А. Г. ГЕЙН

ИНФОРМАТИКА

10 КЛАСС

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

Учебное пособие для общеобразовательных организаций

2-е издание

Москва
«Просвещение»
2017

УДК 372.8:004
ББК 74.263.2
Г29

Гейн А. Г.

Г29 Информатика. Методические рекомендации. 10 класс : учеб. пособие для общеобразоват. организаций / А. Г. Гейн. — 2-е изд. — М. : Просвещение, 2017. — 162 с. : ил. — ISBN 978-5-09-044798-0.

Книга предназначена для учителей, работающих по учебнику «Информатика. 10 класс» авторов А. Г. Гейна, А. Б. Ливчака, А. И. Сенокосова, Н. А. Юнерман. Она полностью соответствует учебнику по структуре и содержанию, в нее включены подробные методические комментарии к теоретическому материалу и практическим заданиям.

Для учителей информатики.

УДК 372.8:004
ББК 74.263.2

ISBN 978-5-09-044798-0

ВВЕДЕНИЕ

Это пособие предназначено для учителей, избравших учебник «Информатика. 10 класс», созданный авторским коллективом в составе А. Г. Гейна, А. Б. Ливчака, А. И. Сенокосова и Н. А. Юнерман. В пособии даны рекомендации по изучению курса в соответствии с требованиями Федерального государственного образовательного стандарта по информатике, утвержденного Министерством образования РФ, и научно-методической концепцией авторов.

В соответствии с указанным стандартом курс информатики в профильном звене школьного образования имеет два уровня — базовый и углубленный. Базовый уровень призван обеспечить поддержку предметов по тем программам, когда информатика и информационные технологии не изучаются углубленно. Поэтому одной из целевых установок изучения информатики на базовом уровне является развитие познавательных интересов, интеллектуальных и творческих способностей через освоение и использование методов информатики и средств информационно-коммуникационных технологий при изучении различных предметов. Это не означает, однако, что курс информатики на базовом уровне решает сугубо прикладные задачи; в нем по-прежнему значительное внимание уделяется фундаментальному компоненту — освоению системы базовых знаний, отражающих вклад информатики в формирование научной картины мира, роль информационных процессов в социальных, биологических и технических системах.

Что касается углубленного уровня, то здесь авторы Стандарта исходят из того положения, что данный курс будет реализовываться в комплексе с углубленным изучением математики. Поэтому в целевых установках они формулируют освоение и систематизацию знаний, относящихся к математическим объектам информатики, овладение умениями строить такие объекты (в том числе логические формулы и программы на формальном языке), развитие алгоритмического мышления, способностей к формализации и т. п.

В концепции углубленного обучения в заключительном звене школьного образования особое место отводится элективным курсам. Однако для них в Стандарте нет установочных положений. Их разработка во многом ложится на плечи учителя, и мы рассчитываем, что наш учебник окажет ему в этом посильную помощь. Поэтому уделим немного внимания самой концепции элективных курсов. Согласно существующей концепции элективные курсы связаны, прежде всего, с удовлетворением индивидуальных образовательных интересов, потребностей и склонностей каждого школьника. Элективные курсы как бы компенсируют во многом достаточно ограниченные возможности как базовых, так и углубленных курсов в удовлетворении разнообразных образовательных потребностей старшеклассников. Такая компенсационная функция может быть двух видов. В первом случае речь идет не столько о компенсации ограниченных возможностей базовых и углубленных курсов в удовлетворении *индивидуальных* образовательных потребностей школьников, сколько о построении содержания образования по информатике, адекватного современному пониманию предмета и содержанию этой отрасли научного знания и деятельности человека. К элективным курсам такого типа следует отнести, по-видимому, курс «Исследование информационных моделей с использованием электронных таблиц» или курс «Технология работы с библиотечными и сетевыми ресурсами» и т. п. Курсы второго типа призваны дать учащимся специальные знания, умения и навыки, которые выходят за рамки общеобразовательных задач углубленного обучения. К элективным курсам второго типа можно отнести, например, курс «Математические основы информатики» или курс по компьютерной графике и т. п. Такие курсы также можно разделить на два подвида: пробные и ориентационные. Первые предназначены для того, чтобы ученик мог попробовать свои силы и «малой кровью» выяснить, то ли это, что ему надо. К таковым относится, например, разработанный элективный курс «Основы математической логики» (Просвещение, 2012). Вторые должны позволить тем, кто уже как бы определился в профессиональных склонностях, увидеть многообразие видов деятельности, связанных с данным выбором.

Третий тип элективных курсов — это курсы, которые не несут общеобразовательной нагрузки и в то же время не являются ориентированными на определенную специализацию. Примером такого курса является курс по подготовке к личным или командным олимпиадам по программированию.

Наконец, четвертый тип элективных курсов — это применение информатики в других предметных профилях: экономике, биологии, филологии и т. д.

Имея в виду использование учебника для проектирования элективных курсов, мы при его создании в большей степени ориентировались на элективные курсы первого и четвертого типов. До определенной степени он может служить основой для разработки элективных курсов второго типа. Что касается курсов третьего типа, то в силу большой специфичности стоящих перед ними задач наш учебник лишь в малой степени может быть полезен для их решения.

Но в первую очередь учебник предназначен для изучения информатики и на базовом и углубленном уровне. К теоретической базе, общей для обоих уровней, мы относим знание основных информационных процессов и особенностей их протекания в компьютеризированной среде, представление об информации и информационных системах, знание общих принципов решения задач с помощью компьютера, понимание того, что значит поставить задачу и построить компьютерную модель, знание основных способов алгоритмизации, а также принципов строения компьютера. Важным компонентом теоретической базы информатики является знание и понимание основных социально-технологических тенденций, связанных с глобальной информатизацией общества.

Приобретение учащимися информационно-коммуникативной компетентности, о чем довольно много говорится сейчас в целевых установках российского образования, для курса информатики является непосредственной целью его изучения. Указанная компетентность подразумевает, что в каждой конкретной ситуации человек способен принять решение, какая информация ему нужна для решения стоящей перед ним задачи, откуда и какими средствами эта информация может быть получена, какая коммуникативная сфера и как должна быть для этого задействована (при этом он должен уметь защищаться от возможного негативного воздействия), какими информационными средствами будет решаться задача и как будет использоваться результат. Обретения такой компетентности нельзя добиться декларациями. Это возможно лишь при условии, что ученик не просто обладает знаниями, но и умеет системно их применять, владеет необходимыми информационными технологиями. Поэтому в нашем учебнике систематично рассматриваются различные жизненные задачи, при объяснении путей решения которых демонстрируются подходы к разрешению указанных вопросов, а учащиеся приобретают умения в использовании своих информационно-технологических знаний. Тем не менее учителю необходимо каждый раз акцентировать внимание учащихся в тех моментах, когда происходит освоение того или иного элемента информационно-коммуникативной компетентности. В предлагаемой нами книге для учителя мы в таких местах стараемся оказать необходимую методическую помощь.

Совершенствование навыков использования информационных технологий опирается на умения работать с готовыми программными средствами: информационно-поисковыми системами, редакторами текстов и графическими редакторами, электронными таблицами, трансляторами с языков программирования и другими инструментальными и прикладными программами. Для реализации такого подхода занятия по информатике делятся на теоретическую и практическую части. На теоретической части создаются компьютерные модели и алгоритмы для решения задач. В ходе практических работ (лабораторных работ в компьютерном классе) учащиеся проводят компьютерные эксперименты.

К данному курсу существует Электронная форма учебника (ЭФУ) — соответствующая по структуре, содержанию и художественному оформлению печатной форме учебника и включающая в себя интерактивные ссылки, расширяющие и дополняющие материал печатного учебника.

Функциональными особенностями ЭФУ является:

- удобный и понятный интерфейс и навигация по ЭФУ;
- работа в онлайн- и офлайн-режимах;
- тестовые задания к каждой теме, разделу учебника;
- возможность добавления материалов, созданных учителем;
- инструменты изменения размера шрифта, создания заметок и закладок.

Использование ЭФУ предоставляет учителю следующие возможности:

- организовать контроль и самоконтроль по результатам изучения темы;
- реализовать технологии мобильного, дистанционного или смешанного обучения;
- реализовать требования ФГОС по формированию информационно-образовательной среды системой электронных образовательных ресурсов и др.

Рассматриваемый учебник входит в учебно-методический комплект по информатике для 10—11 классов общеобразовательной школы. Чтобы представить общий облик комплекта, опишем кратко содержание каждого из учебников.

Учебник для 10 класса

Глава I. Информатика как наука. Понятия информации и информационного процесса. Виды информационных процессов. Информационные объекты. Информатика как наука о процессах получения, хранения, передачи и использования информации. Язык как средство сохранения и передачи информации. Кодирование информации. Универсальность двоичного кодирования. Понятие информационного моделирования. Системный подход в моделировании. Алгоритмы и их свойства. Алгоритмически неразрешимые задачи. Формальный исполнитель. Универсальный исполнитель. Основные направления информатики.

Глава II. Информационная деятельность человека и использование в ней компьютерных технологий. Информационные задачи и этапы их решения. Базы данных. Эксперимент как способ познания. Компьютерная обработка результатов эксперимента. Алгоритм как форма организации процедурной информации. Алгоритмически неразрешимые задачи. Рекуррентные соотношения и рекурсивные алгоритмы. Использование алгоритмов и структур данных для решения задач. Измерение количества информации (формула Хартли).

Глава III. Моделирование процессов живой и неживой природы. Моделирование физических, биологических и социальных процессов. Границы адекватности модели. Вероятностные модели. Датчики случайных чисел и псевдослучайные последовательности. Моделирование случайных процессов. Метод Монте-Карло. Измерение количества информации (формула Шеннона).

Глава IV. Логико-математические модели. Понятие моделей искусственного интеллекта. Элементы логики высказываний. Законы алгебры высказываний. Совершенная дизъюнктивная нормальная форма. Реляционные модели. Логические функции и логические выражения. Предикаты. Кванторы всеобщности и существования. Логика СУБД Access. Построение многотабличной базы данных и работа с ней. Базы знаний и экспертные системы. Реляционная модель экспертной системы. Основы логического программирования.

Глава V. Информационные модели в задачах управления. Понятие управления. Схема разомкнутого управления. Системы с обратной связью. Гомеостаз системы. Управление по принципу обратной связи. Глобальные модели.

Учебник для 11 класса

Глава I. Информационная культура общества и личности. Понятие информационной культуры и информационной грамотности. Социальные эффекты информатизации. Методы работы с информацией. Методы свертывания информации. Моделирование как компонент информационного мировоззрения. Информационные модели в задачах управления.

Глава II. Кодирование информации. Представление информации в компьютере. Понятие кода. Кодовые таблицы. Кодирование числовой информации. Представление чисел в компьютере. Эффекты переполнения и ошибок округления. «Длинная» арифметика. Кодирование цветовой и звуковой информации. Коды, обнаруживающие и исправляющие ошибки. Префиксные коды. Понятие экономности кода. Сжатие информации. Обратимые и необратимые алгоритмы сжатия информации.

Глава III. Основные информационные объекты. Их создание и компьютерная обработка. Создание и обработка текстовых документов. Гипертекст. Основы HTML. Компьютерные словари и системы перевода текстов. Компьютерная обработка графических информационных объектов. Основные форматы графических файлов. Компьютерная обработка цифровых фотографий. Компьютерные презентации. Создание собственных web-страниц.

Глава IV. Телекоммуникационные сети. Интернет. Компьютерные телекоммуникации: локальные и глобальные сети. Топология локальных сетей. Интернет. Поиск информации в глобальных сетях. Электронная почта и другие сервисы. Распределенные информационные ресурсы. Провайдеры, регистрация и удаленный доступ. Правовые аспекты информационной деятельности в глобальных сетях. Защита от информационных атак в Интернете.

Глава V. Исследование алгоритмов математическими методами. Уточнение понятия «алгоритм». Понятия лимитирующей функции и инварианта цикла. Доказательства правильности. Вычислимые функции. Существование невычислимых функций.

Глава VI. Графы и алгоритмы на графах. Простейшие свойства графов. Способы представления графов. Алгоритмы обхода связного графа. Мосты и точки сочленения. Деревья. Каркасы минимального веса.

Глава VII. Игры и стратегии. Дерево игры. Построение стратегии. Инвариант стратегии. Игра как модель управления.

Перед создателями учебников для заключительного звена школьного образования стояли непростые дидактические задачи. Во-первых, мы были обязаны исходить из того, что учащимися в 7—9 классах освоен базовый курс информатики в объеме, предусмотренном Федеральным государственным образовательным стандартом среднего образования. Однако неодинаковое прочтение положений данного стандарта авторами разных учебников, мизерное (и потому совершенно нереальное для полного освоения) время, отводимое на этот курс базисным учебным планом, разнородность условий, в которых преподается данный курс, требуют принятия компромиссных решений как по отбору содержания (в частности, для организации повторения с целью выравнивания стартового уровня), так и по структурированию учебного материала. Во-вторых, учебник призван обслуживать как базовый, так и углубленный компоненты курса информатики старшей школы. Это означает наличие в учебнике двух уровней содержания и изложения учебного материала. В-третьих, создавая учебник, мы стремились, чтобы он оказался применимым не только в школах таких мегаполисов, как Москва или Санкт-Петербург, но и в школах, где доступ к информационным и компьютерным ресурсам весьма ограничен.

Указанные проблемы решены нами следующим образом.

1. Весь материал явным образом распределен по двум компонентам — базовому и углубленному.

2. Главы 1 и 2 в параграфах, относящихся к базовому компоненту, призваны обеспечить выравнивание уровня подготовки учащихся, осваивавших базовый курс информатики в 7—9 классах. Для кого-то из них этот материал в большей своей части окажется повторением пройденного, а для кого-то может быть практически полностью новым. Учителю здесь, в зависимости от сложившейся ситуации, предстоит самостоятельно принимать решение о количестве часов, необходимом для освоения данного материала, форме работы (в частности, организация самостоятельной работы учащихся с учебником) и т. д.

3. В целом глава 1 нацелена на формирование у школьников понятийного ядра информатики как науки. Поэтому при наличии благоприятных условий (т. е. возможности достаточно быстрого достижения необходимого стартового уровня) мы рекомендуем изучение в базовом компоненте и тех параграфов, которые формально отнесены нами к углубленному компоненту. Это обстоятельство надо учитывать при адаптации предлагаемого нами ниже тематического планирования к конкретным условиям преподавания данного курса информатики.

4. Мы исходим из того, что основные инструменты информационных технологий — текстовый и графический редакторы, электронная таблица, СУБД и т. д., а также основы алгоритмизации и какого-либо языка программирования — освоены в базовом курсе информатики 7—9 классов. Поэтому материал, относящийся к указанным вопросам, не рассматривается в теоретической части учебника, но активно применяется в компьютерном практикуме. Мы, однако, и здесь далеки от иллюзий, присущих авторам ФГОС, поэтому в текстах лабораторных работ, относящихся к параграфам первых двух глав, приведены необходимые сведения о работе с указанными инструментами информационных технологий.

5. Описание компьютерного практикума вынесено в отдельный раздел учебника (как это обычно делается с практикумами в учебниках по другим дисциплинам, например физике и химии). Такое решение, на наш взгляд, позволяет учителю более гибко планировать учебное время, нежели при жесткой фиксации места компьютерной лабораторной работы внутри объяснительного текста конкретного параграфа.

6. Указанные выше обстоятельства определили разноплановость лабораторных работ компьютерного практикума. Часть из них обеспечивает (при необходимости) освоение информационных технологий и языка программирования. В этом случае учащиеся должны сначала ознакомиться с объяснительным текстом, приведенным в описании лабораторной работы, разобраться в нем и только затем приступить к ее выполнению. Часть лабораторных работ призвана проиллюстрировать теоретические положения, разобранные в объяснительном тексте соответствующего параграфа. Как правило, текст такой лабораторной работы краток и содержит в основном исходные данные для проведения нужных компьютерных экспериментов. Самая большая часть лабораторных работ носит исследовательский характер — в ходе выполнения такой работы учащиеся открывают новые свойства, новые закономерности, исследуют обнаруженные ими эффекты, производят оптимизацию. Описание такой работы содержит задания и обсуждение результатов, которые получают учащиеся при их выполнении. Текст описания, как правило, довольно большой, и школьники знакомятся с ним порциями сообразно последовательности выполняемых заданий. Учителю в этой ситуации важно настроить учащихся на то, чтобы получаемые учащимися выводы четко формулировались и фиксировались, например, в специальной тетради.

7. Мы учитываем, что 11 класс, особенно второе полугодие, — это время активной подготовки учащихся к выпускным и вступительным экзаменам. Поэтому в 11 классе, на наш взгляд, не следует предлагать материал, требующий от учащихся глубокой и длительной теоретической работы, осваивания сложных понятий. Эта позиция определила распределение материала по учебникам 10 и 11 классов. В нашем учебнике также до определенной степени выдержан принцип модульности, хотя полной независимости глав нет. Это, в частности, связано с тем, что, как говорилось выше, на первом этапе необходимо привести учащихся к единому стартовому уровню. Поэтому главы 1 и 2 должны изучаться в самом начале. Глава 4 «Логико-математические модели» фактически не зависит от глав 3 и 5 и потому может изучаться и сразу после главы 2, и после

главы 5. Некоторые мотивы, по которым мы поставили эту главу между главами 3 и 5, приведены при обсуждении методики преподавания материала указанной главы.

8. Каждая из глав 3—5 разработана так, что в виде отдельного модуля (вместе с примыкающими к ним лабораторными работами) может составить основу элективного курса с тем же названием, что и название главы. Количество часов в этом случае может быть увеличено, а материал расширен. В методических рекомендациях к соответствующей главе мы предлагаем возможные варианты такого расширения.

Для удобства пользования структура данного пособия в основном повторяет структуру учебника для 10 класса — в нем пять глав с теми же названиями, что и в учебнике.

Каждая глава пособия имеет одинаковую структуру: краткая аннотация к главе, за которой следует изложение методических рекомендаций, сформулированных отдельно для каждой изучаемой темы. Эти рекомендации, в свою очередь, содержат:

— формулировку образовательных, развивающих и воспитательных целей;

— указание на место изучаемого материала в курсе и его связь с материалом других тем;

— описание методических приемов, полезных при объяснении теоретического материала;

— разбор возможных вариантов выполнения заданий к параграфам учебника, относящихся к данной теме;

— описание организационных моментов при проведении компьютерного практикума, если таковой предусмотрен тематическим планированием.

* * *

Естественно, книга адресована в первую очередь учителям, работающим по обсуждаемому учебнику. Но мы думаем, что и учитель, работающий по другому учебнику, найдет в ней для себя немало полезного. Мы с благодарностью примем от читателей замечания и предложения, направленные на улучшение книги.

Курс информатики в 10 классе начинается с повторения материала, изученного в соответствии с положениями Государственного стандарта основного общего образования по информатике в 8—9 классах, и восстановления навыков работы на компьютере. В тематическом планировании повторение выделено в отдельную тему. В нее включен частично и материал главы 1, который следует рассматривать как обобщающее и расширенное повторение, ибо для большинства школьников упор на формирование понятийной базы может оказаться непривычным, а с некоторыми понятиями и их трактовками они просто встретятся впервые. В сопровождающем этот материал компьютерном практикуме предусмотрено не только восстановление навыков работы на компьютере и использования информационных технологий, но и дальнейшее их совершенствование.

Одним из краеугольных камней нашего курса является изучение информационного моделирования. Учащихся необходимо привести к пониманию, что успешное решение жизненных задач, в том числе с применением компьютерной техники, возможно лишь при четком соблюдении законов моделирования и формализации. Построение модели, ее реализация средствами информационных технологий, проверка адекватности построенной модели — вот непреложные этапы компетентного подхода к решению задач, возникающих в жизни и деятельности человека информационного общества.

Мы призываем учителя иметь в виду, что главной задачей курса информатики является воспитание у учащихся информационной культуры. Напомним, что в содержание понятия «информационная культура» входят:

— понимание закономерностей информационных процессов;
— умение организовать поиск и отбор информации, необходимой для решения стоящей перед человеком задачи;
— умение оценивать достоверность, полноту, объективность и другие характеристики поступающей информации, в том числе при межличностном информационном общении;
— умение представлять информацию в различных видах, обрабатывать ее посредством подходящих информационных (в том числе компьютерных) технологий;

— умение применять полученную информацию для принятия решений;
— понимание необходимости и умение представить результаты своей информационной деятельности для использования их другими членами общества;
— этичное поведение при использовании информации.

Из этого перечня видно, что фактически в каждой теме курса информатики 10 класса находят свое отражение компоненты информационной культуры. Кроме того, информационная культура личности предполагает наличие у человека определенных компетенций по отношению к продуктам информационной деятельности (созданным как им самим, так и другими людьми), к способам обмена этими продуктами, к способам их хранения, а также по отношению к техническим и программным средствам информационной деятельности. Эти компетенции развиваются у учащихся постепенно, проходя следующие уровни:

— *уровень исполнительской компетентности*: умение точно и правильно создавать информационный продукт или совершать над ним заданную операцию по известной схеме, образцу;
— *уровень технологической компетентности*: умение самому планировать, придумывать схему создания информационного продукта или операций над ним;
— *уровень экспертной компетентности*: умение давать обоснованную качественную оценку информационному продукту, указав его достоинства и недостатки;
— *уровень аналитико-синтезирующей компетентности*: умение на основе анализа готового информационного продукта и технологии обращения с ним предлагать изменения в структуре самого продукта или технологии его изготовления.

Это расшифровка уровней по отношению к информационным продуктам[1]. Аналогично строятся расшифровки по отношению к процессам информационного обмена и хранения информации.

Полагаем, что по крайней мере первый уровень — уровень исполнительской компетентности — был освоен учащимися в базовом курсе информатики 8—9 классов, основным для освоения в 10 классе будет уровень технологической компетентности, хотя в подходящих случаях мы обращаем внимание учителя и на возможность формирования у учащихся компетентностей последующих уровней.

Приступим к обсуждению тем, обозначенных в тематическом планировании курса.

[1] Разработка компетентного подхода к формированию информационной культуры учащихся проводилась совместно с В. В. Мачульским.

ГЛАВА 1

Информатика как наука

Тема 1. Первый урок этой повторительной темы мы рекомендуем провести в форме беседы: вспомнить с учащимися, с какими понятиями они уже знакомы из курса информатики, изучавшегося ими в предшествующих классах. Естественно, что центральным понятием выступает здесь понятие информации. Уже этого замечания достаточно, чтобы перейти к рассмотрению материала § 1, однако предпочтительнее, на наш взгляд, остановиться и акцентировать внимание учащихся на том, что научное знание — это система идеальных моделей и методов их применения для решения задач, возникающих в человеческой практике. Мы употребили здесь словосочетание «идеальная модель» вместо слов «информационная модель» (хотя, как разъяснено в учебнике, научное знание оперирует именно с информационными моделями), поскольку далее понятие «информационная модель» будет описано не на интуитивном уровне, а вполне строго.

Материал § 1 начинается кратким мотивационным введением, после которого учащиеся знакомятся с термином **информация** (или вспоминают, что им об этом понятии известно из предшествующего курса информатики) и определением информатики как науки, изучающей технологию сбора, хранения, переработки и передачи информации. Понятие информации является базовым и, строго говоря, неопределяемым в рамках информатики. Поэтому мы можем лишь охарактеризовать его, описывая свойства и возможные взаимодействия этого понятия с другими понятиями информатики.

Отметим тем не менее, что в учебнике представлены три точки зрения на понятие «информация». Одна из них, которую естественно назвать кибернетической, рассматривает информацию как решающий фактор деятельности живых организмов. Именно кибернетика, из которой выросла современная информатика, стала первой из наук, целенаправленно изучающей информационные процессы в природе и обществе с позиций использования ее в задачах управления. Другая точка зрения состоит в том, что информация — это результат осознания человеком фактов и закономерностей окружающей его действительности. В этом смысле информацию нередко отождествляют со знанием, которым обладает каждый индивидуум и общество в целом. Именно такого взгляда на информацию придерживаются науки, изучающие человека и общество: медицина, социология, психология и т. п. Наконец, третья упоминаемая в учебнике точка зрения состоит в том, что информация — это отражение существующего разнообразия в окружающем человека мире. Именно она позволяет отвлекаться от смыслового содержания информации (что совершенно недопустимо для двух других, упомянутых выше). В частности она позволяет рассматривать как информацию произвольную последовательность сигналов (или символов), что находит свое применение при рассмотрении вопросов передачи и хранения информации сугубо с технологических позиций. Более того, именно такому рассмотрению понятия информации, как правило, отдается предпочтение в большинстве существующих учебников информатики. И именно на этом пути в первую очередь возникают единицы измерения количества информации (биты, байты, килобайты и т. д.) в рамках их применения для измерения информационного объема памяти компьютера или дискеты. Последнее подробно обсуждается в § 4.

Говоря о способах представления информации, надо иметь в виду, что здесь применяются два различных основания для их классификации. Первое основание — это разделение по тому признаку, какой из органов чувств человека участвует в процессе приема информации. Зрением мы воспринимаем визуальную информацию, обонянием — запаховую, осязанием — тактильную, слухом — звуковую и т. д. Иными словами, особенности восприятия информации связаны с устройством и функционированием первой сигнальной системы человеческого организма. В основе второго варианта классификации лежит использование второй сигнальной системы, т. е. членораздельной речи. В этом случае информация кодируется некоторым набором символов, и человек тем самым имеет дело с символьным способом представления информации. Символьная информация может быть представлена и визуальной формой (текст на бумаге), и звуковой (устная речь), и тактильной (азбука для слепых). Исторически, однако, сложилось так, что равноправными считают символьную, графическую и звуковую формы представления информации. Связано это, по-видимому, с тем, что активно разрабатываются и в настоящее время широко используются компьютерные средства обработки текста, графики и звука. Мы в учебнике следуем этой традиции, но приветствовали бы инициативу учителя, который провел бы в этом вопросе более четкое разделение в указанном ключе.

Перейдем к обсуждению заданий. Задание 1 к § 1 мы советуем предложить на дом; выполняя его, учащиеся расширяют свои знания по применению термина «информация». Если в качестве источника примеров служат печатные средства массовой информации, то учащимся можно предложить вырезать соответствующие сообщения и наклеить их в тетрадь. В других случаях они могут переписать сообщение. Необходимо, чтобы при этом учащиеся указывали источник, откуда взята цитата (это первичный элемент информационной культуры и соблюдения авторского права). На следующем занятии можно обсудить наиболее интересные примеры, найденные учащимися. По мере работы с учебником такие примеры у учителя накапливаются, и в последующие годы можно будет приводить их на занятиях для сравнения.

Обсуждая процитированное в задании 2 высказывание Н. Винера, надо обратить внимание учащихся, что, по существу, оно трактует понятие информации так же, как это сделано в объяснительном тексте. Отличие же состоит в том, что это понятие применено исключительно к человеку. Связано это с тем, что в 40-е годы прошлого столетия, когда формировалось современное понятие информации, еще сильны были антропоцентристские позиции, согласно которым человек возвышался над природой (в том числе над другими живыми существами), в частности считалось, что только человеку свойственна разумная деятельность, а следовательно, только применительно к человеку можно говорить о том, что информация имеет смысловое содержание. Современные научные исследования все чаще свидетельствуют об информационной деятельности, присущей многим высшим животным — дельфинам, собакам, обезьянам, некоторым видам птиц и т. д. Они обмениваются информацией, умеют ее извлекать и осознанно (а не инстинктивно) использовать в своей деятельности. Полезно после обсуждения данного задания (мы рекомендуем провести его в классе) попросить найти дома в научно-популярной литературе примеры, свидетельствующие о наличии информационной деятельности у животных. Частично выполнение этого задания предвосхищает задания 6 и 7 из § 2.

Ответ на вопрос задания 3 достаточно очевиден: конечно, одна и та же информация может содержаться в разных информационных объектах. Например, результаты очередного круга футбольного чемпионата могут быть переданы текстовым сообщением, а могут быть представлены таблицей.

Задание 4 репродуктивно — здесь учащиеся должны воспроизвести соответствующие формулировки.

Предназначение § 2 — убедить учащихся, что информация проявляет себя через информационные процессы, среди которых наиболее важными являются получение, хранение, передача, обработка и использование информации. Хотя учащиеся должны знать основные признаки каждого из указанных процессов, не следует, на наш взгляд, очень жестко их разграничивать. Более того, желательно подчеркивать, что в рамках каждого из этих процессов присутствует в той или иной форме информационный процесс другого вида. Скажем, при передаче информации обязательно требуется учитывать процессы ее получения и хранения; получение информации тесно связано с ее обработкой и т. п. Говоря об участии информации в том или ином информационном процессе, мы на самом деле лишь акцентируем внимание на определенные стороны этого процесса, относя его к тому или иному виду.

При обсуждении процесса обработки информации нами выделены два типа обработки: формальная и эвристическая. В большинстве учебников информатики эвристической обработке информации внимания не уделяется вообще. Тем самым человек как активно действующий, творческий компонент информационного мира оказывается за бортом курса информатики. Это, на наш взгляд, противоречит целевым установкам курса информатики, призванного показать ученикам уравновешенную информационную картину мира. Выведение человека за рамки этой картины противоречит принципам материально-информационного единства мира, лишает возможности воспитывать у учащихся конструктивно-критическое отношение к информации, окружающей их в повседневной жизни, делая тем самым недостижимой во всей полноте цель подготовки школьников к жизни в информационном обществе.

Эвристическая линия проводится в учебнике весьма последовательно. Ей специально уделяется внимание и при изучении информационных технологий, и при изучении глобальных информационных ресурсов, и при изучении алгоритмизации. В соответствующих местах данного пособия мы будем особо останавливаться на этом аспекте.

Надо иметь в виду, что эвристика присутствует не только в процессе обработки информации, хотя в учебнике о ней говорится именно в таком контексте. К примеру, мы отмечаем в учебнике, что сохранение информации всегда связано с ее кодированием символами какого-либо языка. Само такое кодирование является формальной процедурой, но разработка кода — процесс эвристический. А при получении информации мы всегда принимаем эвристическое решение, какая именно информация для нас существенна, а какой можно пренебречь. Мы призываем учителя обращать внимание учащихся на эвристические аспекты рассматриваемого учебного материала.

Для адекватной проверки задания 2 приведем наш вариант заполнения таблицы.

Информационный процесс				Неинформационный процесс
Получение информации	Передача информации	Хранение информации	Обработка информации	
б, д		*з*	*в, ж, и, м*	*а, г, е, к, н*

Конечно, при обсуждении этого задания учителю следует проявлять определенную гибкость. Может, например, возникнуть дискуссия, считать ли фотографирование обратной стороны Луны (пункт *д*) получением информации. Разумеется, само фотографирование как физический процесс воздействия света на фоточувствительные элементы не является информационным, но ясно, что цель данного процесса — получение информации. Здесь полезно акцентировать внимание учащихся на том, что любой информационный процесс сопровождается тем или иным физическим процессом, без которого он просто неосуществим.

Заполнение таблицы может быть продолжено за счет примеров, которые будут приводить учащиеся, выполняя задание 3.

Для выполнения задания 6 учащимся, возможно, придется использовать не только учебник биологии, но и дополнительную литературу (например, какую-либо энциклопедию). Наиболее известный способ сохранения и передачи информации в животном мире — это запах. Именно посредством запаховых меток одни животные информируют других того же вида о территориальных притязаниях. Но встречаются и иные способы, например медведи и тигры оставляют следы когтей на деревьях, по которым их соплеменники могут принять решение, стоит ли оспаривать данную территорию. Впрочем, возможны и другие примеры того, как животные сохраняют и передают информацию.

При обсуждении задания 7*б* надо иметь в виду, что, вообще говоря, речь идет не о каких-то особых органах чувств у животных, которых нет у человека. Однако те же органы чувств у животных имеют другой диапазон чувствительности. Например, летучие мыши могут издавать и улавливать ультразвук и пользуются этой способностью для эхолокации, многие высшие животные имеют более острое обоняние, чем человек, рыбы имеют так называемую боковую линию, которая позволяет им улавливать колебания воды, многие животные чувствительны к сейсмическим волнам и т. д.

Поскольку § 2 носит повторительный характер, для экономии времени можно задать его для самостоятельного прочтения дома, а в классе ограничиться последующим разбором заданий и необходимыми комментариями.

Восприятие учениками материала § 3 может быть затруднено в силу вполне естественных причин — он весьма многопланов и идейно насыщен. Начать можно с обсуждения того, как каждый из учащихся понимает, что такое язык. Можно дать предварительно задание найти в словаре толкование этого понятия. К примеру, в словаре С. И. Ожегова приведено шесть толкований слова «язык». Вот два из них, наиболее важных для нас:

«Язык — система звуковых, словарных и грамматических средств, объективирующая работу мышления и являющаяся орудием общения, обмена мыслями и взаимного понимания людей в обществе».

«Язык — система знаков (звуков, сигналов и т. п.), передающих информацию».

В этих толкованиях в первую очередь выделяются две характеристики языка: как язык устроен (звуки, сигналы, слова, грамматические правила, определяющие построение слов и фраз), т. е. синтаксический аспект языка, и его смысловая составляющая (способность нести информацию), т. е. семантический аспект. Выделен и третий аспект — прагматика языка, т. е. его целевое назначение: быть орудием общения, обмена мыслями и взаимного понимания людей.

Рассматривая синтаксический аспект, следует, конечно, подчеркнуть, что базисом любого языка является алфавит, который мы определили как конечный упорядоченный набор символов. На самом деле в математике и информатике иногда приходится использовать и бесконечные алфавиты — они оказываются удобными при построении универсальных моделей вычислительных процессов. В этих случаях можно дать следующее определение «**Алфавит** — рекурсивно перечислимое множество элементарных конструктивных объектов (т. е. объектов, однозначно распознаваемых, попарно различных, воспроизводимых), рассматриваемых безотносительно к их интерпретации вне заданной формальной системы и к их внутренней структуре». Все атрибуты алфавита, указанные в этом определении, разумеется, важны — именно они обеспечивают возможность передачи информации посредством сообщений, организуемых с помощью знаковых систем. К примеру, только различимость знаков «р» и «г» позволяет не путать слова «радость» и «гадость». Рассмотрение безотносительно к внутренней структуре означает, что каждый знак алфавита воспринимается как единое целое и никакая его часть не воспринимается как знаковая составляющая. Например, букву «щ» нельзя рассматривать как результат присоединения к букве «ц» еще одной палочки в середине, с тем чтобы как-то связать роль буквы «щ» в образовании того или иного слова с ролью буквы «ц». Что касается рекурсивной перечислимости, то ее можно понимать как существование алгоритма, перечисляющего элементы алфавита один за другим. Фактически это ослабление требований упорядоченности знаков алфавита и его конечности.

Говоря об алфавите, мы, как сказано выше, отвлекаемся от интерпретации его символов. Однако при рассмотрении построения языка на алфавитной основе (что является одним из центральных моментов в § 3) вопросы об интерпретации, т. е. о придании смысла тем или иным сочетаниям алфавитных символов, становятся весьма существенными. Как правило, у учащихся не вызывает сомнений то, что разнообразные примеры языков, используемых человеком в тех или иных целях, действительно представляют собой языки. Но бросается в глаза разнородность их структуры. К примеру, в § 3 мы упомянули знаки, используемые для регулирования уличного движения (см. рис. 1.2 учебника). Все они являются языковыми знаками, поскольку созданы для сообщения определенной информации. Их совокупность составляет очень простой язык, но все же язык (в дальнейшем мы будем называть его языком типа I). У этих языковых знаков имеется важная особенность: *они неразложимы на меньшие единицы, обладающие значением*. Так, например, стрелка, разрешающая поворот налево, является знаком лишь при условии, что она берется в целом. Отдельные ее части (заостренный кончик, вертикальная часть и т. д.) ничего не означают.

Рассмотрим теперь другой тип языка.

На флоте до введения радиотелеграфии существовал язык (назовем его языком типа II), в котором имелись три элементарные формы: круг, треугольный вымпел и четырехугольный флаг. Значимыми были их различные сочетания, например: вы подвергаетесь опасности, необходима немедленная помощь, да, нет и т. д.

Нетрудно видеть, что в отличие от языка типа I в языке типа II языковые знаки *состоят из частей, повторяющихся в различных комбинациях*. Правда, эти *части сами по себе еще ничего не значат* (и в этом отношении нет никакой разницы между языками типов I и II). Однако возможность сочетания этих частей в различных языковых знаках составляет существенную особенность языка типа II: она делает его *более экономичным* по сравнению с языком типа I. В самом деле, с помощью различных комбинаций всего лишь трех элементарных форм на языке типа II можно передать достаточно большое число фактов, тогда как при пользовании языком типа I нам пришлось бы изобрести столько различных отдельных знаков, сколько существует фактов, подлежащих сообщению. Еще одним примером языка типа II может служить язык обозначений электроники (см. рис. 1.3 учебника).

И наконец, для сравнения с языками типа I и II возьмем какой-либо естественный коммуникативный язык, например русский. Проанализируем какое-нибудь высказывание на этом языке, например: «В этом месте разрешен поворот налево». Оно обозначает тот же факт, что и стрелка из языка типа I. Однако в отличие от стрелки данный языковой знак разложим на части, обладающие собственным смыслом, т. е. на меньшие языковые знаки. Составными частями высказывания прежде всего являются отдельные слова: «в», «этом», «месте», «разрешен» и т. д. Но и слова не предел деления исходного языкового знака. Некоторые из слов, в свою очередь, можно разделить на единицы, имеющие смысловое значение. Так, слово «месте» расчленяется на «мест-» и «-е». Каждая из этих частей обладает определенным значением, окончание «-е», в частности, указывает на единственное число, предложный падеж и 2-е склонение имени существительного. В слове «разрешен» можно выделить в качестве значимых частей «раз-», «-реш-» и «-ен», причем суффикс «-ен» указывает на результат действия, совершенного в прошлом, единственное число и мужской род.

Ясно, что при делении сложного языкового знака на части мы рано или поздно приходим к минимальным значащим единицам, к минимальным языковым знакам, не состоящим из меньших значащих единиц. Например, деление корня «мест-» на какие-либо меньшие элементы дает части, не имеющие смысла в системе русского языка. Следовательно, «мест-» — наименьший языковой знак. Наименьшие значащие элементы данного языка называются **монемами** (термин известного языковеда А. Мартине). Таким образом, языковые знаки обычного языка (не только русского, но и любого другого коммуникативного языка) могут быть разделены на монемы. Ничего подобного нет ни в языке типа I, ни в языке типа II.

Неделимость монем на меньшие единицы, обладающие смыслом, вовсе не означает, что они вообще неделимы. Монемы можно расчленять на части, не имеющие смыслового значения. Например, в монеме «мест-» мы можем выделить четыре элемента: «м», «е», «с», «т». Ни один из этих элементов не является сам по себе носителем какого-либо смысла. Но каждый из них встречается в других монемах, служит материалом для их построения и средством для отличия одной значимой единицы от другой (ср.: «тыл», «тол» или «цел», «мел», «дел» и т. п.). Такие элементы в естественных языках называются **фонемами**.

В языке типа I не существует ничего, что соответствовало бы фонемам обычного языка. Элементы языка типа II, из которых строятся его языковые знаки, аналогичны фонемам и выполняют ту же функцию, что и фонемы: не имея сами по себе значения, они служат материалом для построения значащих единиц и средством отличия одной значащей единицы от другой.

Основное преимущество нашего обычного языка заключается в его экономичности. Пользуясь ограниченным числом фонем и монем, можно построить практически бесконечное число высказываний. Для выражения этих высказываний в языке типа I потребовалось бы и бесконечное число отдельных языковых знаков, что было бы непосильным грузом для нашей памяти. К языкам типа I и II мы прибегаем в тех случаях, когда количество сообщаемых фактов строго ограничено или использование обычного языка по тем или иным причинам нецелесообразно. От шофера, ведущего машину, требуется быстрая реакция. Этого можно достичь не с помощью длинных словесных надписей, прочтение которых требует времени, а посредством легко обозримых компактных броских знаков, не нуждающихся в расчленении на части. Когда один корабль проходит мимо другого, в условиях разыгравшейся бури, удобно воспользоваться для передачи нужной информации тремя «фонемами» языка типа II.

Отметим, что развитие языков программирования, их «очеловечивание» (т. е. приобретение ими черт языков человеческого общения) идет по пути замены фонем монемами. Первым шагом был переход от языков двоичного кодирования команд к автокоду (ассемблеру), в котором для команд применялись имена с вполне ясной мнемоникой, т. е. можно сказать, что эти имена — просто символы другого, более «крупного» алфавита. В языках высокого уровня в качестве символов алфавита выступают имена действий, способов организации действий и организации данных. Наконец, современный этап характеризуется разработкой и применением так называемых визуализированных языков, где символами выступают те или иные графические образы.

Обсуждая с учащимися понятие коммуникативного языка, остановитесь на следующем аспекте. Большинство языков человеческого общения — русский, английский, китайский и т. д. — сложились исторически. Они называются **естественными языками**. Каждый из них несет в себе отпечаток жизни того народа, который им пользуется. Например, в языке чукчей есть свыше 100 слов для обозначения слова «снег». Падающий снег и снег, лежащий на земле, вчерашний снег — это все разные слова. Различать эти «снега» требует жизнь в суровых арктических условиях. Для общения людей, говорящих на разных языках, необходимы переводчики или по крайней мере словари. Поэтому предпринимались попытки создать **искусственный язык**, единый для всего населения земного шара. Наибольшее распространение получил язык эсперанто, хотя он и не стал общим языком всех землян. Языки человеческого общения — неважно, естественные или искусственные — как раз и называются **коммуникативными языками**. К искусственным языкам следует отнести также язык записи чисел цифрами, язык записи алгебраических выражений и, разумеется, все языки программирования.

Даже такое, проведенное весьма кратко рассмотрение феномена языка показывает, что при изучении материала § 3 можно и полезно (особенно в классах гуманитарного профиля) предложить учащимся подготовить сообщения о видах языков, используемых человеком для сохранения и передачи информации.

Основное положение, которое учащиеся должны усвоить, изучив § 3, состоит в том, что каждый язык характеризуется синтаксисом и семантикой. Синтаксис языка определяет то, как из символов алфавита строятся слова, составляющие данный язык. Правила, по которым образуются слова, называются **грамматикой**. В естественных языках грамматические правила выступают не столько средством словообразования, сколько регламентирующим средством, указывающим, какие слова, составленные из символов алфавита данного языка, не принадлежат этому языку. Согласно грамматическим правилам в русском языке нет слов перИход и парАвоз. Однако фраза «Сяпала калуша по напушке» (приведенная в задании 4 из § 3), хотя и содержит слова, не входящие в русский язык, грамматике русского языка полностью соответствует. Иное дело языки, в которых грамматические правила полностью и однозначно определяют, принадлежит ли то или иное слово данному языку или нет. Язык, определяемый грамматикой, называется **формальным**; о формальных языках речь будет идти в § 7.

