

информатика

И. СЕМАКИН, Т. ШЕИНА

Преподавание базового курса информатики в средней школе

методическое
пособие



Б-И-Н-О-М

И Н Ф О Р М А Т И К А

И. СЕМАКИН, Т. ШЕИНА

Преподавание базового курса информатики в средней школе

**методическое
пособие**



Москва
БИНОМ. Лаборатория знаний
2004

УДК 004.9
ББК 32.97
С30

Семакин И. Г.

С30 **Преподавание базового курса информатики в средней школе: Методическое пособие / И. Г. Семакин, Т. Ю. Шеина. — 2-е изд., испр. и доп. — М.: БИНОМ. Лаборатория знаний, 2004. — 540 с.: ил.**

ISBN 5-94774-093-1

Методическое пособие к учебнику И. Г. Семакина и др. «Информатика. Базовый курс» завершает выпуск комплекта (учебник — задачник — методическое пособие) для преподавания курса информатики в 7–9 классах.

Раскрывается концептуальное содержание базового курса, показывается целостность курса, логичность его структуры. Большое внимание уделяется истории школьной информатики, эволюции концепций предмета и общим методическим рекомендациям. Поурочный учебный план, система проверочных тестов, рекомендации по решению задач, дополнительный учебный материал и многое другое делают настоящее пособие хорошим подспорьем для преподавателя.

Для учителей и методистов, использующих учебник авторов, а также для тех, кто захочет разработать свой собственный курс.

УДК 004.9
ББК 32.97

**По вопросам приобретения обращаться:
в Москве**

**«БИНОМ. Лаборатория знаний» (095) 955-03-98, e-mail: lbz@aha.ru
в Санкт-Петербурге**

«Диалект» (812) 247-93-01, e-mail: dialect@sndlet.ioffe.rssi.ru

ISBN 5-94774-093-1

© Семакин И. Г.,
Шеина Т. Ю., 2002
© БИНОМ. Лаборатория
знаний, 2004

Оглавление

Предисловие	5
Часть 1. Введение	12
1. Краткий очерк истории школьной информатики.	12
2. Современная концепция школьной информатики.	15
2.1. Содержание предметной области	15
2.2. Содержание образовательной области информатики	16
3. Авторская концепция содержания базового курса	19
4. Общие методические рекомендации	23
5. Тематическое планирование занятий.	24
Часть 2. Методические рекомендации по изучению разделов курса	29
1. Введение в предмет.	29
2. Информация и знания	37
3. Как измерить информацию (алфавитный подход)	44
4. Содержательный подход к измерению информации.	53
5. История информатики	60
6. Системы счисления.	65
7. Архитектура ЭВМ	80
8. Программное обеспечение ЭВМ.	100
9. Текстовая информация и компьютер	114
10. Графическая информация и компьютер	129
11. Передача информации в компьютерных сетях	141
12. Введение в информационное моделирование	158
13. Базы данных	171
14. Табличные вычисления на компьютере	196
15. Информация и управление.	220

16. Как работает процессор ЭВМ	246
17. Искусственный интеллект и базы знаний	269
18. Введение в программирование	286
Часть 3. Дидактические материалы	
к организации учебных занятий.	334
1. Два года обучения — 68 часов;	
План уроков	336
2. Два года обучения — 136 часов;	
План уроков	341
3. Три года обучения — 102 часа	
3.1. VII класс	351
Домашние задания	367
Практические задания	
для выполнения на компьютере	377
Тесты	387
План уроков. 8 класс.	399
Домашние задания	413
Практические задания	
для выполнения на компьютере	425
Тесты	436
План уроков. 9 класс.	454
Домашние задания	467
Практические задания	
для выполнения на компьютере	476
Тесты	483
Часть 4. Итоговая система знаний и умений	
учащихся по базовому курсу информатики	503
1. Основной уровень знаний и умений	
учащихся по итогам освоения базового курса	
информатики	504
2. Дополнительные знания и умения	
учащихся соответствующие углубленному уровню	
освоения базового курса информатики	510
3. Базовый курс в вопросах и ответах	514
Литература	538

Предисловие

Книги, составляющие учебно-методический комплект (УМК) по базовому курсу информатики, выпускаются в издательстве «Лаборатория базовых знаний», начиная с 1998 года. Последовательно были изданы учебник для 7—9 классов [1], задачник-практикум по информатике (в 2 частях) [2], методическое пособие для учителя [3] и структурированный конспект базового курса [4] — приложение к учебнику. В 2003 году авторами подготовлено обновленное издание учебника для 7—9 классов [5]. Настоящая книга представляет собой новое издание методического пособия для учителя, в котором учтены изменения, внесенные в учебник.

В основу содержания первого издания учебника по базовому курсу информатики были заложены следующие принципы.

- Учебник ориентирован на массовую школу и может быть использован при любом уровне обеспеченности школы компьютерной техникой.
- Содержание учебника соответствует требованиям проекта образовательного стандарта по информатике и Обязательного минимума содержания образования по информатике.
- Содержание учебника (как и всего УМК) обеспечивает возможность многоуровневого обучения информатике в школе.
- Учебник включает в себя теоретическое содержание предмета (с минимально необходимым набором контрольных вопросов и заданий). Материал для практической работы помещается в задачник-практикум.
- В содержании учебника акцент делается на фундаментальное содержание учебного предмета. Центральным системообразующим понятием выступает понятие информации.

- Содержание учебника инвариантно к моделям ЭВМ и версиям программ.

Выполнение требования многоуровневости преподавания информатики в ее теоретической составляющей обеспечивается двухчастной структурой учебника: **ЧАСТЬ 1** — основное содержание базового курса, **ЧАСТЬ 2** — дополнительный материал для углубленного изучения. Многоуровневость практической составляющей обеспечивается содержанием и структурой задачника-практикума. Школьный опыт использования учебника доказал удачность такого решения (по многочисленным высказываниям учителей и методистов). Кроме того, двухчастная структура существенно облегчила работу авторов при подготовке второго издания. Ориентация на многоуровневость нашла подтверждение и в нормативных документах по информатике: Обязательный минимум от 1999 года (уровни А и Б).

Пятилетний опыт использования учебника и всего УМК в школах РФ доказал правильность первоначально заложенных принципов. Комплект получил широкое распространение в массовой школе.

Работая над вторым изданием учебника, авторы сохранили в действии все вышеперечисленные принципы. Внося изменения в содержание учебника, авторы исходили из следующих положений:

- в своей основе учебник остается прежним; модернизация не обесценивает книг первого издания с учетом того, что значительная их часть находится в школьных библиотеках; учителя, получив информацию из публикации в периодике о внесенных изменениях (газета «Информатика». 2003. № 18), смогут, с учетом этого, использовать старые книги;
- обновление производится, прежде всего, в направлении актуализации некоторых тем, необходимость которой назрела за прошедшие 5 лет;

- при обновлении учитывается массовый опыт использования учебника и возникавшие в этом процессе замечания учителей и методистов, высказываемые в периодике, на конференциях и в личных контактах с авторами. В частности, с этим связана реструктуризация первой главы, уточнение отдельных определений;
- исходя из опыта преподавания и с учетом новых требований происходит перемещение некоторых материалов из первой части во вторую и наоборот;
- углубление теоретического содержания курса происходит, в основном, путем дополнения содержания второй части учебника.

В средней школе предмет «Информатика» — сравнительно молодой. Как обязательная дисциплина информатика в программу школы была введена в 1985 году под названием «Основы информатики и вычислительной техники» (ОИВТ). Все недолгие годы своего существования этот предмет оставался, наверное, самым спорным в школьной программе. Менялось название предмета, его место в учебном плане, содержательная концепция, методика преподавания, учебно-методическое и техническое обеспечение. Многочисленные реформаторские процессы, происходившие и продолжающиеся в отечественной системе образования, самым ощутимым образом сказывались на информатике. Время от времени ставится вопрос: «А нужен ли этот предмет вообще в школе? А не слить ли его с другими предметами, например, математикой или технологией?». В федеральном базисном учебном плане 1998 года информатика объединена в одну образовательную область с математикой.

Наверное, ни на одном другом школьном предмете так не отражается региональная специфика, как на информатике. В разных регионах страны различается не только уровень технического обеспечения школ (оснащение компьютерами), но часто и место в региональ-

ном базисном учебном плане, превалирующая содержательная концепция, используемое учебно-методическое обеспечение.

Настоящее пособие посвящено версии базового курса информатики, которая первоначально также носила региональный характер. В 1994 году авторским коллективом, в состав которого входили преподаватели Пермского государственного университета и педагогического института, было подготовлено пособие для учителя «Основы информатики и вычислительной техники в базовой школе» [6]. Некоторые материалы этого пособия публиковались в журнале «Информатика и образование» в серии статей под общим названием «Базовый курс ОИВТ: Пермская версия» [7]. Одновременно с пособием для учителя были изданы рабочие тетради для учащихся с кратким изложением содержания курса и материалами для практических работ. Эти учебные издания позволили уже в 1994/1995 учебном году большинству пермских школ перевести преподавание информатики из старших классов в 8—9 классы.

С выходом в свет учебника [1], которому был присвоен гриф Министерства общего и профессионального образования РФ, «Пермская версия» базового курса информатики приобрела федеральный статус. Сотни школ Пермской области сыграли роль экспериментальных площадок, на которых испытывался и отлаживался базовый курс. Авторы бесконечно благодарны школьным учителям, сотрудничавшим с ними и тем самым непосредственно участвовавшим в отработке курса. Это сотрудничество продолжается.

Согласно Закону об образовании в настоящее время в системе общего среднего образования России содержание обучения определяется концепцией образовательных стандартов. На момент написания этой книги стандарт содержания информатики на законодательном уровне не утвержден. Содержание предмета регламентируется временным документом под названием «Обязательный минимум содержания среднего (полно-

го) общего образования, раздел «Информатика» [8]. «Пермская версия» базового курса полностью обеспечивает обучение в соответствии с минимумом и, в то же время, дает возможность выйти за рамки минимального содержания на более высокий уровень.

