А. Г. ГЕЙН, Н. А. ЮНЕРМАН, А. А. ГЕЙН

ИНФОРМАТИКА

11 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. 11 класс : учеб. пособие для общеобразоват. организаций / А. Г. Гейн, Н. А. Юнерман, А. А. Гейн. — 2-е изд. — М. : Просвещение, 2017. — 240 с. : ил. — ISBN 978-5-09-044797-3

Книга предназначена для учителей, работающих по учебнику «Информатика и ИКТ, 11 класс» авторов А. Г. Гейна, А. И. Сенокосова. Она полностью соответствует учебнику по структуре и содержанию, содержит подробные методические комментарии к теоретическому материалу и практическим заданиям.

УДК 372.8:004
ББК 74.263.2

Учебное издание

Гейн Александр Георгиевич
Юнерман Нина Ароновна
Гейн Андрей Александрович

ИНФОРМАТИКА

Методические рекомендации

11 класс

Учебное пособие для общеобразовательных организаций

Зав. редакцией *Т. А. Бурмистрова*, редактор *О. В. Платонова*, художник *О. П. Богомолова*, художественный редактор *О. П. Богомолова*, компьютерная графика *Е. В. Ветчинкина*, технический редактор и верстальщик *Н. В. Лукина*, корректор *А. К. Рудзик*

Подписано в печать 04.09.13. Формат $60 \times 90^1/_{16}$. Гарнитура Школьная. Печ. л. 15. Заказ №

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

ISBN 978-5-09-044797-3

ВВЕДЕНИЕ

Данное методическое пособие является продолжением пособия для учителя к учебнику информатики для 10 класса и предназначено учителям, работающим по учебнику «Информатика. 11 класс» (авторы: А. Г. Гейн и А. И. Сенокосов). 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. Повторим кратко дидактические установки авторов, изложенные более развернуто в упомянутом пособии для учителя к учебнику для 10 класса.

В соответствии с указанным стандартом курс информатики в заключительном звене школьного образования имеет два уровня — базовый и углубленный. Базовый уровень призван обеспечить поддержку тех 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 лишь в комплексе с углубленным изучением математики. Поэтому в целевых установках формулируется освоение и систематизация 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. Одним из таких элективных курсов может стать курс, ориентированный на подготовку к ЕГЭ по информатике. Поэтому в наших рекомендациях мы достаточно последовательно обсуждаем вопросы, связанные с теми темами, которые традиционно представлены в ЕГЭ. Надо иметь в виду, что ЕГЭ по информатике является экзаменом по выбору, и его разработчики не скрывают, что включаемые в него задания предназначены не для оценки общего уровня подготовки по информатике в рамках, предусмотренных упомянутым выше Стандартом, а отбор в вузы, предъявляющие повышенные требования к подготовке по данному профилю. Такая установка нашла свое отражение, в частности, в том, что в нынешнем ЕГЭ практически отсутствуют задания по базовым информационным технологиям, а основное внимание сосредоточено на вопросах кодирования информации и измерении ее количества, математической логике, алгоритмизации и программированию. Мы постарались учесть это обстоятельство как при создании учебника, так и в предлагаемых вашему вниманию методических рекомендациях[1].

Хотя мы уделили сейчас немалое внимание элективным курсам, надо иметь в виду, что учебник в первую очередь предназначен для изучения информатики и информационных технологий на базовом и углубленном уровнях. 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.

[1] По тем же причинам в тематическом планировании часы на подготовку к ЕГЭ выделены нами только для углубленного уровня изучения информатики.

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.

В учебнике содержится значительное число заданий, обеспечивающих развитие универсальных учебных действий учащихся. В данных методических рекомендациях мы комментируем такие задания, чтобы учителю было легче расставить нужные акценты.

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 являются:

- 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. Перед создателями этого комплекта стояли непростые дидактические задачи. Во-первых, мы были обязаны исходить из того, что учащимися в 8—9 классах освоен базовый курс информатики в объеме, предусмотренном Федеральным государственным образовательным стандартом общего 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, потребовали принятия компромиссных решений как по отбору содержания, так и по структурированию учебного материала. Во-вторых, учебник призван обслуживать как базовый, так и углубленный компоненты курса информатики. 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. 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 уже освоены учащимися. 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.

3. 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.

4. 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 соответствую-

щего 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.

5. Мы учитываем, что 11 класс, особенно второе полугодие, — это время активной подготовки учащихся к выпускным и вступительным экзаменам. 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. На наш взгляд, изучение главы 1 должно предшествовать изучению главы 4, а главы 2 — изучению главы 3. Что касается глав 5—7, то они практически независимы друг от друга и от предшествующих глав. Надо только иметь в виду, что материал этих глав преимущественно относится к углубленному уровню изучения информатики.

6. Каждая из глав 5—7 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 11 класса — в нем семь 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.

Ниже приводится тематическое планирование изучаемого материала, согласованное с учебником «Информатика и ИКТ» для 11 класса. Оно вовсе не носит характер нормативного документа, а выполняет для учителя роль ориентира в распределении времени. Ясно, что конкретные условия реализации курса могут потребовать иных решений в этом вопросе.

* * *

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.

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

В тематическом планировании указано как общее количество часов, отводимых на изучение данной темы, так и их распределение на изучение теоретической части и проведение компьютерных лабораторных работ.

Базовый уровень: *2 ч в неделю, всего 72 ч, из них 42 ч теории + 30 ч компьютерного практикума.*

Углубленный уровень: *4 ч в неделю, всего 140 ч, из них 82 ч теории + 58 ч компьютерного практикума.*

Тема	Базовый уровень			Углубленный уровень		
	Всего часов	Теория	Практика	Всего часов	Теория	Практика
1. Информационная культура общества и личности. Социальные эффекты информатизации. Восстановление навыков работы на компьютере	5	3	2	5	3	2
2. Методы работы с информацией. Свертывание информации	4	3	1	4	3	1
3. Моделирование как базовый элемент информационной грамотности. Моделирование в задачах управления	5	3	2	6	4	2
4. Международные исследования по оценке уровня информационной грамотности учащихся	1	1	0	1	1	0
5. Кодирование числовой информации. Системы счисления. Алгоритмы перевода из системы счисления с одним основанием в систему счисления с другим основанием	4	2	2	7	4	3
6. Кодирование символьной информации. Кодовые таблицы. Кодирование изображений. Универсальность двоичного кодирования	6	4	2	7	5	2

Продолжение

Тема	Базовый уровень			Углубленный уровень		
	Всего часов	Теория	Практика	Всего часов	Теория	Практика
7. Кодирование с заданными свойствами. Коды, исправляющие ошибки. Префиксные коды и алгоритмы сжатия символьной информации. Алгоритмы сжатия видеоинформации. Сжатие звуковой информации	0	0	0	7	5	2
8. Логические основы работы компьютера. Математические основы работы арифметического устройства. Булевы функции. Логика оперативной памяти компьютера	1	1	0	4	4	0
9. Представление чисел в компьютере. Особенности компьютерной арифметики	0	0	0	7	4	3
10. Основные информационные объекты, их создание и обработка. Средства и технологии создания и обработки текстовых информационных объектов. Компьютерные словари и системы перевода текстов. Средства и технологии создания и обработки графических информационных объектов. Компьютерные презентации	23	10	13	23	10	13
11. Телекоммуникационные сети и Интернет. Поисковые системы в Интернете. Сервисы Интернета. Интернет-телефония. Правовые вопросы Интернета. Безопасность и этика Интернета. Защита информации	14	9	5	14	9	5
12. Математические методы исследования алгоритмов. Понятие лимитирующей функции и инварианта	0	0	0	8	5	3
13. Графы и алгоритмы на графах. Определения и простейшие свойства графов. Способы задания графов. Алгоритмы обхода связного графа. Понятие стека. Деревья и каркасы	2	2	0	20	9	11

Продолжение

Тема	Базовый уровень			Углубленный уровень		
	Всего часов	Теория	Практика	Всего часов	Теория	Практика
14. Игровые модели. Граф игры. Стратегия игры. Выигрышные и проигрышные позиции. Инвариант игры. Стратегии на основе инварианта. Функции выигрыша. Стратегии на основе функции выигрыша	3	3	0	13	7	6
15. Повторение. Подготовка к ЕГЭ	0	0	0	12	8	4
16. Резерв учителя	4	1	3	2	0	2
Итого	72	42	30	140	81	59

ГЛАВА 1

Информационная культура общества и личности

Материал, представленный в первой главе учебника, в большей своей части не имеет сегодня аналогов в учебной литературе по информатике для общеобразовательной школы. Здесь в едином узле представлено несколько ключевых линий Федерального стандарта и обязательного минимума:

- формирование информационного мировоззрения;
- социальная информатика;
- особенности обработки информации человеком.

С образовательной точки зрения освоение этих линий можно в концентрированном выражении назвать воспитанием у учащихся основ информационной культуры.

Тема 1. Понятие информационной культуры

Само понятие «информационная культура», появившееся в нашей стране в 70-е гг. прошлого столетия, первоначально функционировало в области библиотековедения и рассматривалось в первую очередь как умение работать с библиографической информацией. В книге профессора Г. Г. Воробьева «Твоя информационная культура», вышедшей в 1988 г., это понятие трактовалось уже иначе — как культура рациональной и эффективной организации интеллектуальной деятельности людей. Научно-технический прогресс и активное внедрение в жизнь компьютерной техники привели к тому, что в середине 90-х гг. под информационной культурой стали понимать владение компьютерными технологиями для обработки информации. Наиболее ярким проявлением этой позиции явился курс «Информационная культура», разработанный под руководством Ю. А. Первина, включавший в себя полностью курс информатики и рассчитанный на реализацию с 1 по 11 класс. Исследования в области информационной культуры, проводившиеся в нашей стране в течение более тридцати лет, способствовали в конечном итоге выделению компонентного состава этого понятия, который и приведен в учебнике. Тем не менее нет никаких оснований считать, что это окончательная редакция содержания понятия «информационная культура». Отсутствие единой точки зрения на вопрос, каким должно быть определение информационной культуры, привело нас к решению не формулировать в объяснительном тексте какое-либо определение этого понятия, которое может быть воспринято учениками как каноническое. Тем не менее мы сочли необходимым (и полезным) познакомить учащихся с другими определениями информационной культуры (см. задание 2 к § 1).

Хотим обратить внимание, что в компонентном составе данного понятия отражены разные образовательные составляющие. Во-первых, это знания и умения, в том числе чисто технологические. Во-вторых, это готовность и способность принимать решения, т. е. действовать в соответствии с воспринятой и преобразованной информацией; в современной педагогике такую готовность и способность связывают с приобретением компетентности в той или иной сфере, в данном случае речь идет об информационной компетентности. В-третьих, это нравственно-ценностная составляющая, выраженная правовыми и этическими нормами. Информационные знания и особенно технологические умения в значительной степени уже присутствовали в предшествующие годы изучения информатики. В предстоящем курсе им также будет уделено внимание (особенно в главе 3). Правовые и этические аспекты также будут неоднократно рассматриваться как в главе 1, так и в главе 4, посвященной глобальным телекоммуникациям. Что касается развития умений принимать информационно обоснованные решения, то речь о них будет идти и в теоретической части, но главная нагрузка здесь ложится на самостоятельную деятельность учащихся, которая организуется через решение специально подобранных заданий и через выполнение работ компьютерного практикума. В нашей книге мы в соответствующих местах будем давать методические рекомендации.

Надо сказать, что термин «информационная культура» не имеет международного аналога; в зарубежных исследованиях фигурирует термин «information literacy» (в дословном переводе — информационная грамотность), о котором мы будем подробно говорить позже, обсуждая § 2.

Методическая проблема в освоении материала § 1 состоит, на наш взгляд, в том, что информационной культуре, как и культуре вообще, нельзя научить; она воспитывается в ходе соответствующей, можно сказать, повседневной информационной деятельности учащихся. Данный параграф имеет своей целью лишь сформировать представление об информационной культуре, ее основных компонентах. Поэтому наиболее естественной формой ознакомления учащихся с этим материалом является беседа, дискуссия и/или выступление учащихся с заранее подготовленными краткими сообщениями.

Вопросы, сформулированные в заданиях 1 и 3 к этому параграфу, носят чисто репродуктивный характер. Они ориентируют учащихся на твердое знание компонентного состава понятия «информационная культура» и понимание содержания понятия «информационное мировоззрение». В последующем при изучении нового материала полезно акцентировать внимание учащихся на том, как проявляются те или иные составляющие этих понятий. Разумеется, при этом следует избегать навязчивой морализации.

О целевых установках задания 2 речь шла выше. Его выполнение требует от учащихся умений вдумчивого анализа. Вот как такой анализ может осуществляться.

Начнем с определения, данного профессором Н. И. Гендиной [1]. Уже первый постулат этого определения: «Информационная культура — одна из составляющих общей культуры человека» — означает, что информационная культура должна наследовать основные характеристики общечеловеческой культуры. Приведем некоторые из них.

1. Культура — ценностно-ориентированное явление. В личностном плане она определяет принципы, позволяющие выделить нормы поведения культурного человека (этика).

2. Культура неразрывно связана с той или иной конкретной цивилизацией (поэтому мы можем говорить о древнегреческой культуре или современной европейской культуре и т. п.). Можно сказать, что культура является отражением цивилизации в форме духовной надстройки.

3. Культура национальна в своем генезисе, интернациональна в своем значении.

4. Развитие культуры — это общественно значимое ее изменение, проявляющееся через артефакты, создаваемые отдельными личностями и интегрируемые в целостное духовное наследие общества. Культура личности проявляется в позитивной общественно значимой направленности деятельности данной личности.

[1] И это определение, и определения других авторов, приведенные в указанном задании учебника, цитируются нами по кн.: Формирование информационной культуры личности: теоретическое обоснование и моделирование содержания учебной дисциплины [Текст] / Н. И. Гендина, Н. И. Колкова, Г. А. Стародубова, Ю. В. Уленко. — М.: Межрегиональный центр библиотечного сотрудничества, 2006 (эта книга сегодня считается одним из наиболее авторитетных источников в вопросах информационной культуры).

На наш взгляд, важно, чтобы в ходе обсуждения данного определения учащимися были выявлены эти четыре характеристики (тем более что они, пусть не в такой конкретно форме, практически прописаны в объяснительном тексте параграфа). Следующим шагом в выполнении этого задания является предложение учащимся привести примеры проявления этих характеристик в информационной сфере, т. е. выработка понимания особенности понятия «информационная культура» по отношению к родовому понятию «культура личности». Поиск таких примеров естественно сформулировать в качестве домашнего задания.

Вполне возможно, что учащиеся сформулируют и другие черты, характерные для понятия «культура». Ни в коем случае не следует отвергать предложения учащихся. Их также надо проанализировать (не исключено, что некоторые из них являются переформулировками или частными случаями уже указанных характеристик) и тоже предложить привести примеры их проявления в информационной сфере.

Второй постулат определения Н. И. Гендиной состоит в обязательном включении информационного мировоззрения в состав понятия «информационная культура». Это почти очевидное требование, поскольку совокупность взглядов человека на окружающий его мир и самого себя, разумеется, является важнейшим компонентом культуры личности. Можно только удивляться, что до Н. И. Гендиной на это практически никто не обращал внимания. Понятие информационного мировоззрения раскрыто в учебнике, поэтому здесь мы на нем подробно останавливаться не будем.

Наконец, последний концентр определения Н. И. Гендиной — это деятельностный компонент культуры. Сюда относится владение способами производства и способами коммуникации, способами сохранения, развития и передачи культурного наследия. Он, конечно, тоже всегда присутствует в расшифровке понятия культуры, что бы мы ни говорили о культуре. К сожалению, в этой части определения преобладает, на наш взгляд, эгоцентричный акцент в деятельности человека: главное здесь — достижение индивидуального комфортного существования в информационном обществе, а это противоречит сформулированному выше пункту 4 в характеристике понятия культуры.

Из этого обстоятельного обсуждения становится видно (и школьники тоже могут обратить на это внимание), что второй и третий постулаты определения Н. И. Гендиной входят в понятие культуры. Но ведь и так сказано, что информационная культура — это часть общей культуры. Какую роль играют эти два концентра? Они очерчивают границы, показывая, какая именно часть общей культуры относится к информационной.

Резюмируя это обсуждение, можно сказать, что **информационная культура** — часть общей культуры человека данного общества, связанная с восприятием, освоением, передачей и использованием информации в рамках той информационной среды, которая создана обществом на том или ином этапе его развития [1]. Если учащиеся в дальнейшем будут пользоваться этим определением, то они обязательно должны уметь развертывать его в те основные постулаты информационной культуры, речь о которых шла выше.

Определение, данное Ю. С. Зубовым, с очевидностью является составной частью определения Н. И. Гендиной. Определение, принадлежащее В. Н. Михайловскому, расставляет акценты иначе. Во-первых, выделяется коммуникативная составляющая, которой отведена главенствующая роль. Во-вторых, указываются особенности мышления. На какие особенности этого определения следует, по нашему мнению, обратить внимание учащихся?

Начнем с того, что коммуникативная составляющая может быть совершенно независимо выделена учащимися в ходе обсуждения понятия «культура» (как о том говорилось выше). И действительно, способы общения, нормы общения, уровень развития сферы общения — все это существенные характеристики культуры общества и личности. Никто сегодня, к примеру, общение даже с высшими лицами государства не начинает со слов «О, великий! Припадаю пред тобой 20 раз на живот и 20 раз на спину...» или «Ваше высочество!..»; все ограничивается словами «Уважаемый господин...». И мы, анализируя третий концентр определения Н. И. Гендиной, также указывали на владение способами коммуникации как на элемент информационной культуры. Принципиальное отличие состоит в том, что у Н. И. Гендиной эти способы направлены на удовлетворение индивидуальных информационных потребностей, а в определении В. Н. Михайловского такого явно эгоцентричного уклона нет. Однако определение В. Н. Михайловского страдает принципиальным недостатком: оно определяет информационную культуру исключительно новыми явлениями в информационной сфере. Получается, что в докомпьютерную и дотелекоммуникационную эпоху никакой информационной культуры не было и быть не могло. На самом деле информационная культура как составная часть общей культуры существовала всегда, но для информационного общества она начинает играть особую роль и прежде всего именно в тех направлениях, которые сформулировал В. Н. Михайловский.

[1] Это определение выработано совместно А. Г. Гейном и О. К. Громовой.

Задание 4 играет особую роль. Во-первых, оно нацелено на повторение понятий, изучавшихся в курсе информатики ранее. Общее понятие технологии обсуждалось в курсе 10 класса: под технологией понимают организацию процесса, обеспечивающую гарантированное получение результата требуемого качества. Компьютерные технологии — это организация процесса обработки информации с помощью программного обеспечения, функционирующего на компьютерной технике. Понятие информационной технологии шире — средства обработки информации вовсе не обязаны быть компьютерными. Однако здесь не только разница в используемых технических приемах, но и методологическая разница — информационные технологии включают в себя целый ряд эвристических моментов наряду с формальными методами преобразования информации. О таких, во многом эвристических моментах информационной деятельности человека будет идти речь в последующих параграфах главы 1. Поэтому данное задание является подготавливающим, в первую очередь к изучению материала из § 2.

Мы рекомендуем задание 4 задать на дом, а затем обсудить его выполнение на уроке.

Без преувеличения можно сказать, что § 2 является ключевым для всей главы 1. Как сказано в заглавии параграфа, информационная грамотность является базовым элементом информационной культуры. И дело не только в том, что составляющие информационной грамотности во многом являются детализирующей расшифровкой компонентов информационной культуры (что само по себе тоже немаловажно), а в том, что в ней сосредоточено все то, что в информационной культуре составляют знания и умения, т. е. то, чему фактически можно научить. Информационная культура — это такая же надстройка над информационной грамотностью, как общая культура человека над его общей грамотностью.

Грамотность служит базисом культуры, ибо это технологии, обеспечивающие человеку как личности восприятие культуры того общества, в котором он живет. Уровень грамотности определяет возможность человека быть приобщенным к культуре. Но даже высокий уровень грамотности (и более обще — образованности) не гарантирует человеку личной культуры. Грамотность может быть передана (и приобретена) через научение, культура всегда результат воспитания. Грамотный человек, воспитанный в рамках одной культуры, как правило, остается грамотным и в рамках другой культуры (близкой с точки зрения уровня цивилизации), однако он нередко оказывается в этой другой культуре на существенно более низком уровне и зачастую воспринимается как малокультурный.

Грамотность обладает той отличительной особенностью, что она направлена на удовлетворение собственных потребностей человека (включая самообразование) в его жизни и продуктивной деятельности. Культура личности — это прежде всего акцент на общественной направленности и нравственно-ценностной ориентации.

Культура общества и личная культура человека (как члена этого общества) — эти понятия связаны проекционной связью: мы воспринимаем человека культурным, соотнося его личные качества (в том числе поведенческие) с культурными установками общества. А вот понятия «грамотность общества» не существует; мы оцениваем грамотность каждого отдельного человека, грамотность общества в целом обсуждается разве лишь как статистическая оценка, позволяющая улавливать тенденции в развитии грамотности членов общества.

Мы предприняли такое обстоятельное обсуждение соотношения между грамотностью и культурой в связи с тем, что до сих пор не утихают споры, не совпадают ли понятия информационной грамотности и информационной культуры. На наш взгляд, разграничение этих понятий необходимо и плодотворно.

Вернемся к рассмотрению материала § 2.

После определения информационной грамотности сразу показано, что оно имеет два блока. Первый из них связан с внешними действиями по формулированию и удовлетворению информационной потребности. Второй блок — это внутренняя работа человека, индуцированная осмыслением полученной информации.

Первый блок нередко в той или иной степени представлен во всех курсах информатики, которые сейчас ведутся в общеобразовательной школе. Иногда это сводится к освоению довольно рутинных приемов организации поиска информации в базах данных, но во многих курсах фактически эти вопросы так или иначе затрагиваются под углом зрения моделирования и формализации. Ведь любое моделирование начинается с определения того, какая информация является существенной для достижения поставленной цели, как эту информацию получить и как она будет представлена. А формализация — это уже непосредственно способ представления запросов и самой информации. Мы в учебнике остановились на минимальном уровне, напомнив о работе с базами данных. Учителю необходимо, на наш взгляд, здесь дать более широкий обзор в соответствии с тем, что уже знают и какими навыками владеют его подопечные. Важно также понимать, что здесь устанавливается прямая связь с материалом § 6, в котором будут повторены основные положения информационного моделирования, которое, мы не сомневаемся, уже изучалось школьниками в предшествующие годы.

Второй блок в большинстве курсов информатики отсутствует, хотя в федеральном компоненте Государственного образовательного стандарта по информатике и информационным технологиям, принятого в 2004 г., прямо указывается на необходимость воспитания у школьников избирательного отношения к полученной информации (см.: Федеральный компонент Государственного образовательного стандарта по информатике и информационным технологиям // Информатика и образование. — 2004. — № 4. — С. 2—33). Такая постановка вопроса тесно связана с развитием у человека так называемого критического мышления. Мы уделим этому отдельное внимание при обсуждении учебного материала, представленного в § 4.

Развернутое определение информационной грамотности, приведенное далее на с. 9 учебника, в основном совпадает с тем определением, которое дал Д. Браун[1], представитель Международной ассоциации школьных библиотек (IASL). Отметим, что это определение не зависит от способов реализации работы с информацией как физических (компьютеров, библиотечных каталогов и т. п.), так и человеческих особенностей (в том числе знания этих средств и умений работать с ними). Такое определение годится и для V в. до н. э., и для XXV в. н. э. На наш взгляд, это — ключевое определение, из которого вытекает все остальное.

Обратим внимание, что развернутое определение информационной грамотности имеет двоякое назначение. Для *учащихся* оно определяет алгоритм грамотной работы с информацией. Было бы чрезвычайно полезным каждому человеку выработать у себя привычку в работе с информацией опираться на этот алгоритм, не пропуская в его исполнении ни одного шага. Разумеется, по мере его освоения, как и большинство алгоритмов, исполняемых человеком, он должен принять свернутую форму: ведь если каждый раз, обрабатывая информацию, человек будет обдумывать, какой шаг алгоритма теперь будет им исполняться, то он уподобится сороконожке, задумавшейся, с какой ноги ей следует сделать очередной шаг.

Для *учителя* данное развернутое определение представляет собой программу[2] обучения: чему учить школьника, что-

---

[1] См.: Браун Д. Р. «Модель решения проблем» как средство развития информационной грамотности: комплексный подход // Библиотека в школе. — 2006. — № 22. — С. 6—11.

[2] Термин «программа» употреблен здесь не в привычном для информатики смысле «алгоритм, записанный на языке команд исполнителя», а в смысле описания целей и задач.

бы он стал информационно грамотным. Каждый пункт этого алгоритмического (с точки зрения учащегося!) описания информационной грамотности является умением, которое, вообще говоря, не формализуемо и представляет собой эвристический элемент. Его освоение — это и есть та учебная задача, на решение которой должны быть нацелены усилия учителя.

Хорошо известно, что методически оправданно отработать каждое умение отдельно, а затем уже отрабатывать весь комплекс умений как единую систему. В данном случае такой подход был бы тем более продуктивен, поскольку умение, представленное любым отдельным пунктом, вовсе не является элементарным и, в свою очередь, требует применения весьма непростых подходов в обучении. Добавим еще, что весьма часто наблюдается совмещение или замещение одних действий другими: например, оценка качества информации (четвертый пункт определения) подменяется собственным отношением к ней (пятый пункт) или наоборот. К сожалению, сегодня для реализации такой методики отсутствует банк необходимых дидактических материалов. Причина здесь, конечно, в том, что педагогика, изучающая проблемы массового образования, только начала заниматься этими вопросами. Мы приводим в учебнике несколько разработанных нами заданий, направленных на отработку отдельных умений информационной грамотности; увы, пока это все, что нам удалось сделать в этом направлении. Мы весьма рассчитываем на инициативу учителя в этих вопросах.

Отметим также, что умения, связанные с формулированием информационной потребности, выработкой стратегии поиска информации, ее поиском и оценкой качества, будут подробнее рассматриваться в § 40 главы 4. А в этом параграфе мы сочли необходимым уточнить понятия, относящиеся к характеристикам информации с точки зрения ее использования: полнота, достоверность, актуальность и объективность.

Переходя к обсуждению вопросов и заданий к § 2, отметим, что, на наш взгляд, ответы на первые четыре вопроса учащиеся легко сформулируют, опираясь на объяснительный текст параграфа. Также не должно вызвать затруднений выполнение задания 5, хотя здесь можно ожидать некоторой дискуссии между учащимися. На наш взгляд, чрезвычайно полезным было бы привлечение в качестве аргументов такой дискуссии извлечений из закона о рекламе, принятого Государственной думой РФ. В этом случае обсуждение задания 5 могло бы быть построено в два этапа: сначала недолгое обсуждение в классе, затем задание на дом подготовить аргументы к продолжению дискуссии, в частности поиск законодательных документов по данному вопросу, и, наконец, на следующем занятии подведение итогов дискуссии (фактически это и есть выполнение пункта *б* данного задания).

Наиболее известный пример, дающий ответ на вопрос задания 6, к тому же воспетый А. С. Пушкиным, — это то, что наблюдаемое человеком движение Солнца по небосводу не соответствует действительности. Как все знают, Земля движется вокруг Солнца, а не наоборот.

При выполнении задания 7 также не исключена дискуссия: кому-то из учащихся данной информации может показаться достаточно. На самом деле в пункте *а* хотелось бы знать, будет ли организовано питание, какой вид транспорта и как будет задействован и т. п. Можно предложить учащимся обсудить задание 7а с родителями: это, на наш взгляд, будет способствовать пониманию детьми того, о чем беспокоятся родители, отправляя их в подобное путешествие. Но, разумеется, предлагать такое обсуждение надо крайне деликатно, ни в коем случае не объявляя его обязательным, поскольку обстановка в семье может оказаться неблагоприятной для подобных обсуждений.

Аналогично строится обсуждение задания 7б.

Задание 8 еще раз обращает внимание учащихся на важность воспитания в себе информационной культуры и освоения информационной грамотности как ее базового компонента.

Начиная обсуждение материала, составившего суть § 3, вернемся к определению информационной культуры, которое дано В. Н. Михайловским. Нам совсем не хотелось бы, чтобы у читателя создалось ощущение поверхностности его подхода в таком важном вопросе, как определение одного из фундаментальных понятий. Надо точно понимать, что понятие «информационная культура» возникло именно в связи с вступлением общества в информационную фазу своего развития. До начала этого процесса не было нужды в таком понятии, так же как не было нужды в строгом понятии алгоритма до тех пор, пока в XX в. не возникла в этом потребность сначала в связи с рядом логико-математических проблем, а затем опять-таки вследствие бурного развития информатики и вычислительной техники. Как мы уже говорили, в определении В. Н. Михайловского в точности зафиксирована необходимость возникновения понятия «информационная культура» в связи с переходом к жизни в информационном обществе. И это мотивирует рассмотрение понятий информатизации и информационного общества в рамках данной главы, хотя, разумеется, эти понятия в той или иной степени уже знакомы учащимся по предшествующему курсу информатики: в соответствии с федеральным компонентом Государственного образовательного стандарта по информатике и информационным технологиям они входят в содержание базового курса 8—9 классов. В частности, поэтому нами принят довольно конспективный стиль изложения материала. Напомнив определение информатизации, мы кратко рассматриваем те аспекты, которые возникают в связи с этим процессом в различных сферах жизни и деятельности человека. Мы считаем полезным, если учитель будет побуждать учащихся к поиску примеров конкретного проявления указанных в учебнике эффектов для каждой такой сферы.

Мы воздержались от того, чтобы формулировать в учебнике определение информационного общества. В научных статьях и учебных пособиях приводятся разные формулировки этого понятия. Но во всех определениях фигурирует по крайней мере одна из тех двух характеристик, которые приведены в нашем учебнике:

- производство информационных товаров и оказание информационных услуг становятся ведущей отраслью хозяйства, и в этой сфере становится занятой основная часть трудоспособного населения;
- каждому члену общества предоставлены равные возможности в получении информации, позволяющей полноценно реализовать себя в профессиональной и общественной деятельности, обеспечивающей реализацию гражданских прав и удовлетворение потребностей в области культуры, повышающей комфортность бытовой сферы.

Конечно, уровень детализации в разных определениях может быть различным, поэтому здесь важно не количество характеристик. Скажем, в определении информационного общества, которое приведено в книге Н. И. Гендиной и др., упомянутой в сноске на с. 13, сформулировано три признака:

- большинство работающих занято в информационной сфере, т. е. сфере производства информации и информационных услуг;
- обеспечена техническая, технологическая и правовая возможность доступа любому члену общества практически в любой точке территории и в приемлемое время к нужной ему информации (за исключением военных и государственных секретов, точно оговоренных в законодательных актах);
- информация становится важнейшим стратегическим ресурсом общества и занимает ключевое место в экономике, образовании и культуре.

Первый и третий признаки в совокупности совпадают с первой из характеристик информационного общества, сформулированных в нашем учебнике. Общий смысл вторых характеристик в обоих определениях тоже близок друг к другу. Имеется, однако, и весьма существенное различие.

В приведенном нами определении речь идет о равенстве возможностей получения информации, причем независимо от того, необходима ли эта информация для развития человека как индивидуума или как социально активного члена общества. Гипотетически можно представить, что этот принцип выполняется и для общества, в котором нет развитой информационной инфраструктуры. Но тогда не будет выполняться первый принцип, и потому такое общество все равно не может рассматриваться как информационное. Из этого обсуждения следует, что в определении информационного общества нельзя ограничиваться формулировкой только второго принципа (что, однако, встречается в литературе, посвященной проблемам информационного общества). Что касается определения, приведенного в книге Н. И. Гендиной, то его основным недостатком является полная правовая несостоятельность. Ведь, кроме военной и государственной, существует коммерческая тайна, медицинская тайна, тайна переписки, конфиденциальные сведения о каждом человеке (например, тайна усыновления). Перечислить в определении все виды «тайн» просто невозможно, кроме того, у каждого типа тайны имеются свои четко обозначенные границы доступа, и нередко они являются ареной весьма упорной борьбы разных сторон. Например, сегодня в западных странах правоохранительные органы добиваются у законодателей разрешения на получение информации об интернет-контактах пользователей, а провайдеры активно этому сопротивляются.

В других определениях информационного общества, которые имеются в литературе (в том числе в школьных учебниках) или Интернете, акцент обычно делается на какой-то одной из двух характеристик; возможно, это последствия истории появления термина «информационное общество» [1]. Мы уже обсудили выше, что ограничиваться только второй характеристикой нельзя. Но нельзя ограничиваться и первой, поскольку тогда можно получить диктатуру с высоко информатизированным производством и мощными средствами информационного подавления. И хотя, как мы отме-

[1] Понятие информационного общества существенно «моложе» понятия «информационная культура». Впервые этот термин был употреблен в выступлениях президента США Б. Клинтона и госсекретаря США А. Гора. И это был в первую очередь термин политического пиара, позволивший уйти от термина «посткапитализм» (который и сегодня рассматривается как синоним термина «информационное общество»), поскольку само слово «капитализм» далеко не у всех вызывает положительные эмоции. Однако здесь, что случается крайне редко, политики попали в точку. Термин оказался весьма продуктивным, ему было дано вполне конкретное содержание, и он начал работать уже в научной сфере — экономике, социологии и т. д.

чали выше, второй из признаков информационного общества, получивший название **принципа информационной открытости**, скорее всего, долго еще будет вызывать споры о границах его применимости, его выполнение является непременным условием наступления той фазы развития общества, которая определяется как информационная.

Нам представляется весьма полезным обсуждение с учащимися, почему в определении информационного общества необходимо присутствие обоих признаков. Это довольно стандартный методический прием: если в определении понятия фигурирует несколько компонентов, то следует продемонстрировать учащимся роль и необходимость присутствия каждого из них.

Как и для § 1, рассмотрение материала § 3 естественно проводить в форме беседы, дискуссии и/или выступлений учащихся с заранее подготовленными краткими сообщениями, весьма полезна подготовка компьютерных презентаций о роли информатизации разных сфер жизни человека и общества. Мы уже кратко обсуждали полезность побуждать учащихся к поиску примеров конкретного проявления указанных в учебнике как позитивных, так и негативных эффектов для каждой такой сферы. Среди негативных социальных явлений, вызванных информатизацией общества, особого внимания заслуживает проблема, связанная с использованием средств глобальных телекоммуникаций для массовых манипуляций человеческим сознанием. Об этом будет идти речь в главе, посвященной Интернету, но постановочные вопросы, позволяющие мотивировать учащихся на изучение методов работы с информацией, здесь весьма своевременны и полезны.

Еще более серьезной проблемой является противодействие нейролингвистическому программированию. Так называют технологии, которые позволяют управлять поведением человека при воздействии на него заранее внедренными в подсознание сигналами. Хотя это тоже относится к информационному взаимодействию между людьми, но в большей степени оно связано с психическими механизмами, и мы не сочли возможным в учебнике уделить внимание данной проблеме, однако в зависимости от профиля класса (в классах социального или психологического профиля) можно предложить учащимся подготовку сообщений на эту тему.

Компьютерный практикум, предусмотренный тематическим планированием, не связан с теоретическим материалом данной темы. Он направлен на восстановление навыков работы на компьютере с использованием изученных ими ранее информационных технологий. Для базового курса информатики можно ограничиться повторением основных приемов работы с текстовым редактором и электронной таблицей. Для профильного курса мы рекомендуем повторить элементы программирования на том языке, который изучался школьниками в 10 классе.

Мы не стали приводить в учебнике тот или иной вариант лабораторной работы компьютерного практикума, ориентированной на такое повторение. Каждый учитель, исходя из уровня подготовленности класса, без труда сам подберет соответствующие задания. Тем не менее ниже мы приводим примеры заданий, которые помогут учителю представить, каким должен быть уровень подготовки учащихся, чтобы у них не возникали проблемы при выполнении заданий компьютерного практикума, предусмотренного данным учебником. Кроме того, задания по программированию фактически совпадают с заданиями, предлагавшимися на ЕГЭ-2008 в части С. Отметим только, что на ЕГЭ эти задания выполнялись учащимися в безмашинном варианте; мы же предлагаем, чтобы учащиеся выполнили их и отладили получившиеся программы на компьютере. Это позволит им лучше вникнуть в алгоритмическую суть предлагаемых задач, организовать эффективный поиск ошибок алгоритмизации и обратить внимание на то, где ими обычно допускаются синтаксические ошибки. На последнем моменте остановимся особо. Дело в том, что те учащиеся, которые берутся за выполнение заданий С4 (здесь оно представлено заданием 4), обычно имеют достаточный опыт реального программирования и привыкли к тому, что транслятор программного кода автоматически указывает им на ошибки синтаксиса. Поэтому, когда они пишут программу в безмашинном варианте, то не обращают на такие ошибки должного внимания. В итоге при проверке этого задания они теряют по крайней мере один первичный балл (а если таких ошибок больше двух, то и два балла). Вот почему мы рекомендуем ставить перед учащимися задачу стараться писать программу без синтаксических ошибок, сохранить созданный ими текст программы, а затем после выявления ошибок при трансляции все эти ошибки отметить в исходном тексте и проанализировать их.

Приведем теперь формулировки предлагаемых заданий и комментарии к их выполнению.

Задание 1. В таблице 1.1 приведены результаты выполнения учащимися заданий с кратким ответом (часть В), предлагавшихся на ЕГЭ по информатике в одном из регионов Российской Федерации [1].

[1] Это реальные данные за 2008 г., о чем можно сообщить учащимся. Возможно, их заинтересует, как формулировались задания этой группы. Мы рекомендуем поощрять (и даже возбуждать) такой интерес. Демонстрационные версии ЕГЭ за разные годы можно найти на сайте ФИПИ (Федеральный институт педагогических измерений).

Таблица 1.1

Задание	Выполнение задания по вариантам, %		
	1	2	3
B1	43,66	52,74	50,34
B2	37,32	39,04	29,53
B3	63,38	60,27	63,09
B4	38,73	46,58	48,32
B5	44,37	54,11	55,70
B6	47,89	58,22	62,42
B7	51,41	42,47	79,87
B8	51,41	62,33	64,43
Среднее по группе B			

а) Внесите предоставленные данные в электронную таблицу и вычислите значения, которые нужно записать в незаполненные клетки таблицы 1.1. (Можно принять, что количество учащихся, писавших каждый вариант, одинаково.)

б) Можно ли на основании полученных данных считать, что в целом варианты имеют приблизительно одинаковую трудность?

в) Постройте столбчатую диаграмму, показывающую для каждого задания процент его выполнения в среднем.

г) Задание считается имеющим базовый уровень трудности, если его выполнило более 60% учащихся, повышенный уровень трудности, если его выполнило от 40 до 60% учащихся, и высокий уровень трудности, если с ним справилось менее 40% учащихся. Используя электронную таблицу (а не вручную!), определите, сколько заданий в каждом варианте имело тот или иной уровень трудности.

Выполнение этого задания будет существенно ускорено, если заранее подготовить файл с листом электронной таблицы, куда уже будут внесены эти данные.

Представленное задание носит чисто учебный характер — каждый его пункт нетрудно выполнить вручную или с помощью калькулятора. В реальных задачах обработки данных ЕГЭ приходится подобные вопросы решать для тысяч исходных данных (ведь не появляются же проценты выполнения сами по себе, они возникают из персональной информации о сдаче экзамена каждым участником). Однако предлагать учащимся вручную заполнить таблицу даже на сотню строк (не говоря уже о тысячах) — неразумная трата времени и сил. Если у учителя имеется такая заполненная электронная таблица, то вполне уместно организовать выполнение этого задания для нее.

Выполняя пункт *а*, учащиеся должны вспомнить правила записи формул в ячейки электронной таблицы, в том числе абсолютную и относительную адресации, поскольку надо потребовать, чтобы при заполнении формульных ячеек использовался режим копирования.

Пункт *б* заставляет учащихся осмыслить ту информацию, которую они получили, выполнив пункт *а*. Результат такого осмысления будет полезно использовать при освоении материала § 5.

Пункт *в* — повторение методов построения диаграмм с помощью соответствующих инструментов Excel.

Выполнить задание пункта *г* можно по-разному. Наиболее простой способ — использовать функцию СЧЁТЕСЛИ.

Возможное заполнение электронной таблицы частично приведено в таблице 1.2.

Таблица 1.2

	A	B	C	D	E
1	Задание	1	2	3	В среднем
2	B1	43,66	52,74	50,34	СРЗНАЧ(B2:D9)
3	B2	37,32	39,04	29,53	...
4	B3	63,38	60,27	63,09	...
5	B4	38,73	46,58	48,32	...
6	B5	44,37	54,11	55,70	...
7	B6	47,89	58,22	62,42	...
8	B7	51,41	42,47	79,87	...
9	B8	51,41	62,33	64,43	...
10	Среднее значение	СРЗНАЧ(B2:B9)	...	...	...
11	Базовый уровень	СЧЁТЕСЛИ(B2:B9; “>60”)	...	...	
12	Повышенный уровень	8-B11-B13	...	...	
13	Высокий уровень	СЧЁТЕСЛИ(B2:B9; “<40”)	...	...	

Задание 2. В этом задании вам предлагается сравнить поведение нескольких функций. Формулы, задающие функции, будут не очень похожи одна на другую, а вот графики... Впрочем, вы все увидите сами.

а) В столбец A введите последовательность значений аргумента с шагом 0,1 от –3 и до 3.

б) В столбце B будут вычисляться значения функции $y = 1 - \frac{x^2}{2}$. Продумайте, какие формулы должны быть записаны в клетках этого столбца. Введите эти формулы.

в) В столбце C будут вычисляться значения функции $y = 1 - \frac{x^2}{2} + \frac{x^4}{24}$. Запишите нужные формулы в клетках этого столбца.

г) В столбце D будут вычисляться значения функции $y = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720}$. Введите соответствующие формулы в таблицу.

д) В столбце E будут вычисляться значения функции $y = \cos x$. Продумайте, какие формулы должны быть записаны в клетках этого столбца. Введите их.

е) Запустите *Мастер диаграмм* и выберите режим *Точечная.* Постройте графики первых трех функций на отдельных диаграммах. На каждой из этих диаграмм постройте еще график функции $y = \cos x$.

ж) Видно, что график третьей функции довольно долго ведет себя так же, как график функции $y = \cos x$. Можно сказать, что функция $y = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720}$ хорошо приближает функцию $y = \cos x$. Найдите на отрезке от –1,5 до 1,5 величину максимального отклонения одной функции от другой. С какой точностью функция $y = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720}$ является приближением функции $y = \cos x$?

з) Функция $y = 1 - \frac{x^2}{2} + \frac{x^4}{24}$ дает менее хорошее приближение. Естественно предположить, что если увеличить число слагаемых, то приближение станет еще лучше. Например, можно рассмотреть функцию $y = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{m}$ с подходящим значением m (подумайте, почему не стоит брать седьмую степень числа x). Попытайтесь с помощью электронной таблицы подобрать значение m так, чтобы на отрезке от –1,5 до 1,5 отклонение составило не более 0,0001.

Все шаги выполнения этого задания расписаны достаточно подробно и вряд ли нуждаются в дополнительных комментариях. Скажем только несколько слов о пункте *з*. Каждая формула, приведенная в этом задании и дающая приближение косинуса, является частичной суммой ряда, сходящегося к этой функции. Школьники, возможно, этого не знают, но могут заметить закономерность, заключающуюся в том, что знаменатель в каждом из слагаемых является факториалом показателя степени. Это наводит на мысль, что m можно взять равным 8!, т. е. 40 320. Действительно, такое значение m годится для получения требуемой точности приближения. Но можно для нахождения подходящего m воспользоваться существующей в Excel надстройкой *Поиск решения* [1]. Результат в этом случае будет отличаться от 40 320, но тем не менее окажется к нему весьма близким. Разумеется, правильным является любое значение m, которое гарантирует требуемое приближение.

Ответ на вопрос о том, почему не стоит брать седьмую степень числа x, требует от учащихся определенной математической культуры. Им надо вспомнить, что функция $\cos x$ является четной, т. е. $\cos(-x) = \cos x$ при любом значении переменной x. Поэтому и для многочлена, дающего приближение этой функции, естественно потребовать выполнения этого свойства. Но тогда ясно, что в таком многочлене могут присутствовать только четные степени переменной.

Задание 3. Дан одномерный массив из 30 целочисленных элементов. Напишите программу подсчета произведения положительных элементов этого массива, проверяя, что в нем есть хотя бы один такой элемент.

Вызывает естественное недоумение отсутствие в тексте задания указаний на то, что должна делать программа, если положительных элементов в массиве не окажется. Тем не менее формулировка этого задания в ЕГЭ была именно такой. Кроме того, в инструкции для экспертов по проверке части С прямо указано, что выполнение оценивается нулем баллов, если составленная программа заканчивает свое выполнение без выдачи результата [2]. Мы предлагаем обсудить эту ситуацию с учащимися; оптимальный, на наш взгляд, выход состоит в том, чтобы в случае отсутствия в массиве положительных элементов программа выдавала сообщение об этом. Приводя это задание, мы преследуем простую цель: должны же школьники быть подготовлен-

[1] Как подключить такую надстройку и как ею пользоваться, описано в лабораторной работе № 1 (с. 264 учебника).

[2] Дословно в инструкции сказано, что задача оценивается в 0 баллов, если результат работы программы не получен: не сохранен в виде переменной, либо не выведен на печать (экран), либо не возвращен при помощи команды return.

ными к тем неожиданностям, которые подстерегают их на государственном экзамене.

Для сведения учителя мы приведем программы, которые согласно инструкции экспертов считаются оптимальными и правильными, на двух языках — Паскале и Бейсике.

<table>
<tr><th>Язык Паскаль</th><th>Язык Бейсик</th></tr>
<tr><td>

```
Const N = 30;
Var a:array [1..N] of integer;
  PPos, NumPos, I: integer;
begin
  PPos := 1;
  NumPos := 0;
  for I := 1 to N do
    if a[I] > 0 then
      begin
        PPos := PPos * a[I];
        NumPos := NumPos + 1;
      end;
  if NumPos > 0 then
  writeln (PPos);
end.
```

</td><td>

```
N = 30
DIM I, PPos, NumPos, A(N) AS
INTEGER
PPos = 1
NumPos = 0
FOR I = 1 TO N
  IF A(I) > 0 THEN
    PPos = PPos * A(I)
    NumPos = NumPos + 1
  ENDIF
NEXT I
IF NumPos > 0 THEN
  PRINT PPos
ENDIF
END
```

</td></tr>
</table>

Если следовать инструкции буквально, то за любое из этих решений следует поставить 0 баллов, поскольку в случае отсутствия положительных элементов в массиве ни та ни другая программа не выводит результата, а в переменной сохраняет значение 1, не имеющее никакого отношения к несуществующему произведению. Утверждение об оптимальности тоже вызывает сомнение — подсчет количества положительных элементов вовсе не нужен для того, чтобы знать, имеется ли в массиве хотя бы один положительный элемент; для этого достаточно иметь сигнальную переменную, например, булевого типа (хорошо известно, что арифметические операции менее эффективны, чем операция присваивания). Но можно вообще обойтись без сигнальной переменной, используя, например, следующий алгоритм:

<u>Алгоритм</u>
<u>цел:</u> *PPos*, *I*, *A*[1:30];
{ *PPos* := –1;
   <u>Делать от</u> *I* := 1 <u>до</u> 30
   { **<u>Если</u>** *A* [*I*] > 0 **<u>то</u>** { *PPos* := ABS(*PPos*)**A*[*I*]; }
   }
   <u>Если</u> *PPos* > 0 **<u>то</u>** { **<u>Сообщить</u>** *PPos*;}
   <u>иначе</u> {**<u>Сообщить</u>** "Положительных элементов нет"; }
}

Задание 4. На вход программе подается последовательность символов, среди которых встречаются и цифры. Ввод символов заканчивается точкой (в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка). Требуется написать программу, которая составит из цифр, встречающихся во входных данных, максимальное число. При составлении итогового числа каждая цифра может быть использована только один раз. Если во входных данных цифры не встречаются, то следует вывести –1.

Например, пусть на вход подаются следующие символы: 14ф73п439.

В данном случае программа должна вывести: 97431.

Мы снова приведем программы, которые согласно инструкции экспертов, проверявших часть С, считаются оптимальными и правильными, на двух языках — Паскале и Бейсике.

<table>
<tr><th>Язык Паскаль</th><th>Язык Бейсик</th></tr>
<tr><td>

```
var a: array[0..9] of boolean;
  c: char;
  i, k: integer;
begin
  for i := 0 to 9 do
    a[i] := false;
  read(c);
  while c<>'.' do
    begin
      if c in['0'..'9'] then
        begin
          k := ord(c) – ord('0');
          a[k] := true
        end;
      read(c);
    end;
  k := 0;
  for i := 9 to 0 do
    if a[i] then
      begin
        k := k + 1;
        write(i)
      end;
    if k = 0 then write(–1);
  writeln
end.
```

</td><td>

```
DIM i, j, k, a(10) AS INTEGER
FOR i = 1 TO 10
  a(i) = 0
NEXT
INPUT c$
DO WHILE NOT (c$ = ".")
  j = ASC(c$)
  j = j – ASC("0") + 1
  if j > 0 and j <= 10 then a(j) = a(j) + 1
  INPUT c$
LOOP
k = 0
FOR i = 10 TO 1 STEP – 1
  IF a(i) > 0 THEN
    k = k + 1
    PRINT CHR$(i + 48 – 1);
  ENDIF
NEXT
IF k = 0 THEN PRINT – 1
END
```

</td></tr>
</table>

Мы воздержимся от комментирования этих программ, хотя представленный в них алгоритм, как правило, существенно отличается от тех, тоже правильных алгоритмов, которые обычно предлагаются учащимися. Поэтому проверку здесь полезно проводить с использованием заранее подготовленных тестов. Важно, чтобы система тестов охватывала все «критичные» ситуации.

Тема 2. Технологии работы с информацией

Слова «информационные технологии» прочно вошли в нашу жизнь. Особенности естественного языка таковы, что создается иллюзия, будто бы это технологии обработки информации. Каждый, конечно, понимает, что информацию обрабатывает человек в своей голове, и никакой технологии здесь нет, а информационные технологии — это инструменты создания и преобразования информационных объектов, в которых эта самая информация фиксируется. Реальные технологии работы с информацией — это те методы и приемы, которые помогают человеку преобразовывать собственно информацию. Речь о некоторых таких методах идет в § 4 и 5.

В § 4 осуществляются первые конкретные шаги, направленные на овладение учащимися информационной грамотностью. Обратившись к развернутому определению информационной грамотности, приведенному в учебнике на с. 9, нетрудно увидеть, что работа с информацией представляет собой свертку девяти указанных там умений в три этапа:

- стартовый этап (сюда входят осознание информационной потребности, формулировка запроса и поиск нужной информации) [1];
- этап осознания полученной информации (сюда входят оценка качества информации, формирование собственного отношения и представление своей точки зрения на эту информацию);
- этап рефлексии (оценка эффективности проделанной работы, перенос полученных знаний на другие задачи и другие сферы деятельности, осознание влияния полученных знаний на собственный внутренний мир).

Сказанное означает, что информационная грамотность — это *не предмет изучения*, а *освоение процесса* и *осознание смысла* работы с информацией. Этот тезис означает, в свою очередь, что методика преподавания здесь принципиально отличается от традиционной, когда учащиеся по рассказу учителя или тексту из учебника знакомятся с материалом,

[1] Стартовый этап по-другому можно было бы назвать поисковым, но термин «поисковый» используется настолько часто в самых разных контекстах, что мы воздержались от этого и употребили нейтральное прилагательное «стартовый».

в котором представлены некие теоретические сведения, правила и алгоритмы деятельности, а затем воспроизводят их на контрольных вопросах и заданиях. Само содержание § 4 может у учителя, привыкшего к такому достаточно традиционному методу обучения, вызвать недоумение: что здесь должны выучить школьники? Ответ простой и парадоксальный: ничего. Но они должны научиться выполнять работу с информацией, соответствующую второму этапу. А для этого начинать следует с того, чтобы предложить им такую работу.

Такое занятие можно построить по-разному. Мы приведем два возможных варианта только для того, чтобы стимулировать творческую фантазию учителя.

Вариант 1. В ходе предыдущих занятий учащиеся уже имели дело с обработкой информации, готовя, например, тот или иной материал по социальным эффектам информатизации. Эти материалы могут быть очень краткими (скажем, примеры положительного и отрицательного влияния информатизации на сферу управления) или более развернутыми (в виде сообщения на какую-то тему). Важно сразу настроить учащихся на то, что речь не идет об обсуждении или особенно оценке автора рассматриваемого материала (возможно, найдутся добровольцы, согласные на то, чтобы именно их материал послужил основой для обучения).

Предоставим слово учителю.

Учитель. Вот материал, представленный Андреем Ивановым. Давайте обсудим его, опираясь на формулировку умений, составляющих основу информационной грамотности. (Здесь можно предложить открыть учебник на с. 9 или обратиться к плакату, на котором эти умения перечислены.) Перед нами уже результат поиска информации, так что о первых трех умениях мы судить не можем. А вот давайте попробуем понять, как можно оценить эту информацию. Начнем с такого свойства, как достоверность. Что, на ваш взгляд, позволяет судить о достоверности?

С этого свойства удобно начать, поскольку в § 2 было сформулировано не только его определение, но и правило подтверждения несколькими источниками. Тем самым у учащихся сразу возникает вопрос об источнике представленного материала: это позиция, высказанная в каком-либо энциклопедическом издании или, например, в полемической статье. Доверие к информации тем выше, чем более мы уверены в квалификации и осведомленности того, кто эту информацию предоставил.

Слово учителю.

Учитель. Можем ли мы на основании представленной информации судить об ее источниках?

Спектр ответов здесь широк. Может оказаться так, что в сообщении ученика нет никаких указаний на информационный источник. Поскольку автор присутствует, то вопрос можно адресовать ему. И ученик начинает понимать, что ссылка на источник — это не вопрос, откуда списал, а проблема оценки достоверности. И здесь можно подчеркнуть: никто и не предполагал, что учащийся, не опираясь на источники информации, может достоверно разобраться в данном вопросе.

Другая крайность: ученик исчерпывающе указывает информационный источник; тогда обсуждается степень доверия к этому источнику.

Второе свойство — объективность. Она во многом определяется целями и задачами, которые преследовали авторы данной информации. Опять-таки фундаментальные источники информации (энциклопедии, словари) заинтересованы в максимальной объективности, а если сообщение преследует популяризаторские цели или мотивирует на развертывание полемики вокруг того или иного вопроса, то вряд ли можно здесь ожидать объективности. Чрезвычайно важно в информации различать факт и мнение о нем. Особенно часто проблема такого различения возникает, когда задается вопрос «Что случилось?»; в ответ обычно дается собственная интерпретация произошедшего события. Именно поэтому в ситуациях конфликта (и даже просто расхождения взглядов) необходимо выслушать обе стороны. Правило простое, но, увы, следовать ему не приучены не только наши дети, но и многие взрослые.

Ответы на вопросы о достоверности и объективности уже настраивают на формирование собственного отношения к информации. Важно, чтобы учащиеся это осознали и продолжили этот процесс.

И опять слово учителю.

Учитель. Наше обсуждение таких свойств, как достоверность и объективность информации, конечно, создает у каждого из нас определенное мнение об этой информации. Но ведь для оценки информации важен и ее смысл. В чем он? Давайте выявление смысла начнем с попытки ответить на совсем простой вопрос: что или кто является главным «действующим лицом» в представленной информации? (Конечно, вопрос здесь может быть сформулирован иначе — он должен быть более конкретным по отношению к обсуждаемому сообщению. Важно только, чтобы он был в первой группе вопросов таксономии Блума.)

В дальнейшем формулируются вопросы из других групп таксономии Блума, с тем чтобы полностью был выявлен смысл данного сообщения. Нам представляется полезным, чтобы на этом этапе обучения учащиеся фиксировали в своих тетрадях как вопросы, так и ответы на них. В последующем им можно предложить сопоставить свои вопросы с группами, указанными в таблице 1.1 из учебника, и подвести итог о роли этих вопросов в выявлении смысла.

Наконец, третий шаг — создание собственного смысла. Здесь мы воздержимся от каких-либо рекомендаций, полностью полагаясь на мастерство учителя.

Вариант 2. Учитель предварительно подбирает 2—3 информационных сообщения, например из периодических изданий или каких-либо других источников. Важно, чтобы они различались по уровню обоснованности (для упражнений в определении степени достоверности и объективности). Кроме того, информация, содержащаяся в сообщениях, должна быть понятной учащимся и вызывать интерес, но в то же время должна быть не слишком примитивной — в противном случае будет трудно обсуждать шаги, связанные с извлечением смысла и созданием собственных смыслов. При этом вовсе необязательно, чтобы сообщения были тематически связанными.

В целом план работы для варианта 2 такой же, как и для варианта 1. Сначала ставится вопрос о достоверности и объективности информации, содержащейся в сообщениях, и обсуждается, как в одном случае достигается бóльшая достоверность (объективность), чем в другом. Основной методический прием здесь состоит в том, что учащимся предлагается сравнить, чем с точки зрения достоверности (объективности) различаются предъявленные им сообщения. Чтобы провести такое сравнение, учащиеся должны сами выработать критерии (в варианте 1, как мы видели, эти критерии им предоставляются учителем). Затем в таком же сравнительном ключе с использованием, как и в варианте 1, вопросов из разных классов таксономии Блума обсуждается процесс извлечения смысла: для какого из сообщений это получается легче и почему. Наконец, заключительный шаг — создание собственного смысла. И здесь снова сравнение того, как осуществляется этот шаг для разных сообщений. Нет прямой зависимости между тем, легко ли проходил процесс извлечения смысла, и тем, как создавался собственный смысл. Если обсуждается более двух сообщений, то нетрудно подобрать сообщения так, чтобы это продемонстрировать.

Мы не приводим здесь примеры каких-либо сообщений, которые можно было бы впрямую использовать для реализации варианта 2, поскольку нам представляется полезным, чтобы эти сообщения были привязаны к тому профилю, в рамках которого происходит обучение. Да и вряд ли подбор таких сообщений может представить трудность для учителя. В качестве образцов подобных сообщений можно взять тексты из задания 4 к § 9.

Еще раз повторим: учитель вправе сам выстраивать методику освоения данного материала в зависимости от уровня возможностей класса, от его профиля, от его собственных вкусов, наконец. Необходимо только, чтобы это не было установкой на простое заучивание.

Продолжая обсуждение понятий, рассматриваемых в § 4, отметим, что создание собственного смысла — это некое домысливание, экстраполяция информации за рамки того, что выразил автор в сообщении. Иногда это называют чтением между строк, иногда — вольной интерпретацией. Нередко реакцией автора на такое домысливание бывает «Вы меня не так поняли», да еще с обидой. А на самом деле он должен быть благодарен — ведь ему помогли понять, что свою мысль он выразил недостаточно точно. Воспитание такого благодарного отношения — тоже важный аспект в привитии информационной культуры.

Создание собственного смысла — это всегда некоторый вывод, который делает человек, пытающийся осмыслить предоставленную ему информацию. Иногда такой вывод может противоречить общепринятой точке зрения. Это далеко не всегда означает, что осмысление информации проведено плохо или неправильно, является продуктом некоторого недомыслия. Во-первых, вовсе не исключено, что имеется несколько решений, во-вторых, сама информация может содержать разные, порой противоречивые взгляды на событие, из которых ученик выбрал наиболее близкий ему в качестве основного, в-третьих, информация может быть неполной, недостоверной или/и необъективной. Роль учителя не в том, чтобы вынести и объявить свой вердикт, а в том, чтобы должным образом направить размышления ученика так, чтобы он увидел, в чем были просчеты в выполнении предыдущего шага осмысления информации — в извлечении смысла и оценке свойств. Но даже если неверный вывод сделан из-за нарушения в логике рассуждений, предпочтительнее не впрямую указывать на допущенные ошибки, а предложить подумать над вопросами, которые помогут ученику обнаружить свои просчеты в логике.

Освоение умений осмысления информации нередко называют развитием критического мышления. Мы не видим особой необходимости обсуждать, насколько этот термин удачен, — он рожден в профессиональной сфере и вполне отражает свое содержание[1]. Однако нам представляется, что употреблять его в широкой практике (в частности, в общении со школьниками) не следует. По своей «одежке» он нередко воспринимается как призыв критиковать всё и вся. В глазах учащихся, и без того склонных в этом возрасте к нигилизму и ниспровержению авторитетов, он является знаменем, и критика для них становится, к сожалению, целью, а не средством, служащим выявлению истины. Мы делаем это предостережение, поскольку намерены указать ниже ряд полезных и содержательных источников по методике развития критического мышления у школьников — они окажутся подспорьем тем учителям, которые захотят уделить данной теме большее внимание. Вот эти источники:

1. Загашев И. О. Учим детей мыслить критически / И. О. Загашев, С. И. Заир-Бек, И. В. Муштавинская. — 2-е изд. — СПб.: Скифия; Альянс «Дельта»; Речь, 2003.

2. Загашев И. О. Критическое мышление: технология развития: Перспективы для высшего образования / И. О. Загашев, С. И. Заир-Бек. — СПб.: Скифия, 2003.

3. Заир-Бек С. И. Развитие критического мышления на уроке / С. И. Заир-Бек, И. В. Муштавинская. — М.: Просвещение, 2004.

4. Красновская В. А. Технология развития критического мышления: [Эл. ресурс]: http://ipk.minsk.edu.by

5. Мяэотс О. Н. Уроки информационной грамотности в школе: метод. рекомендации / О. Н. Мяэотс. — М.: Чистые пруды, 2005. — (Библиотечка «Первого сентября», серия «Библиотека в школе». Вып. 2).

Одно занятие, разумеется, не может привить учащимся обсуждаемые умения и уж тем более превратить их в навык. Поэтому важно, чтобы они были в активе по ходу любой информационной деятельности учащихся. Сейчас школьники в процессе своей учебы в школе готовят немало сообщений, рефератов, презентаций на самые разные темы; мы настоятельно рекомендуем учителю (вообще говоря, любого предмета, а не только информатики) настраивать учащихся на то, чтобы такая подготовка велась с использованием указанных методов обработки информации.

[1] И. О. Загашев, работы которого указаны ниже, приводит следующее определение: критическое мышление — это система мыслительных стратегий и коммуникационных качеств, позволяющих эффективно взаимодействовать с информационной реальностью.

Задания к § 4 направлены на закрепление умений, приобретенных учащимися в ходе учебной работы с информацией. Вряд ли вызовут затруднения задания 1 и 3; что касается задания 2, то ответы в пункте *а* нам также представляются почти очевидными. Суждение 2 относится к первому шагу осмысления информации — это чистое восприятие; суждение 1 могло возникнуть как результат извлечения смысла; суждение 3 — это, без сомнения, создание собственного смысла, ибо оно могло родиться только из сопоставления этих строк с собственными знаниями истории войны 1812 г., ни о каком народе, мирных жителях и т. п. в этих строках М. Ю. Лермонтова нет даже намека.

В задании 2б распределение процесса осмысления информации по шагам уже может показаться не таким очевидным. Впрочем, достаточно ясно, что суждение 1 относится к первому шагу, поскольку в нем практически пересказывается афоризм В. Шкловского. Суждения 2 и 4 следует отнести, скорее всего, ко второму шагу — они фактически говорят об одном и том же, выражая мысль единства, т. е. «собирания» людей вокруг общих интересов, отраженных в книжных коллекциях. Суждение 4, которое может возникнуть как результат осмысления суждений 2 и 4 (в только что рассмотренном ключе), относится к третьему шагу, ибо такой смысл в высказывании В. Шкловского никак не звучит впрямую.