Семантика языка определяет смысловое наполнение слов данного языка. В коммуникативных языках синтаксис и семантика, как правило, тесно переплетены, более того, синтаксис нередко обусловлен семантикой. Примером тому служит, скажем, правило правописания приставок *пре-* и *при-*. Грамматика (т. е. синтаксический аспект) русского языка требует в соответствии со смыслом слова (т. е. семантическим аспектом) писать *прекрасный*, но *приукрасить*. А вот в формализованных и особенно в формальных языках синтаксический и семантический аспекты разведены очень жестко. Например, в языках программирования грамматические ошибки (т. е. неправильное написание слов языка), и только они, диагностируются как синтаксические (syntax error). Тем не менее мы считаем важным, чтобы и для естественных языков учащиеся отчетливо различали эти аспекты. Для выработки понимания разницы между синтаксисом и семантикой полезно разобрать задание 3. С семантической точки зрения лишним словом в указанном наборе, очевидно, является «карандаш», ибо все остальные слова обозначают овощи. С синтаксической точки зрения лишним будет слово «морковь», ибо только оно начинается на букву «м». Можно предложить учащимся самим придумать подобные наборы слов и устроить соревнование, у кого слов в наборе окажется больше.

Основное понятие, непосредственно связанное с семантикой языка, — **формализация**, т. е. закрепление за каждым словом языка однозначно определенного смысла. Применение компьютера для решения жизненной задачи обязательно проходит этапы моделирования и формализации. Особый акцент на этом будет сделан при рассмотрении глав 3 и 4. Пока же учащиеся должны научиться уверенно распознавать случаи применения **формализованного** языка. Способствовать этому призвано задание 4, в котором фразы в пунктах *а, б* и *г* записаны на коммуникативном языке, остальные фразы записаны на формализованном языке.

Информация передается для того, чтобы вызвать соответствующий отклик у ее получателя. В естественном языке слова и предложения связываются друг с другом таким образом, чтобы сформулировать просьбу, недовольство, вопрос, указание, иными словами, так, чтобы вызвать определенную реакцию у получателя сообщения. Если же говорить об информации, получаемой механическим устройством или биологическим органом, то она вызывает однозначную реакцию, лишенную какой-либо степени свободы. Перевод с иностранного языка и инстинктивное поведение являются примерами отклика с ограниченной степенью свободы. Только у высших живых организмов мы обнаруживаем гибкий, оригинальный и творческий отклик, причем у человека — с максимальным количеством степеней свободы. Этот целевой аспект информации будем называть прагматическим. Язык характеризуется еще своей прагматикой, т. е. тем, какова цель его использования. Целью воздействия может быть приведение человека в определенное эмоциональное состояние, побуждающее его к заданным действиям. Примером может служить нейролингвистическое программирование. Как сказано выше, такая прагматика языка определяет информационное поведение с весьма ограниченной степенью свободы.

Принципиально иная прагматика видится нам, когда язык выступает как базисный элемент информационной культуры. Восприятие информации в этом случае ориентировано на проявление максимальной степени свободы.

О прагматической характеристике языка и информации в целом мы подробнее будем говорить в разделе, посвященном социальной информатике, но нам представляется полезным, чтобы уже сейчас учитель коснулся этих вопросов при рассмотрении базовых характеристик понятия «язык».

Оставшееся не прокомментированным задание 5 демонстрирует учащимся, что планарное расположение знаков алфавита вовсе не является таким уж редким. Вот еще два примера. Во-первых, это запись обыкновенных дробей с помощью дробной черты. А в алгебре учащиеся нередко встречаются с «многоэтажными» буквенными выражениями. Здесь уместно вспомнить, что в языках программирования синтаксические правила допускают только линейное расположение знаков алфавита, поэтому привычные математические формулы приходится переписывать, чтобы они удовлетворяли этому синтаксическому требованию. Во-вторых — использование верхних и нижних индексов, скажем, в записи химических формул (см. учебник, задание 4в). Вполне можно ожидать, что учащиеся приведут и другие примеры планарного (и даже пространственного) расположения символов алфавита в словах того или иного языка.

Главный тезис, обсуждаемый в § 4, — для передачи информации с помощью сообщений требуется определенный способ **кодирования**. «Кодирование» выступает здесь как научный термин и не должно в понятийном плане смешиваться учащимися с термином «шифрование». Кодирование имеет главной целью *сохранение* информации и придание ей формы, обеспечивающей полноценную (т. е. без потерь и искажений) передачу информации от источника к получателю. К примеру, в бурно развивающейся *теории кодирования* рассматриваются вопросы помехоустойчивости кодов, создания кодов, автоматически исправляющих ошибки, и т. д. Шифрование же имеет целью *защитить* информацию (уже как-то закодированную) от несанкционированного использования. Вопросы защиты информации будут затрагиваться несколько позже.

Следует понимать, что термин «кодирование» может употребляться в узком смысле и широком. В узком смысле кодирование означает отображение набора знаков и/или слов одного языка в набор знаков и/или слов другого языка. Более широко под кодированием информации понимают сопоставление объектам и отношениям между ними слов некоторого языка, т. е. именование объектов и отношений. Когда речь идет о коде ASCII, то имеется в виду узкий смысл термина «кодирование»; когда в двоичной системе счисления записывается натуральное число (при этом вовсе не нужно заранее располагать записью этих чисел в десятичной или какой-либо еще нумерации), то здесь мы имеем дело с кодированием в широком смысле — каждый объект (в данном случае число) получает имя, представленное словом в двухсимвольном алфавите. В этом же смысле употребляют термин «кодирование», когда иногда говорят, что создание программы по уже имеющемуся алгоритму представляет собой просто кодирование алгоритмов на выбранном языке программирования по несложным правилам.

В § 4 речь в основном идет о кодировании в узком смысле, и оно связывается с развитием технологии передачи и хранения информации. Мы не считаем необходимым пояснять указанное выше различие всем учащимся, но не исключаем, что какой-либо ученик способен интуитивно ощутить его; тогда учитель должен уметь объяснить такому ученику указанные два смысла термина «кодирование».

При изучении представленной темы полезно предложить учащимся по-разному закодировать одну и ту же информацию (к примеру, можно предложить закодировать фразу «Я люблю мороженое»; среди таких кодировок полезно рассмотреть перевод этой фразы на иностранный язык, изучаемый учениками данного класса, и изображение фразы с помощью пиктограмм). Довольно ярким примером кодирования информации является ребус. Рассматривая с учащимися какой-либо конкретный ребус, полезно обсудить наличие в нем двух видов кодирования: рисунки в ребусе кодируют конкретные слова (более точно, последовательность букв), а дополнительные символы (запятые, зачеркивания, вставки) или геометрическое расположение рисунков кодируют указания по преобразованию получающихся последовательностей букв. Чтобы занимательность ребуса не превалировала над учебными целями, ребусы должны быть короткими и легко разгадываемыми (например, такими, как ребус, изображенный на рисунке 1).

После освоения учащимися понятия кодирования им предлагается довольно стандартный путь к введению единиц измерения информационного объема сообщения. Правда, сам термин «информационный объем сообщения» нами не вводится, поскольку его дидактическая ценность нам представляется незначительной. Но при желании учитель

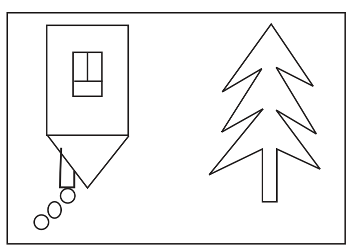

Рис. 1. Ответ: модель

может ввести данный термин и в дальнейшем им пользоваться.

В этом же параграфе обсуждаются вопросы представления и обработки видеоинформации. Важно обратить внимание учащихся на то, что видеоинформация воспринимается человеком значительно легче, чем символьная информация. Это заключение мотивирует высокую желательность представления видеоинформации в компьютере. В свете уже изученного учащиеся должны понимать, что речь тем самым идет о *способах кодирования графической информации*.

Кодирование графической информации тесно связано с различными физическими эффектами, которые обеспечивают создание цветного изображения на экране компьютера и его восприятие человеком. Поэтому изложение данного материала возможно с различным уровнем глубины. Объяснение на первом уровне как раз и дано в § 4 учебника. В этом объяснении констатируется, что почти любой цвет, воспринимаемый человеческим зрением, может быть передан как сочетание в определенной пропорции трех цветов — красного, синего и зеленого. Уровень яркости каждого цвета — величина, конечно, непрерывная, и ее можно менять, крутя, например, ручку настройки. Если же мы хотим, чтобы уровень яркости кодировался каким-то числом, то проще всего это сделать, разбив весь интервал яркости на заданное число частей. Иными словами, мы строим *шкалу яркости*. Отметим, что в графических редакторах довольно часто эта дискретизация интервала яркости скрыта, и пользователь в настройке палитры используемых цветов видит именно «рычажок» непрерывного изменения уровня яркости. Указанные физические и физиологические эффекты (к которым надо добавить еще инерционность человеческого зрения, состоящую в том, что человеческий мозг способен некоторое время хранить изображение; как известно, это создает при дискретном предъявлении кадров иллюзию непрерывности изменения изображения на экране) как раз и лежат в основе создания компьютерных изображений. На них же базируется и кодирование изображений.

Мы не исключаем, однако, что найдутся школьники, которых не удовлетворит данная глубина объяснений. Почему только три цвета выбрано в качестве основных? Почему это именно указанные цвета? И наконец, все ли цвета, которые видит человек, воспроизводимы на экране компьютера? Чтобы не отсылать читателя к специальной литературе, мы кратко ответим на эти вопросы.

Прежде всего вспомним, что видимый человеком свет имеет длину волны в довольно узком диапазоне — от 0,38 мкм до 0,78 мкм. А человеческий глаз обладает специальными рецепторами (называемыми колбочками), чувствительными именно к цвету. При этом каждая из колбочек «специализируется» уже на своем совсем узком диапазоне длин волн. Природа распорядилась так, что в человеческом глазу больше всего оказалось колбочек, чувствительных к цветам, которые мы называем красным, зеленым и синим. (У собак, например, зрение способно воспринимать цвета в диапазоне волн, начинающемся в светло-синем участке спектра и немного захватывающем невидимый человеку ультрафиолет. Человек поэтому долгое время считал, что зрение собак является черно-белым.) Человеческий мозг приспособился из сигналов, поступающих от колбочек трех указанных типов, строить цветную картину видимого человеком мира. Поэтому для создания картинки на экране компьютера естественной кажется идея использовать те три цвета, на которых в основном специализируются колбочки человеческого глаза. Ведь для создания луча своего цвета нужно отдельное техническое устройство его развертки. Даже три устройства совместить не так уж просто, да и недешево. Это ответ на первый и второй вопросы.

Теперь ответим на третий вопрос. Выше уже говорилось, что цвет определяется тем, какой уровень яркости имеет каждый из трех основных цветов. Пусть, например, выбрано 64 градации яркости. Занумеруем их от 0 до 63. В этом случае можно считать, что каждый цвет задается тройкой чисел (x, y, z) и, следовательно, ему соответствует некоторая точка в трехмерном пространстве.

Легко понять, что наборы градаций (5, 15, 30) и (10, 30, 60) основных цветов дают один и тот же цвет, только разной насыщенности. Каждую из этих троек можно воспринимать как вектор в трехмерном пространстве. Ясно, что эти два вектора сонаправлены, просто один вектор составляет половину второго. Это означает, что цвет (без учета насыщенности) определяется не самими уровнями яркости, а отношением этих уровней (в нашем примере $5:15:30$). Тогда без ограничения общности можно считать, что сумма трех членов отношения равна, скажем, 1. Наш пример в этом случае запишется как $0{,}1:0{,}3:0{,}6$. Иными словами, если мы говорим о самом цвете без учета его насыщенности, то он определяется *двумя* числами; третье получается вычитанием суммы первых двух из 1. Но если мы имеем дело с двумя переменными, то их значения можно изображать точкой на плоскости. Тем самым каждому цвету соответствует точка координатной плоскости XOY.

В 1931 г. в качестве Международного стандарта была принята система CIE (Commission Internationale d'Eclairage). В этой системе три основных цвета стандартизированы по длине волны и имеют фиксированные координаты в плоскости XOY, указанные в таблице 1.

Таблица 1

	Красный (*R*)	Зеленый (*G*)	Синий (*B*)
x	0,72	0,28	0,18
y	0,27	0,72	0,08
L, мкм	0,7	0,5641	0,4351

Эксперименты ученых показали, что все цвета, видимые человеком, образуют в плоскости *XOY* область, изображенную на рисунке 2. Основные цвета системы *RGB* (названной так по начальным буквам трех основных цветов) дают вершины внутреннего треугольника, а их произвольная комбинация — это точка внутри полученного треугольника. Следовательно, только цвета этого треугольника можем увидеть мы на экране компьютера, сколько бы градаций каждого из основных цветов ни позволяла передать видеокарта.

Если к трем цветам добавить четвертый, не лежащий внутри треугольника, то на цветовой диаграмме все получаемые цвета заполнят уже четырехугольник с вершинами в точках, изображающих эти цвета. Поэтому, добавив в число основных цветов еще несколько, *можно* расширить палитру цветов, воспроизводимых на экране компьютера. Однако, как уже отмечалось, это обойдется весьма недешево. Кроме того, сколько бы цветов в видимой части спектра мы ни брали, нам никогда не удастся покрыть всю область видимых человеком цветов — она ведь не является многоугольником. А чтобы получить все цвета видимой области в качестве основного, пришлось бы выбрать хотя бы один невидимый цвет. Но как же тогда управлять этим невидимым цветом? Возможности человека ограничены, и это, как обычно, создает ограничения на разработку технологий.

Точно так же с разной степенью подробности может излагаться техническая сторона получения цветного изображения на экране монитора. В учебнике, к примеру, мы просто констатировали, что основные цвета группируются по три (каждая такая группа называется **триадой**). С другой стороны, ясно, что электронные лучи, создающие изображение, пробегают (разумеется, непрерывно) весь экран. Как же тогда получается дискретное изображение в виде триад? А без

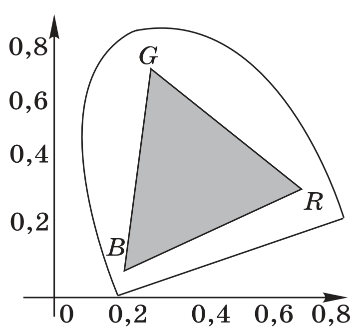

Рис. 2

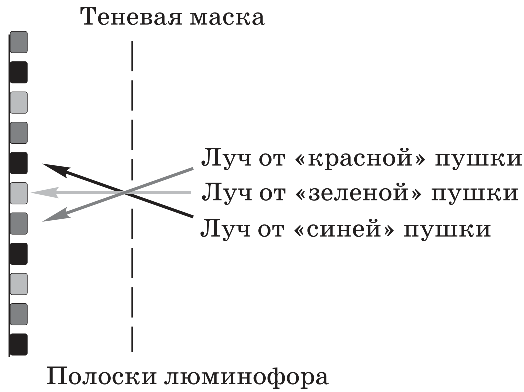

Рис. 3

такой дискретизации опять-таки невозможно цифровое кодирование.

Прежде всего отметим, что электронный луч, т. е. поток электронов, сам по себе цветным не бывает. Цвет создает люминофор, нанесенный микрополосками на внутреннюю сторону экрана. Электронный луч, отвечающий за данный цвет, должен попасть точно на свой люминофор: если он попадет на соседний, то на экране появится цвет, которого быть не должно. Но электронный луч перемещается непрерывно и, значит, неизбежно попадает не на свои люминофоры. Чтобы этого не происходило, между экраном и электронными пушками (так называют источники электронных лучей) помещают **теневую маску**. Она представляет собой металлическую пластину, в которой просверлены тысячи отверстий. Каждое отверстие соответствует одной триаде. Маска располагается так, чтобы каждый из трех лучей, проходя через отверстие, попадал в точности на свой люминофор (рис. 3).

Как и во многих других случаях, в изложении этих вопросов переплетается несколько дидактических линий; в данном случае это линии представления информации и устройства компьютерной техники.

Задания, приведенные в конце § 4, вряд ли могут вызвать затруднения у учащихся. В задании 2 речь идет, разумеется, не о точном числе (оно может меняться от страницы к странице), а о некоторой приближенной оценке. Чтобы получить такую оценку, достаточно подсчитать число символов в нескольких строках (не забывая о пробелах), взять среднее значение и умножить его на число строк. Считая, что один символ соответствует одному байту, мы получаем требуемый ответ.

Для понимания связи между кодированием, алгоритмизацией и программированием полезно представить учащимся различные способы кодирования последовательностей действий. Один из способов представлен в задании 3. Каждому ясно, что решения задач 3а и 3б представляют собой алгоритмы, записанные на формализованном языке из 4 символов. Если бы существовал транслятор с этого языка, то это был бы язык программирования.

Ответ к заданию 3а: ↑↑→↑→↓↓↓→↑↑↑↑→↓↓↓→↑↑↑↑.

Ответ к заданию 3б: ↑↑↑↑←←←←←←←↓↓↓→↓←↓↓↓→→
→↑←←↑→↑↑↑→↓↓↓→↓↓→→→→↑→↓→↑↑↑↑↑←↑→→↓↓↓↓→
→↓←←↓→→→↑↑↑↑↑↑↑↑←←←←←←←↑.

Для задания 3в приведем решение. Поскольку для кодирования маршрута нужно 4 стрелки, каждая из них может быть закодирована двухбитовой последовательностью. Тогда последовательность, построенная при решении задания 3а, перекодируется в 88-битовую последовательность. Аналогично получается ответ для последовательности, являющейся решением задания 3б.

При изучении информатики на базовом уровне материал § 1—4 рассматривается весьма кратко (в течение двух уроков) — как повторение тех тем курса информатики за 8—9 классы, которые обозначены в названиях этих параграфов. В частности, некоторый материал (по усмотрению учителя) может быть оставлен для самостоятельного изучения дома.

Рекомендуется провести первое занятие в компьютерном классе с целью восстановления навыков работы на компьютере и в среде той операционной системы, которая в дальнейшем будет использоваться. В учебнике не приводилось описание лабораторной работы, ориентированной на такое восстановление навыков, поскольку в российском школьном образовании слишком велико разнообразие используемых вариантов интерфейсов операционных систем. В классах с профильным изучением информатики запланировано проведение двух уроков в компьютерном классе. В этом случае второй урок можно посвятить восстановлению навыков работы с текстовым, графическим и звуковым редакторами.

Тема 2. § 5 начинается словами А. П. Ершова, чтобы лишний раз подчеркнуть стержневую роль моделирования для информатики как научной дисциплины. Мы рассчитываем, что в курсе, изучавшемся школьниками в 8 и 9 классах, была в достаточной мере мотивирована необходимость освоения моделирования как метода научного познания и решения задач, возникающих в человеческой практике. Впрочем, не исключается вариант, что из всего изученного в предшествующих классах учащиеся помнят только, как писать программы, или только работу с прикладным программным обеспечением. В таком случае учителю придется уделить внимание мотивированию необходимости изучать методы информационного моделирования. Сделать это можно по-разному; ниже приводится два примера возможного диалога учителя с классом. В них мы не персонифицируем учащихся, указывая везде *Ученики*, подразумевая, что отвечает кто-то один. Хотя все диалоги взяты из реальной практики, нет, конечно, никаких гарантий, что при попытке воспроизвести подобный диалог в другом классе он получится таким же. Тем не менее, все принципиальные моменты диалога должны быть обозначены и зафиксированы учащимися.

Вариант 1

Учитель. На уроках информатики мы с вами уже говорили об информационных процессах. Какие виды информационных процессов нами рассматривались?

Ученики. Получение информации, хранение, передача, обработка и использование.

Учитель. Верно. Давайте внимательно рассмотрим процесс получения информации. Скажем, вас пригласили в гости, но вы мало кого знаете в той компании. Процесс знакомства — это, конечно, процесс получения информации. С чего он начнется?

Ученики. Скорее всего, с представления друг другу.

Учитель. Конечно, вероятнее всего вы первым делом узнаете, кого как зовут. А дальше?

Ученики (обычно после некоторых размышлений). Ну, узнаем, кто чем увлекается, кто где живет...

Учитель. На самом деле вы получаете информацию не только из разговоров. Зрение позволяет вам зафиксировать облик новых знакомых и прежде всего их лица — надо же уметь сопоставлять имена с их владельцами! Вы слышите тембр голоса. Вы фиксируете манеру поведения, и у вас вырабатывается эмоциональное отношение — она вам нравится или не нравится. Впрочем, это уже другой информационный процесс: обработка информации. Поэтому мы остановимся и подумаем, что же произошло с точки зрения информатики?

После возвращения домой вы вряд ли обнаружите у себя в квартире ту же компанию. Но вся полученная вами информация останется с вами. Произошла замена реальных объектов и отношений между ними информацией о них. Этот новый объект, никак не воплощенный материально, называют информационной моделью. Конечно, вы можете сесть и записать свои впечатления. Тогда ваша модель обретет материальный носитель. Правда, все ли удастся перенести на бумагу? Наверно, нет. Так что это будет уже другая модель. Можно сказать, информационно более бедная, чем та, которая имеется в вашем сознании.

Теперь на основании имеющейся модели вы будете решать, хотите ли вы продолжить общение с этой компанией или, быть может, только с отдельными ее представителями либо вообще будете держаться подальше. Можно сказать, что целью моделирования, возможно и неосознаваемой вами, является именно решение вопроса о желательности или нежелательности общения. И заметьте, что получаемая вами информация была существенна именно для решения этого вопроса. Всякая другая информация, даже если она звучала, пропускается, как говорят, мимо ушей. Вам, к примеру, в этой ситуации, вероятнее всего, нет дела до того, сколько часов человек спит или регулярно ли он выполняет домашние задания.

Далее рассматриваются еще один-два примера процессов получения информации, в результате которых строится информационная модель и показывается, что такое построение всегда имеет определенную цель, которая задает критерий существенности информации, отбираемой для данной модели. В заключение делается вывод, что раз модели присутствуют в каждом процессе получения информации, с ними надо познакомиться поближе.

Вариант 2

Учитель. Скажите, пожалуйста, знакомо ли вам слово «модель»? Что этим словом обычно обозначают?

Ученики. Что-то, заменяющее реальный объект.

Учитель. Например?

Ученики. Модель самолета, автомобиля, танка.

Учитель. А зачем нужны модели?

Ученики. Чтобы играть.

Учитель. И только?

Ученики. Ну, на моделях самолетов проверяют, хороша ли их форма для полета... Архитекторы строят модели застройки жилых кварталов, чтобы понять, как это будет выглядеть в натуре...

Учитель. Правильно. А глобус — это модель? И если да, то чего?

Ученики. Конечно, да — земного шара.

Учитель. И для чего же служит эта модель?

Ученики. Чтобы изучать, что где на Земле находится.

Учитель. Изучать — в смысле учить географию. Ведь для того чтобы создать глобус, надо было, как вы знаете, сначала великим путешественникам и ученым изучать саму нашу планету. А еще глобус помещают в кабину космического корабля, и он вращается вокруг своей оси с той же скоростью, с какой корабль движется по орбите. Как вы думаете, зачем там глобус?

Ученики. Чтобы космонавт знал, над какой точкой земного шара он сейчас пролетает.

Можно разобрать еще какие-либо примеры, чтобы убедить школьников в наличии у любой модели целевого атрибута. Этих примеров, как правило, вполне достаточно для понимания школьниками, что моделирование является важным видом человеческой деятельности и его полезно изучать.

Основные понятия, содержание которых раскрывается в данной теме, — это **моделирование, модель, информационная модель.**

Довольно распространено мнение о модели как упрощенном представлении объекта, процесса или явления. Это, вообще говоря, неверно. Что, к примеру, сложного в явлении падения яблока на землю? Однако математическая модель этого явления представляет собой не такую уж простую формулу, выражающую закон всемирного тяготения. Цель моделирования не в стремлении упростить изучаемый объект (процесс или явление), а в представлении его в такой форме, чтобы для исследования интересующего объекта можно было применить имеющийся у человека инструментарий. Скажем, пока в математике изучена только линейная функция, в физике мы можем математически моделировать лишь равномерное прямолинейное движение. Знание квадратичной функции позволяет математически моделировать уже равноускоренное движение. Эти вопросы вполне могут и должны обсуждаться с учащимися в данной теме.

Здесь же вводится основной для информатики класс моделей — информационные модели. Этому предшествует обсуждение понятия модели в смысле копии (другими словами, имитации) объекта, процесса или явления. Такое обсуждение преследует две цели: с одной стороны, снять у учащихся психологическое напряжение, связанное с введением нового понятия, с другой — более контрастно оттенить отличительные характеристики именно информационных моделей. В ходе обсуждения понятия информационной модели полезно предложить учащимся еще раз рассмотреть модели, перечисленные в таблице 1.3, и выделить информационные, каждый раз указывая параметры, характеризующие эти модели, и связи между параметрами.

Отдельного обсуждения заслуживает понятие математической модели. Важно подчеркнуть, что это частный случай информационной модели. Если уровень математической подготовки класса позволяет, то можно уже тут обсудить, что математические модели могут описываться не только языком уравнений и неравенств (более общо, языком функциональных зависимостей), но и, к примеру, языком теории множеств или теории графов (о которых, кстати, речь все равно будет идти).

Но особого внимания заслуживает понятие компьютерной модели, ибо именно реализация информационной модели на компьютере — основа сопряжения информатики с остальными научными дисциплинами и человеческой практикой.

В объяснительном тексте приведена последовательность этапов, которые надо выполнить, чтобы построить информационную модель. Изучая материал последующих глав, учащиеся неоднократно будут ее исполнять, и мы убеждены, что это станет полезным стереотипом. Разумеется, школьники должны понимать, почему необходимо выполнение каждого этапа и почему именно в такой последовательности. Чтобы добиться этого, полезно продемонстрировать учащимся на примерах, к каким негативным последствиям может приводить пропуск этапов или перестановка их выполнения. Приведенное в учебнике высказывание: «Гладко было на бумаге, да забыли про овраги, а по ним ходить» — один из таких примеров.

Чтобы учащиеся легче осознали зависимость выбора факторов от целевых установок, полезно обсудить с ними следующее задание:

Задание 1. По заказу Управления культуры была изготовлена бронзовая статуя девушки с веслом. Определите факторы, существенные для решения каждой из следующих задач:

а) перевезти статую из мастерской в городской парк;
б) установить статую на площадке парка;
в) увеличить посещаемость городского парка;
г) продать статую с аукциона.

У учащихся при обсуждении этого задания полезно спрашивать, для каких задач выделенные ими существенные факторы формализуемы, а для каких нет. Например, в пункте *в* очевидным фактором выступает эстетическая ценность данной статуи. Ясно, что он неформализуем. Наоборот, в пункте *а* в качестве существенных факторов выступают размеры и/или вес статуи. Очевидно, что это формализуемые факторы.

Продемонстрировать влияние средств реализации информационной модели (т. е. исполнителя и языка) на выбор параметров можно следующим примером. Допустим, вы въезжаете в новую квартиру и вам нужно определить оптимальную расстановку мебели. Конечно, можно проводить натурный эксперимент, передвигая шкаф и диван от одной стенки к другой до тех пор, пока не получится приемлемый вариант. Но, наверное, каждый предпочтет сначала все «передвижения» осуществить с компьютерной моделью, а затем уже понравившийся вариант воплотить в жизнь. Факторы здесь понятны: надо, чтобы предметы мебели не «наезжали» друг на друга, не заслоняли окна и двери и т. п. Какую компьютерную технологию выбрать? Если это язык программирования или электронная таблица, то модель придется описывать математическими соотношениями. Например, если комната имеет размеры 6 м в длину и 4 м в ширину, то, выбрав подходящую систему координат, можно записать, что комната — это множество точек, координаты которых удовлетворяют неравенствам $0 \leqslant x \leqslant 6$ и $0 \leqslant y \leqslant 4$. Затем можно похожими неравенствами записать размеры шкафа, дивана и другой мебели и их положение в комнате. После этого придется математическими соотношениями записать ограничения на расположение мебели в квартире и относительно друг друга. И наконец, надо придумать алгоритм поиска оптимального решения. Пожалуй, реализация такой модели оказывается тяжелее натурного перемещения мебели. Совершенно иначе строится компьютерная модель, если в качестве технологии выбран графический редактор. Комнату, а также все предметы мебели можно изобразить графическими примитивами. Перемещая их, довольно легко найти подходящее расположение. Если редактор позволяет строить трехмерные изображения, то вы сможете увидеть проектируемый интерьер в объеме. Цели моделирования и факторы в обоих случаях одинаковы, однако язык описания параметров модели и способ описания связей между параметрами различаются принципиально.

Довольно непростым вопросом для учащихся может оказаться различение понятий **фактор** и **параметр**. В объяснительных текстах учебника не дается точных определений ни того, ни другого понятия. Это не случайность (и тем более не небрежность авторов). Дело в том, что одна и та же характеристика в одной ситуации может рассматриваться как фактор, а в другой — как параметр. Здесь важна *соподчиненность* этих понятий: параметр всегда выступает как средство описания воздействия того или иного фактора. С одной стороны, ветер, без сомнения, является важным фактором того, какая стоит погода. Этот фактор обычно описывается двумя параметрами: скоростью (или силой) и направлением. С другой стороны, если мы говорим о погоде как факторе, влияющем на уборку урожая, то ветер будет выступать одним из параметров, характеризующих влияние погоды.

Следует обратить внимание учащихся на то, что выбор параметров для описания фактора зависит от целей построения модели (или каких-то других ее характеристик). Например, из курса географии учащимся хорошо известно, что существенными факторами, влияющими на климат, являются тепло и влага. Первый из этих факторов может описываться среднегодовой температурой, второй — среднегодовым уровнем осадков. Этих параметров достаточно, чтобы описать основные климатические пояса. Однако если в модели климатического строения мы хотим выделить такие характеристики, как, например, будет ли климат резко континентальным, то фактор тепла будет описываться разницей средних зимних и летних температур. Можно вообще действие фактора тепла описывать набором (т. е. вектором) среднемесячных температур. На самом деле при прогнозировании погоды используется вектор ежесуточных температур, усредненных для данной местности по разным годам. Обсуждение с учащимися подобных примеров позволяет им лучше уяснить суть понятий «фактор» и «параметр».

Уместно отметить также, что получение связей между выделенными параметрами представляет собой чисто научную задачу, которая обычно решается постановкой серии экспериментов. Что же касается выделения существенных факторов, то это, как правило, производится на основе целевых установок: они определяются тем, для каких целей будет использоваться строящаяся модель. Описание факторов набором параметров занимает промежуточное положение: с одной стороны, имеется некоторый произвол в выборе этих параметров, с другой — должна обеспечиваться определенная полнота (степень которой тоже может служить предметом научного исследования). В этой главе основное внимание сосредоточено на выделении существенных факторов и описании их системой параметров. Во второй главе добавится и третий компонент — обработка результатов эксперимента с целью установления зависимостей между параметрами модели. В последующих главах учащимся предстоит решать задачи по построению моделей во всей полноте.

Для закрепления и/или проверки того, насколько учащиеся освоили этапы построения информационной модели, полезно, на наш взгляд, предложить следующее задание:

Задание 2. Ниже приведено несколько задач. Для решения каждой из них требуется построить информационную модель описываемой ситуации. Попытайтесь построить такие модели, а если это не получится, то укажите, какой именно этап построения модели нельзя осуществить и почему.

а) Семья, состоящая из дедки, бабки, внучки, Жучки и кошки, взяв землю в аренду, решила выращивать репу. Потребуется ли привлечение сезонного рабочего (мышки) для сбора урожая?

б) Винни Пух и Пятачок построили ловушку для Слонопотама. Удастся ли его поймать?

в) Винни Пух и Пятачок пошли в гости к Кролику. Сколько бутербродов с медом можно съесть Винни Пуху, чтобы не застрять в двери?

г) Малыш и Карлсон решили по-братски разделить два сладких орешка — большой и маленький. Как это сделать?

д) Аббат Фариа решил бежать из замка Иф. Сколько времени ему понадобится, чтобы осуществить свой замысел?

При выполнении этого задания нужно для задачи из каждого пункта указать либо на отсутствие перечисления в ее условии, какие факторы существенны, а какие нет, либо на невозможность описать существенные факторы системой параметров (иными словами, неформализуемость задачи в рамках выбранной совокупности существенных факторов), либо на невыявленность связей между параметрами, полученными в результате формализации. В некотором смысле самым простым является обоснование того, что в условии задачи нет описания существенных факторов. Для такого обоснования достаточно предъявить две различные, разумные с целевой точки зрения совокупности существенных факторов, приводящих к разным результатам. Например, в пункте *г* допустимы два ответа: 1) большой орех берет Карлсон, маленький — Малыш; существенный фактор — размер потребителя: кто больше (по размеру) или старше, тому больший орех; 2) каждый орех разрезается пополам и каждому дается по половине ореха; существенный фактор — равенство получаемого продукта. Пункт *б* предоставляет пример для второго варианта обсуждения. Совершенно ясно, что в этой задаче существенным фактором является существование Слонопотама (иначе кого ловить?). Однако описать этот фактор какой-либо совокупностью параметров невозможно, поскольку неизвестны признаки существа, которое называется Слонопотамом. Наконец, пункт *в* можно рассматривать как проявление третьего варианта. Действительно, понятно, что существенными факторами здесь являются увеличение «диаметра» Винни Пуха от съеденных бутербродов и неизменность размеров входа в нору Кролика. Для обоих факторов легко указать набор нужных параметров: для первого это число съеденных бутербродов, их объем, «диаметр» Винни Пуха; для второго фактора можно обойтись одним параметром — размером входа в нору. Однако невозможно указать связь между суммарным объемом съеденных бутербродов и увеличением «диаметра» Винни Пуха. Аналогично разбираются остальные две задачи этого задания. Мы убеждены, что шутливый тон задания не помеха для понимания существа дела. Разумеется, учитель вправе подобрать свои примеры задач, позволяющие провести данное обсуждение. На усмотрение учителя мы оставляем и то, как использовать данное задание: в открытом обсуждении в конце первого урока или в начале второго урока по данной теме как проверочную самостоятельную работу.

Адекватность модели — это, без сомнения, один из центральных вопросов всего курса информатики в общеобразовательной школе. Понимание того, что каждая модель ограничена областью своей адекватности, носит без преувеличения мировоззренческий характер, поскольку, с одной стороны, пресекает проявления догматизма (основанного на вере, что та или иная модель пригодна везде и при любых обстоятельствах), с другой стороны, направлено против философского релятивизма, фактически утверждающего, что область применимости любой модели настолько узка, что вообще не имеет смысла говорить о каком-либо знании.

Обнаружение неадекватности имеющейся модели дает толчок к появлению новой, более совершенной модели и тем самым служит причиной прогресса научного знания. Поэтому при изучении данной темы особенно уместны яркие примеры из истории науки. Вот один из примеров.

Вопросы космического устройства мира всегда волновали человека. Космос притягивает внимание человека не только своей романтической безграничностью и загадочностью, но и чисто практическими проблемами: как происходит смена времен года, когда ожидать солнечные и лунные затмения, как по Солнцу и звездам определить свое местонахождение и т. д. Для всего этого необходимо было иметь модель пусть не всего космоса, но хотя бы того ближайшего к Земле окружения, которое сейчас именуется Солнечной системой. Оставляя в стороне представления древних народов о Земле, покоящейся на слонах и черепахах, поговорим о планетарных моделях.

Одной из самых известных была модель, предложенная во II в. н. э. древнегреческим астрономом и математиком

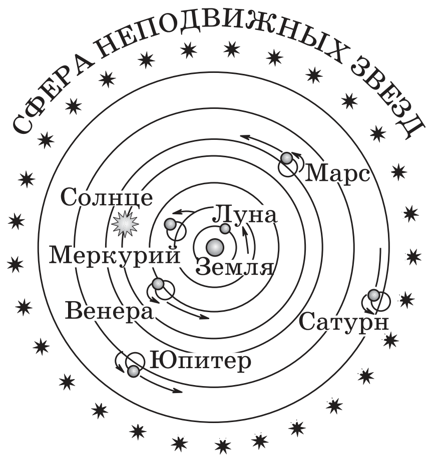

Рис. 4. Система Птолемея

Клавдием Птолемеем, продержавшаяся в истории науки почти 14 столетий. В этой модели в центре располагалась неподвижная Земля, а все планеты (открытые к тому времени) и Солнце вращались вокруг нее по круговым орбитам. Точнее, по круговым орбитам вращались вокруг Земли только Луна и Солнце, а для других планет движение было более сложным — ведь при круговом движении планет их перемещение среди звезд должно быть, как у Солнца или Луны, только «вперед». А наблюдения показывали так называемое «попятное» движение планет.

Рис. 5. Система Коперника

Для объяснения такого движения Птолемей предположил, что каждая планета вращается по некоторой окружности, центр которой как раз и движется по круговой орбите вокруг Земли (рис. 4). Птолемею удалось так подобрать радиусы всех фигурирующих в этой модели окружностей, что она прекрасно согласовывалась с астрономической практикой вплоть до XVI в.

Когда Николай Коперник в 1543 г. предложил гелиоцентрическую модель (рис. 5), то он мотивировал это прежде всего тем, что в ней проще производить астрономические вычисления. Компьютеров в то время не было, и простота вычислений — важный аргумент в борьбе моделей.

Следующий шаг в истории моделей Солнечной системы был сделан Иоганном Кеплером. Сначала И. Кеплер, опираясь на наблюдения датского астронома Тихо Браге, в 1609 г. формулирует первые два своих закона движения небесных тел, второй из которых как раз и гласит, что орбита такого тела представляет собой коническое сечение (т. е. эллипс, параболу или гиперболу), в одном из фокусов которого находится Солнце. Спустя 10 лет И. Кеплер сформулировал третий закон движения небесных тел.

Прошло еще почти 70 лет, и в 1687 г. Исаак Ньютон в своем сочинении «Philosophiae naturalis principia mathematica» на основании трех законов Кеплера выводит формулу для силы притяжения планеты Солнцем и затем, используя эту формулу, приходит к формулировке знаменитого закона всемирного тяготения. Но после того как этот закон (справедливый отнюдь не только для космических тел) был открыт, он стал основой всей небесной механики и теоретической астрономии. Теперь уже из закона всемирного тяготения стали выводить законы Кеплера. И что же оказалось? Выяснилось, что сами законы Кеплера, сыгравшие принципиальную роль в выводе формулы великого закона, справедливы только в том случае, если вокруг Солнца вращается ровно одна планета.

Эта история показывает, что эмпирически построенная модель (в данном случае — три закона Кеплера), сомневаться в адекватности которой нет никаких оснований (ибо родилась она из практики астрономических наблюдений), способна после теоретической «обработки» породить модель, которая однозначно укажет на неадекватность исходной модели или, точнее говоря, резко сузит область ее адекватности.

А что же дальше? В начале XX в. была сформулирована корпускулярно-волновая теория элементарных частиц, согласно которой свет — это не только волна, но и частица. Но тогда в соответствии с законом всемирного тяготения частицы света (их назвали фотонами), проходя мимо космических тел, должны к ним притягиваться. Это значит, что траектория их движения вблизи таких тел должна заметно искривляться. Астрономические наблюдения во время солнечных затмений этот вывод подтвердили.

Однако всеми давно признано, что луч света всегда движется по пути, кратчайшему между точками. Что же получается: прямая не является кратчайшим путем между двумя точками! Но так ли это удивительно? Ведь, к примеру, на поверхности шара кратчайший путь тоже не отрезок между точками, а дуга окружности большого круга. Там по прямой нам мешает двигаться вещество шара. А в пространстве движению мешает сила притяжения между материальными телами, открытая еще Ньютоном. Получается, что геометрия пространства (т. е. вдоль каких линий надо измерять расстояние между двумя точками) зависит от распределения в этом пространстве материальных масс. Эта новая модель пространства, отличная от привычной нам геометрии Евклида, была построена А. Эйнштейном и получила название общей теории относительности. Закон всемирного тяготения, выведенный Ньютоном в предположении, что наше пространство евклидово (неевклидовой геометрии во времена Ньютона просто не было!), привел к появлению модели, в которой геометрия пространства неевклидова.

Мы надеемся, что учитель легко расширит спектр обсуждаемых примеров. Полезно предложить учащимся подготовить сообщения на тему смены научных моделей в связи с проблемой адекватности. Большое поле деятельности здесь представляют физика (например, модели строения атома Томпсона и Резерфорда), химия (теория флогистона), биология (теория эволюции) и др.

Вопрос об адекватности модели напрямую связан с практикой ее использования. При этом термин «практика», как показано в объяснительном тексте, должен пониматься достаточно широко. В первую очередь это применение в реальной деятельности. Здесь уместно еще раз вспомнить и обсудить тезис, что модели нередко возникают в ходе решения жизненных задач и, следовательно, их применимость и востребованность определяется тем, насколько успешным оказалось решение жизненной задачи.

Но, кроме того, адекватность подразумевает согласованность данной модели с другими моделями, образующими *систему* знаний в данной предметной области. Впрочем, построению моделей с более широкой областью адекватности могут препятствовать и другие факторы, нежели ограниченность научной теории и реальной практики. О них в учебнике лишь вскользь упоминается, поэтому мы считаем необходимым более подробно остановиться на этом в комментариях для учителя.

Напомним уже отмечавшийся факт, что создание модели должно учитывать особенности того инструментария, который будет применяться при использовании данной модели. До середины XX в. вычислительные средства были чрезвычайно слабыми, поэтому предпочтение отдавалось моделям, записываемым в аналитической форме (в виде уравнений, неравенств и т. п.). С развитием вычислительной техники ограничения на объем вычислений стали ослабляться, и на первый план стали выходить модели, описываемые дискретными средствами. Например, вместо формулы, выражающей зависимость разыскиваемой величины от исходных величин, интересовались процессом вычисления (как правило, в виде алгоритма) этой величины по исходным данным. При этом отпала потребность в предварительном выводе аналитического соотношения. Нередко построение нужного вычислительного процесса может производиться при более слабых предположениях, чем получение аналитической формулы. А это значит, что модель, выраженная таким вычислительным процессом, имеет более широкую область адекватности, чем аналитическая модель.

Второй фактор, влияющий на расширение области адекватности модели, относится больше к классу информационных моделей. Мы уже упоминали, что для построения информационной модели требуется подходящий язык описания параметров этой модели и связей между ними. Если информационная модель является математической, то требуется язык соответствующего раздела математики. Например, если равномерное прямолинейное движение описывается линейной зависимостью, то прямолинейное равноускоренное движение для описания зависимости пути от времени требует знания квадратичной функции. Произвольное движение по прямой уже описывается обыкновенным дифференциальным уравнением второго порядка. Для описания движения точки на плоскости или в пространстве требуется векторный анализ. Каждая последующая модель включает в себя как частный случай предыдущую, но для ее существования требуется все более развитый язык, в данном случае язык математики. Итак, второй фактор — уровень развития языка описания модели.