Придерживаясь рамок образовательного стандарта, регламентирующего главным образом результаты обучения в виде перечня знаний и умений учеников, учитель имеет право выбора программы обучения, учебника и других средств обеспечения учебного процесса. Информатика — один из первых предметов в школе, в котором появилась возможность выбора средств обучения. Еще в конце 80-х, начале 90-х годов существовали три альтернативных учебника (и соответствующие программы обучения), среди которых учитель мог делать выбор. Это были учебники, составленные авторскими коллективами под руководством А. Г. Кушниренко, А. Г. Гейна и В. А. Каймина. Эти учебники можно отнести ко второму поколению школьных учебников по информатике (первое поколение — учебник под редакцией А. П. Ершова и В. М. Монахова). Учебник [1] и его второе издание [5] следует отнести к третьему поколению учебной литературы по информатике. От предыдущих данное поколение учебников отличается сменой как возрастной ориентации (основная школа, в отличие от старших классов), так и содержательной концепции. Уже сегодня число изданных учебных пособий этого периода превышает число пособий предыдущего периода. Вполне возможно, что их станет еще больше. В связи с этим усложняется задача выбора для учителя.

Понятно, что каждый автор учебника заинтересован привлечь на свою сторону как можно больше учителей. Для того чтобы заинтересовать учителя, его нужно сделать своим единомышленником. Учитель должен сознательно принять авторскую концепцию курса. В этом авторы настоящего пособия видят свою первую цель: раскрыть концептуальное содержание ба-

зового курса, показать целостность курса, логичность его структуры. С точки зрения авторов, учитель должен понимать современную ситуацию в школьной информатике и за «деревьями» сиюминутных проблем видеть весь «лес». Такое понимание позволит ему сделать сознательный выбор. Помочь в этом учителю предназначена вводная глава пособия. В ней отражается ряд концептуальных вопросов, таких как история школьной информатики, эволюция концепций предмета, авторская концепция курса, общие методические рекомендации.

Вторая цель пособия — представить понятийную систему курса и описать методику организации занятий. В первой части приводится тематическое планирование занятий. Однако жестко привязывать материал пособия к этому плану не имеет смысла. В большинстве случаев учителя не смогут строго следовать ему по многим причинам. Поэтому методический материал в пособии структурирован в соответствии с главами и параграфами учебника, а разделение его по урокам — задача учителя.

Вторая часть пособия называется «Методические рекомендации по изучению разделов курса». Эта часть делится на тематические разделы, соответствующие таковым в учебнике. В разделах присутствуют следующие рубрики: основные педагогические цели раздела, перечень изучаемых вопросов, методические рекомендации по изложению теоретического материала, методические рекомендации по решению задач, методические рекомендации по организации практической работы учащихся. Кроме того, в круг задач пособия входит предоставление учителю дополнительного учебного материала, выходящего за рамки учебника. Такой материал присутствует в некоторых разделах второй части. Его использование позволит учителю сделать объяснения на уроках более разнообразными и интересными.

Особенностью учебника [5] является его двухуровневая структура. Первая часть учебника содержит основной материал, обеспечивающий обязательный минимум знаний учащихся. Во второй части расположен дополнительный материал, ориентированный на углубленный уровень изучения базового курса. В ряде разделов второй части методического пособия даются рекомендации по организации изучения курса на разных уровнях.

Третья часть пособия называется «Дидактические материалы к организации учебных занятий». Здесь предлагается три варианта поурочного планирования занятий, рассчитанные соответственно на 68, 102 и 136 учебных часов. Для варианта 102 часа (7—9 классы по 1 уроку в неделю) дана подробная поурочная разработка занятий. Весь материал разделен на содержательные модули и включает в себя:

- поурочный учебный план;
- систему домашних заданий;
- систему заданий для индивидуальной практической работы школьников за компьютерами;
- систему проверочных тестов.

В отличие от первого издания пособия, здесь не приводится система побалльной оценки работы учащихся и описание применения модульно-рейтинговой технологии. При желании учитель самостоятельно может внедрить такую технологию или обратиться к первому изданию методического пособия.

В четвертой части методического пособия дается описание итоговой системы знаний и умений учащихся по базовому курсу информатики, разделенной на два уровня: основной и дополнительный. В этом же разделе представлена система основных понятий курса в форме вопросов и ответов.

Введение

1. Краткий очерк истории школьной информатики

В 1985 году в программу средней общеобразовательной школы СССР был введен новый предмет «Основы информатики и вычислительной техники» — ОИВТ. В этом же году был издан первый школьный учебник ОИВТ под редакцией А. П. Ершова и В. М. Монахова [9]; на базе вузов областных центров страны была проведена курсовая подготовка первых учителей информатики. Согласно базисному учебному плану преподавание информатики велось в 9—10 классах (позднее — в 10—11 классах). Поначалу в большинстве школ информатика преподавалась по безмашинному варианту, поскольку лишь немногие школы могли в ту пору обеспечить своим ученикам доступ к ЭВМ. Счастливыми, имевшими возможность потрогать компьютер своими руками, были ученики некоторых городских школ, попавших под опеку вычислительных центров предприятий и вузов. Компьютер в школе был большой редкостью.

Оказавшись перед фактом всеобщего охвата школ информатикой, правительство страны, Госкомитет СССР по образованию срочно принимают меры по обеспечению преподавания нового предмета. На конкурсной основе разрабатываются школьные компьютеры и программное обеспечение для них, закупается некоторое количество зарубежной техники («Ямаха-MSX» в Японии, «Роботрон» в ГДР). К концу 80-х годов налаживается массовый выпуск отечественных школьных микро-ЭВМ «Корвет» и «Электроника-УКНЦ». На базе отечественного бытового персонального компьютера «Электроника БК-0010» создаются учебные комплекты КУВТ-86. Педагогические вузы начинают подготовку выпускников по специально-

сти «Учитель ОИВТ». В 1986 году начинается выпуск методического журнала «Информатика и образование».

К концу 80-х годов возрастает потребность школ в учебниках и учебных программах по информатике, ориентированных на использование ЭВМ. В результате проведенного в 1987 году конкурса для преподавания информатики в школе были рекомендованы учебники ОИВТ, написанные авторскими коллективами под руководством А. Г. Кушниренко [10] и В. А. Каймина [11]. Немного позднее школам был рекомендован третий учебник, написанный А. Г. Гейном и др. [12]. Программой Госкомитета СССР по образованию от 1991 года [13] было утверждено «троевластие» в школьной информатике. Учитель имел право выбрать любой из трех учебников по своему усмотрению.

Новый этап истории школьной информатики начинается с 1993 года. В этом году был принят новый базисный учебный план для школ Российской Федерации [14], согласно которому преподавание информатики было перенесено в 7 класс (68 часов). Фактически с этого года предмет сменил свое название с «ОИВТ» на «Информатика». Под этим названием он стоит в базисном учебном плане. С этого же времени усиливаются региональные различия в организации преподавания школьной информатики. Например, в Пермской области (представителями которой являются авторы пособия) преподавание информатики переносится в 8—9 классы с той целью, чтобы в старших классах оно при желании могло бы быть продолжено. В школах некоторых регионов информатика так и оставалась в старших классах.

В 1996 году выходит новая редакция Закона РФ «Об образовании», согласно которому содержание образования по всем школьным дисциплинам должно определяться образовательными стандартами. Тем самым отменялись государственные программы по предметам, учителям предоставлялась большая свобода в выборе средств обучения, в организации учебного процесса. Был проведен конкурс проектов образователь-

ных стандартов по всем школьным предметам, в том числе и по информатике. Победил в конкурсе проект, разработанный чл.-корр. РАО, профессором А. А. Кузнецовым [15].

В 1995 году Министерством образования РФ был выпущен документ под названием «Основные компоненты содержания информатики в общеобразовательных учреждениях» [16]. В нем были выделены три этапа овладения основами информатики и формирования информационной культуры в процессе обучения в школе: *пропедевтический* (1–6 классы), *базовый* (7–9 классы) и *профильный* (10–11 классы). Обязательный общеобразовательный минимум подготовки школьников по информатике обеспечивает базовый курс. Именно на него должен быть ориентирован образовательный стандарт.

На момент написания данного пособия образовательные стандарты по дисциплинам не приняты на законодательном уровне. Документом, регламентирующим содержание информатики, является «Обязательный минимум содержания среднего (полного) общего образования, раздел «Информатика» (последняя редакция — в приложении к приказу Минобразования РФ от 30.06.99, № 56). Федеральный базисный учебный план 1998 года снова возвращает обязательную часть курса информатики в старшие классы (10–11).

В наше время компьютер прочно занял место в средней школе. Применяется он не только на уроках информатики, но и при обучении другим предметам, в школьном делопроизводстве, в творческой, исследовательской работе учащихся, для отдыха и развлечений. Происходит массовый переход на IBM-PC-совместимые компьютеры. В образовательную область «Технология» включен раздел «Информационные технологии», под который выделяется учебное время. Содержание предмета «Информатика» меняется в сторону усиления его общеобразовательного веса. Выпускается большое количество учебной литературы по информатике, в потоке которой учителю порой бывает нелегко сориентироваться.

2. Современная концепция школьной информатики

2.1. Содержание предметной области

Содержание любого школьного предмета является отражением на систему образования определенной *предметной области*. Под предметной областью понимается совокупность области знаний и области практической деятельности, базирующейся на этих знаниях. Естественно, что школьный предмет будет иметь статус обязательной общеобразовательной дисциплины только в том случае, если соответствующая предметная область имеет существенное значение для жизни общества. Что же представляет собой предметная область информатики и каково ее значение для современного общества?

Сам термин «информатика» появился в середине 60-х годов XX столетия как гибрид двух слов «информация» и «автоматика» для обозначения науки об автоматизации процессов обработки информации. С этого времени начинается период бурного развития электронно-вычислительной техники и ее внедрения во многие области человеческой деятельности. Однако информатика не сводится только к компьютерам. Современная информатика делится на теоретическую и прикладную. *Теоретическая информатика* включает в себя множество научных дисциплин, для которых общим предметом изучения является информация. Среди них: теория информации, теория алгоритмов, теоретическая кибернетика, математическое и информационное моделирование, дискретная математика, искусственный интеллект и др. Многие из этих дисциплин зародились еще до появления первой ЭВМ. С наступлением компьютерной эпохи их развитие резко ускорилось, появились новые разделы теоретической информатики, такие как «Архитектура ЭВМ», «Теория и методика программирования» и др.