Среди компонентов второго этапа присутствует представление своей точки зрения на информацию, которая была осмыслена. И важно, что речь здесь идет о представлении в первую очередь для себя. К сожалению, практика показывает, что большинство учащихся, считающих, что они хорошо поняли предоставленную им информацию, не могут внятно изложить это свое понимание. И хотя автор слов «Кто ясно мыслит, тот ясно излагает» сегодня не популярен, сами эти слова вовсе не утратили своей актуальности. В таком изложении должно быть отчетливо видно, что́, по мнению учащегося, является в информации главным, каковы связи между разными аспектами, каков вывод. Как правило, исходная информация в таком изложении принимает сжатую форму, и потому этот процесс называют свертыванием информации. И в учебнике при определении понятия «свертывание информации» акцент сделан именно на этом аспекте — на представлении информации в сжатом виде с обязательным сохранением основного смысла исходной информации. Результатом такого сжатия может быть снова текст (в учебнике мы употребили образный оборот «текст-экстракт») или какая-либо графическая форма: таблица, график, схема и т. п.

Текстовые формы свертывания информации весьма разнообразны. Это может быть резюме, конспект, реферат, тезисы, эссе, дайджест и т. д. Информатика — не тот предмет, да и нет достаточного времени, чтобы всерьез осваивать указанные формы свертывания информации. Однако в классах гуманитарного профиля можно предложить учащимся самостоятельно подготовить обзор по этим формам. Нам представляется, что здесь был бы весьма полезен контакт с преподавателем литературы или работником школьной библиотеки. Достаточно подробно текстовые формы свертывания информации рассмотрены в пособии для учащихся «Человек и информация», которое упомянуто в списке литературы для дополнительного чтения, приведенном в конце учебника.

Графические формы свертывания информации обычно более востребованы в курсе информатики. С многими из них учащиеся знакомы по темам, в которых рассматривались вопросы моделирования. Да и в учебном процессе им, конечно, неоднократно приходилось иметь дело с различными схемами, диаграммами, графиками, таблицами. Можно, к примеру, напомнить, что, выполняя задание 1, сформулированное на с. 24, они сначала имели дело с информацией, свернутой в таблицу, а затем преобразовывали ее, строя с помощью Excel несколько диаграмм.

В учебнике основное внимание уделено построению схем. Полезно обратить внимание учащихся, что можно выделить два больших класса: структурные схемы и схемы, отражающие процесс. Один вид структурных схем — кластер — в учебнике представлен достаточно подробно. Схемы взаимодействия обычно отображают «партнерские» связи равнозначных по уровню элементов. Среди схем взаимодействия, которые уже рассматривались учащимися ранее, это схема взаимодействия различных устройств компьютера, схема информационных потоков в компьютере, схемы алгоритмов, схемы управления (в том числе с обратной связью) и т. д. Диаграммы и графики в отличие от схем показывают изменения одних величин относительно других или количественные соотношения между величинами. Надо только иметь в виду, что слово «график» полисемично — ведь говорят «график дежурств по школе» или «график движения автобусов», хотя информация в этом случае обычно представлена не графиком, а таблицей.

Учащиеся уже в средней школе обычно знают, что диаграммы бывают линейчатые (столбчатые) и круговые. В нашем учебнике базового курса для 8—9 классов специально уделялось внимание тому, в каких случаях удобнее использовать круговые диаграммы, а в каких — столбчатые. Учащимся можно предложить самим сформулировать соответствующий критерий. Звучать он может примерно так: столбчатые диаграммы предпочтительнее, когда изображаются данные, изменяющиеся как функция независимого параметра, например времени или номера варианта (как в упоминавшемся задании 1), а круговые — когда изображаются взаимозависимые данные, например компонентный состав вещества или доля отдельных материков на поверхности Земли.

Нет явных рецептов, какую форму свертывания информации предпочтительнее выбрать. Это зависит от многих факторов, в которых среди основных надо назвать цели представления информации. Поэтому все сказанное выше может служить не более чем советом.

Основное внимание в § 5 уделено методам построения схемы в виде кластера. Прием, представленный на рисунке 1.2 учебника, является весьма эффективным, и мы советуем обратить на него внимание учащихся.

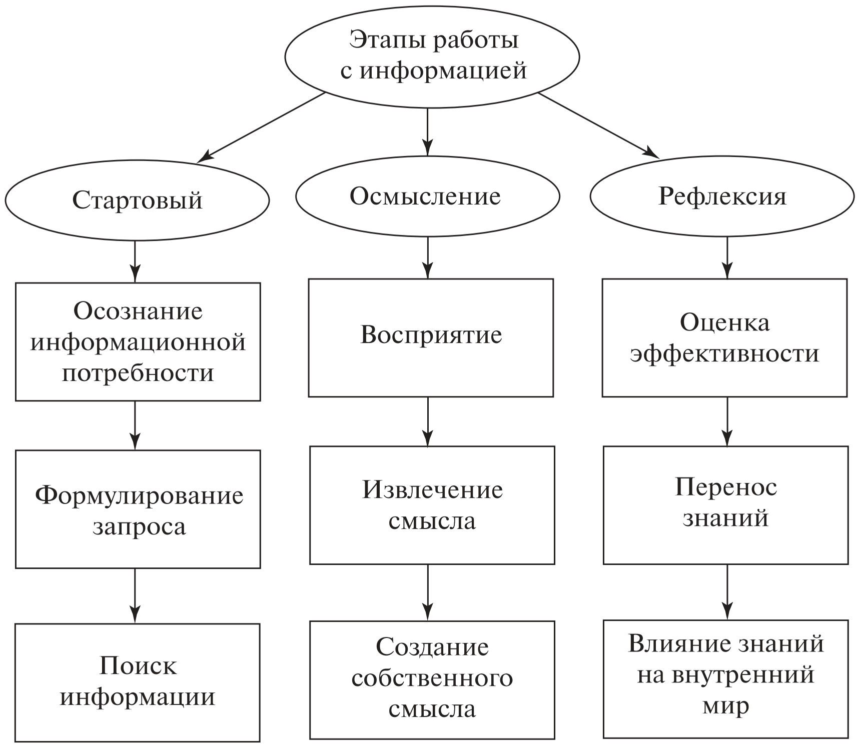


Рис. 1.1

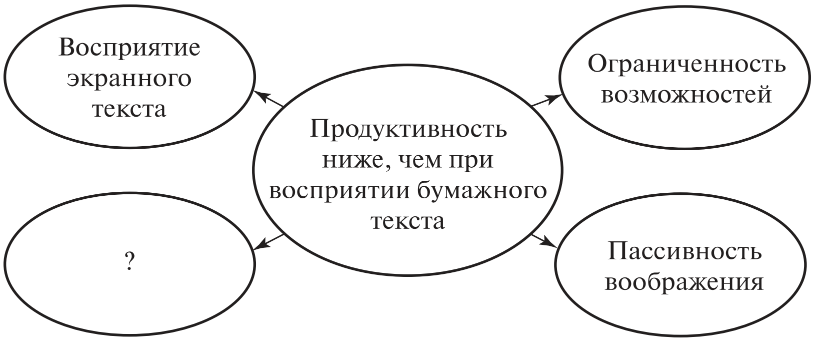


Рис. 1.2

Любой человек в своей работе с информацией должен уметь не только сворачивать ее в ту или иную форму, но и выполнять обратное преобразование — извлекать информацию, поданную в свернутом виде, т. е. уметь работать со схемами, таблицами, графиками, диаграммами, картами. На этих вопросах мы остановимся, обсуждая материал § 9.

Задания к § 5, как и в случае § 4, не могут претендовать на создание и прочное усвоение навыков свертывания информации. Мы и в этом случае рассматриваем представленный материал как средство побуждения учащихся к освоению представленных технологий.

Задание 1 не имеет какого-либо канонического ответа. Учащиеся могут называть разные цели, и чем больше разумных целей будет названо, тем лучше. Задания 2—5 позволяют учащимся повторить основные положения и понятия, связанные со свертыванием информации.

Задание 6 имеет принципиальное значение. Оно показывает, что свертывание информации может происходить по-разному. В частности, это зависит от выбора ключевых слов, т. е. решения человека о том, что является главным в предоставленной ему информации. Многообразие в осмыслении и в последующей работе с информацией — это важнейшая особенность жизни в информационном обществе. И понимание возможности такого многообразия — краеугольный камень в толерантности человека.

В заданиях 7 и 8 нет однозначного ответа. Для задания 7 схема может, к примеру, выглядеть так, как показано на рисунке 1.1. Может возникнуть вопрос, почему внутри каждого этапа шаги располагаются иерархически, а этапы — стартовый, осмысления и рефлексии — находятся на одном уровне. Объяснить это можно тем, что каждый из этапов можно осуществлять автономно (что продемонстрировано, в частности, тем, что этап осмысления отдельно рассматривается в § 4 и 5). А вот для шагов внутри этапов такая автономность малопродуктивна.

Для задания 8 схема может выглядеть так, как показано на рисунке 1.2. Мы специально не приводим заполнение одного из овалов, чтобы не сковывать инициативу. Впрочем, ответ нам представляется очевидным — компьютерные технологии работы с текстом не заменяют, а лишь дополняют работу с бумажными текстами.

Задание 9 отмечено звездочкой (*) не столько потому, что оно трудное, сколько в силу некоторого отхода его темы от основной канвы материала. Тем не менее нам представляется полезным продемонстрировать учащимся, что такой информационный процесс, как обучение, также подчинен общим закономерностям работы с информацией. Отличие в том, что структурированность этапа осмысления здесь предопределена авторами учебных заданий. Более того, мы считаем, что понимание типа задания (репродуктивное, продуктивное, творческое) для некоторых школьников будет способствовать осознанию того, что требуется от них для наиболее полноценного выполнения задания. В данном случае к репродуктивным мы бы отнесли задания 2, 3, 4 и 5а; к продуктивным — 5б, 6 и 8; к творческим — 1, 7 и 9. Но потому задание 9 и является творческим, что здесь вполне возможны другие, вполне аргументированные точки зрения на отнесение заданий к тому или иному типу.

Один час компьютерного практикума, предусмотренный тематическим планированием для профильного уровня, мы рекомендуем использовать для восстановления навыков работы с текстовым редактором. Учитель здесь может предложить задания, которые ему представляются наиболее подходящими для достижения этой цели.

Ни в этих двух параграфах, ни в других разделах учебника мы практически не касаемся третьего, рефлексивного этапа, когда новый опыт, новое знание встраивается в систему личностных смыслов, становясь для человека своим. На наш взгляд, это не может быть предметом изучения. Тем не менее на два аспекта — оценка эффективности проделанной работы и перенос полученных знаний на другие задачи и другие сферы деятельности — можно и нужно, по нашему мнению, обращать внимание учащихся. Это может быть сделано как подведение итогов (чему же мы научились?) или как обращение к имеющемуся опыту при решении новой задачи (давайте вспомним, не приходилось ли нам делать нечто подобное раньше).

Что касается осознания влияния полученных знаний на собственный внутренний мир (третий компонент рефлексии), то это очень деликатный аспект. Учитель здесь может только крайне осторожно, с большим тактом и ненавязчиво подталкивать учеников к тому, чтобы учащиеся осознали те изменения, которые происходят с ними в результате не только осмысления той или иной информации, но и овладения методами работы с информацией, т. е. в связи с повышением их информационной грамотности.

Тема 3. Моделирование — базовый элемент информационной грамотности

Эта тема в предлагаемом курсе информатики играет двоякую роль. С одной стороны, мы хотим обратить внимание учащихся, что обработка информации человеком, рассматривавшаяся в предыдущей теме, — это фактически рассмотрение процесса моделирования под несколько другим углом зрения. В учебнике мы напоминаем учащимся, что первый шаг в построении модели представляет собой определение того, какая информация является существенной для решения задачи, а это есть не что иное, как определение информационной потребности. Последующие шаги (определение средств реализации, описание факторов посредством параметров, установление связей между параметрами) фактически являются некоторой формой осмысления, в ходе которого происходит извлечение смысла и создание собственного смысла, который как раз и представлен построенной моделью. Более того, параллелизм между этапами построения модели и обработкой информации выпукло иллюстрируется сопоставлением, представленным в учебнике в таблицах 1.2 и 1.3.

С другой стороны, эта тема является повторением ядерной части материала, предлагавшегося в учебнике для 10 класса. Если в 10 классе использовался другой учебник, то этот материал, скорее всего, является для учеников новым, и для них весьма краткое, можно сказать, конспективное изложение вопросов моделирования может оказаться недостаточным по объему и трудным для восприятия. Мы в этой ситуации советуем воспользоваться материалом § 5, 6, 11, 43, 46—48 из учебника для 10 класса.

Если учащиеся, осваивающие информатику по плану базового курса, уже знакомы с информационным моделированием в той степени, которая предусмотрена Федеральным компонентом образовательного стандарта по информатике, то эта тема может быть исключена из изучения. В противном случае ее необходимо рассмотреть хотя бы в минимальном объеме.

Для учащихся, избравших профильный вариант изучения информатики, необходимо полностью восстановить свои знания и навыки по данной теме. С этой целью в § 6 воспроизводятся основные определения и методы построения информационных моделей, перечисляются основные виды моделей. Весьма желательно, чтобы учащиеся могли дать определение для модели каждого вида. Для удобства проверки мы ниже воспроизводим эти определения.

Математической называется информационная модель, в которой параметры и связи между ними выражены в математической форме.

Компьютерной моделью называется информационная модель, в которой параметры и связи между ними выражены комплексом программ, реализованных на компьютере.

Информационные модели, в которых параметрами являются атрибуты объектов (т. е. описание их свойств и характеристик), называются **фактографическими**.

Модель, для которой изменение параметров во времени является существенным фактором, называется **динамической**.

Модель, для которой изменением во времени можно пренебречь, называется **статической**.

Модель, для которой существенным является фактор случайности, называется **стохастической**.

Стохастическая модель, в которой для описания параметров используются вероятности случайных событий, называется **вероятностной**.

Разумеется, что от учащихся не требуется дословное воспроизведение этих определений, но их смысл они должны передавать точно. Кроме того, мы рекомендуем предложить учащимся привести примеры моделей каждого вида.

В учебнике при изложении теоретического материала мы исходили из того, что тема моделирования достаточно хорошо изучена учащимися в 8—10 классах. Мы, к примеру, не стали подробно разъяснять понятия фактора и параметра, хотя различение этих понятий нередко представляет для учащихся определенную трудность. Более того, 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. Например, 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 как компонент погоды будет выступать одним из параметров, характеризующих ее влияние. Само выделение существенных факторов, как правило, производится на основе целевых установок. Описание факторов набором параметров занимает промежуточное по-

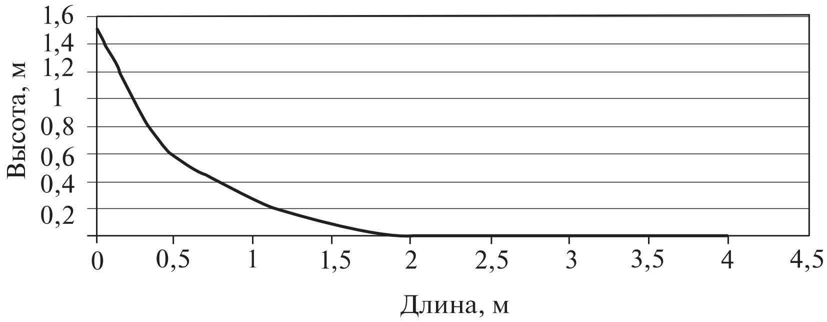


Рис. 1.3

ложение: с одной стороны, имеется некоторый произвол в выборе этих параметров, с другой стороны, должна обеспечиваться определенная полнота. 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 вполне 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 (при необходимости более глубокого повторения данного материала) подобных примеров позволяет им лучше уяснить суть понятий фактора и параметра.

Построение математической модели иллюстрируется решением задачи об оптимальной форме снежной горки. Это классическая задача о траектории наискорейшего спуска, которая в 1696 г. была сформулирована И. Бернулли и в математике положила начало теории вариационного исчисления. Аналитически она была решена еще в XVIII в., и ее решение связано с именами таких великих математиков, как Г. Лейбниц и Л. Эйлер. История решения этой задачи достаточно богата и интересна (известно, к примеру, что это тот редкий случай, когда великий Г. Галилей был обманут своей интуицией — он считал, что такой траекторией является дуга окружности, в то время как на самом деле такой траекторией является циклоида), ей можно посвятить краткое сообщение учащихся.

В соответствии с приведенной выше классификацией эта модель является статической. Существенно новым здесь выступает то обстоятельство, что результатом моделирования статической модели является некоторая линия. В моделях, которые рассматривались в курсе информатики 10 класса, линии появлялись лишь как траектории движения (модель движения тела в среде с сопротивлением, модель броуновского движения) или графики изменения величин в зависимости от времени (модели неограниченного и ограниченного роста биомассы), т. е. в динамических моделях.

Компьютерное исследование построенной модели, описанное в лабораторной работе № 1, показывает, что если на линию спуска не накладывать никаких ограничений, то она обязательно проходит ниже конечной точки. Горку можно строить и с таким профилем, но посреди зимы углубляться в грунт едва ли кому-нибудь захочется. Тем самым возникает задача о траектории наискорейшего спуска с ограничениями [1]. Следуя указаниям, приведенным в тексте лабораторной работы, учащиеся исследуют модель при том ограничении, что в каждой точке высота горки должна быть неотрицательной. На рисунке 1.3 приведена построенная в Excel линия спуска для 14 точек разбиения отрезка [0; 6].

Линии спуска для 11 точек разбиения и 14 точек уже мало отличаются друг от друга, да и время спуска при переходе от 11 к 14 точкам тоже улучшается всего лишь на 0,24 секунды. Поскольку речь не идет о построении трассы для саночных соревнований европейского или мирового уровня, то, по-видимому, на достигнутом результате можно остановиться. Если же учащиеся захотят продолжить компьютерный эксперимент, то следующее значение для числа точек разбиения удобно взять равным 19.

В ходе выполнения этой лабораторной работы важно обсудить с учащимися вопросы адекватности построенной модели и обратить их внимание, что модель, предложенная в объяснительном тексте § 6, оказалась неадекватной. Для воспитания информационного мировоззрения важно подчеркнуть, что неадекватность была выявлена не в ходе натурного эксперимента (т. е. построения горки в действительности), а в результате виртуального эксперимента, проведенного с помощью компьютера.

Разобранная задача допускает самые разнообразные исследовательские продолжения. Дух будет захватывать и в том случае, если, скажем, в середине горки на заданной высоте сделать небольшой трамплин. Как в этом случае будет выглядеть скат до трамплина и после него? Ведь до

[1] О существовании аналитического решения этой задачи нам ничего не известно.

трамплина можно не ставить ограничения на траекторию. Можно вообще перейти к изучению устройства лыжных трамплинов. В общем, эта лабораторная работа может быть развернута в полноценный учебно-исследовательский проект.

Перейдем к обсуждению заданий, помещенных в конце § 6. Ответы на вопросы 1—3 даны в объяснительном тексте параграфа. Задание 4 мы рекомендуем выполнить дома, поскольку в его формулировке в явной форме содержится совет обратиться к соответствующему материалу из учебника для 10 класса. Прямого ответа учащиеся там не найдут, и им придется проанализировать фрагмент § 5, в котором речь идет об адекватности модели. В результате анализа ответ может быть сформулирован так: основными причинами неадекватности являются неправильное или неполное выделение существенных факторов, а также несогласованность с теми научными воззрениями, которые на данном этапе развития науки подтверждены теорией и экспериментом.

Адекватность модели — это, без сомнения, один из центральных моментов 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 некая модель пригодна всегда и при любых обстоятельствах), с другой стороны, направлена 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 движущей силой в развитии научного знания. Поэтому при изучении данной темы особенно уместны яркие примеры из истории науки. Вот задание, в котором обсуждаются некоторые из таких примеров.

Задание 5. а) Как известно, в химии существовала теория флогистона — особого вещества, которое улетучивалось при окислении какого-либо вещества. Какими экспериментами была установлена неадекватность этой модели процесса окисления?

б) Как известно, планетарную модель атома предложил Э. Резерфорд. Каким теоретическим положениям, существовавшим тогда в физике, противоречила эта модель? Почему тем не менее она была признана в то время адекватной? К каким изменениям в физических теориях привело признание адекватности данной модели?

В пункте *а* этого задания имеются в виду опыты Ломоносова и Лавуазье, экспериментально доказавших, что в результате реакции окисления масса вещества увеличивается, а не уменьшается. Эти эксперименты привели к открытию закона сохранения массы при проведении химических реакций, ставшего фундаментом химии и физики.

В пункте *б* планетарная модель была экспериментально подтверждена опытами Э. Резерфорда по бомбардировке атома α-частицами. Однако было неясно, почему электроны движутся по орбитам только вполне определенных радиусов. М. Планк высказал гипотезу, что значение энергии не может быть любым неотрицательным числом, а существует минимальный размер такой энергии, названный квантом, и значение энергии может быть лишь целочисленным кратным этого кванта. Величина кванта настолько мала, что в обычном нашем мире можно пренебречь такой дискретностью множества значений энергии, однако в микромире, т. е. мире элементарных частиц, каждый квант на счету.

Нужно призывать учащихся приводить и другие примеры в истории науки, когда неадекватность существующих моделей обусловливала пересмотр научных концепций. Учащиеся могут, например, вспомнить об опытах Майкельсона, установившего конечность скорости света, что противоречило царствовавшей в то время теории Ньютона, и т. д.

Задание 5 направлено на актуализацию знаний метода дискретизации. Кратко сущность этого метода описана в объяснительном тексте параграфа. Важным является пункт *б* этого задания. Крайне желательно, чтобы учащиеся среди приводимых примеров вспомнили о кодировании цвета (разбиении на градации) и звука (дискретизация по времени и квантование по высоте). Помочь в этом может § 4 из учебника для 10 класса.

Задание 6 нам не представляется трудным, мы рекомендуем выполнить его дома с последующим обсуждением в классе. Цель моделирования — идентификация человека как гражданина страны. Рассматривая паспорт (не только свой, но и родителей, и надо посоветовать им так сделать), учащиеся могут выделить следующие факторы: наименование человека, его возраст, место жительства, семейное положение, внешность. 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 сам представляет собой довольно сложный конгломерат из наименования детей и дат их рождения. Что касается внешности, то это неформализуемый фактор, поэтому он представлен в паспорте фотографией. В обсуждении пункта *в* можно указать на наличие режима смены паспорта при достижении определенного возраста, а также на возможность фиксировать изменения в сведениях о месте проживания, семейном положении и т. п. Из приведенного обсуждения этого задания видно, что оно весьма разнообразно по дидактическим целям: учащиеся должны в очередной раз продемонстрировать понимание различий между понятиями «фактор» и «параметр модели», умение отличать формализуемые факторы от неформализуемых, видеть, в чем проявляется адекватность модели.

Задание 7 не имеет канонического решения, и здесь фантазия учащихся может разыграться не на шутку. Ее надо сдерживать в конструктивных рамках. Как правило, при таком обсуждении учащимися формулируются два фактора — желание друга и собственные финансовые возможности; в крайнем случае, их можно им подсказать. Оба фактора могут быть формализованы: например, первый из них — списком соответствующих пожеланий, второй — указанием допустимой суммы денег. Впрочем, если желание друга выражено в форме «Подарите мне что-нибудь красивое», то вряд ли удастся подобрать подходящий параметр для формализации этого фактора. Красота, как и другие эстетические категории, формализации не поддается.

Что касается задания 8, то практически любая жизненная ситуация содержит в себе ту или иную задачу (и, как правило, не одну). Вы собираетесь на каникулы в туристическую поездку. Как выбрать маршрут? Как организовать поездку, т. е. выбрать вид транспорта, продолжительность поездки и т. п.? А вот другая ситуация: хочется пойти в кино и надо выполнить домашнее задание. Как спланировать свое время? И т. д.

Задание 9 еще раз подчеркивает родственную связь между работой по свертыванию информации и моделированием. Выделение ключевых слов — это фактически формализация того, что следует считать наиболее существенным в представленной информации.

Задание 10 позволяет еще раз акцентировать внимание учащихся на том, что при моделировании востребованными оказываются такие компоненты информационной культуры, как умение организовать поиск и отбор информации, необходимой для решения стоящей задачи, умение оценивать достоверность, полноту, объективность поступающей информации, умение представлять информацию в различных видах.

В задании 11 приведены высказывания известных политиков прошлых веков. Не так важно, кому они принадлежат, поскольку достаточно часто цитируются. Первая из них констатирует неадекватность построенной модели — цель оказалась недостигнутой. Это свидетельствует о том, что при ее построении не были учтены, а может быть, и не выявлены вообще существенные факторы. Вторая фраза означает, что вообще не предпринималось никаких шагов к построению модели, с помощью которой можно было бы прогнозировать развитие ситуации. Едва ли следует устраивать дискуссию, какая из двух ситуаций хуже. Важно другое — оба пункта этого задания имеют прямое отношение к вопросам управления. Тем самым они позволяют стимулировать интерес учащихся к изучению материала очередного параграфа.

Теоретический компонент § 7 в большей своей части направлен на повторение материала, изучавшегося в 10 классе. Поэтому и рекомендации здесь те же, что приведены в книге для учителя к учебнику 10 класса. Новым здесь является, во-первых, ознакомительный рассказ о существующих в настоящее время компьютерных методах, поддерживающих разработку проектов, и средствах управления проектами и, во-вторых, введение и рассмотрение очень важного кибернетического понятия «черный ящик». В объяснении этого понятия мы предлагаем следовать учебнику. Полезно предложить учащимся самим привести примеры использования данного понятия. Но надо иметь в виду, что такое задание может вызвать определенные затруднения. Этому есть, на наш взгляд, два объективных обстоятельства. Во-первых, в школьных дисциплинах, как правило, стремятся вскрыть механизм изучаемого взаимодействия; во-вторых, как инструмент понятие черного ящика используется для описания и анализа очень сложных и, как правило, вероятностных систем. В физике к таким системам относятся в первую очередь газы (в отличие от них многие твердые вещества имеют очень жесткую внутреннюю структуру — кристаллическую решетку). Поэтому, например, изучение поведения газов под воздействием давления и температуры проводится в школьном курсе физики фактически методом черного ящика — без апелляции к молекулярно-кинетической теории, поскольку для ее применения нужны были бы достаточно развитые методы теории вероятностей и математической статистики.

При обсуждении понятия «черный ящик» можно предложить учащимся поразмышлять над вопросом, может ли черный ящик не иметь входов, но иметь выходы. Ответ положительный; самый простой, по-видимому, пример — это обыкновенные часы. Чтобы узнать время, не требуется никаких воздействий и уж тем более не нужно вникать в их устройство, достаточно просто взглянуть на циферблат.

К теме черных ящиков относятся задания 11—13. Формируя набор черных ящиков, мы руководствовались многими мотивами. Прежде всего это — необходимое разнообразие комбинаций входов и выходов. Мы хотим убедить школьников, что число выходов никак не связано с числом входов — довольно распространено ошибочное мнение, что число выходов не превосходит числа входов.

Другой нашей заботой было обеспечение разнообразия типов входных и выходных данных. На входы могут подаваться произвольные последовательности символов, последовательности букв какого-либо алфавита (что позволяет обрабатывать такие последовательности, используя информацию об этом алфавите), целые или натуральные числа. Здесь опять-таки мы стремились реализовать любые комбинации типов входных данных.

Хотим еще раз подчеркнуть, что задача ученика — определить, *что* делает данный черный ящик, и от него вовсе не требуется восстановить алгоритм работы черного ящика — ведь не спрашивается, *как* получается результат. Да и спрашивать во многих случаях неправомочно, например, алгоритмы обработки символьной информации не рассматривались. Кроме того, в принципе нельзя утверждать, что в программе реализован именно тот алгоритм, который «угадан» учащимся, — ведь для такого вывода потребовалось бы тестирование при всех допустимых вариантах исходных данных, перебрать которые за разумное время просто невозможно.

Тем не менее черный ящик являет собой важный пример *формального исполнителя*, управление которым осуществляется программным способом. Поэтому важно еще раз обратить внимание учащихся на ряд свойств, присущих любому формальному исполнителю. Во-первых, нужно разобрать с учащимися, что любая входная информация делится для исполнителя на *воспринимаемую* (можно сказать еще *допустимую*) и невоспринимаемую. На невоспринимаемую информацию формальный исполнитель дает реакцию «Не понимаю». Уже в этом проявляется феномен формальности исполнителя — он и не пытается «додумать», что же хотели ему сообщить. Если перейти на программистскую терминологию, здесь мы имеем ошибку, которая идентифицируется как синтаксическая ошибка.

Во-вторых, даже для допустимых данных алгоритм, заложенный в черный ящик, может не обрабатывать их ввиду и внутренней противоречивости. В программировании такие ошибки обычно иллюстрируются примерами попыток разделить на 0 или извлечь корень из отрицательного числа. В нашем случае подобные ошибки (их обычно называют семантическими) вызывают реакцию исполнителя «Не могу».

Приведем авторские расшифровки черных ящиков (вовсе не исключается, что ученики сформулируют какой-либо иной принцип работы черного ящика; важно лишь то, чтобы все приведенные примеры подходили под предложенный принцип).

Задание 11а. На вход подается произвольная последовательность символов, на выходе — последовательность символов в обратном порядке.

Задание 11б. На вход допускается произвольная последовательность символов русского алфавита, каждый символ заменяется на ему предшествующий в русском алфавите (при этом «а» заменяется на «я»).

Задание 12а. На входы подаются натуральные числа, на выходе — меньшее из этих чисел.

Задание 12б. На входы подаются натуральные числа. Если они одной четности, то на выходе выдается их полусумма; если разной, то меньшее из них. В этом черном ящике спрятана алгоритмическая конструкция ветвления.

Задание 13а. На вход допускается произвольная последовательность символов русского алфавита, подсчитывается количество гласных и согласных в этой последовательности, первое из этих чисел выдается на первый выход, второе — на второй.

Задание 13б. На вход подается натуральное число. Первый выход — наименьший простой делитель этого числа, второй выход — показатель наибольшей степени этого простого числа, которая делит число на входе. Это уже довольно сложная задача. Можно предложить дома написать программу, имитирующую работу этого черного ящика.

Без сомнения, обучающий эффект намного выше, если работа по расшифровке черных ящиков проводится на компьютере. Нами создано специальное педагогическое программное средство «Черные ящики» [1], которое содержит более 100 ящиков различных уровней сложности и различных соотношений входов — выходов.

Из остальных заданий к § 7 в комментарии по выполнению нуждается, на наш взгляд, лишь задание 9: к понятию обратной связи имеют отношения только процессы, описанные в пунктах *б*, *г* и *е* данного задания.

[1] Данный электронный продукт является свободно распространяемым и может быть получен по электронной почте. Для его приобретения можно обращаться по адресу Alexander.Gein@usu.ru

В отличие от § 6, где теоретические положения по моделированию иллюстрировались построением математической модели непосредственно, для понятий, связанных с управлением, рассмотрение задачи на построение математической модели вынесено в отдельный параграф. И связано это не столько с тем, что § 7 и без того весьма насыщен, сколько с тем, что построение модели сопровождается демонстрацией целого арсенала средств, применяемых в реальных исследованиях различных процессов. Прежде всего это использование понятий обратной связи и черного ящика. Именно они позволяют сформулировать задачу в терминах параметров модели. Во-вторых, это понимание необходимости проведения экспериментальных исследований, чтобы изучить и смоделировать поведение черного ящика. Наконец, это методы обработки экспериментальных данных с использованием компьютерных технологий. Что касается последнего аспекта, то напомним, что достаточно подробно вопрос о компьютерной обработке результатов эксперимента обсуждался в § 13 учебника для 10 класса. Мы не стали акцентировать на этом внимание учащихся, по сути повторив соответствующее объяснение в тексте лабораторной работы № 2. Однако при желании учитель может предложить учащимся перечитать указанный параграф.

На наш взгляд, построение модели экономической задачи описано в объяснительном тексте § 8 настолько подробно, что можно поручить учащимся разобраться в нем самостоятельно. А контроль можно осуществить, проверив выполнение задания 2 к этому параграфу.

Вопрос, сформулированный в задании 1, весьма непрост. Использование одной модели для решения различных задач, как правило, возможно в том случае, если после формализации, т. е. описания модели с помощью параметров, мы получили одну и ту же систему зависимостей этих параметров. Предсказать это заранее в большинстве случаев нельзя. Нередко это указывает на какую-то весьма глубокую общность протекания процессов, но природа этой общности может долгое время быть скрытой даже для весьма опытного глаза.

Задание 3 в § 8 относится к весьма важному элементу процесса моделирования — интерпретации результатов. Сам по себе процесс интерпретации, т. е. придание смысла тем или иным сторонам построенной модели или полученным с ее помощью результатам, весьма сложный и тонкий. По существу, он относится к этапу осмысления информации, а как было показано ранее, этап осмысления неформален и весьма непрост. В этой модели можно считать, что коэффициент a показывает уровень благосостояния: чем выше уровень жизни, тем больше людей будет готово заплатить высокую цену, а значит, тем меньше будет коэффициент a. Мы надеемся, что компьютерный эксперимент, предусмотренный лабораторной работой № 2, убедит учащихся в справедливости этого вывода.

Само проведение лабораторной работы № 2 также представляется нам достаточно детализированным. Самым важным, на наш взгляд, здесь является вывод, который по результатам компьютерных экспериментов должны сделать учащиеся: наибольший доход получается тогда, когда предложение товара примерно в полтора раза превышает спрос. Этот достаточно известный закон свободного рынка учащиеся открывают самостоятельно. Уже одно это представляет значительную образовательную ценность. Но важно здесь и то, что учащиеся и в данном вопросе снова имеют дело с процессом интерпретации, т. е. осмысления полученной информации.

Помещенная в конце объяснительного текста § 8 фраза: «У тех, кто дружен с математикой, уже, наверно, руки чешутся исследовать эту функцию математическими методами» — завуалированный призыв к учащимся провести такое исследование аналитическими средствами. Но хотя исследование функции с помощью производной является общеобразовательным элементом математической подготовки (в частности, входит в ЕГЭ по математике, в котором участвуют все выпускники общеобразовательной школы), мы считаем, что предлагать провести исследование функции $D(x)$ нужно только учащимся физико-математического и экономического профилей. Вот как оно проводится. Для функции $D(x)$ вычисляем производную: $D'(x) = (c - ax^2) - \underline{2ax^2} = c - 3ax^2$. Корнями производной являются числа $\pm\sqrt{\frac{c}{3a}}$. Нетрудно проверить, что точкой максимума является число $x_0 = \sqrt{\frac{c}{3a}}$. Это та цена, при которой доход будет наибольшим. А количество проданных билетов по этой цене равно $f(x_0) = c - a(x_0)^2 = c - \frac{\tilde{n}}{3} = \frac{2}{3}c$. Тем самым наибольший доход будет тогда, когда продано $\frac{2}{3}$ от максимально возможного числа билетов. Этот теоретически полученный результат полностью совпадает с выводом, который учащиеся должны сделать по итогам проведенных компьютерных экспериментов.

В свете целевых установок данной главы такое совпадение теоретических результатов и результатов компьютерного эксперимента — важный аргумент в пользу достоверности добытой информации об оптимальной стратегии ценообразования.

Тема 4. Международные исследования по оценке уровня информационной грамотности учащихся

В заключительном параграфе главы 1 мы предлагаем еще раз посмотреть на проблемы, связанные с информатизацией общества, но несколько с иной стороны, нежели это было сделано в § 3. Речь в нем в первую очередь идет об изменении приоритетов в оценке путей и ключевых проблем информатизации. И в нашей стране, и в мировом сообществе основным тормозом процесса информатизации долгое время считалась недостаточная оснащенность компьютерной техникой и информационными технологиями. Ведущие специалисты в области информатизации давно указывали, что главная проблема состоит в том, готова ли бóльшая часть общества к грамотному использованию информационных технологий во всех сферах человеческой деятельности [1]. При этом основной акцент делается не на технологическом компоненте, а на информационной грамотности населения. В последние несколько лет в России во многом благодаря усилиям профессора К. Колина осознаются те гуманитарные проблемы, которые возникли как следствие бурного вступления нашей страны в информационный этап ее развития. К таким гуманитарным проблемам относятся:

- интернационализация рынка труда и образовательных услуг;
- агрессивная информационная политика других государств;
- снижение роли русского языка как коммуникативной основы общества.

Не останавливаясь подробно на анализе этих проблем, отметим, что неадекватное их разрешение гражданами нашей страны во многом связано именно с неумением проводить анализ и осуществлять рефлексию той информации, которая поступает (а нередко весьма жестко навязывается, особенно по интернет-каналам) по указанным направлени-

---

[1] Уместно в связи с этим вспомнить слова Г. Р. Громова, одного из основоположников исследований в области информатизации: «Оценивать уровень информатизации масштабами выпуска ЭВМ (пусть даже и безупречно работоспособными), успехами в реализации престижных проектов искусственного интеллекта и т. д. — это по существу то же самое, что оценивать успехи в развитии, скажем, сельского хозяйства числом тракторов, комбайнов, гектарами пашни, кубокилометрами мелиорации и т. д.» (Громов Г. Р. ЭВМ и информатизация общества: старые мифы и новые проблемы // Микропроцессорные средства и системы. — 1988. — № 5. — С. 85—88). Это было сказано еще в 1988 году!

ям. Все это стоит за той фразой в учебнике, которая звучит так: «Ведь сегодня информационная сфера общества становится главной ареной конкурентной борьбы, а значит, уровень информационной грамотности населения впрямую относится к вопросам национальной безопасности». Важно, что в этом вопросе, как и во многих других, мы постепенно уходим от позиции, что у нас все хорошо только потому, что это у нас. Мы начинаем соотносить ситуацию в нашей стране с тем, как обстоят дела в мировом сообществе. Так что вторая цель § 9 — продемонстрировать учащимся, как осуществляется такое соотнесение.

Надо иметь в виду, что исследования PISA шире, чем проверка уровня информационной грамотности. В частности, это проявляется в том, что задания в PISA сгруппированы по нескольким предметным направлениям: чтение, математика, естественные науки и т. д. Предлагаемые в учебнике задания взяты из группы заданий по чтению; их цель — проверка умений учащихся извлекать информацию при чтении текстов, оценивать ее и формулировать на ее основе те или иные суждения. Отметим, что в ряде заданий PISA формулируются вопросы, относящиеся к *форме* изложения. Поскольку это, на наш взгляд, относится уже к сфере филологии, а не информатики, мы выбрали для учебника те задания, в которых такие вопросы отсутствуют. Ниже мы предлагаем еще одно задание, представляющее собой сокращенный вариант задания из PISA, которое может быть использовано для закрепления навыков содержательной работы с текстовой информацией.

Задание 6. Ниже помещены тексты двух писем, полученных по Интернету. Оба они о граффити — так называют рисунки на стенах или других местах, выполненные без официального разрешения.

«Я киплю от злости, так как в четвертый раз стену школы очищают и перекрашивают, чтобы покончить с граффити. Творчество — это прекрасно, но почему же не найти такие способы самовыражения, которые не причиняли бы лишний ущерб обществу?

По моему представлению, здания, ограда, парковые скамейки сами по себе произведения искусства. И разве не жалко портить эту архитектуру росписью, не говоря уже о том, что используемый для этого метод разрушает озоновый слой. И я не могу понять, почему эти самозваные художники так злятся, когда их так называемые художественные полотна убирают с глаз долой снова и снова.

Хельга».

«У людей разные вкусы. Общество перенасыщено информацией и рекламой. Знаки торговых компаний, названия магазинов. Большие навязчивые плакаты по обеим сторонам улиц. Приемлемо ли все это? В основном да. А приемлемы ли граффити? Некоторые говорят да, некоторые — нет. А спросили ли те, кто ставит рекламные щиты, вашего разрешения? Нет. Тогда должны ли это делать люди, рисующие на стенах? Не просто ли это вариант общения, например ваше собственное имя, названия партий или большие произведения искусств на улице?

Только вспомните о полосатой и клетчатой одежде, появившейся в магазинах несколько лет назад. Модели и цвета были скопированы с разрисованных бетонных стен. Довольно забавно, что и эти модели, и цвета принимаются сегодня в обществе, а граффити в том же стиле считаются ужасными.

Да, трудные времена настали для искусства.

София».

Используйте эти письма для ответов на вопросы, предлагаемые ниже.

а) В чем состоит цель каждого из писем:

- объяснить, что такое граффити;
- выразить свое мнение о граффити;
- продемонстрировать популярность граффити;
- рассказать людям, что очень много средств тратится, чтобы смыть эти росписи?

б) Почему София ссылается на рекламу?

в) С каким из этих двух писем вы согласны? Дайте обоснование своей точки зрения, при этом используя то, что сказано в одном из писем или в них обоих.

г) На основе анализа содержания (*что* говорится) и стиля (*как* говорится) обоих писем объясните, кто из этих двух авторов, по вашему мнению, написал письмо лучше. Обоснуйте свой ответ, ссылаясь на то, как написаны оба или одно из этих писем.

Вопросы, предложенные в этом задании, воспроизведены нами дословно. Однако мы не считаем, что именно в такой постановке они должны прозвучать для учащихся. К примеру, вопрос в пункте *в* заставляет ученика невольно принять одну из точек зрения [1]. На самом же деле вполне возможно, что он не согласен ни с той ни с другой. Или, скажем, он может видеть (и это было бы замечательно!) позитивную сторону в аргументах как одного, так и друго-

[1] Это яркий пример одного из вариантов психологического программирования личности. Человеку предлагается выбрать один из двух вариантов, хотя он может быть не согласен с обоими. Если такая процедура повторяется (возможно, с вариациями) несколько раз, у человека вырабатывается определенный стереотип принятия решения.

го автора. Это подсказывает необходимость поиска компромиссного решения. Воспитание человека, умеющего выслушивать аргументы нескольких сторон, выделять в них позитивный компонент и на его основе принимать конструктивные решения, приемлемые для дискутирующих сторон, — это одна из целей образования в современном обществе. На наш взгляд, такое умение лежит в основе воспитания толерантной личности, где толерантность понимается не как «всеядность» и беспринципное приятие любой позиции, а как конструктивное стремление понять существующее разнообразие точек зрения. В этой связи нам представляется более адекватным сформулировать пункт *в* следующим образом [1]:

в) Какие аргументы за граффити и против них, приведенные в этих письмах, вам представляются достаточно обоснованными? Свой ответ сформулируйте с опорой на то, что сказано в одном из писем или в них обоих.

После обсуждения этого вопроса нам представляется важным, чтобы учащиеся, как и в задании 4в к § 9, оценили знания и приобретенные навыки, полученные ими при выполнении данного задания.

Что касается вопроса в пункте *г*, то его, на наш взгляд, естественно ставить только перед учащимися гуманитарной специализации. В формулировке этого вопроса как раз присутствуют элементы филологического анализа, о которых мы упоминали выше.

В заключение мы хотим информировать учителя, что отношение отечественной педагогической науки к исследованиям PISA далеко не однозначно. На одном полюсе находятся те, кто считает результаты этих исследований приоритетными в формировании образовательной политики страны. Их главный аргумент — соответствие нашего образования мировым стандартам и на этой основе более легкая интеграция в мировую систему образования, в том числе в так называемый Болонский процесс, ориентированный на установление единых для Европы стандартов качества образования и признание на этой основе равенства документов, подтверждающих уровень квалификации, полученной выпускниками учебных заведений разных стран. Их противники считают, что безоговорочное принятие таких стандартов спо-

[1] Мы ранее привели исходную формулировку лишь для того, чтобы учитель, если он сочтет это целесообразным, мог продемонстрировать, как именно формулируются задания PISA (например, для подготовки учеников к прохождению таких испытаний). Мы не исключаем также, что учитель может организовать обсуждение не только ответа на исходную формулировку вопроса *в*, но и саму постановку этого вопроса (в указанном нами ключе).

собно нанести существенный ущерб нашей системе образования, в которой, как известно, приоритетом является фундаментальный, а не прикладной аспект. Мы здесь стоим на компромиссной позиции, считая, что результаты исследований PISA могут и должны служить индикатором уровня освоения информационной грамотности, должны учитываться при выработке стратегии образования, но не могут быть в этих вопросах основополагающими. Именно с этих позиций представлен материал в § 9.

Нам представляется целесообразным, чтобы по крайней мере одно из двух заданий PISA учащиеся выполнили в классе. Весьма желательно, чтобы при этом они не просто давали ответ на тот или иной вопрос, а указывали, на каком основании ими дан именно такой ответ. К примеру, ответ на вопрос 2 задания 1 о годе, соответствующем начальной точке графика, может быть получен из рассмотрения масштаба на оси абсцисс, но может быть определен и из сопроводительного текста, где говорится, что озеро появилось примерно 11 000 лет до н. э., а на графике отражен как раз момент появления озера. Здесь же фактически содержится и ответ на вопрос 3 того же задания. Однако информационно грамотный ответ должен был бы звучать примерно так: «На графике в начале координат отражен момент появления озера — нулевая глубина, а в тексте указано, что это произошло примерно за 11 000 лет до н. э. Этот вывод подтверждается и сопоставлением данной точки с выбранным на оси абсцисс масштабом. Кроме того, это свидетельствует о том, что на графике изображен именно этот момент возрождения озера, а не какой-либо другой». В таком ответе, во-первых, проявляется такой элемент информационной культуры, как сопоставление данных, полученных из разных источников информации (проверка достоверности — информационно грамотный человек всегда должен задаваться вопросом, как еще можно подтвердить полученную информацию); во-вторых, демонстрируется умение анализировать информацию, представленную в разной форме — текстовой и графической. Первый момент особенно важен, поскольку учащиеся приучены прекращать рассмотрение задачи, как только ими получен ответ. И хотя о пользе поиска других путей решения задачи немало сказано в методической литературе (особенно по математике), на практике это воплощается весьма редко. На наш взгляд, учителю информатики здесь предоставляется удобный случай побуждать учеников к поиску различных путей получения информации и выполнения задания.

Так же не вполне однозначен способ нахождения ответа на первый вопрос. Некоторые учащиеся могут предложить найти ответ, исходя из того, что наше время на оси абсцисс фактически изображается границей рисунка, и на этом основании по графику найти требуемую величину. Более грамотное решение состоит в том, чтобы извлечь из сопроводительного текста прямое указание, что глубина озера сегодня практически та же, что 10 000 лет назад, а затем по графику найти эту глубину.

Ответ на четвертый вопрос этого задания достаточно очевиден: древние художники, вероятнее всего, рисовали то, что сами видели. Получить ответ на пятый вопрос тоже несложно — исчезновение этих животных с наскальной живописи произошло примерно в 1700 г. до н. э., т. е. после того, как уровень озера снижался в течение более тысячи лет.

В вопросах и заданиях к § 9 ответы на первые три из них фактически содержатся в объяснительном тексте параграфа. Они призваны еще раз акцентировать внимание учащихся на ключевых положениях предложенного материала. Что касается задания 4, то для выполнения пунктов *а* и *б* надо снова воспользоваться теми умениями работы с информацией, о которых шла речь в § 2, 4 (в частности, использование таксономии Блума для постановки вопросов, способствующих осмыслению информации) и § 5 (в частности, построение кластера по схеме, представленной на рисунке 1.2). Тем самым это задание выступает как обобщающее повторение основных концептов главы. Пункт *в* этого задания затрагивает третий этап работы с информацией — этап рефлексии, т. е. оценки результатов работы с информацией для себя лично. Поэтому проверка выполнения этого пункта — вопрос довольно щекотливый, ибо далеко не каждый ученик готов вынести на публичное обсуждение то, что относится к нему лично. Именно в связи с этим мы рекомендуем пункт *б*, служащий основой для выполнения пункта *в*, обсуждать небольшой (3—4 человека) группой. Тогда и в пункте *в* будет представлена точка зрения этой группы, что снимает указанную выше психологическую проблему.

Мы считаем весьма полезным предложить учащимся выполнить такое же задание для приведенных выше интернет-писем Хельги и Софии. Оно в целом даже проще, чем задание 4 из учебника, поскольку авторы текстов более ярко выражают расхождение в своих позициях.

ГЛАВА 2

Кодирование информации. Представление информации в памяти компьютера

С точки зрения уже имеющихся у школьников знаний содержание этой главы достаточно полярно. С одной стороны, основные сведения о кодировании информации, скорее всего, известны учащимся из базового курса информатики 8—9 классов. Этим вопросам уделялось определенное внимание в 10 классе. Тем не менее Федеральный компонент стандарта полного (среднего) образования предписывает обязательное изучение вопросов, представленных в базовой части этой главы, так, как будто до этого учащиеся ничего об этом не слышали. Наше решение рассматривать эти вопросы в курсе информатики 11 класса продиктовано двумя соображениями. Во-первых, если выясняется, что учащиеся в достаточной степени владеют указанным материалом, учитель получает возможность кратко повторить его, экономя при этом всегда дефицитное время для более основательного изучения других разделов курса. Во-вторых, значительную долю заданий ЕГЭ составляют вопросы, относящиеся к теме «Информация и ее кодирование» — на ЕГЭ-2008 из 28 заданий разделов А и В 10 относились именно к этой теме. И если учащиеся планируют сдавать ЕГЭ по информатике, то целесообразно организовать систематическое повторение данной темы. Изложение материала в базовой части нацелено именно на организацию такого повторения.

С другой стороны, в профильной, углубленной части курса рассматриваются достаточно тонкие вопросы, связанные с методами кодирования информации, позволяющими осуществлять поиск и автоматическое устранение ошибок, а также уменьшение информационного объема (так называемые алгоритмы сжатия информации). Эти вопросы никак не отражены в ЕГЭ, так что объем и глубина изучения данного материала могут варьироваться и целиком зависят от заинтересованности учащихся и целевых установок преподавателя.

Тема 5. Кодирование числовой информации

Кодирование числовой информации — это представление чисел в той или иной системе счисления. С понятием системы счисления учащиеся, скорее всего, знакомы по урокам математики; правда, основное внимание там уделялось, естественно, десятичной позиционной нумерации. Возможно, они знают и о такой непозиционной системе счисления, как римская. С точки зрения информатики акцент здесь должен быть в первую очередь сделан на том, что любая система счисления — это некоторый способ кодирования числовой информации. Можно сказать даже больше: на самом деле это формальный язык над алфавитом, в данном случае из десяти символов, с жестко заданными правилами образования слов в этом языке. Об этом, в частности, говорилось в § 3 учебника для 10 класса. Мы упоминаем здесь об этом лишь потому, чтобы учителю была ясна именно информатическая, а не математическая составляющая понятия «система счисления».

Начать следует, по нашему мнению, с повторения того, что известно учащимся о кодировании символьной информации из предшествующего курса информатики. Это позволит перекинуть мостик от десятичной к двоичной системе счисления — ведь вся информация в компьютере должна быть представлена в двухсимвольном алфавите.

Понятие позиционной системы счисления вводится в § 10, а в § 11 излагаются алгоритмы перевода из одной системы в другую. В этих параграфах мы ограничиваемся действиями лишь с целыми числами. По нашему мнению, учащиеся должны наизусть знать таблицы перевода первых пятнадцати чисел из десятичной в двоичную и шестнадцатеричную системы; эти таблицы также приведены в объяснительном тексте указанных параграфов. Однако в последние два года в демонстрационных вариантах ЕГЭ присутствуют задания на перевод дробных чисел из одной системы в другую. Этим алгоритмам посвящен § 12.

Ответы на вопросы 1—3 к § 10 содержатся в объяснительном тексте. На вопрос 4 ответ отрицателен. Если бы такая позиционная система существовала, то наличие только одной цифры означало бы, что основанием системы служит число 1. Однако любая степень числа 1 равна 1, поэтому запись числа в такой системе равносильна выписыванию цифры этой системы в количестве, равном записываемому числу. Но тогда значение цифры не зависит от занимаемой ею позиции, т. е. такая система не будет позиционной.

Это, однако, не означает, что такая система не используется. Более того, исторически это первая система счисления, которую использовали люди. Зарубки на дереве, насечки на кости животного, узелки на веревке — все это примеры использования данной системы для записи чисел у древних народов. Здесь уместно предложить кому-либо из учащихся подготовить краткое сообщение об истории развития систем счисления у разных народов.

Ответ на задание 5 также содержится в объяснительном тексте § 10, тем не менее мы перечислим преимущества использования двоичной системы счисления:

1) техническое удобство двоичного кодирования;
2) простота реализации действия сложения;
3) сводимость умножения к операциям сдвига и сложения.

Позже, в § 23, будет показано удобство этой системы для представления отрицательных чисел.

Задание 6 из § 10 несложно, но заставляет учащихся осмыслить, какими алгоритмическими конструкциями, сами того не подозревая, они пользовались при изучении в «далеком детстве» операции сравнения чисел. Перед составлением соответствующих алгоритмов учащиеся должны обсудить, как они выполняют сравнение натуральных чисел.

Вот возможные варианты ответов на вопросы задания 6.

```
а) Алгоритм
 { Запросить 1-е число;
   Запросить 2-е число;
   Если (старший разряд первого числа больше
            старшего разряда второго числа)
   то { Сообщить "Первое число больше второго"; }
   иначе {
            Если (старший разряд первого числа меньше
                     старшего разряда второго числа)
            то { Сообщить "Первое число меньше второго"; }
            иначе
                 { Если (младший разряд первого числа
                           больше младшего разряда второго
                           числа)
            то { Сообщить "Первое число больше второго"; }
            Если (младший разряд первого числа меньше
                     младшего разряда второго числа)
            то { Сообщить "Первое число меньше второго"; }
            Если (младший разряд первого числа равен
                     младшему разряду второго числа)
            то { Сообщить "Первое число равно второму"; }
            }           (* конец иначе *)
   }                    (* конец иначе *)
 }                      {* конец алгоритма*}
```

Мы сразу приведем алгоритм для задания *в*, поскольку задание *б* является его частным случаем.

```
в) Алгоритм
 { Запросить 1-е число;
   Запросить 2-е число;
   k := 1;
   Если (k-й слева разряд 1-го числа больше k-го слева
           разряда 2-го числа)
   то { Сообщить "Первое число больше второго"; }
   иначе
           { Если (k-й слева разряд 1-го числа меньше
                    k-го слева разряда 2-го числа)
             то { Сообщить "1-е число меньше 2-го"; }
           }          (* конец иначе *)
   Делать пока (k-й слева разряд 1-го числа равен
                k-му слева разряду 2-го числа) и
                (k-й слева разряд 1-го числа
                не последний)
           { k := k + 1;
           Если (k-й слева разряд 1-го числа больше
                 k-го слева разряда 2-го числа)
           то { Сообщить "1-е число больше 2-го"; }
           Если (k-й слева разряд 1-го числа меньше
                k-го слева разряда 2-го числа)
           то { Сообщить "1-е число меньше 2-го"; }
           }          (* конец цикла *)
   Если (k-й слева разряд 1-го числа равен k-му слева
          разряду 2-го числа)
   то { Сообщить "1-е число равно 2-му"; }
 }
```

Сильным учащимся можно предложить составить алгоритм сравнения любых двух чисел.

Выполнение задания 7 сводится к решению уравнения $3x + 2 = 23$, где через x обозначено основание искомой системы счисления. Ответ: 7.

Более подготовленным учащимся можно предложить следующее дополнительное задание (мы продолжаем нумерацию дополнительных заданий, начатую в главе 1):

Задание 7. а) Может ли десятичное число 371 оказаться в другой системе счисления записанным теми же цифрами, но в обратном порядке (т. е. как 173)?

б) Число 551 записано в десятичной системе счисления. После перевода этого числа в позиционную систему счисления с другим основанием получилось число, записанное теми же цифрами, но в другом порядке. Чему равно основание новой системы счисления?

Ответ: а) да, в системе с основанием 16; б) 21.

Решение. а) Пусть x — основание искомой системы счисления. Это означает, что для x выполнено равенство (в обычной десятичной системе счисления) $x^2 + 7x + 3 = 371$. У этого квадратного уравнения корнями являются 16 и –23. Поскольку основанием системы счисления может быть только натуральное число, то годится лишь 16.

б) Пусть x — основание искомой системы счисления. В этой системе число может записываться либо как 515, либо как 155. Это означает, что для x возможны следующие равенства: $5x^2 + x + 5 = 551$ и $x^2 + 5x + 5 = 551$. Первое из этих уравнений целых корней не имеет, корни второго уравнения равны 21 и –26. Поскольку основанием системы счисления может быть только натуральное число, то годится число 21.

Последующие задания из § 10 относятся к числам, записанным в двоичной системе счисления.

Задание 8 выполняется устно; в зависимости от подготовленности класса его можно выполнить до задания 6 (как подготовку к нему) или после него (как применение разобранных алгоритмов).

Действия над числами в задании 9 мы рекомендуем выполнять столбиком, чтобы лишний раз продемонстрировать высказанный в объяснительном тексте тезис об универсальности алгоритмов выполнения действий столбиком, в которых используются вспомогательные алгоритмы сложения и умножения однозначных чисел в соответствующей позиционной системе счисления, определенные соответствующими таблицами. Как это ни покажется удивительным, только после изучения данного материала многие школьники осознают, почему в начальной школе их так настойчиво заставляли учить таблицы сложения и умножения однозначных чисел — фактически речь шла об овладении этими вспомогательными алгоритмами.

Ответы в задании 9 таковы: $101 + 1101 = 10010$; $101 \cdot 1101 = 1000001$.

В задании 10а ответ очевиден: увеличение числа в двоичной системе в 2 раза равносильно приписыванию 0 в конец числа — ведь умножение на 2 повышает на 1 степень числа 2 в каждом разрядном слагаемом, т. е. сдвигает каждый разряд на один влево, а в разряде единиц появляется 0. Здесь можно услышать от учащихся и другой ответ: умножение на 2 равносильно умножению в двоичной системе на 10, что в свою очередь равносильно приписыванию 0. Однако, чтобы ответить на вопрос, почему умножение на 10 равносильно приписыванию 0, придется все равно рассмотреть представление числа в виде разрядных слагаемых, т. е., по существу, провести вышеприведенное рассуждение.

Два оставшихся пункта задания 10 в комментариях, как нам кажется, не нуждаются.

Учащимся с углубленной математической подготовкой можно вместо задания 10а предложить следующее, более общее задание:

Задание 8. а) Как изменится натуральное число, записанное в системе счисления с основанием b, если к нему справа приписать 0?

б) Любое число, записанное в системе счисления с основанием b, очевидно, увеличится, если в некотором месте между его цифрами вписать 0. В каком месте надо вписать 0, чтобы число увеличилось в наибольшее число раз?

Ответ на вопрос пункта *а* очевиден — число увеличится в b раз. Для получения ответа в пункте *б* запишем произвольное натуральное число n, содержащее k цифр, в виде суммы $x + y$, где y — число, составленное из m последних цифр числа n, перед которыми будет вписан 0. Тогда число, полученное вписыванием нуля, равно $bx + y$. Отношение $\frac{bx + y}{x + y}$ как раз и показывает, во сколько раз новое число больше прежнего. Но $\frac{bx + y}{x + y} = b - \frac{y(b-1)}{x + y} < b$. Это означает, что отношение будет наибольшим, когда 0 приписывается в конец числа — только в этом случае число увеличивается в b раз, во всех остальных случаях это увеличение будет меньше.

Задание 11 из § 10 нам представляется также весьма простым, оно даже проще, чем аналогичные задания с числами в десятичной системе счисления, в частности потому, что старший разряд любого числа в двоичной системе всегда равен 1.

В § 11 продолжается изучение систем счисления и приводятся алгоритмы перевода целых чисел из десятичной системы в систему с другим основанием и обратно. Отметим, что предлагаемый нами алгоритм перевода в десятичную систему основан на так называемой схеме Горнера. Весьма желательно, чтобы учащиеся усвоили эту схему, поскольку она применяется не только для перевода чисел из одной системы счисления в другую, но и для вычисления значения многочлена по заданному значению аргумента — можно без преувеличения сказать, что это главное применение схемы Горнера.

В классах с углубленной математической подготовкой можно предложить к обсуждению принципиальный вопрос: почему каждое натуральное число можно записать в системе счисления с любым основанием $b > 1$ (в том числе и для $b = 10$)? Ответ на этот вопрос несложен, но может оказаться неожиданным для учащихся. Существование такой записи следует из того, что к любому натуральному числу может быть применена схема Горнера, а она как раз и дает нужное представление. У школьников может возникнуть недоумение: ведь схема Горнера применялась к числам, уже записанным в позиционной системе счисления (десятичной). Образуется, как говорят математики, порочный логический круг. Но на самом деле применение схемы Горнера не зависит от того, как изначально были представлены натуральные числа — египетскими иероглифами, в римской нумерации или как-то еще. Важно, что над натуральными числами можно всегда выполнить операцию деления с остатком, ибо цифры — это именно остатки от последовательного деления на основание системы счисления (см. обсуждение задания 8 из § 11, приведенное ниже).

Второй вопрос, над которым полезно задуматься математически ориентированным школьникам, таков: почему представление любого натурального числа в системе счисления с заданным основанием однозначно? Нередко учащиеся отвечают на него, ссылаясь, что алгоритм, представленный схемой Горнера, на каждом шаге своего исполнения определяет очередную цифру однозначно. На это следует возразить, что вполне может существовать какой-нибудь другой алгоритм представления числа в данной позиционной системе счисления, и нет никакой гарантии, что в результате исполнения такого алгоритма получится тот же самый результат. Нетрудно тем не менее понять, что, каким бы алгоритмом ни пользоваться для получения записи натурального числа в заданной системе счисления, последняя цифра всегда равна остатку от деления числа на основание системы счисления, предпоследняя — остатку от деления получившегося частного на основание системы счисления и т. д. А тот факт, что частное и остаток при делении одного натурального числа на другое определены однозначно, — это математическая теорема (которая, правда, в обычном школьном курсе математики формулируется, как правило, без доказательства) [1].

---

[1] Сказанное не следует воспринимать как упрек школьному курсу математики. Доказательство существования и единственности частного и остатка при делении натуральных чисел восходит к аксиоматическому определению натурального числа, ни в коей мере не предусмотренному стандартным курсом школьной математики. Хотим также предупредить, что доказательство существования и единственности представления произвольного натурального числа в позиционной системе счисления, приведенное в книге Е. В. Андреевой «Математические основы информатики. Элективный курс» (М.: Бином, 2005), не вполне корректно (не говоря уже о его неоправданной усложненности). Тем не менее эта книга включена нами в список литературы для дополнительного чтения, приведенный в конце учебника 11 класса, поскольку она имеет ряд несомненных достоинств.

Именно эти свойства позиционной системы счисления с фиксированным основанием — существование и единственность записи *любого* натурального числа с помощью *конечного* количества алфавитных символов — определяют их преимущества перед другими системами счисления. Ведь, к примеру, в римской нумерации конечным набором цифр можно записать только конечное количество чисел. Позиционная система оказалась эффективным способом записи чисел еще и в том смысле, что имеются простые алгоритмы для основных арифметических операций — сложения, умножения, вычитания и деления.

В какой момент провести обсуждение указанных вопросов, решать учителю. Нам представляется, что это удобно сделать в контексте обсуждения с учащимися задания 8 к § 11 (методические рекомендации к выполнению этого задания приведены ниже). Отметим, что рассмотрение указанных вопросов чрезвычайно важно для воспитания культуры алгоритмического мышления. Несколько позже — в главе 5 — подобные вопросы будут обсуждаться систематически.

Особое место занимают алгоритмы свертки двоичной записи чисел в шестнадцатеричную и обратной развертки. В частности, учащиеся должны выучить приведенную в учебнике таблицу перевода в шестнадцатеричную систему первых шестнадцати натуральных чисел. Задание 1а предназначено для проверки знания учащимися алгоритма перевода чисел из двоичной системы в шестнадцатеричную и обратно. В задании 1б предлагается обсудить аналогичные алгоритмы для восьмеричной системы: для перевода числа из восьмеричной системы в двоичную надо каждую цифру расписать в триаду (см. таблицу 2.1), а для обратного перевода нужно разбить запись числа на тройки цифр, отсчитывая их справа налево, и каждую такую триаду записать восьмеричной цифрой. Например: $35_8 = 11101_2$; $10101110_2 = \underline{010}\ \underline{101}\ \underline{110} = 256_8$.