В какой мере учитель может углубиться в обсуждение этих вопросов, зависит от уровня класса и тех задач, которые он ставит перед ним. Мы здесь только наметили возможные пути развития данной темы. Минимальное требование состоит в том, что учащиеся должны понимать наличие четырех факторов, определяющих смену модели: возникновение противоречия с практикой в виде реальной деятельности человека, возникновение противоречия с более общей (и, как правило, более системной) теорией, появление более совершенного языка описания модели и появление более мощных средств реализации модели.

Задания 1—8 к § 5 затруднений не вызывают — в своей основе они репродуктивны. В ходе обсуждения полезно попросить учащихся привести примеры моделей, которые встречались им при изучении физики, химии, биологии, географии, литературы. При этом, естественно, они должны указывать, является ли названная ими модель информационной.

В задании 9 учащиеся должны указать три фактора: начальное состояние тела, вертикальное направление движения и наличие притяжения Земли. А вот сопротивление воздуха существенным фактором в этой модели не признается. Начальное состояние описано двумя параметрами: исходной высотой h_0 расположения тела над Землей и начальной скоростью движения v_0. Фактор вертикальности учитывается, но не описывается никаким параметром (можно сказать, что он описывается отсутствием параметров горизонтального перемещения тела). Наконец, фактор притяжения Земли описывается параметром g.

Задание 10 можно задать домой. Рассматривая паспорт (не только свой, но и родителей, посоветуйте им так сделать), учащиеся могут выделить следующие факторы: наименование человека, его возраст, место жительства, семейное положение. Наименование описывается тремя параметрами: фамилией, именем и отчеством. Возраст описывается тремя параметрами: числом, месяцем и годом рождения. Место жительства описывается значительно большим числом параметров: наименование субъекта Российской Федерации (область, край, автономная республика), наименование населенного пункта (город, деревня, поселок и т. п.), наименование улицы, номер дома, номер корпуса, номер квартиры. Какие-то из этих параметров могут отсутствовать. Семейное положение описывается двумя параметрами: один из них определяет, состоит ли человек в зарегистрированном браке, другой представляет собой довольно сложный конгломерат из наименования детей и дат их рождения.

В § 6 начинается чрезвычайно важный в методологическом отношении концентр изучения материала — понятие системы и системного подхода. Начать можно с рассмотрения примеров системных и несистемных моделей, чтобы подвести учащихся к формулировке определения системы. Важно при этом подчеркнуть, что система обладает таким неформализуемым, вообще говоря, свойством, как *целостность*. Иными словами, совокупность элементов и связей между ними мы считаем системой только в том случае, если исключение из рассмотрения каких-либо связей или элементов данной совокупности приводит к исчезновению у нее возможности выполнять свои функции как единого целого.

Графы являются одним из наиболее употребительных средств для описания систем. В учебнике 10 класса им уделено скромное внимание, при желании учитель может расширить спектр примеров использования графов для решения тех или иных задач информатики. Он может это сделать сам или предложить учащимся подготовить соответствующие сообщения на уроках.

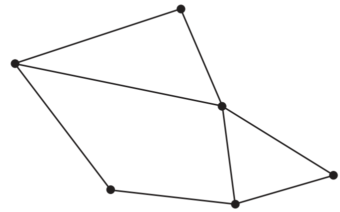

Рис. 6. Граф как модель разных объектов

Задания к § 6 обычно не вызывают трудностей у учащихся. В связи с заданием 7 можно обсудить следующий вопрос: могут ли разные явления описываться одной и той же моделью? Именно для графов легко привести примеры, когда один и тот же граф служит моделью для совершенно различных явлений. К примеру, следующий граф (рис. 6) может показывать, кто с кем знаком в данной компании из 6 человек, а может указывать имеющиеся авиамаршруты между населенными пунктами.

Другой пример, когда разные явления описываются одной и той же моделью — прямо пропорциональной зависимостью. Ею описывается связь пройденного пути, скорости и времени, и такой же прямо пропорциональной зависимостью описывается связь между силой натяжения, удлинением и коэффициентом упругости (закон Гука). Учащиеся должны осознать, что суть не в буквах, которыми обозначены переменные при записи указанных соотношений, а в одинаковости числа параметров и характере взаимосвязей между ними. Вообще, прямо пропорциональная зависимость моделирует такой широкий круг явлений, что можно устроить конкурс, кто больше назовет процессов и явлений, описываемых этой моделью.

Из определения информационной модели вытекает, что она всегда системна. Полезно, на наш взгляд, поставить перед учащимися следующий вопрос: можно ли утверждать, что всякая системная модель является информационной? К числу системных, но не информационных моделей относятся многие натурные модели, например модель автомобиля или самолета, собранная из отдельных деталей, муляж человеческого скелета в кабинете биологии и т. п. То, что каждая такая модель системна, сомнений не вызывает; то, что она не является информационной, тоже ясно — в таких моделях нет никаких параметров, которые описывали бы действие тех или иных факторов. При желании и наличии возможности здесь было бы уместно обсудить еще одно свойство информационных моделей — атрибутивно присущую им знаковую форму представления.

В задании 9 учащиеся должны указать не только длину маршрута, но и сам маршрут в виде последовательности проходимых вершин графа. Для этого вершины графа надо обозначить, например, буквами.

Сопровождающие эту тему две лабораторные работы имеют двоякое назначение. Для базового уровня — восстановление навыков работы с электронной таблицей и текстовым и графическим редакторами (на фоне построения модели соответствующего типа). Поэтому в тематическом планировании для базового уровня им отведено 2 часа. В рамках профильного уровня в лабораторной работе 1 учащиеся осваивают работу с условным оператором, а также построение более сложных математических моделей, возникающих при увеличении числа существенных факторов и соответственно параметров модели. В лабораторной работе 2 на втором часу предусмотрена самостоятельная проектная деятельность по созданию собственных вариантов визитной карточки. Здесь весьма важно проследить прохождение учащимися (быть может, впервые) всей компетентностной цепочки — от исполнительской компетентности, которую они должны были продемонстрировать при создании первой визитной карточки, через технологическую компетентность, проявляющуюся в разработке проекта второй визитной карточки, до экспертной и аналитико-синтезирующей компетентностей. Для этого мы предлагаем разделить класс на небольшие экспертные группы, которые оценят созданные визитные карточки. Предварительно каждая экспертная группа должна разработать критерии оценки (проект критериев учащиеся разрабатывают дома и затем согласовывают их в своей группе). Надо понимать, что разработка критериев — это первый шаг и в аналитико-синтезирующей деятельности, ибо разработка продукта с опорой на определенную критериальную базу как раз и составляет основу аналитико-синтезирующей компетентности. После оценивания созданных визитных карточек надо предоставить учащимся возможность их модернизации.

Тема 3. Для базового уровня в данной теме предполагается только повторение материала курса информатики 8—9 классов в объеме § 7. Учащиеся должны понимать следующие основные положения, относящиеся к алгоритмизации:

— компьютер является формальным исполнителем;
— для использования формального исполнителя требуется

программа, записанная на формальном языке, понятном исполнителю;
- основу программы составляет алгоритм, который представляет собой организованную последовательность действий, допустимых для данного исполнителя, которая направлена на достижение поставленной цели (решения задачи);
- для организации действий в алгоритме применяются конструкции ветвления (в полной и неполной формах), цикла (в формах **Делать пока** и со счетчиком) и выделения вспомогательного алгоритма; последняя используется, в частности, для организации рекурсии;
- запись алгоритма представляет собой последовательность операторов; каждый оператор обозначает либо допустимое действие исполнителя (простой оператор), либо алгоритмическую конструкцию указанного выше типа (составной оператор).

Что касается структур данных, то в этой теме речь идет только о переменных. Переменная, как принято говорить, является атомарной структурной единицей, предназначенной для хранения информации. Учащиеся должны понимать, что за понятием переменной стоят по крайней мере четыре сущности:
- область памяти компьютера, где будет храниться информация;
- имя переменной для указания компьютеру, откуда брать эту информацию;
- значение переменной, т. е. собственно та информация, которая сохраняется посредством данной переменной;
- тип переменной, показывающий, какие действия допустимо совершать над той информацией, которая хранится посредством данной переменной.

Конечно, когда мы обсуждаем составление алгоритмов, можно не акцентировать внимания на физической составляющей этого понятия как области памяти компьютера. Обычно так и поступают, говоря, что каждая переменная характеризуется своим именем, типом и значением. При таком подходе переменную удобно представлять себе как некую емкость (например, ящик), на которую наклеена этикетка с именем и в которую можно «положить» информацию, причем материал, из которого изготовлена данная емкость, определяет то, какого вида информация может быть туда помещена.

Определенную трудность при изучении обычно представляет освоение учащимися операции присваивания. Аналогия между переменной и ящиком условна уже в том смысле, что при заполнении ящика его старое содержимое не исчезает, а операция присваивания безжалостно уничтожает то, что до того хранилось в переменной. Нередко именно утраченные значения переменной или, наоборот, необновленные являются источником ошибок в алгоритмах.

Тем самым, освоив понятие переменной, учащиеся должны знать:

— каждая переменная характеризуется своим типом; в учебнике рассматриваются целый и вещественный числовые типы, символьный (строковый) тип и логический (булевый) тип; тип переменной объявляется специальным оператором (в некоторых языках программирования принимается по умолчанию);
— каждая переменная имеет имя, позволяющее отличать одну переменную от другой и использовать ее в записи операторов;
— каждая переменная имеет значение, соответствующее типу переменной, свое значение переменная получает с помощью оператора присваивания.

Заключительная часть § 7 посвящена так называемым свойствам алгоритмов. Методологической подоплекой появления понятия «свойства алгоритмов» является то обстоятельство, что сам алгоритм — понятие неопределяемое[1]. Аналогия с другими математическими теориями (например, геометрией), где существование неопределяемых понятий заложено в основу соответствующей теории, а сами понятия описываются своими свойствами и связывающими их соотношениями — аксиомами, подсказывает идею описать понятие алгоритма через его свойства. Одно из направлений классической теории алгоритмов развивает именно такую точку зрения. Она легла в основу большинства школьных учебников информатики (и нашла свое отражение в стандарте). Если учитель намерен рассмотреть с учащимися свойства алгоритмов, то им можно предложить воспользоваться этими свойствами, чтобы выполнить следующее задание:

Задание 3. Объясните, почему инструкции, записанные в пунктах *а* и *б* данного задания, не являются алгоритмами.

а) Фрагмент инструкции по приготовлению теста для торта:

...
Добавьте щепотку соли.
Слегка потрясите, чтобы смесь стала комковатой.

[1] См.: Математическая энциклопедия / Под ред. И. М. Виноградова. — М.: Советская энциклопедия, 1977. — Т. 1. — С. 205. При применении понятия «алгоритм» в конкретной ситуации оно соответствующим образом уточняется. С этой целью в понятии алгоритма выделяется 7 параметров, и в зависимости от того, как формализуются эти параметры, получается то или иное уточненное понятие алгоритма. В учебнике 10 класса мы не рассматриваем эти параметры, хотя некоторые уточнения понятия алгоритма фактически обсуждаются в § 8, 9 и 14.

Подогрейте коньяк в маленькой кастрюльке и влейте его в смесь.

...

б) Инструкция по употреблению лекарства от кашля:

Если врач не прописал иначе, то 3—4 раза в день по 15—20 капель, лучше всего в горячей сладкой воде.

В первой инструкции отсутствует свойство детерминированности: ведь исполнителю неясно, требуется ли трясти смесь, пока она вся не возьмется комом, и какой все-таки величины маленькая кастрюлька. Так что это наверняка не алгоритм.

Во второй инструкции не определено, когда должен заканчиваться алгоритм — когда кашель пройдет или когда лекарство кончится? Иными словами, нарушено и свойство детерминированности, и свойство результативности.

Это задание полезно и еще с одной точки зрения. Учащиеся нередко путают или отождествляют между собой свойства детерминированности и результативности. Как отмечено выше, в пункте *а* отсутствует детерминированность, хотя результативность имеется. Приведем пример обратной ситуации — детерминированность есть, но отсутствует результативность. Пусть в алгоритме встретился оператор, в котором фигурирует деление, например:

$a := b/c$.

Исполнителю абсолютно ясно, что нужно делать, т. е. детерминированность присутствует. Однако если c окажется равным 0, то получить результат будет невозможно, т. е. отсутствует результативность. Чтобы алгоритм с таким оператором обладал свойством результативности, нужно до применения этого оператора предусмотреть обработку ситуации, когда $c = 0$. Этот пример показывает, в частности, что отсутствие результативности в составляемых алгоритмах встречается гораздо чаще, чем недетерминированность.

Несмотря на вышесказанное, мы хотим предостеречь учителя от чрезмерного внимания к изучению свойств алгоритмов. Развитие компьютерной техники, теории алгоритмов и практики программирования вносит свои коррективы. Теперь, к примеру, рассматривается не только последовательное, но и параллельное исполнение шагов алгоритма (сравните с трактовкой свойства дискретности). Алгоритм не обязан быть детерминированным — на очередном шаге может осуществляться случайный выбор из некоторого множества возможных результатов. Теория недетерминированных алгоритмов — современная бурно развивающаяся ветвь общей теории алгоритмов. При этом многие недетерминированные алгоритмы приводят тем не менее к однозначно определенному результату. Такие алгоритмы называют *однозначными*.

Конечность алгоритма — это тоже свойство, присущее далеко не всем алгоритмам, используемым на практике. Почти все алгоритмы, применяемые для управления объектами в режиме реального времени, не являются конечными: трудно представить себе конечный алгоритм (т. е. прекращающий работу через конечное число шагов) компьютерного управления полетом космической станции или работой ядерного реактора. Конечно, такие алгоритмы, как правило, включают в себя режим ожидания вмешательства человека (нажатия той или иной клавиши, щелчка мыши и т. п.), но по своему устройству такой алгоритм не является конечным[1]. Впрочем, бесконечно исполняемые алгоритмы играют важную роль в общей теории алгоритмов. Вот пример такого алгоритма, приведенный в Математической энциклопедии[2]. Исполнитель обрабатывает слова, записываемые в алфавите из двух букв a и b. Допустимые действия исполнителя — это переход от слова X к слову Y в одном из двух случаев:

а) если X имеет вид aP, где P — любое слово над данным алфавитом, то $Y = Pb$;

б) если X имеет вид baP, где P — любое слово над данным алфавитом, то $Y = Paba$.

Сам алгоритм выглядит так:

Запросить слово X;

$Y := X$;

Делать пока не получится слово Y, начинающееся буквосочетанием aa

{ Применить к слову Y допустимое действие;

Результат обозначить снова Y;

}

Сообщить Y;

Применим этот алгоритм к слову $babaa$. Цикл должен исполняться, поскольку выполнено условие продолжения цикла. К самому этому слову применимо только второе допустимое действие. Поэтому после первого исполнения тела цикла получаем $baaaba$. Условие продолжения цикла снова выполнено, так что при очередном его исполнении получим $aabaaba$. Это и будет результатом в данном алгоритме.

[1] В научных кругах обсуждается вопрос о том, чтобы подобные алгоритмы лишить «звания» алгоритма и называть их операционной обстановкой. Однако единой точки зрения на этот счет пока нет, и решение не принято.

[2] Математическая энциклопедия издана в 1977 г. Уже тогда было общепризнано, что алгоритмы не обязаны обладать свойством конечности. Тем не менее в школьные учебники по информатике (которая появилась в школе лишь в 1985 г.) это свойство вошло, и на экзаменах у выпускников и абитуриентов о нем спрашивают как об обязательном свойстве алгоритмов.

Однако если применить этот алгоритм к слову *baaba*, то алгоритм будет исполняться бесконечно.

Фактические свойства алгоритмов играют такую же роль, как, например, свойства функций: функция может быть монотонной, а может и не быть; может быть непрерывной, а может и не быть. Если функция монотонна — это хорошо, если непрерывна — еще лучше, но если и не такая, то в этом тоже нет никакой трагедии. Так и алгоритмы: если они обладают указанными свойствами — хорошо, а не обладают (разумеется, на законных основаниях) — печалиться не следует. В какой мере обсуждать данные вопросы с учениками — решать учителю; одно только можно утверждать уверенно: все алгоритмы, составляемые учащимися в ходе изучения курса информатики по нашему учебнику, должны обладать всеми перечисленными в учебнике свойствами (хотя в нем присутствуют и примеры алгоритмов, не обладающих ими).

Обсудим задания к § 7. Ответы на три первых вопроса содержатся в объяснительном тексте. На четвертый вопрос ответ таков: алгоритмы появились раньше письменности. Добывание огня, изготовление простейших орудий, освоение приемов охоты — это все примеры алгоритмов в дописьменную эпоху. Вопросы 5—16 требуют от учащегося воспроизведения знаний теории.

Задание 17 преследует цель продемонстрировать учащимся, что в своей учебной деятельности они уже неоднократно встречались с алгоритмами. Здесь учащиеся могут назвать разные примеры алгоритмов, мы приведем наиболее часто встречающиеся в ответах учеников:

а) на уроках русского языка — алгоритм разбора предложения по членам предложения; алгоритмы проверки безударной гласной в корне; алгоритм морфологического разбора слова;

б) на уроках иностранного (в данном случае английского) языка — алгоритм перехода от утвердительного предложения к вопросительному; алгоритм перехода от Present Indefinite к Present Continuous;

в) на уроках математики — алгоритмы сложения и умножения натуральных чисел столбиком; алгоритм сравнения натуральных чисел; алгоритм сложения дробей; алгоритм приведения подобных членов в многочлене;

г) на уроках литературы — алгоритмы заучивания стихотворений (возможно, у каждого свой); алгоритм написания изложения (или сочинения);

д) на уроках химии — алгоритмы проведения химических опытов (например, титрования раствора, получения реакции медного или серебряного зеркала и т. п.); алгоритм нахождения весовых соотношений для веществ, участвующих в данной реакции;

е) на уроках физкультуры — алгоритм выполнения того или иного комплекса физических упражнений;

ж) на уроках информатики — алгоритмы выполнения работ с помощью тех или иных компьютерных технологий.

Задания 18 и 19 проверяют навыки исполнения алгоритмов. Учащиеся обычно легко распознают, что первый из этих алгоритмов предназначен для нахождения наибольшего числа из трех данных чисел, а второй — для построения серединного перпендикуляра к отрезку.

Алгоритмы, приведенные в задании 20, хотя и несложные, многими учащимися выполняются с ошибками. Источников ошибок два: неправильное понимание условия продолжения исполнения тела цикла и исполнение операции присваивания не с теми значениями переменных, которые имеются к моменту исполнения данного оператора.

Ответ в этом задании учащиеся могут получить посредством составления протокола исполнения приведенных там алгоритмов. Но более эффективным способом является определение значений параметров, при которых нарушается условие продолжения цикла. Так, в задании 20а легко заметить, что значение переменной a все время отрицательно, а значение переменной b увеличивается. Пока b неположительно, левая часть в условии продолжения цикла отрицательна, а правая неотрицательна, так что указанное условие выполняется. При $b = 1$ условие выглядит как $a + 1 < a$ и потому ложно. Поэтому при $b = 1$ тело цикла исполняться не будет. Поэтому ответ в задании 20а таков: $a = -40$; $b = 1$; тело цикла исполняется 3 раза.

В задании 20б ясно, что условие продолжения цикла не выполняется уже в первый раз. Ответ: $a = 25$; $b = 1$; тело цикла не исполняется ни разу.

В задании 20в мы советуем просто составить протокол. Ниже в таблице 2 мы приводим значения a и b после каждого исполнения тела цикла.

Ответ: $a = 3$; $b = 0$; тело цикла исполняется 5 раз.

Таблица 2

Номер исполнения	a	b
1	–2	1
2	–3	3
3	–1	4
4	1	3
5	3	0

В задании 21 ответы таковы: а) 6 раз; б) 4 раза; в) 5 раз; г) 1 раз.

Задание 22а проверяет умение учащихся составить протокол исполнения алгоритма. Задания 22б и 22в мы рекомендуем предлагать только для учащихся, изучающих информатику на профильном уровне. Систематически материал, относящийся к доказательству свойств алгоритмов, будет рассматриваться в 11 классе, поэтому здесь это задание носит проблемный характер и, по существу, лишь демонстрирует ученикам горизонты так называемого доказательного программирования.

Для получения требуемого в пункте *б* обоснования поступим так: после каждого исполнения тела цикла значения переменных m и n будем обозначать теми же буквами с индексом, показывающим количество выполненных исполнений тела цикла. Значит, в начале $m = m_0$ и $n = n_0$ (тело цикла не исполнялось ни разу); после первого исполнения тела цикла получим $m = m_1$ и $n = n_1$; после второго исполнения получим $m = m_2$ и $n = n_2$; после третьего — $m = m_3$ и $n = n_3$ и т. д. Пусть для получения результата потребовалось выполнить тело цикла k раз; тогда в результате имеем $m = m_k$ и $n = n_k$. Поскольку условием окончания цикла яляется $m = n$, получаем $m_k = n_k$. Учитывая сформулированный в задании совет, получаем следующую цепочку равенств:

$$\text{НОД}(m,\ n) = \text{НОД}(m_0,\ n_0) = \text{НОД}(m_1,\ n_1) = \text{НОД}(m_2,\ n_2) = \\ = \ldots = \text{НОД}(m_k,\ n_k) = \text{НОД}(m_k,\ m_k) = m_k.$$

Она показывает, что результатом исполнения алгоритма является НОД(m, n).

Отметим, что в доказательстве был применен важный прием — рассмотрение некоторой характеристики, которая не меняется при исполнении алгоритма. В данном случае такой характеристикой выступал сам разыскиваемый наибольший общий делитель двух чисел — при выполнении алгоритма величина НОД изменяющихся переменных m и n оставалась постоянной. Такую характеристику, которая не меняется при выполнении тех или иных действий, называют **инвариантом**.

Доказательство конечности алгоритма опирается на другое важное в теории алгоритмов понятие — понятие **лимитирующей функции**[1]. Так называют функцию, которая в ходе исполнения алгоритма может принимать лишь конечное число значений, причем каждое из них она принимает ровно один раз. Уже из этого определения ясно, что в исполнении алгоритма, обладающего лимитирующей функцией, число шагов ограничено количеством значений этой функции. Для алгоритма Евклида в качестве лимитирующей функции можно взять переменную $t = m + n$. Действительно, после каждого исполнения тела цикла значение этой функции уменьшается, но тем не менее всегда остается целым положительным числом. Начальное значение t равно

[1] В математической литературе лимитирующую функцию иногда называют **полуинвариантом**, хотя такое название нам представляется менее удачным, поскольку, во-первых, не отражает сути использования этого понятия, а во-вторых, нередко ассоциируется у школьников с понятием инварианта, к которому в общем-то не имеет никакого отношения.

$m_0 + n_0$. Однако между $m_0 + n_0$ и 0 лишь конечное количество натуральных чисел, поэтому количество возможных исполнений тела цикла, а значит, и шагов в алгоритме Евклида всегда конечно.

Разумеется, учащиеся могут предложить и какую-либо иную лимитирующую функцию, например max (m, n). Рассуждения проводятся аналогично.

Учащиеся должны понимать, что отсутствие свойства конечности алгоритма может наблюдаться только при использовании конструкции цикла в форме **Делать пока**, причем в описании цикла должна фигурировать (явно или неявно) переменная, допускающая бесконечное число значений. Лимитирующая функция — это мощный и практический универсальный инструмент доказательства конечности алгоритма. Отметим, что разбор этого задания впервые позволяет содержательно, а не декларативно обсуждать с учащимися такие свойства алгоритмов, как результативность и конечность.

Задание 23 актуализирует знания учащихся в работе с символьными переменными. Исполнив этот алгоритм для нескольких различных значений запрашиваемой переменной, учащиеся легко устанавливают, что предложенный алгоритм переписывает слово в обратном порядке.

Выделенный в тематическом планировании базового уровня один час на компьютерный практикум мы рекомендуем употребить для выполнения лабораторной работы 3. При остром дефиците времени можно пропустить задание, связанное со сравнением скорости работы двух модификаций алгоритма Евклида. В классе профильного уровня мы считаем такое упражнение крайне важным, поскольку оно показывает, как можно экспериментально оценивать эффективность того или иного алгоритма решения задачи. Мы советуем учителю и в дальнейшем предлагать учащимся подобные эксперименты. В 11 классе мы будем обсуждать теоретические аспекты, связанные с эффективностью алгоритмов, но важно уже сейчас готовить для этого почву.

Следующие два параграфа предназначены только для тех, кто изучает информатику на профильном уровне. В каждом из них фактически уточняется понятие алгоритма для того формального исполнителя, который в нем описывается. Надо понимать, что описывается, конечно, не один исполнитель, а целый класс однотипных исполнителей, зависящих от алфавита, посредством которого задаются команды исполнителю, и от числа возможных состояний исполнителя. Учащиеся должны осознавать, что ни один из обсуждаемых исполнителей не является реальным устройством, а представляет собой некую идеальную (точнее, информационную) модель такого устройства. Именно поэтому такие объекты могут изучаться математическими средствами. Для автомата буквами внешнего алфавита (для простоты он выбран двухсимвольным, хотя на самом деле его можно выбрать любым конечным множеством) кодируются команды, которые последовательно будет исполнять автомат, буквами внутреннего алфавита кодируются состояния. Программа для автомата, как и любого формального исполнителя, — это последовательность команд. Следовательно, она представляется словом в символах внешнего алфавита.

Основное отличие автомата от машины Тьюринга (§ 9) состоит в отсутствии у автомата памяти. Переходя в следующее состояние, автомат «не помнит» полученные ранее результаты. Машина Тьюринга, напротив, обладает памятью (ее роль играет лента), и этого оказывается достаточно, чтобы машина Тьюринга оказалась универсальным исполнителем, т. е. таким (гипотетическим) устройством, с помощью которого можно имитировать любое вычислительное устройство, даже самый сложный современный компьютер.

Из двух вариантов представления автомата (таблицей переходов или орграфом) нам предпочтительным кажется вариант орграфа. Он более нагляден, и с его помощью учащиеся обычно легче определяют результат исполнения программы автоматом.

Как было сказано выше, каждое слово — это программа для автомата, язык, распознаваемый автоматом, — это совокупность всех программ, которые может исполнить данный автомат (причем для них, очевидно, выполняются свойства дискретности, результативности и конечности). Поэтому вопрос о том, любой ли язык распознаваем, фактически означает, верно ли утверждение, что для любого набора программ есть автомат, который может исполнить любую программу из этого набора. Можно доказать, что если язык содержит конечное число слов, то такой язык распознаваем. Зато среди бесконечных языков нераспознаваемые языки встречаются часто. Вот один из таких языков: множество слов вида a^nb^n, когда n пробегает множество всех натуральных чисел (запись a^n означает, что буква a повторена n раз).

В начале объяснительного текста § 8 говорится о роли автоматов и необходимости изучения их формально-математическими средствами. Это мотивационное введение можно усилить, предложив учащимся задачу, которую опубликовал Уильям Росс Эшби — один из пионеров кибернетики — в своей книге «Введение в кибернетику»[1]. Вот эта задача.

[1] Росс Эшби У. Введение в кибернетику. — М.: Иностранная литература, 1959.

Задание 4. Представьте себе, что вы получили письмо, которое приведено ниже. Попытайтесь помочь автору этого письма.

«Замогилье»

Дом с привидениями

Дорогой друг!

Некоторое время назад я купил старый дом, но обнаружил, что он посещается двумя странными звуками: Пением и Смехом. В результате он мало подходит для жилья. Однако я не отчаиваюсь, ибо я установил путем практической проверки, что их поведение подчиняется определенным законам, непонятным, но непререкаемым, и что я могу воздействовать на них, играя на Органе или сжигая Ладан.

В течение каждой минуты каждый из этих звуков либо звучит, либо молчит; никаких переходов они не обнаруживают. Поведение же их в последующую минуту зависит только от событий предыдущей минуты, и зависимость эта такова.

Пение в последующую минуту ведет себя так же, как и в предыдущую (звучит или молчит), если только в эту предыдущую минуту не было игры на Органе при молчащем Смехе. В последнем случае оно меняет свое поведение на противоположное (звучание на молчание и наоборот).

Что касается Смеха, то, если в предыдущую минуту горел Ладан, Смех будет звучать или молчать в зависимости от того, звучало или молчало Пение (так что Смех копирует Пение минутой позже). Если, однако, Ладан не горел, Смех будет делать противоположное тому, что делало Пение.

В ту минуту, когда я пишу Вам это, Смех и Пение оба звучат. Прошу Вас сообщить мне, какие действия с Ладаном и Органом должен я совершить, чтобы установить и поддерживать тишину в доме.

Решение этой задачи надо разбирать с учащимися уже после того, как изучен материал § 8. Вот это решение.

Представим Дом с привидениями в виде автомата. Ясно, что у него есть четыре состояния:

q_1 — звучат Смех и Пение;
q_2 — звучит Смех, но нет Пения;
q_3 — молчит Смех, но звучит Пение;
q_4 — молчат оба.

Воздействий на этот автомат тоже четыре:

a_1 — игра на Органе и сжигание Ладана;
a_2 — игра на Органе, но Ладан не сжигается;

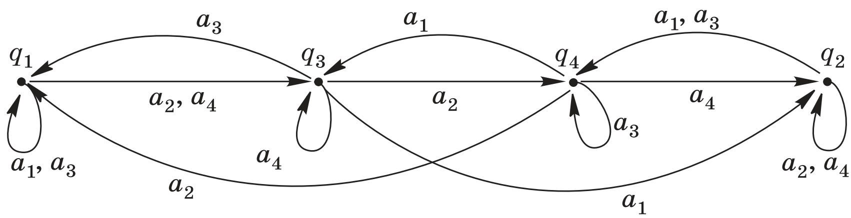

Рис. 7

a_3 — нет игры на Органе, но Ладан сжигается;

a_4 — нет игры на Органе и нет сжигания Ладана.

Читая текст письма, изобразим наш автомат в виде графа (рис. 7).

На этом графе путь от q_1 к q_4 просматривается мгновенно — это, например, a_4a_2. Но чтобы тишина сохранялась, дальше нужно постоянное воздействие a_3. Итак, совет владельцу дома таков:

1) прекратить на 1 минуту игру на Органе и сжигание Ладана;

2) следующую минуту играть на Органе, не сжигая Ладан;

3) со следующей минуты и пока хочется иметь тишину, жечь Ладан, не играя на Органе.

Задачу можно продолжить: ведь неизвестно, в каком состоянии будут находиться Смех и Пение, когда придет это письмо. Возникает вопрос: верно ли, что из любого состояния можно этот автомат перевести в состояние q_4? Положительный ответ легко усматривается, так что можно составить список рекомендаций, как поступать в той или иной ситуации. Но можно поставить такой вопрос: существует ли универсальная последовательность команд — одна на все случаи, которая переводит автомат в состояние q_4? Так сказать, заклинание. Оказывается, существует. Пусть первое действие будет a_4. Оно переведет автомат, в каком бы состоянии тот ни находился, либо в состояние q_2, либо в состояние q_3. Следующим возьмем действие a_2. Оно либо переведет автомат в состояние q_4, либо оставит его в состоянии q_3. Наконец, действие a_3 приведет автомат в состояние q_4. Чтобы оно в нем оставалось, дальше надо все время использовать a_3. Итак, «заклинанием» является слово $a_4a_2a_3$[1].

Перейдем к обсуждению заданий к § 8. Ответы на вопросы 1—3 содержатся в объяснительном тексте.

[1] Слова, переводящие автомат из любого состояния в заданное, называются **синхронизирующими**, а автомат, допускающий синхронизирующее слово, называется **синхронизируемым**. Синхронизируемые автоматы составляют важный класс автоматических устройств, поскольку возможность приведения таких автоматов в некоторое заданное состояние автоматически имеет большое значение.

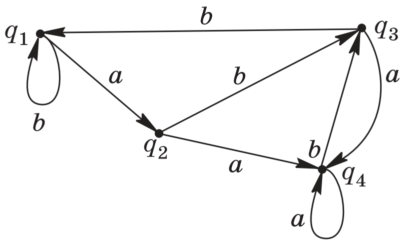

Рис. 8. Автомат из задания 4 к § 8

Задание 4 направлено на отработку понятия «слово воздействует на автомат» (для удобства мы воспроизвели автомат на рисунке 8). В пункте *а* действие буквы a переводит автомат из состояния q_1 в состояние q_2, действие следующей буквы b переводит автомат из состояния q_1 в состояние q_2, следующая буква b переводит автомат из состояния q_2 в состояние q_3, наконец, последняя буква a снова переведет автомат в состояние q_2.

В задании 4*б* аналогично устанавливается, что автомат вернется в состояние q_1. В задании 4*в* автомат перейдет в состояние q_4.

В задании 4*г* ответ зависит от значения n. При $n = 1$ автомат перейдет в состояние q_3. Если же $n \geqslant 2$, то ясно, что блок a^n переведет автомат в состояние q_4, а последующее применение блока b^n переведет автомат в состояние q_1.

При желании (или необходимости) учитель легко расширит спектр подобных примеров. Полезно, например, рассмотреть действие на автомат слов $a^n b a^n$, $(ab)^n$, $b^n a^n$, $(ba)^n$.

В заданиях 5 и 6 учащиеся должны осуществить переход от одной модели представления автомата к другой. Эти задания обычно не вызывают трудностей. Ответы к ним таковы:

Таблица 3

Состояние / Команда	q_1 (K)	q_2
0	q_1	q_2
1	q_2	q_1

Ответ к заданию 5 — таблица 3.

Ответ к заданию 6 — рисунок 9.

У школьников может возникнуть желание самим нарисовать какой-либо автомат и исследовать воздействие на него тем или иным словом. Конечно, такое желание стоит удовлетворить. Надо только тщательно следить, чтобы из каждой вершины выходило столько стрелок, сколько букв во внешнем алфавите.

«Кубик Рубика», «Пятнашки» и другие подобные головоломки представляют собой автоматы: есть некоторое ко-

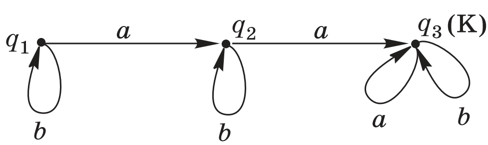

Рис. 9

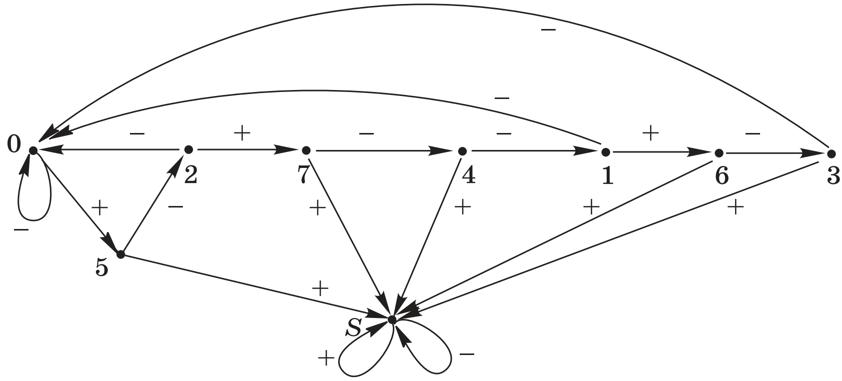

Рис. 10. Автомат «Водокачка»

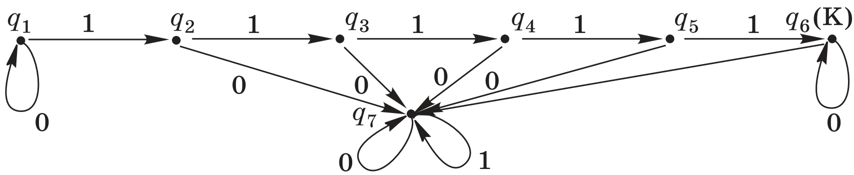

Рис. 11

нечное количество состояний и воздействия, переводящие из одного состояния в другое. Конечно, состояний у каждого такого автомата слишком много, чтобы изобразить его с помощью графа или таблицы, да и воздействий обычно не два, а больше (для «Кубика Рубика» таких воздействий 6 — по числу граней, для «Пятнашек» воздействий 4 — по числу возможных смещений плитки). Иными словами, оба алфавита довольно велики. Поэтому предлагаем рассмотреть с учащимися следующую задачу:

Задание 5. Фирма «Дурацкие игрушки» выпускает игрушку Водокачка. Она представляет собой сосуд, вмещающий 7 л воды, снабженный двумя кнопками. Нажатие одной кнопки приводит к тому, что к имеющемуся в сосуде количеству воды добавляется 5 л. При этом если происходит переполнение сосуда, то он с шумом лопается и выплескивает воду на игрока (после чего игрушка выхо-

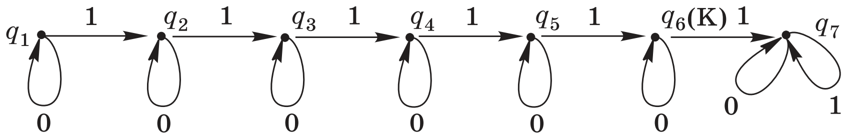

Рис. 12

дит из строя и надо покупать новую). Нажатие второй кнопки приводит к тому, что количество воды в сосуде уменьшается на 3 л, причем если в нем воды было меньше, чем 3 л, то он просто становится пустым. В начальном состоянии сосуд пуст. Постройте информационную модель этой игрушки, рассматривая ее как конечный автомат.

Мы построим модель в форме орграфа. У этого автомата девять состояний, восемь из них удобно обозначить количеством воды, имеющимся в сосуде, а девятое — состояние поломки — обозначим буквой s. Тем самым внутренний алфавит таков: $\{0, 1, 2, 3, 4, 5, 6, 7, s\}$. Внешний алфавит состоит из двух символов, например «+» и «−»: первый обозначает долив в сосуд 5 л, второй — уменьшение воды в сосуде. На рисунке 10 показано, как выглядит требуемый орграф.

Обсуждая с учащимися задание 7, прежде всего стоит заметить, что всякое слово, содержащее ровно две единицы, переводит автомат из начального состояния в конечное. Любое слово, содержащее более двух единиц, переводит автомат из начального состояния в состояние q_4. Если слово содержит одну единицу, то оно переводит автомат в состояние q_2. Наконец, если слово состоит только из нулей, автомат остается в исходном состоянии. Следовательно, язык, распознаваемый данным автоматом, — это множество слов, содержащих ровно две единицы.

Аналогичное рассуждение, проводимое при решении задачи 8, показывает, что язык, распознаваемый заданным там автоматом, состоит из слов, в которых буква a встречается не менее двух раз.

Автомат, который требуется построить в задании 9, представлен на рисунке 11. Труднее всего школьникам дается понимание необходимости такого состояния, в которое будет переходить автомат по всем программам (т. е. словам языка), не удовлетворяющим условию задачи. В автомате, представленном на рисунке 11, таким состоянием является q_7.

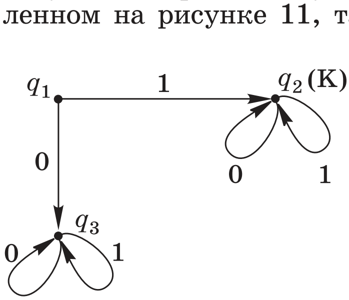

Рис. 13

Автомат, который требуется построить в задании 10, представлен на рисунке 12.

Наконец, автомат, который предлагается построить в задании 11, представлен на рисунке 13.

Автоматы, рассмотренные в объяснительном тексте и заданиях, имели только одно конечное состояние. Определение автомата этого не требует. Нередко бывает удобно иметь несколько конечных состояний. Например, если в автомате, изображенном на рисунке 11, конечными объявить еще и состояния q_2, q_3, q_4, q_5, то такой автомат будет распознавать язык, состоящий из слов, в которых имеется не более пяти единиц, и они идут друг за другом. Полезно задать учащимся вопрос, какой язык будет распознавать автомат, изображенный на рисунке 12, если и в нем объявить дополнительно конечными состояниями q_2, q_3, q_4, q_5. Ответ прост: язык, состоящий из слов, в которых единица встречается не более 5 раз.

Значение материала, изложенного в § 9, многопланово. Прежде всего это фундамент исследований по теории алгоритмов. И первоочередной вопрос — как можно определить, существует ли алгоритм, решающий данную задачу? Конечно, если алгоритм нам предъявлен (например, алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел), то нет сомнений в его существовании. Долгое время математики были убеждены, что для решения любой задачи рано или поздно найдется алгоритм ее решения. Поэтому уже то, что Л. Черч, А. Тьюринг и Э. Пост позволили себе в этом усомниться, нужно рассматривать как достойную восхищения научную смелость и прозорливость[1]. Еще более важным для будущего было другое — понимание ими, что любое доказательство отсутствия алгоритма для решения задачи требует определения понятия «алгоритм». В 30-е гг. XX в., когда этими математиками и логиками закладывались основы теории алгоритмов, речь шла, конечно, об алгоритмах вычисления значений функций. И те функции, для вычисления которых существует алгоритм, Л. Черч назвал вычислимыми функциями. После этого он определил некоторое множество операций над функциями (как бы мы теперь сказали, задал допустимые действия некоторого исполнителя) и описал, какие функции могут получаться, если использовать (в том числе рекурсивно) эти операции. Любые функции, для которых в то время имелись алгоритмы вычислений, попали в это множество функций. Поскольку определения алгоритма не существовало (и как выше отмечалось, это вообще неопределяемое понятие), то Л. Черч высказал гипотезу, что сово-

[1] Впрочем, в те же 30-е гг. К. Гедель доказывает свои знаменитые теоремы о неполноте. Смысл их состоит в том, что существуют истинные, но недоказуемые высказывания (под истинностью высказывания понимается невозможность построить опровергающий пример, а доказуемость высказывания означает наличие цепочки рассуждений, в которой каждое следующее утверждение является непосредственным логическим следствием предыдущих). Наличие доказательства сродни наличию алгоритма, так что идейная перекличка между результатами А. Черча и К. Геделя налицо. Тем не менее результат К. Геделя был настолько шокирующим, что достижения «алгоритмистов» какое-то время оставались в тени.

купность описанных им функций совпадает с классом всех вычислимых функций. Эта гипотеза получила название тезиса Черча и была принята как аксиома. Вот как на самом деле реализован аксиоматический подход в теории — алгоритмом, а вовсе не через весьма сомнительные и эфемерные свойства алгоритмов!

Практически в то же время А. Тьюринг и Э. Пост сформулировали иной подход к определению понятия алгоритма. Каждый из них предложил гипотетическое устройство, которое было способно выполнять несколько очень простых действий. Из этих действий можно составлять инструкции для работы данного гипотетического устройства. Оказалось, что с помощью такого устройства можно вычислять значения функций. Более того, удалось доказать, что для каждого из этих устройств множество функций, которые можно вычислять с его помощью, в точности совпадает с классом функций, рассматривавшихся Л. Черчем. Принимая во внимание тезис Черча, можно сказать, что каждое из устройств — это универсальный вычислитель: все, что алгоритмически можно вычислить с помощью какого-либо устройства, можно вычислить на каждом из этих двух устройств. Применительно к каждому из них высказанное положение называют соответственно тезисом Тьюринга и тезисом Поста.