Компьютер является основным инструментом современной информатики. По своему назначению компьютер — это универсальное автоматическое, программно управляемое устройство для работы с информацией. Поэтому к *прикладной информатике* относятся все области разработки и использования компьютерной техники, ее аппаратных и программных составляющих (в американской терминологии hardware & software). К аппаратным средствам относятся компьютеры, технические средства хранения и отображения информации, передачи данных по сетям. Программное обеспечение компьютерных систем содержит многие тысячи программных продуктов системного и прикладного назначения.

В настоящее время происходит активный процесс *информатизации общества*. Под информатизацией понимается внедрение компьютерной техники и новых информационных технологий (НИТ) в различные сферы производства, общественной и личной жизни людей. Как в начале XX века электрификация привела к значительному росту общественного производства и улучшению бытовых условий жизни людей, так в конце XX, начале XXI века такое же революционное значение для развития человеческого общества имеет информатизация. *Научной и инструментальной базой информатизации общества являются теоретическая и прикладная информатика.*

2.2. Содержание образовательной области информатики

Согласно концепции Федеральных компонентов государственного образовательного стандарта (ФК ГОС) по информатике [15] основными целями обучения являются:

1. Формирование основ научного мировоззрения.
2. Развитие мышления учащихся.
3. Подготовка учащихся к практическому труду, продолжению образования.

Из сформулированного выше принципа о том, что содержание образовательной области является отражением соответствующей предметной области, следует, что в школьной информатике должны найти отражение как теоретическая, так и прикладная часть предметной области информатики. Иначе говоря, в школе на уроках информатики должны изучаться элементы некоторых из перечисленных выше теоретических научных дисциплин, а также компьютерная техника и технологии. Наличие теоретического, научного содержания делает этот предмет элементом фундаментального школьного образования, решает первую из перечисленных выше задач — задачу формирования основ научного мировоззрения учащихся. Согласно проекту ФК ГОС она заключается в формировании представлений об информации (информационных процессах) как одного из трех основополагающих понятий: *вещества, энергии, информации*, на основе которых строится современная научная картина мира. Естественно, что в школьный курс должны войти лишь некоторые элементарные положения теоретической информатики, подобно тому, как в школьную программу входит элементарная математика.

Безусловно, информатика вносит значительный вклад в достижение второй цели образования — развития мышления детей. Хотя в этом отношении проект ФК ГОС значительное место оставляет за процедурным (алгоритмическим) мышлением, необходимо учитывать, что появились новые линии развития навыков у детей в работе с информацией. Это умение находить информацию, необходимую для решения проблемы; определять достаточность этой информации, избавляться от избыточности; распознавать, классифицировать, упорядочивать информацию; структурировать информацию и прочее. Общепризнано, что развитие подобных навыков целесообразно начинать как можно раньше. Данное направление стало содержанием ряда версий пропедевтической информатики (для младших классов), причем как в компьютерном, так и в бескомпьютерном вариантах.

Третья задача — подготовка школьников к практической деятельности, труду, продолжению образования. В проекте ФК ГОС по этому поводу сказано: «Реализация этой задачи связана сейчас с ведущей ролью обучения информатике в формировании компьютерной грамотности и информационной культуры школьников, навыков использования НИТ, важнейших компонентов подготовки к практической деятельности, жизни в информационном обществе».

В Проекте федерального компонента образовательного стандарта по информатике определены основные содержательные линии предмета, обозначено их внутреннее наполнение. Таким образом, в этом документе нарисованы контуры образовательной области информатики в ее инвариантном виде. Вот перечень содержательных линий базового курса:

1. Линия информации и информационных процессов.
2. Линия представления информации.
3. Алгоритмическая линия.
4. Линия компьютера.
5. Линия формализации и моделирования.
6. Линия информационных технологий.

В появившемся позднее (1999 г.) Обязательном минимуме с небольшими вариациями повторяется тот же перечень содержательных линий.

Следует иметь в виду, что данный перечень — это не программа курса, определяющая последовательность его изучения. На базе обязательного минимума строятся программы обучения, учебные планы изучения курса в конкретных регионах, школах, классах. При этом должны учитываться все обстоятельства: возрастные особенности учащихся и выделенное учебное время, имеющиеся в распоряжении средства обучения (компьютеры, программное обеспечение, учебники и пособия), специфика учебного заведения.

Обязательный минимум регламентирует главным образом знания учащихся. Стандартизировать на федеральном уровне практические навыки, связанные с ра-

ботой на компьютере, пока не представляется возможным из-за отсутствия единообразия в обеспечении школ компьютерной техникой. Такая стандартизация возможна лишь на уровне школьного компонента. В тех регионах, где удастся реализовать стандарт на техническое обеспечение школ, возможны единые требования к практическим навыкам учащихся, изложенные в форме регионального компонента стандарта.

3. Авторская концепция содержания базового курса

Содержание авторской версии базового курса информатики, реализованного в учебно-методическом комплекте [1] – [5], находится в полном согласии с концепцией проекта стандарта и обязательного минимума. Вот как выглядит перечень тематических разделов курса (он совпадает с названиями глав учебника):

1. Человек и информация.
2. Первое знакомство с компьютером.
3. Текстовая информация и компьютер.
4. Графическая информация и компьютер.
5. Передача информации в компьютерных сетях.
6. Модели и таблицы.
7. Базы данных.
8. Табличные вычисления на компьютере.
9. Информация и управление.
10. Как работает процессор ЭВМ.
11. Искусственный интеллект и базы знаний.
12. Введение в программирование.

Логика общеобразовательного содержания курса кратко может быть выражена следующим описанием.

1. Рассматривается вопрос о том, что такое информация с позиции человека; анализируется информационная функция человека; отражается роль языков как средства представления информации, а также средства информационных коммуникаций; раскрываются под-

ходы к измерению информации (глава 1: «Человек и информация»).

2. Постулируется положение о том, что компьютер по своей организации моделирует информационную функцию человека. Компьютер — это программно управляемый автомат, способный работать с числовой и символьной информацией, изображением и звуком. Компьютер представляет собой единство аппаратной и программной составляющих. Даются первоначальные сведения об архитектуре ЭВМ и составе программного обеспечения. (Глава 2: «Первое знакомство с компьютером».)

3. Показывается, что прикладное назначение ЭВМ складывается из двух составляющих: компьютер как инструментальное средство работы с информацией и компьютер как средство информационного моделирования.

В разделах, где изучаются текстовые и графические редакторы, а также средства компьютерных телекоммуникаций, учащиеся знакомятся с чисто инструментальным применением ЭВМ. Одновременно изучаются способы компьютерного представления текстовой и графической информации, преобразования информации в процессе передачи по сетям. (Главы 3, 4, 5: «Текстовая информация и компьютер», «Графическая информация и компьютер», «Передача информации в компьютерных сетях».)

4. При переходе к изучению использования компьютера для целей информационного моделирования раскрывается понятие модели. Вводится представление об информационной модели, о видах информационных моделей. Особое внимание уделяется разным формам табличного представления данных — табличным моделям. В дополнительном разделе для углубленного изучения вводятся понятия системы, структуры, графа; даются начальные представления о системном анализе. (Глава 6: «Модели и таблицы».)

5. База данных трактуется как статическая информационная модель объекта; электронные таблицы, так же, как и реляционная БД, — как табличная форма информационной модели, но уже с элементами динамики, математического моделирования предметной области. (Главы 7, 8: «Базы данных», «Табличные вычисления на компьютере».)

6. Применение ЭВМ в управлении процессами основано на винеровской схеме информационной модели управления с обратной связью. В этой теме возникает разговор об алгоритмах автоматического управления самыми разнообразными исполнителями, о типах алгоритмов, о методике их построения. Сам алгоритм трактуется как управляющая информация, необходимая для функционирования системы управления. (Глава 9: «Информация и управление».)

7. Наиболее глубокий уровень в описании архитектуры компьютера — описание устройства и работы процессора, языка машинных команд. Для знакомства с этими вопросами используется простая модель компьютера, предназначенного для работы с целыми числами. (Глава 10: «Как работает процессор ЭВМ».)

8. Линия моделирования имеет две ветви: моделирование объектов и процессов (об этом говорилось выше) и моделирование знаний. Второе направление связано с разделом информатики, называемым искусственным интеллектом. Дается представление о проблемах, решаемых в области искусственного интеллекта. Вводится понятие модели знаний. Дается представление о том, что такое логическая модель знаний. (Глава 11: «Искусственный интеллект и базы знаний».)

9. Программа для ЭВМ — это алгоритм решения задачи, записанный на языке программирования. Описываются особенности построения алгоритмов для работы с величинами. Основным средством современного программирования являются системы программирования на языках высокого уровня. Даются начальные сведения о программировании на языке Паскаль. Опи-

сываются основы технологии решения вычислительных задач с использованием программирования. (Глава 12: «Введение в программирование».)

Из данного описания следует, что базовый курс вводит учащихся в целый ряд научных дисциплин, составляющих информатику: теорию информации, архитектуру ЭВМ, кибернетику, системный анализ, теорию алгоритмов, искусственный интеллект, программирование. Кроме того, базовый курс дает учащимся основные знания в области современных информационно-коммуникационных технологий. Учащиеся получают практические навыки работы с конкретным видом вычислительной техники, с конкретными средствами программного обеспечения общего назначения. Социальная область информатики представлена исторической линией, присутствующей в учебнике: «докомпьютерная» история, история развития технических и программных средств ЭВМ.

Слово «базовый» в названии курса имеет три смысла: во-первых, он дает базовые знания и навыки, позволяющие учащемуся ориентироваться в современной среде компьютеров и программ; во-вторых, эти знания и навыки дают базу для дальнейшего образования в этой области. Это образование может быть продолжено в старших классах школы в форме разнообразных профильных курсов. В-третьих, вторую ступень школьного образования, на которую ориентирован курс, принято называть базовой.