Таблица 2.1

Двоичная система счисления	Восьмеричная система счисления
1	1
10	2
11	3
100	4
101	5
110	6
111	7

Перевод чисел, предложенных в задании 2, учащиеся должны выполнить, используя алгоритм последовательного деления. Вот что у них должно получиться:

$$579 = 4304_5 = 1455_7 = 713_9;$$
$$2468 = 14333_5 = 10124_7 = 3342_9.$$

При выполнении задания 3 учащимся можно предложить для первых двух чисел вычислить десятичное представление чисел по определению, а затем с помощью схемы Горнера. Вот как это выглядит для числа 866_9:

$$866_9 = 8 \cdot 9^2 + 6 \cdot 9 + 6 = 8 \cdot 81 + 54 + 6 = 648 + 60 = 708;$$

8	6	6
8	78	708

А для числа 1287_{11} так:

$$1287_{11} = 1 \cdot 11^3 + 2 \cdot 11^2 + 8 \cdot 11 + 7 =$$
$$= 1 \cdot 1331 + 2 \cdot 121 + 88 + 7 = 1331 + 242 + 95 = 1668;$$

1	2	8	7
1	13	151	1668

Все эти вычисления учащиеся могут проводить с использованием калькулятора. Но важно обратить их внимание на количество действий, которое приходится выполнять, когда вычисления ведутся по определению, и когда — по схеме Горнера. Можно поставить этот вопрос для n-значного числа, чтобы учащиеся убедились, что применение схемы Горнера всегда требует выполнения меньшего числа действий. Мы в предлагаемом курсе не рассматриваем специально вопросы эффективности алгоритмов, но рекомендуем обращать на это внимание учащихся в каждом подходящем случае.

Для остальных чисел из этого задания мы приведем вычисления только по схеме Горнера.

4	5	2	6	7
4	53	638	7661	91 939

5	4	2	3
5	39	275	1928

7	4	2	6	8
7	67	605	5451	49 067

Эти вычисления естественно проводить с помощью калькулятора, поэтому выполнение данного задания можно приурочить к работе в компьютерном классе.

К заданию 4 мы приведем только ответы:
$19 = 10011_2 = 13_{16}$;
$44 = 101100_2 = 2C_{16}$;
$129 = 10000001_2 = 81_{16}$;
$561 = 1000110001_2 = 231_{16}$;
$1322 = 10100101010_2 = 52A_{16}$.

Здесь можно обсудить с учащимися, какой путь получения ответов лучше: сначала перевести в двоичную систему, а затем воспользоваться алгоритмом свертки в шестнадцатеричную или наоборот. На наш взгляд, быстрее к ответам приводит сначала перевод в шестнадцатеричную систему методом деления, а затем использование развертки каждой цифры в двоичный код.

Приведем ответы к заданию 5 (кроме первого табличного ответа; остальные даны в виде применения схемы Горнера; ответом является число в толстой рамке):
$1001_2 = 9$;

1	0	1	0	1
1	2	5	10	21

1	1	1	0	0	1
1	3	7	14	28	57

1	0	1	1	1	1	0	1
1	2	5	11	23	47	94	189

В таком же виде приведем ответы к заданию 6:
$25_{16} = 37$; $4F_{16} = 79$;

1	A	7
1	26	423

A	B	C
16	266	4268

D	1	A	E
13	209	3354	53 678

F	F	F	F
15	265	4255	68 095

В задании 7 учащиеся записывают двоичный код наибольшего семизначного числа — это 1111111_2, затем переводят его в десятичный вид. Получается $2^7 - 1 = 127$.

Этот ответ легко получить, если заметить, что $1111111_2 = 10000000_2 - 1 = 2^7 - 1$. Аналогично получаются ответы для наибольшего 11-разрядного и 15-разрядного чисел.

В задании 8 объяснение, которое требуется дать в пунктах *а* и *б*, основывается на одном и том же представлении записи числа, скажем, c в системе счисления с основанием b. Вот эта запись:

$$c = a_1 \cdot b^n + a_2 \cdot b^{n-1} + \ldots + a_{n-1} \cdot b^1 + a_n \cdot b^0 =$$
$$= (\ldots ((a_1 \cdot b + a_2) \cdot b + a_2) \cdot \ldots + a_{n-1}) \cdot b + a_n.$$

Теперь легко увидеть, что схема Горнера — это в точности следование правилам вычисления значения выражения, содержащего скобки.

На то же самое «скобочное» выражение можно взглянуть с другой стороны — справа налево. Тогда становится ясно, что a_n — это остаток при делении c на b, а выражение в самых внешних скобках — частное при таком делении. В свою очередь, a_{n-1} — это остаток при делении только что полученного частного на b, а выражение в следующих по порядку вложенности скобках — частное при таком делении. И т. д. Это и есть объяснение, почему при последовательном делении с остатком получаются цифры числа в искомой системе счисления.

Для объяснения алгоритма перевода из шестнадцатеричной системы в двоичную надо каждую шестнадцатеричную цифру расписать в двоичной системе как сумму разрядных слагаемых. Если теперь в запись числа шестнадцатеричными разрядными слагаемыми вставить полученную сумму двоичных разрядных слагаемых каждой шестнадцатеричной цифры, то легко заметить, что получается запись числа суммой двоичных разрядных слагаемых. Отсюда уже получается запись числа в двоичной системе. Рассуждение, проводимое в обратном порядке, обосновывает алгоритм перевода из двоичной системы в шестнадцатеричную.

В ЕГЭ довольно активно используются задания, в которых фигурирует восьмеричная система счисления. Ниже приведены примеры двух таких заданий, опубликованные в демонстрационной версии и одном из вариантов ЕГЭ-2008.

Задание 9. а) Вычислите сумму чисел x и y при $x = \mathrm{A6}_{16}$, $y = 75_8$. Результат представьте в двоичной системе счисления.

б) Чему равна разность чисел 101_8 и 100111_2?

1) $1\mathrm{A}_{16}$; 2) 54_8; 3) 42_8; 4) E_{16}.

Как показывает анализ работ учащихся, большинство из них при выполнении пункта *а* переводят каждое из чисел в десятичную систему счисления, затем складывают получившиеся числа и результат переводят в двоичную систему. На самом деле такое решение довольно затратный по времени процесс, а в условиях весьма жесткого лимита времени, которое выделяется на выполнение заданий частей А и В, фактор времени весьма существенен. Более быстрым является перевод сразу обоих чисел в двоичную систему и вычисление суммы в двоичной системе. Вот как это выглядит:

$A6_{16} = 10100110_2$; $75_8 = 111101_2$;

$$\begin{array}{r} 10100110 \\ +\quad 111101 \\ \hline 11100011 \end{array}$$

Ответ: 11100011_2.

В задании *б* тоже удобно число 101_8 перевести в двоичную систему: 1000001_2. Тогда

$$\begin{array}{r} 1000001 \\ -\quad 100111 \\ \hline 11000. \end{array}$$

Но $11000_2 = 30_8 = 1A_{16}$, так что верный ответ: 1.

Регулярное наличие подобных заданий в ЕГЭ показывает, что действиям над числами в двоичной системе счисления надо уделить должное внимание в подготовке учащихся, несмотря на кажущуюся простоту выполнения таких действий.

Отметим еще одно задание демонстрационного варианта ЕГЭ, в котором фигурируют системы счисления.

Задание 10. Укажите все основания систем счисления, в которых запись числа 23 оканчивается на 2.

Решение основывается на том замечании, что после вычитания из числа его последней цифры получившаяся разность обязательно делится на основание системы счисления. В условиях данного задания это означает, что основание системы счисления является делителем числа 21. Тем самым основанием системы счисления может быть только одно из чисел 3, 7 или 21. Каждое из них годится, поскольку $23 = 212_3 = 32_7 = 12_{21}$.

Чтобы проверить понимание учащимися сути позиционного принципа записи чисел, можно предложить им следующее задание:

Задание 11. а) Каким должно быть основание системы счисления, для которой число 100 является нечетным?

б) Верно ли, что в любой системе счисления число 111 нечетно?

Для ответа на первый вопрос достаточно заметить, что в системе счисления с основанием b число $100 = b^2$. Оно будет нечетным тогда и только тогда, когда нечетно само b.

В пункте *б* число $111 = b^2 + b + 1 = (b + 1)\,b + 1$. Произведение $(b + 1)\,b$ всегда четно, поскольку это произведение двух последовательных натуральных чисел и, значит, один из множителей обязательно четен. Следовательно, $(b + 1)\,b + 1$ нечетно при любом b.

Между прочим, это задание демонстрирует учащимся, что формулировки признаков делимости, известные им с 5 класса, тесно связаны со значением основания системы счисления. В частности, известный признак делимости на 2, позволяющий по последней цифре определять, четно число или нечетно, применим только для позиционных систем с четным основанием.

Для учащихся, хорошо владеющих программированием, можно предложить следующее задание, по сюжету продолжающее задачу из приведенного выше дополнительного задания 7.

Задание 12. Найдите все четырехзначные числа в десятичной системе счисления, которые при переводе в систему с другим основанием дают число, записываемое теми же цифрами (причем каждая цифра употребляется столько же раз, сколько она встречалась в записи исходного числа).

Решение задачи может основываться на переборе всех четырехзначных чисел. Такой перебор осуществляется с помощью четырех вложенных циклов со счетчиком по переменным a, b, c и d, представляющим поразрядно цифры искомого четырехзначного числа. Иными словами, искомое число равно $a \cdot 10^3 + b \cdot 10^2 + c \cdot 10 + d$, причем $a \neq 0$. Пусть x — основание новой системы счисления. Тогда то же самое число в системе счисления с основанием x запишем в виде $k \cdot x^3 + l \cdot x^2 + m \cdot x + n$, где (k, l, m, n) — некая перестановка чисел a, b, c и d. Следовательно,

$$x^3 < k \cdot x^3 + l \cdot x^2 + m \cdot x + n =$$
$$= a \cdot 10^3 + b \cdot 10^2 + c \cdot 10 + d < 9999.$$

Из этого неравенства вытекает, что $x \leqslant 21$. На самом деле для каждого конкретного числа эту оценку, как правило, можно значительно улучшить: $x < \sqrt[3]{\dfrac{10^3 a + 10^2 b + 10c + d}{h}}$, где h — наименьшее из ненулевых чисел a, b, c и d. С другой стороны, наибольшее четырехзначное число, записываемое в системе счисления с основанием x, равно $10\,000_x - 1 = x^4 - 1$, т. е. $x^4 > a \cdot 10^3 + b \cdot 10^2 + c \cdot 10 + d \geqslant 1000$. Следовательно, $x \geqslant 6$. Теперь для каждого допустимого значения x осуществляем перевод исходного числа в систему счисления с основанием x. Для этого можно запрограммировать алгоритм, предложенный в § 11. Если после перевода получился тот же набор цифр, что и у исходного числа, значит, число годится.

Первые пять заданий лабораторной работы № 3, которая сопровождает изучение материала из § 11, предназначены для ознакомления с возможностями программного приложения *Инженерный калькулятор* в действиях с числами, представленными в двоичной, восьмеричной и шестнадцатеричной системах счисления. Задание 6 ставит перед учащимися проблемы, возникающие при вычислениях в связи с ограниченностью разрядной сетки. Предлагаемые в этом задании эксперименты демонстрируют явление, которое называется переполнением разрядной сетки. К счастью, *Инженерный калькулятор* устроен так, что эффект переполнения для чисел, записанных в десятичной системе, и чисел, записанных в двоичной системе, проявляется неодновременно (для десятичных чисел на несколько бóльших числах, чем для двоичных чисел), что и позволило организовать предлагаемый эксперимент. Выполнение этого задания преследует две дидактические цели: для учащихся, изучающих только базовый курс, сформировать понимание ограниченности разрядной сетки и на примере переполнения познакомить с возможными негативными эффектами; для учащихся, осваивающих профильный курс, в дополнение к сказанному создать мотивацию к изучению представления чисел в компьютере и особенностей выполнения действий над ними. Как объявлено в тексте лабораторной работы № 3, рассмотрение этих вопросов будет предпринято в § 23. В рамках этой лабораторной работы можно предложить учащимся выяснить, какое наибольшее десятичное число можно перевести в двоичную систему счисления с помощью *Инженерного калькулятора*, какое наибольшее целое число можно возвести в квадрат и т. п.

На выполнение этой лабораторной работы достаточно, на наш взгляд, одного часа учебного времени.

Второй час компьютерного практикума, предусмотренный в тематическом планировании при углубленном изучении информатики, можно уделить программированию задач, связанных с переводом натуральных чисел из одной системы счисления в другую. Эти задания в простейшем варианте позволяют повторить и закрепить навыки программирования основных алгоритмических конструкций, а в более сложном — типы и структуры данных, отличные от числовых.

Задание 13. Составьте алгоритм и напишите программу перевода натуральных чисел из десятичной системы счисления в систему счисления с основанием b. Исходное число и число b запрашиваются у пользователя; $b \leqslant 36$.

Задание 14. Составьте алгоритм и напишите программу перевода натуральных чисел, записанных в системе счисления с основанием b, в десятичную систему счисления, предварительно проверив, что запись исходного числа в b-ичной системе счисления правильна. Исходное число и число b запрашиваются у пользователя; $b \leqslant 36$.

Задание 15*. Число называется **палиндромом**, если оно читается одинаково справа налево и слева направо. Составьте алгоритм и напишите программу поиска такого $b \leqslant 36$, для которого заданное натуральное число, записанное в десятичной системе счисления, является палиндромом в системе счисления с основанием b.

Как уже было сказано, эти задания можно предлагать в двух вариантах — простом, когда $b \leqslant 9$, и усложненном, том, в котором оно сформулировано. Во втором случае учащиеся должны понимать, что при $b > 10$ в записи для обозначения цифр нужно использовать заглавные буквы латинского алфавита, последовательно кодирующие цифры, которые больше 9. Это означает, что результат работы программы естественно хранить в строковой переменной. Отметим, что в задании 15 хранить результат в строковой переменной удобно независимо от того, меньше или больше 10 будут основания рассматриваемых систем счисления. Впрочем, ничто не мешает сначала рассмотреть простой вариант, а позже, после изучения материала § 13, вернуться к усложненному варианту. Дело в том, что при переводе в систему счисления с основанием, большим 10, удобно для получения записи цифры, большей 9, использовать ASCII-код соответствующей латинской буквы [1].

Ниже приведены соответствующие этим заданиям программы на языке Паскаль.

[1] Впрочем, если не оставлять усложненный вариант задания до изучения § 13, можно воспользоваться следующим приемом: организовать символьную переменную, записав туда константу 0123456789ABCDEFGHJKLMNOPQRSTUVWXYZ. Тогда остаток при делении на основание системы счисления будет в точности номером символа в этой строковой константе.

```
program Task13;
var a, b, c : integer;
   res, s : string;
begin
   read(a, b);
   res := '';
   while a <> 0 do
   begin
      c := a mod b;
      if c < 10 then
         begin
            str(c, s);
            res := s + res;
         end else
         begin
            res := chr(ord('A') + c – 10) + res;
         end;
      a := a div b;
   end;
   writeln(res);
end.
```

```
program Task14;
var res, b, c, i, code : integer;
   s : string;
begin
   readln(s);
   read(b);
   res := 0;
   for i := 1 to length(s) do
   begin
      if (s[i] >= '0') and (s[i] <= '9') then
         begin
         val(s[i], c, code);
         res := res * b + c;
      end else
         res := res * b + ord(s[i]) – ord('A') + 10;
   end;
   writeln(res);
end.
```

В программе для задания 15 используются две подпрограммы-функции. Одна из них, предназначенная для перевода числа из десятичной системы в *b*-ичную, фактически повторяет программу Task13; другая осуществляет проверку, является ли палиндромом число, получившееся после перевода. В ней мы использовали булеву переменную, хотя сигнальная переменная могла быть и числовой.

```
program Task15;
var a, b, ans : integer;

function DecToB(a, b : integer): string;
var c : integer;
  res, s : string;
begin
  while a <> 0 do
  begin
    c := a mod b;
    if c < 10 then
      begin
        str(c, s);
        res := s + res;
      end else
      begin
        res := chr(ord('A') + c – 10) + res;
      end;
      a := a div b;
    end;
    DecToB := res;
end;

function IsPalindr(s : string): boolean;
var i: integer;
  res : boolean;
begin
  res := true;
  for i := 1 to length(s) div 2 do
    res := res and (s[i] = s[length(s) – i + 1]);
  IsPalindr := res;
end;
begin
  read(a);
  ans := 0;
  for b := 2 to 36 do
    if IsPalindr(DecToB(a, b)) then ans := b;
  if ans = 0 then write('No solution') else write(ans);
end.
```

Перевод дробных чисел из одной системы счисления в другую (§ 12) мы рекомендуем рассматривать только с теми учащимися, которые изучают информатику углубленно. В этом параграфе материал группируется в три концентра:

- алгоритм перевода десятичной дроби в b-ичную систему счисления;
- правило округления b-ичных дробей;
- алгоритмы перевода двоичных дробей в шестнадцатеричную систему счисления и обратно.

Из двух часов, отводимых на освоение материала этого параграфа, мы рекомендуем первый час посвятить рассмотрению двух первых вопросов, а во второй час рассмотреть третий вопрос и организовать закрепляющее повторение по всей теме.

Алгоритм перевода десятичной дроби в b-ичную достаточно подробно разобран в объяснительном тексте § 12, и объяснение можно, на наш взгляд, выстраивать, следуя этому тексту. В классе с хорошей математической подготовкой можно обсудить вопрос, всегда ли при переводе числа, записанного конечной десятичной дробью, в систему счисления с другим основанием будет получаться конечная или бесконечная периодическая дробь. Ответ на этот вопрос положителен и может быть обоснован теми же рассуждениями, что применяются для доказательства известного учащимся факта о конечности или периодичности десятичной дроби, представляющей произвольное рациональное число. Но в целом мы считаем, что не следует сильно углубляться в проблематику b-ичных дробей в случае произвольного основания b. Этим объясняется и то обстоятельство, что среди упражнений к этому параграфу есть только два задания, относящиеся к b-ичным дробям при b, отличном от 2, 8 и 16. При этом, переводя в десятичную систему счисления смешанное число, мы используем сведение b-ичной дроби к обыкновенной с последующим переводом в десятичную систему счисления отдельно целой части, числителя и знаменателя. Результат можно представить в виде конечной или периодической десятичной дроби, но, вообще говоря, можно ограничиться и представлением в виде обыкновенной дроби (если, конечно, в задании специально не оговорено, в каком виде должен быть представлен результат).

Приведем дополнительное задание, которое позволяет проверить не только технические навыки перевода дробей, но и понимание структуры получающихся записей дробных чисел.

Задание 16. Какая цифра будет стоять на сотом месте после запятой, если число 0,123 из десятичной системы счисления перевести:

а) в пятеричную систему счисления;

б) в семеричную систему счисления?

Решение. Осуществим перевод числа в каждую из систем счисления для того, чтобы выявить период дробной части в каждом из двух случаев (рис. 2.1). В пункте *а* получаем число $0{,}030(14)_5$, а в пункте *б* — число $0{,}(06012155344250266314)_7$. В первом случае видно, что каждая цифра, стоящая на четном месте, начиная с четвертого, равна 1. Поэтому сотая цифра после запятой тоже будет 1. Во втором случае период содержит 20 цифр и начинается сразу после запятой. Поэтому совпадают 20, 40, 60, 80 и 100-я цифры после запятой. Следовательно, для пункта *б* ответ: 4.

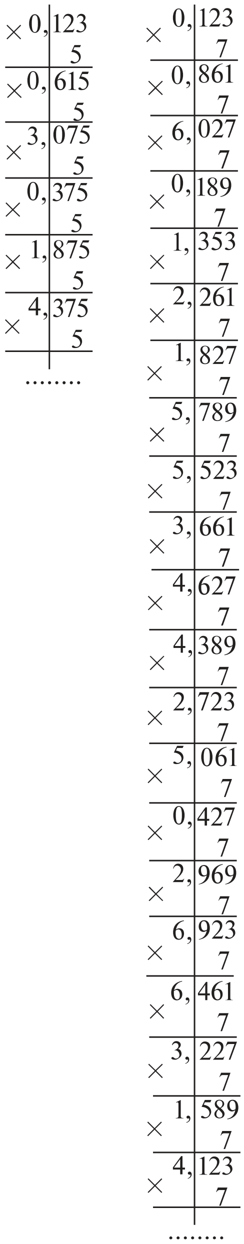


Рис. 2.1

Правило округления дробей в b-ичной системе счисления учащиеся, как правило, способны открыть и сформулировать самостоятельно, хотя нередко, пытаясь сделать это с ходу, учащиеся формулируют стандартное правило, в котором границей, определяющей округление в бóльшую или меньшую сторону, выступает цифра 5. Как правило, в этом случае бывает достаточно предложить им вспомнить определение округления чисел и подумать, какую роль при этом играет цифра 5 в десятичной системе счисления.

Задания 2 и 3, закрепляющие навыки перевода и округления, мы советуем выполнить частично в классе (а оставшиеся пункты упражнений задать на дом) сразу после рассмотрения алгоритмов перевода и правила округления. Вот как может выглядеть выполнение задания 2 для первого и второго из приведенных там чисел:

$$86{,}6_9 = 8 \cdot 9 + 6 + \frac{6}{9} = 78\,\frac{2}{3} = 78{,}(6);$$

$$12{,}87_{11} = 1 \cdot 11 + 2 + \frac{8}{11} + \frac{7}{121} = 13\,\frac{95}{121} = 13{,}(7851239669421487603305).$$

Для остальных чисел мы приведем только ответы:

$$452{,}96_{12} = 638\frac{19}{24} = 638{,}791(6);$$

$54{,}23_7 = 39\frac{17}{49} =$

$= 39{,}(346938775510204081632653061224489795918367)$.

Разумеется, этот 42-значный период получен нами не вручную, а с помощью калькулятора. С учащимися полезно обсудить, как все-таки вычислить такой период, не умещающийся на табло калькулятора.

$74{,}263_9 = 67\frac{219}{729} = 67{,}(300411522633744855967078189)$.

Перед выполнением задания 3 надо обратить внимание учащихся, что при переводе смешанных чисел отдельно осуществляется перевод целой части числа и отдельно его дробной части. Мы рекомендуем первый, второй и пятый примеры выполнить в классе, а два оставшихся выполнить дома. Для контроля приведем ответы к заданию 3:

$86{,}6 = 321{,}3_5$;
$12{,}87 = 22{,}41(3)_5 \approx 22{,}413334_5$;
$40{,}96 = 130{,}44_5$;
$54{,}23 = 204{,}10(3)_5 \approx 204{,}103334_5$;
$74{,}268 = 244{,}113(2)_5 \approx 244{,}113222_5$.

Задания 4 и 5 могут быть предложены как в конце первого урока, посвященного материалу § 12, так и в начале второго урока в качестве повторения и введения в рассмотрение двоичного представления дробей.

Ответы к заданию 4:
$10{,}01_2 = 2{,}25$;
$1010{,}1_2 = 10{,}5$;
$11{,}1001_2 = 3{,}5625$;
$1011{,}1101_2 = 11{,}8125$;
$10111{,}101_2 = 23{,}625$.

Возможно, кто-либо из учащихся заметит, что последнее число, записанное в двоичной системе, получено из предпоследнего сдвигом запятой на один разряд вправо. Это означает, что оно в 2 раза больше предыдущего. А тогда и десятичную запись легко получить из записи предыдущего числа, умножив это число на 2.

Приведем ответы к заданию 5:
$2{,}5_{16} = 2{,}3125$;
$4{,}F_{16} = 4{,}9375$;
$1A{,}7_{16} = 26{,}4375$;
$A{,}BC_{16} = 10{,}734375$;
$D{,}1AE_{16} = 13{,}10498046875$;
$D1{,}AE_{16} = 209{,}6796875$;
$FF{,}FF_{16} = 255{,}99609375$.

Обратите внимание учащихся, что при переводе двоичных и шестнадцатеричных дробей всегда получались конечные десятичные дроби. Обнаруженный эффект обсуждается в задании 6. Ответ на оба вопроса этого задания утвердительный, поскольку при записи обыкновенной дробью всегда получается знаменатель, представляющий собой степень числа 2. В курсе математики 6 класса учащимся объясняют, что если в знаменателе обыкновенной дроби нет простых множителей, отличных от 2 и 5, то такая дробь всегда записывается конечной десятичной дробью.

Алгоритмы перевода двоичной дроби в шестнадцатеричную и обратно обосновываются точно так же, как и для натуральных чисел. Ошибка, которую нередко допускают учащиеся при переводе дробной части числа из двоичной системы в шестнадцатеричную, состоит в том, что учащиеся забывают дополнить последний набор цифр до полной тетрады.

Повторению правил таких преобразований посвящено задание 1а. Выполняя пункт *б* того же задания, учащиеся должны самостоятельно сформулировать соответствующие правила для восьмеричной системы счисления: при переводе из восьмеричной системы в двоичную каждая цифра заменяется ее двоичным кодом; при обратном переводе дробная часть разбивается на триады, начиная от запятой вправо, и каждая триада заменяется восьмеричной цифрой. На закрепление навыков перевода направлены задания 7 и 8.

Приведем ответы к заданию 7:

$11{,}1001_2 = 3{,}9_{16}$;

$1011{,}1101_2 = B{,}D_{16}$;

$100{,}1_2 = 4{,}8_{16}$;

$10{,}101_2 = 2{,}A_{16}$;

$10111{,}101_2 = 17{,}A_{16}$.

Вот ответы к заданию 8:

$2{,}5_{16} = 10{,}0101_2$;

$4{,}7_{16} = 100{,}0111_2$;

$1{,}A7_{16} = 1{,}10100111_2$;

$A{,}BC_{16} = 1010{,}101111_2$;

$D{,}2BE_{16} = 1101{,}00101011111_2$;

$D2{,}BE_{16} = 11010010{,}1011111_2$;

$FF{,}FF_{16} = 11111111{,}11111111_2$.

Полезно обратить внимание учащихся на примеры

$D{,}2BE_{16} = = 1101{,}00101011111_2$

и $D2{,}BE_{16} = 11010010{,}1011111_2$ —

из них видно, что в шестнадцатеричном числе перенос запятой на один разряд вправо соответствует в двоичной записи того же числа переносу запятой вправо уже на 4 разряда. Учащиеся должны уметь объяснить такое соответствие и сформулировать общее правило.

Перевод десятичных дробей в двоичную систему счисления есть и в заданиях ЕГЭ. Вот пример такого задания.

Задание 17. Сколько единиц в двоичной записи десятичного числа 194,5?

Ответ: 4. Самый быстрый способ перевода числа 194 в двоичную систему заключается, на наш взгляд, в переводе этого числа алгоритмом деления в шестнадцатеричную или восьмеричную систему, а затем в расписывании его в двоичную. Поскольку $194 = C2_{16}$, то $194 = 11000010_2$, а $194{,}5 = 11000010{,}1_2$, откуда и получается требуемый ответ.

Приведем также ответы к заданию 9 из § 12: а) 1001,1; б) 110,01; в) 110001 и 110111,001.

Для закрепления навыков перевода из двоичной системы счисления в восьмеричную и шестнадцатеричную и обратно можно предложить учащимся следующее задание:

Задание 18. а) Переведите из шестнадцатеричной системы в восьмеричную числа 25; 4F; 1A7; ABC; D1AE; FF,FF; D1,AE.

б) Переведите из восьмеричной системы в шестнадцатеричную числа 25; 47; 157; 751; 4567; 777,7; 45,67.

Решение основывается на том, чтобы каждое из чисел сначала записать в двоичной системе счисления, а затем сформировать ответ в нужной системе счисления. Например:

а) $25_{16} = 100101_2 = 45_8$;

$D1{,}AE_{16} = 11010001{,}10101110_2 = 321{,}534_8$;

б) $25_8 = 10101_2 = 15_{16}$;

$45{,}67_8 = 100101{,}110111_2 = 25{,}DC_{16}$.

Третий час, отведенный в тематическом планировании на занятие в компьютерном классе, мы рекомендуем посвятить использованию *Инженерного калькулятора* для работы с дробными числами в различных системах счисления.

Для начала можно предложить учащимся вычислить с помощью *Инженерного калькулятора* произведение $101{,}01 \cdot 1010{,}1$, о котором идет речь в задании 9 из § 12. Обнаружив, что в двоичной системе счисления калькулятор воспринимает только целые числа, учащиеся вынуждены искать обходной путь для решения этой задачи. Довольно часто они предлагают каждый множитель перевести в десятичную систему, перемножить получившиеся числа и затем результат снова перевести в двоичную систему. Правда, сам перевод туда и обратно они предполагают выполнить вручную. Все это свидетельствует о сугубо формальном знании позиционного принципа в записи чисел. Учащихся в этом случае следует попросить сформулировать правило умножения десятичных дробей. Напомним эту формулировку: «Чтобы перемножить две десятичные дроби, их надо перемножить как натуральные числа (т. е. не обращая внимания на запятую), а затем в полученном результате отделить после запятой столько десятичных знаков, сколько их в обоих сомножителях вместе»[1]. Нетрудно обосновать, что это правило не зависит от того, в какой системе счисления — десятичной, двоичной, шестнадцатеричной или семеричной — выполняется умножение дробей. Тем самым, чтобы найти предложенное произведение, надо на калькуляторе вычислить произведение целых чисел, полученных простым игнорированием запятой, а затем в полученном результате отделить запятой столько разрядов справа, сколько их суммарно было отделено в обоих множителях. После этого обсуждения учащимся можно предложить найти несколько произведений для чисел в восьмеричной и шестнадцатеричной системах счисления.

Следующий блок заданий относится к переводу дробных чисел из двоичной, восьмеричной и шестнадцатеричной систем счисления в десятичную. Здесь учащиеся должны сформулировать правило, что перемещение запятой на один разряд вправо для числа, записанного в двоичной (восьмеричной, шестнадцатеричной) системе, равносильно его умножению на 2 (соответственно 8 или 16). Тем самым для перевода двоичной дроби в десятичную систему счисления нужно перевести это число как натуральное (т. е. не обращая внимания на запятую), а затем полученное десятичное число разделить на степень числа 2, показатель которой равен количеству знаков после запятой в исходном числе. Например, для числа $110101{,}101_2$ имеем $110101101_2 = 429$, поэтому $110101{,}101_2 = 429/8 = 53{,}625$. Правила для восьмеричных и шестнадцатеричных дробей формулируются аналогично. Подбор примеров для упражнений и определение необходимого их количества мы оставляем учителю.

Учитывая значительное внимание в ЕГЭ к теме «Системы счисления», мы рекомендуем провести контрольную работу.

Рассмотренные нами вопросы далеко не покрывают всего, что сегодня связано с системами счисления, к примеру, совсем не обязательно числа представлять как сумму степеней какого-либо числа, взятых с положительными коэффициентами. Впрочем, вариант, когда в качестве цифр выступают не только натуральные, но и отрицательные числа, весьма активно используется для решения различных теоретических и практических задач. К примеру, в позиционной системе счисления с нечетным основанием b в качестве цифр могут использоваться целые числа в диапазоне от $-\frac{b-1}{2}$ до $\frac{b-1}{2}$. Такая система счисления называется **уравновешенной**. В уравновешенных системах счисления в записи «отрица-

[1] Цитируется по учебнику: Математика, 5 / Л. Н. Шеврин, А. Г. Гейн, И. О. Коряков, М. В. Волков. — М.: Просвещение, 2000.

тельных цифр» вместо знака «–» обычно пишут черту над цифрой. Так, в троичной уравновешенной системе цифрами являются $\bar{1}$, 0 и 1, а в пятеричной — $\bar{2}$, $\bar{1}$, 0, 1 и 2. Приведем таблицу перевода целых чисел от –10 до 10 в уравновешенные троичную и пятеричную системы счисления.

Таблица 2.2

Десятичная система	Троичная система	Пятеричная система	Десятичная система	Троичная система	Пятеричная система
1	1	1	–1	$\bar{1}$	$\bar{1}$
2	$1\bar{1}$	2	–2	$\bar{1}1$	$\bar{2}$
3	10	$1\bar{2}$	–3	$\bar{1}0$	$\bar{1}2$
4	11	$1\bar{1}$	–4	$\bar{1}\bar{1}$	$\bar{1}1$
5	$1\bar{1}\bar{1}$	10	–5	$\bar{1}11$	$\bar{1}0$
6	$1\bar{1}0$	11	–6	$\bar{1}10$	$\bar{1}\bar{1}$
7	$1\bar{1}1$	12	–7	$\bar{1}1\bar{1}$	$\bar{1}\bar{2}$
8	$100\bar{1}$	$2\bar{2}$	–8	$\bar{1}001$	$\bar{2}2$
9	1000	$2\bar{1}$	–9	$\bar{1}000$	$\bar{2}1$
10	1001	20	–10	$\bar{1}00\bar{1}$	$\bar{2}0$

Алгоритмы перевода из десятичной системы счисления в уравновешенную систему с другим основанием и обратно остаются теми же самыми; надо только помнить, что при делении каждый остаток теперь заключен не от 0 до $b - 1$, а от $-\dfrac{b-1}{2}$ до $\dfrac{b-1}{2}$. Вот примеры такого перевода.

а) Переведите число 793 из десятичной системы в уравновешенную пятеричную систему.

Решение:

$$
\begin{array}{llllll}
-793 & \underline{|\,5\quad\;} & & & & \\
\;\;\underline{5\;\;\;} & -153 & \underline{|\,5\;} & & & \\
-29 & \;\;\underline{15\;} & -31 & \underline{|\,5} & & \\
\;\;\underline{25\;} & \;\;-3 & \;\;\underline{30} & -6 & \underline{|\,5} & \\
\;\;-43 & \;\;\;\;\underline{5} & \;\;\;1 & \;\;\underline{5} & \;1 & \\
\;\;\;\;\underline{45} & \;\;\;\;\bar{2} & & \;\;1 & & \\
\;\;\;\;\;\bar{2} & & & & &
\end{array}
$$

Ответ: $793_{10} = 111\bar{2}\bar{2}_5$.

б) Переведите число $1\bar{2}10\bar{1}1$ из пятеричной уравновешенной системы в десятичную систему.

Решение:

1	$\bar{2}$	1	0	$\bar{1}$	1
1	3	16	80	399	404

Ответ: $1\bar{2}10\bar{1}1_5 = 404_{10}$.

Вот только два, но весьма существенных преимущества уравновешенной системы счисления. Во-первых, в обычных системах счисления для записи отрицательных чисел приходится вводить дополнительный символ алфавита — знак «–». В уравновешенной системе этого делать не нужно (см., в частности, таблицу 2.2). Во-вторых, в обычной системе счисления, кроме алгоритма сложения, приходится специально разрабатывать алгоритм вычитания двух натуральных чисел (или сложения чисел с разными знаками); в уравновешенной системе этого делать не нужно. Эти и другие соображения, показывающие более экономный характер уравновешенных систем, привели к созданию в 1958 г. компьютера «Сетунь», работающего в троичной уравновешенной системе. Он был разработан группой научных сотрудников МГУ им. М. В. Ломоносова под руководством Н. П. Брусенцова и назван по имени речки, протекающей недалеко от Московского университета.

При желании предложить учащимся какие-либо задания, связанные с уравновешенными системами счисления, учитель может воспользоваться книгой «Задачник-практикум по информатике и информационным технологиям» А. Г. Гейна, Н. А. Юнерман (М.: Просвещение, 2003).

Тема 6. Кодирование символьной информации. Кодирование изображений

Общие представления о кодировании символьной информации учащиеся получили еще в базовом курсе информатики 8—9 классов. В качестве повторения этот материал был довольно основательно представлен и в курсе 10 класса. В книге для учителя с рекомендациями по преподаванию информатики в 10 классе содержатся основные методические установки по этой теме. Однако понятие кодовой таблицы и описание их видов в учебнике 10 класса не вводилось. Именно знакомству с кодовой таблицей ASCII и некоторыми ее наиболее популярными расширениями посвящен § 13. Важно, чтобы учащиеся знали, что первые 32 символа таблицы ASCII являются управляющими кодами. Разумеется, не следует требовать от учащихся знания кодов каких-либо символов.

В задании 1 к § 13 ответ очевиден — от 0 до 127. Учащиеся иногда ошибаются, считая, что нумерация начинается с 1. В задании 2 ответ отрицательный, поскольку коды десятичных цифр располагаются в основной части таблицы ASCII.

Приведем ответы к заданию 3 (для удобства коды букв мы отделили один от другого пробелами, учащиеся могут дать ответ в виде текста без пробелов): в CP-866 — 141 168

170 174 163 164 160 032 173 165 032 163 174 162 174 224 168 023 173 168 170 174 163 164 160;
в CP-1251 — 205 232 234 238 227 228 224 032 237 229 032 227 238 226 238 240 232 032 237 232 234 238 227 228 224;
в КОИ-8 — 238 201 203 207 199 196 193 032 206 197 032 199 207 215 207 210 201 032 206 201 203 207 199 196 193.

Для выполнения задания 4 от учащихся требуется некоторая наблюдательность. К примеру, в задании 4а встречается символ «±», который присутствует только в таблицах CP-866 и CP-1251, т. е. только одна из них могла быть кодовой при приеме сообщения. Однако в таблице CP-1251 этому символу соответствует код 177, который не кодирует букву ни в какой из этих трех таблиц. Тем самым код приема сообщения — CP-866, а код символа «±» равен 241. Этого достаточно, чтобы написать код исходного сообщения: 205 224 241 242 243 239 232 235 224 032 238 241 229 237 252. Естественно ожидать, что сообщение начинается с заглавной буквы, поэтому код 205 с большей вероятностью соответствует букве «Н» (таблица CP-1251), чем букве «м» (таблица КОИ-8). Используя таблицу CP-1251, получаем сообщение «Наступила осень».

Для задания 4б нетрудно понять, что представленная в сообщении замена прописных букв на заглавные могла происходить только при использовании таблиц CP-1251 и КОИ-8. Нужно только разобраться, в каком порядке они применялись. Если сначала применялась таблица CP-1251, а затем таблица КОИ-8, то буква «с» получилась из буквы «У»; если же сначала применялась таблица КОИ-8, а затем таблица CP-1251, то буква «с» получилась из буквы «Я». И то и другое вполне возможно. Буква «Ъ» могла получиться из «я» (при первом варианте) или из «з». Дальнейшее рассмотрение оформим в виде таблицы (пробелы опущены, а слова разделены двойной линией).

Таблица 2.3

CP-1251	с	Ъ	О	Б	А
CP-1251 → КОИ-8	У	я	и	в	б
КОИ-8 → CP-1251	Я	з	н	а	ю

Уже второе слово однозначно указывает на порядок кодирования. Продолжая действовать по второму варианту, получаем сообщение «Я знаю информатику на отлично».

В задании 4в сразу видно, что вторая кодировка не велась с помощью таблицы CP-1251. Однако если на втором шаге использовалась таблица CP-866, то букве «Б» (второй символ в сообщении) соответствует код 129, которому не соответствует буква русского алфавита ни в какой из двух других кодировочных таблиц. Тем самым на втором шаге использовалась таблица КОИ-8. Вернемся теперь к коду первого символа. Даже если неясно (ввиду полиграфической нечеткости), какой из символов 144—146 здесь представлен, ни одному из них в таблице СР-1251 не соответствует буква русского алфавита. Следовательно, кодировка происходила по схеме СР-866 → КОИ-8. Вот декодированное сообщение: «Столицей России является Москва».

В задании 4г на втором шаге использовалась кодировочная таблица СР-1251. Получаем код исходного сообщения: 141 165 032 167 163 160 239 032 161 224 174 164 160 044 032 173 165 032 225 227 169 225 239 032 162 032 162 174 164 227. Первый символ имеет код 141, а русская буква имеет такой код только в таблице СР-866. Получаем сообщение: «Не зная брода, не суйся в воду». Проблема здесь может возникнуть при декодировании буквы «а», поскольку ее код 160, а он в таблице СР-1251 дает символ пробел.

В задании 4д аналогичный анализ показывает, что не могла использоваться кодовая таблица СР-866. Как и для задания 4б, дальнейшее рассмотрение оформим в виде таблицы.

Таблица 2.4

СР-1251	о	П	Х	У	Н	Д	Х	б	Ю	М	Ъ
СР-1251 → КОИ-8	П	р	и	х	о	д	и	В	а	н	я
КОИ-8 → СР-1251	Н	о	у	с	м	д	ю	А	ч	л	з

Получаем исходное сообщение: «Приходи, Ваня, 1 апреля ко мне в гости».

Компьютерный практикум здесь запланирован только в рамках профильного курса. Это продиктовано дефицитом времени в базовом курсе. Мы советуем тем не менее в Microsoft Word через меню *Вставка/Символ* продемонстрировать, как кодируются различные символы (в том числе в Unicode). Это можно пристыковать к созданию текстовых объектов, которое рассматривается в начале главы 3. Если же речь идет о профильном курсе, то здесь полезно предложить усложненный вариант дополнительных заданий 12—14, о которых речь шла выше. Приведем еще одно дополнительное задание, предназначенное для дальнейшего развития навыков программирования.

Задание 19. Напишите программу, которая при нажатии одной произвольной клавиши (кроме управляющих клавиш) выводила бы сообщение «Это цифра», «Это буква» или «Это другой символ» (режим Caps Lock считается не включенным).

Это задание мы считаем несложным, и с него можно начать компьютерный практикум в этой теме.

С кодированием цвета, которому посвящены § 14—16, на начальном уровне учащиеся знакомы по базовому курсу 8—9 классов, поэтому учитель вполне может опереться на эти знания[1]. Однако рассмотрение теоретических оснований такого кодирования, скорее всего, отсутствовало. Учитель может прямо поставить перед учащимися вопрос: «Почему для представления любого цвета на экране компьютера достаточно использовать цветовые элементы только трех видов — красного, зеленого и желтого?» По существу, теоретическими основаниями для этого служат аддитивный принцип формирования цвета, а также законы трехмерности и непрерывности. Важно, чтобы учащиеся отчетливо понимали, что данные закономерности относятся к восприятию, а не физической сущности света (что весьма подробно обсуждается в учебнике).

Нужно, чтобы учащиеся также различали понятия «кодирование цвета» и «кодирование изображения». Конечно, кодирование цвета является составной частью процесса по кодированию изображения. Однако первым шагом в кодировании изображения является **дискретизация** изображения, т. е. разбиение его на мельчайшие составляющие. В зависимости от выбранной модели разбиения получают либо **растровое** представление изображения, либо **векторное**. В первом случае на изображение как бы накладывается сетка (состоящая обычно из прямоугольников), и внутри каждой ячейки этой сетки (т. е. прямоугольника) изображение считается однородным по цвету (яркости, оттенку, насыщенности). Во втором случае изображение разбивается на некоторые **примитивы** — так называют стандартные фигуры, описываемые, как правило, с помощью подходящих математических формул. Для каждого элемента, полученного в результате дискретизации, описание его характеристик — яркости, насыщенности, цветового оттенка — с помощью тех или иных числовых параметров представляет собой самостоятельную задачу. Дело в том, что такие характеристики — это, как правило, непрерывные величины. Чтобы убедиться в этом, достаточно обратиться к определениям указанных цветовых характеристик, приведенным в § 15. Но ограниченность разрядной сетки ячеек компьютера не позволяет работать с любым значением такой непрерывной характеристики. Поэтому весь диапазон изменения непрерывной величины разбивают на равные части, называемые **уровнями**. Каждому уровню присваивается номер в порядке возрастания величины. Сам процесс такого

[1] Полезно задать предварительное повторение этой темы по учебнику для 10 класса (§ 4).

разбиения называется **квантованием**. В этой главе речь идет именно о квантовании цвета, а особенности растровой и векторной графики рассматриваются в главе 3, начиная с § 34.

Перейдем к обсуждению заданий, помещенных в § 14. Ответ на вопрос задания 1 учащиеся должны знать назубок. Для школьников, изучающих английский язык, подсказкой служит сама аббревиатура RGB — красный (red), зеленый (green) и синий (blue). Важно, чтобы школьники знали, что именно в таком порядке цвета представлены в коде каждого пикселя. Это обстоятельство будет использоваться при выполнении задания 4.

Задание 2 направлено на проверку понимания связи кодирования цвета и координатного представления цветов. Вершине (0; 1; 1) соответствует цвет, полученный смешиванием синего и зеленого, из таблицы 2.8, помещенной в учебнике на с. 65, видно, что это голубой цвет. Непомеченная вершина имеет координаты (0; 1, 0), что соответствует зеленому цвету.

При выполнении задания 3 учащиеся должны вспомнить, что точки с пропорциональными координатами соответствуют одному оттенку цвета разной насыщенности. Поэтому точке с координатами (1/4; 1/4; 0) соответствует темно-коричневый цвет. В свою очередь, точке (3/4; 3/4; 0) соответствует ярко-коричневый (можно сказать, темно-пурпурный) цвет. Что касается пункта *б*, то многие школьники знают, что оранжевый цвет получается смешиванием красного и желтого. Можно считать, что точка, изображающая оранжевый цвет, является серединой ребра цветового куба, которое соединяет вершины, отвечающие за красный и желтый цвета. У этой точки координаты таковы: (1; 1/2, 0).

Для того чтобы выполнить задание 4, учащиеся прежде всего должны учесть особенность кодирования цвета в режиме Hi-Color: первые и последние 5 бит отводятся под кодирование красного и синего цветов, а 6 битов в середине — кодирование зеленого цвета. В пункте *а* мы получаем точку цветового куба с координатами (1; 0; 1). Это пурпур. В пункте *б* получаем точку с координатами (1/2; 1/2; 1/2). Это серый цвет. Наконец, в пункте *в* получаем точку с координатами (1; 1/2; 0). Используя результат задания 3б, делаем заключение, что это оранжевый цвет.

Задание 5 аналогично заданию 4. В пункте *а* записана точка с координатами (0; 1; 1). Это голубой цвет. В пункте *б* опять закодирован серый цвет. В пункте *в* закодирована точка с координатами (1/2; 0; 1/2). А это фиолетовый цвет.

Учителю надо иметь в виду, что представленная в учебнике модель лишь демонстрирует подходы к цветовому кодированию в общетеоретическом плане. Особенности восприятия человеком цветовых оттенков намного сложнее. Поэтому, к примеру, в 24-битной RGB-модели (которая

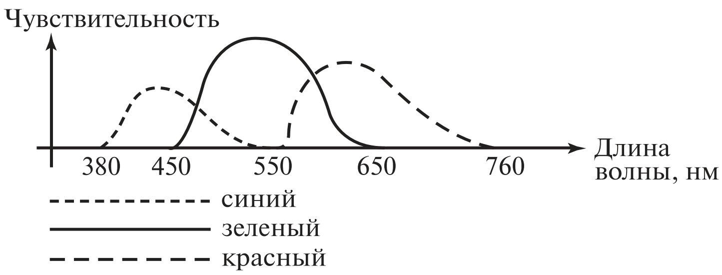


Рис. 2.2

реально используется как цветовой стандарт Интернета) коричневый цвет кодируется как 996600, что на цветовом кубе дает точку (154/256; 103/256; 0) $\approx$ (0,6; 0,4; 0). Это близко к (1/2; 1/2; 0), но все же не совпадает. Мы не исключаем, что дотошные школьники могут знать такие тонкости, и не хотели бы поставить учителя в неловкое положение. На самом деле такая неравномерность в цветовом кодировании (которая, в частности, проявляет себя и в том, что в режиме Hi-Color на кодирование зеленого цвета отводится больше битов, чем для двух других цветов) объясняется особенностями чувствительности человеческого глаза. На рисунке 2.2 представлены данные о чувствительности колбочек разного типа к световым волнам в зависимости от длины волны. Видно, что наибольшую чувствительность человеческий глаз имеет в области зеленого цвета. Это, в частности, означает, что и оттенки зеленого он различает лучше, чем оттенки синего и красного цветов.

В ЕГЭ-2008 впервые предлагалось задание на декодирование цвета. Вот это задание в формулировке, близкой к той, в какой оно присутствовало в одном из вариантов.

Задание 20. Для кодирования цвета фона страницы Интернета используется атрибут bgcolor="XXXXXX", где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом <body bgcolor="00DDDD">?

1) белый; 2) голубой; 3) красный; 4) черный.

Правильный ответ указан под номером 2. Действительно, здесь указана точка цветового куба с координатами (0; 239/256; 239/256), которая близка к точке (0; 1; 1), изображающей голубой цвет. Впрочем, учащимся можно посоветовать идти от противного: белый цвет кодируется FFFFFF, красный — FF0000 (или, по крайней мере, DD0000), а черный — 000000. Ни один из этих кодов не годится, так что остается ответ: голубой.

Знания о цветовой модели HSB, которой посвящен § 15, будут активно использоваться при изучении работы с программой Adobe Photoshop, описанной в главе 3. И хотя при освоении этого программного продукта учащиеся будут работать с разнообразными инструментами редактирования цветов, мы считаем полезным один час компьютерного практикума выделить уже сейчас для работы с компьютерной графикой, чтобы продемонстрировать, как в ней представлена данная модель цветопередачи.

Задания 1—3 к § 15 направлены на повторение основных понятий, относящихся к HSB-модели кодирования цвета. Задание 4 тоже вполне простое: коричневый цвет располагается под пурпурным цветом с меньшей яркостью, кодируется числом 300; оранжевый располагается между красным и желтым, кодируется числом 30; фиолетовый располагается между пурпурным и красным, кодируется числом 330. Учащиеся при выполнении этого задания могут использовать цветное изображение круга Манселла или обратиться к какому-либо графическому редактору, в котором используется HSB-кодирование.

В § 16 рассматриваются особенности воспроизведения цветов на бумаге, которая представляет собой светоотражающую поверхность. Вообще говоря, такое заявление относительно обычного листа бумаги может вызвать у школьников недоумение и даже недоверие. Ведь бумага не зеркало! Здесь можно пояснить (хотя учащиеся могут догадаться об этом и самостоятельно), что различие между зеркалом и бумагой лишь в уровне гладкости поверхности, а не отражательной способности. Если поверхность абсолютно гладкая, то в пучке параллельных лучей каждый луч по законам физики отразится под одним и тем же углом. А если поверхность шероховатая, то угол падения у каждого луча окажется свой и отраженные лучи уже не будут образовывать параллельный пучок, а потому изображение распадется. Тем самым, разглядывая изображение на бумаге, мы видим рисунок, создаваемый отраженными лучами. Какими будут эти отраженные лучи, зависит от типа краски. И об этом подробно рассказано в § 16. Во всех устройствах печати используются краски, играющие роль светофильтров. В целом при объяснении этого материала мы советуем следовать тексту учебника. Впрочем, при дефиците времени материал этого параграфа можно не изучать.

Ответы на вопросы заданий 1 и 2 из § 16 содержатся в объяснительном тексте параграфа. Следуя указаниям, представленным в заключительном абзаце объяснительного текста, достаточно легко выписать формулы, которые предложено получить в задании 3: если в RGB-модели цвет имеет код $(x; y; z)$, то в CMY-модели точка, соответствующая тому же цвету, имеет координаты $(1 - x; 1 - y; 1 - z)$, а значит, эти числа образуют и код этого цвета в CMY-модели. Эти формулы используются и при выполнении задания 4.

Тема 7. Кодирование с заданными свойствами

В этой теме два принципиально различных концентра: первый — коды, обнаруживающие и исправляющие ошибки; второй — экономность кодирования и алгоритмы сжатия. Они в определенном смысле антагонистичны. Обнаружение и исправление ошибок всегда достигается за счет построения избыточных кодов, в то время как экономичность кода требует избавления от избыточности.

О роли избыточности для защиты информации от возможных искажений упоминалось в § 3. Можно сказать, что в естественных языках избыточность присутствует как стихийно сложившаяся система защиты информации от шума. В § 17 речь идет о математических подходах к построению избыточного кодирования с целью обнаружения и даже исправления тех искажений, которые могли возникнуть по тем или иным (прежде всего техническим) причинам. Важно подчеркнуть, что в отличие от «стихийной» избыточности естественных языков математическая теория позволяет строить кодирование так, чтобы заданный уровень надежности (т. е., скажем, высокий процент обнаружения и исправления ошибок) обеспечивался способом кодирования, который был бы наиболее экономичным. Обсуждению такой постановки проблемы посвящена первая половина § 17.

Важнейшим инструментом решения данной проблемы является понятие **расстояние Хэмминга**. Применение слова «расстояние» в столь далекой от геометрии области, какой является кодирование, для учащихся может показаться странным и вызвать у них не только недоумение, но и некоторое отторжение. Здесь уместно, на наш взгляд, обсудить с учащимися вопрос: а что вообще следует понимать под словом «расстояние»? Ниже мы приводим возможный вариант такого обсуждения (отнюдь не призывая учителя следовать ему дословно).

«Может показаться странным, что указанную величину называют расстоянием. Мы привыкли под расстоянием понимать некоторую геометрическую величину, характеризующую удаленность объектов друг от друга. Впрочем, расстояния бывают разные. На плоскости под расстоянием обычно понимают длину отрезка, соединяющего две точки. Но для нас, людей, живущих на земном шаре, естественно расстоянием считать дугу большого круга, проходящего через две рассматриваемые точки (если мы, конечно, не собираемся рыть туннель). В городе расстояние между двумя точками мы будем измерять вдоль улиц, по которым можно добраться из одной точки в другую, и т. д. Математики выяснили, что общими для всех разновидностей расстояний являются три свойства. Вот какие.

Обозначим через $d(a, b)$ функцию, которая двум объектам (например, точкам или последовательностям символов) сопоставляет неотрицательное число, удовлетворяющее следующим условиям:

1) $d(a, b) = 0$ тогда и только тогда, когда $a = b$ (равенство здесь обозначает совпадение объектов);

2) $d(a, b) = d(b, a)$;

3) $d(a, b) \leqslant d(a, c) + d(c, b)$ каким бы ни был объект c.

Смысл первого свойства совершенно ясен; второе утверждает, что объект a удален от объекта b так же, как объект b удален от объекта a; наконец, третье свойство говорит, что дорога через третий объект c всегда длиннее, нежели прямой путь. Третье свойство обычно называют неравенством треугольника за его естественную геометрическую аналогию: сумма двух сторон треугольника всегда больше третьей стороны.

Математики договорились любую функцию, обладающую указанными тремя свойствами, называть **расстоянием**. То, что расстояние Хэмминга обладает свойствами (1) и (2), очевидно. Справедливость третьего свойства вытекает из такого рассуждения. Пусть слова a и b отличаются в некоторой позиции t. Тогда какое бы слово c мы ни взяли, оно в этой позиции будет отличаться по крайней мере от одного из слов a и b. Следовательно, суммируя в правой части $d(a, c)$ и $d(c, b)$, мы обязательно учтем все позиции, в которых различались слова a и b».

Последнее рассуждение в этом фрагменте — это решение задания 11 из § 17. В задании 12 предлагается обосновать утверждение, высказанное в объяснительном тексте без доказательства: поскольку расстояние между любыми двумя кодовыми словами не меньше 3, для каждого слова, содержащего одну ошибку, имеется ровно одно ближайшее к нему кодовое слово. В задании 13 предлагается доказать аналогичное утверждение для кода, у которого минимальное расстояние между кодовыми словами равно 7. Общее утверждение, охватывающее оба эти задания, выглядит так:

Пусть с — минимальное из расстояний между любыми двумя различными кодовыми словами. Тогда для любого слова, содержащего не более (с – 1)/2 ошибок, существует единственное ближайшее к нему кодовое слово.

Доказательство. Поскольку всего кодовых слов конечное число, то для любого слова w, записанного в данном алфавите, найдется кодовое слово, отстоящее от него на минимальном расстоянии. Предположим, что слов на таком минимальном расстоянии оказалось не одно, а несколько. Пусть u и v — два слова, минимально отстоящие от w. Условие, что w содержит не более ошибок, означает, что расстояние от w до некоторого кодового слова (из которого оно получилось в результате ошибок) не превосходит $(c-1)/2$. Но тогда ввиду минимальности для u и v выполнены неравенства $d(w, u) \leqslant (c-1)/2$ и $d(w, v) \leqslant (c-1)/2$ и можно записать следующую цепочку неравенств:

$$c \leqslant d(u, v) \leqslant d(u, w) + d(w, v) = d(w, u) + d(w, v) \leqslant$$
$$\leqslant (c-1)/2 + (c-1)/2 = c - 1.$$

Однако неравенство $c \leqslant c - 1$ невозможно, так что допущение о том, что ближайшее кодовое слово не единственно, было неверным.

Именно то единственное кодовое слово, которое является ближайшим к ошибочному, берется в качестве исходного правильного слова. Поэтому говорят, что такой код автоматически гарантированно исправляет $[(c-1)/2]$ ошибок. Слово «гарантированно» здесь употреблено не случайно. Дело в том, что и в ситуации большего расстояния между ошибочным и ближайшим кодовыми словами такое ближайшее слово может оказаться единственным, и потому его также можно однозначно декодировать. Однако гарантировать такую однозначность для любых слов уже нельзя.

Величину c, т. е. минимальное из расстояний между любыми двумя различными кодовыми словами, называют **кодовым расстоянием** данного кода. Именно этот термин употреблен нами в задании 9.

Для произвольного кодирования задача вычисления кодового расстояния достаточно трудоемка. Даже для тех 10 кодовых слов, которые использованы для кодирования первых десяти натуральных чисел, нужно рассмотреть 45 пар кодовых слов. Это предложено сделать в задании 6 к § 17. Результат выполнения задания 6 представлен нами в таблице 2.5.

Таблица 2.5

0	0000000	0	1	2	3	4	5	6	7	8
1	0001111	4								
2	0010110	3	3							
3	0011001	3	3	4						
4	0100101	3	3	4	4					
5	0101010	3	3	4	4	4				
6	0110011	4	4	3	3	3	3			
7	0111100	4	4	3	3	3	3	4		
8	1000011	3	3	4	4	4	4	3	7	
9	1001100	3	3	4	4	4	4	7	3	4

Заполнена только та часть таблицы, которая располагается ниже диагонали, так как в силу свойства 2, выполненного для произвольного расстояния, клетки, расположенные выше диагонали, заполняются из соображений симметрии.

В математической теории кодирования разработаны методы, позволяющие строить коды, для которых кодовое расстояние находится легко. Вот один из таких подходов. Важную роль в нем играет так называемое побитовое сложение последовательностей по модулю 2. Оно заключается в следующем. Пусть есть две последовательности 0 и 1 (т. е. два слова из символов алфавита $\{0; 1\}$) одинаковой длины. Тогда мы складываем эти две последовательности поразрядно как двоичные числа без переноса единицы в следующий старший разряд. Такую операцию обозначают $\oplus$. Вот пример: $11011 \oplus 01101 = 10110$. В языках программирования эту операцию обозначают XOR, и мы еще не раз с ней встретимся в последующих главах. В теории кодирования эту операцию называют сложением. Она обладает стандартными свойствами:

1) $u \oplus v = v \oplus u$;
2) $(u \oplus v) \oplus w = u \oplus (v \oplus w)$;
3) $u \oplus 00...0 = u$.

Здесь u и v — слова одной длины, записанные только с помощью 0 и 1, а 00...0 — слово, содержащее столько символов, сколько их содержится в u. Слово 00...0 называют **нулевым** — ведь оно играет для данной операции ту же роль, что и обычное число 0 для обычного сложения. Одно свойство этой операции существенно отличает ее от обычного сложения чисел:

4) $u \oplus u = 00...0$.

Весом слова называют количество единиц в нем. Легко видеть, что расстояние между двумя словами равно весу их суммы — ведь единица в сумме будет стоять в точности в той позиции, где исходные слова имели разные символы.

Код называется **линейным**, если сумма любых двух кодовых слов снова является кодовым словом. По-другому такой код называют **групповым**. Поскольку сумма двух одинаковых слов дает нулевое слово (см. свойство 4), любой линейный код содержит нулевое слово. Кроме того, для линейного кода легко доказать следующее утверждение:

Кодовое расстояние линейного кода равно минимальному весу его ненулевых слов.

Действительно, поскольку расстояние между словами — это вес суммы слов, сумма кодовых слов — это снова кодовое слово, то расстояние между любыми двумя кодовыми словами совпадает с весом одного из кодовых слов. Поэтому минимальное расстояние между различными кодовыми словами не меньше минимального веса ненулевых кодовых слов. С другой стороны, вес любого кодового слова — это его расстояние от нулевого слова. Тем самым две указанные величины совпадают.

Код Хэмминга, рассмотренный в объяснительном тексте § 17, является линейным — его создатель сразу строил его как линейный код. Это соображение существенно облегчает выполнение задания 10. Достаточно просто найти попарные суммы уже имеющихся 10 кодовых слов. Важно не только найти недостающие 6 кодовых слов, но и убедиться, что расстояние между любыми двумя словами в таком 16-элементном коде не меньше 3. По определению расстояния пришлось бы проверить $16 \cdot 15/2$ пар. Даже если учесть, что для 45 пар такая проверка проведена при выполнении задания 6, остается проверить еще 75 пар. А вес надо будет подсчитать всего лишь у новых шести слов! Мы настоятельно советуем учителю уделить внимание понятию линейного кода.

Внимательный школьник может заметить, что набор кодовых слов, приведенный в задании 9, содержит те 10 слов, которые перечислены в объяснительном тексте как элементы кода Хэмминга. На самом деле представленный в этом задании код и является полным кодом Хэмминга, который требуется найти в задании 10. Проблема, однако, состоит в громоздкости выполнения пункта *а* задания 9, где, действуя по определению, нужно вычислить расстояние для 120 пар кодовых слов. Если же заметить, что представленный код является линейным, то решение задачи существенно облегчается.

Скептик может возразить, что для проверки линейности кода нам нужно убедиться, что сумма любых двух ненулевых кодовых слов снова является кодовым словом. Ненулевых слов 15, так что проверок придется осуществить $15 \cdot 14/2 = 105$. Что ж, математика и здесь приходит на помощь.

В линейном коде всегда можно выбрать несколько слов, таких, что все остальные слова данного кода получаются как суммы этих слов (необязательно только попарные). В случае кода Хэмминга такими словами можно взять 0001111 (в задании 9 это код буквы «А»), 0010110 (код буквы «Б»), 0100101 (код буквы «Г»), 1000011 (код буквы «З»). Тогда

$0001111 \oplus 0010110 = 0011001$ (код буквы «В»);
$0001111 \oplus 0100101 = 0101010$ (код буквы «Д»);
$0001111 \oplus 1000011 = 1001100$ (код буквы «И»);
$0010110 \oplus 0100101 = 0110011$ (код буквы «Е»);
$0010110 \oplus 1000011 = 1010101$ (код буквы «М»);
$0100101 \oplus 1000011 = 1100110$ (код буквы «Й»);
$0001111 \oplus 0010110 \oplus 0100101 = 0111100$ (код буквы «Ж»);
$0001111 \oplus 0010110 \oplus 1000011 = 1011010$ (код буквы «Л»);
$0001111 \oplus 0100101 \oplus 1000011 = 1101001$ (код буквы «Н»);

0010110 ⊕ 0100101 ⊕ 1000011 = 1110000 (код буквы «К»);

0001111 ⊕ 0010110 ⊕ 0100101 ⊕ 1000011 = 1111111 (код буквы «О»).

Видно, что здесь представлены все возможные суммы кодов букв «А», «Б», «Г» и «З». Если же складывать коды других букв, то это приведет к тому, что в сумме появятся одинаковые слагаемые. К примеру, сложим коды букв «Ж» и «Н» (возникающие одинаковые слагаемые одинаково подчеркнуты):

$0111100 \oplus 1101001 = (\underline{0001111} \oplus 0010110 \oplus 0100101) \oplus$
$\oplus (\underline{0001111} \oplus 0100101 \oplus 1000011) = (\underline{0001111} \oplus \underline{0001111}) \oplus$
$\oplus (0100101 \oplus 0100101) \oplus 0010110 \oplus 1000011 = 0000000 \oplus$
$\oplus 0000000 \oplus 0010110 \oplus 1000011 = 1100110$ — код буквы «М».