Разумеется, доказательство сформулированного выше утверждения выходит за рамки школьного курса. Даже не во всех университетских курсах математики и математической логики оно доказывается. Тем не менее мы считаем, что учитель информатики должен знать и понимать истинное положение дел в этой области. И надеемся, что он сможет с должным пафосом донести до учащихся смысл революционных результатов этих великих математиков.

Однако роль этих результатов не только философско-методологическая. Математики получили инструмент для доказательства того, что та или иная задача алгоритмически неразрешима (подробнее этот вопрос обсуждается в § 14). Более того, чтобы сравнивать эффективность различных алгоритмов, надо иметь «эталонного исполнителя», поскольку в противном случае различия, например, в скорости или сложности будут несопоставимы. Поэтому при теоретических оценках параметров алгоритма нередко считают, что он реализован некоторой машиной Тьюринга.

Исследования алгоритмической разрешимости различных задач с помощью машин Тьюринга, Поста, Минского, нормальных алгорифмов Маркова и других средств, изобретенных математиками с целью изучения алгоритмов, показали, что их надо рассматривать не как вычислительные устройства, а как средства преобразования символьной информации. Ведь формальному исполнителю не важно, почему, к примеру, цепочка *abcpbdh* преобразуется в цепочку *cga*, хотя на самом деле это всего лишь записанный другим кодом пример 123 + 248 (буквами от *a* до *j* закодированы цифры 1, 2, ..., 9, 0, а буквой *p* — знак «+»). Важная мысль, что формальная обработка информации — это всего лишь алгоритмическое преобразование одной цепочки символов в другую, полностью была осознана только после изобретений Тьюринга и Поста (а впервые сформулирована была, по-видимому, ленинградским математиком А. А. Марковым в 1949 г.)[1]. Поскольку основным в информатике как науке является изучение возможности автоматизировать обработку информации, указанные гипотетические устройства выступают не только как универсальные вычислители, но и как универсальные исполнители, т. е. такие формальные исполнители, посредством которых можно имитировать обработку информации любым формальным исполнителем.

Последний тезис является отправной точкой в § 9. На наш взгляд, это оптимальное решение, поскольку никаких теоретических обоснований того, что машина Тьюринга является универсальным вычислителем, на уровне знаний учащихся общеобразовательной школы мы привести не можем. В какой мере расширять в этом вопросе кругозор учащихся — решать учителю. Мы выше лишь наметили канву возможного расширения.

Хотя все универсальные исполнители между собой эквивалентны (и похожи по устройству), мы выбрали машину Тьюринга. Нам важно было продемонстрировать, что инструкция формальному исполнителю, т. е. запись алгоритма, вовсе не обязательно выглядит как последовательность команд. Для машины Тьюринга запись алгоритма представляет собой функциональную схему — прямоугольную таблицу, в которой представлены внешний и внутренний алфавиты машины, а также команды перехода[2].

Чтобы учащиеся освоили работу с машиной Тьюринга, им полезно предложить выполнить программу, представленную функциональной схемой в таблице 1.10 учебника (зада-

---

[1] В истории информатики трудно датировать те или иные достижения, поскольку многие работы были засекречены. Без сомнения, Алан Тьюринг, будучи научным руководителем проекта «Энигма», позволившего англичанам во время Второй мировой войны беспрепятственно читать немецкие военно-морские коды, прекрасно понимал, что созданная им дешифровальная машина — это не вычислительное устройство, а именно средство обработки символьной информации. В открытой печати информация об этом проекте появилась только в 60-е гг. (а позже был даже снят художественный фильм).

[2] Напротив, в машине Поста программа работы машины записывается в привычной форме — последовательностью команд. Поэтому в некоторых пропедевтических курсах информатики авторы предпочитают использовать машину Поста (таковым, к примеру, является курс информатики для начальной школы, разработанный авторами из Санкт-Петербурга).

ние 6). При этом надо предложить им 2—3 различные начальные разметки ленты (к примеру, на первой отмечено три звездочки, на второй — пять, а на третьей — одна). Мы рекомендуем именно это задание обсудить в первую очередь.

Что касается других заданий к данному параграфу, то, отвечая на вопросы 1, 2 и 4, учащиеся должны продемонстрировать знание определений. В вопросе 3 ответ прост: составом внутреннего и внешнего алфавитов. Чтобы получить аргументированный ответ на вопрос 5, можно предложить учащимся попытаться нарисовать орграф машины Тьюринга (если предположить, что она является автоматом), заданной таблицей 1.10 учебника. После нескольких попыток учащиеся приходят к выводу о невозможности представить машину Тьюринга как автомат, поскольку без знания, что записано в клетке, куда переместится считывающее устройство, нельзя однозначно указать, в какое состояние перейдет машина. Лента, как мы уже говорили, играет роль памяти, т. е. работа машины зависит не только от состояния, в котором она находится, и текущей команды, но и от результатов всей предшествующей работы. Однако эти соображения еще нельзя назвать доказательством. Ведь может оказаться, что для представления машины Тьюринга в виде автомата необходимо совершенно другие объекты считать состояниями машины и иные объекты — воздействиями на нее. Например, состояние машины — это тройка (x, y, z), где x — один из символов внешнего алфавита, y — один из символов П, Л или Н, z — один из символов внутреннего алфавита. Доказательным обоснованием невозможности представить машину Тьюринга автоматом является предъявление такой функциональной схемы, работая по которой машина Тьюринга будет делать то, что не может сделать ни один автомат. Заметим, что автомат по любой, заданной ему программе закончит работу за конечное число шагов (т. е. перейдет в некоторое состояние и останется в нем). В то же время несложно написать функциональную схему машины Тьюринга, согласно которой машина на некотором слове, записанном в символах внешнего алфавита, никогда не закончит работу. Пример такой функциональной схемы приведен в таблице 4.

Таблица 4

$A \backslash Q$	q_1	q_2
a_0	$*Нq_0$	$*Нq_0$
$*$	$*Пq_2$	$*Лq_1$

Если на ленте стоит одна звездочка, то машина прекратит работу после первого хода. Если же в начальной клетке и в клетке, соседней справа, стоят звездочки, то машина никогда не закончит работу (зациклится). Тем самым такую машину нельзя изобразить никаким автоматом.

Решение задачи 7 на неформальном уровне довольно очевидно: нужно дойти до правого края строчки из звездочек и поставить в следующую клетку звездочку, а затем остановиться. На привычном алгоритмическом языке это запишется так:

Делать пока (в клетке «*»)
{ Написать в клетке «*»;
Шаг вправо;
}
Написать в клетке «*»;
Остановиться.

Мы советовали бы этот алгоритм записать и учащимся — это поможет им почувствовать, что функциональная схема — это просто иная форма записи алгоритма.

В таблице 5 показано, как выглядит такая функциональная схема:

Таблица 5

$A \backslash Q$	q_1
a_0	$*\text{Н}q_0$
*	$*\text{П}q_1$

Задание 8 мы рекомендуем выполнить дома. Здесь тоже можно порекомендовать сначала записать алгоритм на обычном языке, а затем составить функциональную схему. Возможный вариант схемы показан в таблице 6.

Таблица 6

$A \backslash Q$	q_1	q_2
a_0	$a_0\text{Л}q_2$	$1\text{Н}q_0$
0	$0\text{П}q_1$	$1\text{П}q_0$
1	$1\text{П}q_1$	$2\text{П}q_0$
2	$2\text{П}q_1$	$3\text{П}q_0$
3	$3\text{П}q_1$	$4\text{П}q_0$
4	$4\text{П}q_1$	$5\text{П}q_0$
5	$5\text{П}q_1$	$6\text{П}q_0$

Продолжение

A \ Q	q_1	q_2
6	6Пq_1	7Пq_0
7	7Пq_1	8Пq_0
8	8Пq_1	9Пq_0
9	9Пq_1	0Лq_2

В этой схеме столбец, озаглавленный q_1, обеспечивает поиск самой правой цифры числа, а следующий столбец задает изменение цифр при увеличении числа на 1. Разумеется, это не единственное решение; например, кому-то может оказаться мало состояний q_0, q_1, q_2.

В качестве дополнительных заданий можно предложить учащимся запрограммировать сложение двух натуральных чисел (например, по записи $123 + 248$ машина должна выдать запись $123 + 248 = 371$). В этом случае внешний алфавит дополняется символами «+» и «=». Если создавать алгоритм сложения двух десятичных дробей, потребуется дополнить внешний алфавит символом «,» (о чем, в частности, говорится в учебнике). Более сложным является алгоритм сравнения двух натуральных чисел, его реализацию можно предлагать лишь энтузиастам.

В профильном классе сильные учащиеся могут смоделировать машину Тьюринга на компьютере. В качестве ленты может выступать либо достаточно большой массив, либо строковая переменная. Можно воспользоваться и уже имеющейся компьютерной реализацией машины Тьюринга (разработчик Р. Зартдинов). Программа работает под Windows'95/98/NT/XP и является свободно распространенным продуктом. В частности, ее можно получить, обратившись с запросом по электронному адресу Alexgein@yandex.ru. Отметим, что в том же пакете имеется реализация машины Поста.

Тема 4. Заключительный параграф главы 1 носит обзорный характер, и его рассмотрение может быть построено в форме беседы, сообщений учащихся на заданные или выбранные ими самими темы. В последнем случае мы советуем предложить учащимся самостоятельно прочитать данный параграф, составить его план-конспект, а затем выбрать тот или иной пункт, чтобы по нему сделать развернутое сообщение.

Для проведения компьютерного практикума в профильном уровне обучения мы советуем рассмотреть простейшие методы шифрования. Ниже приведен один из таких способов.

Задание 6. а) Одной из разновидностей шифрования методом простой подстановки для русскоязычных текстов является шифрование методом сдвига.

Выбирается целое число k от 1 до 32. Буквы русского алфавита записываются вдоль окружности по часовой стрелке (так что буква «а» соседствует с буквами «б» и «я»). Затем в сообщении каждая буква заменяется на букву, отстоящую от нее на k букв по часовой стрелке. Знаки препинания и пробел не меняются. Например, если $k = 3$, то «а» заменится на «г», «щ» — на «ь», «ю» — на «б», а фраза «ученье — свет, неученье — тьма» превратится в «цъиряи — феих, рицьиряи — хяп». Составьте алгоритм шифровки и дешифровки русскоязычного текста методом сдвига. Запишите этот алгоритм на изученном вами языке программирования.

б) Какой алгоритм применяется для расшифровки текста, зашифрованного указанным методом?

Разбейтесь на пары. Зашифруйте какой-либо текст (выбрав какое-либо значение k) и передайте его вашему напарнику; значение числа k ему не сообщайте. Постарайтесь расшифровать полученный вами текст.

Разумеется, можно рассмотреть и другие методы шифрования текста и даже предложить учащимся самим придумать таковые.

Приведенный в конце главы 1 тест предназначен для организации самопроверки. Ниже мы предлагаем аналогичный тест для выполнения в классе за счет часов из резерва учителя (если учитель считает целесообразным проведение такого теста).

Контрольный тест по материалу главы 1

Часть 1. При выполнении предложенных ниже заданий укажите номер правильного ответа.

А1. Информационный объем сообщения равен 128 битам. В байтах объем того же сообщения равен:

1) 4; 2) 7; 3) 16; 4) другому, нежели в пунктах 1—3, числу.

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

1) 76 бит; 2) 608 бит; 3) 76 килобайт; 4) 608 килобайт.

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

$$a — 01;\quad b — 110;\quad c — 100;\quad d — 101;\quad e — 10.$$

Двоичной строкой 1010010110101 закодирована следующая последовательность из пяти букв:

1) *ecabd*; 2) *dabbd*; 3) *dabda*; 4) *ecdda*.

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 строкой 01001011101101001 закодирована последовательность *beceda*. Тогда последовательность *abcde* кодируется как:

1) 00101010010101; 2) 001010011100101;
3) 01010110101101; 4) 001010011110101.

А5. Свойство модели отражать элементный состав и существенные связи между элементами называется:

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

А6. Допустимость действия исполнителя — это:

1) разрешение исполнителю выполнить это действие;

2) умение исполнителя выполнить это действие;

3) обязанность для исполнителя выполнить это действие;

4) в пунктах 1—3 нет правильной формулировки.

А7. Дискретность алгоритма означает, что:

1) исполнение алгоритма совершается по шагам;

2) исполнение алгоритма завершается после конечного числа шагов;

3) исполнение алгоритма приводит к получению необходимого результата;

4) в пунктах 1—3 нет правильной формулировки.

А8. Ветвление в неполной 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.

А9. Имеются две кучки камней, в одной 13 камней, в другой 18. Исполнитель имеет три допустимых действия:

а) взять из первой кучки 3 камня и переложить их во вторую;

б) взять из второй кучки 5 камней и переложить их в первую;

в) 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 3 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*.

А10. Значения переменных y и z после исполнения алгоритма, изображенного схемой на рисунке 14, таковы:

1) $y = 2$; $z = 1$; 2) $y = 8$; $z = 2$; 3) $y = 2$; $z = 4$; 4) $y = 8$; $z = 1$.

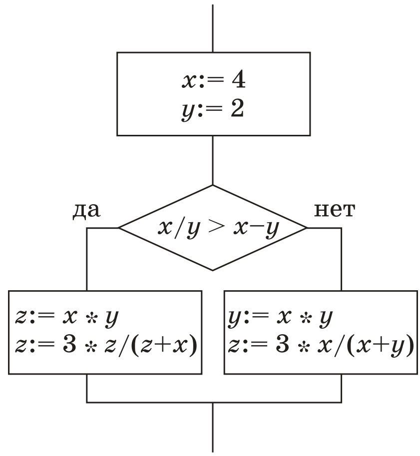

Рис. 14

А11. На рисунке 15 изображена схема алгоритма.

Значения переменных a и b после исполнения этого алгоритма:

1) $a = 10$; $b = -20$; 2) $a = -4$; $b = -2$; 3) $a = 5$; $b = -5$;

4) не совпадают ни с одной парой значений, указанных в пунктах 1—3.

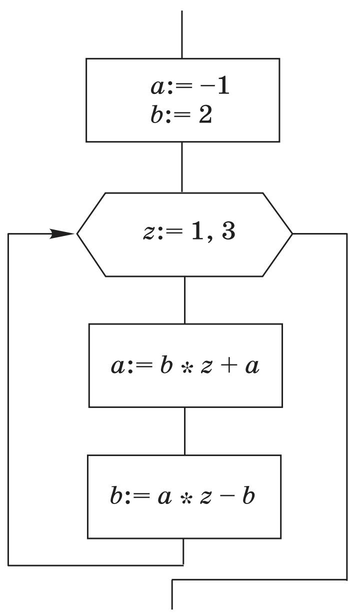

Рис. 15

А12. Дан фрагмент алгоритма:

$a := 1;$
$b := 0;$
<u>Делать</u> **<u>пока</u>** $a + b \geqslant a*b$
{ $a := b + 1;$
$b := 2*a;$
}

Тело цикла будет исполнено:

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

А13. Дан алгоритм:

<u>Алгоритм</u>
<u>сим</u>: A; **<u>цел</u>** X, K, M;
{ **<u>Запросить</u>** A;
$X := 0;$

Если (Часть $(A, 2, 1) =$ Часть $(A, 7, 1))$ **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 2. Значение, присвоенное переменной A после исполнения первого оператора в теле алгоритма, могло быть:

1) ПЕРИНА; 2) МАЛИНА; 3) КАРЗИНА; 4) КАРТИНА.

А14. Дан алгоритм:

Алгоритм
цел: M;
{ **Запросить** M;
Если $(M < 0)$ **то** { $M := -M$; }
{ **Делать** **пока** $(M - 3) * (M - 8) > 0$
{ $M := 2*M$; }
}
}

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

1) 6; 2) 8; 3) 16; 4) другому, нежели в пунктах 1—3, числу.

Часть 2. При выполнении предложенных ниже заданий запишите ответ в виде последовательности символов.

В1. На рисунке 16 изображена транспортная схема между несколькими пунктами. Расстояния между ними указаны в километрах. Кратчайший маршрут, проходящий через все пункты, имеет длину ____ км.

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

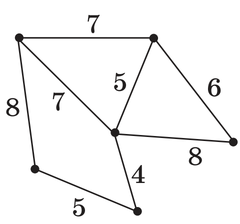

Рис. 16

В3. Дан алгоритм:

Алгоритм
цел: X;
{ **Запросить** X;
Если $((X < 150)$ **и** $(X > 75)$ **то**
{ **Делать** **пока** $X \bmod 2 = 1$
{ $X := (X + 1)/2$; }
}
иначе
{ $X := X - 2$;
Если $(X < 10)$ **то**
{ $X := -X + 3$; }
}

Сообщить значение X;
}
После исполнения этого алгоритма было сообщено число 9. Какое число было присвоено переменной X по команде **Запросить** X? Если таких чисел может быть несколько, запишите все их через запятую в порядке возрастания.

В4. Дан алгоритм:

Алгоритм
сим: A; **цел**: K, M;
{ **Запросить** A;
$M := \text{LEN}\ (A)$;
{ **Делать от** $K := 1$ **до** M
{ $A :=$ Часть $(A,\ 2*K-1,\ 2*K-1)$ +
Часть $(A,\ 2*K-1,\ 1)$ + Часть $(A,\ 2*K,\ M-K)$;
}
}
}

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, чем 2 раза;
4) слово, в котором хотя бы один символ встречается нечетное число раз.

Запишите в порядке возрастания без пробелов номера верных утверждений.

В5. Укажите слово, которое надо поставить вместо пропуска в следующем предложении: «Переменная А1 предназначена для хранения информации о количестве слов в обрабатываемом фрагменте текста; для переменной А1 нужно объявить ________ тип переменной».

Ключи к тесту

Задание	А1	А2	А3	А4	А5	А6	А7	А8	А9	А10	А11
Ответ	3	2	4	4	1	2	1	2	2	4	2

Задание	А12	А13	А14	В1	В2	В3	В4	В5
Ответ	3	4	3	28	циклом	8, 129	13	целый

ГЛАВА 2

Информационная деятельность человека и использование в ней компьютерных технологий

Тема 5. Роль человека в информационных процессах является решающей. Поэтому чрезвычайно важно воспитывать у учащихся культуру информационной деятельности как составную часть информационной культуры личности. Современные концепции формирования информационной культуры исходят прежде всего из того положения, что обладание такой культурой предусматривает наличие у человека информационного мировоззрения и системы знаний и умений, обеспечивающих целенаправленную самостоятельную деятельность по оптимальному удовлетворению индивидуальных информационных потребностей с использованием как традиционных, так и новых информационных технологий. При указанном подходе информационная культура личности совершенно справедливо рассматривается как важнейший фактор успешной профессиональной и непрофессиональной деятельности, а также социальной защищенности личности в информационном обществе. Важно, что информационная культура в контексте общечеловеческой культуры предоставляет осознанную свободу выбора, ограниченную культурными ценностями человеческой цивилизации, поэтому мы и ставим целью формирование у учащихся ценностных ориентиров и ограничителей в использовании информации.

В § 11 как раз совершается перенос акцента с общенаучных трактовок понятий информации и информационного процесса на включенность человека в такие процессы. Важно, чтобы участие в них человека было вполне осознанным и опирающимся на понимание информационных закономерностей.

Характер материала в этом параграфе таков, что его рассмотрение целесообразно перемежать с выполнением заданий. В заданиях примеры могут быть из любой области, даже бытовой. Например, надо купить подарок другу на день рождения. Требуется информация, какой подарок может доставить вашему другу удовольствие. Возможно, вы уже это знаете, а может быть, это знание придется каким-то образом получить (расспрашивая друзей или родителей). Во втором варианте это некий социологический эксперимент, и его надо уметь провести. На следующем этапе вам нужна информация, где можно купить или как изготовить намеченный подарок. Конечно, может оказаться так, что вам снова потребуется получать информацию, но более вероятно, что вы просто знаете об этом или знаете, как быстро получить нужную информацию. Наконец, вы принимаете решение (это и есть результирующая информация), какой подарок вы будете делать и как вы сможете свое решение воплотить на практике. На наш взгляд, полезно, разобрав один пример в классе (кроме того, что описан в учебнике), предложить учащимся дома подготовить еще один (а может, и не один) пример с подробным анализом этапов информационной деятельности.

В задании 2а ответ следует непосредственно из объяснительного текста: получение информации о реальном объекте всегда требует экспериментальной деятельности — иначе мы об этом объекте ничего не узнаем.

В пункте *б* того же задания ответ, на наш взгляд, таков: на втором этапе экспериментальная деятельность не требуется, ибо, если она вдруг понадобилась, значит, у нас было недостаточно исходной информации, и мы должны вернуться на первый этап. Полезно продемонстрировать учащимся, что «гладкое» прохождение всех этапов — ситуация, скорее всего, идеальная. Намного чаще приходится возвращаться, казалось бы, к уже выполненным этапам, чтобы получить недостающую информацию или воспользоваться дополнительными знаниями.

Задание 4 достаточно простое. Его цель, как в закреплении материала данного параграфа, так и в подготовке к восприятию материала главы 4, изучая который учащимся предстоит не только отличать случаи декларативного представления информации от других вариантов ее представления, но и выделять среди них высказывания и предикаты. Ответ в этом задании таков: декларативно информация представлена в пунктах *а, в, г, д, ж*; процедурно — в пунктах *б* и *е*.

В задании 5а декларативное утверждение «Земля вращается вокруг Солнца» нельзя превратить в процедурное, а утверждение «Вода замерзает при температуре 0 °С» можно: «Чтобы заморозить воду, надо понизить ее температуру до 0 °С». Процедурную информацию о получении огня трением можно превратить в декларативную, например, так: «Если достаточно долго тереть друг о друга две плотно прижатые сухие палочки, то возникнет огонь». А вот процедурная информация о ловле рыбы вряд ли может быть преобразована в декларативную: если даже выполнить все указанные действия, рыбка может и не пойматься. Вряд ли можно указать какое-нибудь универсальное правило, которое позволяло бы гарантированно определять, можно ли заданную декларативную информацию превращать в процедурную и наоборот. Да этого и не требуется.

В пункте *б* того же задания авторский ответ таков: утверждения *в, г,* и *д* из задания 4 допускают преобразование в информацию процедурного типа. В свою очередь, утверждения *б* и *е* из задания 4 легко преобразуются в декларативную форму.

Тема, обсуждаемая в § 12, имеет в курсе информатики несколько граней. Во-первых, учащиеся осваивают важный и весьма распространенный класс информационных моделей — фактографические модели. Обычно эти модели представлены базами данных. Говоря об информационых моделях в узком смысле, их иногда даже отождествляют с базами данных. Мы, конечно, не будем придерживаться такого узкого смысла термина «информационная модель».

Во-вторых, информационные модели, с которыми учащиеся до этого имели дело, описывались на уже известном из других дисциплин языке с помощью уже известного инструментария. Например, математические модели описаны на языке математики, модели химических реакций — на языке уравнений этих реакций и т. д. Здесь же язык описания модели (можно сказать, язык таблиц) учащимся практически неизвестен.

В-третьих, совершенно новым для учащихся является инструментарий работы с данным видом моделей. Если для математических моделей таким инструментарием выступает изучаемый в курсе математики математический аппарат, то для данного класса моделей инструментарием выступают системы управления базами данных (СУБД). При этом наличие единых принципов построения и управления базами данных показывает, что мы имеем дело с вполне определенной информационной технологией. К сожалению, нередко изучение этой технологии (впрочем, как и других) низводится к рецептурному освоению того или иного конкретного программного продукта.

Указанные три аспекта показывают, что на самом деле значительную часть курса информатики можно выстроить, отталкиваясь от информационных моделей, представленных базами данных. Частично такое построение выполнено нами в главе 4. В данной главе преследуются более простые цели — повторить основные сведения о базах данных, если учащиеся уже изучали их в 8—9 классах, или познакомить их с ними.

Когда информационная модель представляется базой данных, то в роли параметров этой модели выступают признаки объектов, информация о которых хранится в данной базе данных. В настоящее время вместо термина «признак» обычно используется термин «атрибут». Тем самым для построения базы данных мы должны у совокупности объектов выделить общие признаки, существенные с точки зрения решаемых информационных задач. В объяснительном тексте § 12 этому уделено должное внимание.

Обсуждая, как в данной информационной модели представляются связи между признаками, надо показать роль табличной формы в ее представлении. Важнейшее понятие, которым должен владеть каждый ученик, — это понятие **запроса**.

Система управления базой данных играет роль, аналогичную языку программирования. Можно считать, что это некоторая система команд, обеспечивающая организацию данных и манипулирование ими. Аналогия усиливается тем, что для получения определенного результата с помощью уже имеющейся базы данных требуется исполнить некоторый алгоритм, составленный с использованием команд этой системы. Отличие же состоит в том, что значительную роль в системах управления базами данных играют команды, направленные на создание самой базы. Разумеется, и в языках программирования есть команды организации структур данных (например, массивов), но эти структуры достаточно просты и создаются, можно сказать, одной-двумя командами.

Главная забота СУБД — выдавать ответы на поступающие запросы, и учащиеся должны понимать, как вычисляется ответ на запрос в современных СУБД. Поскольку в реляционных базах данных вся информация представляется в форме таблиц, то это озачает, что СУБД выполняет над таблицами операции, которые преобразуют таблицы с исходной информацией в таблицу-ответ.

Какие же операции над таблицами применяются при вычислении ответа на запрос? Прежде всего это *фильтрация* (выделение тех строк из таблицы, которые обладают заданными свойствами), *проекция* (удаление ненужных столбцов), *объединение* (добавление к одной таблице другой таблицы), *теоретико-множественная разность* (удаление из одной таблицы всех строк, содержащихся в другой таблице) и, наконец, *соединение* таблиц[1]. Подробно операции над таблицами будут рассмотрены в главе 4.

Из приведенного в учебнике описания операции «фильтрация» следует, что **фильтр** — это логическое выражение, а **фильтрация** таблицы — это отбор тех ее строк, для кото-

[1] Заметим, что при «локализации» Access была допущена досадная ошибка — операции соединения (Join) и объединения (Union) стали называться одним словом: *объединение*. Это нередко приводит к большой путанице. Попробуйте-ка научить ребенка сложению и умножению, если вы называете эти действия одинаково!

рых соответствующее выражение истинно. Записывать фильтр можно двояко: либо используя язык SQL (от Structured Query Language — структурированный язык запросов), либо заполняя бланк QBE. Мы избрали второй вариант как более простой для реализации и более легкий для объяснения.

Сделаем одно терминологическое пояснение. Мы пользуемся термином «атрибут» вместо более употребительного на практике термина «поле». Дело в том, что термин «атрибут» принят в теории баз данных, а в документации СУБД Access используется термин «поле». Но там этот термин используется в разных смыслах — то это именно столбец (т. е. атрибут), то это клетка таблицы.

Как обычно, мы предлагаем сценарий лабораторной работы (точнее, двух лабораторных работ: в первой учащиеся создают базу данных, а во второй работают с ней в режиме поиска нужной информации), направленной на приобретение навыков работы с компьютером в рамках изучаемой темы. Инструментом здесь является СУБД Access. Впрочем, если учитель располагает какой-либо другой базой данных, он может воспользоваться ею с единственным условием, что она позволяет отработать все те умения, которые предусмотрены сценариями лабораторных работ 4 и 5.

Задания 3—5 к § 12 не имеют канонического ответа. Возможны различные варианты, и важно организовать их конструктивное обсуждение. В задании 6 отрабатываются первые навыки составления запросов. Обычно это не вызывает трудностей. Надо только при выполнении задания 6г иметь в виду, что Америка состоит из двух континентов — Северной Америки и Южной Америки, поэтому при составлении соответствующего запроса нужно использовать союз ***или***. Сама форма запроса, описанная в условии задания 6, представляет собой логическое выражение. Однако с целью подготовки к работе с бланком QBE можно, если учитель считает это целесообразным, приготовить несколько моделей этого бланка (см. таблицу КП.4 в тексте лабораторной работы 5) и предложить учащимся записать запросы в них. В задании 7а учащиеся нередко совершают ошибку, записывая запрос в виде

Происхождение=ледниковое ***и Глубина***<100.

Ошибочность такого запроса становится особенно ясной, если предложить ученикам зписать запрос для задания 7б — он тогда должен выглядеть точно так же! Но большинству учащихся интуитивно ясно, что ответы на эти два запроса совпадать не обязаны. Это задание рассчитано на развитую логическую культуру анализа высказываний, полученных из предиката связыванием переменной при помощи квантора (в данном случае квантора всеобщности). Систематическое освоение этого материала предусмотрено в главе 4. Имея это в виду, мы рекомендуем предложить учащимся сначала сформулировать утверждение пункта *а* в следующей форме: «Для любого озера верно: если оно ледникового происхождения, то его глубина меньше 100 м». На языке СУБД Access в рамках того, что допускает бланк QBE, такой запрос записать нельзя — он дает возможность записать запрос с квантором существования (существуют ли объекты с заданными значениями атрибутов) и в качестве ответа перечислит все существующие объекты. Поэтому надо перейти к отрицанию: существует озеро неледникового происхождения с глубиной менее 100 м. И если хотя бы одно озеро будет найдено, значит, данное утверждение истинно, а исходное утверждение ложно. Аналогично решается задача 7б.

Задание 8 направлено на профилактику весьма распространенной ошибки — употребление союза ***и*** вместо нужного в этом месте союза ***или***.

Задания 9 и 10 являются подготовительными к лабораторным работам 4 и 5. Если у учащихся не было затруднений с выполнением задания 7, можно в задание 10 добавить вопросы типа: «Верно ли, что у всех девочек день рождения в первой половине года?», «Верно ли, что у каждого мальчика есть сестра?» и т. п. При выполнении лабораторной работы 5 также можно расширить спектр заданий, включив вопросы, позволяющие определить, например, все ли, кто посещает спортивные секции, живут на той же улице, где расположена школа.

На наш взгляд, важно с идейной точки зрения продемонстрировать учащимся применение компьютера как инструмента для обработки экспериментальных данных. Достижению этой цели посвящен § 13. Излагаемый в нем метод наименьших квадратов является одним из наиболее употребительных методов обработки экспериментальных данных. Для обоснования этого метода требуется довольно свободное владение математическим аппаратом, в частности хорошее знание свойств квадратичной функции, поэтому мы рекомендуем рассматривать его только в классах математического профиля (хотя в первых школьных учебниках по информатике этот метод обязательно присутствовал; по-видимому, именно из указанных выше идейных соображений). Что касается компьютерного практикума, то мы считаем важным, чтобы учащиеся познакомились с надстройкой *Поиск решения*, и рекомендуем эту часть лабораторной работы провести независимо от выбранного профиля.

Тема 6. Деление информации на декларативную и процедурную, о чем рассказывалось в § 11, во многом предопределено двумя парадигмами программирования: процедурной и декларативной. К процедурным языкам программирования относятся все те языки, которые задают инструкцию формальному исполнителю в виде алгоритма, т. е. последовательности точных предписаний, что и как делать, чтобы добиться требуемого результата. К декларативным относятся языки логического программирования, например Пролог. Мы пока продолжаем оставаться на позициях процедурной парадигмы и в § 14 изучаем методы алгоритмизации и свойства алгоритмов. Здесь имеется три относительно самостоятельных концентра. Первый из них — повторение понятия вспомогательного алгоритма и подпрограммы, а также рассмотрение метода пошаговой детализации. Второй — эффективное доказательство существования алгоритмически неразрешимой задачи. Третий — зависимость результата исполнения алгоритма от свойств исполнителя.

В названии «вспомогательный алгоритм» отражен главный мотив введения такого понятия: он помогает конструировать более сложные алгоритмы. Этот термин был предложен А. П. Ершовым в самом первом курсе информатики, введенном в общеобразовательную школу в 1985 г. Термин «подпрограмма» появился намного раньше — около 70 лет назад. Приставка «под-» отражает в этом термине подчиненность данного объекта по отношению к некоему более главному объекту (ср.: мастер — подмастерье, полковник — подполковник, чин — подчинение и т. п.). Соподчиненность вспомогательных алгоритмов (и подпрограмм) находит свое отражение как в семантике, так и в синтаксисе: понятия локальных и глобальных переменных, формальных и фактических параметров, правил оформления заголовка вспомогательного алгоритма, обращения к нему и возврата в исходный алгоритм. В большинстве случаев достаточно механизма локальных переменных, поэтому при изучении курса на базовом уровне можно ограничиться только их рассмотрением. Однако есть задачи, в которых обходиться только локальными переменными сложно. В большинстве языков программирования переменная, не объявленная в подпрограмме, но объявленная в основной программе, автоматически считается глобальной. Мы же в алгоритмах, которые будут иллюстрировать решение задач учебника, будем для переменных, используемых в качестве глобальных, каждый раз указывать их статус.

Использование вспомогательных алгоритмов имеет два основных варианта:

а) составление алгоритмов с использованием готовых вспомогательных алгоритмов (восходящее проектирование);

б) составление алгоритмов методом пошаговой детализации (нисходящее проектирование).

В объяснительном тексте восходящее проектирование не рассматривается. Однако оно неявно присутствует в виде подпункта *б* в заданиях 3, 5 и 6, где предлагается воспользоваться составленной ранее подпрограммой-функцией для решения новой задачи.

В объяснительном тексте § 14 мы специально рассматриваем два возможных пути пошаговой детализации решения задачи о разложении натурального числа на простые множители — учащиеся должны осознавать принципиальную возможность вариативности в выборе путей решения практически любой нетривиальной задачи. Такое понимание является важным элементом экспертного уровня компетентности, и мы призываем учителя каждый раз, когда имеется возможность, обращать на это внимание учеников.

Не отвлекаясь на другие концентры данного параграфа, прокомментируем задания к нему, поскольку все, кроме заданий 10 и 11, относятся к данному вопросу.

Ответ в задании 1б таков: в тех случаях, когда результат может быть описан переменной или какой-либо другой структурой данных (например, массивом). Однако массивы, и тем более другие структуры данных, еще не рассматривались в этом учебнике. Поэтому в объяснительном тексте ответу на этот вопрос придана иная форма — синтаксическая: результат описывается одним формальным параметром. Если учащиеся уже знакомы с понятием массива (по курсу информатики 8—9 классов), то можно пояснить, что таким формальным параметром может выступать, например, массив.

В задании 2а учащиеся должны показать, что предложенный алгоритм разыскивает наименьший неединичный делитель натурального числа. Он автоматически будет простым, ибо если у него есть какой-то свой собственный делитель (т. е. отличный от 1 и самого числа), то он будет делителем и исходного числа, причем меньшим чем найденный.

Конечность данного алгоритма (задание 2б) объясняется тем, что переменная p увеличивается на единицу при каждом исполнении тела цикла, но не может принять значение, большее чем n. Это означает, что сама переменная p может быть взята на роль лимитирующей функции (см. обсуждение задания 22в к § 7).

Для задания 3а мы сразу приведем требуемую подпрограмму-функцию.

Функция Показатель_степени (**арг** **цел**: n, p): **цел**
цел: k;
{ $k := 0$;
  Делать **пока** $n \bmod p = 0$
  { $k := k + 1$; $n := n/p$;
  }
  знач $:= k$;
}

Составление алгоритма разложения натурального числа на простые множители (задание 3*б*) выполняется теперь без труда в соответствии со схемой алгоритма, представленной в учебнике на рисунке 2.5.

В задании 4а фактически надо лишь договориться, какое имя в дальнейшем будет иметь подпрограмма-функция вычисления наибольшего общего делителя двух натуральных чисел. Мы советуем для краткости назвать ее НОД. Тогда для вычисления НОД (m, n) достаточно будет написать

$k := \text{НОД}\ (m, n);$

Приведем подпрограмму-функцию, вычисляющую значение функции Эйлера (задание 4б).

```
Функция FunE (арг цел: n): цел
цел: k, m;
{  k := 1;
   Делать от m := 2 до n − 1
   {  Если НОД (m, n) = 1 то {  k := k + 1;  }
   }
   знач := k;
}
```

Впрочем, можно написать подпрограмму, работающую в 2 раза быстрее:

```
Функция FunEm (арг цел: n): цел
цел: k, m;
{  Если n ⩽ 2 то {знач: = 1;  }
   иначе
   {  k := 1;
      Делать от m := 2 до (n − 1)/2
      {  Если НОД (m, n) = 1 то {  k := k + 1;  }  }
      знач := 2*k;
      }
}
```

Если такой алгоритм рассматривается, то учащиеся должны пояснить, почему он тоже вычисляет значение функции Эйлера. Основным в этом обосновании является замечание, что НОД $(m, n) =$ НОД $(n - m, n)$.

Несложно написать подпрограмму-функцию для функции $\psi(n)$, о которой говорится в задании 4в. Вот эта подпрограмма-функция.

```
Функция FunPsi (арг цел: n): цел
цел: k, m;
{  k := 1;
   Делать от m := 2 до n − 1
   {  Если НОД (m, n) = 1 то {  k := k + m;  }
   }
     знач := k;
}
```

Подпрограмма-функция, которую нужно составить, выполняя задание 5, может выглядеть так:

```
Функция Сумма_цифр (арг цел: n): цел
цел: k, m;
{  k := n mod 10;
   m := n div 10;
   Делать пока m > 0
   {  k := k + m mod 10;
      {  m := m div 10;
   знач := k;
}
```

Используя эту функцию, легко составить требуемый алгоритм из задания 5б.

```
Алгоритм Количество_чисел_с_заданной_суммой
цел: k, n, s, m;
{  Запросить n;
   Запросить k;
   s := 0;
   Делать от m := 10^(n−1) до 10^(n−1)
   {  Если Сумма_цифр(m) = k то {  s := s + 1;  }
   }
   знач := k;
}
```

Конечно, этот алгоритм полного перебора n-значных чисел крайне неэффективен. Более того, он не отсекает очевидные случаи, когда чисел, удовлетворяющих данному условию, просто нет, например при $k > 9n$. Поэтому при проведении лабораторной работы 7 надо предложить учащимся набор таких значений k и n, который бы подвел их к идее о необходимости модернизации программы. Например, $n = 20$, $k = 200$ (если учащиеся, не подумав, запустят программу с этими данными, то нужно заметить, что ответ можно получить без всякого компьютера — ясно, что, даже имея 20 девяток, сумму 200 не получить). Однако мы против того, чтобы учащиеся во время работы за компьютером пытались улучшать свой программный продукт. План работы может быть таков: на первом занятии учащиеся вводят и отлаживают подготовленные программы, тестируют их (в том числе на эффективность), обнаруживают возможности для улучшения, а затем дома или на теоретическом занятии разрабатывают улучшенные модификации и только после этого апробируют их в компьютерном классе.

Задача 6а может быть решена разными способами. Поскольку это задача на упорядочение, можно применить любой метод сортировки (например, метод пузырька). И если учитель хочет повторить методы сортировки, то такой подход вполне уместен. А можно воспользоваться тем, что в языках программирования нередко допускается организация цикла с параметром, где в качестве параметра выступает символьная переменная (например, **Делать от** $s := а$ **до** я). Мы предлагаем более «традиционный» вариант (но не уступающий по эффективности).

```
Функция Алфавитное_упорядочение (арг сим: W) : сим
сим: V, VR; цел: k, m;
{  VR := "абвгдеёжзиклмнопрстуфхцчшщъыьэюя";
   Запросить W;
   V := "";
   Делать от k := 1 до 33
   {  Делать от m := 1 до Len(W)
   {  Если Часть(VR, k, 1) = Часть(W, m, 1) то
   {  V := V + Часть(W, m, 1);  }
   }
   }
   знач: = V;
}
```

С помощью построенной подпрограммы-функции легко решается задача 6б. Вот соответствующий алгоритм.

```
Алгоритм Совпадение_перестановок
сим: V, W;
{  Запросить V;
   Запросить W;
   Если Алфавитное_упорядочение(V) = Алфавитное_упорядочение(W)
   то
   {  Сообщить "Можно";  }
   иначе
   {  Сообщить "Нельзя";  }
}
```

Конечно, задача 6а является хорошей подсказкой для решения задачи 6б. Поэтому в сильном классе можно предложить сразу решать задачу 6б, а к задаче 6а прийти как к некоторому шагу в пошаговой детализации решения задачи 6б. В целом идею задачи 6б можно охарактеризовать как один из широко используемых приемов решения математических и программистских задач: приведение разных объектов к одному стандартному виду с последующим сравнением получившихся объектов. В математике, например, для установления, что два многочлена равны, их приводят к стандартному виду — расположению одночленов сравниваемых многочленов по убыванию степеней.

Задача 7 относится к разделу, который в олимпиадной информатике принято называть «вычислительной геометрией». Это не значит, что она слишком трудна для учащихся. Она служит опорным элементом многих олимпиадных задач, связанных с теми или иными геометрическими конфигурациями.

В задании 7а надо прежде всего сформулировать на языке координат, что значит три точки лежат на одной прямой — ведь только в этом случае они не будут образовывать треугольник. Для решения этой задачи лучше всего воспользоваться векторным методом (векторы на плоскости изучались в 9 классе). Пусть A, B и C — три точки плоскости, $\overrightarrow{AB}$ и $\overrightarrow{AC}$ — соответствующие векторы. Ясно, что эти точки лежат на одной прямой в том и только том случае, когда векторы $\overrightarrow{AB}$ и $\overrightarrow{AC}$ параллельны (коллинеарны). В свою очередь, параллельность векторов равносильна их пропорциональности (т. е. один вектор получается из другого умножением на константу). Перейдем к координатной форме представления векторов. Пусть $A(x_1,\ y_1)$, $B(x_2,\ y_2)$ и $C(x_3,\ y_3)$. Тогда $\overrightarrow{AB} = (x_2 - x_1,\ y_2 - y_1)$ и $\overrightarrow{AC} = (x_3 - x_1,\ y_3 - y_1)$. Пропорциональность векторов означает выполнение пропорции $\frac{x_2 - x_1}{x_3 - x_1} = \frac{y_2 - y_1}{y_3 - y_1}$. В математике на этом можно было бы остановиться, но грамотный программист должен предвидеть, что такая форма записи условия параллельности векторов чревата неприятностями, например возникновением ситуации деления на ноль. Поэтому целесообразно записать то же условие в другой форме: $(x_2 - x_1)(y_3 - y_1) = (x_3 - x_1)(y_2 - y_1)$. Теперь легко составить нужный алгоритм.

```
Алгоритм Треугольник
цел: x1, y1, x2, y2, x3, y3;
{  Запросить x1;
   Запросить y1;
   Запросить x2;
   Запросить y2;
   Запросить x3;
   Запросить y3;
   Если (x2 − x1)(y3 − y1) = (x3 − x1)(y2 − y1) то
   {  Сообщить ”Не являются”;  }
   иначе
   {  Сообщить ”Являются”;  }
}
```

В задании 7б ведущей идеей является установление того факта, что заданная точка и вершина треугольника лежат по одну сторону от прямой, проходящей через сторону треугольника, противоположную данной вершине. Если для каждой вершины это будет справедливо, то заданная точка расположена внутри треугольника. Этим рассуждением сформулирована подзадача об определении, лежат ли две заданные точки по одну сторону от прямой. Займемся теперь этой подзадачей.