Современная парадигма школьной информатики начала формироваться в середине 90-х годов. Значительным событием в этом процессе было издание в 1994 году книги «Информатика. Энциклопедический словарь для начинающих» [17]. Впервые в учебном пособии, предназначенном для школы, была раскрыта широкая панорама научного содержания информатики. Эту же линию поддерживает учебник по информатике для педагогических вузов [18], выпущенный в 1999 году. В 2001 году издательством «Академия» выпущен-

но учебное пособие для студентов педагогических вузов «Методика преподавания информатики» [19]. Знакомство с этими книгами окажет большую помощь школьному учителю информатики.

4. Общие методические рекомендации

Методическое мастерство учителя информатики должно быть направлено на то, чтобы за суетой рутинной работы на уроках, решением сиюминутных организационных и технических проблем не потерять главные цели изучения предмета: общеобразовательные, развивающие и прагматические. Решению этих задач должны помочь сформулированные ниже рекомендации.

1. *Информация — центральное понятие курса.* Учитель не должен упускать это из вида при изложении любой темы. Каждый раздел — это разговор об информации и информационных процессах (ЭВМ — это универсальное средство для работы с информацией; алгоритм — это управляющая информация).

2. *Принцип системности.* В процессе изучения курса в сознании учеников строится взаимосвязанная система знаний. Логика курса в целом должна просматриваться как в его структуре, так и в содержании отдельных разделов. Учащиеся должны понять необходимость каждого раздела в его месте в общей структуре курса, увидеть за «деревьями» отдельных тем весь «лес» системы знаний предмета.

3. *Принцип параллельности в освоении фундаментальной и прагматической составляющих курса.* Фундаментальная (общеобразовательная) и прагматическая (технологическая) компоненты курса должны изучаться параллельно. В разных разделах их соотношение различно. Но и в разделах, связанных с ИТ, обязательно присутствует фундаментальная компонента. Для базового курса она первична.

4. *Принцип исполнителя.* Во всех темах, касающихся приложений ЭВМ, проводится методическая концепция: «ЭВМ + прикладное ПО = исполнитель для определенного вида работ с информацией». Здесь можно говорить об архитектуре исполнителя, которая описывается следующими компонентами: среда, режимы работы, система команд, данные (обрабатываемая информация).

5. *Принцип освоения методики самообучения.* Информатика и компьютерные технологии — быстро развивающиеся области. Поэтому человеку, деятельность которого связана с компьютерами, постоянно приходится обучаться. Методическая последовательность изложения материала должна быть такой (прежде всего в разделах, посвященных ИТ), чтобы давать учащимся схему организации самообразования в этом предмете. Необходимо приучать учеников к самостоятельному использованию дополнительной справочной литературы, а также справочных разделов используемых программных продуктов.

6. *Принцип историзма.* Ученики обязательно должны познакомиться с историей информатики, знать основные имена, связанные с ней. Знание исторической канвы помогает сформировать в сознании учеников цельное представление об изучаемой дисциплине, рассматривать ее в контексте истории развития общества. Не должно быть ни одного школьного предмета без «лиц и событий».

5. Тематическое планирование занятий

Предлагаемый тематический план учебных занятий ориентирован на преподавание курса в 7—9 классах общим объемом 102 учебных часа. В графе «Практика» указано время для практической работы на компьютере, а также время для проведения контроля

знаний и умений учащихся (тесты, контрольные работы). Данная таблица не является поурочным планом и служит лишь ориентиром преподавателю при составлении своего рабочего варианта плана уроков. Кроме того, разбивка материала по классам также является условной. В каждой конкретной школе планирование уроков должно быть согласовано с учебным планом школы. Рекомендуемые варианты поурочных планов содержатся в четвертой части настоящего пособия.

**Тематическое планирование учебного времени
для 7 класса, объем 34 ч**

Глава учебника	Тема	Всего часов	Теория	Практика
1	1. Введение в предмет.	1	1	—
	2. Информация и знания. Информация и языки. Информационные процессы.	2	2	—
	3. Измерение информации: алфавитный подход. Единицы измерения информации.	2	1	1
	4. Предыстория информатики. История чисел и системы счисления	3	2	1
2	1. Начальные сведения об архитектуре ЭВМ. Знакомство с основными устройствами ЭВМ. Принципы организации внутренней и внешней памяти компьютера. Устройство ПК.	4	3	1
	2. Виды программного обеспечения (ПО). Системное ПО. Операционные системы; организация диалога с пользователем, файловая система, управление устройствами. Пользовательский интерфейс	7	3	4

Глава учебника	Тема	Всего часов	Теория	Практика
3	1. Тексты в компьютерной памяти: кодирование символов, текстовые файлы.	2	1	1
	2. Текстовые редакторы и практика работы с ними	7	2	5
4	1. Компьютерная графика: области применения, технические средства, принципы кодирования изображения.	2	2	–
	2. Графические редакторы и практика работы с ними	4	1	3

**Тематическое планирование учебного времени
для 8 класса, объем 34 ч**

Глава учебника	Тема	Всего часов	Теория	Практика
5	1. Компьютерные сети: виды, структура, принципы функционирования, технические устройства.	2	2	–
	2. Информационные услуги: электронная почта, телеконференция, Всемирная паутина (Интернет). Практика работы в Интернете	9	4	5
6	1. Понятие информационной модели, информационной структуры. Табличная организация информации	2	2	–

Глава учебника	Тема	Всего часов	Теория	Практика
7	1. Базы данных (БД): основные понятия, типы данных, системы управления БД и принципы работы с ними. Представление о СУБД.	2	2	-
	2. Просмотр и редактирование БД. Условия выбора информации, простые и сложные логические выражения. Логические операции. Удаление и сортировка записей.	5	2	3
	4. Многотабличные БД. Элементы проектирования и создания БД	4	2	2
8	1. Табличные расчеты и электронные таблицы. Структура электронной таблицы. Информация в ячейках таблицы: тексты, числа, формулы.	3	2	1
	2. Абсолютная и относительная адресация, функции. Логические операции и условная функция. Сортировка таблицы.	5	2	3
	3. Построение графиков и диаграмм с помощью электронных таблиц	2	1	1

**Тематическое планирование учебного времени
для 9 класса, объем 34 ч**

Глава учебника	Тема	Всего часов	Теория	Практика
Дополнение к главе 1	1. Содержательный подход к измерению информации. Вероятность и информация.	3	3	2(4)
	2. Технология создания компьютерных презентаций (дополнительная литература)	4	-	4

Глава учебника	Тема	Всего часов	Теория	Практика
9	1. Кибернетика: модели управления. Системы управления с использованием компьютера.	1	1	—
	2. Понятие алгоритма и его свойства. Графический учебный исполнитель: система команд исполнителя, среда исполнителя.	2	1	1
	3. Языки для записи алгоритмов (язык блок-схем, алгоритмический язык АЯ). Линейные, ветвящиеся и циклические алгоритмы. Вспомогательные алгоритмы. Метод пошаговой детализации	8	4	4
10	1. Двоичная система счисления, двоичная арифметика и внутреннее представление данных.	3	2	1
	2. Архитектура учебного компьютера. Понятие о программном управлении компьютером. Язык машинных команд учебного компьютера.	6	3	3
	3. История развития ЭВМ	1	1	—
11	1. Модели знаний и искусственный интеллект. Базы знаний (БЗ) и экспертные системы	2	2	—
	2. Пролог как средство построения БЗ	6	3	3

Методические рекомендации по изучению разделов курса

1. Введение в предмет

Разделы учебника: Введение.

Основные цели. Цель вводного занятия состоит в том, чтобы познакомить учащихся с новым предметом, рассказать, о чем пойдет речь на уроках, чему ученики должны научиться, какое значение для их образования и дальнейшей жизни имеет информатика.

Изучаемые вопросы

- Что такое информатика как наука и как область практической деятельности людей.
- Вещество, энергия, информация — три составляющие окружающего нас мира.
- Роль информации в жизни людей.
- Разделы, изучаемые в курсе информатики (система знаний курса).

Дополнительный материал для вводной беседы

Место информатики в научном мировоззрении

Окружающий нас мир бесконечно многообразен. Бесплодны попытки человека понять любой его объект, любое явление в исчерпывающей полноте. С зарождения науки в древности и до наших дней основным методическим принципом познания является моделирование. Модель — это упрощенное по сравнению с реальностью описание объекта или явления, учитывающее только некоторые существенные, с точки зрения исследователя, его свойства. Моделирование всегда связано с абстрагированием, с выделением общего из множества

частностей. Любой ученый сознает, что понять — это значит найти общность, отвлекшись от частных, а затем объяснить частности через эту понятую общность.

Одним из первых обобщенных, абстрактных понятий науки стало понятие «вещество». В разнообразии материальных объектов ученые пытались увидеть некоторое единство, отыскать «первоматерию», атомы вещества. Эта идея развивалась от философии древней Греции (Демокрит, Эпикур, Лукреций) до самой современной квантовой теории вещества. После разгадки природы вещества, его структуры, казалось, что все в мире можно объяснить, описав его как совокупность взаимодействующих материальных частиц.

Следующим обобщающим понятием в истории науки стало понятие «энергия». Его появление было связано с развитием техники, созданием двигателей, технических преобразователей энергии. Наука стала активно использовать «энергетический язык» в описании природы. Физические, химические, биологические процессы стали рассматриваться с позиции передачи и преобразования энергии. Знаменитая формула Эйнштейна ($E = mc^2$) казалось, окончательно закрепила всеобщность энергетического подхода.

Желая исследовать все более сложные объекты в технике, биологии, обществе, наука встала перед фактом невозможности детального описания их поведения на языке материально-энергетических моделей.

В середине XX века появляется и развивается новая научная дисциплина — кибернетика. Ее основатель (1948) — американский математик Норберт Винер [20]. Термин «кибернетика» на греческом языке означает «искусство управления». *Винер назвал кибернетикой науку об управлении и связи в живом организме и машине.* Объекты, рассматриваемые с позиции кибернетики, принято называть *кибернетическими системами*. В дальнейшем кибернетический подход стал применяться и к описанию социумов (социальных объектов и явлений).

Центральным понятием кибернетики является *информация*. Между элементами кибернетической системы, а также между различными системами имеют место информационные взаимодействия, то есть обмен управляющими сигналами, знаками, командами. В рамках кибернетики не рассматривается физическое, энергетическое взаимодействие, а только информационное.

