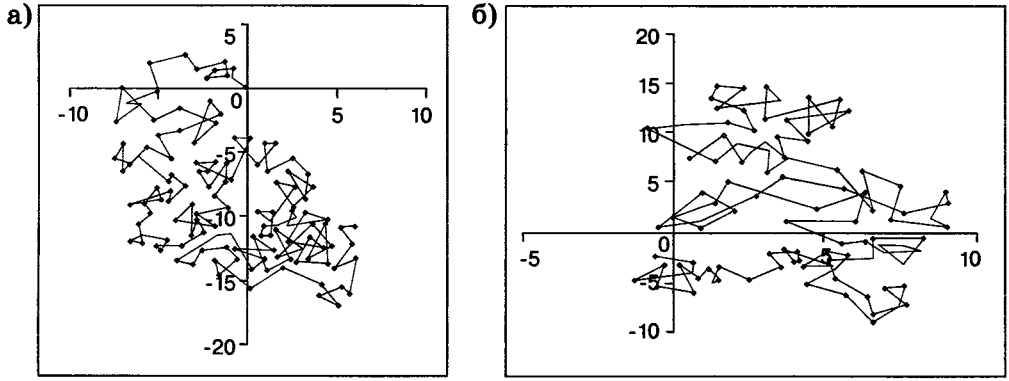


Рис. КП.5 Траектория частицы при $n = 500$

траектория. В мире случайного возможно, что частица вообще будет двигаться в каком-то одном направлении, но это маловероятно.

Попытаемся оценить величину радиуса круга, содержащего траекторию, в зависимости от числа шагов n. Иными словами, мы хотим решить следующую задачу: «*Имеется несколько точек на плоскости. Найдите круг наименьшего радиуса, содержащий все данные точки*».

Мы будем считать, что точки заданы своими координатами. Поэтому для решения задачи требуется найти координаты центра окружности, при которых радиус принимает наименьшее значение. В § 13 мы обсуждали похожую задачу для двух переменных, но там зависимость была линейной. Здесь эта зависимость более сложная, поэтому и метод решения намного сложнее. Его мы здесь излагать не будем, но у вас есть подходящий инструмент решения подобных задач — надстройка *Поиск решения* в электронной таблице Excel. Как применять этот инструмент, описано в лабораторной работе 6. ◀

7 Подготовьте заполнение электронной таблицы для расчета указанных величин. А теперь, проводя компьютерные эксперименты, вычислите в каждом из них отношение квадрата радиуса R наименьшего круга к числу шагов n. Впишите полученные результаты в таблицу.

n	100	200	300	400	500	600	700	800	900
R^2/n									

▶ Числа в клетках получились различные, но обратите внимание, что у многих в классе они близки к одному и тому же числу. Запишите открытую вами закономерность подходящей формулой. ◀

Лабораторная работа 18 (к § 27)

Вычисление площадей и объемов методом Монте-Карло. Моделирование случайных процессов

Познакомившись в § 27 с применением метода Монте-Карло для вычисления площади произвольной фигуры, некоторые из вас могли усомниться, можно ли с помощью вероятностной модели получить достаточно точный результат для такой вполне определенной величины, какой является площадь фигуры. На подобные вопросы отвечает специальный раздел математики — теория вероятностей. Математиками доказано, что метод Монте-Карло действительно применим для нахождения площадей фигур.

Конечно, применяя этот метод, вы получите значение площади лишь приближенно. Но при вычислении по тем или иным формулам (например, площади поверхности прямоугольного стола) вы все равно обязаны провести какие-то измерения, а они-то всегда имеют определенную погрешность. На величину этой погрешности влияет такое большое количество разнообразных и трудноучитываемых факторов, что можно считать ее случайной величиной. Так что от случайности в нашем мире никак не избавиться. А вот компьютерный эксперимент поможет вам убедиться, что метод Монте-Карло дает вполне хорошие результаты.

1 Первый эксперимент по нахождению площади круга проведем, используя алгоритм, записанный в объяснительном тексте § 27. По результатам эксперимента заполните таблицу.

Число точек, N	1000	5000	15 000	50 000	100 000
Площадь, S					

▶ Вы легко подсчитаете без эксперимента, что площадь круга единичного радиуса равна π ($\pi \approx 3{,}14159265358979$). Число, близкое к этому, должно было получиться в вашем эксперименте. При каком из указанных значений N совпали первые три знака после запятой?

Проводя эксперимент для $N = 100\,000$, вы, наверно, заставили компьютер снова 100 000 раз исполнить тело цикла. Но ведь у вас уже был готовый результат для первых 50 000 исполнений. Зачем же так непроизводительно тратить компьютерное время? ◀

2 Модифицируйте алгоритм (а затем и программу) так, чтобы можно было использовать уже имеющиеся результаты вычис-

лений для получения новых. В возможности такого постепенного наращивания результатов, в частности, состоит важное преимущество метода Монте-Карло перед другими вычислительными методами.

3 А теперь с помощью метода Монте-Карло найдите площади фигур, о которых говорится в задании 1 из § 26. Сообщим, что для первой фигуры результат должен быть близким к $\frac{1}{3}$, для второй — к $\frac{1}{4}$, а площадь третьей фигуры вы легко найдете сами.

▶ Метод Монте-Карло можно применить и для вычисления объема тела. Для этого тело надо поместить в куб или прямоугольный параллелепипед и выбрать подходящим образом систему координат. Теперь каждая точка, случайным образом бросаемая в этот параллелепипед, задается тремя своими координатами, т. е. вместо двух чисел нужно генерировать три случайных числа. ◀

4 Составьте математическую модель, алгоритм и программу для вычисления объема шара единичного радиуса.

▶ **Эллипсоидом** называется тело, полученное из шара растяжением вдоль координатных осей. Если исходный шар имел единичный радиус, а его центр находился в начале координат, то после растяжения вдоль оси Ox в a раз, вдоль оси Oy в b раз и вдоль оси Oz в c раз получается эллипсоид, точки которого удовлетворяют неравенству

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}\leqslant 1.$$

Такой эллипсоид полностью содержится в параллелепипеде, задаваемом неравенствами $-a \leqslant x \leqslant a$, $-b \leqslant y \leqslant b$ и $-c \leqslant z \leqslant c$. ◀

5 Найдите методом Монте-Карло объем этого параллелепипеда при $a = 2$, $b = 3$ и $c = 4$. Сравните полученное значение со значением объема, вычисляемого по формуле $V_{эл} = \frac{4}{3}\pi abc$.

6 Проведите расчеты для задачи 3 из § 27. Ваш ответ должен быть близок к 0,234375. В задаче 4 результат должен быть близок к $\frac{13}{24}$.

▶ Это не совпадает с вероятностью события, которое описывается поговоркой «Бабушка надвое сказала», т. е. с вероятностью 0,5, но довольно близко к этому числу. ◀

7 Напоследок желаем успеха в реализации проектов, предложенных вам в заданиях 5 и 6.

Лабораторная работа 19 (к § 32 и 33)

Компьютерное исследование логических формул

Составление таблицы истинности для заданной логической формулы — дело, весьма далекое от творчества. Так почему бы не поручить его компьютеру? Как всегда, это можно сделать и с помощью алгоритмического языка, и с помощью электронной таблицы. Сейчас мы предложим вам вариант использования Excel, а программу для нахождения таблицы истинности вы разработаете самостоятельно.