Пусть прямая проходит через точки $A(x_1,\ y_1)$ и $B(x_2,\ y_2)$, а про точки $C(x_3,\ y_3)$ и $D(x_3,\ y_3)$ как раз и требуется узнать, лежат ли они по одну сторону от прямой AB. Условием, что точка C лежит на прямой AB, является, как мы видели, равенство нулю выражения $(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)$. Если же C не лежит на AB, то это выражение нулю не равно. Оказывается, что *для всех точек, расположенных по одну сторону от прямой, данное выражение всегда имеет один и тот же знак*[1]. В данном случае учитель информатики не должен тратить время на обоснование этого чисто математического утверждения (можно предложить учащимся обратиться за помощью, если им это необходимо, к учителю математики). В связи с этим мы рекомендуем это задание разбирать только в математических классах или с учащимися, готовящимися к выступлению на олимпиадах по информатике.

Итак, условием того, что точки C и D лежат по одну сторону от прямой AB, является положительность выражения

$$((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))((x_2 - x_1)(y_4 - y_1) - (x_4 - x_1)(y_2 - y_1)).$$

Аналогично записываются условия того, что точки A и D лежат по одну сторону от прямой BC и точки B и D лежат по одну сторону от прямой AC. Алгоритм теперь записывается без труда.

В задании 8 надо прежде всего построить математическую модель данной задачи. Пусть x, y и z — количество покупаемых быков, коров и телят соответственно. Тогда условие задачи записывается системой уравнений

$$x + y + z = 100;\ 25\,000x + 16\,000y + 2000z = 500\,000.$$

Второе уравнение полезно переписать в виде $25x + 16y + 2z = 500$. От одного неизвестного полезно избавиться, например от z. Тогда математическая модель задачи будет выглядеть как смешанная система: $23x + 14y = 300$; $x + y \leqslant 100$, x и y — целые неотрицательные числа. Впрочем, из первого уравнения и неотрицательности y следует, что $x \leqslant 300 : 23 \approx 13{,}043$. Следовательно, $x \leqslant 13$. Аналогично делается вывод, что $y \leqslant 21$. Поэтому неравенство $x + y \leqslant 100$ выполняется автома-

[1] Доказать данное утверждение можно многими способами. Например, записать общее уравнение прямой, проходящей через точки A и B, а затем показать, что для точек, расположенных по одну сторону от этой прямой, знак левой части общего уравнения прямой будет одинаков. Но, на наш взгляд, более полезной является интерпретация выражения $(x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1)$ как «ориентированной» площади параллелограмма, построенного на векторах $\overrightarrow{AB}$ и $\overrightarrow{AC}$ (слово «ориентированная» означает, что площадь берется со знаком «+», если поворот на наименьший угол от вектора $\overrightarrow{AB}$ к вектору $\overrightarrow{AC}$ осуществляется в положительном направлении, т. е. против часовой стрелки, и берется со знаком «−» в противном случае). Учащиеся получат одновременно важный инструмент для вычисления площадей параллелограмма и треугольника. Ясно, что точки лежат по одну сторону от прямой, если площади соответствующих параллелограммов ориентированы одинаково.

тически. Кроме того, поскольку $23x = 300 - 14y = 2(150 - 7y)$, число x должно быть четным. Тем самым для x остаются только следующие варианты: 0; 2; 4; 6; 8; 10; 12. Составляем требуемый алгоритм.

```
Алгоритм Покупка
цел: x, y, z;
{  Делать от x = 0 до 12 с шагом 2
   {  Если (300 − 23*x)mod14 = 0 то
      {  y := (300 − 23*x)/14;
         z := 100 − x − y;
      }
      Сообщить ”Количество закупаемых быков”, x;
      Сообщить ”Количество закупаемых коров”, y;
      Сообщить ”Количество закупаемых телят”, z;
   }
}
```

Эта задача предоставляет еще раз возможность обсуждения вопросов оптимальности алгоритмов. У школьников нередко возникает желание написать алгоритм перебора вариантов по всем целочисленным наборам x, y и z, где каждая переменная не превосходит 100. Таких вариантов 1 000 000. Первое оптимизирующее предложение состоит в том, что z можно выразить как $100 - x - y$; тогда количество вариантов при полном переборе будет 10 000, т. е. в 100 раз меньше. Алгоритм, который предложен нами, содержит перебор всего 7 вариантов и в принципе может быть выполнен без компьютера. Ответ: 10 быков, 5 коров и 85 телят.

Задача 9 требует от учащихся определенной математической культуры. Первое замечание, позволяющее оптимизировать перебор вариантов, таково:

Пусть x, y и z — длины отрезков. Поскольку изменение порядка перечисления сторон треугольника самого треугольника не меняет, можно считать, что $x < y < z$. Тогда условие, что из этих трех отрезков можно составить треугольник, выглядит так: $x + y > z$. Тем самым математическая модель задачи записывается следующим набором соотношений: $n = x + y + z$, $x < y < z$, $x + y > z$. Вытекающие из этих неравенств дополнительные ограничения, уменьшающие перебор вариантов, таковы: $x < n/3$, $n/3 < z < n/2$. Вот соответствующий алгоритм:

```
Алгоритм Число_треугольников
цел: m, n, x, y, z;
{  Запросить n;
   m := 0;
   Делать от x = 1 до n/3 − 1
   {  Делать от z = n/3 + 1 до n/2 − 1
      {  y := n − x − z;
         Если (x < y) и (y < z) и (x + y > z) то {  m := m + 1;  }
      }
```

}
Сообщить ”Количество треугольников”, m;
}

Лабораторная работа 7 рассчитана на два часа (но может потребовать и больше времени) и посвящена программной реализации составленных учащимися алгоритмов. Обращаем внимание учителя, что в сценарии лабораторной работы прогоны программ, соответствующих заданиям 7 и 8, поставлены в обратном порядке. Дело в том, что сама программа для задания 8 намного проще, чем для задания 7. Однако решение задачи 8 требует математического моделирования ситуации, в то время как задача 7 сразу сформулирована как математическая.

Второй концентр § 14 — эффективное доказательство существования алгоритмически неразрешимой задачи. В качестве задачи нами выбрана исторически первая задача, для которой была доказана ее алгоритмическая неразрешимость: задача определения по тексту алгоритма, является ли он конечным. В вузовских курсах такое доказательство проводится при том или ином уточнении понятия алгоритма (см. обсуждение § 7 и 9 в данной книге). Как правило, таким уточнением выступает машина Тьюринга или машина Поста. В этом случае задача выглядит так. Ясно, что любую функциональную схему для машины Тьюринга мы можем записать на ее ленту (внешний алфавит будет содержать внутренний алфавит машины и еще символы П, Л и Н). Спрашивается, существует ли функциональная схема какой-либо машины Тьюринга, которая способна по записи на ленте функциональной схемы распознать, закончит ли машина Тьюринга за конечное число шагов работу по этой функциональной схеме. Ответ отрицательный, и его доказательство фактически и приведено в объяснительном тексте § 14. При желании учитель может реконструировать доказательство и провести его с помощью машины Тьюринга. Поэтому данная задача получила название «Проблема останова машины Тьюринга».

Примыкающим к этой теме является задание 11. В нем предлагается составить алгоритм разработки алгоритмов. На первый взгляд может показаться, что это противоречит нередко высказываемому тезису (который, между прочим, звучит и в нашем учебнике), что составление алгоритмов — деятельность эвристическая (другими словами, творческая) и она не может быть передана компьютеру. Разумеется, сам тезис верен. Но учащиеся должны понимать, что в умственной человеческой деятельности всегда есть формализуемая часть, и вот ее-то можно поручить формальному исполнителю. Последнее утверждение — это основа всех исследований в сфере искусственного интеллекта.

В данной задаче по натуральному числу n компьютер должен генерировать текст алгоритма для автоматического устройства. Для простоты можно договориться, что алгоритм будет линейным, т. е. будет содержать только команды на выполнение действий данным автоматическим устройством и не будет иметь алгоритмических конструкций ветвления, цикла и подпрограмм. Собственно, таких команд две: «Добавить 1 л» и «Удвоить количество воды».

Перейдем теперь к обсуждению, каким должен быть наиболее короткий алгоритм для автоматического устройства. На наш взгляд, сначала полезно разобрать более простые задания, чтобы школьники освоились с этим исполнителем. Например, составить для этого исполнителя алгоритм получения чисел 4 и 5. Желательно уже здесь обсуждать, что один и тот же результат может быть получен алгоритмами с разным числом выполняемых действий. Самый «долгоработающий» алгоритм тот, в котором мы будем получать нужное число, используя только второе из допустимых действий — увеличение на 1.

Второе замечание состоит в том, что в решении этой задачи удобнее идти «от конца» к «началу», заменяя действия «удвоить» и «добавить» на «разделить на 2» и «уменьшить». Первый вопрос, который можно задать школьникам, может звучать так: «Из какого числа и каким действием мы можем получить число 100?» Возможны, конечно, два ответа: «Из числа 50 умножением на 2» и «Из числа 99 увеличением на 1». Поскольку хочется как можно скорее прийти к 1, то ясно, что надо выбирать первый вариант. И т. д. Если на каком-то шаге получится нечетное число, то тут выбирать не приходится: ясно, что его можно было получить только увеличением на 1. Вот требуемая цепочка от 1000 к 1:

$$1000 \xrightarrow{:2} 500 \xrightarrow{:2} 250 \xrightarrow{:2} 125 \xrightarrow{-1} 124 \xrightarrow{:2} 62 \xrightarrow{:2} 31 \xrightarrow{-1} 30 \rightarrow$$

$$\xrightarrow{:2} 15 \xrightarrow{-1} 14 \xrightarrow{:2} 7 \xrightarrow{-1} 6 \xrightarrow{:2} 3 \xrightarrow{-1} 2 \xrightarrow{-1} 1.$$

Читая цепочку справа налево, строим нужный алгоритм:

Добавить 1 л;
Добавить 1 л;
Добавить 1 л;
Удвоить количество воды;
Добавить 1 л;
Удвоить количество воды;
Добавить 1 л;
Удвоить количество воды;
Добавить 1 л;
Удвоить количество воды;

```
Удвоить количество воды;
Добавить 1 л;
Удвоить количество воды;
Удвоить количество воды;
Удвоить количество воды;
```

Приведем теперь алгоритм для компьютера:

```
Алгоритм Управление_автоматом
сим: C; цел: n, k;
{ Запросить n;
  C := "";
  Делать пока n > 1
  { Если (n mod 2 = 0) то
    { n := n/2;
      C := C + "2";
    }
    иначе
    { n := n − 1;
      C := C + "1";
    }
  }
  Сообщить "Добавить 1 л;";
  Делать от k := Len(C) до 1 с шагом −1
  { Если Часть (C, k, 1) = 2 то
    { Сообщить "Удвоить количество воды;"; }
    иначе { Сообщить "Добавить 1 л;"; }
  }
}
```

В результате будет напечатан требуемый алгоритм для автоматического устройства. То, что он имеет наименьшее число действий, учащимся обычно представляется очевидным. И доказательство этого факта мы предлагаем разбирать только с учениками, знакомыми с методами математической индукции.

Приведем доказательство. Возьмем произвольное натуральное число n. Обозначим через $a(n)$ самый короткий алгоритм получения числа n из 1 (ясно, что первый раз ничего, кроме действия «Долить 1 л», сделать нельзя), а через $r(n)$ — количество действий в этом алгоритме. Очевидно, что $r(1)=0$. Попытаемся выразить $r(n)$ через значение $r(n-1)$.

Пусть сначала n нечетно, т. е. $n = 2k + 1$. Ясно, что последним действием в алгоритме получения числа n было увеличение на 1, т. е. для получения n нет другого пути, кроме как из $n-1$. Это означает, что кратчайший алгоритм получения нечетного n получается из кратчайшего алгоритма получения $n-1$ приписыванием к нему еще одного действия. Таким образом, $r(2k+1) = r(2k) + 1$.

Пусть теперь $n = 2k$. Покажем методом математической индукции по k, что $r(2k) = r(k) + 1$. При $k = 1$ утверждение очевидно. Пусть теперь $k > 1$ и утверждение справедливо для $k-1$, т. е. $r(2k-2) = r(k-1)+1$. Рассмотрим алгоритм $a(k)$. Приписывая к нему «Умножить на 2», мы получаем алгоритм для вычисления $2k$. Следовательно, $r(2k) \leqslant r(k)+1$. Рассмотрим алгоритм $a(2k)$. Предположим, что последнее действие в этом алгоритме «Увеличить на 1». Тогда $r(2k) = r(2k-1)+1$. По выше доказанному $r(2k-1) = r(2k-2)+1$. В силу предположения индукции $r(2k-2) = r(k-1)+1$. Значит, $r(2k) = r(k-1)+3$. Однако ясно, что $r(k) \leqslant r(k-1)+1$, ибо k из $k-1$ получается одним действием. Поэтому оказалось, что $r(2k) \geqslant r(k)+2$, что противоречит указанному выше неравенству $r(2k) \leqslant r(k)+1$. Полученное противоречие показывает, что последним действием в алгоритме $a(2k)$ обязательно должно быть умножение на 2. Но тогда $r(2k) = r(k)+1$. Шаг индукции доказан, а вместе с ним и все утверждение.

Итак, мы доказали, что $r(n) = \begin{cases} r(n/2)+1, & \text{если } n \text{ четно;} \\ r(n-1)+1, & \text{если } n \text{ нечетно.} \end{cases}$

Из доказательства, кстати, видно, что в самом коротком алгоритме все действия по получению нужного числа, кроме первого действия — получения из 1 числа 2, определены однозначно.

Эта задача не включена в лабораторный практикум, но мы не исключаем, что у школьников может появиться желание выполнить ее компьютерную реализацию. Они могут смоделировать автоматическое устройство, о котором идет речь в задаче, и заставить его реально работать по программе, составленной компьютером. Обычно это вызывает положительный эмоциональный отклик.

Третий концентр данного параграфа — обсуждение того, что результат исполнения алгоритма зависит от свойств исполнителя этого алгоритма. На самом деле это пропедевтика того важного обстоятельства, что для любого компьютера имеются ограничения, связанные с конечностью разрядной сетки представления чисел, и вытекающие отсюда эффекты округления и переполнения. Сама эта тема подробно будет рассматриваться в 11 классе, однако важно, чтобы уже сейчас учащиеся понимали, что алгоритм — это математическая абстракция, в которой предполагается, что все действия выполняются точно. Задание 10 к § 14 как раз демонстрирует данный тезис применительно к такому простому вычислительному устройству, каким является калькулятор.

Обучение решению задач с помощью компьютера, представляющее собой одну из целей курса информатики, не может не включать в себя ознакомления хотя бы с некоторыми наиболее употребительными вычислительными методами. Среди них одним из важнейших является организация вычислений с помощью рекуррентных соотношений. Именно ему посвящена первая половина § 15. Объяснение материала можно строить в соответствии с текстом параграфа, хотя учитель вправе избрать другие примеры рекуррентных соотношений. Мы считаем, что программирование рекуррентных соотношений осваивалось учащимися в 8—9 классах, поэтому данный материал § 15 следует рассматривать как повторение и подготовку к восприятию более сложного материала — рекурсивных алгоритмов.

Перейдем к разбору задания 4, которое относится к теме рекуррентных соотношений.

Задание 4а является незначительной модификацией задачи, разбираемой в объяснительном тексте. Вот как может выглядеть соответствующий алгоритм:

```
Алгоритм Сумма_квадратов_обратных_чисел
вещ: S; цел: N;
{  S := 0;
   Делать от N := 1 до 100;
   {  S := S + 1/N²;
   }
   Сообщить S;
}
```

Для задания 4б возможны два достаточно простых решения: в первом — сначала разыскивается сумма всех чисел и затем вычисляется их среднее арифметическое; во втором — сразу составляется рекуррентное соотношение подсчета среднего арифметического. Приведем алгоритмы, реализующие эти подходы.

```
1) Алгоритм Среднее_арифметическое
   вещ: S, X; цел: J, N;
   {  Запросить N;
      S := 0;
      Делать от J := 1 до N;
      {  Запросить число X;
         S := S + X;
      }
         S := S/N;
         Сообщить S;
   }

2) Алгоритм Среднее_арифметическое_Р
   вещ: S, X; цел: J, N;
   {  Запросить N;
      S := 0;
      Делать от J := 1 до N;
      {  Запросить X;
         S := S + (X − S)/J;
      }
```

```
        Сообщить S;
    }
```

Задание 4в также допускает несколько вариантов решения. Иногда школьники делают ошибку, пытаясь написать в программе выражение $(-1)^{(N-1)}$. Мы приведем короткий, но не вполне очевидный алгоритм, основанный на том наблюдении, что вычисляемая в этом задании сумма положительна при любом N. Чтобы это заметить, достаточно сгруппировать слагаемые парами: 1-е со 2-м, 3-е с 4-м и т. д. (при нечетном N останется еще одно положительное слагаемое). Приведем этот алгоритм.

```
Алгоритм Алгебраическая_сумма
вещ: S; цел: J, N;
{  Запросить N;
   S := 0;
   Делать от J := 1 до N;
   {  S := 1/J–S;  }
   S := ABS(S);
   Сообщить S;
}
```

Здесь через ABS(S) обозначена абсолютная величина числа S; в большинстве языков программирования принято именно такое обозначение этого оператора.

Рекурсия также рассматривается нами на примере классической задачи «Ханойские башни». В заданиях учащиеся сначала выполняют рекурсивные алгоритмы, и только потом им предлагается самим составлять подобные алгоритмы.

В задании 5 алгоритм переписывает каждое слово в обратном порядке. Поэтому ответы таковы: а) срам; б) ропот; в) дорого; г) шалаш.

В задании 6 алгоритм ФОК(x) подсчитывает наименьшее общее кратное первых x чисел. Для установления этого факта надо воспользоваться соотношением НОК$(m, n) = mn/$НОД(m, n).

В задании 7 несложно составить обычный циклический алгоритм для нахождения суммы квадратов цифр натурального числа. Мы, однако, предлагаем составить именно рекурсивный алгоритм, чтобы в этом простом случае учащиеся получили первые навыки составления рекурсивных алгоритмов. Вот возможное решение задачи:

```
Алгоритм Сумма_квадратов_цифр
цел: n;
{  Запросить n;
    n := СКЦ(n);
    Сообщить n;
}
Функция СКЦ (арг цел: n): цел
цел: n;
```

```
{  Если n < 10 то { знач : = n*n;}
   иначе  {  знач: =(n mod 10)*(n mod 10) + СКЦ(n div 10);  }
}
```

Задание 9 — это задача, которую должен уметь решать любой участник олимпиады по информатике. Как правило, эта задача возникает в качестве некоторого шага при пошаговой детализации какой-либо другой задачи, поэтому может потребоваться представить ее результат в форме, отличной от той, в которой дадим мы (например, в виде двумерного массива, где в каждой строке записана одна из перестановок). В частности, если учитель считает целесообразным, можно выполнить это задание после § 16. Приведенный ниже алгоритм использует два массива и механизм глобальных переменных.

```
Алгоритм Перестановки
цел: n, k; лог: A[1: n]; цел: B[1: n];
глоб: n, A[1: n]; B[1: n];
{  Запросить n;
   Делать от k : = 1 до n
   {  A(k) : = Ложь;  }
   Вызвать ПП(1);
}
Алгоритм ПП (арг цел: s)
цел: k;
{  Если s = n + 1 то
   {  Делать от k : = 1 до n
      {  Сообщить B(k);
      }
   }
   иначе
      {  Делать от k : = 1 до n
      {  Если (не A(k)) то
      {  A(k) : = Истина;
         B(s) : = k;
         Вызвать ПП(s + 1);
         A(k) : = Ложь;
      }
    }
   }
}
```

При решении задачи 10 можно договориться, что диски лежат на стержнях № 1 и № 2, причем на стержне № 1 лежит n дисков, а на стержне № 2 лежит m дисков. Результирующую пирамиду дисков будем получать на стержне № 3, который первоначально свободен. Поскольку придется сравнивать диски с разных стержней, то просто знания упорядоченности дисков на каждом стержне недостаточно, нужна еще информация о радиусе каждого диска. Ниже приведено описание алгоритма (в котором, правда, проведена неполная пошаговая детализация, но то, что оставлено читателю, представляется весьма легким, а скрупулезное описание этих шагов, как нам кажется, затемнило бы основную идею).

```
Алгоритм Перенос_2 (арг: n, m, a, b; рез: башня дисков
на стержне № c)
{  Если n = 0 то { Вызвать Перенос_1 (m, b, c) };
   иначе
   {  Если (верхний диск на стержне № a больше радиу-
      са последнего диска на стержне № b) то
      {  Вызвать Перенос_1 (m, b, a);
         Вызвать Перенос_1 (m + n, a, c);
      }
      иначе
      {  На диске № b найти первый диск, радиус которо-
         го больше, чем радиус самого верхнего диска на
         стержне № a;
         В переменной k запомнить число дисков на стерж-
         не № b, радиус которых меньше, чем радиус само-
         го верхнего диска на стержне № a;
         Вызвать Перенос_1 (k, b, c);
         Переложить диск со стержня № a на стержень № b;
         Вызвать Перенос_1 (k, c, b);
         Вызвать Перенос_2 (n − 1, m + 1, a, b);
      }
   }
}

Алгоритм Перенос_1 (арг: k, x, y;
рез: k дисков перенесено со стержня № x на стержень № y)
{  Если k = 1; то
   { Перенести диск со стержня № x на стержень № y;  }
   иначе
{  Вызвать Перенос_1 (k − 1, x, z);
Перенести диск со стержня № x на стержень № y;
Вызвать Перенос_1 (n − 1, z, y);
              }
}

Алгоритм Ханой_2
{  Запросить n;
   Запросить m;
   Запросить набор радиусов дисков на стержне № 1;
   Запросить набор радиусов дисков на стержне № 2;
   Вызвать Перенос_2 (n, m, 1, 2);
}
```

Нетрудно видеть, что на самом деле и сама задача, и предложенный алгоритм — это вариации на тему алгоритма сортировки слиянием, но не для массивов, а для двух стеков (для массивов разрешен прямой доступ к элементам, поэтому там сортировку проводить намного легче). В этом алгоритме рекурсия оказалась распространенной на два вспомогательных алгоритма, причем один из них обращается к другому.

Лабораторная работа 8 к § 15 ориентирована на 3 часа. В сценарий этой работы не включено программирование заданий 9 и 10, поскольку они весьма трудоемки — каждое из них способно забрать академический час — и используют механизм массивов. Но сильным учащимся следует предложить запрограммировать хотя бы задачу 9 (в ходе компьютерного практикума к § 16).

Массивы и основные алгоритмы их обработки, как правило, в достаточном объеме изучаются в 8—9 классах. Поэтому материал § 16 можно рассматривать как повторение. Отметим лишь, что во многих других школьных учебниках двумерный массив принято называть табличной формой организации данных, табличной величиной или просто таблицей. Однако с введением электронных таблиц как активного элемента информационных технологий эти термины стали мешать друг другу. Поэтому мы считаем целесообразным вернуться к исходному программистскому термину «массив», тем более что массив может иметь размерность, выражаемую любым натуральным числом, а не только числом 2. Да и языки программирования имеют дело, конечно, с массивами, а не с мифическими «табличными величинами». Чтобы приучить учащихся к данной абстракции, в заданиях 3 и 4 им предлагается обработать трехмерные массивы, для которых уже нет такой наглядной интерпретации, как таблица.

Задание 1 нацелено на отработку основных операций с элементами массива. В двойном цикле определяется количество 1 и −1 в заданном массиве. Для первого массива это соответственно 7 и 5, поэтому по окончании исполнения алгоритма переменная A имеет значение 7, а переменная B — значение 19. Для второго массива количество 1 и −1 одинаково, поэтому присваивания в обоих условных операторах не исполняются, так что ответ $A = B = 5$. Наконец, для третьего массива условие в первом ветвлении ложно, так что ответ таков: $A = 5$, $B = 12$.

Задание 2 стандартное и в комментариях не нуждается.

Задание 3 направлено на отработку алгоритмов поиска максимальных и минимальных элементов в массивах. Эти алгоритмы тоже стандартны, мы для справки приведем только алгоритм, решающий задачу 3а.

```
Алгоритм Поиск_максимума
цел: k, m, n; вещ: Max, A[1:10, 1:10, 1:10];
{ Делать от k := 1 до 10
   { Делать от m := 1 до 10
      { Делать от n := 1 до 10
         { Запросить A(k, m, n);
         }
      }
}
Max := A(1, 1, 1);
{ Делать от k := 1 до 10
   { Делать от m := 1 до 10
      { Делать от n := 1 до 10
         { Если A(k, m, n) > Max то {  Max := A(k, m, n);  }
         }
      }
   }
}
```

Необходимо обсудить с учащимися то обстоятельство, что представленный выше алгоритм находит только один максимальный элемент массива. Этим задача 3а отличается от задачи 3б, где требуется указать индексы *всех* минимальных элементов. Несмотря на внешнюю схожесть пунктов *а* и *б*, последнее гораздо сложнее — ведь определить индексы всех минимальных элементов можно только после того, как найдено значение минимума. Поэтому при проведении лабораторной работы 9 в тестовых примерах необходимо предусмотреть ситуацию, когда в массиве имеется несколько одинаковых минимальных элементов. Чтобы подготовить соответствующий тестовый массив, можно воспользоваться тем же алгоритмом создания массива, который приведен в этой лабораторной работе для данной задачи, только вместо оператора

$A(I, J, K) := 2*\mathrm{COS}\,(I + 2*J) + 7*\mathrm{SIN}\,(I*I/K);$

надо записать оператор

$A(I, J, K) := \mathrm{INT}\,(2*\mathrm{COS}\,(I + 2*J) + 7*\mathrm{SIN}\,(I*I/K));$

При этом массив *A* можно объявить целочисленным.

Задание 4 не должно, на наш взгляд, вызвать у учащихся каких-либо затруднений, и его вполне можно выполнить дома.

Прежде чем приступить к обсуждению задания 5, нужно, чтобы учащиеся вспомнили формулу расстояния между двумя точками на плоскости, заданными своими координатами. Пошаговая детализация данной задачи показывает, что она разбивается на три подзадачи:

1) генерация всех пар точек;
2) поиск максимального расстояния среди этих пар;
3) поиск всех пар, для которых расстояние максимально.

Вот алгоритм, который решает эту задачу:

```
Алгоритм Максимально_удаленные_пары
цел: k, m, n; вещ: Max, D2, M[1 : 2, 1 : 100];
{ Делать от m := 1 до 100
   { Сообщить "Введите абсциссу точки";
     Запросить M(1, m);
     Сообщить "Введите ординату точки";
     Запросить M(2, m);
   }
   Max := 0;
   Делать от k := 1 до 99
   { Делать от m := k + 1 до 100
     { D2 := (M(1, k) − M(1, m))^2 + (M(2, k) − M(2, m))^2;
     Если D2>Max то { Max := D; }
     }
   }
   Делать от k := 1 до 99
   { Делать от m := k + 1 до 100
     { Если
     Max = (M(1, k) − M(1, m))^2 + (M(2, k) − M(2, m))^2
     то
     { Сообщить "(", M(1, k), ";", M(2, k), "),(", M(1, m),
        ";", M(2, k), ")";
        }
     }
   }
}
```

Отметим, что мы в этом алгоритме подсчитываем не само расстояние между точками, а его квадрат. Нахождение наибольшего элемента от этого никак не страдает, зато каждый раз на одну операцию меньше, а главное, если координаты заданы целыми числами, то все вычисления можно провести в переменных целого типа (что мы и советуем сделать).

Задание 6 по идее решения близко к заданию 5. Учащиеся только должны догадаться, что индексы элементов массива можно рассматривать как координаты центров таблицы. Помочь в этом может рисунок 17, на котором показано, как клеткам таблицы сопоставляются клетки на координатной плоскости, причем первый индекс элементов возрастает не сверху вниз, а снизу вверх.

Задания 7 и 8 представляются нам довольно стандартными, и мы рекомендуем задать их домой с последующей проверкой на компьютерном практикуме. В задании 8 имеется один подводный камень — результат нельзя сразу размещать в том же массиве, что и исходные данные. Поэтому требуется создать вспомогательный (рабочий) массив, а затем из него переписать результаты в исходный массив. На лабораторной работе можно предложить учащимся данное преобразование массива выполнить несколько раз подряд. Прослеживая эволюцию массива, учащиеся могут прийти к

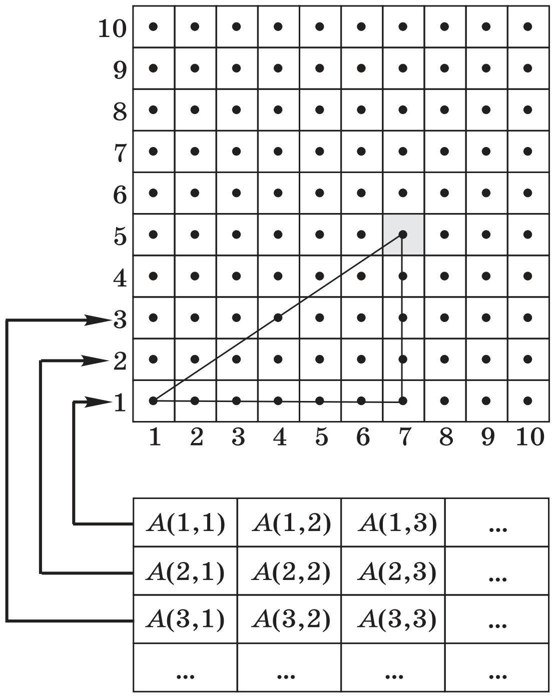

Рис. 17

выводу, что значения элементов становятся все ближе к общему для них всех числу. Пусть попробуют догадаться к какому именно, а в математическом классе можно предложить доказать (или опровергнуть) возникшую гипотезу. Правильный ответ таков: к среднему арифметическому всех чисел исходного массива. Более тонкий вопрос: не может ли случиться так, что процесс зациклится, т. е. будут получаться массивы, состоящие из неравных чисел и отличающиеся друг от друга лишь кольцевым сдвигом? Хотя ответ положительный, его доказательство для произвольных действительных чисел выходит за рамки элементарной математики. Если же исходный массив содержит только целые числа, то имеется доказательство, доступное школьникам даже 9 класса.

О проведении лабораторной работы 9 мы довольно много уже сказали выше, теперь отметим только, что она рассчитана на 3 учебных часа.

Сильным учащимся можно предложить дополнительно следующую задачу:

Задание 7. Задача о ходе конем.

На заданной клетке шахматной доски стоит король. Составьте алгоритм, позволяющий узнать наименьшее число ходов шахматного коня, за которое можно перейти с клетки в левом нижнем углу на клетку, занятую королем.

Естественно, учащимся надо напомнить, что за один ход шахматный конь смещается на 2 клетки в горизонтальном или вертикальном направлении и соответственно на 1 клетку в вертикальном или горизонтальном направлении.

Идея решения основывается на том, чтобы создать массив, моделирующий шахматную доску, и в каждый элемент массива записать, за какое наименьшее число ходов добирается конь до клетки, соответствующей этому элементу. Пусть этот двумерный массив имеет имя A.

Прежде всего заметим, что на любое поле можно добраться не более чем за 63 хода. Поэтому элементу массива, отвечающему за поле с королем, мы присвоим число 64. На начальном поле конь стоит сразу, поэтому элементу $A(1, 1)$ присвоим значение 0. Значение 1 присвоим тем элементам, которые соответствуют полям, до которых можно попасть из начального поля за 1 ход. Теперь в двойном цикле просмотрим все элементы массива и, обнаружив значение 1, запишем 2 в тот элемент массива, который соответствует полю, куда еще не ступала нога лошади, но можно за один ход добраться из поля со значением 1. Разумеется, если этот элемент уже имеет значение 64, то ничего записывать не надо, следует тут же сообщить ответ к задаче. Затем так же поступаем с элементами уже имеющими значение 2 и порождающими значение элементов, равное 3. И т. д. После такого неформального описания можно приступать к составлению алгоритма.

```
Алгоритм Ходом_коня
цел: x, y, k, m, n, A[1...8, 1...8]; лог: q;
{ Запросить x, y;             (*запрашиваются координаты поля, где стоит король*)
  Делать от k := 1 до 8
  { Делать от m := 1 до 8
    A(k, m) := 0;
  }
  A(x, y) := 64;
  Если A(1, 1) = 64 то { Сообщить "За 0 ходов"; }
  иначе
  { Если A(2, 3) = 64 или A(3, 2) = 64 то
    { Сообщить "За 1 ход"; }
    иначе
    { A(2, 3) := 1;
```

```
        A(3, 2) := 1;
      }
    }
q := False;
Делать от n := 1 до 63
  { Делать от k := 1 до 8
    { Делать от m := 1 до 8
      { Если (A(k, m) = n) то
        { Если (k > 1 и m > 2) то
          { Если A(k − 1, m − 2) = 64 то { q := True; }
            иначе
            { Если A(k − 1, m − 2) = 0 то
              { A(k − 1, m − 2) := n + 1; } }
}
Если (k > 2 и m > 1) то
{ Если A(k − 2, m − 1) = 64 то { q := True; }
  иначе
  { Если A(k − 2, m − 1) = 0 то
    { A(k − 2, m − 1) := n + 1; } }
}
Если (k < 8 и m < 7) то
{ Если A(k + 1, m + 2) = 64 то { q := True; }
  иначе
  { Если A(k + 1, m + 2) = 0 то
    { A(k + 1, m + 2) := n + 1; } }
}
Если (k < 7 и m < 8) то
{ Если A(k + 2, m + 1) = 64 то { q := True; }
  иначе
  { Если A(k + 2, m + 1) = 0 то
  { A(k + 2, m + 1) := n + 1; } }
}
Если (k < 8 и m > 2) то
{ Если A(k + 1, m − 2) = 64 то { q := True; }
  иначе
  { Если A(k + 1, m − 2) = 0 то
    { A(k + 1, m − 2) := n + 1; } }
}
Если (k < 7 и m > 1) то
{ Если A(k + 2, m − 1) = 64 то { q := True; }
  иначе
  { Если A(k + 2, m − 1) = 0 то
  { A(k + 2, m − 1) := n + 1; } }
}
Если (k > 1 и m < 7) то
{ Если A(k − 1, m + 2) = 64 то { q := True; }
  иначе
  { Если A(k − 1, m + 2) = 0 то
```

```
{  A(k − 1,  m + 2) := n + 1;   }   }
}
   Если (k > 2 и m < 8) то
   {  Если A(k − 2,  m + 1) = 64  то  {  q := True;  }  }
      иначе
      {  Если A(k − 2,  m + 1) = 0  то
         {  A(k − 2,  m + 1) := n + 1;   }  }
         }
      }
   }                                (*разметка очередного уровня*)
}
Если (q = True) то
      {  Сообщить ”За”, n, ”ходов”;
         n := 64
         }
      }
}
```

Размер 8×8 выбран не только в угоду шахматной фабуле, но и потому, что для такой доски известно, что из углового поля ходом шахматного коня можно прийти на любое другое поле этой доски. Для досок других размеров это уже не так: скажем, для доски размером 3×3 никогда нельзя попасть на центральное поле. Тем не менее задачу можно формулировать для досок любых размеров, только в качестве результата алгоритм должен выдавать либо наименьшее число ходов, либо сообщение, что данное поле недостижимо.

В рассмотренной задаче мы имеем дело с так называемым волновым алгоритмом. Это один из базовых алгоритмов при решении многих задач, основанных на применении теории графов. Систематически волновой алгоритм будет рассматриваться в 11 классе.

В созданной программе значительное место занимают проверки того, не находится ли за пределами доски очередное поле, куда мог бы сходить конь. Избавиться от них позволяет такой прием. Увеличим размеры массива, добавив с каждой стороны по полосе из двух клеток. Ходы из этих клеток мы делать не будем, но рассматривать ход в них разрешается. Тогда программа значительно упрощается.

```
Алгоритм Ходом_коня_м
цел: x, y, k, m, n, A[−1...10, −1...10]; лог: q;
{  Запросить x, y;             (*запрашиваются координа-
                               ты поля, где стоит король*)
   Делать от k := −1 до 10
      {  Делать от m := − 1 до 10
         A(k, m) := 0;
      }
   A(x, y) := 64;
```

```
Если A(1, 1) = 64 то { Сообщить ”За 0 ходов”; }
иначе
   { Если A(2, 3) = 64 или A(3, 2) = 64 то { Сообщить
      ”За 1 ход”; }
      иначе
      { A(2, 3) := 1;
         A(3, 2) := 1;
      }
   }
q := False;
Делать от n := 1 до 63
   { Делать от k := 1 до 8
      { Делать от m := 1 до 8
         { Если (A(k, m) = n) то
            { Если A(k − 1, m − 2) = 64 то { q := True; }
               иначе
               { Если A(k − 1, m − 2) = 0 то
                  { A(k − 1, m − 2) := n + 1; }
               }
            Если A(k − 2, m − 1) = 64 то { q := True; }
            иначе
            { Если A(k − 2, m − 1) = 0 то
            { A(k − 2, m − 1) := n + 1; }
            }
            Если A(k + 1, m + 2) = 64 то { q := True; }
            иначе
            { Если A(k + 1, m + 2) = 0 то
                  { A(k + 1, m + 2) := n + 1; }
            }
            Если A(k + 2, m + 1) = 64 то { q := True; }
            иначе
            { Если A(k + 2, m + 1) = 0 то
                  { A(k + 2, m + 1) := n + 1; }
            }
            Если A(k + 1, m − 2) = 64 то { q := True; }
            иначе
            { Если A(k + 1, m − 2) = 0 то
                  { A(k + 1, m − 2) := n + 1; }
            }
            Если A(k + 2, m − 1) = 64 то { q := True; }
            иначе
            { Если A(k + 2, m − 1) = 0 то
                  { A(k + 2, m − 1) := n + 1; }
            }
            Если A(k − 1, m + 2) = 64 то { q := True; }
            иначе
            { Если A(k − 1, m + 2) = 0 то
                  { A(k − 1, m + 2) := n + 1; }
```

```
    }
            Если A(k − 2, m + 1) = 64 то
              { q := True; }
            иначе
            { Если A(k − 2, m + 1) = 0 то
              { A(k − 2, m + 1) := n + 1; }
            }
          }                    (*разметка очередного уровня*)
    }
    Если (q = True) то
    { Сообщить "За", n, "ходов";
      n := 64;
    }
  }
}
```

Тема 7. В § 17 мы предлагаем познакомить учащихся с еще одним важным вычислительным методом — методом деления пополам. Приоритетность этого метода прежде всего в его достаточной универсальности — класс задач, для решения которых он может применяться, весьма широк, и в первую очередь он используется для решения уравнений. Важно и то, что эта тема подготавливает изучение следующей темы — измерение количества информации. Кроме того, этот метод опирается на уже изученный метод рекуррентных соотношений — каждое последующее приближение к корню вычисляется на основе предыдущего. Полезно также провести обсуждение с учащимися роли точных и приближенных методов решения. Здесь учащихся важно подвести к выводу, что точные методы превалируют в теоретических исследованиях; для практики же, где любые измерения могут быть произведены лишь приближенно, требование абсолютной точности является чрезмерным. Тем самым при решении практических задач, в том числе с помощью компьютера, приоритетны приближенные методы. Разумеется, применение приближенных методов требует учета влияния возникающих погрешностей, поэтому результаты, полученные приближенными методами, должны проверяться еще более тщательно, чем при использовании точных методов.

Мы не видим существенных методических проблем в изучении метода деления пополам, поэтому прокомментируем лишь задания к § 17 и остановимся на особенностях выполнения соответствующей лабораторной работы.

Задание 4а предлагает иной, нежели метод деления пополам, алгоритм нахождения корня на заданном отрезке с заданной точностью. В нем обсуждается простейший метод решения уравнения, когда интервал, на котором располагается корень, разбивается на равные части не больше, чем требуемая точность, и в точках разбиения вычисляется значение функции. Концы того интервала, где происходит смена знака функции, как раз и дают приближенное значение корня с заданной точностью. В данном случае, поскольку шаг увеличения неизвестной переменной равен 0,1, точность, с которой будет найден корень, тоже равна 0,1.

Разумеется, строгого обоснования того, что приведенный в задании многочлен имеет корень на отрезке [0; 1], учащиеся дать не могут — для этого требуется понятие непрерывности функции. Тем не менее у учащихся должно быть интуитивное понимание, что график функции $y = x^3 + 3x - 1$ не может перейти из нижней координатной полуплоскости (при $x = 0$ значение функции равно -1) в верхнюю полуплоскость (при $x = 1$ значение функции равно 3) без пересечения оси абсцисс.

Обсудим теперь задание 4б. Небольшой перебор возможных перестановок двух строк показывает, что единственный вариант, при котором компьютер все-таки сообщит корень, — это перестановка двух последних строк алгоритма. В этом случае компьютер сообщит, быть может, и не одно число, но последнее из них совпадет с тем корнем, который сообщил бы компьютер в исходном алгоритме.

В задании 4в легко показывается, что метод деления пополам нужную точность дает за 4 шага (а не за 9, как может оказаться в алгоритме равномерного деления отрезка).

При изучении информатики на базовом уровне мы предлагаем ограничиться тем сценарием лабораторной работы 10, который приведен в учебнике. Однако в профильных классах полезно предложить воспользоваться данным методом и составленной программой для других уравнений, например для уравнения $x^3 - 3x + 3 = 0$ или/и уравнения $2^x = 3x$. Приступая к решению таких задач, учащиеся должны в первую очередь определить отрезок, внутри которого заключен корень. Для уравнения $x^3 - 3x + 3 = 0$ можно взять следующий отрезок, в котором наверняка находится корень: [− 3; − 2]. Действительно, на левом конце этого отрезка функция $x^3 - 3x + 3$ принимает отрицательное значение, а на правом — положительное. Во втором уравнении надо сначала перенести в одну часть (например, левую) все члены уравнения, содержащие x, после чего для полученной функции, записанной в левой части уравнения, нетрудно указать соответствующий отрезок, на концах которого эта функция принимает разные знаки. Таким отрезком может служить [0; 1]. Выбор нужного отрезка удобно производить, построив предварительно графики соответствующих функций. Это, разумеется, не отменяет последующего обоснования, что выбранные по графику отрезки действительно годятся. Данное задание может быть усложнено (в классах с хорошей математической подготовкой) предложением доказать, что у каждого из уравнений есть только один корень.