Кибернетика породила новый системно-информационный взгляд на природу. Вещество — энергия — информация: это три точки зрения, три стороны, с которых наука сумела посмотреть на бесконечно разнообразный мир.

В 60–70-е годы XX века информатика выделилась из кибернетики как самостоятельная научная дисциплина. Предметом информатики является собственно *информация*, способы ее представления, передачи и обработки. В современном виде информатика оформилась с появлением и развитием электронно-вычислительных машин (ЭВМ).

Что изучается в курсе информатики

Предварительно отметим, что предлагаемый ниже перечень содержательных линий базового курса информатики имеет некоторые отличия от списка содержательных линий, присутствующего в Обязательном минимуме содержания образования по информатике (федеральный компонент) от 1997 и 1999 г. В Проекте образовательного стандарта по информатике, опубликованном в 2003 г., структура содержательных линий вообще отсутствует. Временное отсутствие утвержденного образовательного стандарта дает право авторам предлагать свои варианты структурирования содержания курса информатики.

В своей повседневной жизни, в производственной деятельности человек постоянно имеет дело с тремя упомянутыми выше субстанциями: веществом, энергией, информацией. Мы не можем жить без пищи,

одежды, жилья, предметов быта, транспорта и пр. Все это — материальные объекты (в общем понятии — вещество). Электричество, отопление в наших домах — это энергия. Пресса, радио, телевидение, книги — информация.

Аналогичные примеры можно привести из области производства, которую по отношению к трем названным субстанциям можно разделить на:

- материальное производство (производство сельскохозяйственной и промышленной продукции);
- энергетику (производство, преобразование, транспортировка энергии);
- информационную сферу (получение, хранение, передача, переработка информации).

К последней, например, относятся пресса, образование, наука, связь и пр.

Информатика — современная научная база информационной сферы деятельности людей. *Информатика — это наука, изучающая все аспекты получения, хранения, преобразования, передачи и использования информации.*

Информация и информационные процессы — первая содержательная линия предлагаемого для изучения базового курса информатики.

К определению информации могут быть два подхода: кибернетический и традиционный, возникший достаточно давно, задолго до возникновения кибернетики. В начале изучения информатики следует отталкиваться от традиционного представления о том, что такое информация. Информация — это сведения, знания, сообщения, которые человек воспринимает из внешнего мира через органы чувств (зрение, слух, вкус, обоняние, осязание). Информация хранится в памяти человека, а также в знаковой форме в книгах, магнитных записях и пр.

Что человек делает с информацией? Воспринимает; запоминает или сохраняет в записях; обрабатывает,

в итоге получая новую информацию; передает другим людям.

Здесь учитель может использовать материал § 5 «Предыстория информатики» для краткого рассказа об истории развития средств работы с информацией, в том числе и вычислительной техники. Вершиной этого развития стало создание *электронно-вычислительных машин (ЭВМ)*. В последнее время укоренился термин *компьютер*, от английского *computer* — вычислитель.

Первый компьютер был построен в 1946 году (ENIAC — США). Первые ЭВМ применялись только для математических расчетов. Современные компьютеры работают с любыми видами информации, выполняя любую обработку.

Современный компьютер — это универсальное автоматическое устройство для работы с информацией.

Компьютер (ЭВМ). — вторая содержательная линия курса информатики.

Компьютер — техническое устройство. Поэтому для того чтобы он выполнял определенную работу, ему нужно управлять. Управление работой компьютера происходит с помощью программ (программного обеспечения ЭВМ). Современный компьютер — это единая система аппаратных средств и программного обеспечения. Без программ ЭВМ — мертвое железо. Программное обеспечение современного компьютера — достаточно сложная многоуровневая система.

В состав программного обеспечения современных ЭВМ входит большое число программ, выполняющих разнообразные типовые виды работы с информацией. Например, это текстовые редакторы, позволяющие формировать тексты книг, статей и пр. на ЭВМ; электронные таблицы, позволяющие легко производить самые разнообразные расчеты; базы данных для хране-

ния больших объемов разнообразной информации, в которой ЭВМ быстро находит нужные сведения.

Кроме того, существуют специализированные программные системы, например, для ведения бухгалтерского учета, для автоматического конструирования технических устройств, для перевода текстов с иностранного языка и многие другие.

Все эти программы составлены профессиональными программистами. Но для того чтобы ими воспользоваться, не требуется знания программирования. В последнее время закрепился термин «пользователь ЭВМ» — человек, работающий на компьютере по готовым программам. Перечисленные выше программные средства составляют современные *компьютерные информационные технологии*.

Компьютерные информационные технологии — третья содержательная линия курса информатики.

Как уже говорилось выше, «родительницей» современной информатики была наука «Кибернетика». Кибернетика изучает информационные основы процессов управления. Использование компьютеров для управления (в технике, производстве, общественной жизни) — важная область применения ЭВМ. Важнейшим понятием здесь является понятие *алгоритма управления*. *Алгоритм* — это последовательность команд управления объектом.

Процессы управления (информационные основы) — четвертая содержательная линия курса информатики.

Всякое описание реального объекта, процесса, явления всегда происходит приближенно. Приближенное описание называется *информационной моделью*. Такие модельные описания закладываются в память компьютера и на их основе производятся расчеты, прогнозирующие поведение объекта, управление объектом или процессом и пр. Все это называется компьютерным мо-

делированием. Наиболее сложные информационные модели — это модели знаний человека в различных областях. На их основе создаются компьютерные системы *искусственного интеллекта*.

Информационное моделирование — пятая содержательная линия курса информатики.

Большинство людей, использующих компьютеры — это непрограммирующие пользователи. Первый уровень компьютерной грамотности — пользовательский уровень. Задача базового курса информатики состоит, прежде всего, в освоении учениками этого уровня. Умение программировать — это профессиональный уровень компьютерной грамотности. В рамках базового курса даются лишь начальные сведения о приемах программирования, о языках программирования на ЭВМ.

Программирование — шестая содержательная линия курса информатики.

Подробное изучение программирования может быть предметом профильного курса информатики в старших классах.

Общие методические рекомендации

Специфика информатики состоит в том, что абсолютное большинство учеников имеют высокую первоначальную мотивацию к изучению этого предмета. В их сознании информатика связывается с компьютерами, а научиться работать на компьютере хочет практически каждый. Педагогическая задача учителя, начиная с первого занятия, состоит в том, чтобы, не потеряв положительный заряд этой первоначальной мотивации, настроить учеников на изучение очень серьезного и непростого общеобразовательного предмета. Необходимо дать понять ученикам, что получение практических навыков работы с компьютером — это не самоцель, а всего лишь средство для освоения информатики.

Объем материала, излагаемого на вводном занятии, и уровень его изложения определяются учителем, исходя из имеющегося учебного времени, уровня подготовленности учеников и других специфических условий. В минимальном объеме можно ограничиться содержанием раздела «Введение» учебника.

На первом, вводном занятии закладывается начало будущей системы знаний учеников по предмету. Вся система знаний может быть представлена в виде дерева (иерархической многоуровневой структуры) с перекрестными ссылками. Корень этого дерева («Базовый курс информатики») и шесть ответвлений от него изображены на рис. 1.1. Схема дает наглядное представление о содержательной структуре курса. Эта структура включает шесть содержательных линий. Дальнейшее изучение курса будем рассматривать как продолжение ветвей данной схемы путем детализации их содержания. В схематическом виде такая детализация представлена в пособии [4]. Очень полезно, если рис. 1.1, начиная с

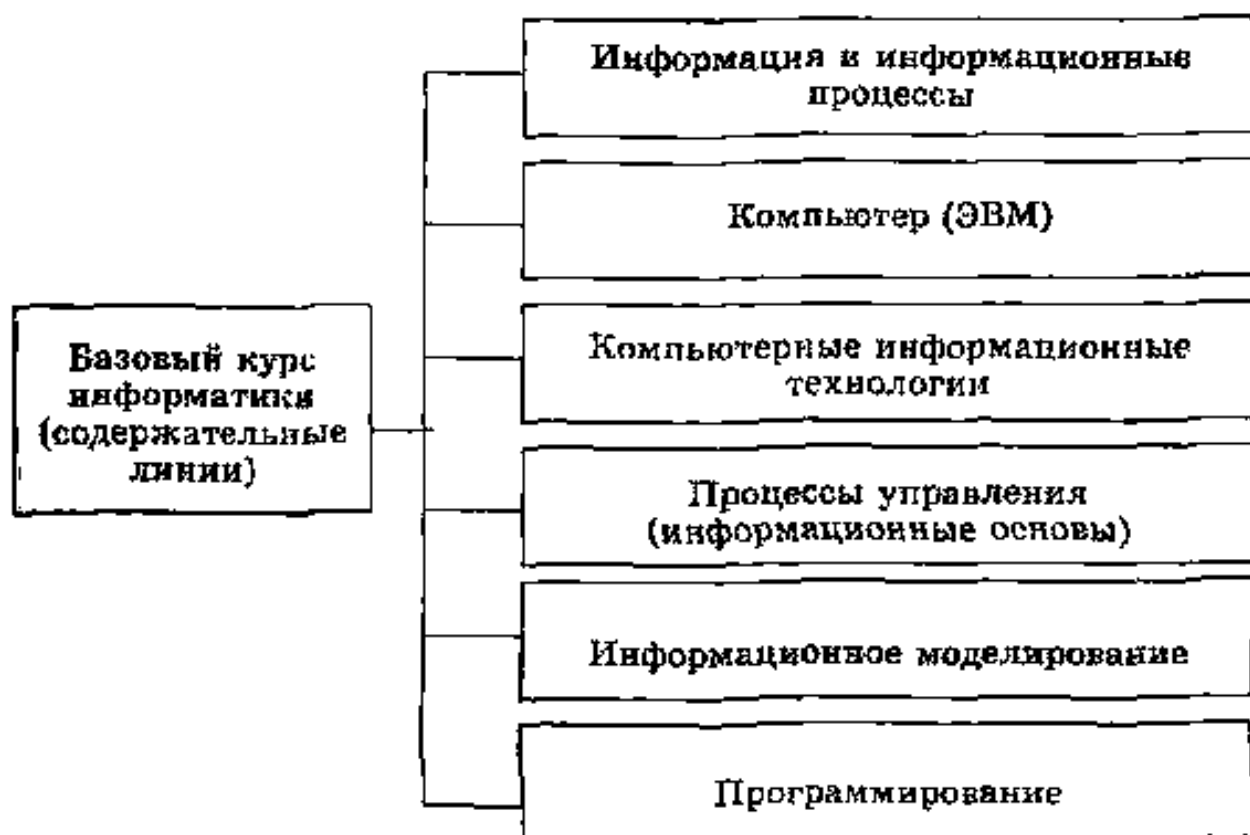


Рис. 1.1

Структура содержания базового курса информатики

первого занятия и на всех последующих, будет присутствовать в классе в виде плаката. Переходя к изучению новой темы, следует возвращаться к схеме, указывая место этой темы в общей структуре предмета.