1 Рассмотрим для примера формулу $\overline{(A \vee B \,\&\, (A \vee \overline{C}))}$. Подготовим заполнение электронной таблицы по образцу таблицы КП.13.

Таблица КП.13

	A	B	C	D
1	A	B	C	$\overline{(A \vee B \,\&\, (A \vee \overline{C}))}$
2	ИСТИНА	ИСТИНА	ИСТИНА	=НЕ(A2;ИЛИ(B2;И(A2;ИЛИ(НЕ(C2)))))
3	ИСТИНА	ИСТИНА	ЛОЖЬ	=НЕ(A3;ИЛИ(B3;И(A3;ИЛИ(НЕ(C3)))))
4	ИСТИНА	ЛОЖЬ	ИСТИНА	...
5	ИСТИНА	ЛОЖЬ	ЛОЖЬ	...
6	ЛОЖЬ	ИСТИНА	ИСТИНА	...
7	ЛОЖЬ	ИСТИНА	ЛОЖЬ	...
8	ЛОЖЬ	ЛОЖЬ	ИСТИНА	...
9	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	...

Отметим, что в столбцах A, B и C вместо слов ИСТИНА и ЛОЖЬ можно писать 1 и 0 соответственно.

В столбце D вы получили значения рассматриваемого выражения при всевозможных наборах значений переменных A, B и C, т. е. на вашем экране расположилась искомая таблица истинности этой логической формулы.

2 А теперь в столбец E введите формулы, позволяющие находить для тех же наборов значений переменных A, B и C значение $\overline{A} \,\&\, \overline{B} \,\&\, C$.

Вы видите, что таблицы истинности для этих формул совпали. Значит, эти формулы равносильны!

3 Выясните аналогичным образом, равносильны ли логические формулы в следующих парах:

1) $\overline{A} \,\&\, B \,\&\, C \vee \overline{A} \,\&\, \overline{B} \,\&\, C \vee B \,\&\, C$ и $(\overline{A} \vee B) \,\&\, C$;

2) $(A \,\&\, B \to C) \,\&\, (B \,\&\, C \to A)$ и $A \,\&\, C \to B$;

3) $\overline{A \,\&\, B \vee \overline{C}}$ и $\overline{C \to A} \vee \overline{C \to B}$;

4) $(A \vee B) \,\&\, (A \,\&\, \overline{C}) \,\&\, (B \vee \overline{C})$ и $A \to C$.

4 А теперь с помощью электронной таблицы решите задачу, похожую на ту, которая разобрана в объяснительном тексте § 33.

З а д а ч а. В нарушении правил торговли подозреваются четыре работника магазина: Антонов, Борисов, Васильев и Денисов. Торговая инспекция выяснила:

1) если Антонов нарушал правила, то вместе с Борисовым;

2) если Борисов нарушал правила, то Васильев нарушал или Антонов не нарушал;

3) если Денисов не нарушал правила, то Антонов нарушал, а Васильев не нарушал.

Кто из подозреваемых наверняка нарушал правила торговли?

5 Обозначьте простые высказывания «Антонов нарушал правила», «Борисов нарушал правила», «Васильев нарушал правила» и «Денисов нарушал правила» логическими переменными и составьте высказывание, описывающее вскрытые торговой инспекцией факты.

6 Заполните электронную таблицу так, чтобы получилась таблица истинности для составленного вами высказывания. Поскольку логических переменных четыре, то эта таблица будет содержать 16 строк. С ее помощью ответьте на вопрос задачи.

7 Дополнительное расследование показало, что если Денисов нарушал правила, то и Борисов нарушал. Выясните, изобличает ли эта информация кого-то еще в нарушении правил и если да, то кого именно.

Лабораторная работа 20 (к § 37)

Соединение таблиц в Access

Выполнив задание 6 из § 34, вы подготовили две таблицы: одна из них содержит информацию о нагрузке учителей (ее атрибуты — Фамилия, Класс, Предмет), другая — о расписании (ее атрибуты — Класс, Предмет, День недели, Номер урока).

1 Создайте эти таблицы в рамках одной базы данных. Вы это делаете не в первый раз, так что надеемся, что проблем у вас не будет. Впрочем, один вопрос наверняка возник: при создании таблицы СУБД Access предлагает какой-либо из атрибутов объявить ключевым. У вас в таблицах вряд ли будет хотя бы один атрибут, который может претендовать на роль ключа: учитель может вести уроки в разных классах, класс бывает на уроках у разных учителей, в каждом классе изучается много предметов и т. д. Так что на предложение СУБД придется ответить отказом.

▶ Теперь давайте обсудим, какие запросы мы хотели бы иметь возможность формулировать нашей базе данных. Например: ведет ли учитель Иванова уроки во вторник? (У вас фамилия учителя может быть другой.) Но тогда нам нужно использовать информацию из обеих таблиц. Как бы поступил человек в этой ситуации? Он в первой таблице нашел бы фамилию «Иванова», посмотрел бы, какой предмет она ведет, затем во второй таблице нашел этот предмет в тех строках, где указан «вторник». Иными словами, он соединил бы таблицы по атрибуту Предмет. Вот и мы поступим так же. ◀

2 Выберите в меню *Сервис* пункт *Схема данных*. В открывшемся окне появился список таблиц, имеющихся в данной БД. Выделите атрибут Предмет в таблице *Нагрузка учителей* и нажатой левой клавишей мыши перемещайте его на такой же атрибут в таблице *Расписание*. После совмещения (об его успешности можно судить по изменению вида курсора мыши) отпустите клавишу мыши. Между атрибутами протянулась «веревочка»: связь установлена. Поскольку мы отказались объявить хотя бы один из атрибутов ключевым, то эта связь имеет тип **«многие-ко-многим»**, хотя никаких значков «∞» на концах «веревочки» не появилось.

3 А теперь создайте запрос (с помощью бланка QBE), посредством которого для данного учителя можно было бы выяснить, в каких классах и в какое время (т. е. указать номер урока) он работает в заданный день недели. Мы предлагаем вам действовать в режиме *Конструктора*. Первым шагом надо в окно бланка QBE занести ваши таблицы. Для этого щелкните в поле окна правой клавишей мыши и в появившемся контекстном меню выберите пункт *Добавить*. Введите обе ваши таблицы, они изобразятся с установленной вами связью атрибутов. А теперь сформируйте нужный запрос и сохраните его под каким-либо именем.

4 Запустите запрос на исполнение и посмотрите на результат. Все ли получилось, как вы задумали? Если да, то поменяйте в запросе фамилию учителя, день недели. Попытайтесь получить аналогичную информацию не для одного учителя, а сразу для нескольких.

▶ Неуверенность в успехе, которая явно чувствуется в предыдущем абзаце, вполне обоснованна. Вернемся к рассмотрению действий человека и еще раз проанализируем ситуацию. После того как будет найден нужный предмет в строке, где указан «вторник», вполне может оказаться, что это урок не в том классе, в котором работает данный учитель. Ведь, к примеру, физику в одном 10 классе может вести один учитель, а в параллельном классе — совсем другой учитель. Если задуматься о теоретических причинах этого явления, то станет ясно, что она кроется в самом определении произведения отношений, которое, как мы говорили, лежит в основе понятия соединения таблиц.