Здесь мы воспользовались всеми четырьмя свойствами операции ⊕. Поэтому при сложении кодов любых двух букв получится код уже имеющейся буквы. Это обосновывает линейность данного кода. Более того, никаких других кодов из выбранных нами четырех с помощью сложения получить уже не удастся.

Отметим еще, что никакое из четырех выбранных нами кодовых слов нельзя получить из трех других сложением. Это означает, что удаление любого из данных четырех слов приведет к тому, что уже не удастся с помощью сложения получить все слова данного кода. Такой набор кодовых слов, из которых сложением можно получить все кодовые слова, но ни одно слово нельзя удалить без утраты этого свойства, называется **базисом линейного кода**. Тем самым выбранные нами четыре кодовых слова образуют базис кода Хэмминга.

Пусть мы выбрали некоторый базис $w_1, w_2, \dots, w_n$ какого-либо линейного кода. Тогда все его кодовые слова получаются как суммы вида $\alpha_1 w_1 \oplus \alpha_2 w_2 \oplus \dots \oplus \alpha_n w_n$, где каждый из коэффициентов $\alpha_1, \alpha_2, \dots, \alpha_n$ равен 0 или 1 в зависимости от того, входит данное базовое слово в сумму или не входит. Заметим, что если хотя бы один коэффициент $\alpha_1, \alpha_2, \dots, \alpha_n$ отличен от 0, то указанная сумма не равна нулевому слову. Действительно, если $\alpha_1 w_1 \oplus \alpha_2 w_2 \oplus \dots \oplus \alpha_n w_n = 00\dots0$ и, к примеру, $\alpha_1 = 1$, то

$$w_1 = w_1 \oplus 00\dots0 = w_1 \oplus (w_1 \oplus \alpha_2 w_2 \oplus \dots \oplus \alpha_n w_n) =$$
$$= (w_1 \oplus w_1) \oplus \alpha_2 w_2 \oplus \dots \oplus \alpha_n w_n =$$
$$= 00\dots0 \oplus \alpha_2 w_2 \oplus \dots \oplus \alpha_n w_n =$$
$$= \alpha_2 w_2 \oplus \dots \oplus \alpha_n w_n,$$

а это означает, что слово w_1 представимо как сумма других слов из базиса, т. е. оно входить в состав базиса не может. То же самое справедливо для любого другого слова, перед которым коэффициент 1.

Из этого замечания вытекает, что для разных наборов коэффициентов $\alpha_1, \alpha_2, \dots, \alpha_n$ получающиеся кодовые слова обязательно будут различными. Действительно, если

$\alpha_1 w_1 \oplus \alpha_2 w_2 \oplus \ldots \oplus \alpha_n w_n = \beta_1 w_1 \oplus \beta_2 w_2 \oplus \ldots \oplus \beta_n w_n$, то $00\ldots0 = (\alpha_1 w_1 \oplus \alpha_2 w_2 \oplus \ldots \oplus \alpha_n w_n) \oplus (\beta_1 w_1 \oplus \beta_2 w_2 \oplus \ldots \oplus \beta_n w_n) = (\alpha_1 \oplus \beta_1)\, w_1 \oplus (\alpha_2 \oplus \beta_2)\, w_2 \oplus \ldots \oplus (\alpha_n \oplus \beta_n)\, w_n$, и потому $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2$, ..., $\alpha_n = \beta_n$. Следовательно, в коде с базисом w_1, w_2, ..., w_n кодовых слов будет столько, сколько различных наборов из 0 и 1 можно составить для коэффициентов α_1, α_2, ..., α_n, а таких наборов в точности 2^n. Итак, в любом линейном коде всегда содержится 2^n слов, где n — число элементов в базисе этого кода. В частности, отсюда вытекает, что в любых двух базисах линейного кода всегда содержится одинаковое число элементов, хотя базис линейного кода может быть выбран неоднозначно. Значит, в семибитовом коде Хэмминга ни в каком его базисе не может быть другое количество слов, нежели 4. Вопрос о том, как находить хоть какой-нибудь базис линейного кода, мы оставляем за рамками нашего рассмотрения.

Этот краткий экскурс в теорию линейных кодов не исчерпывает математического богатства данной темы. В теории кодирования возникают чарующие связи с многочленами и другими фундаментальными структурами высшей алгебры, на которых мы, разумеется, останавливаться не будем. Но и то, что нами рассказано, демонстрирует, что изучение данного материала может по-разному выстраиваться учителем как с содержательной, так и с методической точки зрения.

Продолжим обсуждение задания 9 из § 17. Пункту *а* мы уделили большое внимание. Что касается пункта *б*, то он носит чисто технический характер. Надо разбить приведенное там сообщение на фрагменты по 7 символов, для каждой семибитовой последовательности подобрать ближайшее к ней кодовое слово и затем заменить это кодовое слово той буквой, которую оно кодирует. Вот цепочка действий для получения ответа (исправленные цифры подчеркнуты):
0011011001111101110001100001111101101110110001000010001001101111011000111101110000000100010000111011000011111011 → 0011011 0011111 0111000 1100001 1111011 0111011 0001000 0100010 0110111 1011000 1111011 1000000 0100010 0001110 1100001 1111011 → → 00110$\underline{0}$1 00$\underline{0}$1111 0111$\underline{1}$00 110$\underline{1}$001 1111$\underline{1}$11 011$\underline{0}$011 000$\underline{0}$000 010$\underline{1}$010 0110$\underline{0}$11 10110$\underline{1}$0 1111$\underline{1}$11 $\underline{0}$000000 010$\underline{1}$010 000111$\underline{1}$ 110$\underline{1}$001 1111$\underline{1}$11 → ВАЖНОЕ ДЕЛО ДАНО

Остальные задания § 17 в большей степени направлены на закрепление знаний и умений, которые учащиеся осваивают в соответствии с теоретическим материалом, представленным в объяснительном тексте этого параграфа. В ответах на вопросы заданий 1 и 2 учащиеся должны дать соответствующие определения. Выполняя задание 3, некоторые учащиеся могут растеряться, поскольку все примеры, на которых рассматривалось понятие расстояния в тексте параграфа, приводились над двухсимвольным алфавитом {0; 1}. Надо обратить внимание учащихся, что определение расстояния между словами сформулировано для произвольного, а не только двухсимвольного алфавита. Ответы в задании 3 таковы: а) 4; б) 4; в) 2.

В задании 4 сформулирована обратная задача — по заданному расстоянию определить множество слов, отстоящих от данного слова на этом расстоянии. Учащиеся должны продемонстрировать понимание, что на расстоянии 1 находятся слова, отличающиеся от данного только в одной позиции. Поскольку в исходном слове 5 символов, то и слов на расстоянии 1 должно получиться 5. Вот эти слова: 00101; 11101; 10001; 10111; 10100. Это и есть ответ для пункта *а*. В пункте *б* слов уже будет 10, поскольку выбрать две позиции из пяти можно десятью способами[1]. Вот эти слова (позиции, в которых цифры изменялись, подчеркнуты): $\underline{01}101$; $\underline{0}0\underline{0}01$; $\underline{0}01\underline{1}1$; $\underline{0}010\underline{0}$; $1\underline{10}01$; $1\underline{1}1\underline{1}1$; $1\underline{1}10\underline{0}$; $10\underline{01}1$; $10\underline{0}0\underline{0}$; $101\underline{10}$.

Задание 5 демонстрирует учащимся, что количество слов, отстоящих на заданном расстоянии d от n-символьного слова, не зависит от того, какое слово было взято в качестве исходного. Это количество равно количеству способов, которыми можно выбрать d позиций из n, и имеет специальное название — число сочетаний из n по d. Оно обозначается C_n^d и равно $\frac{n!}{(n-d)!d!}$, где через $n!$ обозначено произведение всех натуральных чисел от 1 до n (читается «n факториал»).

Задание 6 обсуждено выше. В задании 7 в обоих пунктах декодирование производится по тому же правилу, что и в задании 9: приведенное сообщение надо разбить на фрагменты по 7 символов, для каждой семибитовой последовательности подобрать ближайшее к ней кодовое слово и затем заменить это кодовое слово той цифрой, которую оно кодирует. В пункте *а* получаем следующую цепочку преобразований (исправленные цифры подчеркнуты):

$0010111001000010000001000001 \rightarrow$
$\rightarrow 0010111\ 0010000\ 1000000\ 1000001 \rightarrow$
$\rightarrow 001011\underline{0}\ 00\underline{0}0000\ \underline{0}000000\ 10000\underline{1}1 \rightarrow 2008.$

В пункте *б* получается цепочка:

$1001101011010001011100001011 \rightarrow$
$\rightarrow 1001101\ 0110100\ 0101110\ 0001011 \rightarrow$
$\rightarrow 100110\underline{0}\ 011\underline{1}100\ 0101\underline{0}10\ 0001\underline{1}11 \rightarrow 9751.$

---

[1] Хотя комбинаторика — это раздел математики, а не информатики, мы считаем полезным, чтобы такие предварительные комбинаторные оценки учащимися производились. Это приучает их к поиску способов контроля получаемых результатов, что особенно актуально при программировании «многоходовых» задач.

В задании 8 реализован подход, рассматривавшийся в начале § 17, — для повышения помехоустойчивости кода каждое кодовое слово дублируется; в данном случае оно даже повторяется трижды: для a исходным кодом служит 00, для b — 01, для c — 10, для d — 11. Кодовое расстояние равно 3!, что легко проверяется рассмотрением всевозможных пар различных слов (таких пар 6). Алгоритм декодирования и здесь тот же — разбить сообщение на фрагменты по 6 символов, для каждой шестибитовой последовательности подобрать ближайшее к ней кодовое слово и затем заменить это кодовое слово буквой, которую оно кодирует: 110101101011001111110000 → 110101 101011 001111 110000 → → 010101 101010 111111 000000 → *bcda*. Надо обратить внимание учащихся на тот факт, что для первых двух символов в полученном сообщении исправляется одна ошибка, а для двух вторых — две ошибки. Нередко, увидев слово 001111, учащиеся заявляют, что его исправить нельзя, код с кодовым расстоянием 3 исправляет только одну ошибку. Надо еще раз объяснить учащимся, что в данном утверждении речь идет о *гарантированном* исправлении ошибок, но реально может так случиться, что на заданном кодовом расстоянии от заданного слова все равно находится только одно кодовое слово, и тогда оно принимается в качестве правильного. Именно такова ситуация в данном примере — от третьего и четвертого слов в сообщении на расстоянии 2 находится ровно по одному кодовому слову.

Осталось обсудить задание 14. Цели этого задания таковы:

- продолжить воспитание образного восприятия абстрактного понятия расстояния [1];
- установить внутрипредметные связи между темами «Кодирование» и «Графы»;
- связать понятие расстояния между словами с понятием расстояния на графе.

В пункте *a* наиболее удачным нам представляется то изображение графа, которое дано на рисунке 2.3.

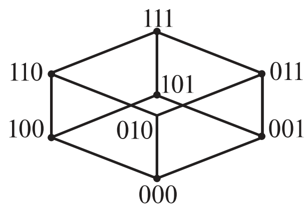


Рис. 2.3

[1] Надо иметь в виду, что учащиеся могут весьма сильно различаться по типу мышления. Для одних учащихся более легким является аналитический способ обработки информации, для других — использование образного представления. Надо стремиться к гармоничному сочетанию этих типов мышления, но не следует пренебрегать возможностью продемонстрировать оба. И тем более надо стараться, чтобы учащиеся с одним типом мышления не имели преимуществ перед учащимися с другим типом.

В теории графов расстоянием между двумя вершинами называют длину кратчайшего пути, ведущего из одной вершины в другую (длину одного ребра полагают равной 1). Это довольно естественное определение, и здесь можно не обсуждать с учащимися, почему такое определение не противоречит общему понятию расстояния, которое обсуждалось выше. Но мы считаем полезным обратить внимание учащихся на то, что расстояние между словами при указанном способе изображения их как вершин графа совпадает с расстоянием между соответствующими вершинами.

В пункте *б* предложение изобразить в виде графа все 128 семибитовых слов, конечно, не реализуемо. Поэтому мы предложили ограничиться лишь теми шестнадцатью словами, которые представлены в коде Хэмминга. Поскольку кодовое расстояние равно 3, то естественно интересоваться, какие именно слова находятся на таком расстоянии. Эти обстоятельства мотивируют постановку задачи, сформулированной в задаче 14б. Для решения задачи учащимся надо проявить определенную наблюдательность. Во-первых, каждое кодовое слово, кроме слов 0000000 и 1111111, имеет вес 3 или 4. Во-вторых, поскольку код является линейным, а расстояние между двумя словами равно весу суммы этих слов, то расстояние между любыми двумя словами данного кода может быть равно одному из трех чисел: 3, 4 или 7. В-третьих, для каждого кодового слова найдется в точности одно другое кодовое слово, сумма с которым равна 1111111. Значит, все кодовые слова разбиваются на такие пары слов, что слова в каждой паре расположены друг от друга на расстоянии 7. Наконец, четвертое наблюдение состоит в том, что сумма слов одного веса всегда дает слово, вес которого отличен от весов слагаемых. Это означает, что слова одного веса в нем не могут находиться на расстоянии 3 друг от друга. Поэтому все слова можно расположить в два ряда, как это показано на рисунке 2.4 (поскольку код Хэмминга фактически представлен в задании 9, мы вместо семибитовых последовательностей на этом рисунке каждое кодовое слово обозначили

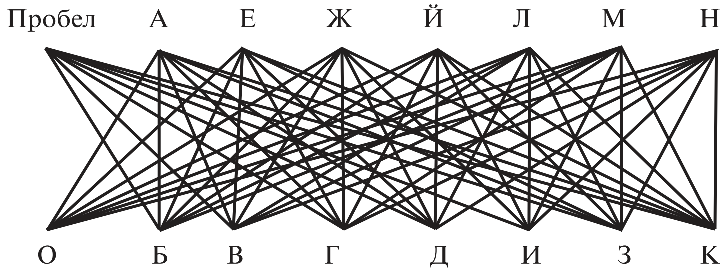


Рис. 2.4

буквой, которую оно кодирует). Прямые вычисления показывают, что каждое слово одного ряда отстоит на расстоянии 3 от семи слов другого ряда, это изображено на рисунке 2.4.

В зависимости от того, как глубоко предполагает преподаватель излагать теорию кодов, обнаруживающих и исправляющих ошибки, на эту тему может отводиться 2 или 3 урока. Компьютерный практикум можно проводить по сценарию, предложенному в учебнике. Мы только советуем, чтобы сообщения, закодированные одним учащимся, после внесения в них ошибок декодировались другим.

Материал § 18 начинает обсуждение другой важной темы теории кодирования — построение кодов, которые давали бы сообщения минимального информационного объема. Алгоритмы построения таких кодов лежат в основе всех программ-архиваторов, и обычно их называют алгоритмами сжатия информации. Здесь важно добиться понимания учащимися нескольких аспектов.

Прежде всего среди таких алгоритмов выделяются обратимые алгоритмы сжатия. Ими называют алгоритмы, которые преобразуют сообщение, закодированное некоторым кодом, в сообщение другой кодировки так, что, во-первых, можно полностью восстановить исходное сообщение и, во-вторых, информационный объем перекодированного сообщения будет меньше информационного объема исходного сообщения. Такое преобразование возможно исключительно ввиду избыточности кода, которым кодировалось исходное сообщение. Устраняя эту избыточность, мы получаем новый код, позволяющий достигнуть нужного эффекта.

Второй класс алгоритмов сжатия основан на избыточности самой информации в данном сообщении. В этом случае эффект достигается за счет того, что в перекодированном сообщении будет присутствовать не вся информация. Но потеря некоторой части информации не отразится существенно на понимании (восприятии) ее человеком. Такие подходы к построению алгоритмов сжатия близки к методам свертывания информации, которые обсуждались в главе 1. Только там речь шла об эвристических методах свертки информации, а здесь мы будем говорить о формальных приемах такой свертки. Надо также иметь в виду, что эти формальные методы применяются преимущественно к визуально воспринимаемой информации и опираются на особенности человеческого зрения, о которых речь шла в § 14 и 15.

Учащиеся должны понимать, что избыточность кода, которая позволяет строить обратимые алгоритмы сжатия, вовсе не является следствием небрежности или неумелости тех, кто такой код разрабатывал. Любой алгоритм сжатия построен с учетом особенностей того языка, который использовался для записи данного сообщения, и более того — с учетом особенностей структуры самого сообщения. Если, к примеру, мы точно знаем, что сообщения представляют собой фразы русского языка, то ясно, что в ASCII будут использоваться коды только заданного диапазона, в частности, наличие первой 1 в кодах символов русского алфавита избыточно, поскольку она присутствует всегда. Следовательно, ее незачем передавать. Таким образом, уменьшение информационного объема происходит за счет того, что мы располагаем дополнительной информацией о самом сообщении. И на самом деле так происходит всегда. Как мы получили дополнительную информацию (о том, что это русскоязычное сообщение), позволившую осуществить сжатие, — вопрос отдельный.

А вот другой взгляд на тот же вопрос, который, вообще говоря, служит идейной основой алгоритма Хаффмана. Хорошо известно, что буквы любого естественного языка встречаются в тексте с разной частотой. Поэтому, кодируя часто встречающиеся буквы более короткими последовательностями из 0 и 1, мы будем получать на достаточно длинных текстах весьма солидную экономию символов 0 и 1. Но именно знание частот появления символов языка — основа в теории измерения количества информации по К. Шеннону (об этом рассказывается в § 28 учебника для 10 класса). И здесь наличие дополнительной информации (о частотах появления букв в сообщениях, представленных с помощью данного языка) позволяет осуществить сжатие информации.

Если проанализировать эти ситуации, то становится ясным, что здесь мы имеем дело с подходом к измерению количества информации, предложенному академиком А. Н. Колмогоровым. Об этом подходе мы рассказывали в § 18 учебника для 10 класса. Напомним фразу из последнего абзаца этого параграфа: «На практике пользоваться определением количества информации по Колмогорову весьма затруднительно, в частности потому, что *нет способов нахождения оптимального алгоритма кодирования для любого наперед заданного сообщения*». Как видно из этой фразы, уже тогда мы объясняли, что вопросы измерения количества информации тесно увязаны с ее оптимальным кодированием. Мы советуем предложить учащимся перечитать этот параграф перед изучением материала, изложенного в § 18 учебника для 11 класса. Впрочем, для учащихся будет не лишним перечитать и § 28 из учебника для 10 класса, чтобы лучше понять, на каких общих закономерностях информатики основывается алгоритм Хаффмана.

Изложение в § 18 подходов к построению обратимых алгоритмов сжатия информации в идейном плане совпадает с тем, что мы только изложили. На наш взгляд, учитель вполне может следовать в этом вопросе тексту учебника, расставляя акценты в соответствии с вышесказанным. Некоторую трудность для учащихся может составить переход от кодов равномерной длины (когда все символы языка кодируются битовыми последовательностями одной длины) к кодам переменной длины. Здесь важно обратить внимание учащихся на существование способа однозначного разбиения последовательности из 0 и 1 на фрагменты, соответствующие кодам символов языка, на котором записано сообщение. В случае равномерного кода такое разбиение очевидно, в случае кодов переменной длины требуются дополнительные ухищрения. Чтобы учащиеся лучше осознали эту проблему, им можно предложить следующие два задания, схожие по формулировке, но весьма отличающиеся по сложности.

Задание 21. Каждая из букв А, К, О, Р, Т закодирована некоторой трехсимвольной последовательностью нулей и единиц. Известно, что слово ТРАКТОР закодировано строкой 101010100110101011010. Запишите двоичную строку, кодирующую: а) слово КРОТ; б) слово АРКА; в) слово КАРТА; г) слово РОТОР.

Задание 22. Каждая из букв А, К, О, Р, Т закодирована не более чем трехсимвольной последовательностью нулей и единиц. Известно, что слово ТРАКТОР закодировано строкой 111011000111001101. Запишите двоичную строку, кодирующую: а) слово КРОТ; б) слово АРКА; в) слово КАРТА; г) слово РОТОР.

Ответ в задании 21 получается совсем легко: после разрезания последовательности 101010100110101011010 на трехсимвольные фрагменты получаем, что Т кодируется как 101, Р — как 010, А — как 100, К — как 110, О — как 011. Соответственно получаем: КРОТ — 110010011101; АРКА — 100010110100; КАРТА — 110100010101100; РОТОР — 010011101011010.

В задании 22 для декодирования букв от учащихся требуются определенные рассуждения. Во-первых, поскольку длина сообщения равна 18, не могут все буквы кодироваться тремя битами (тогда сообщение содержало бы 21 символ), но они не могут быть все и двухбитовыми (тогда сообщение содержало бы 14 символов). Буква Р не может кодироваться одной цифрой, поскольку тогда сообщение содержало бы не более 17 битов. Аналогично и буква Т кодируется не менее чем двумя цифрами. Тем самым Р кодируется либо 01, либо 101. В свою очередь, сочетание ТР кодируется не менее чем четырьмя, но не более чем шестью символами и оканчивается на 01. Значит, ТР кодируется сочетанием 11101. Если Р кодируется как 01, то Т — как 111. Еще раз эта комбинация трех единиц встречается ровно один раз, поэтому получаем, что О кодируется как 0011, что противоречит условию. Следовательно, Р кодируется как 101, а Т — как 11. Тогда сочетание АКТО кодируется последовательностью 1000111001. Поскольку Т в этом фрагменте сообщения должно «содержаться» среди трех последовательных единиц, а О кодируется не более чем тремя битами, получаем, что О — это 001. Тогда АК кодируется последовательностью 10001. Но К не может кодироваться как 001, поскольку этот код занят буквой О. Следовательно, А — это 100, а К — это 01. Теперь уже несложно получить коды требуемых слов:

КРОТ — 0110100111; АРКА — 10010101100;
КАРТА — 0110010111100; РОТОР — 1010011100101.

Возможность такого однозначного декодирования в данном случае обеспечена специальным подбором сообщения, а вообще говоря, для кода переменной длины такая задача может решаться неоднозначно.

Впрочем, данное задание ближе к проблемам криптографии, нежели к вопросам кодирования сообщений заранее известным кодом (мы уже говорили о том, что следует различать цели, которые стоят перед теорией кодирования — возможность сохранения и передачи информации без искажения — и теорией шифрования, для которой главной задачей является защита от несанкционированного доступа и изменения информации). Чтобы учащиеся лучше осознали проблему декодирования при использовании кода переменной длины, полезно рассмотреть пример азбуки Морзе (ее русский вариант приведен в упражнении 5). Длина кодов для букв в этом варианте кодирования согласована с частотой употребления букв в русскоязычных текстах [1]. Поэтому данный код достаточно экономен, что, без сомнения, играло существенную роль в его признании как международного, ибо время, отводимое на связь, всегда было принципиально важным фактором. Однако его существенным недостатком была неоднозначность декодирования, если не пользоваться фактически еще одним символом (кроме всем известных точки и тире) — паузой между кодами символов. Можно для примера предложить учащимся декодировать сообщение, представленное в коде Морзе: — • — — — — —. Здесь есть по крайней мере два осмысленных варианта де-

[1] Слово «согласована» является наиболее подходящим в данной ситуации. Дело в том, что код Морзе создавался, естественно, для латинского алфавита и учитывал частоту употребления букв в английском языке. При его, как сказали бы теперь, локализации в русский сегмент за буквами, имеющими латинский аналог, оставили те же коды, и в целом это не сильно расходится с частотными характеристиками русских букв; остальным буквам коды были сопоставлены действительно в соответствии с частотой их употребления.

кодирования: КОТ и КТО. Ясно, что в такой ситуации автоматизировать процесс декодирования крайне сложно, в частности, все попытки самого С. Морзе предложить автоматическое устройство для такой работы оказались безуспешными. Придуманная немецким инженером Жан Морис Эмиль Бодо идея сделать код равномерной длины позволила автоматизировать процессы кодирования и декодирования сообщений, передаваемых в двухсимвольном алфавите [1]. Тем не менее задача отыскать способ кодирования, с одной стороны, учитывающий частоту появления букв в передаваемых текстах, а с другой — позволяющий построить алгоритм автоматического декодирования, оставалась актуальной. Фактически речь идет о том, чтобы такой алгоритм позволял однозначно определять конец кода для каждого символа алфавита, которым записано сообщение. Осознание такой постановки привело К. Шеннона и Р. М. Фано к мысли, что однозначность будет обеспечена, если никакое кодовое слово не служит началом другого кодового слова. Коды, удовлетворяющие сформулированному условию, называются **префиксными**. Скорее всего, если эта идея была бы высказана раньше на Арабском Востоке, то мы имели бы дело с суффиксными кодами.

После освоения понятия префиксного кода учащиеся должны научиться строить для каждого кода ориентированное дерево, представляющее данный код разметкой на своих дугах. Поскольку из каждой вершины, не являющейся листом, выходит ровно две дуги, то такое дерево обычно называют **бинарным**. С понятием дерева учащиеся будут подробно знакомиться, изучая материал главы 6, поэтому в § 18 мы пользуемся лишь термином «орграф». Однако учитель может, на наш взгляд, ввести термин «дерево» и активно пользоваться им, поскольку учащиеся, скорее всего, знакомы с ним по работе с различными иерархическими структурами организации данных (дерево папок или каталогов и т. п.).

Учащиеся обычно легко понимают главный признак префиксного кода: код является префиксным тогда и только тогда, когда все кодовые слова располагаются на концевых вершинах (листьях) кодового дерева.

На отработку умений строить дерево по заданному коду и определять с его помощью, является ли код префиксным,

[1] Надо иметь в виду, что Бодо в первую очередь изобрел именно код, можно сказать, что это было первое в истории чисто математическое изобретение. Телеграфный аппарат, в котором этот код использовался, был известен и до этого. Тем не менее часто учащиеся, слыша фразы типа «Передать по Бодо» или название картины «Ленин у аппарата Бодо», воспринимают Бодо как изобретателя телеграфного аппарата.

направлены задания 4 и 5. Если учитель сочтет необходимым, то можно предложить еще несколько аналогичных заданий.

Задание 23. а) Дан набор кодовых слов: 00, 10, 010, 101, 110, 1001, 1011, 1101, 1111. Постройте для этого кода соответствующий ему орграф. Является ли этот код префиксным?

б) Выполните такое же задание для кода 00, 01, 10, 011, 100, 101, 1001, 1010, 1111.

в) Выполните такое же задание для кода 00, 10, 010, 110, 0110, 0111, 1110, 1111.

Полезно предложить и «обратное» задание — построить код по бинарному дереву. Вот пример такого задания:

Задание 24. а) На рисунке 2.5 изображен орграф некоторого кода. Для каждой буквы запишите соответствующее ей кодовое слово. Является ли этот код префиксным?

б) «Перестройте» (т. е. не увеличивая количество вершин) граф так, чтобы для тех же восьми букв получился префиксный код. Запишите коды букв в этом случае.

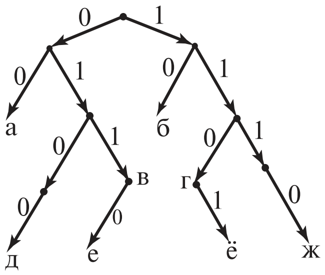
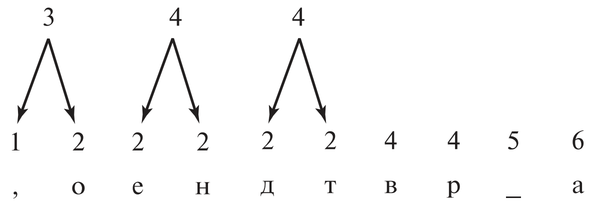


Рис. 2.5

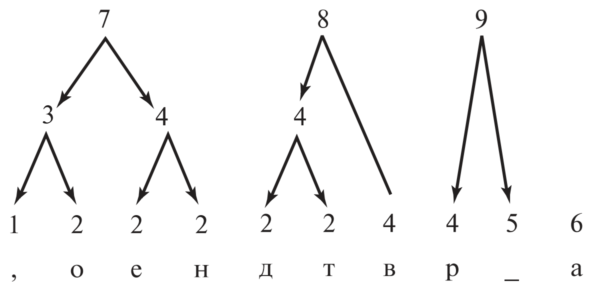
После освоения навыков построения префиксного кода можно перейти к рассмотрению алгоритма Хаффмана. Он достаточно подробно, на наш взгляд, описан и проиллюстрирован примером в учебнике, и мы предлагаем следовать этому изложению материала. В качестве фразы, на которой будет демонстрироваться применение алгоритма Хаффмана, может быть взята какая-либо другая фраза (например, упомянутая в тексте параграфа фраза «КОЛ ОКОЛО КОЛОКОЛА, А КОЛОКОЛ ОКОЛО КОЛА»; она удобна тем, что в ней использовано всего лишь 6 символов: «А», «К», «Л», «О», «пробел» и «запятая», и, следовательно, дерево кода будет не очень громоздким) либо та же самая поговорка о дровах на траве. Второй вариант нам кажется предпочтительным, поскольку в этом случае учитель сможет продемонстрировать, что дерево кода, а значит, и сам код строятся неоднозначно. При построении кодового дерева мы советуем располагать кодируемые символы в порядке возрастания (или убывания) частоты встречаемости их в тексте. В этом случае для фразы «на_дворе_трава,_на_траве_дрова» получится, например, такое расположение:

1	2	2	2	2	2	4	4	5	6
,	о	е	н	д	т	в	р	_	а

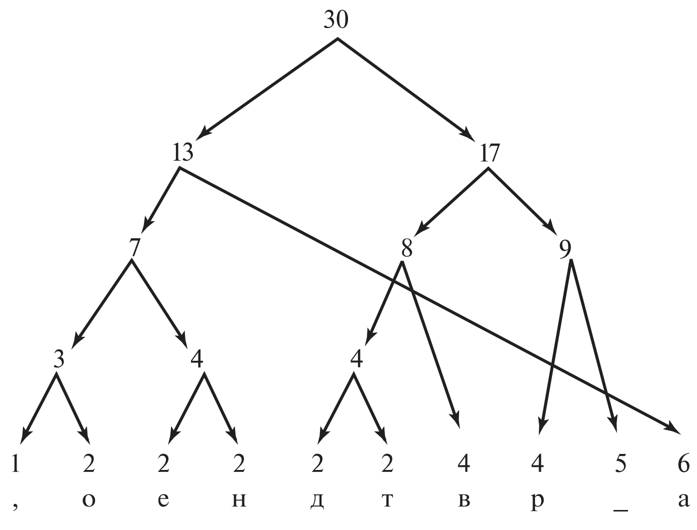
Тогда после первых трех шагов получится следующая картина:



А дальше процесс может развиваться так:



И наконец, получится так:



Дерево оказалось не таким красивым, как в учебнике, но это неважно. Важно, что не только коды самих символов оказались другими, другим стал сам набор кодов. Но коэффициент сжатия остался неизменным! И на это надо обязательно обратить внимание учащихся.

Мы не предлагаем никаких доказательств того факта, что алгоритм Хаффмана является оптимальным среди всех алгоритмов, для которых каждый исходный символ кодируется целым количеством битов, хотя это утверждение и сформулировано в конце объяснительного текста § 18.

Выше мы уже упомянули о заданиях 4 и 5 к данному параграфу. Задания 1 и 3 заставляют учащихся найти в тексте параграфа ответы на соответствующие вопросы. В задании 2 ответ тоже получить несложно. Латинских букв вместе с пробелом 27, значит, для их кодирования достаточно пятибитового кода. Если использовался восьмибитовый код ASCII, то коэффициент сжатия составит $8/5 = 1{,}6$. Если же использовался Unicode, то коэффициент сжатия увеличится вдвое.

При выполнении задания 6 у учащихся, вполне вероятно, будут построены разные кодовые деревья. Поэтому мы не будем приводить ни возможный вариант дерева, ни получающийся для него код. Укажем только, что коэффициент сжатия должен получиться равным $464/175 \approx 2{,}65$. Аналогично в задании 7 коэффициент сжатия будет равен $256/106 \approx 2{,}46$.

На освоение теоретического материала нами отводится 2 часа. В сильном классе можно предложить запрограммировать алгоритм Хаффмана, но для компьютерных экспериментов (вкупе с кодами, исправляющими ошибки, и экспериментами по архивации, описанными ниже) времени, отведенного на практикум, может не хватить. Недостающий час можно позаимствовать из предусмотренного нами резерва учителя.

Необратимые алгоритмы сжатия (§ 19) рассматриваются нами на примере алгоритма JPEG. Здесь снова надо обратить внимание учащихся, что в основе алгоритма лежит идея избыточности информации при восприятии ее человеком. Кроме того, как уже отмечалось при обсуждении кодирования цвета, человеческий глаз оттенки зеленого цвета различает лучше оттенков двух других базовых цветов. Этим объясняется, что в алгоритме сжатия JPEG полностью сохраняется информация о зеленом цвете, а информация о красном и синем цветах, во-первых, дается относительно опять-таки зеленого цвета и, во-вторых, сохраняется только об одном из четырех пикселей.

Математический аппарат, который лежит в основе восстановления исходной картинки после декодирования, едва ли доступен для понимания школьниками, поскольку опирается на современные теории приближения и сглаживания функций. Мы надеемся, что в тексте учебника нам удалось передать дух этих исследований; учащиеся должны почувствовать, что за всеми видеовозможностями современных компьютерных систем стоят сложные математические модели и методы, опирающиеся на самые современные достижения этой науки.

Ответы на вопросы заданий 1—3 фактически даны в объяснительном тексте. Задание 4 по своей сути близко к заданию, которое однажды предъявлялось в демоверсии ЕГЭ. И хотя ни на самом экзамене того года, ни в последующие годы подобные задания не встречались, мы сочли нужным представить его учащимся. Чтобы учащиеся дали на него ответ, мы предлагаем провести соответствующий компьютерный эксперимент, выбирая исходные данные различного объема и затем сжимая их с помощью того или иного архиватора.

Тема 8. Логические основы работы компьютера

Начало § 20 — это повторение того, что учащиеся, как правило, знают из базового курса информатики 8—9 классов. Поэтому мы считаем, что этот параграф может быть задан для самостоятельного изучения с последующим опросом на уроке. Впрочем, вполне возможна ситуация, что данный материал не изучался ими в базовом звене школьного образования, поскольку в Федеральный компонент стандарта по информатике он не входит. Тем не менее авторами большинства учебников он традиционно включается в этот курс (по крайней мере как материал для дополнительного чтения).

Независимо от того, будет ли данный материал выступать как повторение или как новый, хотим обратить внимание на ряд принципиальных моментов, содержащихся в нем. Во-первых, он затрагивает историю компьютерной техники, знание которой, хотя бы схематичное, весьма важно для понимания процессов компьютеризации и информатизации общества. В самом названии «компьютер» заложено первоначальное понимание этого агрегата как инструмента, облегчающего проведение вычислений. Сегодня в ряде публикаций (и даже в некоторых школьных учебниках) можно прочитать, что компьютеры *изначально* создавались для накапливания и обработки больших объемов информации. Это, конечно, не так. Во времена первых компьютеров (их тогда называли ЭВМ) ни о каком накапливании и обработке с их помощью информации речь идти не могла, поскольку быстродействие ЭВМ едва достигало 100 операций в секунду, оперативная память составляла около 20 Кбайт, а самый мощный внешний носитель информации представлял собой магнитную ленту объемом около 100 Кбайт. Осознание, что компьютерные системы — это не вычислительные, а информационные комплексы, приходило постепенно по мере роста мощности компьютеров по всем основным параметрам. При изучении данной темы полезно предложить кому-либо из учащихся подготовить и сделать на уроке сообщение по истории развития и *применения* вычислительной техники.

Во-вторых, изложение физических основ элементной базы компьютера, помимо представления исторического аспекта, убеждает учащихся в необходимости использования двоичного кодирования и основных логических операций: конъюнкции (операция И), дизъюнкции (операция ИЛИ) и отрицания (операция НЕ). Тем самым алгебра логики, изучавшаяся в 10 классе, выступает здесь в новом ракурсе.

Но самое главное — учащимся демонстрируется, что все, так сказать, «интеллектуальное поведение» компьютера основывается на весьма простых логических операциях. Происходит, как говорят, демистификация компьютера.

Физически каждая логическая операция реализуется соответствующим вентилем. В дальнейшем можно не вникать в устройство каждого вентиля, а использовать его как единый элемент. Именно так и сделано при объяснении работы полусумматора, находящего сумму двух однозначных чисел. В последующем полусумматор принимается как самостоятельный элемент и из полусумматоров конструируется сумматор одноразрядных и многозначных чисел.

Обсудим теперь выполнение заданий к § 20.

Ответ на вопрос 1 фактически дан в объяснительном тексте: использование двоичного кодирования продиктовано легкостью технической реализации двух состояний той или иной физической системы. Но в вопросе совсем не случайно употреблено слово «обычно». Дело в том, что с точки зрения экономичности и удобства организации вычислений более эффективной является уравновешенная троичная система счисления, о которой мы рассказывали в этой книге на с. 104.

Приведем ответы к заданию 2:

а) И

x	y	z
0	0	0
1	0	0
0	1	0
1	1	1

б) ИЛИ

x	y	z
0	0	0
1	0	1
0	1	1
1	1	1

НЕ

x	z
0	1
1	0

в) Глядя на таблицу умножения однозначных чисел в двоичной системе и сопоставляя ее с таблицей, заполненной в пункте *а*, легко догадаться, что двоичный мультипликатор однозначных чисел — это просто вентиль И. Ответ в задании 3 дается следующей таблицей:

x	y	z
0	0	1
1	0	0
0	1	0
1	1	1

Иными словами, $z = 1$ тогда и только тогда, когда значения x и y совпадают. Поэтому про данную схему обычно говорят, что она реализует операцию **сравнения** x и y.

Ответ на вопрос, сформулированный в задании 4, прост: сумматор учитывает разряд переноса из младшего разряда слагаемого, а полусумматор этого не делает.

Для решения задачи 5 нужно в первую очередь понять, как представлены исходные данные и результат вычислений. Каждый сомножитель представляет собой двузначное число и, следовательно, описывается двумя переменными: одна переменная отвечает за старший разряд, другая — за младший разряд числа. Тем самым на входе мультипликатора двузначных чисел четыре переменные: две переменные, скажем, x (старший разряд) и y (младший разряд), описывают один множитель, а две другие переменные, скажем, u (старший разряд) и v (младший разряд), описывают другой множитель. Сколько разрядов может иметь результат? Легко понять, что не больше четырех, причем четыре получается, когда 11_2 умножается на 11_2. Значит, для результата тоже надо иметь четыре переменные. Обозначим их через q, r, s и t, считая, что они перечислены по порядку от старшего к младшему (т. е. q — самый старший разряд результата, а t — самый младший). Правила умножения чисел столбиком показывают, что младший разряд t — это результат применения вентиля И к младшим разрядам сомножителей, т. е. y и v. Разряд s получается как младший разряд суммы двух произведений: x на v и y на u. А старший разряд этой суммы, сложенный с произведением x на u, дает разряд r. Наконец, старший разряд последней суммы — это

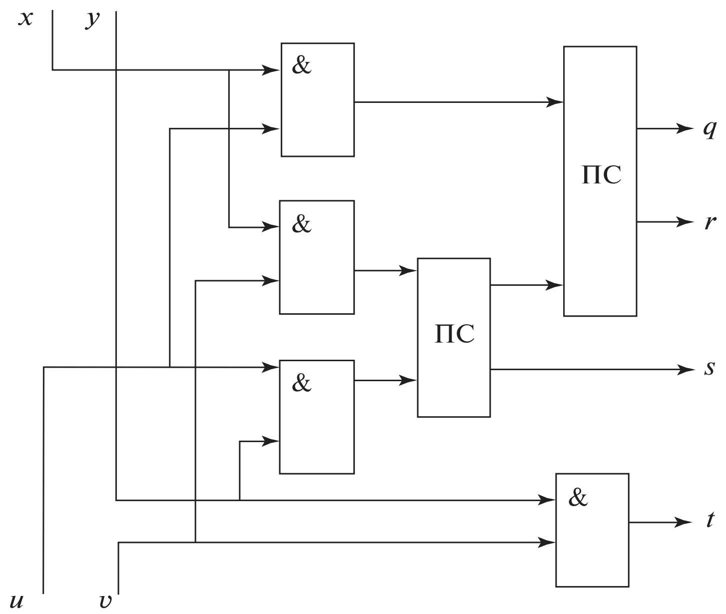


Рис. 2.6

и есть q. Получается схема, которая представлена на рисунке 2.6.

Мы рекомендуем предложить это задание на дом с последующей проверкой его выполнения в классе. После такой проверки естественно поставить перед учащимися проблему, как вообще разрабатывать устройства, которые бы выполняли заданные преобразования с исходными данными, представленными в двоичной кодировке. Это вводит учащихся в проблематику, которой посвящен § 21.

Нередко восприятие учащимися материала этого параграфа затруднено не вполне верными представлениями о понятии «функция», которые сформировались у них в силу того, что с этим понятием они имели дело только в рамках уроков математики, а на этих уроках, к сожалению, они имели дело только с числовыми функциями, да еще преимущественно заданными той или иной формулой. Поэтому на вопрос учителя «Что такое функция?» в лучшем случае прозвучит ответ, что это некоторая зависимость между переменными, а в худшем — что это формула, связывающая значения аргумента и функции. Необходимо преодолеть это крайне узкое (и, по существу, неверное) понимание термина «функция». Сделать это можно двумя путями. В случае если в 10 классе использовался наш учебник, можно предложить учащимся повторить материал, представленный в § 35 этого учебника. Разумеется, при опросе учащихся после такого повторения надо потребовать, чтобы они привели примеры разнообразных (а не только числовых) функций. Важно, что даже внутри самой математики рассматривается много самых разнообразных нечисловых функций. Вот только некоторые примеры:

1. Каждому треугольнику сопоставляется вписанная в него окружность (функция, областью определения которой служит множество треугольников, областью значений — множество окружностей).

2. Каждому отрезку сопоставляется его середина (функция, областью определения которой служит множество отрезков, областью значений — множество точек).

3. Каждому многоугольнику сопоставляется его площадь (функция, областью определения которой служит множество многоугольников, областью значений — множество положительных чисел).

4. Каждой паре, состоящей из положительного числа и точки на плоскости, сопоставляется окружность с центром в этой точке и радиусом, длина которого равна данному числу (функция, областью определения которой служит множество пар, составленных из чисел и точек плоскости, областью значений — множество окружностей на той же плоскости).

Последний пример иллюстрирует тот факт, что аргументов у функции может быть несколько (в данном случае их два: число и точка), и они могут быть, вообще говоря, разной природы.

Булевы функции — это функции, у которых аргументом служат последовательности из нулей и единиц фиксированной длины, а результатом может быть только 0 или 1. Но можно на булеву функцию смотреть и как на функцию n аргументов, каждый из которых также принимает только одно из двух значений: 0 или 1. С подобной ситуацией учащиеся, конечно, уже встречались, когда изучали элементы математической логики, только в той ситуации значение 0 кодировалось символом «Л» (и ему придавался смысл Ложь), а значение 1 кодировалось символом «И» (который интерпретировался как Истина). Разумеется, об этом надо напомнить учащимся и при необходимости предложить им перечитать § 30—32 из учебника для 10 класса.

Если же в 10 классе раздел математической логики не изучался, то можно напрямую следовать объяснительному тексту § 21, более подробно останавливаясь на понятии функции и операций над булевыми переменными. Надо при этом пояснить, что называть функцию от двух переменных операцией над ними — общепринятое соглашение. В этом случае функциональный символ превращается в символ операции и записывается не перед аргументами, а между ними. Впрочем, функцию от одной переменной тоже нередко называют операцией, а символ, обозначающий эту операцию, пишут либо над аргументом (иногда еще и правее), либо перед ним без скобок. Скажем, в алгебре числовых функций мы пишем $-x$, обозначая операцию вычисления числа, противоположного данному, и x^{-1}, обозначая операцию вычисления числа, обратного данному.

Обсуждая способы записи функции, мы коснулись понятия обратной польской записи. Это очень интересный раздел, относящийся к вопросам оптимальной организации вычислений арифметических выражений, однако углубляться в эти вопросы в рамках данной темы мы сочли нецелесообразным. Можно предложить одному-двум учащимся самостоятельно изучить этот вопрос и затем сделать соответствующее сообщение.

Чрезвычайно важной в идейном плане и в целом для воспитания математической культуры нам представляется аналогия между функцией и неким устройством, преобразующим значения аргументов в значение этой функции. Для иллюстрации этой аналогии годятся любые, даже самые бытовые примеры: функцию можно уподобить мясорубке, которая преобразует пищевой продукт из одного вида в другой, или духовке, которая также осуществляет преобразование пищевого продукта. Такая аналогия между функциями и устройствами была проведена фактически, когда вводились вентили И, ИЛИ и НЕ. Только там ситуация была в определенном смысле обратная — для некоторого устройства указывалась функция, которая моделирует его работу. Теперь же идет речь о математическом моделировании процесса соединения нескольких устройств в устройство с новыми продуктивными возможностями. Оказывается, что такому соединению на математическом языке соответствует понятие **композиции функций**. С понятием композиции функций учащиеся знакомы по курсу математики, однако практика показывает, что это понятие осваивается учащимися с большим трудом. На наш взгляд, это происходит именно потому, что оно вводится сугубо формально, за этим не стоит никакого образа физически реального мира. Аналогия между функцией и устройством-преобразователем позволяет создать такой образ. Как минимум, после этого учащиеся намного легче понимают, почему композиция функций дает разный результат при изменении порядка подстановки одной функции в другую. На том же бытовом уровне каждому ребенку, а не только ученику 11 класса понятно, что если сначала мясо измельчить, а затем запечь в духовке, то это будет не то же самое, что сначала его запечь в духовке, а затем измельчить.

После введения основных понятий, относящихся к булевым функциям, формулируется центральная проблема: каков тот минимальный набор элементов, из которых

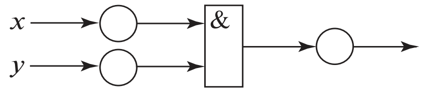


Рис. 2.7

можно собрать устройство, реализующее заданную булеву функцию? В § 21 обосновывается, что из функций &, $\vee$ и $\overline{}$ с помощью операции композиции можно сконструировать любую булеву функцию. Доказательство конструктивно — приводится алгоритм построения формулы для любой функции, которая задана таблицей своих значений. Учащиеся, изучавшие в 10 классе основы математической логики, этот алгоритм могут сформулировать вполне самостоятельно, он почти дословно совпадает с алгоритмом построения логической функции по таблице истинности, который приведен в § 32 учебника для 10 класса.

Но будет ли такой набор минимальным? Оказывается, нет. Можно, к примеру, вполне обойтись только конъюнкцией и отрицанием. Действительно, согласно законам де Моргана $x \vee y = \overline{\overline{x}\ \&\ \overline{y}}$, поэтому вентиль, реализующий операцию $\vee$, можно заменить комбинацией устройств, представленной на рисунке 2.7.

Используя второй закон де Моргана, можно показать, что конъюнкцию можно получить, используя только дизъюнкцию и отрицание. Множество функций, из которых с помощью операции композиции можно получить любую булеву функцию, называется **полным классом** булевых функций. Так что множества {&, $\vee$, $\overline{}$}, {&, $\overline{}$} и {$\vee$, $\overline{}$} являются полными классами, при этом два последних минимальны в том смысле, что удаление из множества любой одной функции уже не дает полного класса. Это достаточно очевидно: с помощью только отрицания вообще нельзя получить функцию от более чем одного аргумента, а с помощью только конъюнкции или только одной дизъюнкции нельзя получить функцию, тожественно равную 1. (В этом месте уместно предложить учащимся получить такую функцию с помощью только операций & и $\overline{}$, а также с помощью операций $\vee$ и $\overline{}$; в первом случае это $\overline{x\ \&\ \overline{x}}$, во втором — $x \vee \overline{x}$.)

Вопрос, можно ли получить любую булеву функцию, используя только конъюнкцию и дизъюнкцию, сложнее. Ответ на него отрицателен, однако обоснование такого ответа требует введения специальных понятий. Математиками получен ответ даже на более общий вопрос: как узнать, будет ли тот или иной набор функций достаточным, чтобы с его помощью построить любую другую булеву функцию? Этот ответ носит название теоремы По́ста. Мы не будем ее формулировать, поскольку, как было объявлено выше, потребовалось бы введение целого ряда новых понятий. Даже для ответа на конкретный вопрос о конъюнкции и дизъюнкции нам необходимо два определения.

Множество функций называется **замкнутым**, если для любого набора функций из этого множества их композиция принадлежит тому же множеству.

Функция $f(x_1, x_2, ..., x_n)$ называется **сохраняющей 0**, если $f(0, 0, ..., 0) = 0$. Этому множеству принадлежат функции &, $\vee$, $\oplus$ и 0, но не принадлежит функция $\overline{}$. Множество всех функций, сохраняющих 0, мы будем обозначать через M_0. Легко проверить, что класс M_0 является замкнутым. А поскольку он не содержит все булевы функции, то с помощью функций только из этого класса нельзя получить любую булеву функцию. Следовательно, и с помощью конъюнкции и дизъюнкции также нельзя построить произвольную булеву функцию [1].

Мы приветствовали бы инициативу учителя обсудить с учащимися рассмотренные здесь вопросы. Это важно не только с тех позиций, что учащиеся видят общую постановку проблем, связанных с конструированием устройств, позволяющих решать те или иные информационные (в частности, вычислительные) задачи, но и знакомство с общеметодологическим подходом к доказательству невозможности тех или иных конструкций. В основе таких доказательств лежит важная идея инварианта — так называют свойство объектов, которые не изменяются при их преобразовании. В нашем случае таким инвариантом служит свойство «сохранять значение 0». Для других множеств функций (о которых говорится в сноске) инвариантом является свойство, вынесенное в название множества. Подробно понятие инварианта будет обсуждаться в § 50 в связи с доказательством правильности работы алгоритма, но уже сейчас можно познакомить учащихся с этим важнейшим понятием, которое, увы, пока присутствует только в олимпиадной математике.

В конце объяснительного текста § 21 ставится проблема минимизации схемы, реализующей заданную булеву функцию. Это целое направление научных исследований, в котором есть свои достижения. Можно поручить учащимся подготовить на эту тему сообщения, опираясь на уже упо-

[1] Аналогично определяется класс функций, сохраняющих 1. Он тоже замкнут, и его обозначают через M_1. Ясно, что и этот класс не является полным. Кроме классов M_0 и M_1, важную роль играют еще три замкнутых, но неполных класса — класс монотонных функций M_2, класс линейных функций M_3 и класс самодвойственных функций M_4. Теорема Поста гласит, что множество функций является полным, если в нем для каждого из пяти указанных классов имеется функция, не принадлежащая этому классу.

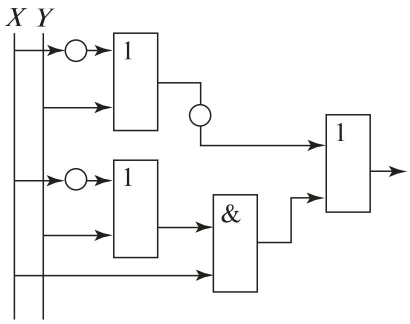


Рис. 2.8

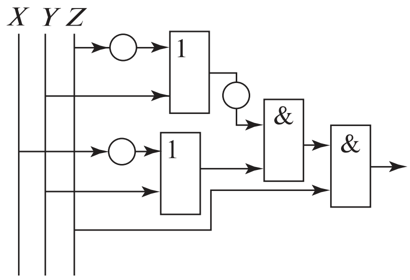


Рис. 2.9

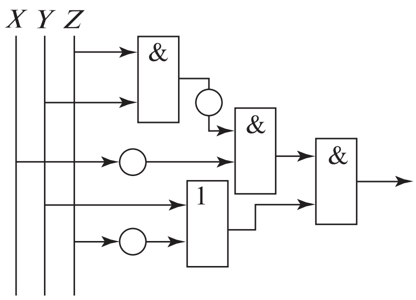


Рис. 2.10

минавшуюся книгу «Математические основы информатики. Элективный курс» Е. В. Андреевой, Л. Л. Босовой, И. Н. Фалиной (М.: Бином, 2005).

Перейдем к обсуждению заданий к § 21. Ответ на вопрос задания 1 в комментариях не нуждается. Задание 2 учащиеся должны выполнить по образцу доказательства ассоциативности конъюнкции, которое разобрано в объяснительном тексте параграфа.

При выполнении задания 3 учащиеся должны прежде всего заменить импликацию X → Y на равносильную формулу $\overline{X} \vee Y$ (доказательство равносильности учащиеся должны были провести, выполняя задание 2б, но дальнейших упрощений каждого из приведенных в этом задании выражений на этом шаге делать не следует). На рисунках 2.8—2.10 приведены ответы к заданию 3.

Затем можно предложить учащимся упростить каждое из выражений и снова построить соответствующие схемы. Ответы, получающиеся в этом случае, приведены на рисунке 2.11. Для формулы в пункте *в* применение закона де Моргана к упрощению не приводит, так что нет оснований к построению преобразованной схемы.

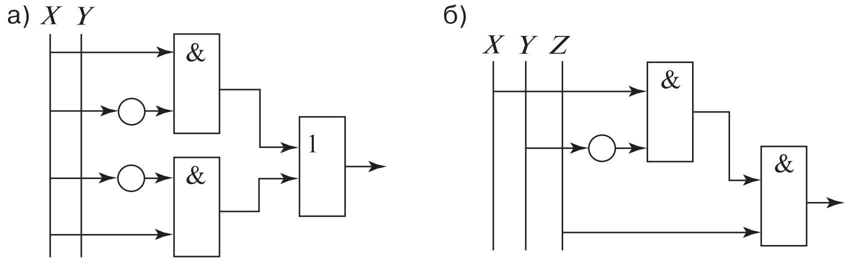


Рис. 2.11. Схемы, полученные после упрощения формулы: а) $\overline{X}$ & $Y \vee X$ & $\overline{Y}$; б) X & $\overline{Y}$ & Z

Задание 4 обычно не вызывает у учащихся затруднений. Соответствующая формула выглядит так: $x \& y \vee \overline{x} \& y \vee \vee x \& \overline{y}$. К этому моменту у учащихся уже достаточно большой опыт составления таблицы значений (этому было посвящено задание 2), так что задание 5а тоже легко выполняется учащимися. Что касается задания 5б, то учащиеся здесь могут просто сравнить полученную таблицу с таблицей 2.19 из учебника и сообщить ответ (мы бы советовали выполнять это задание устно). Но можно подойти к решению и по-другому: преобразовать формулу, полученную в результате выполнения задания 4. Ответ в обоих случаях должен получиться один и тот же: $x \vee y$.

Задания 6 и 7 аналогичны заданиям 4 и 5 (и даже схемы весьма похожи). Но здесь учащиеся должны аккуратно проставить скобки. Ответ к заданию 6: $\overline{(\overline{x \vee y} \vee \overline{x} \& y) \& (x \vee \overline{y})}$. После выполнения задания 7б должен получиться тот же ответ: $x \vee y$, что и в задании 5б.

Задание 8 связано с построением формулы по таблице значений. Здесь учащиеся должны продемонстрировать умение исполнить соответствующий алгоритм. Для разрядной единицы результата получается формула $\overline{x} \& y \& \overline{z} \vee \vee \overline{x} \& \overline{y} \& z \vee x \& \overline{y} \& \overline{z} \vee x \& y \& z$. Для единицы переноса получается формула $\overline{x} \& y \& z \vee x \& \overline{y} \& z \vee x \& y \& \& \overline{z} \vee x \& y \& z$. Мы надеемся, что составить схемы по этим формулам читателю будет уже не очень сложно, и позволим себе не загромождать соответствующими рисунками эту книгу. Что касается пункта *в* этого задания, то мы рекомендуем задать его домой, придав ему некоторый соревновательный дух: кому удастся составить самую экономную схему? На самом деле эти схемы не поддаются сколько-нибудь значительному сокращению — для первой формулы число элементов удается сократить на два, для второй формулы — на четыре. Правда, поскольку реализовать надо сразу обе схемы в одном устройстве, то можно добиться суммарной экономии используемых вентилей за

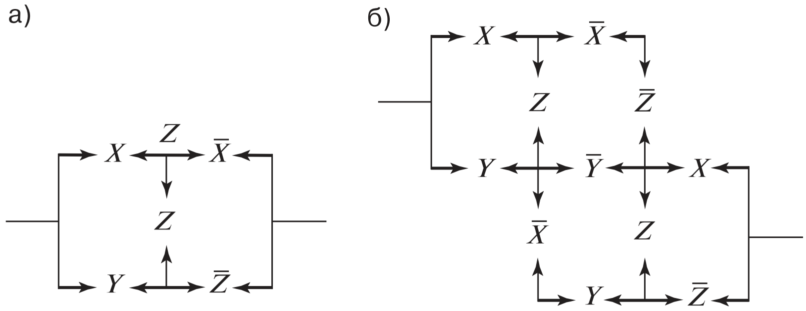


Рис. 2.12

счет многократного использования сигнала из одного вентиля.

Можно предложить учащимся аналогичным образом построить схему для мультипликатора двух двухразрядных чисел (см. задание 5 к § 20).

Здесь можно вернуться к исходным постановкам и отметить, что простейшие переключатели вовсе не обязательно могут быть соединены последовательно или параллельно. Из курса физики учащиеся знакомы с так называемыми мостовыми схемами, в которых один и тот же проводник по отношению к другим может рассматриваться как соединенный параллельно, а может — как соединенный последовательно. В физике возникают задачи расчета сопротивления таких цепей, для нас же важно только то, как проходит по ним сигнал. На рисунке 2.12 приведены примеры двух таких мостовых переключательных схем (переключатели в них обозначены большими латинскими буквами; контакт считается замкнутым, если обозначающая его переменная имеет значение 1, и разомкнутым, если обозначающая его переменная имеет значение 0); сигнал подается на левый контакт схемы, снимается с правого.

Каждая мостовая схема также реализует некоторую булеву функцию. Задача состоит в том, чтобы составить схему из вентилей, которая была бы эквивалентна мостовой (т. е. сигнал на выходе в обеих схемах одинаково зависел бы от значений переменных). Как правило, учащиеся сами приходят к алгоритму решения таких задач. Он состоит из трех шагов:

1) По мостовой схеме составить таблицу значений той функции, которую данная схема реализует.

2) По таблице значений записать формулу для соответствующей булевой функции.

3) По полученной формуле составить вентильную схему.

Вот как этот алгоритм реализуется для схемы на рисунке 2.12а:

Шаг 1. Заполняем таблицу для функции F, выражающей сигнал на выходе схемы.

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Шаг 2. Записываем формулу для функции F:
$F = \overline{X}\ \&\ Y\ \&\ \overline{Z} \vee \overline{X}\ \&\ Y\ \&\ Z \vee X\ \&\ Y\ \&\ \overline{Z} = \overline{X}\ \&\ Y\ \&\ (\overline{Z} \vee \mathrm{Z}) \vee X\ \&\ Y\ \&\ \overline{Z} = \overline{X}\ \&\ Y \vee X\ \&\ Y\ \&\ \overline{Z} = Y\ \&\ (\overline{X} \vee X\ \&\ \overline{Z}) = Y\ \&\ (\overline{X} \vee X)\ \&\ (\overline{X} \vee \overline{Z}) = Y\ \&\ (\overline{X} \vee \overline{Z})$.

Шаг 3. Составляем схему (рис. 2.13).

Для второй схемы алгоритм исполняется аналогично. Учащимся полезно выполнить это задание.

Отметим, что обратного алгоритма — построения мостовой схемы по заданной формуле — пока никто не придумал, хотя в ряде случаев она оказывается более экономной, нежели последовательно-параллельная схема.

Продолжим обсуждение заданий к § 21. Задания 9 и 10 содержат в себе обоснование высказанного в тексте параграфа утверждения, что каждого из элементов — операции Шеффера и операции Пирса — достаточно, чтобы сконструировать любую схему. Действительно, выполнив задание 9а, учащиеся убедятся в том, что с помощью операции Пирса можно сконструировать дизъюнкцию и отрицание, а из них, в свою очередь, конструируется любая булева функция. Первое равенство в задании 9а проверяется, как и ранее, с помощью таблицы значений. Второе тоже можно проверить, составив

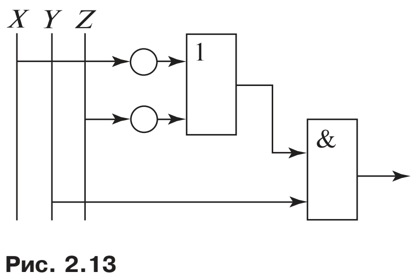


Рис. 2.13

соответствующую таблицу значений, но можно поступить проще. Согласно полученному уже равенству имеем

$$(x_1 \uparrow x_2) \uparrow (x_1 \uparrow x_2) = \overline{x_1 \uparrow x_2} = x_1 \vee x_2.$$

Второе из этих равенств легко увидеть, сопоставив столбцы функций $x_1 \uparrow x_2$ и $x_1 \vee x_2$ из таблицы 2.19 учебника. Что касается задания 9б, то здесь надо воспользоваться одним из законов де Моргана:

$$x_1 \mathbin{\&} x_2 = \overline{\overline{x}_1 \vee \overline{o}_2} = \overline{x}_1 \uparrow \overline{x}_2 = (x_1 \uparrow x_1) \uparrow (x_2 \uparrow x_2).$$

Задание 10 выполняется аналогично; приведем соответствующие формулы:

$$\overline{x} = x \mid x; \quad x_1 \mathbin{\&} x_2 = (x_1 \mid x_2) \mid (x_1 \mid x_2);$$
$$x_1 \vee x_2 = (x_1 \mid x_1) \mid (x_2 \mid x_2).$$

Цели § 22 чисто ознакомительные. Основные понятия, которые должны быть усвоены учащимися: разрядность ячейки, регистр, разрядность процессора, машинное слово, однородность памяти и связанный с этим понятием один из принципов фон Неймана общей архитектуры компьютера. Задания к этому параграфу носят исключительно репродуктивный характер. Ознакомление с материалом может быть осуществлено в разных формах: рассказ учителя, самостоятельное изучение, подготовленные выступления-презентации учащихся.

Тема 9. Представление чисел в памяти компьютера. Особенности компьютерной арифметики

Эта тема является кульминацией всей главы. В ней математические методы, позволяющие представлять числа разного вида и выполнять над ними операции в двоичной системе счисления, проецируются на технологические решения, в которых неизбежно присутствуют ограничения на разрядность ячеек памяти и регистров арифметических устройств, предназначенных для выполнения этих операций.

Начать можно с напоминания, что при использовании компьютера всегда приходится указывать тип данных. Даже для числовых данных различают как минимум два типа — целый и вещественный. Как написано в § 7 учебника для 10 класса, это связано с тем, что алгоритмы выполнения арифметических операций над числами разного типа сильно отличаются один от другого (и мы надеемся, что учащиеся помнят этот тезис). Дело, однако, не только в этом. С учащимися полезно обсудить, что целые числа встречаются в повседневной человеческой практике гораздо чаще вещественных чисел. Ведь количество сотрудников в фирме, количество единиц произведенной продукции (станков, автомобилей и т. п.), количество зданий и многое другое измеряется именно в целых числах, и действия над ними производятся обычно так, что в результате получается точное целое число. А с вещественными числами действия почти всегда совершаются лишь приближенно. Даже если исходное число выражено конечной десятичной дробью, то при переводе в двоичную систему оно, вероятнее всего, окажется бесконечной двоичной дробью и, следовательно, будет округлено. Для большей убедительности можно предложить учащимся оценить, как часто среди десятичных дробей, скажем, с тремя знаками после запятой встречаются дроби, которые при переводе в двоичную систему счисления дают конечную двоичную дробь. Всего десятичных дробей с тремя знаками после запятой имеется 1000 — от 0,000 до 0,999. Запишем каждую такую дробь обыкновенной дробью со знаменателем 1000 (не производя сокращения). Ясно, что среди них в конечную двоичную дробь будут переводиться только те дроби, у которых числитель делится на 125. Таких чисел всего лишь 8 — это 0; 125; 250; 375; 500; 625; 750 и 875. Так что вероятность получить при переводе такой десятичной дроби в двоичную систему счисления конечную дробь равна 0,008, или, выразив эту вероятность в процентах, всего 0,8%.