2. Информация и знания

Разделы учебника: часть I, § 1, 2, 3. **Дополнительный материал:** часть II, разделы 1.4, 1.5.

Основные цели. Раскрыть смысл понятия «информация» в контексте жизни и деятельности человека. Раскрыть понятие информативности сообщения с субъективной (содержательной) точки зрения на информацию. Ввести понятие «информационные процессы» и показать три их разновидности: процесс хранения, процесс передачи и процесс обработки информации. Раскрыть роль языков в информационных процессах.

Изучаемые вопросы

- Чем является информация для человека.
- Декларативные и процедурные знания (информация).
- Три типа информационных процессов.
- Роль органов чувств человека в процессе восприятия им информации.
- Естественные и формальные языки.

Методические рекомендации по объяснению теоретического материала

1. Основные содержательные и методические проблемы изучения всего материала главы 1 «Человек и информация» связаны с проблемой определения понятия «информация». *Информация — фундаментальное понятие науки, поэтому определить его исчерпывающим образом*

через какие-то более простые понятия невозможно. В информатике известны различные подходы к этому понятию, и каждый подход дает свое определение информации. Каждый вариант определения информации обладает некоторой неполнотой. В базовом курсе рассматриваются два различных подхода к информации.

Первый можно назвать *субъективным подходом*, при котором информация рассматривается с точки зрения ее роли в жизни и деятельности человека. С этой позиции *информация — это знания, сведения, которыми обладает человек, которые он получает из окружающего мира.*

Второй подход можно назвать *кибернетическим*, ибо зародился он в кибернетике. Именно этот подход позволяет создавать машины, работающие с информацией. С этой точки зрения *информация — это содержание последовательностей символов (сигналов) из некоторого алфавита.* В таком случае все три вида информационных процессов (хранение, передача, обработка) сводятся к действиям над символами.

При изучении материала § 1 следует отталкиваться от интуитивных представлений об информации, имеющихся у детей. Целесообразно вести беседу в форме диалога, задавая ученикам вопросы, на которые они в состоянии ответить. Не следует сразу требовать от них определения информации, но подвести их к этому определению с помощью понятных вопросов вполне возможно. Вопросы, например, можно задавать в следующем порядке.

— Расскажите, откуда вы получаете информацию?
Наверняка услышите в ответ:

— Из книг, из радио и телепередач ...

Дальше попросите учеников привести примеры какой-нибудь информации, которую они получили сегодня. Например, кто-нибудь ответит:

— Утром по радио я слышал прогноз погоды.

Ухватившись за такой ответ, учитель подводит учеников к окончательному выводу:

— Значит, вначале ты не знал, какая будет погода, а после прослушивания радио стал знать. Следовательно, получив информацию, ты получил новые знания.

Таким образом, учитель вместе с учениками приходит к определению: *информация для человека — это знания, которые он получает из различных источников*. Далее на многочисленных знакомых детям примерах следует закрепить это определение.

2. Разговор перешел к теме анализа человеческих знаний. Следует понимать, что это очень сложный вопрос. Этим вопросом занимается наука «Искусственный интеллект», одна из главных задач которой — анализ и моделирование человеческих знаний. С элементами этой науки ученики познакомятся в главе 11 учебника. В данной теме предлагается один из вариантов классификации знаний: деление знаний (и, следовательно, информации) на *декларативные и процедурные*. Здесь достаточно добиться от учеников умения приводить примеры того и другого типа знаний, используя схему: «Я знаю, что...» (декларативные знания) и «Я знаю, как...» (процедурные знания).

Учителю следует понимать, какое продолжение имеет этот вопрос в дальнейшем содержании базового курса. Когда речь пойдет об информации в компьютере, то ученики узнают, что вся она делится на данные и программы. Данные можно отнести к информации декларативного типа, программы — к процедурной информации.

3. В данной теме используется понятие «сообщение», интуитивно понятное ученикам. Тем не менее, может возникнуть потребность расшифровать это понятие. Сообщение — это информационный поток, который в процессе передачи информации поступает

к приемнику. Сообщение — это и речь, которую мы слушаем (радиосообщение, объяснение учителя), и воспринимаемые нами зрительные образы (фильм по телевизору, сигнал светофора), и текст книги, которую мы читаем и так далее.

Вопрос об информативности сообщения следует обсуждать на примерах, предлагаемых учителем и учениками. Правило: *информативным назовем сообщение, которое пополняет знания человека, то есть несет для него информацию*. Для разных людей одно и то же сообщение с точки зрения его информативности может быть разным. Если сведения «старые», то есть человек это уже знает или содержание сообщения непонятно человеку, то для него это сообщение неинформативно. Информативно то сообщение, которое содержит новые и понятные сведения.

Еще раз хочется подчеркнуть всю познавательную (для учеников) и методическую (для учителя) сложность данного материала. Нельзя отождествлять понятия «информация» и «информативность сообщения». Следующий пример иллюстрирует различие понятий. Вопрос: «Содержит ли информацию вузовский учебник по высшей математике с точки зрения первоклассника?». Ответ: «Да, содержит с любой точки зрения! Потому что в учебнике заключены знания людей: авторов учебника, создателей математического аппарата (Ньютона, Лейбница и др.), современных математиков». Эта истина — абсолютна. Другой вопрос: «Будет ли информативным текст этого учебника для первоклассника, если он попытается его прочитать? Иначе говоря, может ли первоклассник с помощью этого учебника пополнить собственные знания?» Очевидно, что ответ отрицательный. Читая учебник, то есть получая сообщения, первоклассник ничего не поймет, а, стало быть, не обратит информацию из учебника в собственные знания. Введение понятия «информативность сообщения» является первым подходом к изучению вопроса об измерении информации. Если сообщение неинфор-

мативно для человека, то количество информации в нем с точки зрения этого человека равно нулю. Количество информации в информативном сообщении больше нуля.

При объяснении этой темы можно предложить ученикам поиграть в своеобразную викторину. Например, учитель предлагает детям перечень вопросов, ответы на которые они молча записывают. Если ученик не знает ответа, он ставит знак вопроса. После этого учитель дает правильные ответы на свои вопросы, а ученик, записав ответы учителя, отмечает, какие из них оказались для него информативными (+), какие — нет (—). При этом для сообщений, отмеченных «минусом», нужно указать причину отсутствия информации: «не новое» (это я знаю), «непонятное». Например, список вопросов и ответы одного из учеников могут быть следующими:

Вопрос учителя	Ответ ученика	Сообщение учителя	Информативность сообщения	Причина неинформативности
1. Какой город является столицей Франции?	Столица Франции — Париж.	Столица Франции — Париж.	—	Не новое
2. Что изучает коллоидная химия?	?	Коллоидная химия изучает дисперсионные состояния систем, обладающих высокой степенью раздробленности.	—	Непонятное

Вопрос учителя	Ответ ученика	Сообщение учителя	Информативность сообщения	Причина неинформативности
3. Какую высоту и вес имеет Эйфелева башня?	?	Эйфелева башня имеет высоту 300 метров и вес 9000 тонн.	+	

4. Обсуждая проблему восприятия человеком информации из внешнего мира, нужно обратить внимание учеников на то, что человек обладает множеством каналов, по которым в его мозг (память) поступает информация. Эти каналы — наши органы чувств. Их пять: зрение, слух, вкус, обоняние, осязание. Если роль первых двух для восприятия информации очевидна, то для понимания того, что вкусовые и осязательные ощущения, запахи также являются источниками информации, требуется пояснение. Объяснение этому следующее: мы помним запахи знакомых предметов, вкус знакомой пищи, на ощупь узнаем некоторые вещи. Но то, что мы помним — хранится в нашей памяти. Значит, это тоже наши знания, а стало быть — информация.

5. Понятие о языке как некотором знаковом (символьном) способе представления информации также является одним из фундаментальных в информатике. Впервые встречаясь с этим понятием в базовом курсе, следует систематизировать представления учеников, разделив языки на естественные и формальные. Впоследствии ученики еще не раз встретятся с понятием языка в информатике: язык двоичного кодирования,

язык машинных команд, командный язык операционной системы, языки программирования и др. Все это — формальные языки. Ученики также узнают, что одной из целей разработки на ЭВМ систем искусственного интеллекта является достижение понимания компьютером естественных языков.

6. Языковая форма представления информации — это последовательности знаков, сигналов (звуков). Но не всякую информацию, с которой имеет дело человек, можно свести к знакам. Не углубляясь в эти очень сложные вопросы, находящиеся на стыке информатики, физиологии, психологии и других наук, принимаем следующее допущение: всякую информацию, не сводящуюся к символьной форме, будем называть *образной*. *Образная информация* воспринимается всеми органами чувств человека.

7. Фундаментальное понятие информатики — *информационные процессы*. Следует дать понять ученикам, что какую бы сложную работу с информацией не выполнял человек, она сводится к трем составляющим: *хранению, передаче, обработке*. Объяснение этих вопросов также можно проводить в форме диалога. Следует отталкиваться от жизненного опыта учеников, их школьной практики. Ключевыми понятиями здесь являются: *память, носители информации, источник информации, приемник информации, канал связи*. При обсуждении различных вариантов обработки информации следует обратить внимание учеников на то, что обработка всегда происходит по определенным правилам, которые сами являются информацией (*процедурной*).