Может быть, тогда нам соединить таблицы по атрибуту Класс? Давайте попробуем. ◀

5 Снова вызовите режим *Схема данных* и сделайте изменение связи. Для этого проще всего навести курсор мыши на связующую атрибуты «веревочку» и щелкнуть правой кнопкой. В открывшейся таблице укажите, какие теперь атрибуты надо связать. Сохраните новую связь и начинайте эксперименты с запросами.

▶ Результат от сделанных изменений стал только хуже. Наш учитель теперь ведет вообще не свои предметы. Попытайтесь объяснить, почему так получилось.

Еще раз вернемся к действиям человека с этими двумя таблицами. Найдя во вторник нужный предмет, человек посмотрит, в каком классе этот урок проводится, и затем по первой таблице проверит, ведет ли данный учитель уроки в этом классе. Иными словами, связь между таблицами устанавливается не только по атрибуту Предмет, но и по атрибуту Класс. ◀

6 Свяжите в ваших таблицах атрибуты Предмет и Класс.

Вот теперь на все запросы будут получаться корректные ответы.

▶ Обдумывая пройденный путь проб и ошибок, нетрудно сделать вывод, что успех нам обеспечило то обстоятельство, что набор двух атрибутов Предмет и Класс является ключом для первой таблицы. Не случайно разработчики данной СУБД настойчиво советуют искать ключевой атрибут при создании таблицы — это убережет от многих неприятностей. Если же назначить ключевой атрибут не удастся, нужно очень внимательно проектировать соединение таблиц.

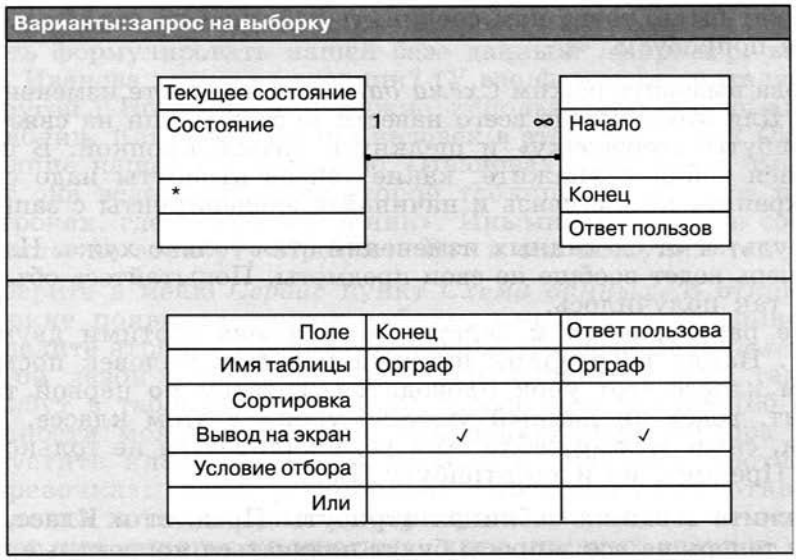
И напоследок сделаем еще одно замечание. Мы соединяли атрибуты с одинаковыми именами. Конечно, это необязательно: в разных таблицах одно и то же содержание может иметь разные имена. Как писал У. Шекспир: «Роза пахнет розой, хоть розой назови ее, хоть нет». И СУБД Access допускает соединение таблиц по атрибутам с разными именами, если только у этих атрибутов одинаковый тип. Если же вы попытаетесь связать текстовый атрибут с числовым, то СУБД укажет вам на ошибку. ◀

Лабораторная работа 21 (к § 39)

Создание экспертной системы с помощью Access

1 Прежде всего надо создать те таблицы, которые описаны в тексте § 39, при этом полезно объявить ключевым атрибут Состояние в таблицах *Имена*, *Вопросы* и *Текущее состояние*.

2 Создаем запрос *Варианты*, который позволяет выбрать те дуги орграфа, по которым пользователь может двигаться из текущего состояния к следующему. Но предварительно надо со-



Поле	Конец	Ответ пользова
Имя таблицы	Орграф	Орграф
Сортировка		
Вывод на экран	✓	✓
Условие отбора		
Или		

Рис. КП.6 Образец заполнения бланка запроса *Варианты*

единить таблицы *Текущее состояние* и *Орграф*, связав атрибуты Состояние и Начало.

3 Сформируйте запрос, руководствуясь рисунком КП.6.

4 Теперь нужно предложить пользователю выбрать одно из этих ребер и поместить *Конец* этого ребра в таблицу *Текущее состояние*. Все это мы реализуем с помощью формы с полем со списком. Эта форма будет иметь единственное поле, в котором и будет выпадать список доступных в данном состоянии ребер (рис. КП.7).

▶ Чтобы построить эту форму, надо вызвать *Конструктор* форм (рис. КП.8).

После щелчка по кнопке ОК появляется *Конструктор* форм и панель элементов. На рисунке КП.9 показано, как выглядит

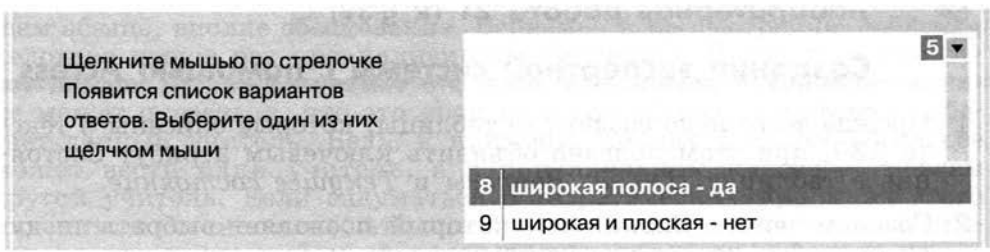


Рис. КП.7 Форма для выбора ответа пользователем

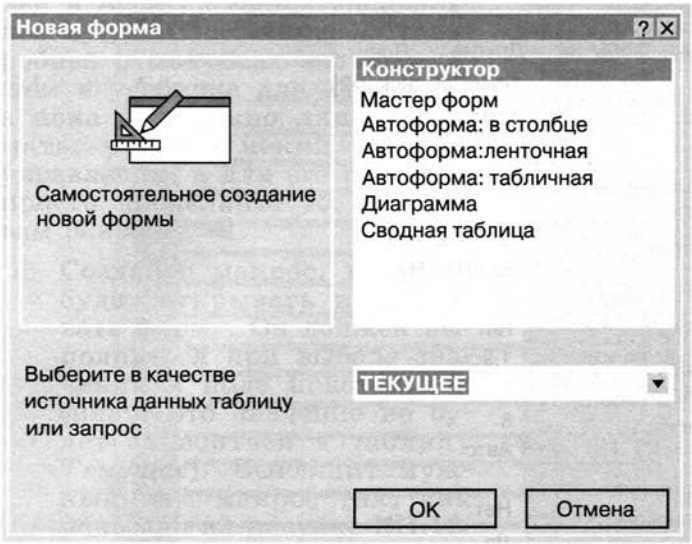


Рис. КП.8 Вид меню *Создать новую форму* с выбранной таблицей *Текущее состояние*