В конце главы мы снова возвращаемся к главному понятию курса — понятию «информация». Это понятие многогранно и, без сомнения, весьма сложно. На наш взгляд, методически целесообразно знакомить учащихся с этим понятием в течение всего курса информатики, который изучается в школе (при этом все равно что-то еще останется и вузовскому курсу). Многие грани понятия «информация» уже были раскрыты в предшествующих темах. Мы говорили об информационных процессах и информационных ресурсах, об информационной этике и опасностях, связанных с доступностью информации, о тех благах, которые обусловлены информационными возможностями, предоставляемыми человеку. Мы обсуждали (на языке моделей) вопросы существенности и адекватности той информации, которая используется при решении жизненных задач. Учащиеся знакомы уже и с единицами измерения информационного объема. Теперь можно связать эти единицы с понятием количества информации. Сделано это в довольно традиционной манере; методическая новинка — опора на метод половинного деления, изложенный непосредственно перед этим в качестве метода решения уравнений, хотя в последних учебниках (возможно, не без влияния учебников данного авторского коллектива) этот прием становится достаточно стандартным. Здесь же вводится техническое понятие приемника информации, которое отличается от обыденного понимания тем, что мы не требуем от приемника осмысления той информации, которую он получил. Речь идет, можно сказать, о *формальном приемнике* информации.

В начале урока полезно вспомнить о единицах измерения информационного объема и обратить внимание на технологический характер этого определения. Одна и та же фраза, записанная на магнитный носитель дважды, будет занимать в 2 раза больше места, и, значит, ее информационный объем будет в 2 раза больше, чем информационный объем той же фразы, записанной однократно. Но ясно, из такого повторения мы не получим больше информации, чем из фразы, произнесенной однократно. Точно так же, кадр на экране дисплея имеет тот самый огромный информационный объем, который подсчитывался в § 4, даже если одна половина кадра просто белая, а другая — черная. Итак, имеется принципиальное различие между термином «информационный объем сообщения» и интуитивным пониманием выражения «количество информации». После этого можно подвести учащихся к той мысли, что под получением информации мы можем понимать получение ответов на интересующие нас вопросы. От этого тезиса уже легко перейти к обсуждению игры «Угадайка».

Возможно, есть школьники, которые эту игру знают. Тогда естественно сразу сформулировать стратегию игры и указать минимальное число вопросов, гарантирующих угадывание задуманного числа. Если же школьникам эта игра неизвестна, то полезно предложить им несколько раз сыграть друг с другом, чтобы найти оптимальную стратегию и указать необходимое число задаваемых вопросов. Некоторое время надо посвятить обоснованию найденной стратегии. Для этого можно воспользоваться объяснительным текстом к § 18, составляющим основу данной темы, и заданием 4 к тому же параграфу (разбор этого задания приводится ниже). После обсуждения игры можно сформулировать основной тезис: получение одного бита информации означает уменьшение неопределенности вдвое. И уже после этого вскрывается взаимосвязь между техническим и семантическим подходами к измерению количества информации. Фактически — и на это следует обратить внимание учащихся — оказывается, что задача измерения количества информации является плохо поставленной: если считать несущественным тот фактор, что информация имеет смысловое содержание, то в измерении количества информации в данном сообщении мы приходим к измерению информационного объема, т. е. технологическому пониманию этой величины. Если же данный фактор считать существенным (что, конечно, более естественно), то о количестве информации, извлекаемой из сообщения, мы можем судить по уменьшению неопределенности рассматриваемой ситуации после получения данного сообщения. Как и ранее, здесь мы снова сталкиваемся с необходимостью решать вопрос на основе модельного подхода. Далее в учебнике обсуждаются другие модели измерения количества информации.

Связь между подходом к измерению количества информации на основе уменьшения неопределенности и технологическим подходом, при котором измеряется объем двоично закодированного сообщения, демонстрируется на математическом материале — игре «Угадайка». На самом деле играть в такую игру можно (и действительно в нее увлеченно играют) не только на числах, но и на любых объектах. Можно загадывать животное или литературного героя, разгадывать причины каких-либо событий. Как показывает метод деления пополам, наиболее информативны те вопросы, которые делят множество вариантов на две примерно равные половины. Например, если задуман какой-то объект, то эффективнее первым вопросом узнать, живой он или неживой, чем спросить, не является ли этот объект предметом мебели. Для желающих подробнее обсудить с учащимися возможность выяснения информации с помощью вопросов с бинарным ответом, мы приводим примеры нескольких заданий.

1. Чем сильнее его бьют по одному и тому же месту, тем он лучше выполняет свою функцию. О чем идет речь?

2. Я встретил ее случайно. Я пытался ее достать, но она уходила все дальше и дальше. Я принес ее домой в руке. Что это?

3. В небольшой город приехал в командировку инженер, поселился в одноместный номер в гостинице. Вечером лег спать, но полночи не мог заснуть. Потом встал, набрал телефонный номер, дождался, когда поднимут трубку, ничего не сказал, положил трубку, лег и спокойно заснул. Почему?

4. Он совершил кражу, и хотя об этом все узнали, люди ему благодарны. Кто это?

Ответы: 1. О гвозде. 2. Заноза. 3. Из соседнего номера доносился храп, мешавший спать; после звонка сосед проснулся, и храп прекратился. 4. Прометей.

Далее в учебнике прослеживается связь между уменьшением неопределенности при ответах на бинарные вопросы (так обычно называют вопросы, имеющие только два варианта ответа) и двоичным кодированием. Если ответ «Да» кодировать как 1, а ответ «Нет» кодировать как 0, то последовательность вопросов порождает последовательность нулей и единиц так, как это продемонстрировано в учебнике.

Еще одна модель измерения количества информации — это подход, предложенный А. Н. Колмогоровым. В классах базового уровня изучения информатики его можно опустить.

Прокомментируем теперь задания к § 18. В задании 2 для числа 128 ответ очевиден: 7 вопросов. В случае 100 чисел ответ тот же, поскольку 6 вопросов гарантированно определяют число, самое большее из 64 претендентов.

В задании 3а требуется узнать одно из 32 нечетных чисел. Значит, все по той же формуле требуется 5 вопросов, чтобы гарантированно угадать задуманное число. В задании 3б также требуется определить одно из 32 чисел, меньших 1000 и являющихся квадратами целых чисел, — это 0^2, 1^2, 2^2, ..., $31^2 = 961$ (заметим, что 32^2 уже больше 1000). Значит, и здесь хватит пяти вопросов, чтобы наверняка угадать число.

В задании 4 надо сначала понять, почему для восьми чисел не хватит двух вопросов. Действительно, какой бы вопрос ни был задан, один ответ «Да»—«Нет» охватывает не менее четырех чисел. Но тогда ясно, что если задуманное число попало как раз в эту группу, то вторым вопросом никак не удастся определить, какое из чисел этой группы было задумано. Теперь уже нетрудно показать, что для шестнадцати чисел не хватит трех вопросов: один ответ «Да»—

«Нет» охватывает не менее восьми чисел, а на них осталось только два вопроса. Полезно такими же рассуждениями показать, что даже для задуманных девяти чисел нельзя гарантировать угадывание за три вопроса.

В задании 5 химический антураж чисто внешний; фактически надо выбрать одно число (химический номер элемента) из менее чем 128 чисел. Значит, вполне достаточно семи вопросов.

В задании 6б вывод тот же, какой уже делался раньше: знания, которыми обладает приемник информации, позволяют уменьшать объем передаваемого сообщения с сохранением содержательной информации в этом сообщении.

Приведенный в конце главы 2 тест предназначен для организации самопроверки. Ниже мы предлагаем аналогичный тест для выполнения в классе за счет часов из резерва учителя (если учитель считает целесообразным проведение такого теста).

Контрольный тест по материалу главы 2

Часть 1. При выполнении предложенных ниже заданий укажите номер правильного ответа.

А1. Дан фрагмент однотабличной базы данных (в первой строке указаны имена атрибутов; в скобках приведены дополнительные сведения об атрибутах).

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 < 20 000 ***и*** Высота > 100 или Глубина < 200», равно:

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

А2. Локальные переменные вспомогательного алгоритма — это:

1) переменные, используемые только во вспомогательном алгоритме;

2) аргументы вспомогательного алгоритма;

3) переменные, указанные в заголовке вспомогательного алгоритма;

4) переменные, посредством которых передаются данные во вспомогательный алгоритм.

А3. Размерность массива — это:

1) наименьшее значение какого-либо индекса массива;

2) наибольшее значение какого-либо индекса массива;

3) количество элементов в массиве;

4) в пунктах 1—3 нет правильной формулировки.

А4. Дан алгоритм:

```
Алгоритм
вещ: X, Y, M[1 : 5, 1 : 5]; цел: K;
{  X := 0;
   Y := 0;
   Делать от K := 1 до 5
   {  X := X + M(K, K);
      Y := Y + M(K, 6 − K);
   }                              (*конец цикла*)
   Если (X > Y) то
      {  Сообщить M(1, 3) + M(5, 3);
      }
   иначе
      {  Сообщить M(3, 1) + M(3, 5);
      }                           (*конец ветвления*)
}
```

Этот алгоритм был исполнен для массива M, описанного таблицей.

−1	5	3	1	−1
6	2	−3	7	0
1	−4	2	−5	3
2	2	−5	4	1
0	6	−4	−1	3

После исполнения сообщено:

1) −1; 2) 10; 3) 4; 4) другое число.

A5. 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 X + Y;
}
```

Этот алгоритм был исполнен для массива M, описанного 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, нежели указанное в пунктах 1—3.

A6. Некто задумал натуральное число, не превосходящее 256, а затем сообщил: «Задуманное мною число является кубом целого числа». Количество бит информации, содержащейся в этом сообщении, равно:

1) 3; 2) 5; 3) 8; 4) другому, нежели в пунктах 1—3, числу.

A7. Дан алгоритм:

```
Алгоритм
вещ: 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 $K = 2000$ этот алгоритм исполняется в течение 0,3 с. На том же компьютере этот алгоритм при $K = 100\,000$ будет исполняться:

1) ≈ 7,5; 2) ≈ 15 с; 3) ≈ 125 с; 4) ≈ 12,5 мин.

Часть 2. При выполнении предложенных ниже заданий запишите ответ в виде числа.

В1. В квадратной таблице размером 8×8 четверть клеток закрашена красным цветом, еще четверть — синим, еще четверть — зеленым, остальные оставлены белыми. Требуется установить, где они располагаются (т. е. узнать номер ряда и номер столбца, на пересечении которых находится клетка). Получена информация, что клетка закрашена синим цветом. Количество полученной информации (в битах) равно _______.

В2. Дан алгоритм и подпрограмма-функция.

Алгоритм
цел: K, M;
{ **Запросить** M;
$K :=$ ФОКУС (M);
Сообщить M/K;
}
Подпрограмма-функция ФОКУС (**арг цел**: M): **цел**
{ **Если** $M < 10$ **то** { **знач**: $= M$; }
иначе { **знач**: $= (M \text{ mod } 10) + 2*$ ФОКУС (M div 10); }
}

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

В3. Дан алгоритм, в работе которого используются две подпрограммы-функции.

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

Подпрограмма-функция A (**арг цел**: M): **цел**
{ **Если** $M < 3$ **то** { **знач**: $= M$; }
иначе { **знач**: $= \text{Int}(M/3) + B(M-1)$; }
}

Подпрограмма-функция B (**арг цел**: N): **цел**
{ **Если** $N < 3$ **то** { **знач**: $= N$; }
иначе { **знач**: $= 3*N - A(N-2)$; }
}

Результат работы алгоритма при $M = 10$ равен _______.

Ключи к тесту

Задание	А1	А2	А3	А4	А5	А6	А7	В1	В2	В3
Ответ	2	1	4	3	1	2	4	2	8	20

ГЛАВА 3

Моделирование процессов живой и неживой природы

Путеводной нитью профильного курса информатики в 10 классе мы избрали информационное моделирование. Конечно, учащиеся уже знакомились с ним и в курсе информатики 8—9 классов, мы уделили ему значительное внимание в двух предыдущих главах. Теперь предполагается углубленное изучение информационного моделирования на примерах моделей из различных научных областей. Поэтому бóльшая часть материала данной главы относится к профильному уровню. В то же время, как уже отмечалось ранее, освоение информационного моделирования лежит в основе научного мировоззрения и обеспечивает человека научной методологией в понимании и исследовании природы и общества. Особое внимание уделяется проблеме адекватности при использовании построенных моделей для решения жизненных задач. Тем самым определенная часть материала должна быть освоена и учащимися базового уровня. Нам представляется, что для этого достаточно ограничиться материалом параграфов, отнесенных к базовому уровню.

Тема 8. Мы открываем изучение этой темы рассмотрением математических моделей, которые используются в физике. Причин для этого несколько. Во-первых, физика как научная дисциплина наиболее математизирована (мы говорим: *доктор* или *кандидат физико-математических наук*). Нередко потребности физики в адекватном математическом аппарате приводили к существенному развитию самой математики. Во-вторых, среди всех школьных предметов физика также остается основным «поставщиком» математических моделей, причем учащиеся должны не только знать эти модели, но и применять их для решения разнообразных физических задач. Учитель может предложить учащимся назвать ряд математических моделей, с которыми они уже успели познакомиться к 10 классу. Среди них наверняка прозвучит одна из наиболее изученных моделей механического движения — движение тела под действием силы тяжести. Редко находится школьник, который сомневается в справедливости этой модели. Однако простейшие расчеты (на примере капли дождя) заставляют усомниться в том, что эта модель хорошо описывает физическую реальность. Существенный фактор, не учтенный при построении исходной модели, довольно очевиден — сопротивление воздуха. Разумеется, вопрос о том, как описать действия этого фактора на движущееся тело с помощью тех или иных параметров, должны решать физики, и мы здесь пользуемся готовым результатом экспериментального исследования, который гласит, что сила сопротивления пропорциональна квадрату скорости. Здесь, между прочим, полезно упомянуть, что на самом деле такая закономерность справедлива не для всех значений скорости, а только для скоростей, близких к скорости звука (330 м/с $\pm$ 100 м/с). При меньших скоростях сила сопротивления пропорциональна первой степени скорости, а при сверхзвуковых (превышающих более чем в 2 раза) пропорциональна третьей степени скорости и даже пятой.

Само построение модели мы предлагаем провести в соответствии с планом § 19. Что касается заданий к этому параграфу, то задания 1 и 2 репродуктивны. Задание 3 ориентирует учащихся на актуализацию методов подбора параметров модели по имеющимся экспериментальным данным. Обсуждением этого задания мы напоминаем, что такие методы рассматривались в § 13 и сопровождающей его лабораторной работе.

При обсуждении задачи 4 можно напомнить учащимся, что большинство тел, движущихся с высокими скоростями в атмосфере Земли, просто сгорают. Происходит это из-за того, что работа силы сопротивления воздуха переходит в тепловую энергию. Количество этой энергии проще всего рассчитать, исходя из закона сохранения энергии. Действительно, если бы тело двигалось без сопротивления среды и имело бы скорость v_1, то обладало бы кинетической энергией, равной $\frac{mv_1^2}{2}$ (здесь m — масса тела). В среде с сопротивлением оно имеет меньшую скорость, скажем v_2. Кинетическая энергия равна $\frac{mv_2^2}{2}$. Разница энергий — это и есть потери на теплоту. Чтобы узнать изменение температуры, надо воспользоваться математической моделью процесса теплопередачи. Он описывается формулой $cm(t - t_0)$, где c — удельная теплоемкость, t_0 — начальная температура, t — температура в момент, когда скорость равна v_2. Следовательно, $t - t_0 = \frac{v_1^2 - v_2^2}{2c}$. Из этой формулы видно, что изменение температуры зависит только от удельной теплоемкости и не зависит от массы тела. Конечно, такой вывод верен, если предположить, что теплопередача в окружающее пространство отсутствует, что в реальном полете, разумеется, не так. Чтобы при выполнении лабораторной работы 11 можно было узнать повышение температуры тела при его движении в среде с сопротивлением, надо предварительно запастись значениями удельной теплоемкости для разных материалов. Нужно поручить это самим школьникам; в помощь учителю в таблице 7 мы привели значения удельной теплоемкости для некоторых веществ.

Таблица 7

Вещество	Железо	Углерод	Резина	Стекло	Бумага
Удельная теплоемкость, кДж/(кг · К)	0,452	0,502	2,09	0,67	1,51

Задание 5 подготавливает к восприятию материала § 20.

В § 20 построенная математическая модель подвергается двум преобразованиям. Во-первых, производится дискретизация модели, во-вторых, осуществляется переход от декларативной формы представления модели (в виде некоторых уравнений) к процедурной — составляются две компьютерные модели: алгоритмическая и табличная. Здесь важно еще раз обратить внимание учащихся на то, что выбор инструмента информационной технологии влияет на построение модели.

Задания 2—4 из § 20 ориентированы на подготовку к лабораторной работе 11. При этом мы рекомендуем, чтобы во время проведения компьютерного эксперимента одна половина группы, присутствующей на занятии в компьютерном классе, использовала алгоритмическую модель (программно реализованную на изучаемом ими языке программирования), а другая половина пользовалась электронной таблицей. Каждой из групп надо предложить вывести на экран траекторию движения тела, а также графики зависимости скорости, высоты и дальности полета тела от времени. В компьютерных экспериментах мы рекомендуем начальную скорость выбирать от 300 до 400 м/с, а начальную высоту от 700 до 1000 м. Эти параметры фиксируются на протяжении одного компьютерного эксперимента, а угол, начиная с 0°, варьируется с шагом 5—10°. Важно обратить внимание учащихся на две особенности, которые отчетливо наблюдаются на графиках. Первая траектория полета мало напоминает привычную параболу — в конце тело движется почти вертикально; это обстоятельство на графике дальности полета выражается тем, что график дальности приближается к некоторой горизонтальной линии. Вторая: по графику изменения скорости видно, что и она стабилизируется на некоторой величине. Иными словами, компьютерный эксперимент позволил обнаружить следующий физический эффект: при движении в среде с сопротивлением с течением времени это движение становится практически равномерным и вертикальным. Нужно предложить учащимся осмыслить полученный результат с физической точки зрения. Согласно закону Ньютона равномерное прямолинейное движение осуществляется в отсутствие движущей силы. В рассматриваемой ситуации это означает, что сила тяжести уравновешивается силой сопротивления. Иными словами, при равномерном вертикальном движении выполнено соотношение $mg = kmv^2$. Следовательно, $v = \sqrt{\frac{k}{g}}$. А теперь можно сравнить, близко ли полученное в компьютерном эксперименте значение скорости при уже почти равномерном движении к этому числу, найденному из теоретических соображений. Естественно, что эти значения у учащихся окажутся близкими (если это не так, то, следовательно, в компьютерной модели допущена ошибка).

Полученный результат достоин удивления — оказывается, конечная скорость не зависит ни от начальной скорости, ни от высоты, ни от угла, под которым начинается движение. Можно предложить учащимся убедиться в этом, проводя компьютерный эксперимент с разными значениями начальной высоты и начальной скорости. Мы специально не стали обсуждать все эти факты, поскольку хотим, чтобы у школьников появилось ощущение сделанного открытия, и надеемся, что учитель сможет с необходимым пафосом подать эту творческую ситуацию учащимся. На самом деле в современной физике компьютерный эксперимент как средство исследования и обнаружения физических эффектов находит все большее применение. Мы данной лабораторной работой смоделировали процесс такого исследования.

Первая часть лабораторной работы 11 занимает, как правило, 2 учебных часа. По ее окончании мы рекомендуем обсудить задание 5 из § 20. Компьютерный эксперимент позволил обнаружить некоторый физический эффект, однако для доказательства того, что он имеет место всегда, эксперимента недостаточно, сколько бы этих экспериментов ни проводилось. Доказательство может быть проведено только аналитическими методами, и после этого открытая закономерность обретет силу теоретического (а не эмпирического) закона. Но аналитические методы не всегда позволяют произвести расчеты для конкретных исходных данных. Поэтому компьютерные модели оказываются более востребованны, когда речь идет о практических применениях той или иной аналитической модели.

Кроме того, в конце первой части лабораторной работы 11 с учащимися следует обсудить план решения задачи 3 из § 20, в частности наметить применение метода деления пополам для нахождения оптимального угла.

Во второй части лабораторной работы 11 компьютерный эксперимент проводится с моделями, построенными при решении задач 3 и 4.

В § 21 и 22 проводится построение и компьютерное исследование моделей на примерах из экологии. Изучение материала этих параграфов преследует следующие цели:

— закрепить понимание, что модель зависит от параметров, описывающих воздействие существенных факторов на систему;

— подготовить учащихся к рассмотрению понятия адекватности модели.

В обеих моделях (модель неограниченного роста и модель ограниченного роста) рассматривается фактически один существенный фактор — влияние окружающей среды на популяцию. Однако это влияние может описываться по-разному. В учебнике приводятся два варианта описания: одним параметром — коэффициентом прироста и двумя параметрами — коэффициентом прироста и коэффициентом перенаселенности (выражающем тот факт, что в конкретных экологических условиях окружающая среда способна выдержать лишь ограниченное давление со стороны данной популяции и будет препятствовать росту ее численности).

В предшествующем абзаце мы дали как краткое обоснование основных идей, положенных в основу моделей неограниченного и ограниченного роста, так и ответы на вопросы 1—4 к § 21. Важно после рассмотрения этих двух моделей акцентировать внимание учащихся на следующем: здесь они столкнулись не только с проявлением довольно очевидного свойства, что строящаяся модель зависит от того, какими будут параметры, описывающие существенный фактор, но и с тем, что для одного и того же фактора может быть взята разная система параметров (и это, конечно, приводит к появлению разных моделей). Задание 5 направлено на подготовку к лабораторной работе 12.

Тонким дидактическим моментом в этой теме является то, что, хотя модель неограниченного роста отвергается по результатам компьютерного эксперимента, у учащихся не должно возникнуть ни разочарования (вроде как занимались бесполезной работой, когда строили эту модель), ни негативного отношения к данной модели. Для нее тоже существует область адекватности, в которой она оказывается весьма полезной. Одним из мотивов, отвергающих разочарование и негативную реакцию, может быть, к примеру, то, что для выяснения непригодности какой-либо модели она, как правило, должна быть построена. Более того, здесь полезно отметить, что компьютерный эксперимент лучше натурного уже тем, что позволяет проверить выводы и отвергнуть непригодную модель, не нанося ущерба реальному миру.

Проводить лабораторные работы мы рекомендуем в соответствии со сценарием, представленным в учебнике. На рисунке 18 приведен график роста массы растений в тундре. Аналогичные графики получаются для пустынь и степей. Для тайги график выглядит несколько иначе: в некоторые годы масса растений превышает предельно допустимую величину. Тогда в последующие годы она становится ниже этого уровня. Иными словами, масса колеблется около устойчивого уровня, который был определен как предельно допустимый.

Можно продолжить компьютерные эксперименты в двух направлениях. Во-первых, предложить изменить начальную массу, скажем, сделать ее больше, чем допустимая для данной территории. Пусть $M(0) = 16\,000$. На рисунках 19 и 20 приведены получающиеся для этой массы графики массы растений в тундре и тайге соответственно.

И в этом случае наблюдается выход на стабильный уровень, но с иным характером изменения массы. Но если взять начальную массу 18 000 т, то в таежных условиях растительность погибнет (в таблице это выражено появлением отрицательного числа в столбце «Масса»).

Во-вторых, можно предложить эксперимент с увеличением коэффициента размножения (например, у бактерий, насекомых и даже рыб он намного выше).

Для $k = 2{,}57$ получающаяся картина представлена на рисунке 21. Налицо определенная периодичность в чередовании высокого и низкого уровней численности. Не этим ли объясняется чередование годов с высоким уровнем заболевания гриппом и низким? Ведь иммунитет от гриппа сохраняется очень недолгое время.

А вот если какой-нибудь биолог в ходе многолетних наблюдений получит данные, изображенные на рисунке 22, он может и не догадаться, что это все та же модель ограниченного роста (при $k = 2{,}78$).

Тема адекватности модели, на которой сосредоточено все внимание в § 22, без сомнения, одна из важнейших тем всего курса информатики в общеобразовательной школе. Мы уделяли ей внимание на протяжении всего курса, но при этом не предлагали никаких подходов к решению проблемы адекватности средствами информационных технологий. В § 2 мы демонстрируем один из вариантов решения проблемы адекватности.

Следует признать, что § 22 (вместе с лабораторной работой 13) является, по-видимому, одним из самых трудных. Прежде всего надо прояснить для учащихся само понятие области адекватности данной модели. В самом общем виде область адекватности — это совокупность всех тех ситуаций, в которых применима данная модель. Если эта модель

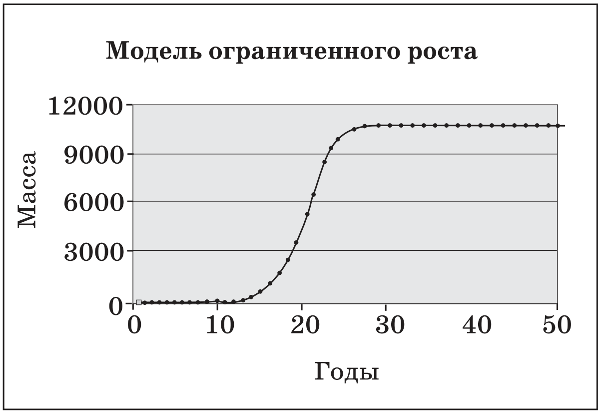

Рис. 18. График изменения массы растений в тундре

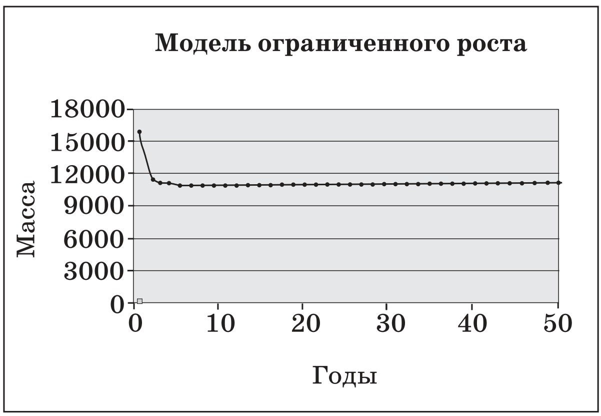

Рис. 19. График изменения массы растений при начальной массе 16 000 т в тундре

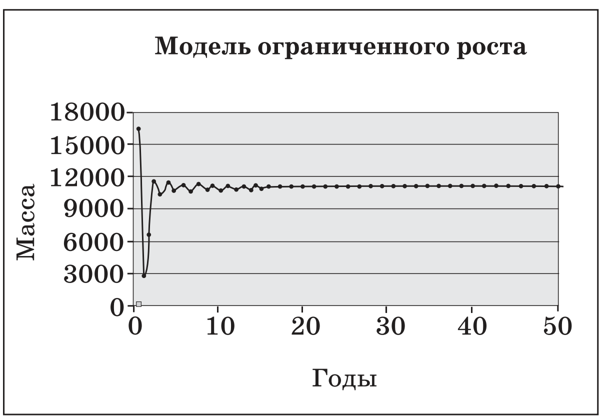

Рис. 20. График изменения массы растений при начальной массе 16 000 т в тайге

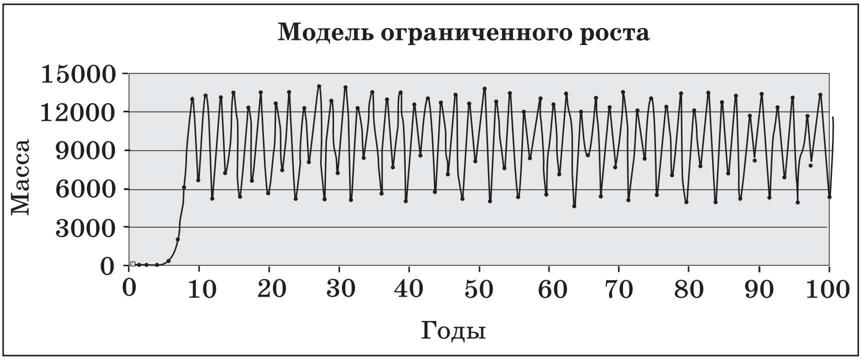

Рис. 21. График изменения массы при $k = 2,57$

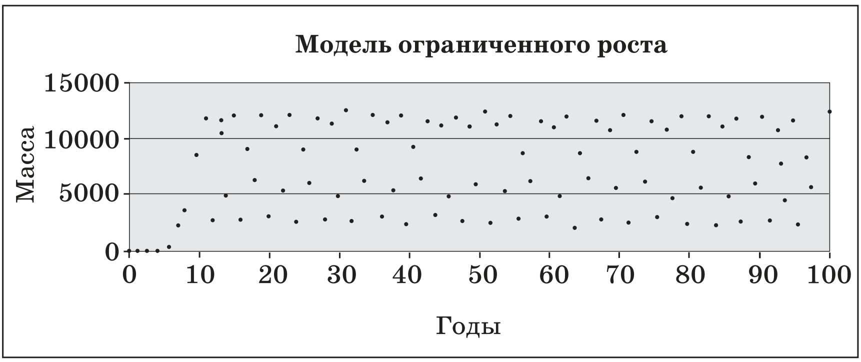

Рис. 22. График изменения массы при $k = 2,78$

информационная (и, значит, она описывается набором параметров), то область адекватности можно представлять себе как область допустимых значений этих параметров, т. е. тех значений, при которых не нарушается свойство адекватности модели. Наконец, если модель математическая, то мы можем попытаться в математических терминах (в частности, если параметры числовые, то системой неравенств) описать область адекватности модели. Поскольку модели неограниченного и ограниченного роста математические, то и границу области адекватности нужно искать в виде связи между их параметрами, выраженной в математической форме. Как было выяснено ранее, параметрами этих моделей являются начальная масса $M(0)$, коэффициент прироста k, предельная масса L, время n и масса $M(n)$ живых организмов через n лет. Нужно еще сформулировать критерий адекватности. В данном учебнике предлагается оценивать адекватность относительной величиной отклонения массы, рассчитанной по модели неограниченного роста, от массы, рассчитанной по модели ограниченного роста. Выбор численного значения критерия, вообще говоря, произвол экспериментатора. Обычно это диктуется погрешностями измерения в реальных экспериментах. Мы взяли в качестве критерия отклонение не более чем на 10%. Можно было бы выбрать другое число, это не меняет сути дела.

Следующий шаг состоит в том, что мы предполагаем наличие такой функции $f(k, L)$, что при n, меньших числа $f(k, L)$, модель неограниченного роста удовлетворяет сформулированному нами критерию адекватности. Осталось теперь подобрать эту функцию, опираясь на эксперименты с моделью. Сценарий лабораторной работы как раз и описывает постановку таких экспериментов.

Разумеется, искать сразу функцию двух переменных довольно сложно. Поэтому зафиксируем какое-нибудь значение k (в учебнике предлагается $k = 1$). Когда возникла и окрепла в ходе численных экспериментов гипотеза о виде функции при фиксированном k, переменная k «размораживается», и проводится поиск подходящей зависимости уже с переменной k. Конечно, можно считать счастливой удачей, что в данном случае вид функции $f(k, L)$ оказался довольно простым. Нередко исследователю приходится кропотливо изучать поведение функции (например, с помощью графиков) и надеяться, что в конце концов его осенит счастливая догадка. В какой мере этот поиск будет смоделирован при выполнении данной лабораторной работы, решать учителю. Мы же уверены, что, когда поиск увенчается победой, это доставит удовольствие учащимся.

Можно усложнить задачу, предложив учащимся найти ограничивающую функцию, зависящую не только от k и L, но и от величины p, устанавливающей численное значение критерия адекватности (в нашем случае $p = 10\%$).

В качестве итога изученному в § 22 полезно рассмотреть этапы решения задачи с помощью компьютера. Наглядное представление об этих этапах и их взаимодействии с уже рассмотренными базовыми понятиями информатики дает схема, изображенная на рисунке 23.

Прокомментируем еще задание 4 к § 22. В приведенной таблице довольно легко выделяется участок равномерного бега: от датчика 2 до датчика 16. После датчика 16 движение равнозамедленное, и можно предложить учащимся найти соответствующее ускорение и скорость спортсмена в тот момент, когда его бег стал замедляться. После этого уже можно лучше оценить и участок разгона. В целом это довольно сложная задача, допускающая различную глубину своей постановки.

Модель ограниченного роста — одно из величайших научных достижений. Она служит основой для построения многих других моделей, связанных с динамикой популяции. В § 23 показано, как эта модель применяется для построения модели эпидемии гриппа. Позже, в § 44, на ее основе будет построена модель управления возобновимыми ресурсами, а в § 47 она будет расширена до модели «Хищник—жертва».

После освоения довольно трудного материала § 22 материал § 23 учащимися должен восприниматься как достаточно несложный. Единственный наш методический комментарий состоит в том, что уточнение модели должно производиться после того, как выполнена первая часть лабораторной работы 14 (пункты 1 и 2). Вторая часть этой лабораторной работы проводится по уточненной модели эпидемии гриппа, а также по модели, составленной при решении задачи 4 из § 23.

Тема 9. Хотя в обыденной жизни любой человек сталкивается со случайными событиями практически постоянно, научное понятие «случайность» в российской общеобразовательной школе появилось совсем недавно. Произошло это благодаря модернизации школьного математического образования, в концепцию которого начиная с 2004 г. входит

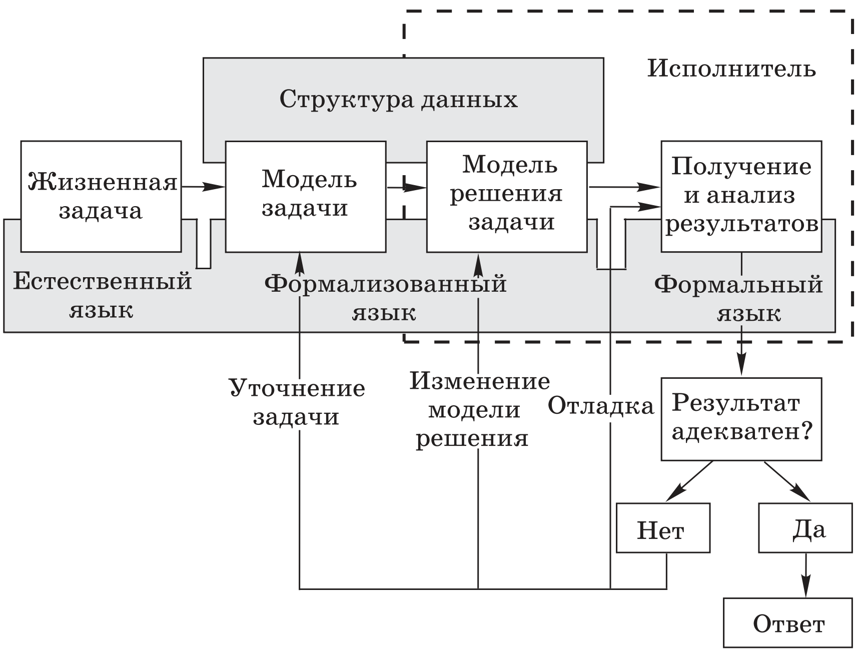

Рис. 23. Этапы решения жизненной задачи

знакомство учащихся с вероятностно-статистической линией. Это означает, что использование нашего учебника может происходить в весьма широком диапазоне условий — возможно, что учащиеся неплохо знакомы с основными понятиями теории вероятностей и элементами математической статистики, а возможно, обо всем этом слышат впервые. Поэтому первая цель § 24 — ввести необходимые понятия и терминологию. Вторая цель — показать, каковы подходы к моделированию случайных процессов. Наконец, третья цель — убедить учащихся в том, что случайность подчиняется некоторым закономерностям, которые можно изучать и моделировать с помощью компьютера. В зависимости от подготовленности класса учителю предстоит решить, как построить план занятия и сколько времени уделить достижению указанных целей.

Идея случайности первоначально отражала научную идею невозможности учесть все факторы, действующие на окружающую действительность. Чтобы точно предсказать, куда попадет летящая стрела, надо точно знать и уметь учитывать вес, форму стрелы, силу натяжения тетивы, силу и направление ветра (и даже изменение этих параметров во время полета), влажность воздуха и многое, многое другое. Поэтому можно понимать полет стрелы как некий случайный процесс, который характеризуется вероятностными параметрами. К аналогичным постановкам приводит рассмотрение социальных явлений, где невозможно учесть личные и общественные предпочтения каждого индивида, его психологические и физические особенности, поэтому социология оперирует вероятностными категориями при описании социальных явлений. Однако исследование физиками явлений микромира привело к выводу, что случайность — это неотъемлемая черта природы. Принцип неопределенности Гейзенберга показывает, что невозможно одновременно точно указать положение и импульс элементарной частицы. Корпускулярно-волновая теория, подтвержденная экспериментами по дифракции элементарных частиц, показывает, что правомерно говорить лишь о вероятности нахождения частицы в той или иной точке пространства. Детерминистская картина мира, которую школьное образование вкладывало в головы выпускников как единственно верную, оказалась не только односторонней, но и прямо противоречащей природе вещей.

Но здесь опять-таки важно избежать прямолинейного восприятия — случайность правит миром, значит, детерминизм надо отправить на свалку истории. Учитель в очередной раз должен подчеркнуть, что детерминизм и стохастика — это всего лишь два разных подхода к созданию моделей окружающего мира. Признание или непризнание случайности существенным фактором при моделировании определяет то, какая модель получится — стохастическая (мы назвали такую модель вероятностной, чтобы не вводить лишнюю терминологию) или детерминированная. А правомерность выбранного подхода определяется адекватностью построенной модели. Конечно, для теннисного мячика тоже можно записать уравнение Шредингера и найти соответствующую ему длину волны, но соотношение массы и скорости для него таково, что длина волны будет в миллионы раз превосходить длину траектории движения, так что рассматривать для него волновой эффект бессмысленно (например, объяснять промах наличием дифракции).

Второй важный момент в формировании вероятностного подхода к описанию картины мира заключается в том, что, хотя наступление случайного события — это единичный акт, говорить о вероятности события можно лишь в том случае, когда многократно (в идеале — бесконечно) повторяются одни и те же условия, в которых данное событие может произойти. Каждое такое воспроизведение условий принято в теории вероятностей называть **испытанием**. И мы в учебнике также пользуемся этим термином. Конечно, реально мы не можем произвести серию бесконечных испытаний и потому ограничиваемся достаточно большим их числом. Сколько надо провести испытаний, чтобы выявить ту или иную вероятностную характеристику случайного события? На этот вопрос как раз и отвечает математическая статистика. Мы (без всяких математических обоснований) касаемся этого трудного вопроса только один раз — в лабораторной работе 15 при проверке датчика случайных чисел.

Тем не менее проводить «вручную» необходимо длинную серию испытаний, конечно, невозможно. И в этом одна из методических проблем изучения теории вероятностей на уроках математики — все подобные эксперименты проводятся умозрительно, а значения вероятностей либо постулируются (например, объявляется, что стрелок попадает в мишень с вероятностью 0,8), либо получаются на основе косвенных соображений (например, все грани кубика равноправны, поэтому выпадение любой из них равновероятно). Компьютерный эксперимент дает возможность получить вероятностные характеристики статистическим путем, и этим информатика помогает математике в изучении сложной вероятностной темы.

Обсудим задания к § 24. Задания 1 и 2 вряд ли вызовут затруднения у учащихся; ответы к заданиям 3 и 4 содержатся в объяснительном тексте. В задании 5 только для пункта *б* можно получить значение вероятностей посредством умозаключений — все исходы равноправны, так что вероятность каждого из них равна 1/90.

В § 25 прежде всего обсуждается понятие равномерного распределения значений случайной величины. Поскольку учащимся придется иметь дело в первую очередь с дискретными случайными величинами (и, более того, с величинами, принимающими лишь конечное число значений), то мы назвали равномерно распределенной случайную величину, у которой вероятности всех значений одинаковы. Непрерывная случайная величина (т. е. такая, значения которой заполняют некоторый сплошной промежуток, быть может, всю числовую прямую) называется равномерно распределенной, если для любых двух интервалов одинаковой длины, лежащих в области значений случайной величины, вероятность попадания в каждый из них одинакова. Дело в том, что нередко используются датчики случайных чисел, которые генерируют непрерывную случайную величину, поэтому школьник должен понимать (хотя бы интуитивно), что, значит, такая величина распределена равномерно.

Далее рассматривается понятие датчика случайных чисел и приводится один из примеров — датчик фон Неймана. В задании 6а обсуждается вопрос об обязательной периодичности ДСЧ Неймана. Как показано в объяснительном тексте, для этого достаточно убедиться, что какое-то число в данной последовательности обязательно повторится. Но это действительно произойдет, поскольку различных n-значных чисел лишь конечное число. Отметим, что все используемые сегодня компьютерные ДСЧ периодичны.

В задании 6б предлагается построить непериодическую последовательность n-значных чисел. Пример может быть таким. Рассмотрим число $\sqrt{2}$, его дробную часть разобьем на блоки по n цифр и выпишем эти блоки в виде последовательности. Если какой-то блок начинается с цифры 0, заменим ее на цифру 1. Получится последовательность n-значных чисел. Предположим, что она периодична. Пусть период равен m и начинается с k-го числа. Запишем числа последовательности одно за другим без пробелов, перед первым напишем цифру 1 и отделим ее запятой. Получится бесконечная десятичная дробь. Она периодична: ее период содержит nm цифр и начинается с kn-й цифры после запятой. Значит, это число рационально. Но если оно и отличается от $\sqrt{2}$, то только единицами, периодически стоящими после kn-й цифры, и, быть может, еще несколькими единицами, стоящими до этой цифры. Можно сказать, что к $\sqrt{2}$ прибавлено указанное число из нулей и единиц. Но оно тоже рационально. Получается, что само число $\sqrt{2}$ рационально, а это, как известно, неверно. Почему же нельзя использовать построенную таким способом последовательность в качестве псевдослучайной? Да потому, что она не дает равномерного распределения[1]!

Обсудим остальные задания к § 25. Задания 1 и 2 предполагают воспроизведение определений из текста параграфа. В задании 3 учащиеся обычно затрудняются в определении источника случайных чисел. Одним из таких источников может быть измерение (например, можно измерять рост учеников данного класса). Можно подсчитывать количество дней в месяце, когда были осадки. Можно подсчитывать количество машин, проезжающих через данный перекресток в течение, скажем, пяти минут. И т. д.

В задании 4 подразумевается получение равномерно распределенной случайной величины. Второй способ не годится, поскольку вряд ли можно ожидать, что количество страниц в справочнике — это круглое число сотен. А тогда требуемые исходы не равновероятны. Третий способ также, скорее всего, не годится: у людей имеется психологическое предпочтение называть либо маленькие числа, либо, наоборот, близкие к верхней границе. Четвертый способ не годится, поскольку выигравший номер уже не участвует в дальнейших розыгрышах, так что вероятность каждого следующего номера уже не та, что у предыдущего. Первый способ можно применять при условии, что справочник совершенно новый — если им долго пользовались, то он будет открываться не на случайных, а на наиболее часто используемых страницах. Первые три способа можно предложить учащимся провести как соответствующий натурный эксперимент и проверить, является ли полученная ими последовательность чисел равномерно распределенной.