В варианте углубленного изучения базового курса следует использовать материал раздела 1.5 части II учебника, где рассказывается о схеме передачи информации К. Шеннона и о теории кодирования. Здесь даются важные для информатики понятия *кодирования и декодирования*.

8. Во многих учебниках информатики, так или иначе, рассматривается вопрос о роли информации для человека, о взаимодействии человека и информации. Обычно такие разделы носят названия «Информация вокруг нас» или какие-то другие, близкие к этому. Глава 1 нашего учебника называется «Человек и информация». Помимо упомянутых выше традиционных педагогических целей в ней реализуется еще одна цель: описать *информационную функцию человека*, то есть рассмотреть человека как высокоразвитую «биологическую информационную машину». Зачем это нужно? Для того, чтобы в дальнейшем, при рассказе об устройстве ЭВМ, воспользоваться *дидактическим приемом аналогии*, основанном на следующем тезисе: *компьютер — это техническая модель информационной функции человека*.

3. Как измерить информацию (алфавитный подход)

Разделы учебника: часть I, § 4. Дополнительный материал: часть II, раздел 1.3.

Основные цели. Познакомить учащихся с методом измерения информации в символьном сообщении. Ввести понятие алфавита, мощности алфавита. Научить детей вычислять количество информации, заключенное в тексте, составленном из символов определенного алфавита. Ввести основные единицы измерения информации, установить связь между ними.

Изучаемые вопросы.

- Что такое алфавит, мощность алфавита.
- Представление о двоичном алфавите.
- Что такое информационный вес символа в алфавите.
- Что такое бит, байт, килобайт, мегабайт.

- Как измерить информационный объем текста с алфавитной точки зрения.

Методические рекомендации по объяснению теоретического материала

1. Рассматриваемый в этой теме подход к измерению информации является альтернативным к содержательному подходу, который излагается в разделах 1.1, 1.2 части II учебника. *Здесь речь идет об измерении количества информации в тексте (символьном сообщении), составленном из символов некоторого алфавита. К содержанию текста такая мера информации отношения не имеет.* Поэтому такой подход можно назвать объективным, то есть не зависящим от воспринимающего его субъекта. Алфавитный подход — это единственный способ измерения информации, который может применяться по отношению к информации, циркулирующей в информационной технике, в компьютерах.

2. Опорным в этой теме является понятие *алфавита — конечного множества символов, используемых для представления информации.* Число символов в алфавите называется мощностью алфавита (этот термин взят из математической теории множеств). В основном содержании базового курса алфавитный подход рассматривается лишь с позиции *равновероятного приближения.* Это значит, допускается предположение о том, что вероятности появления всех символов алфавита в любой позиции в тексте одинаковы. Разумеется, это не соответствует реальности и является упрощающим предположением. Изучение этого вопроса с учетом разной вероятности встречаемости символов в тексте возможно лишь на углубленном уровне. Такой материал дается в части II учебника, в разделе 1.3. Он может изучаться после знакомства с разделами 1.1, 1.2, где рассматривается вероятностный подход к измерению информации.

3. В рассматриваемом приближении количество информации, которое несет в тексте каждый символ (i), вычисляется из уравнения Хартли: $2^i = N$, где N — мощность алфавита. Величину i можно назвать информационным весом символа. Если данная тема изучается в 7 классе, то возникает методическая проблема, связанная с тем, что ученики еще не знакомы из курса математики с операцией возведения в степень. В § 4 учебника фактически вводится определение операции возведения в целую степень, как многократного умножения числа самого на себя. Ученики 7 класса вполне в состоянии понять это правило. Знакомство с возведением в степень в курсе информатики окажется полезной пропедевтикой для изучения этой темы в математике.

4. Полезно обсудить с учениками следующий вопрос: какова минимальная мощность алфавита, с помощью которого можно записывать (кодировать) информацию?

Предположим, что используемый алфавит состоит всего из одного символа, например, «1». Интуитивно понятно, что сообщить что-либо с помощью единственного символа невозможно. Но это же доказывается строго с точки зрения алфавитного подхода. Информационный вес символа в таком алфавите находится из уравнения: $2^i = 1$. Но поскольку $1 = 2^0$, то отсюда следует, что $i = 0$ битов. Полученный вывод можно проиллюстрировать следующим образным примером. Представьте себе толстую книгу в 1000 страниц, на всех страницах которой написаны одни единицы (единственный символ используемого алфавита). Сколько информации в ней содержится? Ответ: несколько, ноль. Причем такой ответ получается с любой позиции, как с содержательной, так и с алфавитной.

Минимальная мощность алфавита, пригодного для передачи информации, равна 2. Такой алфавит называется двоичным алфавитом. Определение единицы измерения информации — бита дается в связи с введением представления о двоичном алфавите: один символ

двоичного алфавита несет 1 бит информации. С этим обстоятельством ученики снова встретятся, когда будут знакомиться с алфавитом внутреннего языка компьютера — языка двоичного кодирования.

5. В § 4 вводятся дополнительные единицы измерения информации. Следует обратить внимание учеников на то, что в любой метрической системе существуют единицы основные (эталонные) и производные от них. Например, основная физическая единица длины — метр. Но существуют миллиметр, сантиметр, километр. Расстояния разного размера удобно выражать через разные единицы. Так же обстоит дело и с измерением информации. 1 бит — это исходная единица. Следующая по величине единица — байт. Байт вводится как информационный вес символа из алфавита мощностью 256. Поскольку $256 = 2^8$, то 1 байт = 8 битов. Мы снова встречаемся с темой, которая является своеобразной пропедевтикой к будущему изучению компьютера. Уже в рамках данной темы можно сообщить ученикам, что компьютер для внешнего представления текстов и другой символьной информации использует алфавит мощностью 256 (во внутреннем представлении любая информация в компьютере кодируется в двоичном алфавите). Фактически, для выражения объема компьютерной информации, в качестве основной единицы используется байт.

6. Количество информации во всем тексте (I), состоящем из K символов, равно произведению информационного веса символа на K : $I = i \times K$. Эту величину можно назвать информационным объемом текста. Такой подход к измерению информации еще называют *объемным подходом*.

Представляя ученикам более крупные единицы измерения информации: килобайт, мегабайт, гигабайт, следует обратить их внимание на то, что мы привыкли приставку «кило» воспринимать, как увеличение в 1000 раз. В информатике это не так. Килобайт больше байта в 1024 раза, а число $1024 = 2^{10}$. Так же относится

и «мега» по отношению к «кило» и так далее. Тем не менее, часто при приближенных вычислениях используют коэффициент 1000.

Методические рекомендации по решению задач

В условиях задач по обсуждаемой теме связываются между собой следующие величины: мощность символьного алфавита — N ; информационный вес символа — i ; число символов в тексте (объем текста) — K ; количество информации, заключенное в тексте (информационный объем текста) — I . Кроме того, при решении задач требуется знать связь между различными единицами информации: бит, байт, Кбайт, Мбайт.

Задачи, соответствующие уровню минимального содержания базового курса, рассматривают лишь приближение равновероятного алфавита. В задачах для углубленного уровня обучения используется более реальное предположение о неравновероятности появления символов. В таком случае, появляется еще один параметр — вероятность символа (p).

Для решения задач полезно, чтобы ученики записали в тетради таблицу значений целых степеней двойки: от 2^3 до 2^{10} .

Пример 1.

[5]: задание 7 к § 4. *Два текста содержат одинаковое количество символов. Первый текст составлен в алфавите мощностью 32 символа, второй — мощностью 64 символа. Во сколько раз отличается количество информации в этих текстах?*

В равновероятном приближении информационный объем текста равен произведению числа символов на информационный вес одного символа:

$$I = K \times i.$$

Поскольку оба текста имеют одинаковое число символов (K), то различие информационных объемов определяется только разницей в информативности сим-

волов алфавита (i). Найдем i_1 для первого алфавита и i_2 для второго алфавита:

$$2^{i_1} = 32; \text{ отсюда } i_1 = 5 \text{ битов};$$

$$2^{i_2} = 64; \text{ отсюда } i_2 = 6 \text{ битов}.$$

Следовательно, информационные объемы первого и второго текстов:

$$I_1 = K \times 5 \text{ битов}; \quad I_2 = K \times 6 \text{ битов}.$$

Отсюда следует, что количество информации во втором тексте больше, чем в первом в $6/5$ раз или в 1,2 раза.

Пример 2.

Задачник-практикум: раздел 1.3, задача № 23.
Объем сообщения, содержащего 2048 символов, составил 1/512 часть мегабайта. Каков размер алфавита, с помощью которого записано сообщение?

Переведем информационный объем сообщения из мегабайтов в биты. Для этого данную величину умножим дважды на 1024 (получим байты) и один раз — на 8:

$$I = 1/512 \times 1024 \times 1024 \times 8 = 16\,384 \text{ битов}.$$

Поскольку такой объем информации несут 2048 символа (K), то на один символ приходится:

$$i = I/K = 16\,384 / 2048 = 8 \text{ битов}.$$

Отсюда следует, что размер (мощность) использованного алфавита равен $2^8 = 256$ символов.

Пример 3.

Задачник-практикум: раздел 1.3, задача № 29. Для записи сообщения используется 64-символьный алфавит. Каждая страница содержит 30 строк. Все сообщение содержит 8775 байтов информации и занимает 6 страниц. Сколько символов в строке?

Обозначим: X — число символов в одной строке. Тогда число символов во всем тексте представится выражением:

$$6 \times 30 \times X = 180 \times X.$$

Зная мощность алфавита ($N = 64$), определим информационный вес символа:

$$2^i = 64; \text{ отсюда } i = 6 \text{ битов.}$$

Теперь запишем уравнение, в котором слева стоит произведение полного числа символов в тексте на информационный вес одного символа, а справа — информационный объем всего текста, переведенный в биты:

$$180 \times X \times 6 = 8775 \times 8.$$

Отсюда:

$$X = (8775 \times 8) / (180 \times 6) = 65.$$

Ответ: в одной строке текста содержится 65 символов.

Пример 4.