Рис. КП.9 Вид конструктора форм и панели инструментов

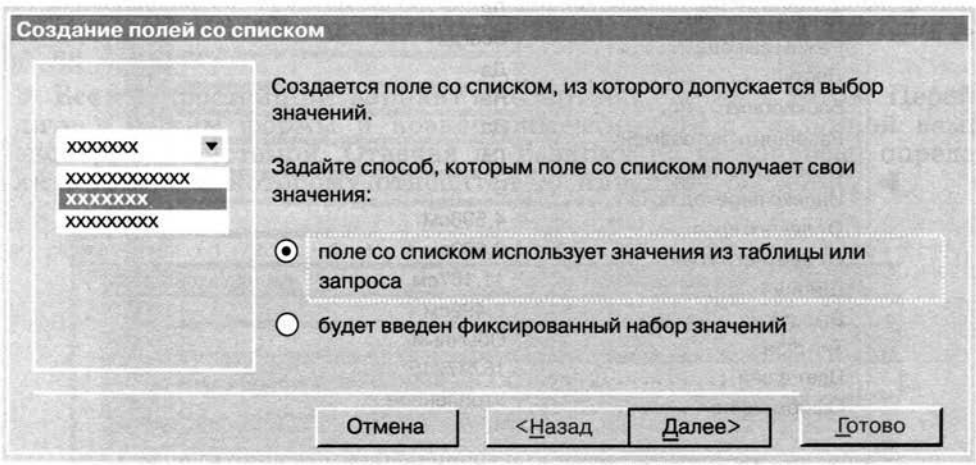


Рис. КП.10 Диалоговое окно при создании поля со списком

Конструктор **форм, панель инструментов и инструмент** ***Поле со списком****, которым мы дальше намерены воспользоваться.**

Расширим область данных, потянув ее мышью за границы. Выберем инструмент ***Поле со списком*** **и нарисуем им поле. После этого система предлагает диалог, от которого мы откажемся, нажав кнопку** ***Отмена*** **(рис. КП.10).**

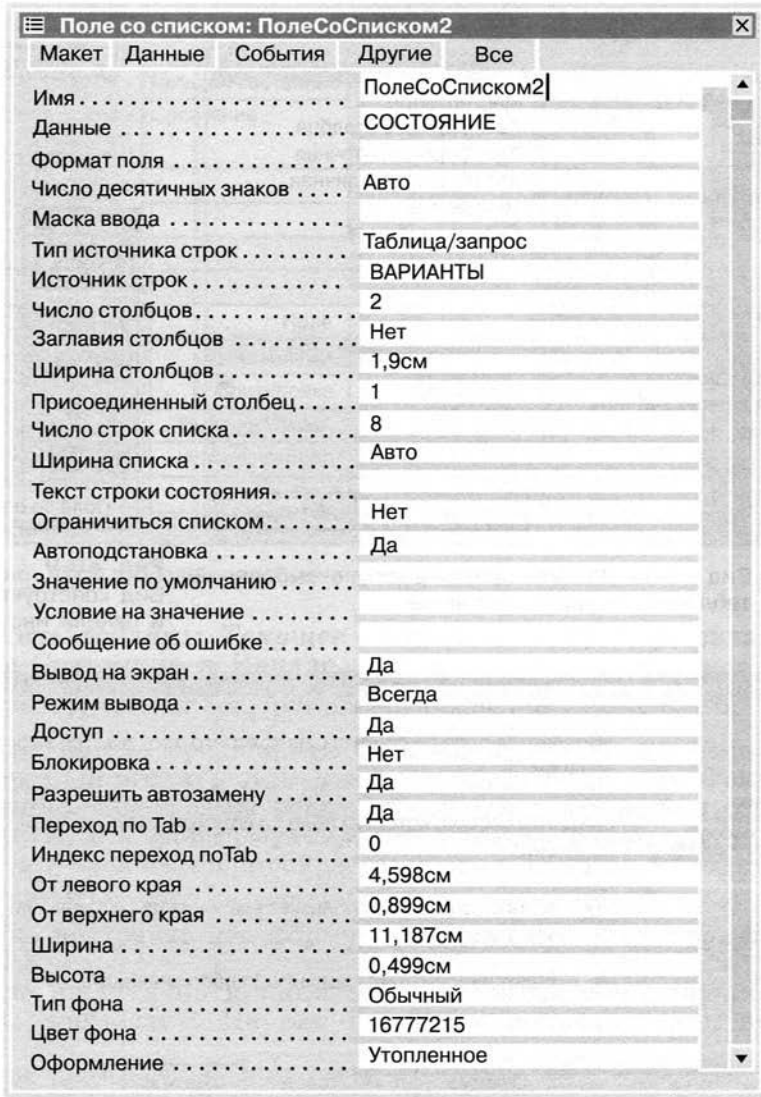


Рис. КП.11 Изменение свойств поля со списком

Щелчком мыши внутри поля выделим его.

Нарисовав поле со списком, из меню *Вид-Свойства* вызываем его свойства. В появившемся окне (рис. КП.11) присвоим свойствам следующие значения: Данные — *Состояние*, Источник строк — *Варианты*, Число столбцов — 2, Ширина столбцов — 1,9.

Но «вручную» открывать и закрывать эту форму весьма неудобно. Надо, чтобы это выполнялось автоматически. Для этих це-

лей в Access имеется специальное средство — макросы (подробнее о макросах мы расскажем в учебнике для 11 класса, а пока вам важно лишь понимать, что это некий элемент управления, а для его создания просто проделайте то, что мы вам скажем). ◀

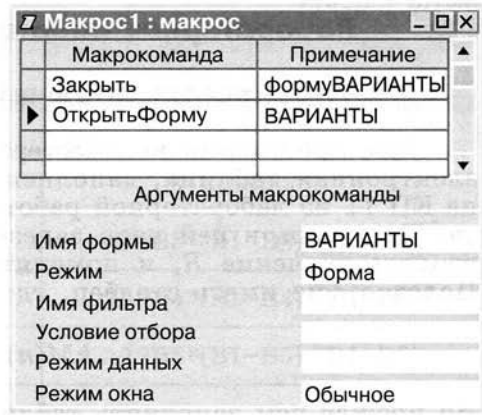


Рис. КП.12 Окно макроса, открывающего и закрывающего форму *Варианты*

5 Создайте макрос, который будет открывать и закрывать форму. Он должен выполняться при выборе значения в поле подстановок (иначе это значение не будет заноситься в таблицу *Текущее*). Выглядит нужный нам макрос так, как показано на рисунке КП.12.

6 Теперь этот макрос нужно связать с обновлением поля со списком формы *Варианты*. Для этого нужно снова выделить в режиме *Конструктора* поле со списком, открыть его свойства и приписать событию *После обновления* значение «Макрос1» (рис. КП.13). Теперь после обновления поля будет выполняться *Макрос1*.