Задание 7 рекомендуется задать на дом, а его обсуждение проводить после того, как изучен материал § 26 и 27, а также проведены соответствующие им лабораторные работы.

Особое место занимает лабораторная работа 15. Она содержит довольно много теоретических сведений, а ее дидактические цели весьма разноплановы. Во-первых, учащиеся знакомятся с тем, как организовать получение случайной последовательности с помощью компьютерного датчика случайных чисел, используя изучаемый ими язык

[1] Можно предложить более простой и наглядный пример. Запишем бесконечную десятичную дробь по следующему правилу: пишем 0,1, затем пишем цифру 2, затем два раза — цифру 1, снова цифру 2, затем три раза — цифру 1, снова цифру 2, затем четыре раза — цифру 1 и т. д. Получившаяся дробь непериодична. Из нее «нарежем» последовательность n-значных чисел — здесь и нуль заменять не потребуется. Она будет также непериодична. Легко, однако, видеть, что, чем более далекие числа мы будем брать, тем больше будет подряд идущих чисел, состоящих исключительно из 1. Ни о каком равномерном распределении здесь и речи быть не может. Более того, частота появления числа из n единиц стремится к 1.

программирования. Во-вторых, углубляется понимание самого феномена случайности. Поверхностное восприятие нередко приводит к тому, что, например, жесткое чередование исходов, скажем, при бросании монеты, обеспечивающее относительную частоту каждого исхода равной в точности 0,5, отождествляется со случайным выпадением орла или решки. Совпадение относительной частоты с теоретически подсчитанной вероятностью не является гарантией случайности процесса! В случайном процессе обязательно должны наблюдаться отклонения относительной частоты от вероятности. И эта мысль доводится до учащихся в тексте лабораторной работы. Мы надеемся, что учитель сможет при необходимости эту мысль поддержать.

Например, для некоторых случайных величин, равномерно распределенных, математиками получены критерии того, как должна относительная частота отклоняться от вероятности: оказывается, в данном случае эти отклонения подчинены так называемому нормальному закону распределения. Такому же закону подчинены, например, случайные ошибки при измерении величин. Вообще нормальный закон распределения — один из наиболее часто встречающихся случаев распределения случайной величины, быть может, даже более частый, чем равномерное распределение. Объяснение этому дает так называемая центральная предельная теорема Лапласа. Мы ее, разумеется, не формулируем, но приведенная в учебнике формула — прямое следствие этой теоремы. Дальнейшая работа по проверке ДСЧ строится с использованием электронной таблицы (в частности, указанная функция НОРМСТРАСП — это сокращение слов «нормальное статистическое распределение»).

У читателя может возникнуть вопрос: почему насыщенный теоретическим материалом текст помещен в лабораторный практикум, а не в какой-либо параграф? Дело в том, что теоретическим рассмотрениям должны предшествовать компьютерные эксперименты и отделять теорию от практики в данном случае методически нецелесообразно. Впрочем, у учителя может быть иная точка зрения.

Третий целевой аспект состоит в том, чтобы продемонстрировать, что и «фирменное» программное обеспечение отнюдь не идеально. Конечно, не следует воспитывать абсолютный нигилизм в этом вопросе, однако правило «Доверяй, но проверяй» весьма актуально и в этой области человеческой деятельности.

В § 26 строится модель системы массового обслуживания. Мы выбрали магазин, хотя с таким же успехом можно было выбрать, например, поступление звонков на телефонную станцию. Учитель может поручить учащимся построить такую модель самостоятельно. В целом мы предлагаем следовать учебнику при объяснении материала и выполнении лабораторной работы 16 — здесь ситуация противоположна той, что была в предыдущей лабораторной работе: весь необходимый теоретический материал изложен в объяснительном тексте параграфа.

В заданиях к этому параграфу впервые предлагается полностью почти самостоятельно построить математическую модель и подготовить проведение компьютерного эксперимента (задание 4). Мы употребили слово «почти», поскольку существенные факторы и описывающие их параметры указаны явно: в первом случае — это только случайное изменение направления движения (при постоянной скорости), во втором — случайным изменениям подвержены и направление, и скорость частицы. Впрочем, мы не бросаем ученика на произвол судьбы — в описании лабораторной работы 17 приводится и соответствующий алгоритм, и заполнение электронной таблицы.

Метод Монте-Карло, которому посвящен § 27, является одним из мощных методов вычисления площадей и объемов (в общем случае — кратных интегралов). Основан он на интуитивно ясном замечании, что вероятность попадания равномерно распределенной случайной величины в заданный интервал пропорциональна длине этого интервала. Аналогично для равномерно распределенной пары случайных величин (а пару чисел естественно рассматривать как координаты точки на плоскости) вероятность попадания в некоторую фигуру пропорциональна площади этой фигуры. Поэтому если мы моделируем процесс бросания точки в некоторую область, то площадь этой области пропорциональна вероятности попадания в нее, а, следовательно, при большом числе бросаний пропорциональна относительной частоте попадания.

На проведение лабораторной работы 18 мы запланировали 3 часа. Первая часть этой лабораторной работы (задания 1—5) как раз посвящена нахождению площадей и объемов методом Монте-Карло. При этом задания 1 и 2 нацелены на то, чтобы преодолеть возможное предубеждение учащихся против применения вероятностных методов для нахождения точных величин. Наши эксперименты показывают, что 100 000 бросаемых точек обеспечивают вычисление в числе π четырех знаков после запятой. Если этого окажется недостаточно, продолжите компьютерный эксперимент так, как это описано в задании 2 данной лабораторной работы. Именно после работы с методом Монте-Карло учащиеся, как правило, начинают проникаться доверием к результатам, полученным с помощью вероятностных методов, — до этого им любые подобные результаты кажутся малообоснованными и недостоверными.

В задании 3 (компьютерная реализация решения задачи 1 из § 27) площади первых двух фигур школьники, скорее всего, точно найти не могут — для этого они должны уметь интегрировать степенную функцию. А именно, площадь фигуры, изображенной на рисунке 3.4, *а*, равна $\int_0^1 x^2dx = 1/3$, а на рисунке 3,4, *б* — $\int_0^1 x^3dx = 1/4$. Найти площадь треугольника, изображенного на рисунке 3.4, *в*, труда не составляет.

Второй час, отводимый на выполнение лабораторной работы 18, мы рекомендуем посвятить выполнению задания 6 (компьютерная реализация решений задач 2 и 3 из § 27).

Построение модели для задачи 2 несложно. Примем 12 часов дня за начало отсчета, т. е. точку 0. Тогда время прихода резидента задается формулой 2*ДСЧ(1) − 1. Обозначим это число t_R. Время прихода агента задается точно такой же формулой. Обозначим это число t_A. Условие, что они встретятся, можно записать так: $|t_R - t_A| \leqslant 1/4$ (ибо 15 мин — это четверть часа). Модель построена. Соответствующий алгоритм выглядит так:

```
Алгоритм Встреча_с_агентом
цел:m, n, k; вещ: t1, t2;
{  m := 0;
   Запросить n;                    (*количество испытаний*)
   Делать от k := 1 до n
   {  t1 := 2*ДСЧ(1) − 1;
      t2 := 2*ДСЧ(1) − 1;
      Если ABS(t1 − t2) ⩽ 0,25 то  {  m := m + 1;  }
   }
   Сообщить m/n;
}
```

С учащимися можно обсудить геометрическую модель той же задачи. Пара чисел (t_R, t_A) задает в координатной плоскости точку. В силу принятых нами соглашений $-1 \leqslant t_R \leqslant 1$ и $-1 \leqslant t_A \leqslant 1$. Это значит, что такие точки заполняют квадрат со стороной 2 (рис. 24). Условие встречи, как мы видели, записывается неравенством $|t_R - t_A| \leqslant 1/4$. Множество соответствующих этому неравенству точек изображено закрашенным шестиугольником.

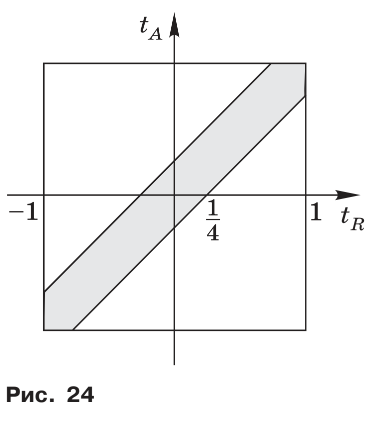

Рис. 24

Тогда вероятность встречи — это отношение площади закрашенного шестиугольника к площади квадрата. Площадь шестиугольника легко вычислить, если заметить, что она равна площади всего квадрата минус площадь двух незакрашенных треугольников. Катеты этих треугольников равны 7/4. Значит, площадь шестиугольника равна $4 - 49/16 = 15/16$. Отношение этого числа к площади квадрата равно 15/64, или 0,234375. Это и есть искомая вероятность.

Аналогично разбирается задача 3 из § 27. Условием того, что уравнение имеет действительные корни, является неотрицательность дискриминанта. Так что алгоритм выглядит весьма похожим образом.

Алгоритм Вероятность_корня
цел:m, n, k; **вещ**: p, q;
{ $m := 0$;
  Запросить n; (*количество испытаний*)
  Делать **от** $k := 1$ **до** n
  { $p := 2$*ДСЧ(1) − 1;
    $q := 2$*ДСЧ(1) − 1;
    Если $p^2 - 4*q \geqslant 0$ **то** { $m := m + 1$; }
  }
  Сообщить m/n;
}

Геометрическая модель тоже строится аналогично задаче 2. Точки с координатами (p, q) при указанных в задаче ограничениях заполняют такой же квадрат, как и в предыдущей задаче. Условие $p^2 - 4*q \geqslant 0$ задает множество точек под параболой $q = \frac{p^2}{4}$. Площадь под этой параболой приходится считать с помощью интеграла, как это было сделано в задаче 1а. Точный ответ приведен в учебнике.

Задания 5 и 6 из § 27 представляют собой небольшие проекты. Их можно выполнять небольшими группами и затем представлять на обсуждение классу. В частности, полезно уделить внимание дизайну каждого из проектов.

Задание 7 подготавливает учащихся к восприятию материала § 28. В этом параграфе углубляются знания учащихся о количестве информации как меры снятой неопределенности и обосновывается формула Шеннона. Для этого учащимся необходимо освоить вероятностный подход к измерению неопределенности. То, что неопределенность (как фактор) описывается с помощью вероятности (которая тем самым выступает параметром описания данного фактора), школьникам интуитивно ясно. Аргументация требуется для того, чтобы объяснить, почему степень неопределенности связана с вероятностью через логарифм. А для этого приходится знакомить их с «правилом произведения»: вероятность произведения независимых событий равна произведению вероятностей. Мы не формулируем это правило, а только иллюстрируем его примерами подбрасывания двух кубиков и двух монет. Если учащиеся знают указанное правило из курса математики, то можно (и полезно) прямо на него сослаться. Если же они с этим правилом незнакомы, можно предложить им рассмотреть опыт с одновременным подбрасыванием кубика и монеты. Список равновозможных исходов опыта выглядит тогда так: (1, «орел») (2, «орел») (3, «орел») (4, «орел») (5, «орел») (6, «орел») (1, «решка») (2, «решка») (3, «решка») (4, «решка») (5, «решка») (6, «решка»).

Общее число исходов равно произведению числа исходов в опыте подбрасывания кубика и числа исходов в опыте подбрасывания монеты. Неопределенность же исхода в данном опыте с одновременным подбрасыванием равна сумме неопределенностей опытов с подбрасыванием каждого из предметов отдельно, так что снова справедлива формула $H(km) = H(k) + H(m)$, где $k = 6$, а $m = 2$.

Обоснование формулы Шеннона и ее использование существенно опирается на свойства логарифмической функции. Поэтому данный материал может рассматриваться только в тех классах (преимущественно математических), где логарифмическая функция уже изучена.

Перейдем к обсуждению заданий данного параграфа. Ответ на первый вопрос таков: формула Шеннона соответствует пониманию информации как проявления разнообразия в окружающем мире. Эта мера не зависит от свойств приемника информации (в частности, от его осведомленности).

В задании 2 ясно, что вероятность выпадения осадков равна сумме вероятности дождя и вероятности выпадения снега, т. е. 0,8. Тогда соответствующая неопределенность равна $-0{,}8 \log_2 0{,}8 - 0{,}15 \log_2 0{,}15 \approx 0{,}6681$. Тем самым, неопределенность оказалась меньше, чем 15 июня.

При обсуждении этой задачи учащиеся могут задать вопрос: не могло ли случиться так, что осадки были в виде дождя со снегом (а если такой вопрос не возник у учащихся, его вполне может задать сам учитель)? Ответ на него таков: если бы имел место случай одновременного выпадения двух видов осадков, то это должно было учитываться при подсчете вероятности как для дождя, так и для снега. Тогда сумма всех вероятностей была бы больше 1. Однако это не так, следовательно, случаев одновременного выпадения дождя и снега не наблюдалось[1].

[1] На языке алгебры событий наступление хотя бы одного из двух событий A или B описывается как сумма этих событий: $A + B$; одновременное наступление двух событий называется произведением этих событий и записывается AB. В теории вероятностей доказывается формула $P(A + B) = P(A) + P(B) - P(AB)$, где через P обозначена вероятность соответствующего события. Если события одновременно не происходят, то $P(AB) = 0$ и, следовательно, $P(A + B) = P(A) + P(B)$. В противном случае $P(A + B) < P(A) + P(B)$.

В задании 3а применение формулы Шеннона дает значение $-20/30\ \log_2 2/30 - 10/30\ \log_2\ 1/30 \approx 4{,}24$. Полученное число и есть среднее количество вопросов, за которое можно узнать ответ, если применять оптимальную стратегию. В пункте *б* того же задания надо попытаться подобрать стратегию, которая давала бы близкую вероятность гарантированного угадывания ответа. Никакого общего рецепта для подбора оптимальной стратегии не существует, надо просто пробовать различные стратегии и для каждой из них оценивать среднестатистическое значение числа вопросов. Приведем пример. Пусть первый вопрос таков: верно ли, что задуманное число больше 10? С вероятностью 1/3 мы получаем ответ «да», с вероятностью 2/3 получаем ответ «нет». Далее все числа равноправны (в каком бы из оставшихся диапазонов они ни находились), поэтому целесообразно применить метод половинного деления. Пусть ответ был «нет». Вопросом: «Верно ли, что задуманное число больше 5?» — множество чисел от 1 до 10 разбивается на два пятиэлементных множества. Пусть ответ будет «нет». Следующий вопрос: «Верно ли, что задуманное число больше 3?» — разбивает на два подмножества, однако ответы не равновероятны: ответ «да» имеет вероятность 2/5, ответ «нет» имеет вероятность 3/5. В первом случае нам далее потребуется один вопрос, во втором случае с вероятностью 1/3 потребуется один вопрос и с вероятностью 2/3 — два вопроса. На рисунке 25 представлено дерево вариантов (правая ветвь в точности совпадает с левой, поэтому мы не стали ее рисовать).

Разглядывая левую веточку этого графа (т. е. для чисел $\leqslant 10$), видим, что 5 вопросов потребуется в 4 случаях, а для 6 случаев потребуется задать только 4 вопроса. Тем самым, исходная вероятность 2/3 попадания на левую ветвь делится в пропорции 4 : 6. Значит, общая вероятность получить 5 вопросов равна $0{,}4 \cdot 2/3 = 8/30$, а для 4 вопросов она равна $0{,}6 \cdot 2/3 = 12/30$. Аналогично для правой ветки имеем вероятность 5 вопросов $0{,}4 \cdot 1/3 = 4/30$, а вероятность 4 вопросов — $0{,}6 \cdot 1/3 = 6/30$. Тогда среднее число вопросов равно $5 \cdot 8/30 + 4 \cdot 12/30 + 5 \cdot 4/30 + 4 \cdot 6/30 = 4{,}4$. Как мы видим, она несколько больше, чем та, которая получается по формуле Шеннона. Это легко интуитивно понять — ведь первое деление пополам не учитывало вероятности загадывания чисел. Более разумная стратегия — первым вопросом разбить множество чисел на множество чисел, меньших 10, и множество чисел, больших или равных 10. Подсчитать среднее количество задаваемых вопросов в этом случае мы оставляем читателю.

Такой подсчет, однако, — вещь непростая, для учащихся наверняка неосуществимая. Поэтому для проверки оптимальности стратегии можно предложить им провести компьютерный эксперимент. Для этого последовательно случайным образом генерируются натуральные числа от 1 до 20, причем так, чтобы вероятность числа от 1 до 10 была в 2 раза выше, чем для чисел от 11 до 20. Каждое такое число объявляется задуманным, и к нему учащиеся применяют свою стратегию определения этого числа, подсчитывая количество потребовавшихся вопросов. После нескольких таких испытаний (не меньше 40) подсчитывается среднее значение числа вопросов и сравнивается с шенноновским. Ниже приведен алгоритм, позволяющий генерировать нужную псевдослучайную последовательность чисел.

Алгоритм Генерация_2 : 1
цел: m, k;
{ $m := 1 + \mathrm{INT}(30{*}\text{ДСЧ}(1))$;
Если $m > 10$ **то** { $m := m - 10$; }
Сообщить m;
}

Мы рекомендуем также предложить учащимся провести эксперименты с другим распределением вероятностей загадываемых чисел.

В задаче 4 нетрудно написать алгоритм определения фальшивой монеты. Первым действием мы на каждую чашу весов кладем по 333 монеты. Если весы оказались в равновесии, то фальшивая монета среди оставшихся 334 монет. Если же какая-то чаша перевесила, то фальшивая монета

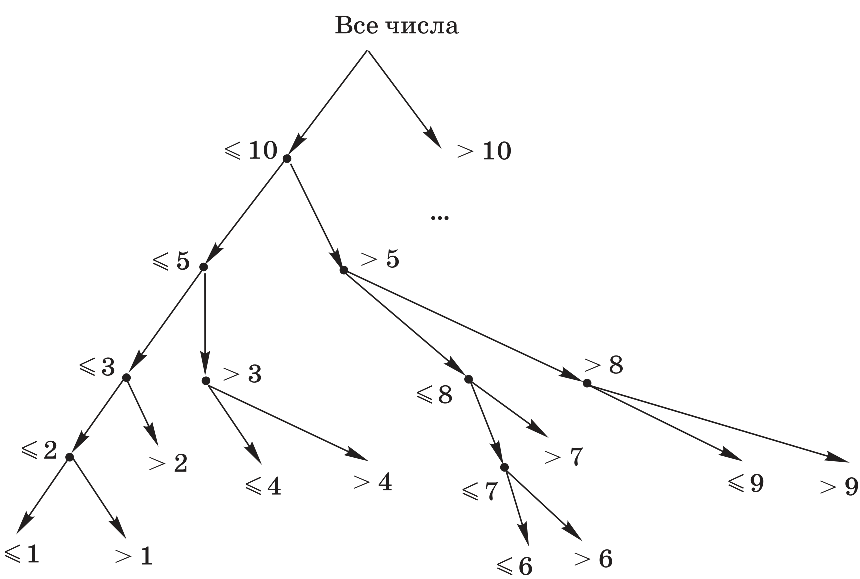

Рис. 25

лежит на другой чаше. Вторым действием на каждую чашу весов кладем по 111 монет из той группы монет, где оказалась фальшивая. Снова возможны два случая: весы оказались в равновесии, тогда монета среди оставшихся то ли 111, то ли 112 монет, либо равновесия нет, и тогда монета лежит на той чаше весов, которая легче. Третьим действием кладем на чаши весов по 37 монет, при этом рядом с весами осталось то ли 37, то ли 38 монет. В зависимости от результатов взвешивания определяем, в какой группе монет оказалась фальшивая. Если она оказалась в группе из 37 монет, то кладем на чаши весов по 12 монет (четвертое взвешивание), а еще 13 остается рядом. Если же она оказалась в группе из 38 монет, то для четвертого взвешивания кладем на чаши весов по 13 монет, а 12 монет остается рядом. Аналогично после пятого взвешивания фальшивая монета будет локализована не более чем в 6 монетах. После шестого взвешивания фальшивая монета будет либо выделена, либо окажется одной из 2 монет. Во втором случае седьмым взвешиванием мы выделим фальшивую монету.

Почему нельзя выделить фальшивую монету за меньшее число взвешиваний? С позиций измерения количества информации ответ можно обосновать так. Мы знаем, что для выделения одного элемента из множества однотипных элементов требуется не менее $\log_2 1000$ бит информации. А за каждое взвешивание мы получаем один из трех возможных исходов: весы находятся в равновесии (фальшивая монета среди оставшихся), левая чаша легче правой (фальшивая монета на левой чаше), левая чаша тяжелее правой (фальшивая монета на правой чаше). Значит, за одно взвешивание мы получаем $\log_2 3$ бит информации. За x взвешиваний можно получить не более чем $x\log_2 3$ бит информации. Если за x взвешиваний удалось выделить монету, то должно выполняться неравенство $x\log_2 3 \geqslant \log_2 1000$. Поэтому $x \geqslant \frac{\log_2 1000}{\log_2 3} = \log_3 1000 \approx 6{,}2877$. Но число взвешиваний должно быть целым, поэтому $x \geqslant 7$.

Решение задачи 5а основано на следующем соображении: двузначных чисел 90. Для выделения одного из них требуется $\log_2 90$ бит информации. После сообщения требуется выделить уже одно число из 36 чисел, т. е. не хватает $\log_2 36$ бит информации. Значит, получено $\log_2 90 - \log_2 36 = \log_2 2{,}5$ бит информации. Тот же результат можно получить, используя понятие неопределенности. Требовалось выделить одно число из 90, затем стало нужно выделить одно число из 36. Неопределенность уменьшилась в $90/36 = 2{,}5$ раза; следовательно, получено $\log_2 2{,}5$ бит информации.

Аналогичное рассуждение для задания 5б показывает, что данное сообщение несет $\log_2(900/225) = 2$ бита информации (учащиеся нередко ошибаются, подсчитывая количество трехзначных чисел, меньших 325, — это числа от 100 до 324 включительно).

В задании 6 ответ зависит от того, разрешается ли ставить 0 на первое место в номере машины. По «официальной версии» (т. е. по рассказу К. Чапека) разрешается. Тогда у нас имеется 1000 возможных номеров машин, а из трех указанных цифр различными перестановками можно получить всего лишь 6 вариантов. Значит, количество информации, полученное полицией, равно $\log_2(1000/6) \approx 7{,}38$ битов. Если же учащиеся примут договоренность, что 0 на первом месте стоять не может (и не следует устраивать по этому вопросу долгую дискуссию), то общее число трехзначных чисел 900, а допустимых перестановок указанных трех цифр всего лишь 4. Тем самым в этом случае количество информации, полученное полицией, равно $\log_2(900/4) \approx 6{,}9657$ битов.

Задание 7 аналогично задаче о неопределенности погоды, разобранной в объяснительном тексте параграфа. Мы принимаем, что студенты оценивают свои знания вполне объективно, поэтому для Иванова вероятности оценок таковы: отлично — 0,25; хорошо — 0,4; удовлетворительно — 0,3 и неудовлетворительно — 0,05. Для Петрова соответствующие вероятности таковы: 0,15; 0,5; 0,25; 0,1. Подсчитываем величину энтропии оценок для каждого студента:

$H_{\text{Иванов}} = -\,0{,}25 \cdot \log_2 0{,}25 - 0{,}4 \cdot \log_2 0{,}4 - 0{,}3 \cdot \log_2 0{,}3 - 0{,}05 \times$
$\times \log_2 0{,}05 \approx 1{,}766;$

$H_{\text{Петров}} = -\,0{,}15 \cdot \log_2 0{,}15 - 0{,}5 \cdot \log_2 0{,}5 - 0{,}25 \cdot \log_2 0{,}25 - 0{,}1 \times$
$\times \log_2 0{,}1 \approx 1{,}743.$

Видно, что неопределенность с оценкой у них почти одинакова, но все же у Иванова меньше.

В задании 7б получаем у Иванова вероятность сдать экзамен 0,95, а у Петрова — 0,9. Соответствующие энтропии таковы:

$H_{\text{Иванов}} = -\,0{,}95 \cdot \log_2 0{,}95 - 0{,}05 \cdot \log_2 0{,}05 \approx 0{,}2864;$

$H_{\text{Петров}} = -\,0{,}9 \cdot \log_2 0{,}9 - 0{,}1 \cdot \log_2 0{,}1 \approx 0{,}4690.$

По вопросу сдачи экзамена, как видно из полученных чисел, неопределенность различается весьма существенно.

Поскольку материал главы 3 предназначен в основном для углубленного изучения, тестовые задания, предложенные в учебнике учащимся для самоконтроля, не имеют аналогов ни в ЕГЭ, ни в Централизованном тестировании. Поэтому и мы не предлагаем дублирующий тест для проведения контрольного мероприятия в классе.

ГЛАВА 4

Логико-математические модели

Логика со времен Аристотеля была призвана моделировать человеческое мышление. Его загадки и ученых, и людей, далеких от науки, волнуют, быть может, даже больше, чем тайны неживой природы. Информация и ее преобразование человеческим умом — это ключевые элементы мышления человека. Однако большинство школьных учебников информатики не идут дальше общих деклараций на эту тему и простейших сведений об алгебре высказываний. И это вполне соответствует базовому уровню изучения информатики в общеобразовательной школе. Что касается профильного уровня, то нам представляется важным познакомить учащихся с современными методами и информационными средствами в этой области, тем более что системы искусственного интеллекта, реализующие эти методы, все в большей мере используются при построении самых разнообразных информационных продуктов.

Еще в первой главе отмечалось, что основным направлением современной информатики является автоматизация информационных процессов на основе применения компьютерной техники. Возможность такой автоматизации обеспечивается построением формальных моделей. Это означает, что и в вопросах перекладывания на плечи компьютера каких-то функций человека, связанных с его умственной деятельностью, основное внимание также сосредоточено на рассмотрении формальных моделей человеческого мышления. Но в этой сфере тот сугубо «численно-ориентированный» математический аппарат, который школьниками изучается и воспринимается как собственно математика, оказывается уже недостаточным[1]. До определенной степени это уже было продемонстрировано в главе 1, где были введены понятия формального языка, автомата, машины Тьюринга, графа, т. е. целого ряда абстрактных математических объектов, играющих принципиальную роль в теоретическом обосновании многих положений информатики. Без преуве-

[1] Речь идет о том, что для учащихся общеобразовательной школы математика — это дисциплина, рассматривающая исключительно числовые или конкретные геометрические объекты и операции над ними (включая функциональные соотношения). На самом деле современная математика имеет совершенно иной облик: числа и геометрические фигуры в большинстве случаев выступают в ней всего лишь как модели тех абстракций, которыми она оперирует.

личения можно сказать, что математика в своем развитии во многом обязана возникающим задачам, имеющим, по сути, информатическую природу. В этой главе мы знакомим школьника еще с двумя важнейшими математическими структурами — предикатами и бинарными отношениями. Их изучение началось в середине XX в. Тем не менее эти молодые теории лежат в основе большинства современных компьютерных средств. В частности, язык предикатов лежит в основе так называемого логического программирования, а теория бинарных отношений составляет основу всех реляционных баз данных (в том числе СУБД Access). Фактически речь идет о самых современных и в то же время фундаментальных аспектах математики, имеющих решающее прикладное значение в информатике.

Глава 4 открывается параграфом, носящим постановочный и мотивирующий характер. В нем обсуждаются затронутые выше вопросы моделирования мыслительной деятельности человека, в частности проблема использования компьютерной техники для принятия решения. Эта проблема многогранна — ей посвящены труды ученых, и в то же время она являет собой благодатное поле деятельности для писателей-фантастов. Беседа и возможные обзоры на эту тему, самостоятельно выполненные учащимися, — вот, на наш взгляд, наиболее адекватная форма введения учащихся в эту проблематику. Отметим, что поднимаемые в этом параграфе вопросы найдут свое продолжение в главе 5, где будут рассматриваться кибернетические аспекты информатики.

Тема 10. Эта тема, возможно, повторяет то, что изучалось учащимися в курсе информатики 8—9 классов. По крайней мере они, конечно, уже знакомы с основными логическими операциями и конструировали с их помощью сложные высказывания для применения таковых в качестве условий в конструкциях ветвления и цикла. Разумеется, на это знание нужно опереться, в частности оно способно сыграть мотивационную роль. Фактически § 30—33 имеют своей целью систематизировать знания учащихся в разделе «Математическая логика». Первое, на что следует направить усилия учителю, — это добиться четкого понимания учащимися, что такое высказывание. Учащиеся обычно легко отличают высказывания от вопросительных или побудительных предложений, но часто путают их с предикатами, т. е. высказывательными формами, содержащими свободные переменные. Поэтому мы начинаем с примеров таких высказывательных форм, где переменная присутствует явно (например: «Число x рационально»), и лишь затем переходим к высказывательным формам, где переменная обозначена каким-либо именем (например: «Четырехугольник является параллелограммом»). Надо обратить внимание учащихся (здесь или позже) при изучении предикатов, что в русском языке квантор всеобщности нередко подразумевают, но не употребляют явно. Типична, к примеру, такая формулировка: «В прямоугольнике диагонали равны». Строго говоря, надо было бы формулировать так: «В любом прямоугольнике диагонали равны», чтобы не было свободной переменной, обозначенной словом «прямоугольник». Тем не менее небрежное отношение к связыванию переменных кванторами — источник многих ошибок у школьников.

Второй аспект, возникающий при переходе от весьма неформализованного русского языка к языку математической логики, состоит в особенностях употребления местоимений. Нередко местоимение играет роль свободной переменной. Скажем, во фразе «Кто-то что-то с кем-то делал» три свободные переменные, обозначенные местоимениями «кто-то», «что-то» и «кем-то». Их можно заменить объектами: «Коля с Васей пошли в кино». Теперь мы имеем высказывание, которое будет истинным или ложным в зависимости от того, пошел ли Коля с Васей в кино. Однако во фразе «Вот кто-то с горочки спустился» (см. задание 1ж из § 30) местоимение «кто-то» выступает в качестве связанной переменной — точный смысл этой фразы таков: «Существует x, который спустился с горочки». Тем самым это повествовательное предложение является высказыванием — либо был кто-то, спустившийся с горочки, и тогда утверждение истинно, либо никого не было, и тогда утверждение ложно. В целом мы снова видим здесь проявление уже обсуждавшегося ранее эффекта формализации, но теперь применительно к той ситуации, что язык математической логики выступает формализацией естественного языка. А любая формализация как один из этапов моделирования неоднозначна и зависит от разных факторов, в данном случае от контекста.

В связи с выше сказанным мы хотим предостеречь учителя от чрезмерного увлечения обсуждением высказываний, относящихся к различным бытовым ситуациям. Их формализация далеко не однозначна и может порождать бесполезные дискуссии, отвлекающие от освоения учебного материала (и потому вредные).

Новыми для учащихся могут оказаться операции разделительной дизъюнкции (исключающего ***или***), импликации и равносильности. Особое внимание следует уделить импликации. Работа в алгоритмах с условными выражениями в форме **Если**... **то**... и прочтение импликации как такого же словесного оборота нередко порождают у учащихся неверное представление, что в импликации посылка обязана быть истинной. Иными словами, учащиеся должны отчетливо понимать, что высказывание, являющееся импликацией, и конструкция ветвления в алгоритме — это совершенно разные сущности, совпадающие только по форме словесного оформления. Чтобы закрепить требуемое различение, можно предложить учащимся следующее задание:

Задание 8. Дан алгоритм:
Алгоритм Простая_логика
цел: x, y; **лог**: A, $\overline{B}$;
{ **Запросить** x;
Запросить y;
$A := (x > y)$;
$B := (x \text{ mod } y = 0)$;
Если $(A \to B)$ **то** { $x := x + y$; }
иначе { $y := x*y$; }
Сообщить x, y;
}

Что будет сообщено после выполнения этого алгоритма, если по командам **Запросить** вводятся следующие значения:
а) $x = 5$; $y = 2$; б) $x = 2$; $y = 5$; в) $x = 10$; $y = 2$; г) $x = 0$; $y = 7$.

Ответы в этом задании таковы: а) 5 и 10; б) 7 и 5; в) 12 и 2; г) 7 и 7.

Конечно, в программах на реальном языке программирования импликация в условии встретиться не может, так что предложенный нами алгоритм носит чисто учебный характер.

Обратим внимание читателя еще на одно обстоятельство. В конце объяснительного текста § 30 вводится понятие простого высказывания. Это сугубо рабочее понятие и ему не следует придавать сколько-нибудь академический характер. В зависимости от контекста одно и то же высказывание может играть роль простого, а может выступать как составное. Например, высказывание «Родился ребенок» можно рассматривать как простое, можно считать его составным, поскольку оно эквивалентно дизъюнкции высказываний «Родился мальчик» и «Родилась девочка»[1].

Приведем ответы к заданиям § 30.

В задании 1 высказывания приведены в пунктах *а, б, ж, з, и*. В пунктах *г* и *е* приведены высказывательные формы. В остальных пунктах предложения не являются повествовательными.

---

[1] Конечно, начитанные дети могут оспорить равенство «Родился ребенок» = «Родился мальчик» или «Родилась девочка», ссылаясь на пушкинскую «Сказку о царе Салтане». Ведь там говорится: «Родила царевна в ночь не то сына, не то дочь, не мышонка, не лягушку, а неведому зверюшку». И царь этому сообщению вроде бы поверил, даже гонца хотел повесить. Но жизнь, конечно, расставила все по своим местам. Однако это шуточное обсуждение в очередной раз показывает, что разделение высказывания на более простые требует осмотрительности.

В задании 2 все высказывания истинны. Конечно, истинность высказывания в пункте *г* лежит на совести историков, но мы в объяснительном тексте специально подчеркивали, что логика не несет ответственность за истинность высказываний, касающихся конкретных фактов. Устанавливать их истинность — забота тех научных дисциплин, к ведению которых эти факты относятся. Истинность высказывания в пункте *в* является следствием определения необъятности: необъятное — это то, что никому нельзя объять. Установить истинность высказывания в пункте *д* сложнее — для этого можно изучить семь календарей невисокосных лет, в которых 1 января последовательно приходится на понедельник, вторник, среду, четверг, пятницу, субботу и воскресенье, а также семь аналогичных календарей для високосных лет. Разумеется, этот пункт лучше задать домой.

Задание 4 является математизированным вариантом задания 2. Истинность высказываний в пунктах *а* и *в* подтверждается соответствующими математическими теоремами; в пунктах *д* и *ж* устанавливается несложными рассуждениями. Высказывания в пунктах *б, е, з* и *и* ложны. Это устанавливается приведением соответствующих примеров (в пункте *е* таким примером служит число 1, в пункте *и* — число 1001). В пункте *г* приведено высказывание, которое на сегодняшний день не доказано, но и не опровергнуто. Тем не менее это, конечно, высказывание, поскольку ясно, что сформулированное утверждение или истинно, или ложно.

Задания 5—7 обычно не вызывают затруднений у школьников. В задании 8 отрабатывается понимание истинности импликации. Каждый раз, как только посылка импликации ложна, сама импликация истинна. Поэтому истинными являются высказывания в пунктах *б, в, г, д*.

Для задания 9 (а—в) в таблице 8 приведем искомые таблицы истинности.

Таблица 8

X	Y	$\overline{Y} \to \overline{X}$	$X \to (Y \to X)$	$\overline{X \vee Y} \to (X \,\&\, Y)$
И	И	И	И	И
И	Л	Л	И	И
Л	И	И	И	И
Л	Л	И	И	Л

Высказывание в пункте 9 *г* является тождественно истинным.

В § 31 основой изучения нового материала являются понятия равносильности логических выражений и их тождественных преобразований. Первое, важно акцентировать внимание учащихся, что при замене в логической формуле какой-либо ее части, которая сама является логической формулой, на ей равносильную, получается формула, равносильная исходной. Значительную помощь в усвоении этого оказывает аналогия с тождественными преобразованиями алгебраических выражений. Важно также внушить учащимся мысль, что, имея набор базовых тождеств, можно производить тождественные преобразования, какими бы странными эти тождества ни казались. Современная алгебра (как раздел *современной* математики) как раз изучает операции на множествах, удовлетворяющих тем или иным, иногда довольно экзотическим тождествам или похожим на них соотношениям.

Второе, на чем нужно акцентировать внимание учащихся, — это особенность математической логики, состоящая в том, что для доказательства равенства двух логических формул не обязательно производить тождественные преобразования; достаточно сравнить их таблицы истинности. Составление таблиц истинности и их сверка представляют собой совершенно формальную работу, ее можно поручить компьютеру. Первая часть лабораторной работы 19 как раз посвящена реализации этого подхода.

Чтобы избежать изобилия скобок при записи логических выражений, мы ввели приоритеты операций. Они совпадают с общепринятыми, но надо иметь в виду (и предупредить об этом школьников), что существуют языки программирования и поисковые системы, которые это соглашение не поддерживают. Решение о введении приоритетов в выполнении действий воспринимается учащимися как вполне естественное — достаточно апеллировать к аналогии с числовыми выражениями.

В целом объяснительный текст данного параграфа нам представляется несложным, перейдем поэтому к обсуждению заданий.

В задании 2а таблица истинности для первого высказывания приведена в объяснительном тексте § 30, таблицу истинности для второго высказывания учащиеся должны были составить, выполнив задание 9а, к тому же параграфу. Сравнивая две эти таблицы, учащиеся приходят к выводу о равносильности данных высказываний. Обычно к этому времени учащиеся профильных классов с углубленным изучением математики уже знакомы с понятиями прямого, обратного и противоположного утверждений[1]. Равносильность двух данных утверждений означает, что исходная

[1] Напомним, что если теорема записана в форме импликации $A \to B$, то обратной теоремой называется утверждение $B \to A$, а противоположной теоремой — утверждение $\overline{A} \to \overline{B}$. Соответственно теорема $\overline{B} \to \overline{A}$ называется обратной противоположной.

теорема имеет место тогда и только тогда, когда выполняется обратная противоположная теорема. Обычно этот факт доказывается с использованием метода «от противного». Здесь же доказательство абсолютно формально. Обсуждение этого феномена обычно производит сильное впечатление на учащихся и демонстрирует им, что благодаря математической логике истинность некоторых утверждений может быть получена сугубо формальными и даже синтаксическими средствами. Последнее утверждение (о синтаксическом характере вывода утверждений) надо было бы подкрепить доказательством равносильности двух данных высказываний с помощью тождественных преобразований. Это полезно предложить учащимся проделать самостоятельно. А затем отметить, что тождественные преобразования (неважно, алгебраических или логических выражений) производятся без обращения к смыслу (семантике) преобразуемых выражений, т. е. являются чисто синтаксической процедурой.

Равносильность высказываний в пунктах *б* и *в* также легко проверяется. Полезно обратить внимание учащихся, что равносильность высказываний в пункте *в* выражает распределительный закон конъюнкции относительно исключающего ***или***. В обоих примерах можно предложить учащимся ответить на вопрос: останутся ли равносильными высказывания в этих парах, если конъюнкцию заменить дизъюнкцией? Ответ положительный.

В пункте *г* в каждой строке, где $X = \text{Л}$ и $Z = \text{Л}$, значения приведенных высказываний различны. Тем самым они не равносильны.

Задания 3 и 4 выполняются аналогично.

В задании 5, как и во всех заданиях с формулировкой «Упростить выражение», возникает вопрос: что считать упрощенным видом данной формулы? В алгебре логики можно дать такой ответ: полученное выражение должно содержать минимально возможное число переменных, соединенных знаками конъюнкции и дизъюнкции, а отрицание должно относиться только к переменным. Это обосновано двумя соображениями: во-первых, как уже говорилось, в процедурных языках программирования не применяются логические операции, отличные от конъюнкции, дизъюнкции и отрицания, во-вторых, такое преобразование подготавливает школьников к восприятию дизъюнктивной нормальной формы, с которой они встретятся при изучении следующего параграфа. Отметим, что ответ в таких заданиях неоднозначен, так что утверждение о неправильности ответа, полученного кем-либо из школьников, требует проверки, что он не равносилен правильному ответу. Разумеется, школьники не должны доказывать, что получившаяся у них формула действительно содержит минимально возможное число переменных — на сегодняшний день не существует эффективного метода, позволяющего решать подобную задачу за приемлемое время.

При выполнении задания 6а могут быть предъявлены разные доказательства. Приведем одно из наиболее коротких:

$$(X \vee Y) \,\&\, Y = (X \,\&\, Y) \vee (Y \,\&\, Y) = (X \,\&\, Y) \vee Y = Y.$$

Первое равенство получается применением распределительного закона (под номером 6), второе равенство — применением закона идемпотентности (под номером 8), третье равенство — применением первого из законов поглощения (под номером 17).

В задании 6б из тождества $\overline{X \vee Y} = \overline{X} \& \overline{Y}$ (второй закон де Моргана) подстановкой вместо X и Y их отрицаний получаем тождество $\overline{\overline{X} \vee \overline{Y}} = \overline{\overline{X}} \,\&\, \overline{\overline{Y}}$. Учитывая закон двойного отрицания (формула 19), правую часть преобразуем к виду $X \,\&\, Y$. Применив к обеим частям отрицание и воспользовавшись еще раз законом двойного отрицания, получаем первый закон де Моргана. Аналогично второй закон де Моргана выводится из первого.

Тавтологичность формул в задании 7 может проверяться составлением для них таблиц истинности или с помощью тождественных преобразований. Например,

$$(X \to X) = \overline{X} \vee X = \text{И}.$$

Первое равенство получается применением формулы 20, а второе — формулы 14.

Первая формула в задании 7, несмотря на ее простоту, заслуживает обсуждения. К сожалению, не так уж редко мы сталкиваемся с рассуждениями типа: «Нечто имеет место, потому что оно имеет место». Классическим примером является исторически известное утверждение «На солнце не может быть пятен, потому что их там быть не может». Истинность данной импликации никак не может повлиять на истинность самого утверждения X. Эта простая мысль должна быть осознана всеми учащимися. Мы надеемся, что после этого учителям математики реже придется слышать утверждения типа: «Этот треугольник прямоугольный, потому что он прямоугольный». Или рассуждения, напоминающие рассуждения «от противного», но таковыми не являющиеся: «Предположим, что утверждение верно, докажем, что оно верно». Во всех случаях проявляет себя все та же тавтология $X \to X$.

Задание 8 в своей идейной основе не отличается от задания 7. Приведем ответы к этому заданию: в пунктах *а* и *б* приведены тавтологии, формула в пункте *в* тавтологией не является.