Самая главная причина, почему при вычислениях на компьютере приходится различать целые и дробные числа, состоит в том, что в памяти компьютера и в регистрах его арифметического устройства целые и дробные числа представлены по-разному. Об этом и рассказывается в § 23 и 24 нашего учебника.

Начинать естественно с представления целых чисел. Учащиеся уже знакомы с двоичной системой счисления, поэтому целесообразность ее применения для записи натуральных чисел не вызывает у учащихся сомнений. Ясно также, что на кодирование знака числа требуется еще один бит, но, чем именно — 0 или 1 — кодировать знак «–», неважно, надо просто принять некоторую договоренность. В рамках этой договоренности мы получаем так называемый **прямой код** целого числа.

Однако выполнение операций над целыми числами, записанными в прямом коде, наталкивается на существенные трудности. И трудность эта вовсе не «компьютерная». Чтобы учащиеся убедились в этом, им можно предложить изобразить алгоритм сложения чисел, который они осваивали в 6 классе, когда начали изучать действия с числами разных знаков, считая, что алгоритм сложения положительных чисел уже освоен. Мы вместо схемы алгоритма приведем запись на русском алгоритмическом языке, который использовали и в учебнике 10 класса.

```
Алгоритм Сложение
 { цел: a, b;
   Запросить a;
   Запросить b;
   Если (a и b одного знака) то
   {       Сложить модули чисел a и b;
           Приписать результату общий знак чисел a и b;
   }
   иначе
   {       Если (модуль числа a больше модуля числа b)
           то
           {      Из числа модуля числа a вычесть модуль
                  числа b;
                  Приписать результату знак числа a;
           }
           Если (модуль числа a больше модуля числа b)
           то
           {      Из числа модуля числа a вычесть модуль
                  числа b;
                  Приписать результату знак числа a;
           }
           Если (модуль числа a равен модулю числа b)
           то
           {      Объявить результатом число 0;
           }
   }
 }
```

Возможно, что учащиеся удивятся, как им раньше удалось усвоить такой сложный алгоритм, в который встроено еще несколько вспомогательных алгоритмов-функций (вычисление суммы, вычисление разности, определение, сравнение двух положительных чисел). Придумать простое устройство, реализующее такой алгоритм, весьма непросто. Даже сложение двух натуральных чисел, представленных в двоичном коде, реализуется, как было показано в § 20, с помощью целой батареи одноразрядных сумматоров. Выход один — менять способ кодирования для отрицательных чисел [1]. Такой способ кодирования был придуман. Он называется **дополнительным кодом**. Оказалось, что ограни-

[1] Здесь можно вспомнить об уравновешенной троичной системе счисления (см. с. 104—106), в которой положительные и отрицательные числа записываются единообразно и операции выполняются по единому, привычному для всех алгоритму сложения.

ченность разрядной сетки, которая обычно воспринимается как отрицательное свойство, приводящее к различным негативным последствиям (например, потере точности при вычислениях), может оказаться весьма полезным. Именно это свойство позволяет после записи отрицательных чисел в дополнительном коде действовать с ними так же, как с положительными числами.

Отдельного внимания заслуживает число 0. Для него прямой код (если мы его рассматриваем как +0) и дополнительный код (если мы его рассматриваем как –0) совпадают. Так что компьютер, как и люди, фактически считает нуль числом без знака.

Учащиеся должны освоить алгоритм построения дополнительного кода, который приведен в учебнике, и уметь его обосновывать (см. задание 3 к § 23). Тем, кто занимается программированием, важно также знать разрядность представления целых чисел в форматах Integer, Longint и Shortint.

Выполняя задание 1, учащиеся должны продемонстрировать не только формальное знание понятия «дополнительный код», но и понимание причин его введения.

В задании 2 учащиеся должны провести довольно формальные вычисления: 4 ячейки содержат 32 бита, так что диапазон кодируемых целых чисел составляет от -2^{31} до $2^{31} - 1$.

Задание 3 мы фактически обсудили выше.

В задании 4 признаком отрицательности числа служит 1 в старшем (крайнем левом) разряде. Тем самым в пункте *а* отрицательными являются следующие числа: 10111011; 10001001; 11111111. Дополнительный код для этих чисел будет соответственно таким: 11000101; 11110111; 10000001. В пункте *б* отрицательными являются числа: 1011101110111011; 1011101110001001; 1111111111111111. Для них дополнительный код выглядит так: 1100010001000101; 1100010001110111; 1000000000000001.

Задание 5 также, на наш взгляд, не вызовет у учащихся затруднений. Ответы в нем таковы: а) 11110110; 10011011; 10111110; б) 1111111010100001; 1110000101011101; 1001100111000001.

Приведем ответы к заданию 6: а) –211; –166; –144; б) –53843; –42134; –33376.

Для того чтобы получить ответ в задании 7, заметим, что в прямом коде четное число заканчивается нулем. При переходе к дополнительному коду этот нуль заменится на 1, однако после прибавления 1 последняя цифра станет снова нулем. Таким образом, отрицательное число четно тогда и только тогда, когда последняя цифра его дополнительного кода 0.

Задание 8 подготавливает учащихся к восприятию материала § 25. Поэтому его можно разбирать непосредственно перед изучением компьютерных операций над целыми числами. Но можно рассматривать его и как расширенное закрепление материала § 23. Поскольку в пункте *а* фактически содержится 6 примеров на выполнение операции сложения, то мы советуем сложение первых двух чисел осуществить в классе, а остальные примеры задать на дом. Вот как это может выглядеть. Поскольку второе число отрицательно (признак — 1 в старшем разряде), запишем его в дополнительном коде: 11101011. А теперь складываем с первым числом:

$$\begin{array}{r} 01001010 \\ +\;11101011 \\ \hline 100110101 \end{array}$$

Однако код содержит только 8 разрядов, поэтому старший, девятый разряд, содержащий 1, гасится. Ответ: 00110101.

Приведем остальные 5 ответов:

- сумма 01001010 и 11010111 равна 11110011 (надо иметь в виду, что это дополнительный код отрицательного числа, в десятичной системе счисления это –13);
- сумма 01001010 и 00110110 равна 10000000 (получилось –128, что не может не вызвать недоумения — ведь складывали положительные числа);
- сумма 10010101 и 11010111 равна 10010100 (после отбрасывания старшего, девятого разряда получается правильный ответ, в десятичной системе это –108);
- сумма 10010101 и 00110110 равна 00100001 (после отбрасывания старшего, девятого разряда получается правильный ответ);
- сумма 11010111 и 00110110 равна 11011111.

Таким образом, ровно один раз у нас произошло неверное вычисление; как будет объяснено в § 25, это проявил себя эффект переполнения разрядной сетки.

В пункте *б* уже 12 случаев. При вычитании перемена операндов действия местами влияет на результат — меняет его знак на противоположный, а в случае записи отрицательного числа в дополнительном коде меняется полностью результат вычитания. Здесь важно продемонстрировать учащимся, что вычитание можно заменить сложением по формуле $a - b = a + (-b)$. Если b при этом отрицательно и записано в прямом коде, то надо просто обнулить старший разряд и выполнить сложение; если же положительно, то надо превратить его в противоположное число и записать в дополнительном коде. Дальнейшее выполнение задания аналогично пункту *а*.

Знакомство с эффектами компьютерной арифметики целых чисел учащиеся продолжат при выполнении лабораторной работы № 5. Использование *Инженерного калькулятора* позволяет легко иллюстрировать различные эффекты. Важно, чтобы учащиеся сами обнаруживали и объясняли эти эффекты. Роль учителя — акцентировать на них внимание и комментировать нужным образом.

Если класс достаточно сильный, то можно дополнительно углубить знания в вопросах компьютерной арифметики целых чисел. Мы не случайно везде (и в учебнике, и здесь, в книге для учителя) записываем целые числа, не опуская ведущие нули. Этим мы хотим подчеркнуть, что для компьютера число — это просто некоторое машинное слово. Операции компьютер выполняет над машинными словами, а уже мы их интерпретируем как сложение, вычитание, умножение чисел, которые представлены этими машинными словами. Однако раз речь идет просто о машинных словах, то над ними можно выполнять и другие, в том числе булевы операции. Напомним, что, рассматривая коды, исправляющие ошибки, мы ввели операцию $\oplus$ побитового сложения двух кодов (см. с. 119) и там же объявили, что нередко ее обозначают Xor. В *Инженерном калькуляторе* такая операция располагается на правом краю панели. Учащиеся могут проверить работу этой операции. Но, кроме операции Xor, там же располагаются клавиши с операциями And, Or и Not. Это операции побитового И, ИЛИ и НЕ. Это именно побитовые операции! Учащимся надо предложить поэкспериментировать с этими операциями и убедиться, что они функционируют именно так. Конечно, для этого числа на табло надо помещать в двоичной системе счисления (тогда они приобретают вид машинного слова, правда, ведущие нули на табло не высвечиваются, но подразумеваются!). Например, 1011 And 101 = 0001 = 1; 1011 Or 101 = 1111. Надо только иметь в виду, что при использовании операции Not количество разрядов регулируется выбором количества байтов на табло. Так, если набрать 1011 и нажать клавишу Not, то при выборе режима «1 байт» получится 11110100, а при выборе режима «4 байта» получится уже 11111111111111111111111111110100.

Затем учащимся можно предложить эксперименты с этими операциями применительно к натуральным числам, записанным в десятичной системе. Пусть они узнают, чему равно 27 And 13 или 35 Or 18. В этом случае каждое из чисел переводится в двоичную систему и к получившимся машинным словам применяются эти операции. Важно, что и компьютер целые числа обрабатывает точно так же. Если немного поразмыслить, то это не покажется слишком удивительным — ведь на самом деле компьютер и имитируемый им калькулятор работают по одним и тем же принципам. Поэтому в программе на языке Паскаль можно написать оператор:

$a := b$ **and** $(b - 1)$.

Учащимся можно предложить определить, как преобразует этот оператор натуральное число b.

Следующий вопрос: при каких b число будет равным нулю? Оказывается, что в том и только в том случае, когда b — это степень числа 2. Почти наверняка учащиеся решали задачу «Составить алгоритм, позволяющий определить, является ли заданное натуральное число степенью двойки» (она позволяет проиллюстрировать применение циклической конструкции в форме «Пока»). Решить такую задачу можно одним оператором и без всяких циклов. В олимпиадном программировании, где фактор времени работы программы весьма существенен, подобные ухищрения весьма полезны. Мы также предлагаем поручить учащимся выяснить, какой результат можно получить, используя команды:

а) $a := b$ And $(b + 1)$;
б) $a := b$ And $(-b)$;
в) $a :=$ (Not b) And $(b + 1)$;
г) $a :=$ (Not b) And $(b - 1)$;
д) $a := b$ Xor $(b - 1)$.

Учащиеся могут и сами придумать различные комбинации побитовых и арифметических операций, а затем поэкспериментировать.

На самом деле математиками доказана следующая теорема:

Теорема. Функция, преобразующая машинные слова в машинное слово, может быть реализована посредством побитовых операций сложения, вычитания, И, ИЛИ, НЕ тогда и только тогда, когда каждый бит результата зависит только от битов исходных операндов той же позиции и правее нее.

Мы не будем приводить здесь доказательство этой теоремы.

Выше мы уже обсудили, что действия с дробными числами осуществляются всегда приближенно. Но что еще важнее — дробные числа в компьютере всегда представляются в нормализованном виде. Надо сказать, что раньше (скажем, четверть века назад) в курсе математики всегда обсуждалась нормализованная запись числа, поскольку учащиеся должны были уметь выполнять вычисления на логарифмической линейке. На смену логарифмическим линейкам пришли калькуляторы, затем микрокомпьютеры. И теперь с представлением чисел в нормализованном виде учащиеся встречаются разве лишь на уроках физики, когда обсуждают размеры атома или, наоборот, звезд и планет. Необходимость рассматривать действия с числами, представленными в такой форме, исчезла. Мы вовсе не ратуем за восстановление в курсе математики раздела вычислений с помощью логарифмической линейки, но учитель должен понимать, насколько непривычной для учащихся является данная проблематика. Поэтому начало § 24 мы посвятили довольно обстоятельному введению понятия нормализованного числа и связанных с этим понятием терминов «мантисса» и «порядок». Однако и этого еще недостаточно, чтобы адекватно описать компьютерное представление вещественных чисел — дело в том, что в таком представлении используется понятие машинного порядка. Не следует также игнорировать наше замечание в самом конце § 24, где сказано, что представление вещественных чисел зависит и от используемого языка программирования.

С учащимися также надо обсудить вопрос, который не затронут в § 24: что представляет собой нормализованное представление числа 0? Конечно, в житейской практике представлять 0 в нормализованном виде никогда не приходится. Однако при компьютерных вычислениях такое вполне может случиться. Обычно учащиеся высказывают естественную гипотезу, что мантисса такого числа равна 0. Что касается порядка этого числа, то обычно порядок тоже принимается равным 0 (хотя ясно, что его величина роли не играет). Не менее естественным выглядит и решение у нулевого числа считать нулевым машинный порядок. Это позволяет не различать представление 0 с фиксированной и плавающей запятой. Решение этого вопроса опять-таки зависит от реализации представления вещественных чисел в конкретном языке программирования [1].

Первые три задания к § 24 направлены на закрепление основных понятий и терминов, представленных в этом параграфе. В задании 4 каждое из чисел необходимо перевести в двоичную систему счисления, затем нормализовать и записать в требуемом машинном коде. Вот как выглядит эта процедура для числа 325:

$325 \to 101000101 \to 0{,}101000101 \cdot 2^9 \to$
$\to 0\ 1001001\ 101000101000000000000000$

(для наглядности мы разделили пробелом разряды, отведенные под знак числа, машинный порядок и мантиссу; учащиеся могут давать ответ записью без пробелов).

Остальные ответы к этому заданию таковы:

б) $-2{,}75 \to -10{,}11 \to -0{,}1011 \cdot 2^2 \to$
$\to 1\ 1000010\ 101100000000000000000000$;

[1] Этим, в частности, обусловлено стремление на олимпиадах по программированию входные данные представлять целыми числами. Вещественные числа если и возникают в таких задачах, то, как правило, в результатах работы программы. Но сказанное небесполезно учитывать не только в олимпиадной, но и в повседневной программистской практике.

в) $0{,}15 \to 0{,}001(0011) \to 0{,}1(0011) \cdot 2^{-2} \to$
$\to$ 0 0111110 10011001100110011001010

(в круглых скобках, как обычно, записан период; кроме того, при записи мантиссы пришлось выполнить округление);

г) $-0{,}0125 \to -0{,}0000(0011) \to -0{,}(1100) \cdot 2^{-6} \to$
$\to$ 1 0111010 11001100110011001100.

В задании 5 сначала требуется выполнить переход от машинной записи к обычному числу:

00111101 11001100110011001101 $\to$
$\to 0{,}11001100110011001101 \cdot 2^{-3} \to$
$\to 0{,}1000000014901161193847656 25_{10}$.

Как видно из этой записи, ошибка возникает уже в девятом разряде после запятой. Абсолютная погрешность равна 0,0000000014901161193847656 25, а относительная в 10 раз больше: 0,000000014901161193847656 25[1]. Но в обычных инженерных вычислениях такой точности, как правило, бывает достаточно.

Ответ на вопрос задания 5в довольно очевиден: возникновение 1 в последнем разряде — эффект округления.

Чтобы получить ответы на вопрос задания 6, необходимо оценить порядок числа в его двоичном представлении. Для этого не требуется каждое из чисел переводить в двоичную систему счисления (но если учащиеся предложат такой путь, то не следует им отказывать пройти по нему по крайней мере для одного из чисел — пусть еще раз убедятся, что применение математических знаний способно существенно облегчить решение задачи). Заметим, что нормализованная запись числа a в двоичной системе — это его представление в виде $\pm m \cdot 2^p$, где $\frac{1}{2} \leqslant m < 1$, а p — целое число. Значит, выполнены неравенства $\frac{1}{2} \cdot 2^p \leqslant |a| = m \cdot 2^p < 2^p$, откуда следует, что $\log_2 |a| < p \leqslant \log_2 |a| + 1$. Учитывая, что p — целое число, получаем, что p этими неравенствами определено однозначно. В пункте *а* для порядка p получается значение 58 (те, кто добросовестно перевел число $0{,}19 \cdot 10^{18}$ в двоичную систему счисления, получили

1010100011000000111111111001001011010100110000000000000000

и смогли убедиться, что после нормализации порядок действительно равен 58 — данное число содержит в своей

[1] Нередко оказывается, что учащиеся не помнят определения абсолютной и относительной погрешностей. Приведем для справки эти определения. **Абсолютной погрешностью** называется абсолютная величина разности между точным и приближенным значениями. **Относительной погрешностью** называется отношение абсолютной погрешности к точному значению числа.

записи в точности 58 цифр). Тем самым данное число представимо четырехбайтным кодом. Правда, оно подвергнется округлению, поскольку в мантиссу помещается лишь 24 разряда. Вот как запишется это число:

$$0\ \ 1111010\ \ 101010001100000100000000.$$

Возможно, учащимся захочется узнать, насколько «пострадало» исходное число от такого округления. Переведем машинное представление обратно в десятичную систему. Получается $0{,}190000007326203904 \cdot 10^{18}$. Ошибка все в том же девятом разряде.

В пункте *б* порядок получается равным 63. Это число также представимо четырехбайтным кодом. В пункте *в* порядок числа равен 64, и, следовательно, это число уже нельзя представить в данном формате. В пункте *г* порядок равен –64, и, таким образом, данное число может быть представлено четырехбайтным кодом. А вот в пункте *д* порядок числа равен –67, так что это число не представимо четырехбайтным кодом. Наконец, для числа $-0{,}25 \cdot 10^{-19}$ (пункт *е*) порядок оказывается равным –65, поэтому это число тоже не представимо в рассматриваемом формате. Это число является почти пограничным — число $0{,}28 \cdot 10^{-19}$ еще представимо четырехбайтным кодом. Можно предложить учащимся с помощью *Инженерного калькулятора* выяснить, какое наибольшее и какое наименьшее положительное вещественное число допускает указанное машинное представление.

Ограниченность разрядной сетки накладывает существенный отпечаток на выполнение действий даже в случае целых чисел. Частично рассмотрение этих эффектов начато еще в § 23, однако термин «переполнение разрядной сетки» вводится и подробно обсуждается для операций сложения и умножения целых чисел лишь в § 25.

Но основное внимание в § 25 сосредоточено на действиях с действительными числами. Прежде всего учащиеся должны осознать необходимость выравнивания порядков компонентов действия при выполнении операций сложения и вычитания. Как отмечено в учебнике, такое выравнивание может приводить к обнулению мантиссы одного из операндов и, по существу, к игнорированию одного из компонентов действия. Нам представляется весьма полезным, чтобы учащиеся не просто прочитали в учебнике соответствующий пример, а сами открыли указанный эффект, выполнив сложение (или вычитание) над подходящими числами. Впрочем, это пожелание относится не только к данному эффекту, но и ко всем остальным эффектам, о которых рассказывается в тексте § 25.

В заключительном абзаце объяснительного текста § 25 мы предупреждаем учащихся о весьма распространенной ошибке: для данных вещественного типа не следует в конструкциях ветвления и цикла в качестве условия записывать равенство или неравенство двух чисел. Операция сравнения двух чисел компьютером осуществляется через сравнение разности этих чисел с нулем. Возможность возникновения любой из четырех ситуаций, перечисленных в § 25, означает, что компьютером будет выработано неверное логическое значение, и исполнение алгоритма пойдет по неверному пути. Вместо сравнения на точное равенство $a = b$ (или неравенство $a <> b$) следует использовать условие $\text{abs}\,(a - b) < e$ (или соответственно $\text{abs}\,(a - b) > e$) для заранее выбранного числа e, которое служит оценкой погрешности округления. Как мы видели, обычно ошибка округления возникает в девятом разряде после запятой, поэтому в качестве числа e обычно берут 10^{-9}.

Ответ на вопрос задания 1 однозначен: основной причиной негативных эффектов компьютерной арифметики является ограниченность числа разрядов в ячейке и регистрах арифметического устройства.

В задании 2а переводим числа 100 и 50 в двоичную систему счисления, получается 1100100 и 110010. В однобайтовом коде получаем следующее представление: 01100100 и 00110010. Их сумма равна 10010110. Это отрицательное число, поэтому оно записано в дополнительном коде. Тогда модуль этого отрицательного числа равен 1101010_2. Значит, получилось число –106.

В задании 2б аналогично получаем

$$-80_{10} \to -1010000_2 \to 10110000;$$
$$-64_{10} \to -1000000_2 \to 11000000;$$
$$10110000 + 11000000 = 101110000 \to 01110000.$$

Это код положительного числа 1110000_2, в десятичной системе счисления получаем 112.

Задание 3 требует от учащихся хорошего знания так называемой арифметики остатков, или, говоря языком высшей алгебры, действий в кольце вычетов по модулю 2^m, где m — количество разрядов в коде числа. О том, что это действия в кольце вычетов по указанному модулю, фактически объявлено в самом конце § 23, где объясняется, что дополнительный код отрицательного числа n — это не что иное, как число $2^m + n$. Поскольку для действий над остатками справедливы все те законы, которые выполняются для операций над целыми числами, то сохранится и сочетательный закон для сложения. Вообще говоря, у учащихся могут появиться сомнения в его справедливости, так как возможна ситуация, когда при сложении чисел a и b возникает переполнение, а за счет отрицательности слагаемого c это переполнение не наступает.

Продемонстрируем выполнение задания 4.

а) Здесь порядки слагаемых одинаковы; складывая мантиссы, имеем $0{,}101 + 0{,}1 = 1{,}001$. Следовательно, результат требуется нормализовать. Ответ: $0{,}1001 \cdot 2^4$.

б) Выравниваем порядки до большего показателя — второе слагаемое записывается как $0{,}0001 \cdot 2^2$. Сложение мантисс дает 0,11. Дополнительная нормализация не требуется. Ответ: $0{,}11 \cdot 2^2$.

в) Выравниваем порядки до большего показателя — второе слагаемое записывается как $0{,}01 \cdot 2^{-1}$. Сложение мантисс дает –0,0111. Следовательно, результат требуется нормализовать. Ответ: $0{,}111 \cdot 2^{-2}$.

Продемонстрируем выполнение задания 5.

а) Сумма порядков равна 6. Произведение мантисс равно 0,0101, значит, требуется нормализация. Ответ: $0{,}101 \cdot 2^5$.

б) Сумма порядков равна 1. Произведение мантисс равно 0,01011, значит, требуется нормализация. Ответ: $0{,}1011 \cdot 2^0$.

в) Сумма порядков равна –3. Произведение мантисс равно 0,01011, значит, требуется нормализация. Ответ: $0{,}1011 \cdot 2^{-4}$.

При обсуждении задания 6 учащиеся довольно быстро осознают, что умножение мантисс не может приводить к нарушению сочетательного закона[1]. Значит, проблемы возникают из-за суммирования порядков. Ясна и причина возникновения проблем — ограниченность числа разрядов, отводимых под порядок, а самое главное, здесь нет механизма дополнительного кода для отрицательных порядков. Приведем конкретный пример. Вычислим выражения $(0{,}1 \cdot 2^{33} \times 0{,}1 \cdot 2^{33}) \times 0{,}1 \cdot 2^{-30}$
и $0{,}1 \cdot 2^{33} \times (0{,}1 \cdot 2^{33} \times 0{,}1 \cdot 2^{-30})$.

При вычислении произведения в скобках для первого выражения сумма порядков равна 11000010, однако для порядка отведено 7 разрядов, поэтому старший разряд будет утерян. Остается 1000010. Кроме того, для нормализации порядок требуется уменьшить на 1, так что окончательно имеем

0 1000001 10000000000000000000000.

[1] Учащиеся могут настаивать на том, что возникающие ошибки округления при перемножении в разном порядке способны привести к разным результатам. И это действительно так. Но возникающая разница в результатах вычислений затрагивает максимум два последних разряда мантиссы, так что можно считать, что в этом случае сочетательный закон выполняется в пределах допустимой погрешности. А вот проблемы с суммированием порядков приводят к принципиальным различиям в результатах.

Это соответствует числу $0{,}1 \cdot 2^1$, т. е. 1. После умножения на $0{,}1 \cdot 2^{-30}$ и нормализации получаем $0{,}1 \cdot 2^{-30}$.

Нетрудно видеть, что для второго выражения сначала получится $0{,}1 \cdot 2^2$, а затем $0{,}1 \cdot 2^{34}$. Тем самым результаты не совпали.

Можно предложить учащимся убедиться, что для чисел с плавающей запятой ассоциативный закон нарушается не только для умножения, но и для сложения. Если класс достаточно сильный, то подобрать подходящие числа для иллюстрирующего примера учащиеся могут самостоятельно. Если вы не уверены в том, что учащиеся способны это сделать, им можно предложить следующее задание:

Задание 25. По правилам компьютерной арифметики вычислите суммы $(-0{,}111 \cdot 2^{11} + 0{,}1 \cdot 2^{13}) + 0{,}11 \cdot 2^{-12}$ и $-0{,}111 \cdot 2^{11} + (0{,}1 \cdot 2^{13} + 0{,}11 \cdot 2^{-12})$. Убедитесь, что для этих чисел нарушается сочетательный закон.

Важно также продемонстрировать учащимся, что эффект переполнения может возникать не только при сложении целых чисел в формате с фиксированной запятой, но и при сложении нормализованных чисел. Именно на это направлено задание 7 из § 25.

Задание 8 демонстрирует то, как может зависеть результат от организации вычисления одной и той же величины. Ответ в пункте *а* положителен: оба алгоритма подсчитывают сумму чисел, обратных первым N натуральным числам. Однако число $\frac{1}{N}$ для значения N, указанного в пункте *б*, при записи его в нормализованном двоичном коде имеет порядок, равный –29. В алгоритме, приведенном слева, переменная S уже после двух исполнений тела цикла будет равна 1,5, т. е. в нормализованном виде представлена как $0{,}11 \cdot 2^1$. А дальше порядок переменной S будет только возрастать. Поскольку выбрано четырехбайтное представление, выравнивание порядков для S и $\frac{1}{N}$ приведет к тому, что мантисса числа $\frac{1}{N}$ при сложении с уже накопленным значением переменной S станет нулевой. На самом деле это будет происходить даже гораздо раньше, так что, начиная с некоторого K, значение S перестанет изменяться. Если же суммирование ведется в обратном порядке, то сначала будут складываться близкие по степени малости числа, их сумма будет все время вполне соизмерима по величине порядка с очередным слагаемым. И несмотря на то что в конце все равно будет выполнено округление до 24 значащих цифр мантиссы, результат исполнения правого алгоритма получится иным, нежели результат для левого алгоритма. Мы настоятельно советуем подкрепить это умозрительное рассуждение соответствующим компьютерным экспериментом (в лабораторной работе № 6 это задание 9).

Это задание тесно связано с группой заданий 10—12 из лабораторной работы № 6. В них изучается фактически тот же алгоритм суммирования чисел, обратных числам натурального ряда. Из проведенного выше обсуждения ясно, что алгоритм зацикливается ввиду того, что переменная S перестает изменяться, как только N становится достаточно большим (например, больше чем 10^8). Для константы 10, с которой проводится эксперимент при выполнении задания 11, мы не указываем точное значение N, поскольку оно может оказаться разным для разных версий используемых языков программирования, но получится примерно 22 000. А квадрат этого числа $\approx 4{,}84 \cdot 10^8$, что как раз близко к границе зацикливания. В задании 12 из лабораторной работы № 6 учащимся предлагается экспериментально определить наибольшее значение константы (мы советуем ограничиться натуральными числами), для которого программа еще не зацикливается. Предыдущие два задания показывают, что оно заключено между 10 и 20. Опять-таки в зависимости от используемой версии языка программирования эта константа получается 17 или 18.

Результат, который требуется получить в задании 10 — найти N, для которого S станет больше 20, можно попытаться вычислить, реализуя алгоритм суммирования в обратном порядке (он приведен в задании 8 из § 25). Ведь оценка для N уже получена: $\approx 4{,}84 \cdot 10^8$. Прежде всего нужно выяснить (с помощью указанного алгоритма), какое значение получит переменная S при $N = 4{,}84 \cdot 10^8$. По полученному результату учащиеся определяют, нужно ли еще увеличивать значение N или, наоборот, требуется его уменьшить. На самом деле это значение N больше, чем необходимо, почти в 1,7 раза. Поэтому поиск нужного значения N здесь можно организовать методом деления пополам. Ответ: 272 404 868 (он может получиться несколько иным опять-таки из-за различий в языковой реализации).

Эту экспериментальную работу мы предлагаем проводить на втором часе компьютерного практикума, предусмотренного лабораторной работой № 6. А первый час практикума посвятить выполнению заданий 1—8.

ГЛАВА 3

Основные информационные объекты. Их создание и компьютерная обработка

Мы уверены, что одиннадцатиклассники в большинстве своем уже хорошо владеют базовыми технологиями создания таких информационных объектов, как текст, таблицы и диаграммы, рисунки в растровой и векторной графике. Освоение этих технологий предусматривается Федеральным стандартом по информатике в базовом звене образования (8—9 классы), и, как правило, в этой части указанный стандарт выполняется в полном объеме. Тем не менее Федеральным стандартом по информатике в заключительном звене школьного образования предусмотрено практически полное повторение вопросов, изучавшихся ранее. Поэтому учитель в рамках реального процесса обучения здесь должен сориентироваться как по учебному времени, отводимому на изучение материала данной главы, так и по расстановке акцентов в его изучении учащимися. На наш взгляд, рассмотрение многих тем может происходить в режиме самостоятельного изучения (а фактически расширенного повторения) материала с привлечением дополнительной литературы (которая имеется в изобилии) и последующим обсуждением в классе. Основное внимание следует уделить освоению практических навыков по использованию базовых технологий создания и обработки информационных объектов разной природы.

В то же время учащиеся должны осознавать, что информационные технологии нужны не сами по себе, а для решения информационных задач, возникающих в разнообразной человеческой практике. Напомним, что владение информационными технологиями является важнейшим составным элементом информационной культуры личности. Однако в реальности он становится таким элементом только в том случае, если их освоение не самоцель, а средство совершенствования такой культуры. Поэтому мы призываем учителя постоянно иметь это в виду и демонстрировать учащимся связи материала данной главы с тем, что они уже знают, изучив главу 1 учебника.

В аналогичной книге для учителя, посвященной методике преподавания информатики в 10 классе [1], мы достаточно подробно остановились на проблеме развития информационной компетентности учащихся. В ней отмечено, что данная компетентность развивается постепенно, проходя несколько уровней. Вот эти 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 спланировать, придумать схему создания информационного продукта или операций над ним;
- уровень экспертной компетентности: умение дать 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.

Мы призываем учителя при изучении учащимися материала этой главы, где речь идет прежде всего именно о создании информационных объектов, постоянно уделять внимание вопросам развития компетентностей на соответствующем уровне. Мы полагаем, что уровень исполнительской компетентности освоен учащимися полностью в ходе предшествующего изучения информатики. Более того, и уровень технологической компетентности также, вероятнее всего, достаточно высок. А вот к развитию экспертной и аналитико-синтезирующей компетентностей следует приложить отдельные усилия.

Тема 10. Основные информационные объекты, их создание и обработка

В § 26 и 27, по существу, мы напоминаем учащимся основные функции текстовых процессоров, позволяющих создавать и редактировать текстовые документы. Именно самостоятельное (в домашних условиях) знакомство с материалом § 26 представляется нам наиболее подходящей формой его изучения. Полезно предложить учащимся составить краткий справочник тех возможностей редактирования текста, которые описаны в этом параграфе (отметим, что значительная их часть указана в задании 8). После этого на уроке в классе

[1] См.: Гейн А. Г. Информатика и информационные технологии: кн. для учителя: метод. рекомендации к учеб. 10 кл. — М.: Просвещение, 2008.

можно разобрать задания, предложенные в конце § 26. Они почти все устные (кроме заданий 9—11) и позволяют сразу выяснить уровень усвоения предложенного материала. Задание 10 опирается на результаты, полученные при выполнении задания 9. Ответ в этом задании положителен. В профильном классе можно сформулировать следующую задачу:

Задание 26. Имеется прямоугольник со сторонами a и b, причем $a > b$. Его разрезали на два равных прямоугольника. Оказалось, что получившиеся прямоугольники подобны исходному прямоугольнику. При каком отношении сторон a и b это возможно?

Решение. Поскольку у меньшего прямоугольника стороны равны b и $a/2$ (рис. 3.1), то их подобие означает выполнение пропорции $a : b = b : a/2$. Следовательно, $a^2 = 2b^2$. Значит, $\frac{a}{b} = \sqrt{2} \approx 1{,}41421$.

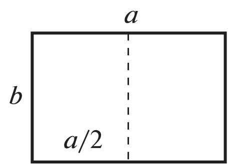


Рис. 3.1

Заметим, что 297 : 210 = 1,41429, т. е. размеры стандартного листа бумаги подобраны именно так, чтобы указанное разрезание давало лист той же формы.

Задание 11 является подготовительным к выполнению первой части лабораторной работы № 7. Напомним, что составление плана — это один из весьма распространенных методов свертывания информации. План (в смысле текстового, а не графического свертывания информации) можно определить как информационную модель текста, представляющую собой систематизированный перечень разделов и подразделов, характеризующих содержание и отражающих композицию текста. План всегда имеет иерархическую структуру. Для его составления требуется разбить текст на смысловые фрагменты, найти в каждом ключевые слова и на их основе озаглавить каждый из выделенных фрагментов. Учащиеся уже много раз на уроках (особенно на уроке литературы) составляли планы, однако вовсе не исключено, что четко сформулировать определение плана и алгоритм его составления им удастся не сразу.

Бóльшая часть заданий лабораторной работы № 7 направлена не только на освоение и закрепление умений пользоваться возможностями текстового редактора, но и на повторение основных приемов свертывания информации, рассмотренных в главе 1. Учащимся классов с гуманитарным уклоном можно предложить познакомиться еще с одной стратегией смыслового свертывания. Она получила название «синквейн» (от французского cinq — пять), поскольку построенный по этой стратегии информационный текстовый объект обязательно содержит 5 строк.

Синквейн создается по строго определенной схеме:

1-я строка — одно ключевое слово, определяющее содержание синквейна; как правило, это существительное;

2-я строка — два прилагательных или причастия, характеризующие данное понятие;

3-я строка — три глагола, обозначающие действие в рамках заданной темы;

4-я строка — короткое предложение из четырех слов, раскрывающее суть темы или отношение к ней;

5-я строка — синоним ключевого слова либо образ, раскрывающий смысл; как правило, снова существительное.

Синквейн позволяет описать суть понятия и осуществить рефлексию на основе полученных знаний. Синквейн используется на стадии послечтения для встраивания нового опыта, новых знаний в систему уже имеющихся личностных смыслов. Вот примеры четырех синквейнов, созданных старшеклассниками, участниками одного литературного кружка г. Челябинска, по повести А. Платонова «Котлован».

Котлован. Глубокий, черный. Зияет, пугает, поглощает. Строить общий дом — рыть могилу. Бездна.	Котлован. Огромный, глубокий. Копать, надеяться, умирать. Надежда на светлое будущее. Могила.
Котлован. Сырой, бездонный. Разрушает, толкает в бездну, несет гибель. Земля растерзана фанатиками идеи. Могила.	Котлован. Страшный, бездуховный. Отучает думать, радоваться жизни. Самое страшное — бездумная вера. Крах.

Как видите, при всем лаконизме синквейны очень образны, метафоричны, эмоциональны, экспрессивны. Они являются способом образного обобщения, синтеза эмоционального восприятия и понимания смысла. И использование этой стратегии смыслового свертывания воспитывает у учащихся соответствующие черты мышления.

Конечно, уроки информатики — не место, где следует подобным вопросам уделять систематическое внимание, однако демонстрировать подобные технологии (особенно в классах гуманитарной и социальной профессиональной направленности) нам представляется чрезвычайно важным и полезным.

В § 27 рассказывается о возможностях повышения выразительности создаваемых информационных текстовых объектов, предоставляемых текстовым редактором. Это в первую очередь использование табличной формы, а также внедрение рисунков и диаграмм. Применение этих средств повышает визуализацию структуры информационного объекта, а следовательно, способствует повышению его содержательной емкости. Собственно говоря, о применении таблиц, схем, рисунков как средства свертывания информации речь шла в главе 1. Здесь же акцент снова делается на технологической стороне. Но повторим — ни в коем случае не следует забывать, что не сами технологии, а умение с их помощью реализовывать содержательно емкие информационные объекты является, на наш взгляд, главной целью изучения этих технологий в заключительном звене школьного образования. Поэтому лабораторная работа № 8 разработана нами как продолжение лабораторной работы № 7, в ней предлагается информационно усилить тот объект, который был создан в предшествующей лабораторной работе.

Отдельным вопросом стоит овладение редактором математических формул. Наверно, для классов гуманитарного профиля этот материал не нужен; что касается физико-математического и информационно-технологического профилей, то им его изучение предписано Федеральным стандартом по информатике.

Вопросы, сформулированные в заданиях к § 27, имеют своей целью закрепить теоретические знания. Ответы на них содержатся в тексте параграфа.

Приступая к изложению материала, представленного в § 28—32, мы снова стояли перед дилеммой. Вполне возможно, что учащиеся в достаточной мере знакомы как с самим понятием гипертекста, так и со средствами его создания и редактирования. Многие школьники к 11 классу уже имеют личные web-страницы. В этом случае уроки по этой части темы 10 можно построить в виде бесед и сообщений учащихся с параллельным выполнением заданий из лабораторных работ № 9—11. Если же учащимися этот материал осваивается впервые, то можно следовать объяснительному тексту этих параграфов, в котором с достаточной, на наш взгляд, степенью подробности описаны возможные процедуры создания гипертекстовых документов. Принципиально важным здесь является понимание учащимися «контейнерного» принципа вложенности тегов. Но это, как правило, не вызывает затруднений, поскольку и по форме, и по существу дела данный принцип ничем не отличается от организации вложенных друг в друга алгоритмических конструкций, с чем они уже многократно имели дело ранее.

Выполнение заданий к § 28 обычно не вызывает трудностей. Из заданий к § 29 приведем для контроля ответы к заданию 5: правильно расположены контейнеры лишь в пункте *а* этого задания. Мы советуем предложить ученикам, начиная с этого параграфа, создавать свой рабочий справочник по тегам. Пример такого справочника приведен ниже (мы добавили туда часто применяемые теги, которые не упомянуты в тексте учебника).

Краткий справочник по HTML

Элемент	Значение
<HTML> ... </HTML>	Указатель начала и окончания описания электронного документа на языке HTML
<HEAD> ... </HEAD>	Заголовок
<BODY> ... </BODY>	Тело документа
<TITLE> ... </TITLE>	Название документа
	Базовый адрес
	Поисковый документ
	Общая гипертекстовая ссылка
<H?> ... </H?>	Размер символов (? = 1, 2, ..., 6)
 	Перевод строки
<P> ... </P>	Начало абзаца
	Отображение без изменений с учетом пробелов
<HR>	Разделитель — горизонтальная линия, идущая через весь экран
<IMG SRC="FILE: ... >	Вставка изображения из файла (с указанием пути к файлу и его имени)
<A HREF="URL...>...</A>	Ссылка на URL. Задание подписи или адреса
<MARQUEE> ... </MARQUEE>	Вывод бегущей строки
<UL> ... </UL>	Неупорядоченный список
<LI> ... </LI>	Элемент списка
<OL> ... </OL>	Упорядоченный список
<CENTER> ... </CENTER>	Центрирование текста
<I> ... </I>	Курсив
<B> ... </B>	Жирный шрифт
<U> ... </U>	Подчеркивание
<S> ... </S>	Перечеркнутый текст
<BIG> ... </BIG>	Увеличенный размер шрифта
<SMALL> ... </SMALL>	Уменьшенный размер шрифта
_{...}	Подстрочные символы
^{...}	Надстрочные символы
<FONT COLOR="?"> ... </FONT>	Цвет текста (? — имя цвета)
<CITE> ... </CITE>	Цитирование
<CODE> ... </CODE>	Отображение примеров кода
<SAMP> ... </SAMP>	Последовательность символов
<KBD> ... </KBD>	Ввод символов с клавиатуры
<TABLE> ... </TABLE>	Начало и конец таблицы
<TR> ... </TR>	Начало и конец строки в таблице
<TD>... </TD>	Начало и конец содержимого ячейки в таблице
<VAR> ... </VAR>	Переменная
<DFN> ... </DFN>	Определение
<Q> ... </Q>	Текст, заключенный в скобки
<AU>... </AU>	Автор
<INS> ... </INS>	Вставка текста
<DEL> ... </DEL>	Удаление текста
<TEXTAREA> ... </TEXTAREA>	Поле для ввода текста

Мы привели этот мини-справочник, во-первых, для удобства самого учителя и, во-вторых, чтобы учитель мог легко ответить на оперативно возникающие вопросы учащихся (хотя нужно научить учеников самостоятельно разыскивать ответы в соответствующих источниках, в том числе электронных).

Если учащиеся в достаточной мере владеют основами создания гипертекстовых документов, то с ними можно обсуждать более сложные вопросы: как оптимально с точки зрения восприятия информации должна быть организована гипертекстовая страница? Визуальный уровень информационного продукта определяется сочетанием смыслового содержания страницы с ее внешним видом, формой представления. Применяя те или иные методы и приемы создания информационных продуктов, важно понимать особенности восприятия различных элементов содержания вероятными пользователями. Обычно на гипертекстовой странице можно выделить следующие базовые элементы:

- текст;
- поля ввода;
- кнопки;
- гиперссылки;
- пиктограммы;
- списки;
- графические изображения.

Поясним некоторые из этих терминов.

Поле ввода — элемент интерфейса, с помощью которого человек передает информацию в информационную систему. Различают следующие виды полей:

- текстовое поле;
- поле выбора из списка (выпадающий список);
- маркер (чек-бокс);
- переключатель (радиокнопка).

Кнопка — элемент, воздействие на который вызывает динамическую реакцию системы, например вызов нового диалогового окна.

Гиперссылка — основной элемент управления перемещением пользователя по информационному продукту. Виды гиперссылок:

- внутренняя (якорь, переход в верхнюю часть страницы);
- внешняя (навигационная ссылка, действие).

Пиктограмма — миниатюрный графический образ, обозначающий какое-либо действие или объект.

Список — элемент, используемый для перечисления однотипных структурных единиц.

Эти элементы, организованные специальным образом, влияют на восприятие человеком представленной на странице информации, управляют его вниманием. Разумеется, способность воспринимать информацию у каждого человека индивидуальна. Но есть и много общего в психологии восприятия. Как часто многие, проходя мимо огромной рекламной растяжки, не замечают того, что на ней написано, несмотря на размер букв, их яркий цвет и даже заманчивость предложения. В то же время ваш взгляд может остановиться на каком-либо объявлении, напечатанном черным по белому обычным шрифтом. Подобные явления легко объяснимы, если использовать понятие «фокус внимания человека». Мы рассмотрим это понятие с точки зрения контакта человека и компьютера.

Фокус внимания человека при взаимодействии с компьютером — это место на экране монитора, куда направлен взгляд и где он сознательно сосредоточен.

В задачи разработчика информационного ресурса входит управление фокусом внимания пользователя, а именно:

- привлечение внимания к элементу содержания;
- удержание внимания с целью постижения (понимания) смыслового содержания элемента;
- перемещение внимания к последующим элементам содержания и так до тех пор, пока цель пользователя или разработчика информационного ресурса не будет достигнута.

По мнению психологов, наиболее распространена следующая последовательность восприятия человеком содержимого web-страницы:

- исходные точки — левый верхний угол экрана или центр экрана;
- направление дальнейшего перемещения — построчно, слева направо, сверху вниз.

Однако существуют факторы, способствующие изменению последовательности восприятия. К ним относятся:

- статические графические изображения (например, фотография, рисунок, чертеж);
- динамические графические изображения (например, бегущая строка, видеоролик, анимированный рекламный баннер).

Между прочим, такие изображения могут полностью «закрыть» восприятие более важных смысловых элементов, поэтому их использование должно быть минимальным. К сожалению, школьники (да и не только они) на первых порах создания web-страниц чересчур увлекаются подобными эффектами.

Хотя на содержательном уровне Интернету будет посвящена следующая глава, учащимся, активно владеющим технологиями создания web-страниц, полезно предложить работу по оцениванию организации главной страницы какого-либо сайта. Вот примерный план такой работы:

- Дать описание внешнего вида web-страницы, характеризующего визуальный уровень восприятия информационного продукта.
- Составить схему размещения основных смысловых блоков на странице.
- Описать структуру web-страницы, определив логические связи между отдельными ее элементами.

Описание внешнего вида может быть дано в виде таблицы со следующими столбцами:

№	Название элемента	Способ представления (вид информации)	Назначение элемента

А схема размещения и описание структуры могут быть выполнены в виде подходящих графов.

Мы советуем обращаться к практике создания гипертекстовых страниц и в дальнейшем, например при освоении работы с графическими объектами (этому будут посвящены несколько последующих параграфов той же главы).

В § 33 весьма кратко дан обзор компьютерных словарей и систем перевода текстов. Здесь, на наш взгляд, нет особых теоретически значимых знаний; реально освоение таких средств происходит в ходе конкретной практической работы. Много зависит от языка и качества реализации автомата-переводчика, от характера текста — литературные произведения таким способом лучше не переводить, в нем слишком много метафор, образов фразеологических оборотов. Каждый, кто пользуется подобной системой, адаптирует ее под себя и адаптируется к ней сам. Мы даже не стали планировать для этого параграфа компьютерный практикум, хотя при желании учитель может в зависимости от профиля класса предложить те или иные задания к переводу с помощью подходящей системы. К примеру, если ваши ученики регулярно участвуют в интернет-олимпиадах по программированию (сайты www.olympiads.ru и www.neerc.ifmo.ru/school), то при проведении олимпиад с международным участием задания выкладываются на английском языке, в этом случае участникам приходится использовать и словари, и системы перевода текстов. Полезно познакомить учащихся и с толковыми словарями, имеющимися в Интернете.

Задания 3 и 4 к этому параграфу мы советуем выполнить дома, но с обязательной последующей проверкой

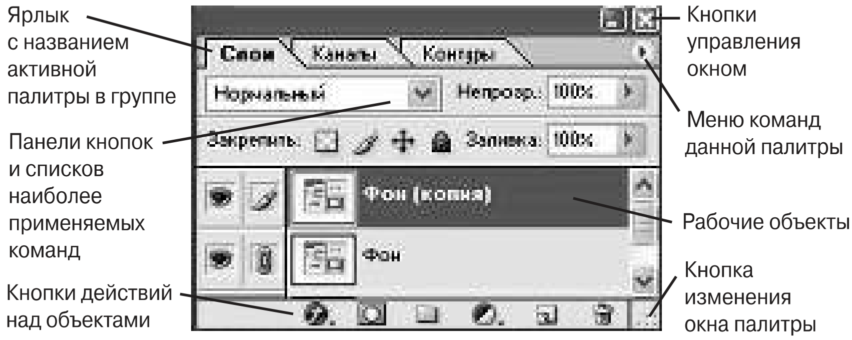


Рис. 3.2

в классе. Задание 5 можно сразу обсуждать в классе — причина описанного явления ясна обычно всем: неоднозначность смысла слов естественного языка, его неформализованность. Домой же можно предложить в качестве необязательного задания придумать или отыскать еще несколько фраз, в которых может наблюдаться подобный эффект. Разумеется, это задание более подходит для классов с углубленным изучением того или иного иностранного языка.

В § 34 и 35 учащиеся более глубоко знакомятся с обработкой графических объектов, представленных в растровых форматах. Основным инструментом, который здесь рассматривается, выступает Adobe Photoshop, но в случае его отсутствия можно воспользоваться любым другим растровым графическим редактором. При изучении этого материала вполне уместно предложить заинтересовавшимся учащимся воспользоваться дополнительной литературой, в частности той, которая указана в конце учебника. Основное внимание здесь, на наш взгляд, должно быть уделено практике использования различных инструментов этого мощного графического пакета. Этому способствуют предложенные лабораторные работы № 12—14, выполнение которых в зависимости от профильной направленности класса может занимать от 3 до 6 часов. К примеру, в лабораторной работе № 12 мы только в одном абзаце упомянули понятие палитры. На самом деле это целый набор разнообразных инструментов (рис. 3.2). Они предназначены для работы с изображением, и каждая посвящена одной теме, например выбору рабочих цветов, работе со слоями, хранению созданных контуров и действий над ними и т. д.

Следует рекомендовать учащимся открывать во время работы только необходимые им палитры и размещать их по правому краю рабочего поля, чтобы они не мешали созданию изображения. Совместно используемые палитры удобно группировать. В одном диалоговом окне группы может быть несколько вкладок — отдельных палитр. Переход между ними — щелчок по ярлычку с названием. Палитры можно вытаскивать из окна группы за ярлычок, можно, наоборот, помещать в окно группы. Ниже описаны приемы работы с некоторыми палитрами, с которыми полезно знакомить школьников.

Палитра **Навигатор** позволяет изменять масштаб отображения рисунка с помощью кнопок с треугольниками и движкового регулятора, управлять просмотром фрагментов изображения. Для этого имеется красная рамка, показывающая область изображения, видимую в окне рисунка, если масштаб велик. Рамку можно перемещать движением мыши при нажатой левой клавише.

Палитра **Инфо** показывает в зависимости от выбранного инструмента текущую информацию о координатах курсора мыши, размере области выделения, цвете точки под курсором в режиме RGB и CMYK и др.

Каждый раз, когда вы выполняете какое-то действие, оно записывается в виде строки в палитру **История (Протокол)**. Эта палитра позволяет отменять любое количество действий. Для этого надо нажать левую кнопку мыши на ползунке в левой части последней строки и протащить ее вверх по строкам отменяемых действий. Строки отмененных действий поблекнут. Отмененные строки не исчезают, их всегда можно снова вернуть, протащив мышью ползунок вниз по отмененным строкам.

В лабораторной работе № 13 речь идет еще об одной палитре — палитре слоев. Работа со слоями описана в учебнике.

Уже сказанное ясно показывает, что возможно построение и небольшого (объемом до 12—18 часов) элективного курса, ориентированного на более глубокое изучение средств компьютерной графики.

Обсуждая в § 2 понятие информационной грамотности, мы отдельным элементом выделили умение представлять свою точку зрения, новые знания и понимание или решение проблемы. Мы почти не останавливались на этом аспекте, хотя на самом деле это один из важнейших элементов информационной культуры. Его значимость определяется целым рядом причин.

Во-первых, результатом решения любой информационной задачи, т. е. результатом деятельности человека по обработке информации, выступает некоторый информационный продукт, который имеет презентационную ценность не только для того, кто решал информационную задачу, но и для лиц из окружения этого человека (иногда узкого, а иногда очень широкого) [1].

Во-вторых, решение вопроса о том, как будут представлены результаты работы с информацией, играет определяющую роль и в организации поиска нужной информации, и в подборе средств, используемых для ее обработки. Пожалуй, сильнее всего этот аспект ощущают школьные библиотекари, хотя делают они это, даже не задумываясь над тем, что выясняют у читателя именно форму презентации. Ведь, получая запрос от читателя, они нередко начинают выяснять: «Тебе для чего? Для доклада? Ах, олимпиада скоро, тогда тебе совсем другую книжку нужно». Говоря языком информатики, форма презентации результатов обработки информации является существенным фактором в решении информационной задачи. Можно смело утверждать, что, приступая к решению каждой информационной задачи, надо прежде всего понять, зачем, кому и в какой форме будут представлены итоги этой работы.

---

[1] Отвлекаясь от основной канвы наших рассуждений, сделаем несколько важных, на наш взгляд, замечаний. В настоящее время достаточно широкое распространение получила практика подготовки реферата или проведения учебно-исследовательской работы. Выполнение такой работы нередко оценивает лишь учитель. В итоге учащиеся берут информацию из Интернета (причем нередко даже не внося в нее хотя бы косметические изменения), списывают друг у друга или старших учащихся, которые выполняли такую же работу в предшествующем году. Учителя пытаются с этим бороться, сопоставляя полученные работы с имеющимися у них образцами таких же работ и уличая мнимых авторов в плагиате. Скажем прямо — это информационно безграмотное решение, и потому его эффективность весьма мала. Если презентационный круг состоит только из одного учителя, то единственная цель ученика заключается в том, чтобы любым путем сдать ему работу. Совершенно иная ситуация складывается, когда работа ученика выносится на обсуждение и оценивание широкого круга — одноклассников, учащихся одной параллели, общешкольное, с привлечением родителей и т. д. В такой ситуации ни один ученик не захочет, чтобы в него тыкали пальцем, говоря, что он не способен сделать что-то сам, а может лишь списать готовенькое. И чем шире презентационный круг, тем сильнее этот эффект. Ясно, что на уроках такое организовать невозможно, хотя бы ввиду ограниченности времени. Для этого необходима локальная компьютерная сеть (лучше всего в рамках всей школы и как минимум в рамках компьютерного класса), где функционировали бы соответствующие сайты. Иными словами, нужна компьютерная информационная инфраструктура школы. Мы не будем сейчас останавливаться на том, как ее организовать и поддерживать. Заинтересовавшемуся читателю мы предлагаем ознакомиться с нашими подробными рекомендациями на этот счет в книге для учителя «Информатика и информационные технологии. Методические рекомендации к учебнику 9 класса» авторов А. Г. Гейна, А. И. Сенокосова, Н. А. Юнерман (М.: Просвещение, 2008).

В-третьих, изначальное определение формы представления результатов работы с информацией позволяет сформулировать критерии оценки такой работы.

В целом работа, направленная на решение информационной задачи, иными словами, на удовлетворение информационной потребности, может быть условно разделена на шесть этапов.

1. Формулировка и уточнение информационной потребности, определение круга источников информации, выбор формы презентации результатов работы.
2. Определение средств и выработка алгоритма поиска информации.
3. Поиск и локализация информации.
4. Качественная оценка и отбор полученной информации.
5. Обработка, компоновка и интерпретация полученной информации.
6. Подготовка к презентации и презентация итогов работы.

Пункт 1 относится к постановке информационной задачи. Многое из этого пункта уже обсуждалось ранее. Пункты 2 и 3 будут подробно обсуждаться в главе 4. Об оценке уже имеющейся информации и ее отборе, как и ее интерпретации, мы тоже довольно много говорили в ходе обсуждения материала главы 1. В данный момент основное внимание сосредоточено на обсуждении пункта 6.

Все сказанное выше относится к презентации результатов информационной деятельности человека вообще, а необязательно тех из них, которые создаются с помощью компьютерных средств. И поэтому, приступая к рассмотрению с учащимися материала § 36, где речь идет о создании презентаций с помощью Power Point, необходимо, по нашему мнению, внушить учащимся важность учета тех положений, которые мы провозгласили выше.

Технология изготовления презентации изложена в учебнике, а в лабораторной работе № 15 приведен план практического занятия по освоению Power Point. Впрочем, список подготавливаемых презентаций может быть расширен с учетом пожеланий и фантазии учащихся.

Мы советуем ознакомиться с достаточно устоявшимися правилами оформления компьютерной презентации. Если говорить кратко, то они состоят в следующем[1].

Содержание

1. На слайдах презентации не пишут весь тот текст, который произносит докладчик (во-первых, в этом случае сам

[1] Ни в коем случае не следует воспринимать эти правила как догму. Нередко из них бывают исключения, продиктованные целевыми установками презентации. Но решение о каждом таком исключении должно быть продуманным и аргументированным для автора презентации.

факт произнесения доклада теряет смысл, так как аудитория обычно умеет читать, а во-вторых, длинный текст на слайде плохо воспринимается и только мешает слушанию и пониманию смысла).

2. Текст на слайде должен содержать только ключевые фразы (слова), которые докладчик развивает и комментирует устно.

3. Если презентация является основой устного доклада, то по европейским и американским правилам второй слайд должен содержать краткое перечисление всех основных вопросов, которые будут рассмотрены в докладе. Это нечасто встречается у нас даже на «взрослых» конференциях, но практика показывает, что данное правило чрезвычайно полезно: дисциплинирует докладчика, концентрирует внимание слушателей, а кроме того, во время создания такого слайда от автора требуется очень четко выделить и сформулировать ключевые проблемы доклада.

4. Если презентация имеет характер игры, викторины или какой-либо другой, требующей активного участия аудитории, то на каждом слайде должен быть текст только одного шага или эти шаги должны появляться на экране не сразу, а один за другим, постепенно.

Оформление

1. На первом слайде пишут не только название презентации, но и имена авторов (в ученическом случае и руководителя проекта) и дату создания.

2. Каждую прямую цитату, которую комментирует или даже просто приводит докладчик (будь то эпиграф или цитаты по ходу доклада), размещают на отдельном слайде обязательно с полной подписью автора (имя и фамилия, инициалы и фамилия, но ни в коем случае одна фамилия, исключение — псевдоним). Допустимый вариант — две небольшие цитаты на одну тему на одном слайде, но не больше.

3. Все схемы и графики должны иметь названия, отражающие их содержание.

4. Подбор шрифтов и художественное оформление слайдов должны не только соответствовать содержанию, но и учитывать восприятие аудитории. Например, сложные рисованные шрифты часто трудно читаются, тогда как содержание слайда должно восприниматься все сразу, одним взглядом. Цвета фона и шрифтов должны быть достаточно контрастными (но исключительно черно-белая палитра выглядит уныло и быстро утомляет слушателей). Не следует злоупотреблять красным цветом.

5. На каждом слайде выставляют колонтитул, включающий фамилию автора и/или краткое название презентации и год создания, номер слайда.

6. В конце презентации представляют список использованных источников, оформленный по правилам библиографического описания.

7. Правила хорошего тона требуют, чтобы в конце презентации была выражена благодарность тем, кто прямо или косвенно помогал в работе над ней.

8. Обычно на последнем слайде выражена благодарность слушателям за их внимание к работе.

Результаты выполнения заданий лабораторной работы № 15 обязательно должны выноситься на обсуждение, как минимум в рамках класса. Нельзя упустить прекрасную возможность для развития экспертной и аналитико-синтезирующей компетентностей. Важно при этом, чтобы при оценивании любой работы обязательно соблюдались следующие условия:

- выделялись положительные моменты в представленном продукте (ибо критиковать всегда легче);
- критические замечания должны относиться к работе, а не к личным способностям автора (недопустимы замечания в духе «у него руки не оттуда растут»);
- все комментарии, как положительные, так и отрицательные, должны быть аргументированы (не принимаются оценки типа «мне нравится» или «мне не нравится» без дальнейшей расшифровки, что именно нравится или не нравится и почему);
- критерии оценивания должны быть стандартизованными и объективными.

Хотя условия на критерии записаны последним, четвертым пунктом, на самом деле это ключевой пункт в формировании экспертной компетентности в целом и выполнения первых трех условий. Правильным решением является предварительное обсуждение критериев, по которым будет оцениваться информационный продукт (неважно даже, будет это презентация или нечто иное). В случае оценивания презентаций основу таких критериев могут составить те 4—8 пунктов, в которых выражены основные требования к компьютерным презентациям. Эти требования могут быть детализированы применительно к конкретной ситуации, могут быть сформулированы дополнительные критерии (и они могут быть даже более важными, поскольку будут отражать конкретную специфику).

Аналитико-синтезирующая компетентность будет проявлять себя в том, что наряду с замечаниями будут формулироваться конструктивные предложения по совершенствованию информационного продукта и технологические подходы к их реализации.

ГЛАВА 4

Телекоммуникационные сети. Интернет

Можно было бы повторить то, что сказано в преамбуле к главе 3: возможно, что к 11 классу учащиеся чувствуют себя в Интернете как рыба в воде. Поэтому цель данной главы — воспитание информационно грамотного пользователя телекоммуникациями.

Ощущение учащимися, что они все про Интернет знают, создает дополнительные проблемы с мотивацией к изучению данной темы. Одним из выходов здесь является использование сети Интранет, в которой возможно применение всех инструментов, существующих в глобальной сети, но без обращения к этой сети. Это довольно эффективный и распространенный вариант решения проблемы. Можно полагать, что в ближайшем будущем он станет основным, поскольку оплата интернет-трафика постепенно перекладывается на плечи самих образовательных учреждений, и не исключено, что руководство этих учреждений захочет минимизировать использование реального Интернета. Другой вариант — знакомство с Интернетом проводить в рамках строго регламентированной проектной деятельности учащихся. И это тоже хорошо зарекомендовавший себя вариант.

На наш взгляд, надо знакомить учащихся с одной важной тенденцией в развитии информационного общества — переходом к обществу, в котором системообразующей опорой являются знания. В обиход все шире входит и соответствующий термин «общество знаний». Чтобы разобраться в причинах, вызвавших к жизни данную тенденцию, вернемся в 70-е гг. прошлого столетия, когда возникло понятие информационного кризиса. Напомним, оно связано с тем, что объем научно-технической информации удваивался примерно в течение 3—4 лет, и это приводило к тому, что дешевле было повторить исследования, чем найти описание результатов в научно-технической литературе. Внедрение компьютеров и соответствующего программного обеспечения с последующим объединением информационных банков в глобальной сети Интернет, казалось бы, разрешило эту проблему. Действительно, требуемые результаты теперь можно найти, кликнув мышкой. Однако это оказался джин, выпущенный из бутылки. Концепция информационного общества, когда каждому жителю планеты должна быть предоставлена возможность получения и размещения любой информации (с известными оговорками, о которых речь шла в главе 1), привела к лавинообразному росту количества информации в Интернете.

Обилие *информации* уже давно воспринимается как нечто само собой разумеющееся. По оценкам экспертов, около 80% журналистов обращаются к Интернету в поисках новостей, но лишь 20% находят ту информацию, которая им необходима. Менеджеры многих фирм считают, что не могут эффективно работать без получения большого объема информации, но вместе с тем загрузка большим объемом данных снижает эффективность работы, препятствует нормальному функционированию корпорации. Это состояние, успевшее получить название «синдром информационной усталости», свидетельствует об избытке информации и недостатке знаний.

Стало ясно, что в сложившейся информационной ситуации уже недостаточно опираться лишь на традиционный путь превращения информации в знание через обработку этой информации вручную именно тем человеком, которому это знание потребовалось. Общая идея автоматизации информационных процессов путем применения компьютерной техники привела в 80-х гг. XX в. к возникновению так называемых **систем искусственного интеллекта** (с ними мы знакомили учащихся в главе 4 учебника для 10 класса), которые позволяют создавать и использовать базы знаний. **Знание** — это обобщенная и систематизированная информация, позволяющая решать различные практические и научно-познавательные задачи на основе тех или иных логических методов вывода. Как мы уже обсуждали, такие методы могут быть формальными (и потому поддаваться автоматизации), а могут быть эвристическими, т. е. являться сугубо человеческой прерогативой.

При поиске информации пользователь, как правило, представляет характер того, что может получить. Например, если требуется узнать, как по железной дороге доехать до нужной станции, то ясно, что надо запросить на соответствующем сервере информацию о расписании поездов. Другое дело, если он хочет познакомиться, скажем, с понятием бизнес-проекта: что это такое, как он разрабатывается, как внедряется, по каким критериям оценивается его успешность и т. д. Если обратиться к поисковой системе Интернета со словом «бизнес-проект», вряд ли будет получено желаемое знание. Скорее всего, вам предложат познакомиться с конкретными такими проектами в разных сферах предпринимательской деятельности или отправят в интернет-магазин купить книжку, в названии которой это словосочетание встречается. Иными словами, при поиске знаний пользователь рассчитывает получить нечто, до тех пор ему неизвестное, и познать его. Стандартный информационный поиск с этой работой справляется малоэффективно — ведь сам пользователь далеко не всегда может вразумительно сказать, что же ему нужно. В помощь такому поиску появилось новое направление в обработке информации — «глубинный анализ текстов» Text Mining.