▶ Все — простейший вариант экспертной системы готов. Перейдите в режим формы и поэкспериментируйте с созданной вами экспертной системой. Отвечая по-разному на ее вопросы, определите класс, к которому относится то или иное растение. ◀



Рис. КП.13 Привязка макроса к обновлению поля со списком

Лабораторная работа 22 (к § 44)

Управление добычей возобновляемых ресурсов

Для проведения компьютерного эксперимента вам потребуется электронная таблица, заполненная примерно так же, как таблица КП.11 из лабораторной работы 12 по модели ограниченного роста. Но теперь в ней надо зарезервировать клетку, куда мы будем вводить значение R, и поменять формулу для вычисления $M(n)$. Полезно еще иметь столбец, где указывается ежегодный прирост:

$$kM(n)\frac{L - M(n)}{L - M(0)}.$$

Но прежде чем заполнять таблицу, уточним постановку задачи.

Моря и океаны — это, конечно, основной источник для рыбного промысла. Но мы ведем речь об управлении, а управлять рыбными ресурсами морей и океанов никто еще не научился. Да и модели, используемые для описания гидросферы Земли, относятся к так называемым глобальным моделям, о которых у нас пойдет речь лишь в § 49. Поэтому будем строить управление добычей рыбы в неком рыбоводческом хозяйстве. Оно имеет водоем, и в некоторый момент времени (от которого и будет вестись отсчет) в него запустили рыбу. Десять лет она там росла и размножалась без всякого отлова (кроме небольших контрольных отловов для наблюдения за состоянием рыбы). Затем решили начать ее промышленную добычу. Оптимальный размер этой добычи нам и предстоит определить.

1 Заполните электронную таблицу по образцу таблицы КП.13. В ней в клетках B3—B13 вычисляется масса рыбы в те годы, когда отлова не было, а затем производится ежегодный отлов в количестве R тонн. В угловых скобках записано обозначение величины, которая должна быть занесена в данную клетку в качестве начального значения.

2 Теперь определим исходные данные. Пусть $L = 10\,000$ т, $M(0) = 1$ т, $k = 2$. Пусть принято решение ежегодно отлавливать 1000 т рыбы.

3 Выполните соответствующие вычисления в электронной таблице.

4 Постройте графики изменения массы и прироста рыбы по годам.

5 Обратите внимание: со временем величина прироста совпадает с забираемой массой! Случайность? Проверим: попробуем вылавливать 3000 т.

Выполните соответствующие вычисления в электронной таблице. Постройте графики изменения массы и прироста рыбы по годам. Тот же эффект!

Таблица КП.13

	A	B	C
1	<*k*>	<*L*>	<*R*>
2	Год	Масса рыбы	Прирост
3	0	<M(0)>	A1*B3*(B1–B3)/(B1–B3)
4	A3+1	B3+C3	A1*B4*(B1–B4)/(B1–B3)
5	A4+1	B4+C4	A1*B5*(B1–B5)/(B1–B3)
6	A5+1	B5+C5	...
...	...	...	...
13	A12+1	B12+C12	...
14	A13+1	B13+C13–C1	...
15	A14+1	B14+C14–C1	...
...	...	...	...

6 Поэкспериментируйте еще с несколькими значениями R. Но не жадничайте — не берите R больше чем 6000 т.

Фактически вы сейчас «открыли» важное свойство живой природы — явление *саморегуляции*. Природа стремится восстановить то стабильное состояние, из которого ее вывели тем или иным способом.

7 А теперь попробуйте изъять 6000 т.

Выполните соответствующие вычисления в электронной таблице. Постройте графики изменения массы и прироста рыбы по годам.

Видите: через 6 лет масса рыбы вдруг стала отрицательной. Реально такого быть не может. Просто это означает, что от вашего хозяйствования рыба погибла.

Значит, 6000 т — это слишком много. Ну а какую же наибольшую массу все-таки можно отлавливать?

8 Организуйте и проведите соответствующий компьютерный эксперимент.

▶ Вот мы и выяснили, каким должно быть здесь оптимальное управленческое решение. ◀

Лабораторная работа 23 (к § 45)

Организация посещений парка

1 Прежде всего надо определить коэффициенты b и c в формуле для A. Заполните электронную таблицу так, как указано в объяснительном тексте § 44, и найдите значения b и c.

2 Теперь электронную таблицу заполните так, чтобы можно было находить K по N и N по K (о том, как нужно заполнить таблицу, вы размышляли, выполняя задание 1 к § 45).

3 Найдите максимально допустимое количество посещающих, если парк будет работать пять дней в неделю.

4 Будем считать, что двух дней подряд хватает для полного восстановления парка. В каком режиме должен работать парк, чтобы его могли посещать 150 человек ежедневно?

Лабораторная работа 24 (к § 47)

Лисы и кролики

1 Подготовим заполнение электронной таблицы для проведения запланированного нами компьютерного эксперимента так, как показано в таблице КП.14.

Таблица КП.14

	A	B	C	D
1	*k*	Год	Масса кроликов	Масса лис
2	<*k*>	0	<M(0)>	<L(0)>
3	*c*	B2+1	(1+A2–A4*D2)*C2	(1+A6*(C3–C2))*D2
4	<*c*>	B3+1	(1+A2–A4*D3)*C3	(1+A6*(C4–C3))*D3
	s	…	…	…
	<*s*>	…	…	…
		…	…	…
		…	…	…
51		B50+1	(1+A2–A4*D50)*C50	(1+A6*(C51–C50))*D50

▶ Как обычно, в угловых скобках записаны обозначения величин, которые должны быть занесены в качестве начальных данных. ◀

2 Заполните электронную таблицу указанным образом. Проведите компьютерный эксперимент по исходным данным, указанным в таблице КП.15.

3 Используя компьютер, постройте графики $M(n)$ и $L(n)$ как функций от n.

Таблица КП.15

k	c	s	$M(0)$	$L(0)$
3,7	0,009	0,0001	5000	460
2,8	0,006	0,0002	5000	210

Литература для дополнительного чтения

Андреева Е. Математические основы информатики. Элективный курс: учеб. пособие / Е. Андреева, Л. Босова, И. Фалина. — М.: БИНОМ, 2005.

Бекаревич Ю. Самоучитель Access 2000 / Ю. Бекаревич, Н. Пушкина. — М.: BHV, 2001.

Вирт Н. Алгоритмы + структуры данных = программы. — М.: Мир, 1985.

Вирт Н. Систематическое программирование. Введение. — М.: Мир, 1977.

Гончаров А. Microsoft Access 7.0 в примерах. — СПб.: Питер, 1997.

Грабер М. Введение в SQL. — М.: Лори, 1997.

Задачник-практикум по информатике и информационным технологиям: кн. для учащихся 7—11 кл. общеобразоват. учреждений / А. Г. Гейн, Н. А. Юнерман. — М.: Просвещение, 2003.

Информатика: задачник-практикум. В 2 т. / под ред. И. Г. Семакина, Е. К. Хеннера. — М.: ЛБЗ, 1999. — Т. 1, 2.

Информатика: энциклопедический словарь для начинающих / под ред. Д. А. Поспелова. — М.: Педагогика-Пресс, 1994.

Информатика как наука об информации / под ред. Р. С. Гиляревского. — М.: Фаир-Пресс, 2006.

Информационная культура: учеб. пособие для ср. шк. 10—11 кл. / Т. А. Матвеева и др. — Смоленск: Ассоциация XXI век, 2006.