В задании 9 преобразуем левую часть формулы:

$(\overline{X \vee A}) \vee (\overline{X \vee \bar{A}}) = (\bar{X} \,\&\, \bar{A}) \vee (\bar{X} \,\&\, \bar{A}) = \bar{X} \,\&\, (\bar{A} \vee A) = \bar{X} \,\&\, \text{И} = \bar{X}.$

Поэтому $\bar{X} = B$, откуда $X = \bar{B}$.

В задании 10а ответ отрицателен: если положить $X = \text{И}$ и $Y = \text{И}$, а $Z = \text{Л}$, то какую бы формулу для F мы ни взяли, высказывание, записанное в этом пункте, будет ложным. В задании 10б ответ также отрицателен: если положить $X = \text{Л}$ и $Y = \text{И}$, то для любой формулы F высказывание, записанное в этом пункте, будет ложным. Дополнительно можно предложить вопрос: существует ли такая формула F, при подстановке которой в формулу $F \,\&\, X \vee \bar{F} \,\&\, (\bar{X} \vee \bar{Y})$ эта формула становится тавтологией? Здесь ответ положителен: в качестве F можно взять X.

В § 32 приводится алгоритм построения логической формулы по заданной таблице истинности. Поскольку любая логическая функция от любого числа переменных может быть задана своей таблицей истинности, фактически в этом параграфе показано, что **для любой логической функции существует формула, задающая эту функцию**. В алгебре ничего подобного нет — алгебраическими выражениями (т. е. выражениями, построенными с помощью переменных и алгебраических операций) вовсе не исчерпываются не только все числовые функции вообще, но даже все непрерывные функции.

Между прочим, благодаря тому что каждая логическая функция полностью определяется своей таблицей истинности, нетрудно подсчитать, сколько всего существует различных логических функций от n переменных. Их столько же, сколько можно составить различных таблиц. А таблиц существует $2^{(2^n)}$ — каждая такая таблица содержит 2^n строк наборов значений аргументов, и в каждой строке может реализовываться любой из двух вариантов значения функции: Истина или Ложь.

Формула, которая получается из таблицы истинности применением описанного в § 32 алгоритма, имеет довольно специальный вид, который, в свою очередь, имеет специальное название: **совершенная дизъюнктивная нормальная форма**, сокращенно СДНФ. Она показывает, что на самом деле для записи любой логической функции достаточно только трех операций: дизъюнкции, конъюнкции и отрицания. Этим в частности объясняется то обстоятельство, что именно эти операции используются в языках программирования для записи условий в алгоритмических конструкциях[1]. Конечно, получающаяся формула в СДНФ весьма неэкономна с точки зрения количества используемых в ней переменных и опе-

[1] В 11 классе мы покажем, что любую логическую функцию можно записать, используя одну подходящую логическую операцию.

раций. Как мы уже отмечали, проблема построения экономных формул с заданной таблицей истинности пока не решена, хотя некоторые оптимизирующие алгоритмы существуют.

Эти два феномена — возможность задать любую логическую функцию с помощью таблицы истинности и наличие алгоритма построения логической формулы по таблице истинности — составляют основу для решения задач 5 и 6 из § 32. Впрочем, учащиеся довольно часто догадываются, какая формула годится, и без обращения к таблице истинности. Например, в задании 6а для функции F можно использовать формулу $(A \& B) \vee (B \& C) \vee (A \& C)$. Отметим, что эта формула представляет собой дизъюнктивную нормальную форму (ДНФ), но не является СДНФ, где в каждом конъюнкте присутствуют все переменные. В частности при построении F с помощью таблицы истинности у учащихся получится иная формула (но, конечно, равносильная данной).

Если есть время, то можно познакомить учащихся с **совершенной конъюнктивной нормальной формой** (СКНФ). Она получается, если в таблице истинности выбрать строки, которые соответствуют ложному значению функции. Затем для каждой такой строки пишется дизъюнкция тех переменных, которые в этой строке имеют значение Ложь, и отрицание тех переменных, которые в этой строке имеют значение Истина. Конъюнкция всех получившихся дизъюнктных выражений и есть требуемая конъюнктивная нормальная форма. Ясно, что для функции, у которой ложных значений меньше чем истинных, СКНФ получается более экономной, чем СДНФ.

Изложение в § 33 методов решения логических задач с помощью формул математической логики скорее дань традиции, чем реальная необходимость в изучении этих методов. Обычно ответ в подобных задачах может быть получен без применения формализма математической логики. Школьники это видят, и у них создается негативное впечатление: применяется нечто сложное и малопонятное к вещам, которые почти очевидны. Мы постарались подобрать задачи (как в объяснительном тексте, так и в заданиях к параграфу), для которых этот эффект проявляется, как нам кажется, в минимальной степени. Сюда же примыкает вторая часть лабораторной работы 19. Привлечение компьютера (т. е. формального исполнителя) к решению таких задач демонстрирует школьникам, что решение подобных задач не представляет большого труда. Тем самым акцент переносится с собственно решения задачи на формализацию процесса решения, а это есть главная дидактическая цель данного параграфа.

Выполнение заданий к этому параграфу осуществляется по образцам, предъявленным в объяснительном тексте. Вот как, например, можно выполнить задание 1.

Обозначим через X высказывание «Джон не встречал Смита», через Y высказывание «Смит был убийцей», через Z высказывание «Джон лжет» и через U высказывание «Убийство произошло после полуночи». Тогда следующие функции имеют значение Истина:

$X \to Y \vee Z;$
$\overline{Y} \to X \,\&\, U;$
$U \to Y \vee Z;$
$\overline{U}.$

Запишем конъюнкцию всех истинных высказываний:

$$\text{И} = (X \to Y \vee Z) \,\&\, (\overline{Y} \to X \,\&\, U) \,\&\, (U \to Y \vee Z) \,\&\, \overline{U} =$$
$$= (\overline{X} \vee Y \vee Z) \,\&\, (Y \vee (X \vee U)) \,\&\, (\overline{U} \vee Y \vee Z) \,\&\, \overline{U} =$$
$$= (\overline{X} \vee Y \vee Z) \,\&\, (Y \vee (X \,\&\, U)) \,\&\, \overline{U} =$$
$$= (\overline{X} \vee Y \vee Z) \,\&\, ((Y \,\&\, \overline{U}) \vee (X \,\&\, U \,\&\, \overline{U})) = (\overline{X} \vee Y \vee Z) \,\&\, (Y \,\&\, \overline{U}) =$$
$$= (X \vee Y \vee Z) \,\&\, Y \,\&\, \overline{U} = Y \,\&\, \overline{U} = Y \,\&\, \text{И} = Y.$$

Следовательно, Y истинно.

Аналогично выполняются и два других задания к этому параграфу.

Тема 11. В этой теме учащиеся знакомятся с математическим аппаратом, который служит теоретической основой всех реляционных баз данных. Два первых параграфа этой темы (§ 34 и 35) посвящены введению понятия отношения, способам записи отношений и выделению одного из важнейших классов отношений — функциональных отношений. В одном из вариантов реформы школьного образования предлагалось в курсе математики знакомить школьников с теорией отношений и в частности вводить понятие функции, но, к сожалению, этот проект не был принят, так что эти общематематические понятия приходится осваивать в курсе информатики.

В § 34, кроме введения понятия отношения, демонстрируются две возможные формы записи отношения: указанием свойства пар (троек, четверок и т. п.) элементов, находящихся в данном отношении, и в виде таблицы, в которой элементы, связанные данным отношением, явно выписаны. Учащиеся должны уметь переходить от задания отношения через свойство к записи этого отношения таблицей. На отработку этого умения направлено задание 5 из § 34.

Ответ в задании 3 из § 34, разумеется, положительный. Более того, он важен для § 35, где как раз и рассматриваются функциональные отношения.

Задание 4 призвано подкрепить учащимся декларацию того факта, что отношения встречаются практически всюду. Скажем, в биологии отношение определяется тем, кто для кого служит пищей, в химии — взаимодействуют ли два данных вещества.

Задание 6 из § 34 подготавливает к выполнению лабораторной работы 20. Учащимся надо посоветовать, информацию о каких классах лучше взять. Крайне желательно, чтобы в таблице фигурировали учителя, работающие одновременно в нескольких классах, одинаковые предметы, которые ведут разные учителя, одинаковые по названию предметы в разных параллелях (например, литература в 9 классе и литература в 11 классе). Это задание должно быть задано домой с последующим обсуждением в классе и, возможно, с объединением результатов отдельных учеников в одну большую таблицу.

Понятие функциональной зависимости, о котором фактически идет речь в § 35, относится к числу фундаментальнейших понятий не только в математике, но и вообще в науке. К сожалению, в школьном курсе математики это понятие связывается почти исключительно с числовыми функциями. Даже естественные геометрические примеры почти никогда не рассматриваются с этой точки зрения. Вот только наиболее часто употребляемые функциональные соотношения в геометрии:

1. Каждому многоугольнику сопоставляется его периметр.
2. Каждому многоугольнику сопоставляется его площадь.
3. Каждому треугольнику сопоставляется вписанная в него окружность.
4. Каждому многоугольнику сопоставляется наименьший радиус окружности, которая его целиком содержит.
5. Каждому отрезку сопоставляется его середина.
6. Каждому треугольнику сопоставляется точка пересечения его высот.
7. Каждому многоугольнику сопоставляется сумма его внутренних углов.

Этот список легко продолжит каждый ученик.

А вот функциональные отношения в физике:

1. Каждому веществу сопоставляется температура его плавления.
2. Каждому веществу сопоставляется его плотность.
3. Каждому проводнику сопоставляется величина его сопротивления.
4. Каждому веществу сопоставляется его удельная теплоемкость.

И т. д. Мы предлагаем учителю уделить внимание примерам функциональных отношений, которые являются не числовыми, а хотя бы «геометрическими», «физическими» или «химическими».

Роль понятия «функция» для теории реляционных баз данных состоит в аккуратном и ясном введении понятия «ключ».

Понятие ключа играет немаловажную роль в теории и практике баз данных. Поскольку в лабораторном практикуме мы ориентируемся на СУБД Access, посмотрим, что говорится о ключе, если выбрать пункт меню *Помощь*. Там термин «ключ» употреблен с двумя разными прилагательными: «первичный ключ» и «внешний ключ». Первичный ключ определен как «одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ используют для связывания таблицы с внешними ключами в других таблицах». В свою очередь, внешний ключ — «это одно или несколько полей (столбцов), содержащих ссылку на поле или поля первичного ключа другой таблицы. Внешний ключ определяет способ объединения таблиц». Ясно, что, пока речь идет об одной таблице, внешний ключ нам не интересен (хотя и с ним не все гладко).

Рассмотрим, к примеру, таблицу Браки с атрибутами Муж и Жена. Будем предполагать, что мы живем в высоконравственной христианской стране, где у каждого мужа может быть только одна жена, а у каждой жены — только один муж. Тогда, согласно приведенному определению, у нашей таблицы два ключа. Каждый из ее атрибутов — ключ. Но у таблицы Access может быть только один ключ!

Следовательно, приведенное определение не вполне корректно. Кроме того, оно не содержит даже намека на одно немаловажное обстоятельство. Дело в том, что ключ в Access назначается пользователем. Ну как же можно не сказать ему об этом?!

Чтобы прояснить ситуацию, удобно ввести понятие **возможного ключа**.

Набор атрибутов данного отношения (т. е. таблицы) назовем возможным ключом, если это отношение функционально и данный набор атрибутов может служить ее аргументом.

Итак, пусть данное нам отношение функционально, и мы определили для него возможные ключи. Теперь нужно сказать, что из всех возможных ключей пользователь может выбрать только один и назначить его ключом для данной таблицы (в терминологии Access — первичным ключом). В Access это делается с помощью кнопки или команды, называемых одинаково: *Ключевое поле* (меню *Правка* или контекстного меню в режиме *Конструктор таблицы*). На бланке *Конструктор таблицы* ключ помечается специальным символом.

Мы предлагаем определить **ключ** так: **ключ — это один из возможных ключей, назначаемый пользователем** (в Access — с помощью команды или кнопки *Ключевое поле*).

Это определение конструктивно. Оно, во-первых, показывает, что ключ назначается пользователем (и даже содержит указание на способ назначения), а во-вторых, что ключом может быть назначен не любой атрибут (или набор атрибутов), а лишь являющийся возможным ключом.

Что произойдет, если пользователь назначил некий атрибут ключом? Во-первых, Access проверит, может ли предложенный пользователем атрибут быть ключом. Если нет, то на экране появится соответствующая диагностика. Во-вторых, Access будет следить, чтобы соответствующее отношение и впредь оставалось функцией. Система не допустит изменений таблицы, нарушающих это свойство.

Отметим, кстати, что у пользователя есть и другой способ превращения отношения в функцию. Это так называемый уникальный индекс. Но ради экономии места мы здесь его касаться не будем.

Мы совсем ничего не сказали, зачем нужен ключ. Позже мы остановимся на этом подробно, и выяснится, что, назначив атрибут ключом или уникальным индексом, пользователь получает возможность создавать определенного типа связи; кроме того, это позволяет ускорить поиск информации по данному атрибуту.

Мы надеемся, что сказанного достаточно, чтобы проникнуться тем, насколько важно учащимся понимать роль функциональности отношения для последующего разумного (а не кнопконажимательского) отношения к Access.

Обсудим задания к § 35. В задании 2 отношение функционально, но, как ни покажется странным, аргументом является ребенок: именно для него отец определен однозначно, а не наоборот.

В задании 3 нетрудно догадаться, что в одиночку ни один из атрибутов не может претендовать на роль аргумента в функциональном отношении, а значит, и быть ключом. Однако совокупность двух атрибутов, например Название и Автор, уже можно взять в качестве ключа.

В задании 4 ни один из атрибутов сам по себе ключевым не является, однако любая пара уже может служить ключом. Это означает, что данное отношение функционально.

В задании 5а еще одним ключевым набором является пара атрибутов — количество электронов на слоях M и N. Задание 5б мы рекомендуем рассматривать только в классах физико-химического профиля.

Особая роль отводится § 36. В нем вводится несколько важных понятий: предиката, свободной и связанной переменной, квантора существования и квантора всеобщности[1].

[1] Со связанной переменной в другом ее обличии учащиеся встречаются и на уроках математики. Например, в выражении $a = \underset{i=1}{\overset{n}{\varnothing}} x_i^2$ переменные a и n свободны, об их значении имеет смысл спрашивать. А переменная i присутствует в связанном виде, она только указывает диапазон изменения; точно так же, как и переменная, связанная квантором всеобщности или существования, указывает, что здесь требуется рассмотреть все элементы области определения предиката.

Любое содержательное утверждение — математическая теорема, физический закон или закон социальной жизни — представляет собой высказывание, полученное из некоторого предиката связыванием всех его переменных. В этом тезисе два важных аспекта. Первый — учащиеся должны понимать, что в любом высказывании *все* переменные должны быть связаны. К сожалению, учащиеся нередко допускают утверждения, содержащие свободные переменные. Иногда, конечно, связывающий квантор можно определить по контексту[1]. Второй — мы уже отмечали, что в традиции коммуникативного языка отсутствие квантора надо воспринимать как квантор всеобщности. Именно эти языковые вольности нередко служат причиной логических ошибок.

Второй аспект более тонкий и сложный: при связывании переменных кванторами важен порядок применения кванторов существования и всеобщности. Значительное число учащихся убеждены, что порядок кванторов роли не играет. Важно разубедить их в этом. Учащиеся должны понимать, что выражение «существует x такой, что для любого y...» означает, что элемент x обслуживает все возможные значения y сразу. А выражение «для любого y существует x...» означает, что нужный элемент x выбирается в зависимости от того, какой элемент y рассматривается. Разумеется, это различие должно не только обсуждаться на абстрактном уровне, но и демонстрироваться разнообразными примерами (ряд таких примеров содержат задания 5 и 6).

Важным обсуждаемым в этом параграфе аспектом является правило построения отрицания для высказываний с кванторами. Здесь два дидактических момента. Во-первых, правильное преобразование кванторных приставок при построении отрицания — замена квантора всеобщности квантором существования и, наоборот, при обязательном сохранении порядка следования переменных. Во-вторых, правильное построение отрицания бескванторной части. В заданиях к § 36 предлагаются упражнения на построение

---

[1] Вот два примера. В высказывании «Если диагонали параллелограмма равны, то этот параллелограмм — прямоугольник» подразумевается квантор всеобщности: для *любого* параллелограмма. В высказывании «Если две стороны и угол между ними одного треугольника равны двум сторонам и углу между ними другого треугольника, то треугольники равны» подразумевается квантор существования: *существуют* две стороны одного треугольника и угол между ними и *существуют* две стороны другого треугольника. А вот пример посложнее: «Если сумма противоположных углов четырехугольника равна 180°, то в этот четырехугольник можно вписать окружность». Какой квантор стоит здесь: «если *существует* пара противоположных углов с суммой 180°, то...» или «если для *любой* пары противоположных углов сумма равна 180°, то...»? Такой вопрос нередко ставит школьника в тупик. После размышления многие учащиеся с удивлением обнаруживают, что здесь достаточно квантора существования.

отрицаний. Входящие в них высказывания оформлены как предложения русского языка. Учитель может, если считает необходимым, предложить учащимся построить отрицание высказывания, записанного логической формулой. Пример такого упражнения содержит тестовое задание А4.

Полезно продемонстрировать учащимся, что в случае конечной области определения логической функции квантор всеобщности можно заменить конъюнкцией соответствующих предикатов, где вместо связанной переменной подставлены поочередно все элементы области определения. Квантор существования в этом случае можно заменить аналогичной дизъюнкцией. Допуская вольность речи, можно сказать, что квантор всеобщности — это «бесконечная конъюнкция» по всем элементам области определения, а квантор существования — это аналогичная «бесконечная дизъюнкция». Мы обсуждаем это в § 37, но учитель может сделать это и раньше.

Поскольку любая логическая функция априори может принять только два значения Истина и Ложь, естественно интересоваться множеством тех значений переменных, при которых предикат принимает значение Истина (по определению предиката, что при всех оставшихся значениях аргумента он принимает значение Ложь). Нахождение таких множеств (т. е. решений уравнения $P(x_1, x_2, \ldots, x_n) = \text{И}$) облегчает учащимся понимание предиката как функции. В то же время отбор таких решений под названием «фильтрация» применяется в любой базе данных. В школьной математике такие задачи хорошо известны как задачи с параметром в весьма разнообразных и изощренных формулировках.

Идейная и техническая насыщенность § 36 привела к тому, что и комплект заданий в этом параграфе получился довольно обширным.

Задание 2а продолжает пример, приведенный в объяснительном тексте. В задании 2б для первой пары значений y и z предикат истинен, ибо для x нужное значение существует; для второй пары значений он ложен; для третьей снова истинен: в качестве x надо взять пустое слово. На самом деле это задание — первый и совсем маленький шаг в понимании школьниками, что логическая формула — это всего лишь синтаксическая конструкция, а ее семантика, т. е. содержательное утверждение, которое она выражает, зависит от интерпретации входящих в нее символов. Если переменные интерпретируются как числа, а знак «+» как сложение, знак «=» как равенство, формула наполняется одним смыслом, и можно говорить, что она истинна или ложна при тех или иных наборах значений переменных. Если переменные интерпретируются как слова, а знак «+» как конкатенация слов, знак «=» как равенство, то формула также становится осмысленной, и снова можно говорить о ее истинности или ложности на заданном наборе значений переменных. Можно предложить учащимся еще каким-либо образом проинтерпретировать переменные и знак операции (что касается знака «=», то его в логике договариваются всегда интерпретировать одинаково — как совпадение значений). Например, можно считать, что переменные обозначают некоторые множества, а знак «+» означает операцию объединения множеств. Получим еще одну интерпретацию той же самой формулы.

Оказывается, что существуют формулы, которые принимают значение Истина при любой интерпретации входящих в нее переменных и знаков операций. Как и в случае с высказываниями, такие формулы называются тавтологиями. Сами тавтологии большого интереса не представляют. Но если к тавтологиям добавить некоторое множество формул, называемых аксиомами, и указать правила вывода (другими словами, правила тождественных преобразований), то множество формул, которые можно получить с помощью этих правил, образует так называемую аксиоматическую теорию. Важно, что если имеется какая-то интерпретация аксиом, в которой все аксиомы истинны, то и все формулы теории оказываются истинными в этой интерпретации. С точки зрения логики именно эта теория и есть доказательное знание рассматриваемой области. Если мы берем аксиомы геометрии, мы получаем геометрию, если аксиомы натуральных чисел, получаем теорию чисел и т. д. Конечно, информатика не место для подробных рассмотрений этих вопросов, но общий взгляд на них нам представляется полезным.

Задание 3 несложно. В пунктах *а* и *б* выражения имеют по две свободные переменные и одну связанную. В пункте *в* приведено высказывание, оно не содержит свободных переменных.

Ответы в задании 4 таковы: а) $x = 1$; б) $x = 2$. В пункте *б* учащиеся иногда ошибочно включают в ответ число 1. В этом случае им надо напомнить, что простым называется натуральное число, имеющее ровно два делителя. Число 1 не является ни простым, ни составным.

О заданиях 5 и 6 было сказано выше. При их выполнении важно акцентировать внимание учащихся на содержательное различие утверждений, в которых используется разный порядок применения к переменным кванторов всеобщности и существования (при одинаковом порядке переменных, связанных этими кванторами). В задании 7 еще раз обращается внимание на различность утверждений, получаемых перестановкой связанных переменных. Это различие подчеркивается выводом, что из истинности предиката

$\exists y \forall x(P(x,y))$ следует истинность предиката $\forall x \exists y(P(x, y))$. На самом деле формула

$$\exists x \forall y(P(x, y)) \to \forall x \exists y(P(x, y))$$

тождественно истинна (т. е. является тавтологией). Но формально-логическое доказательство этого факта опирается на утверждения логики предикатов, неизвестные школьникам. На содержательном уровне обсуждение тождественной истинности этой импликации (в том духе, как это сделано выше) весьма целесообразно.

Задания 8 и 9, с одной стороны, аналогичны заданиям алгебры высказываний, показывающим, как из простых высказываний с помощью логических операций конструируются сложные высказывания. С другой стороны, они подготавливают учащихся к восприятию языка логического программирования, где существуют так называемые встроенные предикаты, и надо уметь выражать через них предикаты, приводящие к решению задачи. У школьников при решении этих задач иногда возникает искушение дать рекурсивное определение требуемого предиката. В этом случае надо объяснить, что на данном этапе этого не допускается, пусть подождут до изучения логических языков программирования. Отметим, что ответы в этих заданиях не определены однозначно, могут существовать разные способы выражения одного предиката через другие предикаты.

Приведем возможные ответы к этим заданиям.

Задания 8. а) $\overline{N(x)}$; б) $N(x) \,\&\, D(y, x)$; в) $D(x, y) \,\&\, D(y, x)$. Здесь используется понятное, но редко применяемое свойство натуральных чисел: если числа делятся друг на друга, то они равны. Поэтому данный пункт задания выполняется учащимися дольше, чем два предыдущих.

г) $\forall y(D(y, x))$. Трудность этого задания в том, что учащиеся должны догадаться до введения связанной переменной. У некоторых учеников возникает желание использовать предикат $x \leqslant y$, что недопустимо, ибо в условии строго оговорено, какими предикатами можно пользоваться.

д) $D(x, z) \,\&\, D(y, z) \,\&\, \forall t((D(x, t) \,\&\, D(y, t)) \to D(z, t))$. Здесь использовано свойство наибольшего общего делителя, состоящее в том, что он делится на любой другой общий делитель.

е) $\forall z(D(x, z) \,\&\, D(y, z) \to \forall s(D(s, z))$. В этом задании учащиеся нередко забывают квантор $\forall z$. В результате получается предикат с тремя свободными переменными x, y и z. Конечно, истинным он будет только при $z = 1$, но это не означает, что он описывает указанное в пункте *е* свойство, где фигурируют только две свободные переменные.

ж) $(\forall y(D(x, y) \to D(y, x) \vee \forall z(D(z, y)))) \,\&\, (\exists z(\overline{D(z, x)}))$. Предикат $\exists z(\overline{D(z,\ x)})$ выражает тот факт, что простое число не равно 1. О нем учащиеся нередко забывают.

з) Предикат получается комбинацией предикатов из пунктов *ж* и *а*:

$$(\forall y(D(x,\ y) \to D(y,\ x) \vee \forall z(D(z,\ y)))) \,\&\, (\exists z(\overline{D(z,\ x)}))\&\overline{N(x)}.$$

и) Этот предикат является очевидной комбинацией предикатов из пунктов *ж* и *е*. Важно, чтобы учащиеся видели, что в этом пункте приведено высказывание, и потому полученная ими формула не должна содержать свободных переменных. Вот как может выглядеть ответ:

$$\forall x((\forall t(D(x,\ t) \to (D(t,\ x) \vee \forall z(D(z,\ t)))) \,\&\, (\exists z(\overline{D(z,\ x)}))) \to$$
$$\to \forall y(\forall z(D(x,\ z) \,\&\, D(y,\ z) \to \forall sD(s,\ z))) \vee D(y,\ x))).$$

Важно проследить, чтобы область действия квантора $\forall x$ распространялась на все выражение; нередко учащиеся допускают ошибку в расстановке скобок, из-за которой действие этого квантора ограничено лишь той частью, которая записана до импликации, стоящей перед $\forall y$. В таком случае получается предикат со свободной переменной x.

к) Здесь тоже записано высказывание. Вот возможный вариант формулы:

$$\forall y \exists x((\forall t(D(x,t) \to (D(t,x) \vee \forall z(D(z,\ t)))) \,\&\, (\exists z(\overline{D(z,x)})) \,\&\, D(y,x)).$$

Задание 9 аналогично заданию 8. В пункте *а* предикат может выглядеть так: $\exists z(U(x,\ z,\ y))$. Имея этот предикат, нужные предикаты в пунктах *б–г* можно «списывать» с соответствующих пунктов задания 8. Однако в ряде случаев удобно пользоваться прямо предикатом U. Например, в пункте *в* можно записать $U(x,\ x,\ x)$. Чтобы построить предикат, описанный в пункте *д*, полезно переформулировать условие так: любые два простых делителя числа x равны, а само x не равно 1.

Задания 10 и 11 затруднений обычно не вызывают.

Задание 12 — типичное задание на уравнение с параметром. Оно перебрасывает межпредметный мостик от информатики к школьной алгебре. Предлагать его целесообразно только учащимся с хорошей математической подготовкой.

Тема 12. В § 37—39, относящихся к данной теме, учащимся демонстрируется, что математический аппарат, который осваивался ими в предыдущих параграфах этой главы, служит основой функционирования СУБД Access. Надо прямо сказать, что для большинства школьников (даже обучающихся в классах с углубленным изучением математики) сама математика воспринимается как наука, носящая прежде всего функционально-числовой характер, а в информатике ее роль в основном ограничена построением и обоснованием тех или иных вычислительных алгоритмов. На самом деле математика XX в. изучает абстрактные структуры, иногда весьма причудливые, для которых числовая оболочка всего лишь средство кодирования информации. Математическая логика, с которой познакомились в этой главе учащиеся (конечно, весьма поверхностно), — один из таких разделов современной математики, она является фундаментом для разработки любой реляционной базы данных. Корректность работы БД (а это означает, что при работе различных пользователей с базой данных в ней не возникает противоречий) обеспечивается математическими теоремами, выполнение условий которых, как правило, закладывается в конструкцию СУБД с помощью так называемых ограничений целостности данных и правил каскадного обновления данных. Мы не рассматриваем данные вопросы подробно, но при выполнении лабораторной работы 20 учащиеся в экспериментальной работе обнаруживают необходимость следовать определенным правилам построения связей между данными, выражаемыми на языке умножения отношений. Непосредственное отношение к этой лабораторной работе имеет задание 3 из § 37, но важную роль играет также задание 4, которое мы рекомендуем обсудить в классе.

В целом мы советуем при изучении этой темы следовать объяснительному тексту параграфов и сценариям лабораторных работ 20 и 21.

Тема 13. Изучение логических языков программирования относится к дополнительному материалу. Изложение материала носит ознакомительный характер, компьютерный практикум проводится в том случае, если в компьютерном классе установлена какая-либо версия языка Пролог, например Турбопролог или Пролог-Д.

ГЛАВА 5

Информационные модели в задачах управления

Тема 14. В этой теме реализуется требование Обязательного минимума содержания образования по информатике познакомить учащихся с информационными процессами в управлении. Центральным понятием здесь является понятие обратной связи. Именно ему и его применению в решении задач управления уделяется главное внимание в данной главе.

Что касается модельных задач, на которых раскрываются понятия управления и обратной связи, то здесь мы руководствовались соображениями их общезначимости, так что эти задачи — задача о потреблении возобновимых ресурсов и экономическая задача ценообразования — не нуждаются в создании дополнительной мотивации к их изучению.

В § 43 вводится понятие управления. Но управление здесь осуществляется по разомкнутой схеме, т. е. без организации обратной связи. Роль компьютерного эксперимента видна в этом случае особенно выпукло: необходимо выбрать такое управляющее воздействие, которое бы обеспечило достижение цели управления и в то же время не привело к разрушению системы. Это означает, что нужно решить соответствующую задачу прогноза; в нашем случае это осуществляется построением подходящей компьютерной модели, позволяющей отслеживать функционирование системы при разных управляющих воздействиях. Такое построение модели естественно разбить на два этапа: сначала смоделировать функционирование самой системы, а затем «встроить» в эту базовую модель управляющее воздействие. В § 44 такой базовой моделью является модель ограниченного роста, изучавшаяся в § 21. Поэтому рассмотрение материала § 44 нужно предварить повторением модели ограниченного роста.

Ответ на вопрос 1 к § 44 содержится в объяснительном тексте параграфа, остальные задания направлены на то, чтобы учащиеся твердо усвоили основные компоненты определения управления. К этим компонентам относятся в частности такие характеристики управляющего объекта, как наличие целевых установок (целеполагание управления) и возможностей воздействия на управляемый объект, а также характеристики управляемого объекта, выражающие его функциональные возможности, и допустимые на него воздействия. Поэтому ответы на вопросы задания 2 удобно оформить в виде таблицы 9.

Таблица 9

Пункт задания	Цель управления	Управляемый объект		Управляющий объект	
		Название	Допустимые воздействия	Название	Возможные управляющие воздействия
а	Провести поезд по маршруту	Локомотив	Изменение скорости	Машинист	Увеличивать и уменьшать силу тока
б	Произвести посадку пасажиров в самолет	Пассажиры данного рейса	Передача информации	Диспетчерская служба аэропорта	Оповещение с помощью радио, табло
...	...	...	...	...	...

В задании 3 все процессы, кроме рисования пейзажа, могут рассматриваться как управленческие. Но от учащихся надо требовать, чтобы они при этом точно указывали цели управления, управляющий и управляемый объекты и характеристики, позволяющие осуществлять управление.

Аналогично строятся ответы на вопрос 5.

О содержании и назначении § 45 сказано выше. Чтобы ответить на вопрос 2 к этому параграфу, надо прежде всего вспомнить предположения, которые были сделаны при построении модели ограниченного роста. Эти предположения фактически выписаны в конце объяснительного текста § 21. Мы их здесь воспроизведем, чтобы легче было проводить дальнейшее обсуждение.

Предположения модели ограниченного роста:

— прирост возобновимого ресурса за единицу времени пропорционален уже имеющейся величине этого ресурса;

— сущестует некоторое предельное значение возобновимого ресурса;

— коэффициент прироста ресурса за единицу времени пропорционален разности между максимально возможным значением ресурса и величиной ресурса, имеющегося на данный момент времени.

К этим предположениям мы должны добавить предположения относительно изъятия ресурса. Вот это предположение:

— величина ежегодно изымаемого ресурса постоянна.

Вот эти четыре предположения, выделяющие существенные факторы и позволяющие описать их числовыми параметрами, и задают модель потребления возобновимых ресурсов.

При обсуждении вопроса 2 желательно поговорить о выявлении параметров модели потребления возобновимых ресурсов, что, без сомнения, полезно само по себе, но, кроме того, пригодится для выполнения задания 3. Здесь опять-таки к параметрам, описывающим модель ограниченного роста, надо добавить параметр R, задающий ежегодный уровень потребления. Вот полный список параметров модели потребления возобновимых ресурсов:

— начальная величина возобновимого ресурса $M(0)$;
— предельное значение ресурса L;
— коэффициент пропорциональности a в формуле для коэффициента прироста ресурса;
— время n;
— величина R ежегодного потребления возобновимого ресурса.

Формула, указывающая связь между этими параметрами, приведена в объяснительном тексте § 45.

Задание 3 направлено на подготовку к лабораторной работе 22. Это задание рекомендуется выполнить дома и обсудить его в начале выполнения лабораторной работы 22. Поскольку учащиеся уже знакомы с каким-либо языком программирования, желательно предложить им составить алгоритм и написать программу для вычисления количества ресурса через n лет согласно модели потребления возобновимых ресурсов. При этом на лабораторной работе 22 одна половина учащихся работает с электронной таблицей, другая — с созданной программой. Это позволит сравнить применение двух весьма различных на первый взгляд инструментов компьютерных технологий и оценить достоинства и недостатки каждого из них.

При использовании электронной таблицы как инструмента компьютерного моделирования нужно предложить учащимся построить график зависимости количества ресурса от времени.

Беря для R последовательно разные значения, например 5000, 5500, 5800, 6000, 6100, можно наблюдать одну и ту же картину: масса рыбы стабилизируется на некотором уровне, зависящем от значения R. Однако переход значения R через 6400 оказывается для природы непосильным бременем — вся популяция погибнет. Так, меняя значение R, учащиеся подбирают наибольшее допустимое значение изымаемого ресурса. Значение 6300 еще годится, а 6350 уже не годится. Значение 6310 годится, а 6340 приведет к гибели на 44-м году. Значение 6330 сдвигает границу краха на 84-й год и т. д.

В таблице 10 приведены соответствующие программы на языках QBasic и Pascal.

Таблица 10

QBasic	Pascal

INPUT ”Введи число лет”, N
INPUT ”Введи величину изымаемого ресурса”, R
M = 10000
L = 11000
K = 1.8
A = K/(L – M)
FOR I = 1 TO N
P = A*M*(L – M)
M = M + P – R
PRINT ”Прирост за год равен”, P
PRINT ”Ресурс после”, I
PRINT ”лет составляет”, M
NEXT I
	Var R, M, L, K, A : real;
	Var I : integer;
	Begin
	readln (N, R);
	M : = 10000;
	L : = 11000;
	K : = 1.8;
	A : = K/(L – M);
	for I : = 1 to N do
	begin
	P : = A*M*(L – M);
	M : = M + P – R;
	write (‘Прирост за год равен’, P, ‘Ресурс после’, I, ‘лет составляет’, M);
	end;
	End.

Один из важнейших выводов, который должны сделать учащиеся в результате выполнения данной лабораторной работы, состоит в обнаружении явления саморегуляции и стремлении ее к некоторому положению равновесия. Уже здесь целесообразно обсудить, что саморегулирование системы возможно только в том случае, если информация с выхода снова поступает на вход системы. Это должно рассматриваться как подготовка к введению понятия обратной связи.

Понятие обратной связи, хотя и появилось в кибернетике (т. е. науке об управлении), разумеется, связано не только с управлением, но и с понятием устойчивости системы к внешним воздействиям. Поэтому, кроме понятия обратной связи, важную роль играет понятие **гомеостаза** системы — положения равновесия, устойчивого при внешних воздействиях. По-видимому, нельзя эффективно объяснить понятие обратной связи и без введения понятия **регулятора**.

Вопросы 1—4 к § 46 направлены на проверку знания понятий, введенных в объяснительном тексте параграфа. Успешное выполнение задания 5 обеспечивается усвоением главного принципа: *изменение состояния объекта должно* *быть связано с этим состоянием*. Поэтому среди примеров, приведенных в задании 5, к взаимодействию по принципу обратной связи относятся те, которые перечислены в пунктах *а, в* и *г*. Обсуждение этого задания для каждого пункта должно начинаться с выделения пары *управляющий объект→управляемый объект*. Если имеется обратная связь, то от управляемого объекта к управляющему должна существовать передача информации, корректирующей управление. В указанных выше пунктах соответствующие объекты и обратная связь легко усматриваются. Что касается взаимодействия дождя и человека с зонтом, то, как ни распределяй роли управляющего и управляемого объектов, нельзя получить эффект обратной связи. Действительно, какова здесь пара *управляющий→управляемый*? *Человек→дождь* (человек управляет дождем)? Или *дождь→человек* (дождь управляет человеком)? Казалось бы, так можно сказать. Но какова тогда обратная информация от человека к дождю? Еще возможная пара *человек→зонт*. Но в чем состоит информационное общение человека с зонтом? Других объектов в этом примере нет, так что еще какие-то варианты управления по принципу обратной связи здесь тоже найти не удастся. Аналогично (и даже еще проще) разбирается пункт *д* задания 5.

В задании 2 к § 47 вводится понятие равновесия экологической системы как такого ее состояния, при котором параметры системы не меняются со временем. Положение равновесия не обязательно является гомеостазом системы, хотя гомеостаз — это обязательно равновесное состояние.

Из заданий к § 47 отметим также задание 3, поскольку обычно учащимися положительные обратные связи воспринимаются как деструктивные и психологически к ним складывается негативное отношение. Эволюция подразумевает развитие положительных тенденций, и положительная обратная связь здесь оказывает ускоряющее влияние. Действительно, чем больше особей с прогрессивным признаком, тем больше их приспособляемость, и тем самым воспроизводится больше особей с данным признаком.

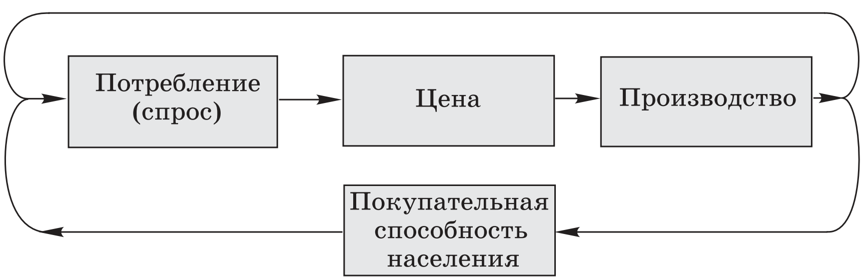

Рис. 26

Выполнение лабораторной работы 24 предлагается проводить в соответствии с приведенным в учебнике сценарием.

В § 48 соединяются понятия управления и принципа обратной связи. В теоретическом плане главное внимание следует уделить обоснованию большей эффективности управления по принципу обратной связи по сравнению с разомкнутым управлением. В объяснительном тексте § 48 этому посвящен пример со светофором, а в системе заданий — задачи 6 и 7. Ниже мы обсудим решение задачи 6; что касается задачи 7, то мы пока воздержимся от обнародования рецептов решения данной проблемы. С точки зрения принципа обратной связи решение лежит на поверхности, поэтому тем более интересным будет обсуждение этой задачи с учащимися.

В задании 4 сначала надо предложить учащимся определить положительные и отрицательные связи в трехзвенной схеме. Фактически это сделано в объяснительном тексте: из сказанного там вытекает, что связи *спрос→цена* и *цена→производство* положительны, а связь *производство→спрос* отрицательна. Чтобы встроить еще один черный ящик, надо понять, на что он влияет и что на него влияет. Ясно, что снижение покупательной способности населения уменьшает спрос, а рост производства увеличивает покупательную способность, поскольку вызывает рост доходов. Это значит, что обе связи положительны. Схема же выглядит так, как показано на рисунке 26.

Мы считаем, что человек, вступающий во взрослый мир, должен знать о глобальных проблемах и моделях, позволяющих рассматривать компетентно эти проблемы. Именно этому и посвящен заключительный параграф учебника. Задание 2 в этом параграфе — это снова приглашение к обсуждению (если хотите, дискуссии), подводящему итог изучению главы. Здесь мы, конечно, рассчитываем на опыт и знания учителя.

О распределении часов из резерва мы несколько раз говорили в ходе различных обсуждений, но в целом оставляем этот вопрос на усмотрение учителя. Желаем нашим коллегам успешной работы с учебником.

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ НА I ПОЛУГОДИЕ

ТЕМА	Базовый уровень			Углубленный уровень		
	Всего часов	Теория	Практика	Всего часов	Теория	Практика
1. Информация и информационные процессы. Язык как средство сохранения и передачи информации. Кодирование информации. Универсальность двоичного кодирования. Восстановление навыков работы на компьютере с основными средствами информационных технологий	6	4	2	6	4	2
2. Понятие информационной модели. Системный подход в моделировании	6	3	3	7	3	4
3. Алгоритмы и их свойства. Конечные автоматы. Распознаваемые языки. Машина Тьюринга	2	1	1	10	7	3
4. Основные направления информатики	1	1	—	3	1	1
5. Декларативная и процедурная информация. Простейшие базы данных. Обработка экспериментальных данных	7	4	3	8	4	4
6. Вспомогательный алгоритм. Метод пошаговой детализации. Понятие подпрограммы. Алгоритмические неразрешимые задачи. Рекуррентные и рекурсивные алгоритмы. Обработка массивов	7	3	4	14	5	9
7. Метод деления пополам. Количество информации (формула Хартли)	3	2	1	5	3	2

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ НА II ПОЛУГОДИЕ

ТЕМА	Базовый уровень			Углубленный уровень		
	Всего часов	Теория	Практика	Всего часов	Теория	Практика
8. Моделирование процессов живой и неживой природы. Нахождение границ адекватности модели	9	4	5	21	8	13
9. Датчики случайных чисел и вероятностные модели. Метод Монте-Карло	5	3	2	15	6	9
10. Высказывания. Операции над высказываниями. Алгебра высказываний. Дизъюнктивная нормальная форма	3	2	1	6	5	1
11. Отношения. Реляционные модели. Функциональные отношения. Предикаты. Кванторы. Логические основы реляционных баз данных	4	2	2	9	6	3
12. Экспертные системы	3	2	1	7	3	4
13. Основы логического программирования	—	—	—	5	4	1
14. Понятие управления. Понятие обратной связи. Построение управления по принципу обратной связи. Глобальные модели	7	4	3	11	6	5
15. Повторение. Резерв учителя	7	3	4	14	5	9
Итого	70	38	32	140	70	70

ОГЛАВЛЕНИЕ

Учебное издание

Гейн Александр Георгиевич

ИНФОРМАТИКА

Методические рекомендации

10 класс

Учебное пособие для общеобразовательных организаций

Зав. редакцией *Т. А. Бурмистрова*
Редактор *О. В. Платонова*
Младший редактор *Е. А. Андреенкова*
Художественный редактор *О. П. Богомолова*
Технические редакторы *Е. А. Сиротинская, Р. С. Еникеева, Н. В. Лукина*
Корректоры *Д. Б. Трубникова*

Подписано в печать 09.09.13. Формат $60 \times 90^1/_{16}$. Гарнитура Школьная.
Печ. л. 10,25. Заказ №

Открытое акционерное общество «Издательство «Просвещение».
127521, Москва, 3-й проезд Марьиной рощи, 41.