Перед технологией Text Mining поставлены следующие задачи:

- автоматическая классификация, выявление смысловых взаимосвязей отдельных фрагментов и понятий, выраженных в тексте;
- составление осмысленных рефератов, резюмирующих знания, содержащиеся в текстовых массивах больших объемов.

Возможности данной технологии могут применяться маркетологами (например, при анализе рынка), политиками, бизнесменами, учеными — всеми, кто активно участвует в современных информационных, общественно-политических и бизнес-процессах.

Методы Text Mining активно используются в таких областях, как:

- политические исследования — геополитика, анализ предвыборной ситуации, деятельности партий, общественных организаций, политических деятелей;
- конкурентная разведка — обобщенный анализ деятельности конкурентов, их PR-активности, клиентской базы;
- анализ рынков — выявление основных тенденций в производстве и потреблении товаров и услуг определенных видов;
- анализ новых технологий — в различных сферах науки, бизнеса, безопасности;
- анализ достижений науки, образования и культуры — выявление основных тенденций в указанных областях человеческой деятельности.

Поэтому при выборе будущей профессии и определении сферы личных интересов в какой-либо из указанных областей сегодняшнему школьнику весьма полезно владеть этой технологией. Если учитель сочтет возможным познакомить учащихся с подобными технологиями, мы бы это только приветствовали.

Перейдем к методическому обсуждению вопросов, представленных в учебнике.

Тема 11. Телекоммуникационные сети и Интернет

Изучение темы начинается с рассмотрения локальных и глобальных компьютерных сетей (§ 37 и 38). Рассматриваются технические аспекты такого соединения, обсуждаются полезные эффекты, возникающие при таком соединении, вводятся соответствующие термины, такие, как: сетевая плата, сервер, топология сети, модуляция и демодуляция, модем, протокол информационного обмена и т. д. Кратко представлена история появления и развития глобальных сетей, в том числе Интернета. Вопросы и задания к этим параграфам ориентированы на закрепление полученных знаний и носят в основном репродуктивный характер.

Поскольку данная тема рассматривается нами в рамках базового курса, то учащимся дается только самое общее представление об организации сетей. При более пристальном взгляде на излагаемые вопросы надо иметь в виду, что на самом деле в различных сетях, вошедших и вновь подключаемых в Интернет, действуют весьма разнообразные протоколы. Чаще всего протоколы разрабатываются не отдельными компаниями, а независимыми комитетами, в которые входят представители различных организаций. Такие протоколы могут быть открытыми, что позволяет использовать их бесплатно в системах различных видов. Однако существуют и закрытые протоколы, за использование которых требуется платить.

Надо понимать, что при передаче данных на каждом компьютере сети работает сразу несколько протоколов, отвечающих за реализацию различных функций. Протоколы удобно разделять по уровням в зависимости от того, для выполнения какой функции они предназначены. Эти уровни вместе составляют так называемую эталонную модель OSI (Open Systems Interconnection — открытая системная связь)[1]. На рисунке 4.1 представлена схема работы протоколов семиуровневой модели OSI.

Удобно представлять, что протоколы, работающие на разных уровнях, образуют стек, который называют **стеком**

[1] Модель OSI появилась в 1983 г. как результат работы двух организаций: Международной организации по стандартизации и Сектора стандартизации телекоммуникаций Международного телекоммуникационного союза. Предполагалось, что предложенная модель станет основой для всех сетевых протоколов, однако в коммерческой форме этот проект так и не был реализован. Поэтому модель OSI в основном используется в качестве обучающего и справочного пособия, а также для теоретических исследований.

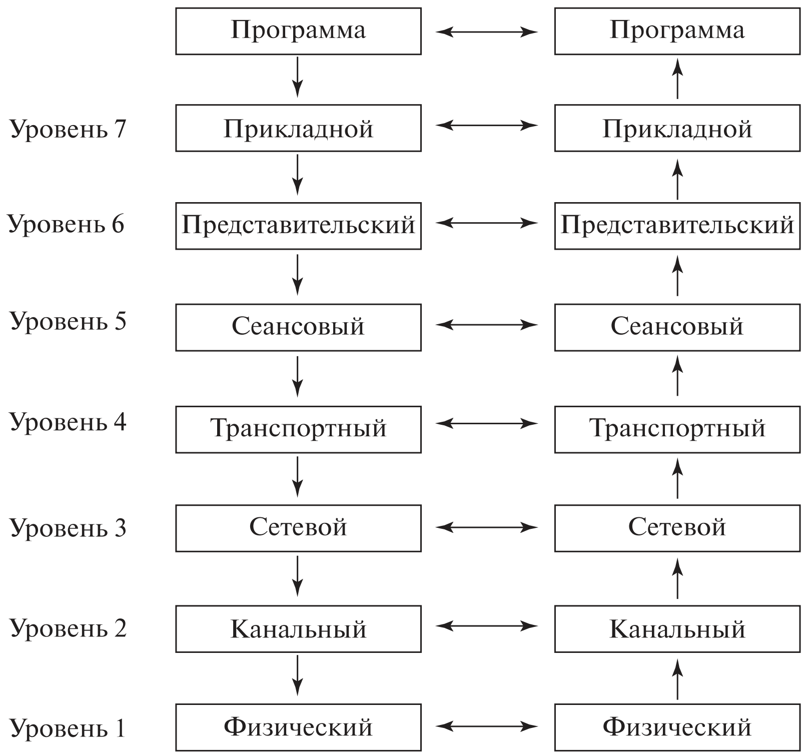


Рис. 4.1

протоколов. На включенном в сеть компьютере протоколы работают совместно, причем протоколы соседних уровней обслуживают друг друга. При отправке данных протокол более высокого уровня выполняет некоторые функции, необходимые для работы протокола, расположенного ниже. При приеме данных те же действия компьютером-получателем выполняются в обратном порядке. При этом протоколы отправителя и приемника, расположенные на одном уровне, выполняют однородные функции. Например, если на третьем уровне при передаче данных протокол отвечает за их кодирование, то протокол третьего уровня компьютера-приемника отвечает за их дешифровку. Реальная связь между компьютерами сети осуществляется протоколами физического уровня, но одновременная согласованная работа всех протоколов может восприниматься как единое целое.

На каждом уровне к результату обработки данных на предыдущем уровне добавляется заголовок, свидетельствующий о прохождении очередного уровня, и трейлер (trailer — прицеп) со служебной информацией. Так, на канальном уровне в трейлер помещается дополнительный код, позволяющий обнаруживать и исправлять ошибки (см. § 17). В целом это напоминает отправку обычного письма: содержательная информация вкладывается в конверт, на конверте надписывается адрес, потом ставится штамп почтового отделения и т. д. Итогом прохождения всех уровней становится **пакет**, готовый к передаче по сети. Когда пакет достигает места назначения, то при прохождении к запрашиваемой информации в обратном порядке с него снимаются все заголовки и трейлеры. Мы в своем рассказе на этом остановимся, поскольку более глубокое проникновение в данный вопрос потребовало бы еще немало страниц и времени.

В § 39 обсуждаются проблемы маршрутизации и адресации в Интернете. И здесь мы тоже воздержались от подробных объяснений и термина «маршрутизация», и термина «пакет». Маршрутизацией называется процесс выбора в сети самого эффективного маршрута для передачи пакетов от системы-отправителя к системе-получателю. В Интернете часто от одного компьютера к другому можно добраться разными путями. Сети специально проектируются с избытком путей, чтобы при сбое на одном из них пакет все же добрался до адресата. Выполняя лабораторную работу № 16, учащиеся смогут выяснить, по какому маршруту поступила к ним информация от того компьютера, на котором она располагается.

Об адресации в Интернете рассказано больше и подробнее. В ЕГЭ по информатике 2008 г. впервые появилось задание, связанное с точечно-десятичной формой IP-адреса (до этого рассматривались только URL-адреса). Вот одно из заданий.

Задание 27. Во время следствия были обнаружены четыре обрывка бумаги (рис. 4.2), и криминалисты установили, что на них записаны фрагменты одного IP-адреса, которые условно обозначили буквами А, Б, В и Г. Необходимо восстановить IP-адрес.

2.16	19	.55	8.131
А	Б	В	Г

Рис. 4.2

В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

Решение. Поскольку каждое из чисел, заключенное между двумя точками, должно быть не более 256, то нет соединения фрагментов вида АБ, ВА, ВБ, ВГ, ГА и ГБ, т. е.

после В поставить нечего. Следовательно, В — последний фрагмент IP-адреса. Тогда после Г может идти только В. Так как соединения АБ быть не может, то единственный вариант БАГВ, т. е. 192.168.131.55. Такой IP-адрес действительно возможен. Ответ: БАГВ[1].

Как видно, для выполнения этого задания достаточно ограничения, приведенного в объяснительном тексте § 39, что каждое из чисел, стоящее в IP-адресе, не более чем 256. На самом деле для числа, стоящего до первой точки, максимально возможное значение 223, ведь в двоичном коде этот байт самое большее может быть 11011111. Вполне можно ожидать, что в будущем учащимся придется воспользоваться и этим знанием.

Мы же сейчас фактически получили ответ на вопрос задания 4 к § 39: не может IP-адрес начинаться на 249. Аналогично получается ответ и на задание 5. Первый байт для сетей класса А может начинаться самое большее на 01111111, следовательно, в IP-адресе первое десятичное число не более 127. Для сетей класса В диапазон получается от 128 до 191, ибо $191 = 10111111_2$. Для сетей класса С диапазон фактически уже вычислен — от 192 до 249. Теперь уже легко получить ответ на вопрос задания 6: сетей класса В может существовать лишь 64, т. е. их в 2 раза меньше, чем сетей класса А.

Остальные задания этого параграфа нам представляются достаточно легкими.

В § 40 даны общие механизмы работы поисковых систем в Интернете. Учащимся полезно их понимать хотя бы в общих чертах, как это и представлено в данном параграфе, поскольку это способствует формированию умений составлять запрос и прогнозировать ответ поисковой системы. В частности, в ЕГЭ регулярно присутствуют задания (весьма простые) на оценку количества ответов в сети Интернет в зависимости от формы запроса. Вот одно из таких заданий.

Задание 28. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

[1] Подобные задания пока редко встречаются в учебной литературе. Чтобы создать еще несколько вариантов, учителю достаточно написать какой-либо IP-адрес, «порезать» его на части и получить задание. Однако восстановление адреса может оказаться неоднозначным. Поэтому мы приведем еще два варианта с заведомо однозначным восстановлением.

а)

.32	3.142	21	2.11
А	Б	В	Г

б)

5.253	212.1	13	.24
А	Б	В	Г

Для обозначения логической операции ИЛИ в запросе используется символ |, а для логической операции И — символ &.

1	волейбол | баскетбол
2	волейбол & баскетбол
3	волейбол
4	волейбол | баскетбол | правила

Решение. Союз И никогда не увеличивает количество ответов, удовлетворяющих данному запросу, а союз ИЛИ, наоборот, никогда не уменьшает количество ответов. Поэтому ответ таков: 2, 3, 1, 4.

Интернетная фабула в этом задании почти ни при чем. Для его выполнения вполне достаточно четко представлять себе работу операторов И и ИЛИ. Но надо также отчетливо понимать используемую в нем интернетную терминологию.

Есть в этом задании и некая идеализация работы поисковой системы. Неявно предполагается, что поисковик осуществляет просмотр всех страниц Интернета, что, как объяснено в § 40, вовсе не так: он просматривает только уже проиндексированную им часть Интернета. Если бы вместо баскетбола стояло нечто намного менее распространенное, то не исключено, что в пунктах 1—3 ответы были бы одинаковыми.

Задания к § 40, как и к § 37, и § 38, нацелены на закрепление полученных знаний и носят в основном репродуктивный характер. Обратим лишь внимание на задание 6, в котором фактически еще раз до учащихся доводится мысль, высказанная в конце объяснительного текста, что «привычка» поисковой системы работать в первую очередь с проиндексированными страницами может приводить к выдаче ссылок на уже не существующие электронные ресурсы.

Изучение материала § 40 сопровождается выполнением практической работы в компьютерном классе, сценарий которой представлен в лабораторной работе № 17. Выполняя ее, учащиеся знакомятся с работой основных инструментов. Нам представляется и здесь ставить перед учащимися вопросы, связанные с воспитанием информационной грамотности и экспертной компетентности. Для каждого из предложенных к просмотру сайтов можно обсуждать такие вопросы, как управление фокусом внимания, базовые элементы и общая структура, организация интерфейса. Интересен сравнительный анализ по указанным характеристикам. Он позволяет высказать предположения об основных целях каждого из сайтов и увязать эти цели с теми дизайнерскими решениями, которые были выявлены учащимися. Скажем сразу, что целевое назначение сайтов настолько разное, что различия бросаются в глаза. Первый сайт является культурно-просветительским, в то время как второй — политико-пропагандистским. Будет очень хорошо, если учащиеся осознают (и продемонстрируют), что столь различные цели достигались тем не менее с использованием примерно одних и тех же технологических средств.

Следующий блок параграфов образует смысловой и учебный концентры, посвященные организации поиска информации в Интернете. Поиск информации — это, по сути, путь следования от источника к источнику, пока не будет получена требуемая информация. Здесь учащиеся должны научиться отвечать в первую очередь на вопросы: с чего я начну поиск, к чему обращусь дальше, если не найду нужную информацию или ее часть? А дальше не простое следование намеченному пути, а еще и умение по ходу поиска сразу отсеивать информацию (или источники), не соответствующую требованиям, определенным еще на первом этапе, который, напомним, звучал так: формулировка и уточнение информационной потребности, определение круга источников информации, выбор формы презентации результатов работы. Если не научить ученика сразу отслеживать соответствие получаемой информации заданным параметрам, то раньше или позже он растеряется перед количеством полученных материалов, несопоставимых по уровню, степени достоверности и нужности.

Чтобы оптимально проложить путь поиска, надо иметь достаточно отчетливое представление о существующих возможностях и правилах поиска. Наиболее общей характеристикой поиска здесь выступает разделение поиска на два типа: адресный и тематический. В § 41 на примерах показывается, в чем заключается различие этих типов поиска [1]. Кроме того, в том же параграфе обсуждается несколько понятий, относящихся к оценке эффективности поиска, — это понятия релевантности, полноты и точности.

Компьютерный практикум к этому параграфу опирается не только на лабораторную работу № 18, но и на задание 2, сформулированное в самом параграфе. Разница в том, что в лабораторной работе построение пути поиска фактически прописано в самом сценарии работы. Выполняя задание 2, учащиеся должны проделать ту предварительную работу по организации поиска, о которой мы только что говорили. Нам представляется методически целесообразным предло-

[1] Эти иллюстрирующие примеры любезно предоставлены О. К. Громовой.

жить выполнить задание 2 дома, затем обсудить намеченные пути поиска в классе и лишь после этого реально осуществлять поиск в глобальной сети (для этого, вероятнее всего, придется использовать дополнительное время, представленное в тематическом планировании как резерв учителя).

Ответы к остальным заданиям из § 41 легко разыскиваются в его объяснительном тексте.

Изучение материала, представленного в § 42, — это мостик от Интернета, рассматриваемого как явление сугубо информационной сферы, к Интернету как социально значимому явлению нашей эпохи. Отсюда потянется ниточка к рассмотрению социальных явлений, связанных с интернетизацией общества, проблем защиты информации и т. д. Но уже сама постановка вопроса об Интернете как сервисной (т. е. обслуживающей) системе означает смещение взгляда на Интернет как на некий социально значимый объект в жизни общества.

В настоящее время опубликовано достаточно много материалов по ресурсам, имеющимся в Интернете. Можно выдать учащимся задания по поиску таких материалов и подготовке сообщений самими школьниками о возможностях Интернета. Те или иные материалы могут быть найдены в самом Интернете. При наличии выхода в Интернет желательно продемонстрировать по крайней мере некоторые из тех сервисов, которые описаны в объяснительном тексте § 42 (на это отводится 1 час компьютерного практикума).

Одним из социально значимых вопросов является выбор жизненного пути выпускниками школы. И можно смело сказать, что их малая информированность в вопросах послешкольного и послевузовского обустройства жизни весьма отрицательно сказывается не только на самих выпускниках, но и на обществе в целом. К примеру, сегодня зафиксировано значительное превышение числа выпускников экономического и юридического профилей, тем не менее желающих получить образование по этим специальностям по-прежнему весьма много. Но перепроизводство специалистов в какой-либо отрасли всегда сопряжено с целым рядом негативных последствий. Во-первых, превышение предложения над спросом всегда приводит к снижению цены, т. е. оплата труда работников в данной сфере начинает уменьшаться, и, в частности, падает престижность данной профессии. Во-вторых, деньги на обучение специалистов оказались потраченными впустую. В-третьих, трудоустройство невостребованных специалистов потребует вложения новых средств на их переподготовку. Поэтому вопрос о перспективах рынка труда и о выборе сферы дальнейшей деятельности или образования — это то, что обязательно должно быть в сфере внимания ученика. У нас же обучение тому, как школьник должен подходить к решению подобных вопросов, не фигурирует в программе никакого школьного предмета. Информатика — это не обществознание, в котором было бы уместно рассматривать подобные вопросы. Но где и как можно получить информацию для решения таких вопросов, вполне вписывается в ту область, которой занимается информатика. Именно поэтому мы предлагаем выполнить лабораторную работу № 19, в которой предусмотрено знакомство учащихся с соответствующими информационными ресурсами. Скорее всего, учащиеся впервые именно при выполнении этой работы столкнутся с такой непростой вещью, как составление собственного резюме. Мы не случайно предлагаем делать это не в личном порядке, а в обсуждении с товарищами. Вполне может получиться так, что именно товарищ укажет на те способности и личные особенности автора резюме, которые будут характеризовать его с лучшей стороны.

Задумывая эту лабораторную работу, мы, конечно же, имели в виду и весьма существенный воспитательный аспект — научить видеть в человеке то хорошее, что есть обязательно в каждом из нас. Очень важно, чтобы здесь не началась лесть по принципу «Кукушка хвалит петуха за то, что хвалит он кукушку». Нам представляется, что творческому Учителю здесь есть где развернуть свои способности.

Материал об интернет-телефонии (§ 43) носит сугубо иллюстративный характер. Его можно предложить для самостоятельного чтения, а при остром дефиците времени пропустить.

Совместное обсуждение темы этики Интернета и его опасностей является, на наш взгляд, наиболее удачной формой изучения этих вопросов. Не следует смаковать неэтичные формы поведения, помня, что дурные примеры заразительны. Но проявлять инициативу в самозащите и защите сети от неэтичного и тем более противозаконного поведения необходимо. Важный вывод, который должны сделать для себя учащиеся после изучения этой темы: каждый клиент сети несет ответственность за качество помещаемой им в сеть информации и не должен допускать действий, представляющих опасность для функционирования сети. В частности, учащиеся должны знать, что при обнаружении вируса пользователь должен отключиться от сети и предпринять антивирусные меры. В случае работы в корпоративной локальной сети необходимо также незамедлительно сообщить об обнаружении вируса администратору сети. Более подробно эти вопросы представлены нами в § 44 учебника.

Бытует мнение, что учащиеся на практике должны применять имеющиеся в распоряжении компьютерного класса антивирусные средства. Мы оставляем это на усмотрение учителя, но никакой лабораторной работы по данному вопросу нами не предусмотрено.

В настоящее время вопросы информационной безопасности стали рассматриваться не только как внутренняя проблема сетевого сообщества, но и как проблема устойчивого существования человеческого общества в целом. Виртуализация экономики, размещение в виртуальном пространстве значительных информационных ресурсов — все это привело к тому, что преднамеренные или даже непреднамеренные нарушения функционирования этого виртуального мира могут привести к тяжелым, а иногда и катастрофическим последствиям. В § 45 приводится соответствующая терминология, позволяющая на внутригосударственном и межгосударственном уровнях формулировать условия и требования к информационной безопасности субъектов информационных отношений. В заданиях к этому параграфу основное внимание уделено обсуждению вопросов, к каким последствиям для экономической деятельности могут привести те или иные нарушения требований информационной безопасности. Однако экономика — далеко не единственная (хотя и очень важная) сфера жизни человеческого общества. Поэтому мы приветствовали бы инициативу учителя обсудить с учащимися вопросы, аналогичные по своей постановке тем, которые сформулированы в заданиях 3 и 4, но применительно, например, к политической деятельности или научной сфере.

Заключительный параграф этой главы предлагает еще раз взглянуть на защиту информации с различных точек зрения: правовой, этической, технологической. В частности, в заданиях 4 и 5 к этому параграфу настоятельно проводится мысль, что преднамеренные действия, приведшие для субъектов информационного взаимодействия к негативным эффектам, являются не только противозаконными, но и уголовно наказуемыми.

Что касается технологической информации о вирусах и антивирусной защите, в настоящее время имеется много источников, в которых представлена соответствующая информация. Надо предложить учащимся подготовить сообщения по данной теме и обсудить их в той или иной форме.

Наконец, в задании 6 к § 46 предлагается еще раз, уже после многократных обсуждений различных аспектов информационной безопасности, высказаться на тему этики в Интернете.

ГЛАВА 5

Исследование алгоритмов математическими методами

Материал этой главы предназначен исключительно для профильного курса информатики. Его освоение требует от учащихся весьма значительной математической и логической подготовки. Тем не менее Федеральный компонент образовательного стандарта по информатике в профильном варианте предписывает изучение таких вопросов, как доказательство правильности алгоритмов, их выполнимости (в частности, конечности), результативности и т. п. Мы надеемся, что смогли в нашем учебнике рассказать об этом на достаточно доступном для школьников языке. В то же время мы понимаем, что перед учителем стоит весьма нелегкая задача помочь учащимся в освоении совершенно новой для них области теоретической информатики — теории алгоритмов. Скажем прямо, широкого опыта в преподавании данной темы не существует, все наши рекомендации опираются на практику преподавания в специализированном математико-информационном классе одной из школ г. Екатеринбурга. Тем не менее мы полагаем, что наш опыт окажется полезным.

Тема 12. Математические методы исследования алгоритмов

Разумеется, алгоритмы вовсе не изобретение XX в., когда появились первые электронно-вычислительные машины. Они даже намного древнее своего названия, произошедшего от искажения слов аль-Хорезми. Некоторые из алгоритмов навсегда ушли из человеческой практики. Никто, например, сейчас не производит действия с дробями по тем алгоритмам, которые применялись древними египтянами. Другие, столь же давно изобретенные алгоритмы и сегодня играют важную роль. Примером тому служит всем известный алгоритм Евклида вычисления наибольшего общего делителя натуральных чисел (Евклид и предположить не мог, что придуманный им способ будет называться алгоритмом). А вот сказать, что такое алгоритм, оказалось непросто. А надо. И дело не только в том, что требуется как-то отличать алгоритмы от всяческих инструкций другого рода. Важнее уметь для имеющейся задачи определять, существует ли алгоритм, ее решающий. Обычно если алгоритм есть и нам его предъявляют, то ни у кого нет сомнения в том, что все в порядке[1]. А вот как доказать отсутствие алгоритма, если не знаешь, что это такое?

Математика и здесь оказалась наиболее востребованной. Основных причин для этого, по-видимому, две. Во-первых, само это понятие формировалось именно в сфере действий над математическими объектами: числами (как целыми, так и дробными), алгебраическими выражениями, геометрическими фигурами и величинами. Во-вторых, именно в математике развивались методы строгих доказательств, приведшие в конечном счете к появлению особой дисциплины — математической логики.

Довольно стандартное определение алгоритма, которое бытует в школьных учебниках информатики, таково: алгоритм — это последовательность допустимых для некоторого исполнителя действий, направленная на достижение определенной цели. Вполне пристойное определение, ничуть не хуже, чем определения, даваемые в большинстве естественно-научных дисциплин[2]. Проблема, однако, в том, что при таком определении существование или несуществование алгоритма зависит от выбора исполнителя. Вот иллюстрация этой ситуации на геометрическом материале.

Всем известен алгоритм построения с помощью циркуля и линейки перпендикуляра к прямой, проходящего через данную точку. Слова «с помощью циркуля и линейки» — это и есть краткое задание допустимых действий исполнителя. А вот если исполнитель вооружен только одной линейкой, то нетрудно доказать, что с помощью и при посредстве только этого инструмента построить перпендикуляр уже невозможно. В свою очередь, с помощью циркуля и линейки нельзя решить задачу удвоения куба, т. е. построить куб, объем которого в точности в 2 раза больше

---

[1] Сомнения, конечно, возникают, тот ли результат получился после исполнения данного алгоритма, при любых ли начальных данных он работает и т. п. Но это сомнения иного рода: никто не сомневается в том, что предъявленный нам объект — это алгоритм.

[2] Вот примеры: «Электромагнитная волна — возмущение электромагнитного поля, распространяющееся в пространстве» (Касьянов В. А. Физика: 11 класс. — М.: Дрофа, 2002), «Мельчайшие структуры, способные к самовоспроизведению, называются клетками» (Рувинский А. О. и др. Общая биология: 10—11 классы. — М.: Просвещение, 1993), «Вирус — это неклеточная форма жизни» (там же, хотя каждому здравомыслящему человеку видно, что две последние фразы абсолютно противоречат друг другу).

объема данного куба. Но если к циркулю и линейке добавить еще какие-либо инструменты (т. е. расширить список допустимых действий), то эта задача станет разрешимой. Создается ощущение, что вопрос о существовании алгоритма, решающего ту или иную задачу, — это вопрос о наборе допустимых действий используемого исполнителя.

Ощущение это оказывается, однако, обманчивым [1]. И чтобы найти путь, который позволил бы обсуждать алгоритмы, не обращаясь явно к понятию исполнителя, необходимо изучить общие свойства алгоритмов.

Методологической подоплекой появления понятия «свойства алгоритмов» является идея объявить, что сам алгоритм — понятие неопределяемое. В «Математической энциклопедии» в статье «Алгоритм» про него именно так и сказано [2]. Аналогия с другими математическими теориями (например, геометрией), где существование неопределяемых понятий заложено в основу соответствующей теории, а понятия описываются своими свойствами и связывающими их соотношениями — аксиомами [3], подсказывает идею описать понятие алгоритма через его свойства. Именно она легла в основу большинства школьных учебников информатики (и нашла свое отражение в стандарте). Список свойств невелик: дискретность, массовость, конечность, результативность, детерминированность [4]. Однако далеко не всегда бывает легко убедиться в том, что предложенный алгоритм обладает требуемым свойством и дает нужный результат. Но сначала расшифруем, что означает каждое из свойств.

Дискретность. Под дискретностью понимается то, что алгоритм состоит из описания последовательности тактов обработки, организованной таким образом, что в начальный момент задается исходная ситуация, а в каждый следующий момент ситуация преобразуется на основе данных, полученных в предшествующие такты обработки. Дискретность алгоритма означает, что он исполняется по шагам: каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.

---

[1] Уже одно это заявление может вызвать шок, поскольку школьникам практически во всех учебниках твердят, что нельзя определить алгоритм без определения исполнителя.

[2] См.: Математическая энциклопедия / Под ред. И. М. Виноградова. — М.: Советская энциклопедия, 1977. — Т. 1. — С. 205.

[3] Одна из наиболее известных аксиом, связывающая понятия точки и прямой, такова: через любые две точки можно провести прямую, и притом только одну.

[4] Этот список свойств учащиеся должны знать, по крайней мере, с 10 класса — он приведен в нашем учебнике в § 7, который так и называется «Алгоритмы и их свойства».

Детерминированность. Это свойство означает, что на каждом шаге исполнения алгоритма (см. свойство дискретности) однозначно определено преобразование объектов среды исполнителя, полученных на предшествующих шагах алгоритма. По-другому это свойство алгоритма называют **точностью**.

Результативность. Это свойство подразумевает, что каждое действие в алгоритме после своего завершения создает ситуацию, в которой все имеющиеся объекты однозначно определены. Если это по каким-либо причинам невозможно, то алгоритм должен сообщить, что решения задачи не существует. Это свойство является обязательным для любого алгоритма.

Обычно говорят о результативности алгоритма в целом, подразумевая, что алгоритм предназначен для решения той или иной задачи. Как правило, одновременно с результативностью в целом подразумевается и **конечность** алгоритма, т. е. завершение его работы за конечное число шагов (при этом количество шагов может быть заранее неизвестным и различным для разных начальных ситуаций).

То свойство алгоритма, что в нем могут использоваться только допустимые действия исполнителя, нередко называют **понятностью** алгоритма (иногда **эффективностью** или **элементарностью** действий). Понятность является обязательным свойством любого алгоритма.

Массовость. С точки зрения практической ценности алгоритма важно, чтобы множество допустимых начальных ситуаций было для него достаточно большим (иногда говорят: потенциально бесконечным). Например, известный из начальной школы алгоритм сложения столбиком двух натуральных чисел является массовым, поскольку применим к любой паре натуральных чисел.

Несмотря на вышесказанное, мы хотим предостеречь учителя от чрезмерного внимания к изучению свойств алгоритмов. На это есть две причины.

Во-первых, развитие компьютерной техники, теории алгоритмов и практики программирования вносит свои коррективы. Теперь, к примеру, рассматривается не только последовательное, но и параллельное исполнение шагов алгоритма (ср. 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. Пример такого алгоритма приведен в Математической энциклопедии [2] и воспроизведен нами в § 47 под именем Преобразование_слов. И примеры слов, которые фигурируют в тексте этого параграфа при обсуждении указанного алгоритма, тоже взяты из Математической энциклопедии.

Во-вторых, в общей теории алгоритмов нет и речи ни о каких свойствах. Они появляются только тогда, когда четко сформулирована задача, для которой надо установить наличие или отсутствие алгоритма с заданными свойствами. Поэтому при применении понятия «алгоритм» в конкретной ситуации оно соответствующим образом уточняется. С этой целью в понятии алгоритма выделены семь параметров, и в зависимости от того, как формализуются эти параметры, получается то или иное уточненное понятие алгоритма. Мы не будем сейчас вдаваться в подробности определения этих семи параметров, а заинтересовавшихся читателей отсылаем к уже неоднократно упоминавшейся статье «Алгоритм» в Математической энциклопедии.

Фактически 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:

[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 (см. сноску на с. 204 (4)) 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.

все алгоритмы, составляемые учащимися в ходе изучения курса информатики, должны обладать всеми перечисленными свойствами.

Прокомментируем теперь задания к § 47. Чтобы ответить на вопрос задания 1, учащиеся должны четко понимать, что значит дать определение. Это означает объяснить значение нового понятия через уже ранее определенные. Сказать, что все понятия, фигурирующие в объяснении, что такое алгоритм, можно считать ранее определенными, очевидно, нельзя. Более того, данное рассуждение показывает, что в любой научной дисциплине — физике, химии, биологии, филологии и т. д. — неизбежно присутствуют неопределяемые понятия. На наш взгляд, было бы весьма полезно обсудить с учащимися эту ситуацию и даже предложить им привести примеры неопределяемых (на самом деле!) понятий.

Ответ на вопрос, сформулированный в задании 2, приведен в объяснительном тексте параграфа.

Приведем решение задачи 3а. Для этого последовательно исполним алгоритм для каждого из восьми трехбуквенных слов. Мы будем записывать результаты преобразования входного слова в теле цикла в виде цепочки.

$aaa \to a$ (условие продолжения цикла не выполнено, тело цикла не исполняется ни разу);

$aab \to b$ (условие продолжения цикла не выполнено, тело цикла не исполняется ни разу);

$aba \to bab \to baba \to baaba \to abaaba \to baabab \to ababa$-$ba \to \ldots$ (легко понять, что в ходе указанных преобразований не может появиться слово, начинающееся с сочетания *aa*, следовательно, алгоритм никогда не прекратит работу);

$abb \to bbb \to \ldots$ (абсолютно ясно, что слово, начинающееся с сочетания *bb*, меняться в теле цикла не будет, в то же время условие продолжения цикла остается все время выполненным, так что алгоритм никогда не прекратит работу);

$baa \to aaba \to ba$ (алгоритм завершил работу);

$bab \to \ldots$ (это сочетание мы уже встречали выше);

$bba \to \ldots$ (ситуация с двумя буквами *b* в начале уже обсуждалась);

$bbb \to \ldots$ (эта ситуация уже тоже рассмотрена).

Итак, к словам *aaa*, *aab*, *baa* алгоритм применим, а к остальным нет.

В аналогичной форме приведем решение задачи 3б, только здесь слов будет уже 16.

$aaaa \to aa$ (условие продолжения цикла не выполнено, тело цикла не исполняется ни разу);

$aaab \to ab$ (условие продолжения цикла не выполнено, тело цикла не исполняется ни разу);

$aaba \to ba$ (условие продолжения цикла не выполнено, тело цикла не исполняется ни разу);

aabb → *bb* (условие продолжения цикла не выполнено, тело цикла не исполняется ни разу);

abaa → *baab* → *ababa* → *babab* → *bababa* → *babaaba* → ... (легко понять, что в ходе указанных преобразований не может появиться слово, начинающееся с сочетания *aa*, следовательно, алгоритм никогда не прекратит работу);

abab → *babb* → *bbaba* → ... (слово, начинающееся с сочетания *bb*, в теле цикла не меняется, в то же время условие продолжения цикла остается все время выполненным, так что алгоритм никогда не прекратит работу);

abba → *bbab* → ... (слово, начинающееся с сочетания *bb*, в теле цикла не меняется, в то же время условие продолжения цикла остается все время выполненным, так что алгоритм никогда не прекратит работу);

abbb → *bbbb* → ... (слово, начинающееся с сочетания *bb*, в теле цикла не меняется, в то же время условие продолжения цикла остается все время выполненным, так что алгоритм никогда не прекратит работу);

baaa → *aaaba* → *aba* (алгоритм завершил работу);

baab → ... (это сочетание мы уже рассмотрели выше);

baba → *baaba* → *abaaba* → *baabab* → *abababa* → ... (с этим сочетанием мы встречались в пункте *а*);

babb → *bbaba* → ... (слово, начинающееся с сочетания *bb*, в теле цикла не меняется, в то же время условие продолжения цикла остается все время выполненным, так что алгоритм никогда не прекратит работу).

Оставшиеся четыре слова *bbaa*, *bbab*, *bbba* и *bbbb* начинаются с *bb*, и потому алгоритм никогда не прекратит работу.

Ответ: алгоритм применим к словам *aaaa*, *aaab*, *aaba*, *aabb* и *baaa*.

В задании 4 приведен пример недетерминированного алгоритма — ведь при каждом исполнении тела цикла выбор плодов не определен. Но давайте исследуем, как меняется количество бананов на дереве, когда Садовник срывает два плода. Пусть x — количество бананов перед очередным исполнением тела цикла. Тогда возможны три варианта.

1) Садовник срывает два апельсина. Тогда вырастает апельсин, а количество бананов не меняется, т. е. остается равным x.

2) Садовник срывает два банана. Тогда вырастает апельсин, а количество бананов уменьшается на 2, т. е. становится равным $x - 2$.

3) Садовник срывает один апельсин и один банан. Тогда вырастает банан. Следовательно, количество бананов не изменилось, т. е. осталось равным x.

Эти рассмотрения показывают, что количество бананов на дереве при исполнении тела цикла не меняет своей четности (если оно было нечетным, то нечетным и останется;

если же было четным, то останется четным). В конце исполнения алгоритма остается один плод. Если первоначально количество бананов было четным, то и после исполнения алгоритма оно должно быть четным, т. е. последний плод — это не банан. Следовательно, это апельсин. Если же первоначально количество бананов было нечетным числом, то и в конце количество бананов должно быть нечетно, т. е. тот единственный плод — это банан. Итак, вид плода после исполнения алгоритма зависит только от того, четно или нечетно было количество бананов изначально.

В решении этой задачи важную роль сыграло найденное нами свойство — сохранение четности числа бананов. Такое свойство, которое не меняется при исполнении алгоритма, называют **инвариантом**. Подробно понятие инварианта обсуждается в § 50, однако ввести его можно и нужно уже при рассмотрении этой задачи.

Отметим еще одно обстоятельство. Мы доказали, что результат данного алгоритма зависит только от начальных данных и не зависит от исполнения алгоритма. Как было сказано выше, такой алгоритм называется однозначным.

Задание 5 имеет ту же природу, что и задание 4, — в нем сформулирован недетерминированный алгоритм, и требуется проверить, будет ли он однозначным. Но это задание намного легче предыдущего: нетрудно догадаться, что ответом здесь всегда будет сумма всех записанных чисел, увеличенная на $n - 1$, где n — количество чисел, записанных первоначально на доске. Мы рекомендуем эту задачу решить самостоятельно дома.

В задании 6 положительный ответ на вопрос, является ли предложенный алгоритм однозначным, получается из других соображений. Здесь надо проанализировать условие окончания цикла. Алгоритм закончит свою работу в том и только в том случае, если все тома сочинений В. Скотта будут расположены в порядке возрастания. Ясно, что есть только один вариант расположения томов в порядке возрастания, так что результат исполнения алгоритма определен однозначно.

В задании 7 мы снова возвращаемся к вопросам нахождения области применимости того или иного алгоритма.

В задании 7а исходным данным является целое число N. Ясно, что алгоритм применим к любому натуральному числу. Ответ: множество всех целых чисел. Здесь можно услышать возражение, что цикл будет исполняться лишь при натуральных значениях N. Но и в этом случае алгоритм завершает работу, сообщая нулевое значение для S.

В задании 7б в роли начальных данных может выступать любая последовательность символов. Результат исполнения алгоритма имеет смысл лишь в случае, когда эта последовательность является именем человека, но надо напомнить учащимся, что любой алгоритм исполняется формально, т. е. без учета смысла обрабатываемой информации.

В задании 7в единственным ограничением является возможность вычисления дроби $1/(a + d*K)$, а именно: знаменатель не должен обращаться в 0. Запишем это условие, подставив вместо d выражение $(b - a)/N$, получается $a + \frac{b-a}{N} K \neq 0$. Выразив из этого неравенства K, имеем $K \neq \frac{aN}{a-b} = \frac{N}{1-\frac{b}{a}}$. Если число $\frac{N}{1-\frac{b}{a}}$ является целым и заключено в пределах от 1 до N, то при одном из выполнений тела цикла произойдет остановка исполнения алгоритма из-за деления на 0. Но выраженные ограничения на исходные данные a, b и N не являются хорошей формой ответа. Давайте порассуждаем. Чтобы указанное число было положительным и меньшим, чем N, необходимо и достаточно, чтобы число $-\frac{b}{a}$ было положительным. А чтобы число $\frac{N}{1-\frac{b}{a}}$ было целым, необходимо (но недостаточно!), чтобы число $-\frac{b}{a}$ было рациональным. Пусть $-\frac{b}{a}$ представлено несократимой дробью $\frac{p}{q}$, где p и q — натуральные числа. Тогда $\frac{N}{1-\frac{b}{a}} = \frac{N}{1+\frac{p}{q}} = \frac{qN}{p+q}$. Поскольку p и q — взаимно простые числа (ибо дробь $\frac{p}{q}$ несократима), число $\frac{qN}{p+q}$ будет целым тогда и только тогда, когда $p + q$ является делителем числа N. Теперь можно описать множество тех наборов значений переменных, которые не являются допустимыми в качестве исходных данных для данного алгоритма. Если N целое неположительное, то допустимыми будут любые значения a и b, поскольку цикл не будет исполняться ни разу. Пусть N — натуральное число и m — его натуральный делитель. Выберем произвольное натуральное число q, не превосходящее m. Положим $p = m - q$. Тогда допустимыми будут такие пары чисел a и b, для которых $\frac{b}{a} \neq -\frac{p}{q}$.

После этих довольно абстрактных рассуждений полезно рассмотреть какой-либо конкретный пример. Пусть $N = 6$, тогда возникают четыре возможных варианта: $m = 1$, $m = 2$, $m = 3$ и $m = 6$.

При $m = 1$ для q есть только одна возможность: $q = 1$. В таком случае $p = 0$. Но тогда и $b = 0$. Ясно, что в этом случае уже для $K = 1$ в теле цикла возникает ситуация деления на 0, каким бы ни было значение a.

При $m = 2$ для q есть две возможности: $q = 1$ и $q = 2$. При $q = 2$ снова $p = 0$ и снова $b = 0$, а этот случай уже рассмотрен. И вообще случай, когда $p = 0$, можно больше не рассматривать, поскольку ответ всегда будет один: $b = 0$, a — любое. Если $q = 1$, то $p = 1$. Тогда $a = -b$, деление на 0 возникает при $K = 3$.

При $m = 3$ для q есть три возможности: $q = 1$, $q = 2$ и $q = 3$. При $q = 3$ снова $p = 0$. Если $q = 1$, то $p = 2$. Тогда $b = -2a$, деление на 0 возникает при $K = 2$. Если же $q = 2$, то $p = 1$. Тогда $b = -\frac{1}{2}a$; деление на 0 возникает при $K = 4$.

При $m = 6$ для q есть шесть возможностей: $q = 1$, $q = 2$, $q = 3$, $q = 4$, $q = 5$ и $q = 6$. Соответственно для p получаются значения $p = 5$, $p = 4$, $p = 3$, $p = 2$, $p = 1$, $p = 0$. Последнее значение новой ситуации не дает, а для остальных получаются следующие соотношения между a и b:

$$b = -5a;\ \ b = -2a;\ \ b = -a;\ \ b = -\frac{1}{2}a;\ \ b = -\frac{1}{5}a.$$

Часть вариантов нам уже встречалась, новых здесь ровно два — первый и последний.

Поскольку a и b — любые действительные числа, пару $(a; b)$ удобно представлять точкой на координатной плоскости с абсциссой a и ординатой b. Тогда область применимости алгоритма при $N = 6$ будет представлена так, как это показано на рисунке 5.1 (как это принято в математике, пунктирными линиями обозначены множества точек, которые надо удалить из плоскости).

Вообще говоря, поскольку исходные данные представлены тремя переменными: a, b и N, то и область применимости этого алгоритма представляет собой ту часть трехмер-

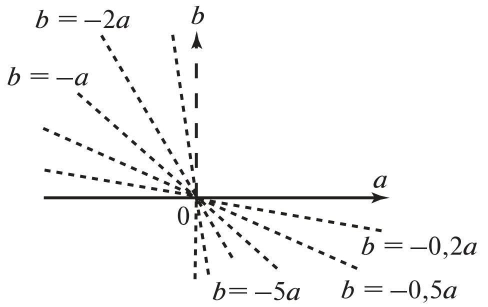


Рис. 5.1

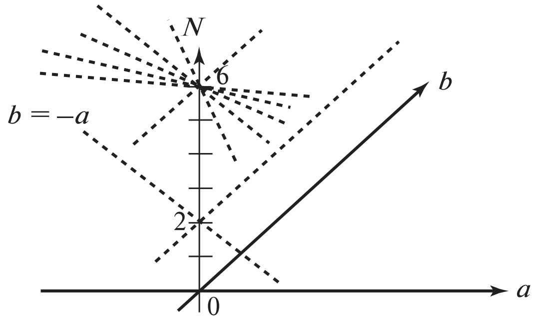


Рис. 5.2

ного пространства, в котором для каждого натурального значения N в соответствующей плоскости, параллельной плоскости $a0b$, удалены точки нескольких прямых (см. рис. 5.2, на котором изображены лишь изученная нами плоскость при $N = 6$ и плоскость при $N = 2$).

В задании 7г учащиеся должны установить, для каких положительных значений a подкоренное выражение будет положительным. Классическое неравенство между средним арифметическим и средним геометрическим гласит, что $\frac{a^x + a^{-x}}{2} \geqslant \sqrt{a^x a^{-x}} = 1$, причем равенство достигается тогда и только тогда, когда $a^x = a^{-x}$, т. е. $a = 1$ или $x = 0$. Если же a неположительно, то ветвление исполняться не будет и алгоритм завершит работу без выдачи результата (пользователь в этой ситуации по внешним признакам и без вмешательства в работу компьютера не может определить, закончилось исполнение алгоритма или он зациклился). Следовательно, область применимости этого алгоритма состоит из множества пар $(a;\ 0)$, где a — произвольное положительное число, и пар $(1,\ x)$, где x — произвольное действительное число.

В задании 7д ситуация похожа на ситуацию, представленную в задании 7г. Условие ветвления выбрано так, что алгоритм всегда заканчивает работу, однако результат он сообщает в том и только в том случае, если $\cos x = 1$. Тем самым алгоритм применим для $x = \frac{\pi}{2} + 2\pi k$, k — произвольное целое число.

Такое задание убедительно демонстрирует учащимся, что вопрос о применимости алгоритма к тем или иным данным весьма непрост. А этот вопрос должен быть решен до того, как алгоритм приступит к обработке введенных данных. Этим мотивируется необходимость изучения методов, позволяющих обосновывать применимость алгоритма для тех или иных исходных данных. В § 48 некоторые из этих методов обсуждаются. Но с точки зрения информатики важно не столько освоение учащимися этих методов (вполне достаточно, чтобы они разобрались с тем, как эти методы применяются), сколько понимание полезности и даже необходимости теорем чистого существования. Дело в том, что даже очень сильные в программировании учащиеся (а потом и студенты) нередко высказывают точку зрения, что изучение методов классической математики в информатике (и в программировании) является лишней и бесполезной нагрузкой на их головы. И вообще, для решения любой задачи достаточно, по их мнению, составить программу и отладить ее на компьютере. В § 48 мы демонстрируем, что этого абсолютно недостаточно, если не обосновать заранее, что запрограммированный алгоритм действительно способен решить данную задачу. Можно сказать, что этот параграф несет прежде всего методологическую нагрузку и уже во вторую очередь знакомит учащихся с методами доказательной алгоритмизации.

Мы понимаем, что от учителя, взявшегося изложить учащимся данную тему, потребуется вникнуть в весьма непростые математические рассуждения. Мы постарались сделать их как можно более подробными и прозрачными. Вполне возможно, что сильным учащимся можно поручить рассмотреть эти рассуждения самостоятельно, а затем представить их классу на обсуждение. Учитель в такой ситуации выступает дирижером этого обсуждения.

Перейдем к обсуждению заданий к § 48. При выполнении задания 1 полезно не только получить ответ на сформулированный в нем вопрос, но и узнать мнение учащихся о том, какую роль играют теоремы чистого существования в теории алгоритмов.

В задании 2 учащиеся обычно без труда объясняют, почему данный алгоритм, если, конечно, он сработает, даст ответ на поставленную задачу. Тем не менее полезно уточнить, в какой мере они понимают, почему используется оператор Y := mod (X, 100000) и почему в качестве начального присваивания выбрано Y := 2. Что касается обоснования, почему алгоритм обязательно закончит работу за конечное число шагов, то для этого требуется доказать теорему существования такой степени числа 3, которая будет давать 1 в остатке при делении на 100 000. Приведем формулировку доказательства этой теоремы.

Теорема. Существует такое натуральное число $n > 1$, для которого число 3^n дает остаток 1 при делении на 100 000.

Доказательство. Рассмотрим последовательность чисел $1; 3; 3^2; 3^3; \ldots; 3^{100000}$. В ней 100 001 число. Однако различных остатков при делении на 100 000 ровно 100 000. Поэтому среди указанных чисел обязательно найдутся два числа с одинаковым остатком. Это означает, что разность этих двух чисел делится на 100 000. Пусть одно из этих чисел — число 3^k, а другое — 3^m, причем можно считать, что $k > m$. Тогда их разность $3^k - 3^m = 3^m (3^{k-m} - 1)$ делится на 100 000. Но множитель 3^m не имеет общих делителей с числом 100 000, поэтому на 100 000 делится второй множитель $3^{k-m} - 1$. Роль числа n играет разность $k - m$.

В задании 3 алгоритм аналогичен алгоритму, разобранному в объяснительном тексте параграфа. Здесь запрограммировано вычисление последовательности $X_1 = 1$; $X_2 = 3$; ...; $X_n = X_{n-1} - 2X_{n-2}$ для каждого $n > 2$. Требуется установить, верно ли, что среди членов этой последовательности обязательно встретится число, большее, чем любое наперед заданное число M. Вот первые 15 членов этой последовательности:

1; 3; 1; –5; –7; 3; 17; 11; –23; –45; 1; 91; 89; –93; –271.

Некоторые наблюдения можно сделать уже сейчас. Например, можно заметить, что все числа в этой последовательности нечетные. Обосновать это совсем несложно — ведь каждое следующее число отличается от предыдущего четным слагаемым. В частности, в этой последовательности не может встретиться число 0.

А еще можно заметить, что после двух отрицательных чисел идет три положительных числа, а потом снова два отрицательных. Попытаемся это обосновать. Запишем цепочку равенств:

$$X_n = X_{n-1} - 2X_{n-2} = (X_{n-2} - 2X_{n-3}) - 2X_{n-2} = \\ = -(X_{n-2} + X_{n-3}).$$

Из этого равенства видно, что если имеются два рядом стоящих члена последовательности, имеющие одинаковый знак, то через один шаг в этой последовательности появится число противоположного знака. Поэтому более чем трех подряд идущих членов одного знака быть не может. Значит, в нашей последовательности как угодно далеко встречаются как положительные, так и отрицательные числа. С другой стороны, если после положительного числа идет отрицательное, то следующее снова будет отрицательное, поскольку из отрицательного числа вычитается удвоенное положительное; если же после отрицательного числа идет положительное, то следующее снова будет положительное — из положительного числа вычитается удвоенное отрицательное. Это означает, что около каждого положительного числа по крайней мере один его сосед (число, идущее перед ним, или число, идущее сразу после него) тоже будет положительным числом. И то же самое справедливо для отрицательных чисел: у каждого отрицательного числа по крайней мере один сосед тоже отрицательное число. Так что числа одного знака идут в этой последовательности парами или тройками.

Пусть теперь X_n и X_{n-1} — два члена последовательности, имеющие одинаковый знак. Тогда, как показано выше, $X_{n+2} = -(X_n + X_{n-1})$, а значит, $|X_{n+2}| = |X_n + X_{n-1}| > \max\{|X_n|; |X_{n-1}|\}$, поскольку в последовательности нет нулей. Но рядом с X_{n+2} находится член последовательности того же знака, что и X_{n+2}. Поэтому еще через 2 или 3 члена будет элемент последовательности, абсолютная величина которого теперь уже строго больше, чем $|X_{n+2}|$, а по знаку этот член будет снова противоположен X_{n+2}. Нетрудно понять, что тогда как среди положительных членов последовательности, так и среди отрицательных будут элементы, как угодно большие по абсолютной величине. Для положительных чисел это как раз и означает, что найдется член последовательности, который будет больше числа M, какое бы M ни было выбрано первоначально.

В задании 4 сформулировать задачу, для решения которой предназначен алгоритм, несложно: «Указать наименьшее натуральное число N, для которого сумма чисел, обратных квадратам всех натуральных чисел от 1 до N, не меньше 20». Чтобы установить, конечен ли алгоритм, надо прежде всего убедиться, что такое N существует. Обычно учащиеся опрометчиво высказывают предположение, что алгоритм конечен, поскольку сумма увеличивается после каждого исполнения тела цикла. Им кажется очевидным, что на каком-то шаге эта сумма обязательно станет больше 20 (и вообще любого наперед заданного положительного числа). Однако это не так. Обозначим через S_K сумму первых K слагаемых, тогда можно записать такое равенство:

$$S_N = 1 + 1/2^2 + 1/3^2 + 1/4^2 + 1/5^2 + \ldots + 1/(N-1)^2 + 1/N^2.$$

Из него нетрудно видеть, что

$$\begin{gathered} S_N < 1 + 1/(1 \cdot 2) + 1/(2 \cdot 3) + 1/(3 \cdot 4) + \ldots + \\ + 1/(N-2)(N-1) + 1/(N-1)N = \\ = 1 + (1/1 - 1/2) + (1/2 - 1/3) + (1/3 - 1/4) + \ldots + \\ + (1/(N-2) - 1/(N-1)) + (1/(N-1) - 1/N) = 2 - 1/N < 2. \end{gathered}$$

Значит, какое бы N мы ни взяли, сумма S не превзойдет 2, т. е. требуемого N не существует, а алгоритм зациклится.

Задание 5 совершенно аналогично заданию 4. Поэтому в пункте *а* ответ можно сформулировать так: «Указать наименьшее натуральное число N, для которого сумма чисел, обратных всем натуральным числам от 1 до N, не меньше заданного числа M». Однако, столкнувшись в задании 4 с ситуацией, когда сумма как угодно большого числа положительных слагаемых все-таки остается ограниченной некоторой константой, учащиеся уже осторожнее высказывают свои гипотезы. Ответ в пункте *б* положителен: при любом значении M алгоритм за конечное число шагов закончит работу. Чтобы обосновать это, запишем следующую цепочку неравенств:

$$\frac{1}{2^n+1}+\frac{1}{2^n+2}+\ldots+\frac{1}{2^{n+1}}>\frac{1}{2^{n+1}}+\frac{1}{2^{n+1}}+\ldots+\frac{1}{2^{n+1}}=$$
$$=\frac{2^n}{2^{n+1}}=\frac{1}{2}$$

(здесь в левой части записано 2^n слагаемых, каждое из которых потом заменено на меньшее число $\frac{1}{2^{n+1}}$). Но тогда

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{8}+\ldots+\frac{1}{2^n}+\frac{1}{2^n+1}+\frac{1}{2^n+2}+\ldots+\frac{1}{2^{n+1}}=$$
$$=1+\frac{1}{2}+\left(\frac{1}{3}+\frac{1}{4}\right)+\left(\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{8}\right)+\ldots+$$
$$+\left(\frac{1}{2^n+1}+\frac{1}{2^n+2}+\ldots+\frac{1}{2^{n+1}}\right)>$$
$$>1+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\ldots+\frac{1}{2}=1+\frac{n+1}{2}>M$$

при $n > 2M - 3$.

Следовательно, сумма чисел, обратных натуральным числам, становится больше, чем любое наперед заданное число M.

В задании 6 учащиеся прежде всего должны увидеть, что функция СКВ подсчитывает сумму квадратов цифр заданного числа. Тогда становится ясным, что в теле цикла алгоритма Преобразование строится последовательность чисел, в которой каждое последующее число равно сумме квадратов цифр предыдущего числа. Исследуем поведение этой последовательности для всех вариантов первого числа.

1) $1 \to 1 \to 1 \to \ldots$

В этом случае процесс будет продолжаться бесконечно.

2) $2 \to 4 \to 16 \to 37 \to 58 \to 89 \to 145 \to 42 \to 20 \to 4 \to \ldots$

Дальше пойдет повторение тех преобразований, что были после числа 4.

3) $3 \to 9 \to 81 \to 65 \to 61 \to 37 \to \ldots$

Дальше пойдет повторение тех преобразований, что были после числа 37 в предыдущей последовательности.

4) $5 \to 25 \to 29 \to 85 \to 89 \to \ldots$

Во-первых, уже не надо исследовать число 4, а сразу можно заняться числом 5. Во-вторых, ясно, что в этой последовательности дальше пойдет повторение тех преобразований, что были после числа 89 в последовательности 2.

5) $6 \to 36 \to 45 \to 41 \to 17 \to 50 \to 25 \to \ldots$

Дальше пойдет повторение тех преобразований, что были после числа 25 в предыдущей последовательности.

6) $7 \to 49 \to 97 \to 130 \to 10 \to 1 \to \ldots$

Дальше пойдут единицы.

7) $8 \to 64 \to 52 \to 29 \to \ldots$

Число 2 уже было в последовательности 4.

8) $11 \to 2 \to \ldots$

Уже было...

9) $12 \to 5 \to \ldots$

Уже было...

10) $13 \to 10 \to \ldots$

Ничего нового...

Число 14 то же самое, что 41, а 41 уже было.

11) $15 \to 26 \to 40 \to 16 \to \ldots$

Числа 16, 17, 18 (точнее, 81) уже были.

12) $19 \to 82 \to 68 \to 100 \to 1 \to \ldots$

Теперь сделаем некоторые наблюдения. Во-первых, если одно число отличается от другого лишь перестановкой цифр, то последовательности для них совпадают. Во-вторых, если одно число отличается от другого тем, что в каком-то разряде стоит 0, то последовательности для них совпадают. В-третьих, для каждого из чисел, с которого начинается последовательность, эту последовательность надо строить до того момента, пока в ней не встретится число, уже раньше присутствовавшее в какой-либо последовательности. Эти замечания позволяют для многих чисел сразу сказать, как ведет себя последовательность, которую они определяют.

А теперь в таблице 5.1 вычеркнем те числа, которые у нас либо уже встретились, либо удовлетворяют высказанным замечаниям.

Таблица 5.1

~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~
~~11~~	~~12~~	~~13~~	~~14~~	~~15~~	~~16~~	~~17~~	~~18~~	~~19~~	~~20~~
~~21~~	22	23	~~24~~	~~25~~	26	27	~~28~~	29	~~30~~
~~31~~	32	33	34	35	36	~~37~~	38	39	~~40~~
~~41~~	~~42~~	43	44	~~45~~	~~46~~	47	48	~~49~~	~~50~~
~~51~~	~~52~~	53	~~54~~	55	~~56~~	57	~~58~~	59	~~60~~
~~61~~	62	~~63~~	~~64~~	~~65~~	66	67	68	69	~~70~~
~~71~~	72	~~73~~	74	75	76	77	78	~~79~~	~~80~~
~~81~~	~~82~~	83	84	~~85~~	86	87	88	89	~~90~~
~~91~~	92	93	~~94~~	95	96	~~97~~	98	99	~~100~~

Даже в этой таблице еще много необследованных чисел. А за ее пределами таких чисел бесконечно много. Но давайте порассуждаем. Пусть число x записано с помощью n цифр. Тогда сумма квадратов его цифр не превосходит $81n$, поскольку каждая цифра не больше 9. Рассмотрим неравенство $10^{n-1} > 81n$. Нетрудно показать (например, с помощью математической индукции), что оно справедливо для всех натуральных $n \geqslant 4$. Но тогда получаем, что $x \geqslant 10^{n-1} > 81n$. Следовательно, если x четырехзначно или еще больше, то его сумма квадратов обязательно меньше чем x. Иными словами, последовательность, начинающаяся с такого x, будет убывающей по крайней мере до тех пор, пока в ней не появится трехзначное число. Но если x трехзначно, то следующий член последовательности не превосходит $81 \cdot 3 = 243$. Таким образом, для трехзначных чисел мы должны исследовать только те числа, которые не превосходят 243. Среди таких трехзначных чисел наибольшую сумму квадратов цифр имеет число 199. Эта сумма равна 163. Значит, обследовать надо только числа, не превосходящие 163. Среди таких чисел наибольшую сумму квадратов имеет число 99. Она равна 162. Дальше подобными рассуждениями уменьшить диапазон обследуемых чисел уже не удается. Значит, надо обследовать все числа от 1 до 162. Мы уже немало сделали. Но еще есть над чем поработать.

Обследовать эти числа вручную (даже вооружившись калькулятором) — дело трудоемкое. Но можно составить программу, ведь теперь мы точно знаем, для каких чисел надо строить последовательность. Таблицу придется дополнить еще тремя строками (не войдут только числа 161 и 162, для них можно дать ответ отдельно). Мы предлагаем такому обследованию посвятить часть компьютерного практикума. В этом случае надо обратить внимание учащихся, что мы исследуем один алгоритм с помощью другого алгоритма и его реализации на компьютере. На самом деле это один из примеров того, как применяется компьютер к решению теоретических вопросов математики.

В результате выяснится, что, с какого бы числа ни начиналась последовательность, она либо стабилизируется на числе 1, либо попадает в цикл

$$4 \to 16 \to 37 \to 58 \to 89 \to 145 \to 42 \to 20 \to 4 \to \ldots$$

Тем самым, чтобы алгоритм, рассматриваемый в задании 6, был конечным, необходимо и достаточно, чтобы одно из чисел a и b было равно 1, а другое равнялось бы какому-то числу из указанного цикла.

Задание 7 обычно вызывает меньшие трудности, чем предыдущие задания. Поэтому мы предлагаем задать его домой. При нечетных K и M алгоритм завершает работу, даже не входя в тело цикла. Рассмотрим оставшиеся три комбинации четности и нечетности K и M. В каждом из этих случаев тело цикла будет исполняться.

Пусть оба числа четны. Тогда после исполнения тела цикла возможны два варианта: либо M станет нечетным, а K не изменится и потому останется четным, либо M станет четным, а K станет либо четным, либо нечетным. В любом случае хотя бы одно из чисел K или M будет четным.

Пусть K нечетно, а M четно. Тогда M останется четным, т. е. снова хотя бы одно из двух чисел после исполнения цикла будет четным.

Наконец, пусть K четно, а M нечетно. Тогда либо M станет четным, и неважно, каким по четности станет K, либо M останется нечетным, но тогда K останется прежним, т. е. четным.

Итак, в любом случае после исполнения тела цикла хотя бы одно из чисел K или M остается четным. Следовательно, алгоритм никогда не закончит свою работу.

Лабораторная работа № 20, которая сопровождает § 48, рассчитана на два урока. Выполняя ее, учащиеся будут вынуждены бороться с теми эффектами компьютерной арифметики, которые они наблюдали, изучая материал главы 2. В заданиях 1 и 2 им предстоит преодолевать трудности, связанные с ограниченностью разрядной сетки для переменных целого типа, а в задании 5 они будут бороться с потерей точности для переменной вещественного типа. Забегая вперед, скажем, что в этом задании надо реорганизовать алгоритм так, чтобы суммирование шло от меньших слагаемых к большим. Такой прием обсуждался в лабораторной работе № 6 при выполнении заданий 10—12.

Прокомментируем еще задание 4 из этой лабораторной работы. Функция $P\ (x)$, как нетрудно догадаться, рекурсивно подсчитывает произведение цифр числа. Важно понять, что в последовательности значений переменной y, которая вычисляется в основном алгоритме, обязательно встретится число, одна из цифр которого будет равна 0. А тогда произведение цифр окажется равным нулю, последовательность стабилизируется на этом числе, и исполнение алгоритма закончится. Однако догадаться об этом не так уж просто. Предлагаемый нами компьютерный эксперимент подскажет учащимся причину, по которой последовательность стабилизируется. Само же доказательство не так уж сложно для математически грамотного школьника.

В § 49 учащимся демонстрируется другой метод доказательства того, что алгоритм завершает работу за конечное число шагов. Как и для § 48, мы рекомендуем изучать этот материал коллективными усилиями учащихся и преподавателя.

Задания 1—3 позволяют проверить, как учащиеся усвоили теоретический материал. Задание 4а легкое, в нем лимитирующей функцией выступает сама переменная n. Действительно, после каждого исполнения тела цикла значение n уменьшается. Поскольку условием завершения цикла является $n > 1$, на каком-то шаге это условие будет нарушено и исполнение алгоритма закончится.

В задании 4б школьники, как правило, интуитивно ощущают, что для любого n алгоритм за конечное число шагов прекратит работу, поскольку деление пополам действует «сильнее», чем прибавление 5. Однако строгое обоснование конечности алгоритма здесь не так уж просто. Вот один из наиболее простых, по-видимому, вариантов.

Рассмотрим алгоритм:

```
Алгоритм
  цел: n, I;
   { Запросить n;
     I := 1;
     Делать пока (n > 1)
          { Если (n нечетно) то { n := (n + 5)/2;
            I := I + 2; }
            иначе { n := n/2; I := I + 1; }
          }          (*конец цикла*)
     Сообщить I;
   }
```

Легко понять, что этот алгоритм эквивалентен алгоритму из задания 5. (**Эквивалентными** называются алгоритмы, для которых совпадает множество допустимых начальных данных и которые при одних и тех же допустимых начальных данных дают одинаковые результаты. Если с учащимися подробно рассматривать вопрос эквивалентности алгоритмов, то проще это делать с такой точки зрения: у нас имеется черный ящик, который понимает оба алгоритма, алгоритмы эквивалентны, если по тем данным, которые подаются на вход, и результатам на выходах нельзя определить, по какому именно алгоритму работал в данном случае черный ящик.) Это, в частности, означает, что исходный алгоритм конечен тогда и только тогда, когда конечен и эквивалентный ему алгоритм. Но в эквивалентный алгоритм встроена лимитирующая функция INT $((n + 5)/2)$: она принимает только целые положительные значения, при $n > 5$ убывает при каждом исполнении тела цикла, поскольку $(n + 5)/2 < n$, при $n = 2$ и $n = 3$ увеличивается на 1, при $n = 4$ и $n = 5$ не меняет своего значения. Это означает, что зацикливаться данный алгоритм может только при $n = 2$, 3, 4 или 5. Однако непосредственное исполнение алгоритма для этих значений n показывает, что работа алгоритма заканчивается за конечное число шагов.