Касаткин В. Н. Введение в кибернетику. — Киев: Радянська школа, 1976.

Клоксин У. Программирование на языке Пролог / У. Клоксин, К. Мелиш. — М.: Мир, 1987.

Кольман Э. Занимательная логика / Э. Кольман, О. Зих. — М.: Наука, 1966.

Ульман Дж. Д. Введение в системы баз данных / Дж. Д. Ульман, Дж. Уидом. — М.: Лори, 1999.

Шафрин Ю. А. Информационные технологии. — М.: ЛБЗ, 1999.

Предметный указатель

А

автомат 36
адекватность модели 23
адрес ячейки 218
адресация абсолютная 219
— относительная 219
алгоритм 28
— вспомогательный 74
— рекурсивный 82
алфавит 13
— внешний (машины Тьюринга) 42
— внутренний (машины Тьюринга) 42
анализ информации 57
аргумент вспомогательного алгоритма 75
— отношения 162
арность отношения 162
атрибут ключевой 165
— объекта 61
— обязательно присутствующий 64
— отношения 162

Б

база данных 60
— — иерархическая 62
— — реляционная 63
— — сетевая 62
— знаний 183
байт 16
банк данных 61
бит 15
бланк QBE 228
блок логического вывода 176
блок ячеек 219

В

вероятность 118
вершина графа 26
видеоинформация 7
высказывание 147
высказывания равносильные 152
— тождественно равные 152

Г

гарнитура 222
гигабайт 16
гомеостаз 207
грамматика языка 13
граф 26
— нагруженный 26
— ориентированный 26

Д

датчик случайных чисел 121
детерминированность алгоритма 32
диапазон изменения индексов 84
дискретизация 19
— звука 18
дискретность алгоритма 31
ДСЧ 121
дуга орграфа 26

З

запрос 60
— на выборку 227
знания 58
значение переменной 29

И

имя вспомогательного алгоритма 74
— массива 84
— отношения 162
— переменной 29
индекс элемента массива 84
информация декларативная 58
— символьная 7
исполнитель 10
— универсальный 40
испытание 118
источник информации 8

К

канал связи 8
квантор всеобщности 168
— существования 167
кегль 222
кибернетика 199
килобайт 16
когнитолог 177

конечность алгоритма 32
контекст 12

Л
логика 145
— математическая 145
— формальная 145

М
массив 84
— двумерный 84
— линейный 84
— одномерный 84
массовость алгоритма 33
машина Тьюринга 40
мегабайт 16
метод Монте-Карло 129
— наименьших квадратов 69
— половинного деления 87
— пошаговой детализации 73
моделирование 21
модель 21
— вероятностная 119
— глобальная 214
— информационная 23
— компьютерная 23, 108
— математическая 23
— реляционная 159
— фактографическая 61

Н
носитель информации 7

О
обработка информации формальная 10
— — эвристическая 9
объект информационный 7
объявление переменной 31
оператор присваивания 29
орграф 26
отладка 28
отношение 161
— бинарное 162
— тринарное 162
— функциональное 165
ошибка семантическая 29
— синтаксическая 28

П
параметры модели 21
— фактические 74
— формальные 74
переменная 29
— логическая 152
— свободная 167
— связанная 167
подпрограмма 74
подпрограмма-функция 74
понятность алгоритма 32
последовательность, равномерно распределенная 120
— псевдослучайная 122
правило вывода 145
— рекурсивное 191
предикат 167
приближение равномерное 230
приемник информации 8, 91
признак объекта 61
произведение отношений 174
программа 73
— логическая 183
программирование логическое 183
процедурный стиль программирования 183
процесс информационный 8

Р
размерность массива 84
ребро графа 26
результат вспомогательного алгоритма 74
результативность алгоритма 32
рекуррентное соотношение 81
рекурсия 82
решатель 176
робот промышленный 36

С
саморегуляция системы 263
связь между атрибутами 167
— по типу многие-ко-многим 256
— — — один-к-одному 174
— — — один-ко-многим 174
— обратная 206
— — положительная 207
— — отрицательная 206
СДНФ 156
семантика языка 13
серия испытаний 118
сеть 26
— семантическая 26

синтаксис языка 13
синтез информации 58
система 25
— динамическая 201
— информационная 61
— искусственного интеллекта 45
— управления базой данных 60
— экспертная 45
системный подход 25
— эффект 24
слово 12
совершенная дизъюнктивная нормальная форма 156
соединение таблиц 174
сортировка 64
структура древовидная 62
СУБД 60
схема функциональная 43
счетчик цикла 31

Т

тавтология 154
терабайт 16
тип булев 31
— вещественный 31
— логический 31
— массива 84
— переменной 31
— символьный 31
— целый 31
тождественно истинная формула 154
тождественное преобразование 152
тождество 152

У

управление 199
— по принципу обратной связи 211
утверждение целевое 183

Ф

фильтр 173
фильтрация 173
флажок 229
формализация 13
формула логическая 152
формулы равносильные 152
— тождественно равные 152
функция голосования 152
— целевая 230

Ц

цель моделирования 21

Ч

частота 118
— относительная 118

Ш

шрифт 222
— наклонный 222
— подчеркнутый 222
— полужирный 222

Э

эксперт 175
электронно-вычислительная машина 7
эллипсоид 253
эмерджентность 24
энтропия 134

Я

язык 12
— коммуникативный 12
— распознаваемый 38
— — автоматом 38
— формализованный 13
— формальный 28
ячейка электронной таблицы 218

UNICODE 16

КЛЮЧИ К ТЕСТОВЫМ ЗАДАНИЯМ

Глава 1

Задание	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
Ответ	3	1	3	2	3	3	2	3	3	4	1

Задание	A12	A13	A14	A15	A16	A17	A18	A19	A20
Ответ	3	4	4	2	4	2	4	3	4

Задание	B1	B2	B3	B4	B5	B6	B7
Ответ	моделированием	27	циклом	144	24	присваивания	символьный

Глава 2

Задание	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
Ответ	3	2	2	3	1	1	3	2	3	3	3

Задание	B1	B2	B3
Ответ	3	25	8

Глава 3

Задание	A1	A2	A3	A4	A5	A6	A7	B1	B2
Ответ	2	2	3	2	2	3	3	315467	123

Задание	B3	B4	B5	B6
Ответ	1АГД, 2ББН, 3БЕЖ, 4ДЗИЛМ, 5ДКЛМ, 6АВГД	135	псевдослучайной	дискретизацию

Глава 4

Задание	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
Ответ	4	2	2	3	3	2	1	1	2	2	4	2

Задание	B1	B2	B3	B4
Ответ	ключевым	4	истинно	146

Глава 5

Задание	A1	A2	A3	A4	B1	B2	B3
Ответ	2	2	3	3	положительной	управлением	1346

Оглавление

Уважаемые старшеклассники! 3

Глава 1. Информатика как наука 4

§ 1. Информация 5
§ 2. Информационные процессы 8
§ 3. Язык как средство сохранения и передачи информации 12
§ 4. Универсальность двоичного кодирования 15
§ 5. Информационное моделирование 21
§ 6. Системный подход в моделировании 24
§ 7. Алгоритмы и их свойства 28
§ 8. Формальный исполнитель: автомат 36
§ 9. Универсальный исполнитель 39
§ 10. Основные направления информатики 44

Итоги главы 1 47

Проверь себя 49

Глава 2. Информационная деятельность человека и использование в ней компьютерных технологий 56

§ 11. Информационные задачи и этапы их решения 57
§ 12. Применение компьютера для решения простейших информационных задач 60
§ 13. Эксперимент как способ познания. Компьютерная обработка результатов эксперимента 66
§ 14. Алгоритм как форма организации процедурной информации 72
§ 15. Рекуррентные соотношения и рекурсивные алгоритмы 80
§ 16. От переменной к массиву 84
§ 17. Решение уравнений методом половинного деления 86
§ 18. Измерение количества информации 89

Итоги главы 2 96

Проверь себя 97

Глава 3. Моделирование процессов живой и неживой природы 102

§ 19. Моделирование физических процессов ..... —
§ 20. Компьютерное исследование модели движения в среде с сопротивлением 105
§ 21. Моделирование процессов в биологии 109
§ 22. Границы адекватности модели 112
§ 23. Моделирование эпидемии гриппа 116
§ 24. Вероятностные модели 117
§ 25. Датчики случайных чисел и псевдослучайные последовательности 120
§ 26. Моделирование случайных процессов 123

§ 27. Метод Монте-Карло ... 129
§ 28. Еще раз об измерении количества информации ... 132
Итоги главы 3 ... 138
Проверь себя ... 140

Глава 4. Логико-математические модели ... 144
§ 29. Понятие моделей искусственного интеллекта ... 145
§ 30. Элементы логики высказываний ... 146
§ 31. Законы алгебры высказываний ... 151
§ 32. Как построить логическую формулу ... 155
§ 33. Решение логических задач средствами математической логики 157
§ 34. Реляционные модели ... 159
§ 35. Функциональные отношения ... 164
§ 36. Логические функции и логические выражения ... 167
§ 37. Логика СУБД Access ... 171
§ 38. Базы знаний и экспертные системы ... 175
§ 39. Реляционная модель экспертной системы ... 177
§ 40. Знакомимся с логическим программированием ... 183
§ 41. Запросы в базе знаний на Прологе ... 187
§ 42. Встроенные предикаты в логических языках программирования. Простейшие программы ... 189
Итоги главы 4 ... 192
Проверь себя ... 195

Глава 5. Информационные модели в задачах управления ... 199
§ 43. Что такое управление ... —
§ 44. Сколько можно взять у природы ... 202
§ 45. Задача о лесопарке ... 203
§ 46. Учимся у природы правильной организации управления ... 206
§ 47. Изучаем системы с обратной связью ... 209
§ 48. Управление по принципу обратной связи ... 211
§ 49. Глобальные модели ... 214
Итоги главы 5 ... —
Проверь себя ... 215

Компьютерный практикум ... 217
Лабораторная работа 1 (к § 5). Обработка числовой информации с помощью электронной таблицы ... 218
Лабораторная работа 2 (к § 6). Обработка текстовой и графической информации ... 222
Лабораторная работа 3 (к § 7). Программирование основных алгоритмических конструкций ... 224
Лабораторная работа 4 (к § 12). Фактографическая модель «Класс» ... 225

Лабораторная работа 5 (к § 12).
Поиск информации в базе данных 227
Лабораторная работа 6 (к § 13).
Компьютерная обработка экспериментальных данных 229
Лабораторная работа 7 (к § 14).
Метод пошаговой детализации 231
Лабораторная работа 8 (к § 15).
Рекуррентные соотношения и рекурсивные алгоритмы 232
Лабораторная работа 9 (к § 16).
Программы для обработки массивов ..... —
Лабораторная работа 10 (к § 17).
Решение уравнений 234
Лабораторная работа 11 (к § 20).
Модель движения в среде с сопротивлением 235
Лабораторная работа 12 (к § 21).
Модели неограниченного и ограниченного роста 239
Лабораторная работа 13 (к § 22).
Поиск границ адекватности модели 242
Лабораторная работа 14 (к § 23).
Компьютерная модель эпидемии гриппа 244
Лабораторная работа 15 (к § 25).
Проверяем датчик случайных чисел 245
Лабораторная работа 16 (к § 26).
Компьютерная модель системы массового обслуживания 248
Лабораторная работа 17 (к § 26).
Моделирование броуновского движения ..... —
Лабораторная работа 18 (к § 27).
Вычисление площадей и объемов методом Монте-Карло. Моделирование случайных процессов 252
Лабораторная работа 19 (к § 32 и 33).
Компьютерное исследование логических формул 254
Лабораторная работа 20 (к § 37).
Соединение таблиц в Access 255
Лабораторная работа 21 (к § 39).
Создание экспертной системы с помощью Access 257
Лабораторная работа 22 (к § 44).
Управление добычей возобновляемых ресурсов 262
Лабораторная работа 23 (к § 45).
Организация посещений парка 264
Лабораторная работа 24 (к § 47).
Лисы и кролики ..... —
Литература для дополнительного чтения 265
Предметный указатель 266
Ключи к тестовым заданиям 269

Учебное издание

Гейн Александр Георгиевич
Ливчак Александр Борисович
Сенокосов Александр Иванович
Юнерман Нина Ароновна

ИНФОРМАТИКА И ИКТ

Учебник для 10 класса общеобразовательных учреждений

Базовый и профильный уровни

Зав. редакцией *Т. А. Бурмистрова*
Редактор *О. В. Платонова*
Младшие редакторы *Е. А. Андреенкова, С. В. Дубова*
Художник *О. П. Богомолова*
Художественный редактор *О. П. Богомолова*
Компьютерная верстка и техническое редактирование: *О. Ю. Мызникова, Н. В. Кондратьева*
Компьютерная обработка рисунков: *Г. М. Дмитриев*
Корректоры *Е. В. Барановская, О. Н. Леонова, Г. Н. Смирнова, И. В. Чернова*

Налоговая льгота — Общероссийский классификатор продукции ОК 005-93—953000. Изд. лиц. Серия ИД № 05824 от 12.09.01. Подписано в печать 05.03.12. Формат $70\times90^{1}/_{16}$. Бумага офсетная. Гарнитура SchoolBookCSanPin. Печать офсетная. Уч.-изд. л. 18,06 + 0,50 форз. Тираж 5000 экз. Заказ № 31075 (II-Гз).

Открытое акционерное общество «Издательство «Просвещение». 127521, Москва, 3-й проезд Марьиной рощи, 41.

Отпечатано в полном соответствии с качеством предоставленных материалов в ОАО «Смоленский полиграфический комбинат». 214020, г. Смоленск, ул. Смольянинова, 1.

УДК 373.167.1:004
ББК 32.97я72
И74

На учебник получены положительные заключения Российской академии наук (№ 10106-5215/15 от 31.10.07) и Российской академии образования (№ 01-364/5/7д от 22.10.07)

Авторы:

А. Г. Гейн, А. Б. Ливчак, А. И. Сенокосов, Н. А. Юнерман

И74 **Информатика и ИКТ** : учеб. для 10 кл. общеобразоват. учреждений : базовый и профил. уровни / А. Г. Гейн, А. Б. Ливчак, А. И. Сенокосов, Н. А. Юнерман. — 2-е изд. — М. : Просвещение, 2012. — 272 с. : ил. — ISBN 978-5-09-028705-0.