В задании 5 учащиеся, конечно, должны сначала понять, что делает этот алгоритм по ходу своей работы. В первом внутреннем цикле для каждой строки подсчитывается количество единиц и нулей, стоящих в этой строке. Затем, если число единиц больше числа нулей, нули превращаются в единицы, а единицы — в нули [1]. И так обрабатываются все строки.

Во втором цикле аналогичная процедура осуществляется над столбцами. Сигнальная переменная k определяет окончание цикла, если в каждой строке и каждом столбце количество нулей больше количества единиц. После манипуляций со столбцами в строках условие окончания цикла может снова нарушиться, а после очередной манипуляции со строками может нарушиться условие окончания цикла для столбцов. Тем самым представляется вполне вероятным, что для какой-то начальной конфигурации нулей и единиц в массиве алгоритм может работать бесконечно. Но здесь опять выручает лимитирующая функция — количество нулей во всей таблице после каждого исполнения тела цикла **Делать пока** ($k = 1$): если исполнялся хотя бы один внутренний цикл (и тогда значение $k = 1$), то количество нулей возрастает. Но оно не может быть больше общего числа элементов в таблице.

В задании 6 описан алгоритм нахождения простых чисел, известный как «решето Эратосфена». Он был бы конечен в том и только в том случае, если бы простых чисел было конечное число. Однако еще Евклид доказал, что простых чисел бесконечно много. Мы не будем приводить это довольно известное доказательство, и его не надо требовать от учащихся — в обосновании бесконечности алгоритма они могут просто сослаться на это утверждение, которое обычно доказывается (или по крайней мере формулируется) в курсе математики.

Алгоритм, представленный в задании 7 к § 49, очень похож на алгоритм из задания 7 к § 48. Учащиеся могут подумать, что у авторов учебника склероз и они еще раз предложили фактически тот же самый алгоритм (изменился только результат выдачи — вместо $K + M$ сообщается $K * M$)[2]. Но это совсем другой алгоритм, поскольку внутри каждого ветвления меняются значения обеих переменных,

---

[1] Алгоритму дано название «Мимикрия», поскольку в нем меньшинство «перекрашивается» в значение большинства.

[2] На самом деле более вероятно, что учащиеся и не вспомнят об алгоритме из задания 7 к § 48. В этом случае учитель, на наш взгляд, должен обратить внимание учащихся на схожесть этих алгоритмов и предложить им обсудить, в чем проявляются различия между этими алгоритмами.

а не одной (как в том алгоритме). Чтобы нагляднее продемонстрировать отличие этих алгоритмов, полезно предложить учащимся исполнить этот алгоритм, например, для $K = 6$ и $M = 2$. Алгоритм из задания 7 к § 48 на этой паре значений зацикливается, а данный алгоритм завершает работу с выдачей числа 15 в качестве результата.

По-прежнему, если K и M оба нечетны, тело цикла не исполняется ни разу и алгоритм завершает свою заботу выдачей произведения K и M.

Пусть теперь хотя бы одно из чисел четно. Рассмотрим возможные случаи.

1) K нечетно, а M четно. Тогда первое ветвление не исполняется. Во втором ветвлении если $M/2$ четно, то K остается нечетным; если $M/2$ нечетно, то K становится четным, так что в этой паре все равно ровно одно число четно, а другое нечетно.

2) K четно, а M нечетно. Если $K/2$ четно, то M остается нечетным и второе ветвление не исполняется. Если $K/2$ нечетно, то M становится четным и мы находимся в ситуации, рассмотренной в случае 1. Значит, и в этом случае после исполнения тела цикла одно из чисел K и M четно, а другое нечетно.

Подведем промежуточный итог: если числа K и M разной четности, то после исполнения тела цикла они снова будут разной четности, и, следовательно, будет выполнено условие продолжения цикла. Значит, для таких пар K и M алгоритм зацикливается.

3) K четно и M четно. Запишем $K = 2^s \cdot a$, $M = 2^t \cdot b$, где a и b — нечетные натуральные числа, а s и t оба натуральные.

Исследуем сначала случай, когда $s = t$. Тогда после исполнения первого ветвления $K = 2^{s-1} \cdot a$, $M = 2^{s-1} \cdot (2b + a)$. В скобках у M записано нечетное число, т. е. у новых значений K и M показатели степеней множителя 2 останутся одинаковыми, но уменьшатся на 1. Второе ветвление (если оно будет исполняться) также оставит одинаковым показатель степеней множителя 2, уменьшив его еще на 1. Это рассуждение показывает, что в данном случае в качестве лимитирующей функции можно взять показатель множителя 2: после каждого исполнения тела цикла он уменьшается (на 2 или 1), и так как отрицательным он быть не может, то через несколько таких исполнений он станет равным 0. Это означает, что K и M превратятся в нечетные числа и исполнение алгоритма закончится.

Теперь исследуем случай, когда $s < t$. Тогда после исполнения первого ветвления $K = 2^{s-1} \cdot a$, $M = 2^{s-1} \cdot (2^{t-s+1}b + a)$. Поскольку число $2^{t-s+1}b + a$ нечетно, у новых значений K и M показатели степеней множителя 2 стали равными.

Следовательно, они и дальше будут оставаться такими. Значит, и в этом случае исполнение алгоритма заканчивается через конечное количество шагов.

Исследуем случай, когда $s > t + 1$. После исполнения первого ветвления $K = 2^{s-1} \cdot a$, $M = 2^t \cdot (b + 2^{s-1-t} \cdot a)$. Число $b + 2^{s-1-t} \cdot a$ нечетно, так что перед исполнением второго цикла $t < s - 1$. Но тогда во втором цикле показатели степеней множителя 2 у чисел K и M станут равными. Это означает, что и в этом случае исполнение алгоритма заканчивается через конечное количество шагов.

Наконец, исследуем случай, когда $s = t + 1$. После исполнения первого ветвления $K = 2^t \cdot a$, $M = 2^t \cdot (b + a)$. Число $b + a$ четно, так что показатель степени множителя 2 у числа M будет больше, чем t. Если $b + a$ делится на 4, то перед исполнением второго ветвления мы находимся в уже рассмотренной ситуации и знаем, что приводит к выравниванию показателей степеней множителя 2, после чего исполнение алгоритма закончится за конечное число шагов. Если же $b + a$ не делится на 4, то $b + a = 2b_1$, где b_1 нечетно. Значит, $M = 2^{t+1} \cdot b_1$. После исполнения второго ветвления значение M станет равным $2^t \cdot b_1$, а значение K будет $2^t \cdot (b_1 + a)$, причем $b_1 + a$ четно. Если $b_1 + a$ делится на 4, то, как уже было рассмотрено выше, произойдет выравнивание степеней множителя 2 и алгоритм завершит работу за конечное число шагов. Пусть снова $b_1 + a = 2a_1$, где a_1 — нечетное число. Таким образом, к началу очередного исполнения цикла ситуация такова: $K = 2^{t+1} \cdot a_1$, $M = 2^t \cdot b_1$, где a_1 и b_1 — нечетные числа. Вспоминая, что $s = t + 1$, мы видим, что у K и M изменились только нечетные множители. Проследим изменение этих множителей: $b_1 = \dfrac{a+b}{2}$, $a_1 = \dfrac{a+b_1}{2} = \dfrac{a + \frac{a+b}{2}}{2} = \dfrac{3a+b}{4}$.
Найдем разность чисел a_1 и b_1:

$$a_1 - b_1 = \frac{3a+b}{4} - \frac{a+b}{2} = \frac{a-b}{4}.$$

Это означает, что при $a \neq b$ числа a_1 и b_1 тоже не равны, причем разность между ними вчетверо меньше, чем разность между предшествующими нечетными множителями. Если мы исполним тело цикла еще раз, то появятся числа a_2 и b_2, которые опять будут не равны, а их разность уменьшится вчетверо по сравнению с предыдущей. Значит, по отношению к первоначальной разности $a - b$ она уменьшится уже в 16 раз, но будет оставаться при этом целым числом. Если алгоритм зацикливается, то процесс возникновения чисел a_n и b_n продолжается бесконечно. Но это невозможно, поскольку $a_n - b_n = \dfrac{a-b}{4^n}$, и, следовательно, становится меньше любого натурального числа, оставаясь в то же время натуральным числом. Полученное противоречие показывает, что зациклиться алгоритм может только в том случае, если $a = b$. Но это означает, что при четных K и M алгоритм может зациклиться лишь в том случае, когда K ровно в 2 раза больше M. И действительно, если $K = 2M$, то нетрудно проверить непосредственным исполнением алгоритма, что после каждого исполнения цикла числа K и M оставались теми же, что были до его исполнения, и потому алгоритм будет исполняться бесконечное количество раз.

Теперь мы можем сформулировать ответ: алгоритм применим к тем и только к тем парам (K; M), для которых K и M одной четности и в случае четных K и M их отношение не равно 2.

Продолжая сравнение алгоритма, предложенного в задании 7 из § 48, и только что рассмотренного алгоритма, можно предложить учащимся рассмотреть следующий алгоритм:

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

Он отличается от исходного алгоритма только действием **Сообщить**. Ясно, что область применимости у измененного алгоритма осталась та же. Можно предложить учащимся исполнить этот алгоритм для нескольких различных допустимых исходных данных. После этого учащиеся обычно с удивлением обнаруживают, что результат исполнения алгоритма (если только он заканчивает работу) — это всегда сумма исходных чисел. Но при внимательном рассмотрении становится ясно, что величина $K + M$ для чисел K и M, полученных после исполнения первого ветвления, останется той же, что была до его исполнения. Точно так же и со вторым ветвлением. Следовательно, величина

K и M не меняется при исполнении алгоритма. Напомним, что такую величину называют **инвариантом** (мы упоминали о нем при обсуждении задания 4 к § 47). Подробному обсуждению понятия «инвариант» будет посвящен следующий параграф, но полезность этого понятия можно продемонстрировать уже сейчас. Ведь числа K и M будут разной четности тогда и только тогда, когда их сумма нечетна. Поскольку сумма чисел K и M остается одной и той же, то в результате исполнения цикла мы снова получаем числа разной четности и потому исполнение алгоритма никогда не закончится. Тем самым применение инварианта дает нам другое доказательство зацикливания алгоритма для тех пар, у которых разная четность (вместо рассмотрения случаев 1 и 2 в предыдущем доказательстве).

Приступим к обсуждению задания 8. Фактически в нем представлен недетерминированный алгоритм, поскольку суть его заключается в том, что в массиве случайным образом выбираются два элемента, один из них заменяется полусуммой, а другой — абсолютной величиной полуразности. Надо напомнить учащимся, что разобранный в объяснительном тексте алгоритм библиотекаря тоже является недетерминированным (и очень похожим на данный — там в одномерном массиве меняются местами два произвольно выбранных числа (конечно, с точки зрения информатики можно считать, что переставляются не сами книжки, а номера томов), а здесь также два произвольных числа заменяются на результаты действий над ними). Тем не менее использование лимитирующей функции позволило доказать его конечность. Но...

Пусть в массиве есть два элемента, в которых числа не равны. Один из них имеет номер I, другой — номер $J \neq I$. Пусть I-й элемент равен числу a, J-й — числу b. Конечно, вероятность, что при исполнении алгоритма будут выбраны именно эти элементы, равна $2 \cdot \frac{1}{30} \cdot \frac{1}{29} = \frac{1}{435}$. Тем не менее это вовсе не исключено. Поскольку y окажется отличным от нуля, состоится второе исполнение тела цикла, при котором с той же вероятностью $\frac{1}{435}$ снова могут быть выбраны те же элементы. Нетрудно проверить, что тогда $x = \frac{1}{2}\max(a, b)$, а $y = \frac{1}{2}\min(a, b)$. Значит, $x \neq y$ и $y \neq 0$. Следовательно, цикл опять будет исполняться. И в третий раз может так оказаться, что будут выбраны те же элементы массива. Снова y окажется неравным нулю. И в четвертый раз совершенно случайно могут быть выбраны те же элементы. При этом исполнении тела цикла получится, что $x = \frac{1}{4}\max(a, b)$, а $y = \frac{1}{4}\min(a, b)$. Ясно, что такой процесс может продолжаться бесконечно. Из этого рассмотрения вытекает, что единственный вариант, когда алгоритм гарантированно заканчивает работу за конечное число шагов, — это случай равенства всех чисел в массиве (независимо от того, натуральные они или любые другие).

Это задание позволяет продемонстрировать учащимся важное отличие детерминированных (т. е. привычных им) алгоритмов от недетерминированных. В обычном алгоритме завершение или незавершение работы алгоритма за конечное число шагов полностью определено тем, какие начальные данные предъявлены ему для обработки. В данной задаче при одном и том же наборе начальных данных алгоритм может закончить работу, а может зациклиться. Почему может произойти зацикливание, мы уже обсудили. Но ведь с вероятностью $\frac{1}{900}$ при выполнении тела цикла может быть выбран один и тот же элемент ($I = J$). Тогда алгоритм завершит свою работу за одно исполнение цикла. Все это открывает совершенно новую область исследования алгоритмов — вероятностные свойства алгоритмов. Можно, к примеру, вычислять вероятность того события, что предъявленный алгоритм закончит работу за конечное число шагов. Но это вопросы, далеко выходящие за рамки даже профильного курса школьной информатики.

Заключительный параграф главы 5 посвящен еще одному математическому инструменту исследования алгоритмов — понятию инварианта. В объяснительном тексте дан образец оформления проверки того, что некоторое свойство является инвариантом. Мы советуем в целом придерживаться этого образца. Но в целях экономии места мы будем просто указывать инвариант и проводить с ним дальнейшие рассмотрения в вольной форме.

В задании 2 к § 50 результатом работы алгоритма является квадрат запрашиваемого числа k. Для доказательства этого достаточно проверить, что после каждого исполнения тела цикла переменной m присвоено значение, равное $(n + 1)^2$, где n — текущее значение счетчика цикла.

В задании 3 учащимся сначала можно предложить исполнить этот алгоритм для нескольких пар чисел m и n. Вот возможный набор таких пар: $m = 4$, $n = 7$; $m = 4$, $n = 6$; $m = 3$, $n = 3$. Среди разных гипотез, которые высказывают школьники (особенно относительно выводимого значения переменной y), обычно присутствует правильная. Она такова: сообщаемое значение x — это наибольший общий делитель чисел m и n (далее мы будем обозначать его НОД (m; n)), а для y — это наименьшее общее кратное чисел m и n (далее мы будем обозначать его НОК (m; n)).

Чтобы обосновать это, поищем возможные инварианты. Один из инвариантов обнаружить нетрудно. Поскольку для любых чисел a и b у пар чисел (a, b), $(a, b - a)$ и $(a - b, b)$ наибольший общий делитель один и тот же, то, сколько бы раз цикл ни исполнялся, величина НОД (x, y) остается неизменной. Тем самым функция $f(x, y) =$ НОД (x, y) является инвариантом цикла. Другой инвариант, зависящий от всех четырех переменных, фигурирующих в цикле, найти несколько сложнее. Им является, например, функция $g(x, y, u, v) = xv + yu$. Действительно, если $x < y$, то, после того как в очередной раз исполнение тела цикла закончилось, новое значение функции g подсчитывается по формуле

$$g(x, y - x, u, v + u) = x(v + u) + (y - x)u = \\ = xv + yu = g(x, y, u, v),$$

если же $x > y$, то значение функции g после очередного исполнения тела цикла подсчитывается по формуле

$$g(x - y, y, u + v, v) = (x - y)v + y(u + v) = \\ = xv + yu = g(x, y, u, v).$$

Значит, функция $g(x, y, u, v)$ тоже инвариант.

Можно построить и еще сколько угодно инвариантов. Например, сумма функций f и g тоже, очевидно, является инвариантом. Нужно ли так много инвариантов?

Действительно, инвариантов всегда можно построить бесконечно много. А вот какой из них окажется полезным, это непростой вопрос.

Используют инвариант обычно так: вычисляют его значение до первого исполнения тела цикла, а затем после последнего его исполнения. Например, до исполнения цикла инвариант $f(x, y)$ имеет значение НОД $(m; n)$, поскольку в этот момент $x = m$, а $y = n$. А когда закончилось исполнение цикла (всего цикла, а не однократное исполнение тела цикла!), из условия продолжения цикла следует, что $x = y$. Поэтому значение $f(x, y) = f(x, x) =$ НОД $(x; x) = x$. Но по окончании цикла x служит одним из результатов работы алгоритма. Следовательно, один из результатов работы алгоритма — это наибольший общий делитель чисел m и n. Причем мы доказали, что он именно такой!

Попробуем теперь с помощью инвариантов определить, что является вторым результатом работы алгоритма. Для этого построим еще один инвариант:

$$h(x, y, u, v) = \frac{g(x, y, u, v)}{2f(x, y)}.$$

До начала исполнения цикла имеем

$$h(x, y, u, v) = h(m, n, m, n) = \frac{g(m, n, m, n)}{2f(m, n)} = \frac{mn}{\text{НОД}(m, n)}.$$

А теперь найдем значение инварианта h после завершения исполнения цикла:

$$h\ (x,\ y,\ u,\ v) = h\ (x,\ x,\ u,\ v) = \frac{g\,(x,\,x,\,u,\,v)}{2f\,(x,\,x)} = \frac{x\,(u+v)}{2x} = \frac{u+v}{2}.$$

Тем самым инвариант $h\ (x,\ y,\ u,\ v)$ по окончании исполнения цикла дает нам второй результат работы алгоритма. Но через аргументы этот инвариант выражается как $\frac{mn}{\text{НОД}\,(m,\,n)}$. В общем-то это уже ответ на пункт *б* данного задания. Однако хорошо известно (по крайней мере в классах углубленной математической подготовки), что

$$\frac{mn}{\text{НОД}\,(m,\,n)} = \text{НОК}\ (m;\ n).$$

Такое использование инварианта — доказательное выявление того, что является результатом работы алгоритма, — может показаться счастливым случаем. Однако теоретики современного программирования (Э. Дийкстра в книге «Дисциплина программирования», Д. Грис в книге «Наука программирования» и др.) высказывают аргументированную точку зрения, что всегда имеется инвариант, описывающий результат работы цикла. Более того, по их мнению, стратегия конструирования цикла состоит в изначальном определении инварианта, описывающего результат, и лимитирующей функции, которая гарантирует конечность исполнения цикла. Если эти два объекта определены, то дальше следует довольно формальная процедура, позволяющая построить цикл, имеющий заданный инвариант и заданную лимитирующую функцию. Если инвариант и лимитирующая функция построены для исходной задачи правильно, то в алгоритме гарантированно не будет ошибок. О чем еще мечтать программисту?! Однако построение как инварианта, так и лимитирующей функции — дело весьма непростое. Если же вернуться к рассматриваемому нами алгоритму из задания 3 к § 50, то здесь в роли лимитирующей функции может выступить $\min\ (x,\ y)$. Она принимает только целые неотрицательные значения, причем при каждом исполнении тела цикла ее значение уменьшается. Бесконечно этот процесс продолжаться не может, значит, на некотором шаге алгоритм прекратит работу.

В задании 4 в переменной c подсчитывается количество общих элементов в массивах K и M. Инвариантом цикла в момент исполнения при некоторых значениях параметров a и b является сумма значения параметра c и количество общих элементов в массивах $K\ [a : 20]$ и $M\ [b : 30]$.

ГЛАВА 6

Графы и алгоритмы на графах

На первый взгляд материал этой главы противоположен тому, что предлагалось к изучению в главе 3. И действительно, глава 3 относится к разделам ярко выраженного пользовательского направления в школьной информатике, в то время как тематика графов считается вполне заслуженно некоей абстрактной и весьма математизированной областью науки и практики. Однако в методике изучения этих тем есть много общего. Во-первых, напомним, что графы выступают как одно из универсальных средств описания системных моделей. Фактически любая схема, любое изображение структуры — это всегда граф, даже если мы не хотим пользоваться этим термином для описания того объекта, с которым имеем дело. Таким образом, граф — это некий язык и некая технология для описания разнообразных объектов и явлений. И в этом функция графов сродни функции информационных технологий. Во-вторых, так же как для информационных технологий, широкое применение графов на современном уровне невозможно без использования компьютера. Поэтому в данной главе, как и в главе 3, компьютерный практикум занимает значительное место.

Тема 13. Свойства графов, представление графов и алгоритмы

С понятием графа учащиеся в соответствии с Федеральным стандартом общего образования должны быть знакомы по курсу информатики еще с 9 класса. Кроме того, о графах речь шла и в § 6 нашего учебника для 10 класса в связи с повторением понятия «системная модель», поэтому на материал § 51 можно смотреть как на повторение, но при этом надо понимать, что многие сформулированные там вещи могут оказаться для учащихся абсолютно новыми. Это, в частности, относится к таким утверждениям, как лемма о рукопожатиях и теорема о существовании вершин одинаковой кратности. В задании 6 к этому параграфу данные утверждения выступают как необходимые условия существования графа.

Действительно, можно сразу сказать, что граф с набором степеней вершин, указанным в задании 6а, не существует, поскольку в этом наборе нет двух вершин одинаковой степени. Аналогично не бывает графа с набором степеней, указанным в задании 6б, так как в этом наборе имеется нечетное число вершин нечетной степени, что противоречит лемме о рукопожатиях. Для наборов степеней вершин, предложенных в заданиях 6в и 6г, примеры соответствующих графов легко строятся. Это иногда порождает у учащихся неверное мнение, что набор чисел, удовлетворяющий двум данным утверждениям, является набором степеней вершин подходящего графа. Надо добиться понимания от учащихся, что эти условия не являются достаточными для того, чтобы граф с заданным числом вершин и заданными степенями этих вершин существовал. Для этого можно предложить им выполнить следующее задание.

Задание 29. Существует ли граф с 7 вершинами и следующим распределением степеней вершин: 1, 1, 2, 3, 4, 5, 6?

Оба условия: четность количества нечетных вершин и существование двух вершин одинаковой степени — выполнены. Однако граф с таким набором степеней, очевидно, не существует. Действительно, наличие вершины степени 6 показывает, что такая вершина должна быть смежной со всеми остальными вершинами, в частности с обеими вершинами степени 1. В то же время вершина степени 5 должна быть смежной со всеми вершинами, кроме одной. Поэтому она должна быть смежна по крайней мере с одной из вершин степени 1. Но тогда эта вершина степени 1 оказалась смежной уже с двумя вершинами — противоречие.

Обсуждая другие задания к § 51, отметим, что при выполнении заданий 4а и 4г учащиеся нередко забывают около каждой вершины нарисовать петлю, показывающую, что вершина находится в данном отношении сама с собой. Впрочем, такие петли должны появляться у некоторых вершин во всех пунктах задания 4, кроме пункта *б*.

Задания 7 и 8 решаются применением леммы о рукопожатиях.

Важную методологическую роль играет задание 15. Обсуждая его, учащиеся должны дать ответ на очень непростой вопрос: как доказать, что один граф нельзя перерисовать в другой граф? Ясно, что если один граф можно перерисовать в другой, то у этих графов одинаковое число вершин, одинаковый набор степеней вершин, одинаковое, например, количество циклов длины 3 и т. д. Иными словами, у одинаковых, но по-разному нарисованных графов есть много инвариантов — свойств, которые не зависят от того, как граф изображен. Поэтому если хотя бы для одного инварианта его значение для двух графов различно, то невозможно один граф перерисовать в другой. Как мы видим, здесь учащиеся снова встречаются с применением понятия «инвариант», которое изучалось ими в § 50.

Задание 16 может оказаться сложным для учащихся. Для доказательства надо выбрать ученого, у которого наибольшее количество друзей. Пусть этих друзей n. Тогда, во-первых, у каждого из друзей этого ученого разное количество друзей, так как все они имеют общего друга, выбранного нами ученого. Во-вторых, у каждого из них не больше чем n друзей, поскольку нами выбран ученый с максимальным количеством друзей. Следовательно, для этих людей должны реализоваться все варианты количества друзей — будет ровно 1 человек, у которого 1 друг, ровно 1 человек, у которого 2 друга, ..., ровно 1 человек, у которого n друзей. Значит, есть ученый, у которого ровно 1 друг.

Выполнение остальных заданий к этому параграфу большого труда не составляет.

Материал, представленный в § 52, носит чисто технический характер. Обработка графа с помощью компьютера возможна лишь в том случае, если перекодировать его в числовом виде. Некоторые способы такого кодирования графа как раз и представлены в этом параграфе. Сопровождающая этот параграф лабораторная работа № 21 направлена на подготовку учащимися программ, позволяющих строить графы и производить преобразование их из одного способа задания в другой. Фактически это комплекс некоторых вспомогательных программ, которые будут использоваться в последующих лабораторных работах. Кроме того, активным участникам различных олимпиад по информатике и программированию такие программы позволяют создавать тесты для самопроверки решения тех задач, в которых используются графы. Отметим, что программы, которые обсуждаются в § 52 и 53, — это обязательный минимум любого олимпиадника-программиста. Дело в том, что алгоритмы обхода графа, которым посвящен § 53, — это, вообще говоря, алгоритмы произвольного грамотного перебора, а переборные задачи не такая уж редкость в олимпиадных соревнованиях, да и в жизни в целом.

В объяснительном тексте § 53 приведены алгоритмы обхода графа. С точки зрения реализации алгоритма это далеко не самое оптимальное решение. В действительности важную и принципиальную роль в организации переборного процесса играет понятие стека. Это понятие представлено в описании лабораторной работы № 22, и учащимся предложено провести компьютерный эксперимент, подтверждающий, что использование такой структуры данных оптимизирует процесс поиска при реализации поиска в глубину. В лабораторной работе № 23 аналогичные рассмотрения проводятся для алгоритма поиска в ширину.

Чрезвычайно важным классом графов являются деревья. Этот тип графов удостоился неоднократного упоминания в Федеральном стандарте образования по информатике. Поэтому мы сочли необходимым этот вид графов рассмотреть значительно подробнее, чем другие виды, например двудольные графы, планарные и т. п. Важными понятиями при изучении свойств деревьев являются понятия моста и точки сочленения. О них рассказывается в весьма небольшом по объему § 54.

Сами деревья и их характеристические свойства рассматриваются в § 55. Они возникают как остовы произвольных графов. В этом параграфе еще не указаны причины, чем вызван интерес к остовам графов; об этом более подробно идет речь в § 56. Основную часть объяснительного текста этого параграфа занимает доказательство того, что алгоритм Краскала, во-первых, конечен и, во-вторых, на самом деле дает каркас минимального веса. Это доказательство весьма непростое. Оно, в частности, заставляет учащихся снова вспомнить понятие лимитирующей функции. Нам представляется, что разбирать данное доказательство следует только в классах с углубленной математической подготовкой, в то время как понять сам алгоритм Краскала вполне по силам и тем учащимся, которые не отличаются глубокими математическими познаниями. Этот же комментарий следует сделать относительно задания 8 — помечающая его звездочка в данном случае означает, что предлагать это задание следует только математически сильным учащимся, хотя само доказательство проводится по тому же плану, что и для алгоритма Краскала.

Остальные задания к § 56 особых затруднений, по нашему мнению, вызвать не должны.

ГЛАВА 7

Игры и стратегии

Без преувеличения можно сказать, что эта глава аккумулирует в себе аспекты теоретической информатики. Здесь мы снова говорим о понятии информации, об алгоритмах, о кодировании, о моделях, об управлении. При желании весь курс информатики мог бы быть выстроен вокруг игровой проблематики и каждое изучаемое понятие могло бы иллюстрироваться тем, какое место оно занимает в играх. Причины этого заключаются в том, что любая игра — это модель взаимодействия человека с окружающим миром. И в зависимости от того, на чем сосредоточено внимание, мы оказываемся в той или иной области информатики. Едва начав обсуждение понятия игры, мы вынуждены сразу указать на то, какую роль в этом понятии играют основополагающие понятия информатики: информация, модель, информационное взаимодействие, управление. Уже в первом параграфе этой главы мы обсуждаем деление игр на игры с полной и неполной информацией, и учащимся придется вспомнить о таком свойстве информации, как полнота, которое обсуждалось в § 2. Процесс взаимодействия играющих сторон удобно изображать графом. Мы рассматриваем только конечные игры, для которых представляющие их графы являются деревьями, но, вообще говоря, многие игры представляются орграфами более сложной природы. Такое ограничение в наших рассмотрениях связано, в частности, с тем, что в главе 6 мы практически совсем не касались орграфов (опять-таки в связи с дефицитом учебного времени).

Любая игра — это всегда управляемая деятельность. Вся она или какие-то ее фрагменты нередко оказываются алгоритмизуемыми. А в применяемых для игр алгоритмах можно встретить все виды алгоритмических конструкций, в том числе рекурсию. Здесь мы тоже ограничились наиболее простыми примерами.

Даже этот, бегло составленный перечень наглядно показывает справедливость высказанного нами тезиса — обсуждая игры, можно изучить всю теоретическую информатику. Но мы пошли другим путем, поставив эту тему в заключительную позицию. Тем самым, рассматривая игры, мы кратко повторим весь пройденный курс.

Тема 14. Игра как модель управления

«Что наша жизнь? Игра!» В этой поэтической фразе для учителя информатики заключено смысла гораздо больше, чем для любого другого человека. С одной стороны, она в емкой форме повторяет то, что мы сказали выше, — игра может служить моделью любой сферы человеческого существования. С другой стороны, игра — это не только модель, но зачастую и сама человеческая жизнь. Когда человек сидит за карточным столом в казино, или выходит на футбольное поле в составе команды, или садится за игровую приставку, игра для него — это сама жизнь, с ее страстями, выбросами адреналина, радостью побед и горечью поражений. Это означает, что понятие игры имеет так же много оттенков, как и понятие информации. Здесь тоже существует бытовое понимание этого слова, которое встречается наиболее часто, и потому именно в таком виде оно прежде всего присутствует в сознании школьника. В разных науках есть своя трактовка этого понятия уже как научного термина, например в психологии или этологии (науке о поведении животных). В информатике (и кибернетике) игра понимается как формализованная модель взаимодействия нескольких сторон, из которых по крайней мере одна преследует определенные цели.

При разъяснении понятия игры нам кажется полезным проведение указанной выше аналогии между этим понятием и понятием информации. Учащиеся должны точно понимать, какой смысл вкладывается в термин «игра» при рассмотрении его в курсе информатики. И так же как в случае понятия «информация», при любом его понимании существует некоторое «ядро» этого понятия (а именно сведения, позволяющие осуществлять целенаправленную деятельность), для понятия «игра» таким ядром является понятие стратегии. Даже играя в рулетку, где по самому определению этой игры выигрыш или проигрыш — дело случая, игрок, как правило, придумывает для себя ту или иную линию поведения.

Объявив, что игра — это формализованная модель, мы обрекли себя на выбор формализованного языка описания этой модели. Поскольку процесс игры заключается в последовательном переходе от одной позиции к другой, то естественным языком описания таких переходов становятся ориентированные графы. Мы сразу ограничили класс рассматриваемых игр в учебнике, поэтому все фигурирующие графы являются деревьями[1]. Причин для этого несколько. Во-первых, это один из наиболее простых классов игр, и вполне естественно начинать знакомство с теорией игр именно с него[2]. Во-вторых, именно эти игры фигурируют в заданиях Единого государственного экзамена, а несмотря на всю незамысловатость таких игр, эффект неожиданности весьма нежелателен при прохождении каких бы то ни было аттестационных испытаний. В-третьих, их изучение в этом месте курса не требует введения дополнительного математического аппарата.

В § 57 вводится в рассмотрение указанный класс игр и даются соответствующие терминологические пояснения. Центральным понятием этого параграфа является понятие выигрышной стратегии. Учащиеся должны знать это понятие и уметь объяснять его содержание.

При выполнении задания 1 к § 57 учащиеся должны вспомнить определение модели и показать, почему оно здесь применимо. Действительно, реальный объект оказывается замененным на другой объект, в котором борьба противостоящих сторон описана так, что она учитывает не все, а только существенные факторы ведения борьбы.

Перечисление того, чем характеризуется любая игра (задание 2), приведено в самом начале § 57: определенностью исходных позиций и игрового материала, результата игры (когда игра считается оконченной и что именно считать выигрышем, поражением или ничьей), а также определенностью допустимых действий каждого из игроков и порядка выполнения ими ходов. Обычно игроки равноправны, т. е. у них одинаковый игровой материал, одинаковый набор допустимых действий, и ходы они совершают строго по оче-

---

[1] Методически целесообразно предложить учащимся рассмотреть примеры игр с полной информацией, для которых орграф игры не является деревом. К таковым в первую очередь относятся игры, допускающие многократное повторение позиций (шахматы, шашки, го и т. д.). Тогда в орграфе игры появляется цикл. Конечность таких игр обеспечивается специальными оговорками в правилах (типа трехкратное повторение позиции, вечный шах и т. п.).

[2] Это не единственная точка зрения на то, с чего надо начинать знакомство учащихся с теорией игр. Многие авторы придерживаются мнения, что можно сразу изучать так называемые матричные игры.

реди. Однако бывают игры, где это не так, например игра «Волк и четыре собаки» [1].

Ответ на вопрос задания 4 положительный. Именно для того, чтобы каждая из этих игр обязательно была конечной, соответствующим образом введено понятие «ничья». Надо прямо сказать, что почти все игры, в которые реально играют люди, являются конечными, хотя иногда их конечность обеспечивается внеигровыми факторами (например, продолжительность игры по времени). Нередко конечность игры определяется счетом определенных достижений. Один из примеров гипотетически бесконечной игры — волейбол, поскольку окончанием игры служит конъюнкция двух условий: достижение определенного счета и разницы в счете не менее двух мячей. Мы считаем нужным проведение подобных обсуждений с учащимися, чтобы они лучше усвоили понятие конечности игры.

Вопросы в заданиях 5—7 заставляют учащихся воспроизвести соответствующие определения. Ответом в задании 8 может быть следующая формулировка: непроигрышной стратегией для данного игрока называется правило совершения ходов этим игроком, при соблюдении которых его противник не может выиграть, какие бы ходы он ни совершал. Понятие непроигрышной стратегии нужно в том и только в том случае, если в качестве результата игры предусмотрена ничья.

На рисунках 7.1 и 7.2 изображены деревья, которые требовалось построить в заданиях 9а и 9б.

---

[1] Напомним, в чем состоит эта игра. На шахматном поле вдоль одной его стороны на полях черного цвета стоят 4 белые шашки (собаки), а на противоположной стороне на одном из полей стоит черная шашка (волк). Ходы выполняются по очереди, пропускать ход запрещено. За один ход игрок передвигает на соседнее по диагонали и свободное от других шашек поле одну из своих шашек, причем собакам разрешено двигаться только вперед, а волк может и отступать. Если собакам удается окружить волка так, что тому некуда сделать ход, выиграл игрок, игравший собаками. В противном случае выигрывает волк. Кто делает ход первым — волк или собаки, можно решать подбрасыванием монетки. Эта игра конечна (поскольку каждая из собак может сделать не более семи ходов), с полной информацией. В частности, для этой игры может быть построено дерево вариантов: каждая позиция описывается положением пяти шашек на поле с указанием допустимых переходов из одной позиции в другую. Построение такого дерева вручную — дело трудоемкое (и потому игра интересна), поэтому можно предложить учащимся написать соответствующую программу, а затем по построенному дереву определить, для какого игрока существует выигрышная стратегия (которая, между прочим, допускает формулировку в виде правила выбора ходов). Можно обобщить эту игру на доски разных размеров.

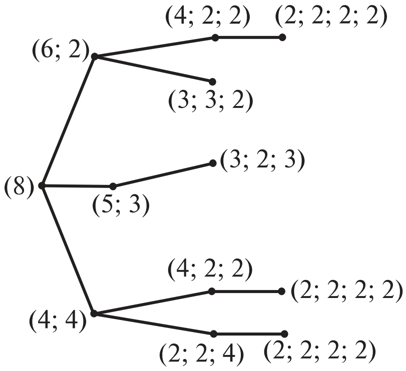

(8)
(6; 2)
(4; 2; 2)
(2; 2; 2; 2)
(3; 3; 2)
(5; 3)
(3; 2; 3)
(4; 4)
(4; 2; 2)
(2; 2; 2; 2)
(2; 2; 4)
(2; 2; 2; 2)

Рис. 7.1

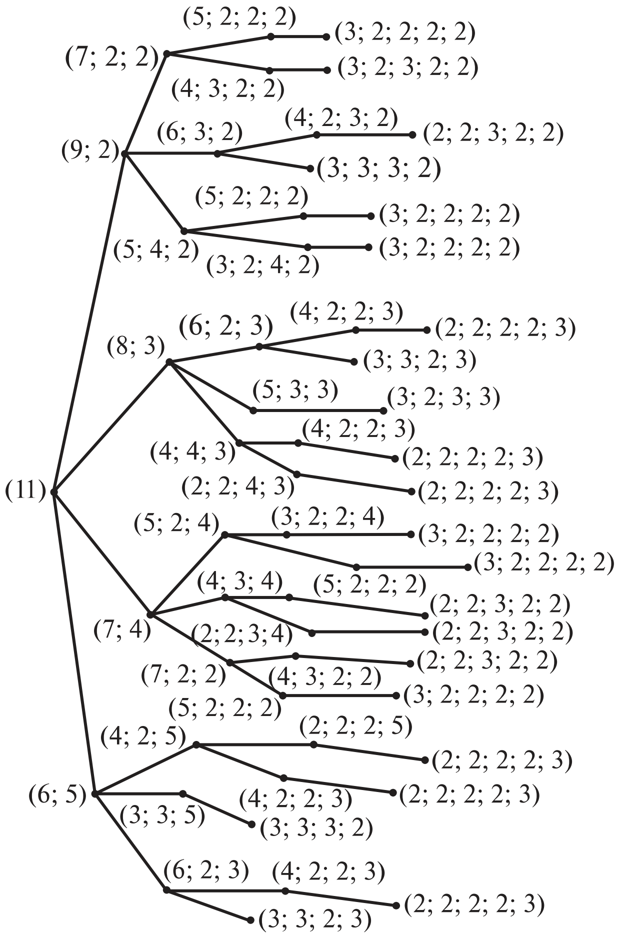

(11)
(9; 2)
(7; 2; 2)
(5; 2; 2; 2)
(3; 2; 2; 2; 2)
(4; 3; 2; 2)
(3; 2; 3; 2; 2)
(6; 3; 2)
(4; 2; 3; 2)
(2; 2; 3; 2; 2)
(3; 3; 3; 2)
(5; 4; 2)
(5; 2; 2; 2)
(3; 2; 2; 2; 2)
(3; 2; 4; 2)
(3; 2; 2; 2; 2)
(8; 3)
(6; 2; 3)
(4; 2; 2; 3)
(2; 2; 2; 2; 3)
(3; 3; 2; 3)
(5; 3; 3)
(3; 2; 3; 3)
(4; 4; 3)
(4; 2; 2; 3)
(2; 2; 2; 2; 3)
(2; 2; 4; 3)
(2; 2; 2; 2; 3)
(7; 4)
(5; 2; 4)
(3; 2; 2; 4)
(3; 2; 2; 2; 2)
(3; 2; 2; 2; 2)
(4; 3; 4)
(5; 2; 2; 2)
(2; 2; 3; 2; 2)
(2; 2; 3; 2; 2)
(2;2;3;4)
(2; 2; 3; 2; 2)
(7; 2; 2)
(4; 3; 2; 2)
(3; 2; 2; 2; 2)
(5; 2; 2; 2)
(6; 5)
(4; 2; 5)
(2; 2; 2; 5)
(2; 2; 2; 2; 3)
(4; 2; 2; 3)
(2; 2; 2; 2; 3)
(3; 3; 5)
(3; 3; 3; 2)
(6; 2; 3)
(4; 2; 2; 3)
(2; 2; 2; 2; 3)
(3; 3; 2; 3)

Рис. 7.2

Для игры, сформулированной в задании 9а, для первого игрока есть выигрышная стратегия: первым ходом он делит кучку на две равные части, переводя противника в позиции (4; 4), после чего на любой ход второго игрока отвечает тем, что делит единственную оставшуюся кучку из 4 камней снова на две равные части.

Для игры, сформулированной в задании 9б, первый игрок при правильной игре второго всегда проигрывает. Действительно, если первый игрок разделит кучку на части $9 + 2$ или $8 + 3$, то фактически игра для второго игрока свелась уже к разобранным случаям, и мы знаем, что у игрока, начинающего в такой позиции, есть выигрышная стратегия. Если же первый игрок разбивает кучку на 7 и 4, то любой ответ второго игрока ставит первого игрока в проигрышную позицию. Если же первым ходом игрок делит кучку на 6 и 5 камней, то второй игрок делит кучку из 6 камней на две части: 4 камня и 2 камня, после чего первый игрок проигрывает.

Мы не воспроизводим деревья, возникающие при решении задач 9в, 10 и 11, поскольку они слишком громоздкие. Мы также оставляем на усмотрение учителя, для каких из этих заданий следует настаивать, чтобы учащиеся нарисовали соответствующий граф. Но крайне важно, чтобы учащиеся для каждой игры четко определили, что в ней является позицией. Для игр, описанных в задании 10, позиция — это множество нарезанных к этому моменту прямоугольников. Для игр из задания 11 позиция — это пара чисел, задающая количество камней в кучках.

Отметим еще, что задание 11 по своей постановке совпадает с заданиями С3, из года в год повторяющимися в ЕГЭ по информатике. Официальное решение в точности предусматривает предъявление дерева игры, правда, как говорят авторы заданий, не полного, а усеченного (полное дерево просто не поместится на бланк), т. е. без рассмотрения всех возможных вариантов ходов. Но, во-первых, учащемуся далеко не сразу может быть ясно, какими ребрами этого дерева можно пренебречь, а во-вторых, требуется обоснование того, что именно этими ребрами и вершинами можно пренебречь. Поэтому выполнению данного задания целесообразно уделить особое внимание. Скажем сразу, что уже в следующем параграфе будет предложен более экономный, на наш взгляд, метод решения подобных задач, имеющий существенно более прозрачное обоснование того, что выбираемая стратегия является выигрышной.

Мы не даем изображения деревьев для игр, о которых говорится в заданиях 9в, 10 и 11, но ответ на вопрос, для какого игрока существует выигрышная стратегия, мы дадим. Проще всего обстоит дело с игрой, описанной в задании 9в. В ней выигрывает первый игрок, ему достаточно расщепить кучку из 13 камней на две кучки, оставив в одной 11 камней, а в другой 2 камня. Кучка из 2 камней дальше не делится, а с кучкой из 11 камней игрок, делающий первый ход, проигрывает.

Для задания 10а ответ таков: выигрывает игрок, делающий первый ход. Здесь имеется несколько выигрышных стратегий, одна из них такова. Первым ходом игрок режет прямоугольник на две части: одну размером 2×7, другую — 3×7. Дальше, чтобы не проиграть, каждый из этих прямоугольников можно резать только поперек стороны, которая содержит 7 клеток. Когда второй игрок делает такой ход на одном из прямоугольников (при этом не отрезается полоска шириной 1), первый игрок своим очередным ходом режет аналогичным образом второй прямоугольник (рис. 7.3). И в дальнейшем первый игрок поступает так же, как второй игрок, только на другом прямоугольнике (если не был отрезан прямоугольник шириной 1).

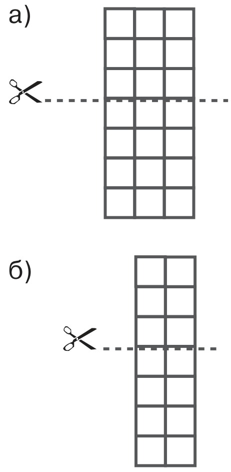


Рис. 7.3

Вообще говоря, это пример так называемой симметричной стратегии, речь о которой пойдет позже, в § 59.

В задании 10б ответ тот же, что и в задании 10а. Здесь игрок сразу разрезает прямоугольник на два одинаковых прямоугольника размером 5×9 и далее следует описанной выше симметричной стратегии.

В задании 10в снова выигрывает первый игрок, однако стратегия здесь достаточно сложная.

В задании 11а указанная позиция является проигрышной для того, кто делает первый ход; в задании 11б, наоборот, указанная позиция выигрышная для начинающего игру. Обоснование этого приводится на с. 202—205.

Вопрос в задании 12 побуждает учащихся вспомнить понятие лимитирующей функции. Если при каждом ходе игрока значение такой функции меняется, то игра обязательно закончится за конечное число шагов. Для игры, описанной в объяснительном тексте § 57, такой функцией является количество кучек. При каждом ходе значение этой функции увеличивается на 1, с другой стороны, никакое ее значение не может оказаться больше чем $\frac{n}{2}$, поскольку в каждой кучке по условию не меньше 2 камней. Это ответ на задание 13.

Для игры, описанной в задании 10, в качестве лимитирующей функции можно взять количество клеток, не стоящих на границе никакого из нарезанных прямоугольников. При каждом ходе количество таких клеток убывает. Однако оно не может стать отрицательным.

Совсем просто назвать лимитирующую функцию для игры, описанной в задании 11. В качестве такой функции можно взять сумму камней в обеих кучках. При каждом ходе ее значение возрастает, однако игра прекращается, как только это значение станет больше 20.

Уже при построении игровых деревьев учащиеся замечают, что многие позиции повторяются. И обычно они высказывают предложение обозначать эти позиции одной и той же вершиной дерева. Более того, для задачи, рассматриваемой в § 57, одинаковыми будут позиции, которые отличаются лишь порядком чисел. Тогда граф, приведенный в учебнике на рисунке 7.3, примет вид, показанный здесь на рисунке 7.4, а граф, изображенный на рисунке 7.2, примет вид, показанный на рисунке 7.5.

Каждый из этих графов перестал быть деревом (и потому на каждом ребре мы теперь указали направление), он стал намного проще для восприятия и анализа.

Мы рекомендуем предложить учащимся проанализировать и другие построенные ими графы с целью их аналогичного преобразования. В этом случае идея изучения позиций (т. е. вершин графа), попадание в которые обеспечивает выигрыш, становится более прозрачной[1]. Совокупность всех незаключительных позиций игры разбивается на два множества: множество выигрышных позиций и множество проигрышных позиций. Обозначим множество выигрышных позиций буквой V, а множество проигрышных позиций буквой P. Чем характеризуются эти множества? Это можно описать совершенно формально. Позиция x является выигрышной (т. е. $x \in V$), если *существует* ход (т. е. дуга орграфа), ведущий из x в заключительную позицию или позицию из множества P. В свою очередь, позиция x является проигрышной (т. е. $x \in P$), если любой ход из этой позиции ведет в позицию из множества V. Это позволяет прямо на орграфе игры расставить знаки «+» и «–» по следующему правилу: знаком «–» отмечаются все заключительные и проигрышные позиции, а знаком «+» отмечаются все выигрышные. Фактически это некий рекурсивный алгоритм: сначала знак «–» выставляется у всех заключительных вершин, затем выставляется знак «+» у каждой вершины, для которой конец выходящей из нее дуги помечен знаком «–», затем выставляется знак «–» у каждой вершины, для которой все выходящие из нее ду-

[1] Более того, мы советуем вернуться к рассмотрению задания 10 из § 57 и построить такой более компактный орграф, чтобы с его помощью найти оптимальную стратегию.

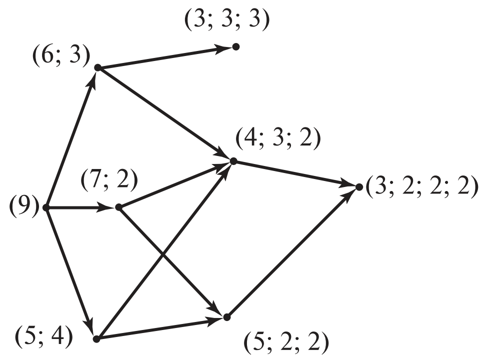

(3; 3; 3)
(6; 3)
(4; 3; 2)
(7; 2)
(9)
(3; 2; 2; 2)
(5; 4)
(5; 2; 2)

Рис. 7.4

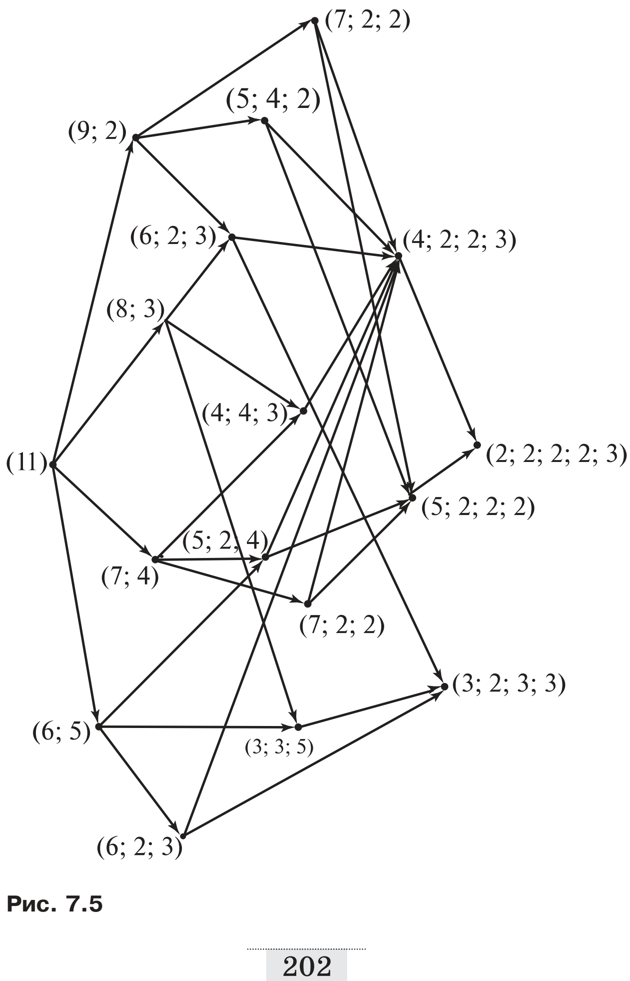

(7; 2; 2)
(5; 4; 2)
(9; 2)
(6; 2; 3)
(4; 2; 2; 3)
(8; 3)
(4; 4; 3)
(11)
(2; 2; 2; 2; 3)
(5; 2; 2; 2)
(5; 2, 4)
(7; 4)
(7; 2; 2)
(3; 2; 3; 3)
(6; 5)
(3; 3; 5)
(6; 2; 3)

Рис. 7.5

ги имеют конец, помеченный знаком «+», и т. д. На рисунках 7.6—7.9 показано, как этот процесс выглядит для графов, изображенных на рисунках 7.4 и 7.5 (чтобы картина была прозрачнее, мы на этих орграфах убрали подписи позиций около вершин).

После такой разметки становится очевидным, какой игрок выигрывает: тот, кто ходит первым, или тот, который ходит вторым[1]. Более того, стратегия игры также становится очевидной: надо искать дугу, выходящую из вершины, отмеченной знаком «+», и ведущую в вершину со знаком «–». Именно это правило сформулировано в § 58. Кроме того, там для игр, в которых каждая позиция описывается значениями двух переменных, предложено иное изображение: кодирование позиций клетками прямоугольной таблицы или, что фактически то же самое, точками на координатной плоскости. Вот как, к примеру, может выглядеть такое изображение для игры, предложенной в задании 11 к § 57. В таблице 7.1 представлены два первых шага заполнения (клетки заключительных позиций не только отмечены знаком «–», но и закрашены).

Таблица 7.1

n																					
20	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
19	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
18	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
17	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
16	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
15	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
14	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
13	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	
12	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	
11	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	
10	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	
9	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	
8	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	
7	+	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	
6			+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	
5						+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	
4						+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	
3						+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	
2							+	+	+	+	+	+	+	+	+	+	+	+	–	–	
1							+	+	+	+	+	+	+	+	+	+	+	+	+	–	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	*m*

[1] Глядя на граф, изображенный на рисунке 7.9, видно, что в игре, представленной в задании 9б, выигрывает второй игрок.

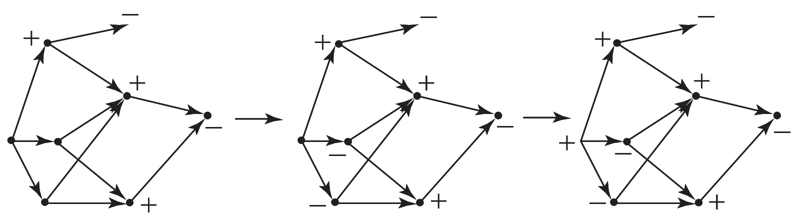


Рис. 7.6

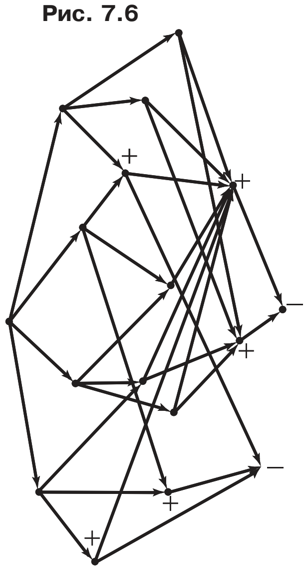


Рис. 7.7

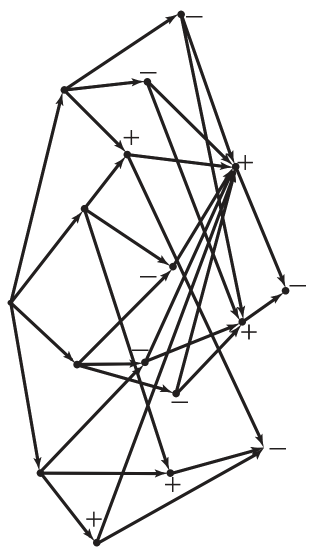


Рис. 7.8

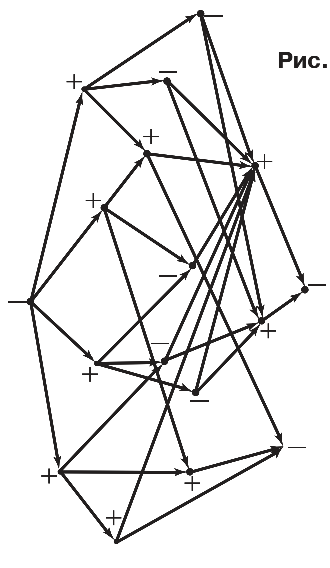


Рис. 7.9

Таблица 7.2

n																				
20	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
19	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
18	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
17	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
16	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
15	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
14	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–
13	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–
12	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–
11	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–
10	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–
9	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–
8	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–
7	+	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–
6	+	–	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–
5		+		+	–	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–
4					+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–
3						+	+	+	+	+	+	+	+	+	+	+	+	–	–	–
2		+			+	–	+	+	+	+	+	+	+	+	+	+	+	+	–	–
1						+	+	+	+	+	+	+	+	+	+	+	+	+	+	–
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	m

В таблице 7.2 представлены два следующих шага, причем вновь появившиеся позиции со знаком «–» тоже закрашены.

В таблице 7.3 представлены очередные два шага, и вновь появившиеся позиции со знаком «–» закрашены.

До полного заполнения осталось еще раз проставить знаки «–» и «+», но мы остановимся и заметим, что начальные позиции, которые указаны в задании 11 из § 57, уже размечены (в нашей таблице эти клетки обведены толстой рамкой). Сразу становится ясно, что в позиции (4; 4) начинающий игрок при правильной игре второго игрока проигрывает, а в позиции (2; 3) начинающий игрок выигрывает.

Хотя таблица выглядит довольно громоздкой, она занимает не больше места, чем граф игры, а заполняется очень легко. Да и заполнять минусами так много заключительных позиций тоже необязательно, достаточно обозначить границу, переход через которую уже дает выигрыш. Но са-

Таблица 7.3

n																					
20	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
19	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
18	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
17	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
16	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
15	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
14	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
13	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	
12	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	–	
11	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	–	
10	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	–	
9	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	–	
8	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	
7	+	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	
6	+	–	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	
5	–	+	–	+	–	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	
4	+	–	+	–	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	–	
3		+		+	–	+	+	+	+	+	+	+	+	+	+	+	+	–	–	–	
2		+	+	–	+	–	+	+	+	+	+	+	+	+	+	+	+	+	–	–	
1				+	–	+	+	+	+	+	+	+	+	+	+	+	+	+	+	–	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	m

мое главное — для обоснования, почему выигрывает тот или иной игрок, уже фактически не требуется никаких комментариев[1]. Кроме того, мы видим, что одна таблица обслуживает сразу все возможные варианты начальных позиций (варианты заданий С3 нередко различаются именно этим).

[1] Мы, однако, хотим предупредить, что поскольку изучение методов решения игровых задач практически отсутствует в подавляющем большинстве учебников информатики, то и эксперты, проверяющие на местах часть С, зачастую не готовы к проверке таких заданий, если они решаются не тем методом, который указан в рекомендациях по проверке. В Свердловской области, где проходила апробация данного учебника, мы ожидали экзаменационных работ, в которых будет применен данный метод решения игровой задачи (и наши ожидания оправдались), поэтому на курсах подготовки экспертов этот метод специально рассматривался. Однако за другие регионы мы ответственности не несем.

Переход от рассмотрения дерева игры к формулированию стратегии на языке выигрышных и проигрышных позиций — только одна из принципиально важных идей, которые обсуждаются в § 58. Вторым идейным концентром выступает дальнейшее уточнение понятия стратегии. Напомним, что в § 57 стратегия была определена как правило совершения ходов, при соблюдении которого игрок добивается выигрыша при любых ответных ходах другого игрока (или других игроков, если их несколько). Такое определение оказывается весьма похожим на определение алгоритма, которое дается в некоторых школьных учебниках информатики [1]. Но даже если мы объяснили различие между стратегией и алгоритмом (алгоритм — это организованная последовательность действий некоторого исполнителя, направленная на решение определенной задачи, а стратегия — правила, следуя которым надо выстраивать последовательность действий), у учащихся, как правило, остается ощущение непонимания. И если в такой момент попросить их привести примеры, демонстрирующие это различие, то обычно они этого сделать не могут. Дело в том, что и стратегия нередко описывается в форме алгоритма, только такой алгоритм предназначен не тому исполнителю, который будет осуществлять действия по преобразованию позиции (передвигать шахматную фигуру, забирать камни из кучки, разрезать прямоугольник на части и т. п.), а тому, кто будет давать указания исполнителю на выполнение определенных действий. В учебнике это продемонстрировано соответствующим примером и делается вывод, что стратегия нередко бывает представлена алгоритмом планирования. Такое понимание стратегии позволяет, с одной стороны, приучить учащихся избегать суетного употребления этого термина (которое, увы, нередко встречается в обыденной жизни, да и не только в ней), а с другой стороны, использовать его для обозначения подходящих под это определение реалий.

[1] Процитируем только один из учебников (Информатика: Учеб. для 6—7 кл. средней шк. / Под ред. Н. В. Макаровой. — СПб.: Питер, 1998): «Алгоритм — совокупность правил выполнения определенных действий, обеспечивающих решение задачи». Вот, к примеру, задача — изобличить преступника. Следователь выполняет определенные действия для решения этой задачи. Имеются правила выполнения этих действий: нельзя производить обыск без санкции прокурора. Эти правила собраны в так называемый Процессуальный кодекс. Но никому не придет в голову назвать Процессуальный кодекс алгоритмом. Или другой пример, более близкий школьной жизни. Ученик в перемену хочет перекусить в школьном буфете. Он для этого выполняет некоторые действия, можно даже представить их в виде алгоритма. Существуют правила поведения ученика в школе, например запрещающие толкаться. Но разве совокупность правил поведения — это алгоритм?

Третий идейно важный аспект, который рассматривается в § 58, — это понятие эквивалентных игр[1]. Вообще говоря, замена одного объекта на эквивалентный ему в том или ином смысле — стандартная идея, лежащая в основе моделирования. Так что и здесь можно смотреть на замену одной игры на другую игру как на процесс моделирования. Однако важно подчеркнуть, что в данном случае оба объекта совершенно равноправны, и поэтому не имеет принципиального значения, что для чего выступает моделью. Фактически мы имеем дело с описанием одного и того же процесса на разных языках. Умение видеть это и тем более самостоятельно переходить от описания на одном языке к описанию на другом — чрезвычайно важный элемент мышления, развивать который нужно при каждом удобном для этого случае. Практика ЕГЭ показала, что, когда в 2008 г. вместо уже привычных задач про кучки с камешками была сформулирована задача о точках на плоскости, многие выпускники не смогли увидеть, что это «та же Дуня в другом сарафане».

Наконец, четвертый аспект — понятие эвристики. Скажем сразу: единой точки зрения на понятие эвристики нет. Психологи обычно понимают под эвристикой решение, принимаемое интуитивно. Другая крайняя точка зрения состоит в том, что эвристика — это алгоритм, который на данный момент не получил достаточного обоснования. С этой точки зрения любая эвристика по мере накопления знаний должна превратиться в алгоритм. Мы в учебнике привели точку зрения, которую высказывают многие ученые, но конкретно эта формулировка принадлежит Анри Пуанкаре. Она занимает промежуточную позицию между двумя сформулированными выше. С одной стороны, правило, сокращающее перебор вариантов, действительно может со временем превратиться в алгоритм. С другой стороны, это правило может возникать на интуитивной основе. Однако от чисто интуитивного решения оно отличается тем, что должно быть четко сформулировано. Можно образно сказать, что эвристика — это формализованная интуиция.

Отметим в связи с этим, что сформулированный нами вопрос в задании 3 может оказаться весьма непростым для учащихся. Главное, на наш взгляд, отличие определения эвристического способа обработки информации и определения эвристики состоит том, что в первом случае речь идет о принятии решения в ходе исполнения последовательности действий по обработке информации, а эвристика — это правило, сформулированное до начала обработки информа-

[1] В теории игр используется более точный термин — изоморфизм игр. Однако термин «изоморфизм» сложнее для понимания школьником, поэтому мы остановились на понятии эквивалентности.

ции. Тем самым эвристики могут быть включены в тот или иной алгоритм (такие алгоритмы получили название эвристических, хотя само сочетание «эвристический алгоритм» выглядит несколько противоречиво).

Для заданий 1 и 2 из § 58 ответы даны в объяснительном тексте параграфа, задания 3 и 4 разобраны выше. При выполнении задания 5 некоторые учащиеся предлагают рисовать трехмерный параллелепипед (трехмерный массив, аналог двумерной таблицы) и в нем размечать кубики знаками «+» и «–». Это разумная идея, хотя и трудно реализуемая. Мы же просто приведем ответы в виде троек чисел, в которых на первом месте записано количество камней в первой кучке, на втором — количество камней во второй кучке, на третьем — количество камней в третьей кучке. Итак, для пункта *а* проигрышными позициями являются (0; 1; 1), (1; 0; 1), (1; 1; 0), (0; 2; 2), (2; 0; 2), (2; 2; 0), (0; 3; 3), (0; 4; 4), (1; 2; 3), (1; 3; 2), (1; 4; 5), (2; 4; 6); для пункта *б* проигрышными позициями являются (0; 1; 1), (1; 0; 1), (1; 1; 0), (0; 2; 2), (2; 0; 2), (2; 2; 0), (0; 3; 3), (3; 0; 3), (3; 3; 0), (0; 4; 4), (0; 5; 5), (1; 2; 3), (1; 3; 2), (2; 1; 3), (2; 3; 1), (3; 2; 1), (3; 1; 2), (1; 4; 5), (1; 5; 4), (2; 4; 6), (2; 5; 7), (3; 4; 7), (3; 5; 6). Важно, чтобы это задание было выполнено и проверено до того, как начнется изучение материала в § 59, поскольку рассматриваемый в этом параграфе пример игры опирается на решение этой задачи.