Учебник полностью охватывает материал, предназначенный как для базового, так и для профильного уровня обучения. В курсе основное внимание на базовом уровне преподавания информатики уделяется расширенному освоению информационных технологий для применения их к решению разнообразных жизненных задач. Это делает предлагаемый курс привлекательным для всех учащихся независимо от того, выбрали они гуманитарное или естественно-научное направление своего обучения. Материал, предназначенный для изучения на профильном уровне, содержит более глубокое изложение основ теоретической информатики и нацелен на подготовку к ЕГЭ.

УДК 373.167.1:004
ББК 32.97я72

ISBN 978-5-09-028705-0

© Издательство «Просвещение», 2008
© Художественное оформление.
Издательство «Просвещение», 2008
Все права защищены

Язык записи алгоритмов	Pascal
1. Заголовок программы	
Алгоритм {...}	**program** <имя>**;**
2. Описание переменных	
	var *(*Обязательно перед описанием переменных*)*
вещественные переменные	
вещ: <список переменных>;	<список переменных>**: real;**
целочисленные переменные	
цел: <список переменных>;	<список переменных>**: integer;**
символьные переменные	
сим: <список переменных>;	<список переменных>**: string;**
массивы	
вещ: <имя массива> [1:100; 1:100];	<имя массива>**: array** [1..100, 1..100] **of real;**
3. Оператор условия	

Если <условие> **то**
{<условие>;
...}
иначе {<оператор>;...}
	if <условие> **then**
	begin
	<оператор>**;** ...
	end
	else
	begin
	<оператор>**;** ...
	end;

Конструкция **иначе** (**else**) может отсутствовать	
4. Оператор цикла по условию	

Делать пока <условие>
{<оператор>;...}
	while <условие> **do**
	begin
	<оператор>**;** ...
	end;

Язык записи алгоритмов	Pascal
5. Оператор цикла со счетчиком	

Делать от <переменная> := <выражение>
до <выражение> **с шагом** <выражение>
{ <оператор>;... }
	for <переменная>**: =** <выражение>
	to <выражение> **do**
	begin
	<оператор>**;** ...
	end;

	Если нужен шаг –1, то вместо **to** пишут **downto**
6. Операторы ввода и вывода информации	
Запросить <имя переменной>;	**readln (**<список переменных>**);**
Сообщить <имя переменной или/и сообщение в кавычках>;	**writen (**'<сообщение>',<список переменных>**);**
7. Подпрограммы и функции	

Подпрограмма <имя подпрограммы>;
(**арг:** <список параметров>,
рез: < список параметров>)
{<оператор>; ...}
	Procedure <имя> **(**<параметры>**:** <тип>**;**
	var <результаты>**:** <тип>**);**
	begin
	<оператор>**;** ...
	end;

Функция <имя>: (**арг:** <типы аргументов>:
<список переменных>): <тип>
{ <оператор>;
...
знач := <результат>;
}
	function <имя> **(**<параметры>**:** <тип>**):**
	<тип>**;**
	begin
	<оператор>**;** ...
	<имя функции> **:=** <результат>**;**
	end;

8. Обращение к подпрограмме	
Вызвать <имя подпрограммы> (**арг**: <список параметров>; **рез**: <список параметров>);	<Имя> **(**<список параметров>**);**
9. Обращение к функции	
<переменная> := <Имя функции> (<список фактических параметров>);	<переменная> **:=** <Имя функции> **(**<список фактических параметров>**);**

Язык записи алгоритмов	QBasic
1. Заголовок программы	
Алгоритм {...}	Заголовка программы нет
2. Описание переменных	
вещественные переменные	
вещ: <список переменных>;	Если описание переменной отсутствует, то она вещественная
целочисленные переменные	
цел: <список переменных>;	Знак **%** после имени переменной или **DEFINT** <список переменных>
символьные переменные	
сим: <список переменных>;	знак **$** после имени переменной или DEFSTR <список переменных>
массивы	
вещ: <имя массива> [1:100; 1:100];	**DIM** <имя массива> **(1 to** 100, 1 **to** 100**)**
3. Оператор условия	

Если (<условие>) **то**
{<условие>;...}
иначе {<оператор>;...}
	IF (<условие>**) THEN**
	<оператор>
	...
	ELSE
	<оператор>
	...
	ENDIF

Конструкция **иначе** (**else**) может отсутствовать	
4. Оператор цикла по условию	
Делать пока (<условие>) {<оператор>;...}	**WHILE (**<условие>**)** <оператор> <оператор> ... **WEND**

Язык записи алгоритмов	QBasic
5. Оператор цикла со счетчиком	
Делать от <переменная> := <выражение> **до** <переменная> **с шагом** <выражение> { <оператор>;... }	**FOR** <переменная> = <выражение> **TO** <выражение> **STEP** <выражение> <оператор> ... **NEXT** <переменная>
6. Операторы ввода и вывода информации	
Запросить <имя переменной>;	**INPUT** " <уточняющее сообщение>" <список переменных через запятую>
Сообщить <имя переменной или/и сообщение в кавычках>;	**PRINT** " <сообщение>",<список переменных через «,» или «;» >
7. Подпрограммы и функции	
Подпрограмма <имя подпрограммы>; (**арг:** <список параметров>, **рез:** <список параметров>) {<оператор>; ...}	**SUB** <имя> (<параметры>) <оператор> **ENDSUB**
Функция <имя>: (**арг:** <типы аргументов>: <список переменных>): <тип> { <действие>; ... **знач** := <результат>; }	**FUNCTION** <имя> (<параметры>) <оператор> ... <имя>= <результат> **END FUNCTION**
8. Обращение к подпрограмме	
Вызвать <имя подпрограммы>; (**арг**: <список параметров>; **рез** <список параметров>);	**CALL** <имя> (<список параметров>)
9. Обращение к функции	
<переменная> := <Имя функции> (<список фактических параметров>);	<переменная> := <Имя функции> (<список фактических параметров>)

ИНФОРМАТИКА и ИКТ



БАЗОВЫЙ • ПРОФИЛЬНЫЙ УРОВНИ

Учебно-методический комплект по информатике авторского коллектива под руководством А. Г. Гейна содержит:

- А. Г. Гейн, А. Б. Ливчак, А. И. Сенокосов, Н. А. Юнерман
  Информатика и ИКТ. 10 класс
- А. Г. Гейн, А. И. Сенокосов
  Информатика и ИКТ. 11 класс
- А. Г. Гейн, А. А. Гейн
  Тематические тесты для 10 и 11 классов
- А. Г. Гейн
  Задачник-практикум. 10 —11 классы
- А. Г. Гейн
  Книги для учителя. 10 и 11 классы
- А. Г. Гейн
  Информатика. Рабочие программы. 10 —11 классы

Программы для проведения практических занятий и методические рекомендации размещены на сайтах:

http://kadm.math.usu.ru (*на страничке А. Г. Гейна*) и http://www.prosv.ru

ISBN 978-5-09-028705-0



ПРОСВЕЩЕНИЕ
ИЗДАТЕЛЬСТВО