В связи с попытками использовать для решения трехмерный аналог таблицы можно предложить учащимся следующее задание:

Задание 29. Два игрока играют в следующую игру. Перед ними лежат три кучки камней, в первой из которых 2 камня, во второй — 3 камня, в третьей — 4 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или удваивает число камней в какой-то кучке, или добавляет по 2 камня в каждую из кучек. Выигрывает игрок, после хода которого либо в одной из кучек становится не менее 15 камней, либо общее число камней во всех трех кучках становится не менее 25. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

На самом деле здесь надо построить орграф, в котором каждая позиция присутствует ровно один раз, и разметить в нем вершины знаками «+» и «–».

В задании 6а нужно провести определенное комбинаторное рассмотрение. Сначала все просто. После того как поставлен первый символ (пусть для определенности это будет крестик), остается 8 вариантов хода. Значит, из корневой вершины дерева выходит 8 ребер. Следующий ход может быть выбран 7 способами, значит, из каждой вершины выходит уже по 7 ребер, затем из каждой вершины выходит по 6 ребер, потом по 5, далее по 4. Уже на этом шаге в графе получается $8 + 56 + 336 + 1680 + 6720 = 8800$ ребер. И вот здесь надо задуматься, поскольку некоторые из вершин этого дерева уже могут оказаться листьями. Таких вершин столько, сколько есть различных позиций с тремя крестиками подряд и где-то стоящими двумя нуликами. Существует ровно 8 позиций, в которых 3 крестика идут подряд: они расположились на одной из трех горизонталей, на одной из трех вертикалей или на одной из двух диагоналей. Из оставшихся 6 клеток 2 заняты нуликами, это может быть сделано $\frac{6 \cdot 5}{2} = 15$ способами. Тем самым имеется 15 позиций, после которых нет продолжения. Надо только понять, на каких ребрах расположатся эти вершины, поскольку каждая из них будет фигурировать в этом графе не единожды. Одна и та же позиция возникает при разной очередности простановки крестиков и нуликов. Для крестиков таких вариантов $3! = 6$, а для нуликов $2! = 2$. Тогда общее число вариантов равно $6 \cdot 2 = 12$. Значит, количество позиций, которые будут листьями на этом графе, равно $12 \cdot 15 = 180$. Для оставшихся $6720 - 180 = 6540$ вершин из каждой такой вершины выходит по 3 ребра, т. е. в графе окажется еще $6540 \cdot 3 = 19\,620$ ребер. Опять некоторые из вершин этих ребер окажутся листьями — в точности те, которые соответствуют позициям, где есть три подряд стоящих нулика, но нет трех подряд идущих крестиков. Вариантов выстраивания трех нуликов в ряд по-прежнему 8. Количество вариантов расположения трех крестиков при заданном расположении трех нуликов равно $\frac{6 \cdot 5 \cdot 4}{2 \cdot 3} = 20.$ Для каждого варианта, когда нулики располагаются в одной строке или одном столбце, есть по 2 варианта расположения крестиков в параллельной строке или параллельном столбце. Значит, общее число позиций равно $6 \cdot 18 + 2 \cdot 20 = 148$. Снова подсчитаем, в скольких вершинах из 19 620 стоят эти позиции: $148 \cdot 3! \cdot 3! = 5328$. Следовательно, число ребер, из которых выходит по 2 ребра, равно $19\,620 - 5328 = 14\,292$. Значит, количество новых ребер равно 28 584. Таким образом, в графе игры уже $8800 + 19\,620 + 28\,584 = 57\,004$ ребра. Среди 28 584 вершин какие-то опять окажутся листьями, а из остальных выходит по одному ребру. Мы уже не будем подсчитывать количество этих вершин (это делается аналогично, и терпе-

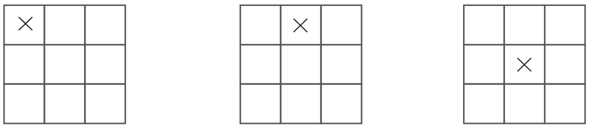


Рис. 7.10

ливый читатель может сделать это самостоятельно), поскольку главная цель задания — показать, насколько большим может оказаться дерево игры, — достигнута. Эти вычисления способны остудить пыл даже у самых ярых сторонников программировать любую задачу, не оценивая предварительно, в какие объемы данных это может вылиться. Но важнее, что эти вычисления показывают, насколько эффективнее оказывается идея рассмотрения не дерева игры, а множества позиций.

Что касается заданий 6б и 7, то они весьма трудоемки. Мы предлагаем задавать их выполнение в «распределенном» режиме. Например, при выполнении задания 6б после первого хода возникает 3 позиции (в оговоренных условиях не различать позиции, отличающиеся поворотом и симметрией). Они представлены на рисунке 7.10.

Теперь каждую позицию дальше анализирует отдельный ученик. К анализу получающихся вариантов снова подключаются другие учащиеся и т. д. Надо постоянно отслеживать эквивалентные варианты и тем самым уменьшать перебор. Даже при таком «распределенном» решении этой задачи перебор получается весьма значительным. Это мотивирует на поиск принципиально других путей решения. Один из таких путей описан в § 60.

В задании 8 легко увидеть, что в каждом столбце (да и в каждой строке) может стоять не более одного знака «–». Значит, количество проигрышных позиций, т. е. число n, не превосходит $\min(k, m)$. Важно, чтобы этот ответ учащиеся получили до выполнения лабораторной работы № 27, которая представляет данный параграф в компьютерном практикуме. На выполнение этой работы требуется, по нашему мнению, не менее одного полного урока.

В § 59 учащиеся знакомятся, пожалуй, с самым мощным и эффективным средством построения стратегии, построением инварианта стратегии. Суть понятия «инвариант» та же, что и раньше, — неизменность при выполнении тех или иных операций. Для игр инвариантом выступает свойство, которое выполняется для всех проигрышных позиций и не выполняется ни для какой выигрышной позиции. Иными словами, это свойство нарушается при переходе от проигрышной позиции к выигрышной и восстанавливается при переходе от выигрышной позиции к проигрышной.

В учебнике инвариант стратегии продемонстрирован на примере игры Ним. Учащиеся не так легко воспринимают операцию поразрядного сложения, если они не встречались с ней ранее. Именно поэтому мы настоятельно рекомендовали познакомить учащихся с этой операцией при изучении кодов, исправляющих ошибки (см. обсуждение темы 7). Более того, использование свойств этой операции облегчает как доказательство того, что данное свойство является инвариантом стратегии, так и нахождение нужного хода для перевода выигрышной позиции в проигрышную. Действительно, пусть $a \oplus b \oplus c = s$, где $a \oplus b \oplus c$ — количество камней в кучках. Если $s \neq 0$, то выбираем ту кучку, где количество камней больше, чем s. Пусть, например, это c. Тогда рассмотрим $d = c \oplus s$. Это то число, которое должно остаться после выполнения хода. Действительно,

$$a \oplus b \oplus d = a \oplus b \oplus (c \oplus s) = (a \oplus b \oplus c) \oplus s = s \oplus s = 0.$$

Если же $a \oplus b \oplus c = 0$, то изменение любого слагаемого равносильно поразрядному сложению с некоторым ненулевым числом. Но тогда и сумма станет неравной нулю.

Эти рассуждения лежат в основе решения задач 4 и 5 из § 59. В задании 4 мы прямо рекомендуем запрограммировать данную игру с помощью операции XOR, а в задании 5 легко видеть, что данный инвариант — поразрядная сумма количеств камней в кучках равна нулю — верен и в обобщенной игре Ним.

Значительное внимание в этом параграфе уделено поиску игрового инварианта на основе так называемых симметричных стратегий. Иногда это дословная геометрическая симметрия, а иногда это некоторая аналогия. Для лучшего освоения этой идеи учащимся можно сначала предложить несколько игр, в которых симметрия используется в буквальном смысле. Вот примеры таких игр.

Задание 30. На прямоугольный стол по очереди выкладываются пятаки, причем никакой пятак не должен налегать на ранее положенные монеты. Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре?

Задание 31. В каждой клетке доски 11×11 стоит шашка. За ход разрешается снять с доски любое количество подряд идущих шашек либо из одного вертикального, либо из одного горизонтального ряда. Выигрывает игрок, снявший последнюю шашку. Кто выигрывает при правильной игре?

В игре, описанной в задании 30, первый игрок кладет свой пятак так, чтобы центр монеты совпадал с центром стола. Затем на каждый ход второго игрока он кладет монету симметрично относительно центра стола. Инвариант проигрышных позиций — симметричность расположения выложенных на стол монет относительно центра стола.

В игре, описанной в задании 31, игрок, делающий первый ход, снимает все шашки из вертикального ряда, образующего ось симметрии для данного квадрата. После этого на любой ход второго игрока первый игрок отвечает ходом, симметричным относительно этой оси (этот ход автоматически получается на другой половинке доски, нежели ход второго игрока). Инвариант проигрышных позиций — симметричность расположения оставшихся шашек относительно указанной оси симметрии.

В заданиях к § 59 преимущественно рассматриваются именно симметричные стратегии. Выполняя задание 3, учащиеся обычно легко догадываются, что одним ходом первый игрок может выровнять число конфет в обеих коробках и тем самым свести задачу к той, что была разобрана в объяснительном тексте.

В задании 6 описана классическая для учебников информатики игра Баше. Она довольно часто встречается на математических олимпиадах, поэтому не исключено, что учащиеся ее знают [1]. Поэтому мы сразу опишем выигрышную стратегию. Если n делится на $k + 1$, то выигрывает второй игрок, придерживаясь следующего правила: если первый игрок взял m камней, то ему следует взять $k + 1 - m$ камней. Если же n не делится на $k + 1$, то выигрывает первый игрок. Для этого он сначала берет из кучки столько камней, каков остаток при делении на $k + 1$ числа n, а затем следует указанной стратегии. Инвариантом проигрышных позиций является, как выше объяснено, нулевой остаток при делении числа камней в кучке на $k + 1$.

В задании 7 при $k = m$ представленная в нем игра является классической игрой Баше. Поэтому имеет смысл рассматривать только случаи, когда $k > m$ и когда $k < m$.

При $k > m$ всегда выигрывает первый игрок. Его действия таковы. Если n не делится на $m + 1$, то он ведет себя так, как будто ему разрешается брать не более m камней, т. е. играет в обычную игру Баше. Если n делится на $m + 1$, то первым ходом он забирает $m + 1$ камней, после чего играет в обычную игру Баше по вышеприведенному правилу.

[1] Она разбирается, например, в учебнике: Математика: Учеб. для 5 кл. общеобразоват. учреждений / Л. Н. Шеврин, А. Г. Гейн, И. О. Коряков, М. В. Волков. — М.: Просвещение, 2001.

Пусть теперь $k < m$. Если $n \leqslant k$, то первый игрок забирает все камни и выигрывает. Если же $n > k$, то после хода первого игрока второй игрок оказывается в ситуации, рассмотренной в предыдущем абзаце. Следовательно, он выигрывает.

В игре, описанной в задании 8, всегда выигрывает первый игрок. Своим ходом он соединяет отрезком две точки так, чтобы по каждую сторону от этой хорды осталось по 9 точек. Затем применяется симметричная стратегия.

Идея решения задачи 9а близка к идее решения задачи 8. Первым ходом игрок в случае нечетного числа монет забирает одну монету, лежащую в середине, разрывая тем самым цепочку монет на два одинаковых куска. Затем он реализует симметричную стратегию. Если же количество монет четно, то из середины цепочки надо забрать 2 монеты и дальше действовать симметричным образом.

В задании 9б всегда выигрывает второй игрок. Дело в том, что после первого хода игра становится эквивалентной игре из пункта *а*. Практика, однако, показывает, что учащиеся далеко не сразу это замечают.

В задании 10 пусть для определенности m четно. Если $m > 2$, то можно сразу применить симметричную стратегию, разрезав первым ходом прямоугольник на две равные части. Надо только вовремя прекратить следование этой стратегии — как только противник отрежет полоску шириной в 1 клетку, надо тут же от нее отрезать требуемый квадрат 1×1. Если же $m = 2$, то резать полоску можно лишь поперек длинной стороны, причем так, чтобы каждый раз получались полоски длиной не менее 2. Теперь в этой постановке легко узнать задачу о разбиении на кучки так, чтобы в каждой из получающихся кучек было более одного камня (задача из объяснительного текста § 57). Ее решению посвящено задание 11, поэтому задание 10 надо предлагать после того, как выполнено задание 11.

В задании 11а фактически снова применяется симметричная стратегия. Выполняя задание пункта *б*, учащиеся должны сначала экспериментально обнаружить, для каких небольших нечетных n позиция оказывается проигрышной. Ранее были проанализированы для n значения 9, 11 и 13 (см. объяснительный текст § 57 и задание 9 к этому параграфу). Легко понять, что значение 7 тоже проигрышное. Возникает гипотеза, что проигрышными для нечетных значений n являются все позиции, дающие в остатке 3 при делении n на 4, т. е. $n = 4k + 3$, где k — натуральное число. Она верна и доказывается методом математической индукции.

В компьютерном практикуме материал этого параграфа представлен в лабораторной работе № 28. По-видимому, на ее выполнение потребуется два урока. Мы оставляем на усмотрение учителя решать, какие требования предъявлять к интерфейсу тех игровых программ, которые составляют учащиеся.

Заключительный параграф возвращает учащихся к постановкам, прозвучавшим в самом начале изучения главы. В той модели, которая обсуждается в § 60, представлены все аспекты управления, включая канал обратной связи. Полигоном для изучения данного метода служит игра «Крестики-нолики». Напомним, что ранее мы обсуждали мотивационный аспект, позволяющий аргументировать применение оценочного подхода к построению игровой стратегии в этой игре. Мы вполне отдаем себе отчет, что материал здесь довольно сложный, поэтому нами было принято решение предъявить программу игры в явном виде, что и сделано в лабораторной работе № 29. Один час наверняка уйдет на реализацию, отладку и эксперименты с программой игры в обычные «Крестики-нолики». Следующие 2 часа можно потратить на выполнение заданий 3 и 4 этой лабораторной работы. Если имеется достаточный резерв времени, то можно реализовать проект по программированию игры «Реверси», описанной в задании 5 к § 60 [1].

[1] Довольно подробный разбор построения алгоритма для игры «Реверси», доступный для большинства учащихся, приведен в кн.: Информатика: Учеб. для 8—9 кл. с углубл. изуч. информатики / А. И. Сенокосов, А. Г. Гейн. — М.: Просвещение, 1995.

Подготовка к Единому государственному экзамену по информатике

В тематическом планировании, рекомендуемом нами для реализации предлагаемого курса информатики, предусмотрен раздел «Подготовка к ЕГЭ». Мы убеждены, что изучение курса информатики в соответствии с нашим учебником полностью обеспечивает подготовку к ЕГЭ[1]. Тем не менее практика показывает, что сама форма экзамена и стиль его проведения являются весьма существенным фактором, влияющим на успешность его сдачи выпускниками. Поэтому, с одной стороны, мы поместили в конце учебника контрольно-измерительные материалы в тестовой форме, близкие по содержанию к тому, что обычно предлагается на ЕГЭ. Они предназначены для самостоятельной работы учащихся с целью самооценки уровня подготовленности к экзамену. С другой стороны, мы призываем учителя активно использовать материалы по подготовке к ЕГЭ, размещенные в открытом сегменте интернет-портала Федерального института педагогических измерений (ФИПИ).

В этом разделе книги мы хотим дать некоторые рекомендации, основанные на опыте работы как с учащимися, сдававшими впоследствии ЕГЭ по информатике, так и в составе группы экспертов, проверявших часть С.

Прежде всего следует сказать, что ЕГЭ по информатике является экзаменом по выбору, и составителями он позиционируется как экзамен, ориентированный на выпускников, которые планируют продолжить обучение в вузах по специальностям, где предъявляются повышенные требования к подготовке по информатике и программированию. Это предопределило выбор тем, по которым ведется проверка знаний и умений, а также весьма высокий уровень требований к алгоритмической и программистской подготовке. Достаточно сказать, что три из четырех заданий части С относятся именно к алгоритмизации и программированию, причем повышенного и высокого уровней сложности. При этом составите-

[1] Наша убежденность основывается на результатах сдачи ЕГЭ выпускниками, обучавшимися по этим учебникам. Средний балл школьной оценки составил 4,9 (здесь не учтены 3 призера Всероссийской олимпиады по информатике, которым согласно Положению автоматически выставлялось 100 баллов).

ли ЕГЭ прямо указывают, что такие темы, как «Социальная информатика», «Основные устройства информационных и коммуникационных технологий», «Технология обработки текстовой информации», даже не представлены в заданиях ЕГЭ. Кроме того, по сравнению с ЕГЭ по курсу математики на ЕГЭ по информатике планка школьной отличной оценки установлена весьма высоко. Чтобы ее получить, необходимо полностью выполнить задания частей А и В и по крайней мере одно задание (причем самое сложное) из части С. При выполнении ряда заданий существенную роль играет уровень математической подготовленности выпускников. В частности, у многих участников ЕГЭ он оказался недостаточным для успешного выполнения задания С1. Наблюдая за изменениями вариантов год от года, можно отметить, что требования к математической подготовке выпускников, сдающих информатику, неуклонно возрастают.

Все это приводит к выводу, что принимать участие в этом экзамене целесообразно только тем учащимся, которые прошли обучение в рамках углубленного курса информатики и предполагают поступление в те вузы, где данный предмет является доминирующим в учебной программе.

По нашему убеждению, эту информацию следует довести до сведения учащихся не позже начала 11 класса. К сожалению, многие учащиеся, успешно освоившие информационные технологии, наивно предполагают, что это может стать залогом успешной сдачи ЕГЭ по информатике. Увы, это ошибочная точка зрения.

Определившись с контингентом учащихся, предполагающих сдавать ЕГЭ по информатике, обсудим, какие шаги, по нашему мнению, следует предпринять учителю для организации подготовки к этой форме экзамена. Такая подготовка имеет два аспекта: содержательный и технологический.

Содержательный аспект

Работа учителя по подготовке учащихся к ЕГЭ в содержательном аспекте может быть разбита на три фазы: постановочную, подготовительную и контролирующую. В постановочной фазе учитель определяет основные параметры работы по подготовке учащихся к ЕГЭ. Подготовительная фаза — это деятельность учителя по созданию дидактических материалов, работая с которыми учащиеся приобретают необходимые знания и умения по выполнению заданий ЕГЭ. Наконец, контролирующая фаза — это та основная часть работы, в ходе которой происходит научение учащихся выполнению заданий ЕГЭ, а также период комплексной проверки готовности учащихся к ЕГЭ, по результатам которой осуществляется необходимая (и заблаговременная) коррекция процесса подготовки.

Постановочная фаза

1. Анализ, что будет и чего не будет на очередном ЕГЭ.

На первом шаге этот анализ основывается на сопоставлении кодификатора и плана экзаменационной работы. Отметим, что разработчиками ЕГЭ весь обязательный минимум, который является составной частью федерального компонента Государственного стандарта общего образования, разложен на составляющие элементы и каждому элементу присвоен код. Эта информация содержится в кодификаторе, который представлен на сайте ФИПИ. В свою очередь, для каждого задания, содержащегося в тесте, указан код того составляющего элемента, на проверку которого он, по мнению составителей, направлен. Сделано это и в демонстрационной версии варианта экзамена, которая также располагается на сайте ФИПИ. Поэтому учителю прежде всего надо, используя демоверсию, выделить в кодификаторе те темы и элементы, на изучение которых требуется обратить особое внимание, поскольку они войдут в вариант очередного ЕГЭ.

2. Анализ того, как именно выделенные на первом шаге элементы будут представлены в ЕГЭ.

Напомним, что в ЕГЭ три части: А, В и С. Во-первых, надо выявить, какие элементы в каких частях представлены. Во-вторых, для каждого элемента надо определить, какой вид деятельности предусмотрен составителями ЕГЭ для этого элемента. В-третьих, оценить уровень трудности задания, в котором данный элемент представлен.

При выполнении этого этапа подготовки определяется уровень, на котором будет вестись подготовка, — базовый (в этом случае прогнозируемая максимально возможная школьная оценка — 4) или профильный (т. е. с ориентацией на поступление в вуз, где предъявляются повышенные требования к подготовке абитуриента по информатике). Все остальные фазы подготовки ведутся с учетом решений, принятых на этом этапе.

Фаза подготовки

1. Подготовка комплектов учебных (тренировочных) тестовых заданий на каждый тематический блок элементов в той форме, в какой они встречаются в демонстрационном варианте.

Варианты этих заданий могут использоваться по мере изучения той или иной темы, представленной в ЕГЭ. Главная цель — научить учащихся основным эффективным приемам выполнения заданий ЕГЭ по данной теме, дать им возможность привыкнуть к форме предъявления материала по этой теме.

2. Подготовка комплексного варианта ЕГЭ.

Основная цель — научить учащихся выстраивать стратегию прорешивания варианта ЕГЭ, которая, исходя из персональных достижений учащегося, определяет порядок выполнения им заданий. Надо прямо сказать, что учащиеся поставлены в весьма жесткие временные рамки, поэтому, определяя оптимальную стратегию выполнения заданий, учащийся должен уметь быстро и адекватно оценивать, стоит браться за данное задание немедленно или отложить его выполнение на более позднее время.

В качестве варианта для такого тестирования может быть взят демонстрационный вариант ЕГЭ за прошедший и текущий учебный год. Имеющаяся практика подготовки к ЕГЭ по информатике показывает целесообразность проведения как минимум двукратного комплексного тестирования в ходе подготовки.

Фаза контроля

1. Провести тренировочные тематические тестирования. Определить основные проблемы в подготовке по отдельным темам. Осуществить коррекцию планов подготовки.

2. Провести тренировочное тестирование по комплексному варианту. Выявить темы, нуждающиеся в дополнительной проработке.

Технологический аспект

Только на первый взгляд кажется, что оформление — дело простое.

1) Если ученик никогда до этого не работал с бланком ЕГЭ, то ему на проставление ответа (часть А) или вписывание ответа (часть В) требуется больше времени, чем ученику, уже работавшему с таким бланком. Главная причина замедления — боязнь испортить бланк. И его действительно нельзя портить — бланк не заменяется. А это время, которого на самом деле не так много.

2) Допускается исправление ошибочных ответов. Но и это осваивать во время экзамена уже поздно.

3) Надо избегать появления каких бы то ни было дефектов на бланке № 1. Бланк сканируется и обрабатывается автоматически! Дефекты могут восприниматься как ошибки.

4) При вписывании ответов (часть В) нужно придерживаться указанных образцов символов. Без привычки это тоже трудно. Каждый символ (например, запятая) должен стоять в отдельной клеточке, причем в отведенном для него месте клеточки (не должна стоять запятая в середине клетки, поскольку тогда распознающей системой она будет восприниматься как другой символ).

5) При описании решений части С не должно быть ничего лишнего. Все должно уместиться на бланке, причем в естественной последовательности. Учащихся необходимо приучать к мысли, что эксперты, проверяющие часть С, не должны разгадывать их ребусы и кроссворды.

Высказанные нами тезисы о подготовке учащихся к сдаче ЕГЭ показывают, что, хотя в тематическом планировании часы на такую подготовку стоят в конце, их нужно разумно распределить на весь учебный год (по крайней мере начиная со второй четверти).

Ниже представлен разбор заданий частей А и В демонстрационного варианта ЕГЭ-2008 [1]. Для каждого задания:

- приводится правильный ответ;
- обсуждаются наиболее эффективные пути выполнения задания;
- рассматриваются типичные трудности, возникающие у учащихся при выполнении задания;
- обсуждается возможное варьирование задания и направления его изменений в будущем, приводятся примеры заданий в альтернативных формулировках.

Условия заданий воспроизведены нами дословно. Также сохранена форма предъявления заданий. Кроме того, мы воспроизводим стандартную преамбулу об обозначениях и приоритетах логических операций, которая сопровождает каждый вариант ЕГЭ.

1. Обозначения для логических связок (операций):

а) *отрицание* (инверсия, логическое НЕ) обозначается как $\neg$ (например, $\neg A$);

б) *конъюнкция* (логическое умножение, логическое И) обозначается как $\wedge$ (например, $A \wedge B$) либо & (например, A & B);

в) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается как $\vee$ (например, $A \vee B$) либо | (например, A | B);

г) *следование* (импликация) обозначается как $\rightarrow$ (например, $A \rightarrow B$);

д) символ 1 используется для обозначения истины (истинного высказывания), символ 0 — для обозначения лжи (ложного высказывания).

---

[1] Что касается части С, мы не включили ее в этот разбор по двум причинам. Во-первых, она имеет традиционную форму изложения ответов. Во-вторых, три задания из четырех относятся к программированию, которым учащиеся большей частью овладевали в ходе компьютерного практикума. На наш взгляд, научиться программированию без использования компьютера невозможно. Однако здесь учащихся подстерегает немалая опасность. Большинство сильных учащихся привыкли к тому, что ошибки синтаксиса исправляются в ходе трансляции программ. К сожалению, утрата ими навыков самостоятельного синтаксического контроля может привести к тому, что они будут терять баллы из-за допущенных синтаксических ошибок. Кроме того, чисто механически при простановке знака «;» точка прописывается слабее, что иногда приводит к ее потере при сканировании. Так рождаются ошибки-фантомы, которые тем не менее вполне реально влияют на оценку.

2. Два логических выражения, содержащие переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения $A \to B$ и $(\neg A) \vee B$ равносильны, а выражения $A \vee B$ и $A \wedge B$ нет (значения выражений разные, например при $A = 1$, $B = 0$).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), эквивалентность (равносильность). Таким образом, $\neg A \wedge B \vee C \wedge D$ совпадает с $((\neg A) \wedge B) \vee (C \wedge D)$. Возможна запись $A \wedge B \wedge C$ вместо $(A \wedge B) \wedge C$. То же относится и к дизъюнкции: возможна запись $A \vee B \vee C$ вместо $(A \vee B) \vee C$.

Часть А

A1. В кодировке Unicode на каждый символ отводится два байта. Определите информационный объем слова из двадцати четырех символов в этой кодировке.

1) 384 бит; 2) 192 бит; 3) 256 бит; 4) 48 бит.

Ответ: 1.

Обсуждение. В первую очередь учащиеся должны обратить внимание на то, что все результаты даны в битах, а в условии кодировка оговорена в байтах. Если количество информации будет выражено в байтах, то получится число 48 и ошибочно может быть выбран ответ 4. С другой стороны, если невнимательный ученик не учтет, что каждый символ кодируется двумя байтами, и сыграет роль обычно формирующийся стереотип, что каждый символ кодируется 8 битами, то получится число 192 и будет выбран ответ 2. Выбор числа 256 наименее вероятен, ибо оно возникает как результат возведения числа 2 в восьмую степень, т. е. это просто количество символов, которое можно закодировать однобайтовыми последовательностями из 0 и 1. Конечно, свою лепту в неправильный выбор ответа вносят арифметические ошибки, которые учащиеся допускают весьма регулярно.

Вариативность задания.

1) В реальных вариантах ЕГЭ вместо указания длины сообщения может быть приведено само сообщение. В этом случае учащийся должен самостоятельно найти длину сообщения. При этом возникают ошибки, связанные с тем, что в качестве символа не учитываются пробелы и знаки препинания.

2) Другим направлением варьирования данного задания является определение информационных параметров сообщения (длина сообщения, информационный объем и т. п.)

по аналогичным параметрам сообщений в разных кодировках. Вот пример такого задания:

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 240 бит. Какова длина сообщения в символах?

3) Можно ожидать, что задания этого типа будут перенесены из части А в часть В. В этом случае учащиеся должны быть особенно внимательны к требованиям, в каких единицах следует записать ответ.

A2. Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 18 различных сигналов?

1) 6; 2) 5; 3) 3; 4) 4.

Ответ: 3.

Обсуждение. Это задание на самом деле не связано напрямую с информатикой, а относится к разделу математики, традиционно называемому комбинаторикой. Учащиеся, знакомые с комбинаторикой хотя бы на начальном уровне, сразу должны определить, что n лампочек дают 3^n различных комбинаций своих состояний. Поскольку выполняется двойное неравенство $3^2 = 9 < 18 < 27 = 3^3$, двух лампочек недостаточно, а трех вполне хватает. Конечно, учащиеся могут рассуждать, непосредственно выписывая комбинации лапочек. Для одной лампочки на табло таких комбинаций по условию 3, для двух лампочек их получится 9, для трех — уже 27, из чего делается вывод о том, какой из предложенных вариантов ответа правильный. К неправильным ответам может привести как раз слабое знание комбинаторных схем. Например, ответ 6 получается, если 18 разделить на 3 (т. е., по мнению учащихся, необходимое количество лампочек, обозначенное через x, удовлетворяет уравнению $3x = 18$).

Вариативность задания.

1) В реальных вариантах ЕГЭ нередко варьируется количество возможных состояний кодирующего элемента (например, лампочки могут находиться в двух, а не в трех состояниях, клетки некоторой доски покрашены одним из трех цветов и т. п.).

2) Другим направлением варьирования данного задания является изменение конфигурации расположения кодирующих элементов: например, лампочки располагаются в два ряда одинаковой длины. Если, скажем, по-прежнему тре-

буется кодировать 18 сообщений, то потребуется два ряда по 2 лампочки в каждом, поскольку двух рядов по одной лампочке будет недостаточно.

3) В демоверсии 2009 г. это задание перенесено (с соответствующей модификацией) в часть В.

А3. Для передачи секретного сообщения используется код, состоящий из десятичных цифр. При этом все цифры кодируются одним и тем же (минимально возможным) количеством битов. Определите информационный объем сообщения длиной в 150 символов.

1) 600 бит; 2) 750 бит; 3) 1200 бит; 4) 60 байт.

Ответ: 1.

Обсуждение. Это задание по своей сути получено «скрещиванием» заданий А1 и А2: сначала надо найти длину кодирующей последовательности для одного символа (задача типа А2), а затем информационный объем всего сообщения (задача типа А1). Поскольку $2^3 = 8 < 10 < 2^4 = 16$, получаем, что цифры кодируются четырехбитовыми последовательностями. Умножая 4 на 150, получаем 600, что и определяет выбор ответа.

Вариативность задания.

1) Возможности варьирования здесь те же, что и в заданиях А1 и А2.

2) Другим направлением варьирования данного задания является замена задачи на обратную (например, по заданному информационному объему найти длину сообщения).

3) В реальном варианте задание было усложнено тем, что длина кода определялась из одних данных, а длина сообщения — из других. Вот пример такого задания:

В велокроссе участвуют 159 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 70 велосипедистов?

1) 70 бит; 2) 70 байт; 3) 159 бит; 4) 159 байт.

А4. Сколько единиц в двоичной записи десятичного числа 194,5?

1) 5; 2) 6; 3) 3; 4) 4.

Ответ: 4.

Обсуждение. Самый быстрый способ перевода числа 194 в двоичную систему заключается, на наш взгляд, в переводе этого числа алгоритмом деления в шестнадцатеричную или восьмеричную систему, а затем в расписывании его в двоичную. Поскольку $194 = C2_{16}$, то $194 = 11000010_2$. В свою очередь, $0{,}5 = \frac{1}{2} = 0{,}1_2$. Следовательно, $194{,}5 = 11000010{,}1_2$, откуда и получается требуемый ответ. Конечно, этот путь решения предполагает, что учащиеся владеют алгоритмом быстрого перевода чисел из шестнадцатеричной (восьмеричной) системы в двоичную.

Сильные учащиеся хорошо помнят степени числа 2 и осуществляют перевод в двоичную систему разложением числа в сумму степеней двойки. Конечно, такой путь решения тоже приемлем.

Вариативность задания.

1) В 2008 г. впервые в вариантах ЕГЭ появился перевод дробных чисел в двоичную систему счисления. Можно ожидать, что это тенденция, а не разовое явление.

2) Предложенный нами путь решения (через промежуточный перевод в шестнадцатеричную систему) продемонстрирован еще и потому, что традиционно в вариантах ЕГЭ на этом месте стоит задание, связанное с переводом чисел из одной системы, основание которой является степенью числа 2, в другую систему счисления с аналогичным основанием. Вот пример такого задания:

Число $A87_{16}$ *представьте в восьмеричной системе счисления.*

Оптимальный путь решения — перевести число в двоичную систему, преобразовав каждую шестнадцатеричную цифру в тетраду, а затем в полученном двоичном числе разбить все цифры на триады и каждую триаду записать восьмеричной цифрой.

Можно ожидать, что в будущих вариантах ЕГЭ аналогичные преобразования будут предложены уже для дробных чисел, поэтому полезно отработать алгоритмы преобразования шестнадцатеричных и восьмеричных дробей в двоичные и обратно.

3) В некоторых вариантах ЕГЭ вместо количества единиц требуется подсчитать количество ведущих нулей. Учащиеся должны понимать, что слово «ведущий» означает, что запись числа должна начинаться с цифры 1, если только 0 не означает «нуль целых». Неведущие нули могут возникать при расписывании шестнадцатеричных и восьмеричных цифр в двоичное представление.

4) Еще одна вариация этого задания была представлена в варианте второй волны. Вот задание такого типа:

Дано $A = 2528$, $B = AC16$. *Какое из чисел* C, *записанных в двоичной системе*, *отвечает условию* $A < C < B$?

1) 10101010_2; 2) 10101110_2;
3) 10101011_2; 4) 10101100_2.

A5. Вычислите сумму чисел x и y при $x = \mathrm{A6}_{16}$, $y = 75_8$. Результат представьте в двоичной системе счисления.

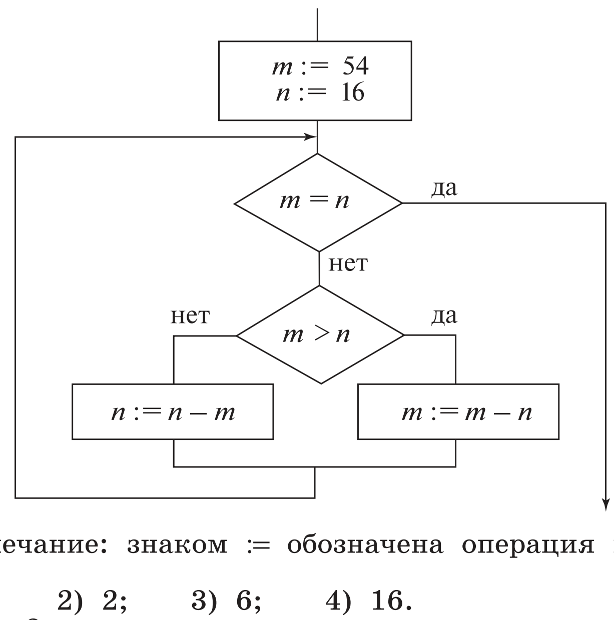
1) 11011011_2; 2) 11110001_2;
3) 11100011_2; 4) 10010011_2.

Ответ: 3.

Обсуждение. Здесь, как и в задании А4, оптимальный путь решения состоит в переводе каждого из чисел в двоичную систему счисления с последующим выполнением сложения. Поскольку $\mathrm{A6}_{16} = 10100110_2$ и $75_8 = 111101_2$, то их сумма равна 11100011_2. Как показывает практика, многие школьники предпочитают сначала перевести каждое число в десятичную систему счисления, затем найти сумму и результат перевести в двоичную систему. Такой многоходовый путь дает гораздо больше поводов допустить ошибку в вычислениях, чем тот способ, который предложен выше. Видимо, именно этим объясняется существенно более низкий процент выполнения этого задания по сравнению с заданием А4.

Вариативность задания. В вариантах предшествующих лет форма задания не менялась. В целом же оно может варьироваться в тех же направлениях, что и задание А4.

A6. Определите значение переменной m после выполнения фрагмента алгоритма.



Примечание: знаком := обозначена операция присваивания.

1) 1; 2) 2; 3) 6; 4) 16.

Ответ: 2.

Обсуждение. Данной схемой представлен алгоритм Евклида нахождения наибольшего общего делителя двух натуральных чисел. Этот алгоритм обязательно изучается в базовом курсе школьной информатики, поэтому учащиеся могут просто распознать его и получить ответ, не исполняя алгоритм по шагам. Конечно, такое распознавание весьма существенно экономит время выполнения задания, хотя и чревато ошибкой в том случае, если на самом деле алгоритм окажется лишь похожим на уже известный.

Если же учащиеся не распознали алгоритм и исполняют его по шагам, то типичной ошибкой является выполнение тела цикла либо на один раз меньше, либо, наоборот, на один раз больше.

Вариативность задания.

1) В данном примере использовано вложение конструкции ветвления в конструкцию цикла. Это наиболее сложное сочетание алгоритмических конструкций из тех, которые реально встречаются в вариантах ЕГЭ. Однако надо готовить учащихся к тому, что в будущем могут встретиться именно вложенные конструкции: цикл в ветвление, ветвление в цикл, двойной цикл и т. п.

2) В демоверсии 2009 г. это задание перенесено в часть B.

A7. Определите значение целочисленных переменных **a** и **b** после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический

a = 3 + 8 * 4
b = (a\10) + 14
a = (b MOD 10) + 2
'\ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно
	a := 3 + 8 * 4;
	b := (a div 10) + 14;
	a := (b mod 10) + 2;
	{div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}
		a := 3 + 8 * 4
		b := div (a, 10) + 14
		a := mod (b, 10) + 2
		| div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно |

1) a = 0, b = 18; 2) a = 11, b = 19;
3) a = 10, b = 18; 4) a = 9, b = 17.

Ответ: 4.

Обсуждение. При исполнении этого линейного алгоритма от учащегося не требуется ничего, кроме внимательности. Ошибки, которые ими допускаются, либо носят арифметический характер, либо связаны с тем, что учащиеся перепутали действие операций div и mod.

В а р и а т и в н о с т ь з а д а н и я.

1) Более сложным получается задание, если при его выполнении приходится учитывать приоритет операций.

2) Другой вариант усложнения задания этого типа состоит в том, что в ходе выполнения команд происходит переприсваивание значений переменных. Вот пример такого задания из демоверсии ЕГЭ другого года:

Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический

a = 42
b = 14
a = a\b
b = a*b
a = b\a
\ — *стандартная операция, вычисляющая результат деления нацело первого аргумента на второй*
	a := 42;
	b := 14;
	a := a div b;
	b := a*b;
	a := b div a;
	{div — *стандартная операция, вычисляющая результат деления нацело первого аргумента на второй*}
		a := 42
		b := 14
		a := div(a, b)
		b := a*b
		a := div(b, a)
		| div — *стандартная функция, вычисляющая результат деления нацело первого аргумента на второй* |

1) a = 42, b = 14; 2) a = 1, b = 42;
3) a = 0, b = 588; 4) a = 14, b = 42.

A8. Значения двух массивов A [1...100] и B [1..100] задаются с помощью следующего фрагмента программы:

Бейсик	Паскаль	Алгоритмический
FOR n = 1 TO 100 A (n) = (n – 80) * (n – 80) NEXT n FOR n = 1 TO 100 B (101 – n) = A (n) NEXT n	for n := 1 to 100 do A [n] := (n – 80) * (n – 80); for n := 1 to 100 do B [101 – n] := A [n];	нц для n от 1 до 100 A [n] = (n – 80) * (n – 80) кц нц для n от 1 до 100 B [101 – n] = A [n] кц

Какой элемент массива B будет наибольшим?
1) B [1]; 2) B [21]; 3) B [80]; 4) B [100].
Ответ: 4.

О б с у ж д е н и е. Принципиальным моментом в решении этой задачи является то, что учащемуся не нужно исполнять этот алгоритм — немыслимо вычислить 100 значений для заполнения массива A, а затем переписать эти значения в виде массива B. Учащийся, рассмотрев этот алгоритм, должен понять, что в первом цикле вычисляется набор значений квадратичной функции, а во втором цикле полученные значения выписываются в обратном порядке.

Следовательно, само максимальное значение элементов массива A и массива B будет одним и тем же, но стоять они будут на симметричных (относительно середины массива) местах. Далее выполнение этого задания может идти двумя путями.

1) Найти наибольшее значение квадратичной функции на отрезке [1; 100]. Поскольку функция равна $(x - 80)^2$, т. е. ее график представляет собой параболу, ветви которой направлены вверх, то наибольшее значение достигается на концах отрезка. Ясно, что для данной функции таким концевым значением аргумента является 1. Поэтому в массиве B наибольший элемент будет B [100].

2) Поскольку какой-то из предъявленных четырех вариантов ответа обязательно правильный, достаточно вычислить только четыре элемента B [1], B [21], B [80] и B [100], после чего выбрать из них наибольший. Имеем B [1] = = A [100] = 400, B [21] = A [80] = 0, B [80] = A [21] = 3481 и B [100] = A [1] = 6241. Поэтому в массиве B наибольший элемент будет B [100].

Вариативность задания.

1) Вместо одномерного массива может предлагаться двумерный массив. В этом случае учащиеся, как правило, уже не могут воспользоваться своими математическими представлениями о поведении функции — исследование поведения функции от двух переменных в школьной математике отсутствует. В этой ситуации остается использовать второй из разобранных подходов.

2) Это задание может быть перенесено в часть B. Тогда учащийся должен использовать первый подход.

3) В реальных вариантах ЕГЭ-2008 учащимся предлагалось выбрать одно из описаний того, для какой цели предназначен данный алгоритм. Вот пример такого задания:

Дан фрагмент программы, обрабатывающей двумерный массив A *размером* $n \times n$.

Бейсик	Паскаль	Алгоритмический

k = 1
FOR i = 1 TO n
c = A (i, i)
A (i, i) = A (i, k)
A (i, k) = c
NEXT i
	k := 1;
	for i := 1 to n do
	begin
	c := A [i, i];
	A [i, i] := A [i, k];
	A [i, k] := c
	end
		k := 1
		нц для i от 1 до n
		c := A [i, i]
		A [i, i] := A [i, k]
		A [i, k] := c
		кц

Представим массив в виде квадратной таблицы, в которой для элемента массива A [i, j] *величина* i *является* *номером строки, а величина* j — *номером столбца, в котором расположен элемент. Тогда данный алгоритм меняет местами:*

1) *два столбца в таблице*;
2) *две строки в таблице*;
3) *элементы диагонали и k-й строки таблицы*;
4) *элементы диагонали и k-го столбца таблицы.*

A9. Для какого из указанных значений числа X истинно высказывание

$((X < 5) \to (X < 3)) \wedge ((X < 2) \to (X < 1))$?

1) 1; 2) 2; 3) 3; 4) 4.

Ответ: 2.

Обсуждение. Возможный путь решения этой задачи состоит в нахождении множества всех значений переменной X, при которых данная логическая функция истинна. По нашему мнению, этот путь требует значительного времени и, что не менее важно, чреват промежуточными ошибками. Более эффективным, на наш взгляд, является путь, состоящий в вычислении для каждого предложенного значения X соответствующего значения логической функции. Вот как это выглядит:

$((1 < 5) \to (1 < 3)) \wedge ((1 < 2) \to (1 < 1)) = (И \to И) \wedge (И \to Л) = И \wedge Л = Л$;

$((2 < 5) \to (2 < 3)) \wedge ((2 < 2) \to (2 < 1)) = (И \to И) \wedge (Л \to Л) = И \wedge И = И$;

$((3 < 5) \to (3 < 3)) \wedge ((3 < 2) \to (3 < 1)) = (И \to Л) \wedge (Л \to Л) = Л \wedge И = Л$;

$((4 < 5) \to (4 < 3)) \wedge ((4 < 2) \to (4 < 1)) = (И \to Л) \wedge (Л \to Л) = Л \wedge И = Л$.

Следовательно, верным является второй ответ.

Отметим, что два последних вычисления можно было бы не проводить, поскольку известно, что правильный ответ ровно один. Однако для этого надо быть уверенным, что в ходе вычислений не была допущена ошибка. Можно сказать, что два последних вычисления осуществляют функцию косвенного контроля правильности полученного ответа — ведь если где-нибудь еще раз получилось значение Истина, то, значит, надо провести перепроверку.

Вариативность задания.

1) Можно ожидать, что некая модификация этого задания будет перенесена в часть B. Вот возможный вариант:

Сколько существует натуральных чисел X, *для которых истинно высказывание*

$(X < 10) \wedge ((X < 5) \to (X < 3))$?

2) В демоверсии 2009 г. в части B помещена следующая вариация этого задания:

Каково наибольшее целое число X, при котором истинно высказывание

$(50 < X \cdot X) \to (50 > (X + 1)(X + 1))$?

В этой ситуации уже нельзя выполнить задание простым перебором возможных вариантов, здесь необходимо проанализировать логическое условие, затем решить получающуюся систему неравенств и среди решений выбрать наибольшее целое число. Сначала отметим, что импликация истинна, если посылка ложна. Значит, при X, принимающем значения от –7 до 7, предъявленное высказывание будет истинным. Следовательно, ответ в этом задании не меньше 7. Если же X больше 7, то посылка в импликации будет истинной, а тогда для истинности всей импликации должно выполняться неравенство $50 > (X + 1)(X + 1)$. Однако при $X > 7$ это невозможно. Ответ: X = 7.

A10. Укажите, какое логическое выражение равносильно выражению $\neg(A \lor \neg B \lor C)$.

1) $\neg A \lor B \lor \neg C$; 2) $A \land \neg B \land C$;
3) $\neg A \lor \neg B \lor \neg C$; 4) $\neg A \land B \land \neg C$.

Ответ: 4.

Обсуждение. Стандартный путь решения таких задач — составление таблицы истинности для каждого из выражений и проверка на совпадение. Несмотря на кажущуюся трудоемкость, многие учащиеся выполняют это задание именно этим способом (спасает обычно то, что расхождение в таблицах истинности наблюдается не позже третьей строки, а верный вариант ответа редко бывает четвертым, как в данном случае). Но конкретно в этом задании проще всего воспользоваться законом де Моргана и правилом двойного отрицания.

A11. Символом F обозначено одно из указанных ниже логических выражений от трех аргументов X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
1	1	1	1
1	1	0	1
1	0	1	1

Какое выражение соответствует F?

1) $X \lor \neg Y \lor Z$; 2) $X \land Y \land Z$;
3) $X \land Y \land \neg Z$; 4) $\neg X \lor Y \lor \neg Z$.

Ответ: 1.

Обсуждение. Стандартный путь решения таких задач — составление таблицы истинности для каждого из предложенных вариантов функции F при тех значениях переменных, которые присутствуют в таблице. Ясно, что если при заполнении таблицы на каком-то наборе переменных значение не совпало со значением функции F, то дальше можно не продолжать. В частности, поэтому можно избрать несколько иной путь: для одного набора переменных вычислить значения всех тех функций, которые указаны в качестве вариантов ответа, — часть из них почти наверняка можно будет тут же отбросить, потом также протестировать оставшиеся на следующем наборе и т. д., пока не останется одна функция.

A12. Грунтовая дорога проходит последовательно через населенные пункты A, B, C и D. При этом длина дороги между A и B равна 80 км, между B и C — 50 км и между C и D — 10 км. Между A и C построили новое асфальтовое шоссе длиной 40 км. Оцените минимально возможное время движения велосипедиста из пункта A в пункт B, если его скорость по грунтовой дороге 20 км/ч, по шоссе 40 км/ч?

1) 1 час; 2) 1,5 часа; 3) 3,5 часа; 4) 4 часа.

Ответ: 3.

Обсуждение. По замыслу составителей ЕГЭ для выполнения этого задания учащиеся должны нарисовать нагруженный граф, у которого на ребрах проставлено время движения по данному ребру, и затем найти кратчайший путь на графе. Правда, пункт D здесь оказывается совсем ни при чем, а из A в B есть ровно два варианта, которыми можно добраться.

Вариативность задания.

1) В реальных вариантах ЕГЭ обычно используется граф с большим количеством ребер и вершин. В этом случае учащиеся должны уметь осуществить разумный перебор возникающих вариантов путей. Наиболее подготовленные учащиеся могут, конечно, применить какой-либо известный алгоритм поиска кратчайшего пути в нагруженном графе.

2) Надо ожидать, что в будущем в этом месте будут представлены задачи переборного типа, необязательно связанные с графами. Уже в реальной версии 2008 г. была предложена следующая задача:

Между четырьмя крупными аэропортами, обозначенными кодами ALK, DOR, LTA *и* MIV, *ежедневно выполняются авиарейсы. Приведен фрагмент расписания перелетов между этими аэропортами*:

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
MIV	DOR	05:40	08:00
ALK	DOR	07:10	10:05
LTA	ALK	09:10	11:30
LTA	MIV	09:35	12:30
DOR	MIV	10:00	12:20
DOR	ALK	10:15	12:45
MIV	ALK	11:20	16:15
ALK	LTA	12:20	15:15
ALK	MIV	14:40	19:40
MIV	LTA	14:50	16:55

Путешественник находится в аэропорту ALK *в полночь* (0:00). *Определите самое раннее время, когда он может оказаться в аэропорту* MIV.

1) 12:20; 2) 12:30; 3) 16:15; 4) 19:40.

Следует отметить не вполне однозначную трактовку вопроса в этом задании: неясно, имеется в виду самое раннее время суток или продолжительность путешествия. Отметим, что на этой неоднозначности построены неверные варианты ответов.

A13. Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11 соответственно). Если таким способом закодировать последовательность символов ГБАВ и записать результат шестнадцатеричным кодом, то получится:

1) D2; 2) 132; 3) 3102; 4) DBAC.

Ответ: 1.

Обсуждение. Официальное решение предполагает, что учащиеся перекодируют последовательность букв в двоичную последовательность и получившееся двоичное число переведут в шестнадцатеричный код. Однако легко понять, что в получающейся двоичной последовательности будет всего лишь 8 символов. Следовательно, в шестнадцатеричном коде может присутствовать лишь два символа. После этого выбор правильного варианта ответа становится очевидным.

Вариативность задания. В реальных вариантах ЕГЭ как этого, так и прошлых лет встречалось задание, связанное с декодированием последовательности. Вот пример такого задания:

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв — из двух бит, для некоторых — из трех). Эти коды представлены в таблице:

m	n	o	p	r
000	11	01	001	10

Определите, какой набор букв закодирован двоичной строкой 01100110001001.

1) orpmro; 2) ororpp; 3) orprpp; 4) orormro.

A14. В формировании цепочки из четырех бусин используются некоторые правила. В конце цепочки стоит одна из бусин P, N, T, O. На первом месте — одна из бусин P, R, T, O, которой нет на третьем месте. На третьем месте — одна из бусин O, P, T, не стоящая в цепочке последней. Какая из перечисленных цепочек могла быть создана с учетом этих правил?

1) PORT; 2) TTTO; 3) TTOO; 4) OOPO.

Ответ: 4.

Обсуждение. Составителями ЕГЭ это задание по кодификатору отнесено к теме «Формальное исполнение алгоритма, записанного на естественном языке». Фактически же никакого алгоритма, т. е. последовательности действий, приводящей к заданному результату, здесь нет. С некоторой натяжкой это задание можно было бы отнести к логическому программированию, но надо заметить, что таковое в стандартном школьном курсе информатики не изучается. На самом деле здесь просто требуется проверить, какая из предъявленных последовательностей удовлетворяет всем условиям, сформулированным в задании. Первому правилу удовлетворяют все цепочки, т. е. оно ничего не отвергает. Второе правило фактически дает одно содержательное ограничение: буквы, стоящие на первом и третьем местах, обязательно различны. Это исключает второй вариант ответа. Наконец, третье правило означает, что не может в конце оказаться двух одинаковых букв, что отвергает ответ 3, и буквы R на третьем месте, что отвергает вариант 1.

Составители явно рассчитывают на сформированный у учащихся стереотип проверять условия в том порядке, в каком они предъявлены. Именно поэтому первое правило ничего не отвергает, второе отвергает только один вариант, а самое строгое правило помещено последним. Зная эти приемчики составителей, можно ориентировать учащихся на проверку правил, начиная с последнего, — это сэкономит время.

Другой вариант выполнения этого задания состоит в проверке каждой цепочки, удовлетворяет ли она сформулированным условиям. Но и в этом случае отчетливо видно, что никакого исполнения какого бы то ни было алгоритма не осуществлялось.

Вариативность задания.

1) В заданиях этого типа усложнение происходит за счет увеличения числа логических условий или формы их высказываний. К примеру, в данном варианте второе и третье правила высказаны в виде конъюнкции двух условий. Можно ожидать применения и более сложных логических конструкций.

2) Иногда меняется фабула задания. Это может также вызвать определенное замешательство, прежде чем учащиеся поймут, что это «та же Дуня, только в другом сарафане». Вот пример такого «переодевания».

Дешифровщику необходимо восстановить поврежденный фрагмент сообщения, состоящий из четырех символов. Имеется достоверная информация, что использовано не более пяти букв (А, О, Б, В, Г), *причем на втором месте стоит один из символов* А, Г, О, В, *на первом — одна из букв* Б, В, Г, О, *которой нет на третьем месте, на третьем месте — одна из букв* О, А, Б, В, *не стоящая в слове на втором месте, на четвертом месте — одна из букв* А, Б, В, Г, *которой не было на первом месте. Появилась дополнительная информация, что возможен один из четырех вариантов. Какой?*

1) ВВАА; 2) БОБА; 3) ОВВА; 4) ГОАГ.

А15. Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

символ «?» (вопросительный знак) означает ровно один произвольный символ;

символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, какое из указанных имен файлов удовлетворяет маске

?a???*

1) dad1; 2) dad22; 3) 3daddy; 4) add444.

Ответ: 2.

Обсуждение. Основные ошибки, допускаемые учащимися, состоят в том, что они предполагают возможность пустой последовательности символов вместо знака «?» и, наоборот, считают, что вместо знака «*» обязательно должен стоять хоть какой-нибудь символ.

Вариативность задания. В реальном варианте предлагалось полное имя файла. Вот соответствующий пример.

*Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:*

символ «?» (вопросительный знак) означает ровно один произвольный символ;

символ «» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.*

Определите, какое из указанных имен файлов удовлетворяет маске **ban?*.?xt**:

1) ban.txt; 2) banan.xt; 3) bank.txt; 4) bank.xt.

Казалось бы, это непринципиальное изменение задания потребовало от учащихся больше времени на его выполнение. Причина — наличие отвлекающей внимание точки.

A16. Из правил соревнования по тяжелой атлетике.

Тяжелая атлетика — это прямое соревнование, когда каждый атлет имеет три попытки в рывке и три попытки в толчке. Самый тяжелый вес поднятой штанги в каждом упражнении суммируется в общем зачете. Если спортсмен потерпел неудачу во всех трех попытках в рывке, он может продолжить соревнование в толчке, но уже не сможет занять какое-либо место по сумме двух упражнений.

Если два спортсмена заканчивают состязание с одинаковым итоговым результатом, высшее место присуждается спортсмену с меньшим весом. Если же вес спортсменов одинаков, преимущество отдается тому, кто первым поднял победный вес.

Таблица результатов соревнований по тяжелой атлетике:

Ф.И.О.	Вес спортсмена	Взято в рывке	Рывок с попытки	Взято в толчке	Толчок с попытки
Айвазян Г. С.	77,1	150,0	3	200,0	2
Викторов М. П.	79,1	147,5	1	202,5	1
Гордезиани Б. Ш.	78,2	147,5	2	200,0	1
Михальчук М. С.	78,2	147,5	2	202,5	3
Пай С. В.	79,5	150,0	1	200,0	1
Шапсугов М. Х.	77,1	147,5	1	200,0	1

Кто победил в общем зачете (сумме двух упражнений)?

1) Айвазян Г. С.; 2) Викторов М. П.;
3) Михальчук М. С.; 4) Пай С. В.

Ответ: 1.

Обсуждение. Составителями ЕГЭ предполагалось, что учащиеся должны составить по словесному описанию условий селективную формулу и с ее помощью получить ответ. Этот путь позволяет избежать ошибок, которые могут возникать из-за трудностей удерживать в поле зрения сразу несколько условий, причем связанных операциями конъюнкции или дизъюнкции. Нередко ими учитывается значение только одного операнда.

Вариативность задания. 1) В реальных вариантах ЕГЭ вместо словесного описания предлагалась логическая формула запроса к базе данных. Вот пример такого задания.

Результаты тестирования представлены в таблице:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в ней удовлетворяет условию
«Пол = ‘ж’ И Химия > Биология»?
1) 5; 2) 2; 3) 3; 4) 4.

2) Другим направлением варьирования данного задания можно ожидать увеличение числа условий и создание более сложной формулы отбора записей. Именно эта тенденция прослеживается в последние годы.

A17. Для хранения растрового изображения размером 32×32 пикселя отвели 512 байт памяти. Каково максимально возможное число цветов в палитре изображения?
1) 256; 2) 2; 3) 16; 4) 4.
Ответ: 3.

Обсуждение. Это достаточно стандартная задача. Основные ошибки возникают из-за невнимательности к единицам измерения, а также не вполне отработанным навыкам действий со степенями числа 2.

Вариативность задания. 1) Нередко предлагается какая-либо обратная задача, например: задано количество цветов и размер растра, требуется определить объем памяти.

2) Можно ожидать, что подобное задание будет перенесено в часть В.

3) В реальном ЕГЭ-2008 под этим номером было задание, связанное с RGB-кодированием. Оно подробно разобрано в данной книге на с. 89 (см. задание 20).

A18. Дан фрагмент электронной таблицы:

	A	B	C
1	10	20	= A1 + B$1
2	30	40	

Чему станет равным значение ячейки C2, если в нее скопировать формулу из ячейки C1?
Знак $ обозначает абсолютную адресацию.
1) 40; 2) 50; 3) 60; 4) 70.
Ответ: 2.

Обсуждение. Ответ 70 получается, если не учтен знак абсолютной адресации. Остальные ответы малоправдоподобны. При подготовке тренировочных вариантов лучше формировать варианты ответов, подбирая различные комбинации абсолютной и относительной адресации, которые могут возникать из-за непонимания различий этих операций.

Вариативность задания. 1) Задание может усложняться за счет копирования как в строки, так и в столбцы (пример такого задания приведен под номером А14 на с. 323 учебника).

2) В реальных вариантах 2008 г. это задание, наоборот, было упрощено.

Дан фрагмент электронной таблицы:

	A	B
1	0,5	1,5
2	9,5	= A1 + A2/5

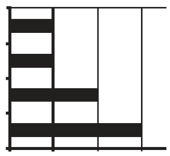
Чему станет равным значение ячейки B2, если значение ячейки A1 увеличить на 4, а значение ячейки A2 уменьшить на 2?
1) 3,5; 2) 2; 3) 8; 4) 6.

A19. Дан фрагмент электронной таблицы:

	A	B	C	D
1		3	4	
2	= C1 – B1	= B1 – A2 * 2	= C1/2	= B1 + B2

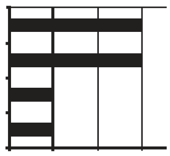
После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A2 : D2. Укажите получившуюся диаграмму.

1)

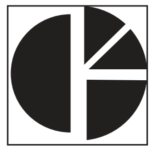


2)

3)



4)



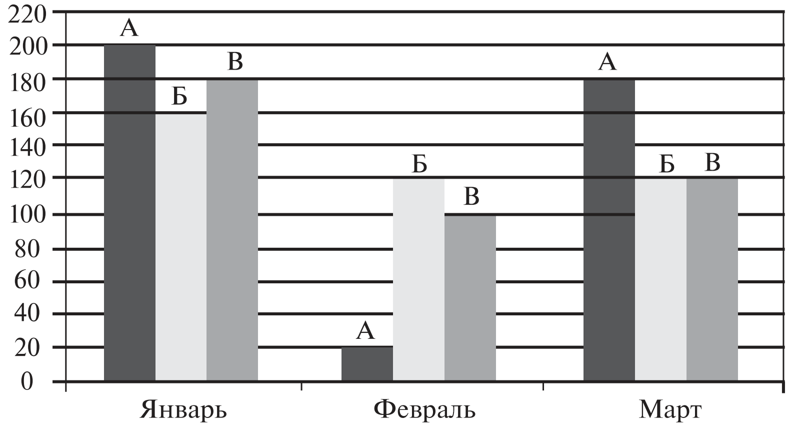
Ответ: 4.

<u>Обсуждение.</u> К сожалению, учащиеся допускают ошибки, связанные с определением порядка действий (возможно, они воспринимают электронную таблицу как обычный калькулятор; именно для предупреждения подобных ошибок мы в нашем практикуме в основном предлагали использовать *Инженерный калькулятор*). Конечно, вносят свою лепту и ошибки невнимательности.

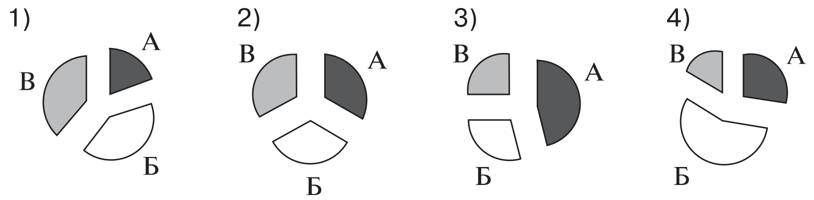
<u>Вариативность задания.</u>

В реальном ЕГЭ-2008 под этим номером было задание, связанное с преобразованием диаграмм из одного вида в другой. При этом учащиеся должны осуществить двойное преобразование информации: сначала из графической формы в числовую, а затем снова в графическую. Фактически здесь осуществлялась проверка тех элементов информационной культуры, о которых шла речь в главе 1 нашего учебника.

На диаграмме показаны объемы выпуска продукции трех видов: А, Б и В — за каждый из месяцев первого квартала.



Какая из диаграмм правильно отражает соотношение объемов выпуска этих видов продукции за весь квартал?



A20. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	**вниз**	**влево**	**вправо**

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

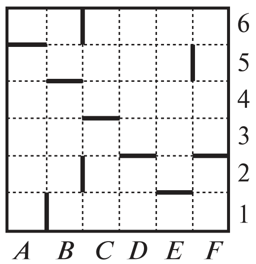
Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл ПОКА < *условие* > *команда* выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ остановится в той же клетке, с которой он начал движение?

НАЧАЛО
ПОКА < **справа свободно** > **вправо**
ПОКА < **сверху свободно** > **вверх**
ПОКА < **слева свободно** > **влево**
ПОКА < **снизу свободно** > **вниз**
КОНЕЦ
1) 1; 2) 0; 3) 3; 4) 4.
Ответ: 4.



Обсуждение. При выполнении этого задания учащиеся должны заметить, что контур обхода всегда представляет собой прямоугольник, и учесть направление обхода. При этом рисовать сам контур крайне вредно — он будет мешать проверке на допустимость в следующей клетке. Разумеется, важную роль играет культура организации перебора вариантов.

Результаты выполнения этого задания на реальном ЕГЭ колеблются в разных вариантах от 16 до 27%. Это намного ниже того, что прогнозировалось изначально разработчиками варианта. Причина низкого выполнения этого задания кроется, на наш взгляд, не в том, что учащиеся не умеют выполнять подобные задания (тем более что аналогичное задание было предъявлено в демоверсии), а в том, что требовалось до двадцати раз проделать довольно трудоемкую процедуру (требующую на свое исполнение от 20 до 30 секунд и очень большого внимания). Кроме того, каждый раз процедура выполнялась на одном и том же чертеже, где было уже зафиксировано и предыдущее ее исполнение, что, несомненно, сказывалось отрицательно. В условиях дефицита времени, выделяемого на выполнение заданий групп A и B, это неизбежно влекло ошибки. Отметим, что для этого задания количество учащихся, отказавшихся от выбора ответа, наибольшее среди всех заданий группы A. Мы предполагаем, что реально учащиеся на каком-то шаге отказывались от полного выполнения этого задания и пытались частично угадать ответ. Легко подсчитать, что вероятность угадывания правильного ответа как раз и составляет 25%.

Вариативность задания. Фактически отсутствует.

ОГЛАВЛЕНИЕ