Задачник-практикум: раздел 1.3, задача № 34. *Выяснить, сколько битов информации несет каждое двузначное целое число (отвлекаясь от его конкретного числового значения).*

Эта задача относится к углубленному уровню изучения темы. К решению задач такого типа ученики могут подойти по-разному и в итоге получить разные результаты. Важно, чтобы каждый вариант решения был бы учеником хорошо обоснован. Если эту задачу учитель предполагает использовать как демонстрационную, то рекомендуется следующий подход.

Замечание, записанное в условии задачи в скобках, обозначает, что двузначное число рассматривается лишь как последовательность двух символов-цифр. Первая цифра может быть любой из диапазона от 1 до 9 (для многозначных целых чисел впереди нули не пи-

шутся); вторая — от 0 до 9. Следовательно, для записи цифр используются разные алфавиты. Информационный вес первой цифры находится из уравнения:

$$2^{i_1} = 9; \text{ отсюда } i_1 = 3,16993 \text{ бита.}$$

Результат получен из таблицы решений данного типа показательного уравнения, имеющейся как в учебнике, так и в задачнике. Для второй цифры аналогично имеем:

$$2^{i_2} = 10; \text{ отсюда } i_2 = 3,32193 \text{ бита.}$$

Отсюда суммарное количество информации равно 6,49186 бита. Результат получен с точностью до пяти знаков после запятой.

Другой подход, который могут предложить ученики, следующий. Будем рассматривать представление двузначного целого числа в компьютерном символьном алфавите. Все десятичные цифры входят в этот алфавит. Тогда каждый символ несет 1 байт информации. Следовательно, двузначное целое число несет 2 байта, или 16 битов информации.

Если ученик предложит такой вариант решения, его не следует отвергать как неверный. С «позиции компьютера» это решение верное. Однако, желательно показать ученикам первый вариант, более строгий с математической (теоретической) точки зрения.

Пример 5.

[5]: часть II, дополнение к главе 1, задание 16. *Определить свою скорость восприятия информации при чтении вслух и про себя.*

Данное задание носит творческий характер. Ученик должен сам спланировать эксперимент. План может быть следующим: взять книгу и выбрать в ней страницу, заполненную текстом. Желательно, чтобы этот текст был новым для ученика, но понятным, то есть информативным. Подсчитать число символов в тексте. Для этого нужно определить среднее число сим-

волов в строке, число строк на странице. Умножив эти два числа, получим число символов во всем тексте. Разумно допустить, что для набора текста книги использован компьютерный алфавит, мощность которого равна 256. Следовательно, каждый символ несет 1 байт информации. Таким образом, общее число символов равно информационному объему текста в байтах. Далее нужно читать текст вслух, измеряя по секундомеру время чтения. Скорость чтения должна быть такой, чтобы ученику было понятно содержание текста. Проверить это можно, попытавшись пересказать прочитанное. Если ученик ничего не запомнил, значит он не воспринял информацию, и скорость чтения следует уменьшить. Окончательный ответ получается путем деления объема информации на время в секундах.

Например, выберем для чтения стр. 29 из учебника. На странице 40 строк; в каждой строке в среднем по 50 символов (пробелы тоже нужно считать). Следовательно, на странице 2000 символов и информационный объем текста равен 2000 байтов. Время чтения вслух — 140 секунд. Значит, скорость восприятия информации при чтении вслух равна $2000/140 = 14,3$ байт/сек.

Повторение такого же эксперимента с чтением «про себя» может дать более высокий результат.

Полезно обратить внимание учеников на то, что для более точной оценки средней скорости чтения желательно брать текст большего размера. Различные фрагменты текста могут оказаться разными по степени сложности восприятия. Чем текст больше, тем результат ближе к объективному среднестатистическому.

4. Содержательный подход к измерению информации

(тема для углубленного изучения)

Разделы учебника: Часть II, разделы 1.1, 1.2.

Основные цели. Дать учащимся представление о подходе к измерению информации с содержательной точки зрения. Ввести единицу измерения информации — бит. Научить детей вычислять количество информации в частном случае сообщения о событии с известной вероятностью (из данного конечного множества).

Изучаемые вопросы.

- Бит как единица уменьшения неопределенности знаний.
- Количество информации в сообщении об одном из N равновероятных событий.

Методические рекомендации по объяснению теоретического материала

1. С содержательной точки зрения количество информации, заключенное в сообщении, связано с тем, насколько это сообщение уменьшает неопределенность знаний принимающего его человека. С этой позиции определение бита — единицы измерения информации — может оказаться сложным для понимания. В этом определении содержится незнакомое детям понятие «неопределенность знаний». Прежде всего, нужно раскрыть его. Учитель должен хорошо понимать, что речь идет об очень частном случае: о сообщении, которое содержит сведения о том, что произошло одно из *конечного множества (N) возможных событий*. Например, это сообщение о результате бросания монеты, игрового кубика; вытаскивания экзаменационного билета и т. п. Неопределенность знания о результате некоторого события — это число возможных вариантов результата. Для монеты число вариантов равно 2, для кубика — 6, для билетов — 30 (если на столе лежало 30 билетов).

2. Еще одной сложностью является понятие *равновероятности*. Здесь следует оттолкнуться от интуитивного представления детей, подкрепив его примерами. *События равновероятны, если ни одно из них не имеет преимущества перед другими*. С этой точки зрения выпадения орла и решки равновероятны; выпадения шести граней кубика равновероятны. Полезно привести примеры и *неравновероятных* событий. Например, в сообщении о погоде в зависимости от сезона сведения о том, что будет дождь или снег, могут иметь разную вероятность. Летом наиболее вероятно сообщение о дожде, зимой — о снеге, а в переходный период (в марте или ноябре) они могут оказаться равновероятными. Понятие «более вероятное событие» можно пояснить через родственные понятия: более ожидаемое, происходящее чаще в данных условиях. В рамках базового курса не ставится задача понимания учениками строгого определения вероятности, умения вычислять вероятность. Но представление о равновероятных и неравновероятных событиях должно быть ими получено. Ученики должны научиться приводить примеры равновероятных и неравновероятных событий.

При наличии учебного времени полезно обсудить с учениками понятий «*достоверное событие*» (событие, которое обязательно происходит) и «*невозможное событие*». От этих понятий можно оттолкнуться, чтобы ввести интуитивное представление о мере вероятности. Достаточно сообщить, что вероятность достоверного события равна 1, а невозможного — 0. Это крайние значения. Значит, во всех других промежуточных случаях значение вероятности лежит между нулем и единицей. В частности, вероятность каждого из двух равновероятных событий равна $1/2$. В классах с углубленным изучением информатики можно рассмотреть раздел 1.2 «Вероятность и информация» части II учебника.

3. В учебнике дано следующее определение единицы информации: «*Сообщение, уменьшающее неопределенность знаний в 2 раза, несет 1 бит информации*».

Немного дальше приводится определение для частного случая: «Сообщение о том, что произошло одно событие из двух равновероятных, несет 1 бит информации». Учитель, предпочитающий индуктивный метод объяснения, может начать со второго определения. Обсуждая традиционный пример с монетой (орел-решка), следует отметить, что получение сообщения о результате бросания монеты уменьшило неопределенность знаний в два раза: перед подбрасыванием монеты было два равновероятных варианта, после получения сообщения о результате остался один единственный. Далее следует сказать, что и для всех других случаев сообщений о равновероятных событиях при уменьшении неопределенности знаний в два раза передается 1 бит информации. Примеры, приведенные в учебнике, учитель может дополнить другими, также хорошо предложить ученикам придумать свои примеры. Индуктивно, от частных примеров учитель вместе с классом приходит к обобщенной формуле: $2^i = N$. Здесь N — число вариантов равновероятных событий (неопределенность знаний), а i — количество информации в сообщении о том, что произошло одно из N событий.

Если N известно, а i является неизвестной величиной, то данная формула превращается в показательное уравнение. Как известно, показательное уравнение решается с помощью функции логарифма: $i = \log_2 N$. Здесь учителю предоставляются два возможных пути: либо с опережением уроков математики объяснить, что такое логарифм, либо «не связываться» с логарифмами. Во втором варианте следует рассмотреть с учениками решение уравнения для частных случаев, когда N есть целая степень двойки: 2, 4, 8, 16, 32 и так далее. Объяснение происходит по схеме:

Если $N = 2 = 2^1$, то уравнение принимает вид: $2^i = 2^1$, отсюда $i = 1$.

Если $N = 4 = 2^2$, то уравнение принимает вид: $2^i = 2^2$, отсюда $i = 2$.

Если $N = 8 = 2^3$, то уравнение принимает вид: $2^i = 2^3$, отсюда $i = 3$ и так далее.

В общем случае, если $N = 2^k$, где k — целое число, то уравнение принимает вид $2^i = 2^k$ и, следовательно, $i = k$. Ученикам полезно запомнить ряд целых степеней двойки хотя бы до $2^{10} = 1024$. С этими величинами им предстоит еще встретиться в других разделах.

Для тех значений N , которые не являются целыми степенями двойки, решение уравнения $2^i = N$ можно получать из приведенной в учебнике таблицы 1.1. Со всем не обязательно говорить ученикам, что это таблица логарифмов по основанию 2. Например, желая определить, сколько же битов информации несет сообщение о результате бросания шестигранного кубика, нужно решать уравнение: $2^i = 6$. Поскольку $2^2 < 6 < 2^3$, то следует пояснить ученикам, что $2 < i < 3$. Заглянув в таблицу, узнаем (с точностью до пяти знаков после запятой), что $i = 2,58496$ бита.

Методические рекомендации по решению задач

Задачи по теме раздела 1.1 связаны с использованием уравнения $2^i = N$. Возможны два варианта условий задач: 1) дано N , найти i ; 2) дано i , найти N .

В случаях, когда N равно целой степени двойки, желательно, чтобы ученики выполняли вычисления «в уме». Как уже говорилось выше, полезно запомнить ряд целых степеней числа 2 хотя бы до 2^{10} . В противном случае следует использовать таблицу 1.1, где рассматриваются значения N от 1 до 64.

Для основного уровня изучения базового курса предлагаются задачи, связанные с сообщениями о равновероятных событиях. Ученики должны это понимать и обязательно качественно обосновывать, используя термин «равновероятные события».

Пример 1.

[5]: часть II, раздел 1.1, задание № 4. *Сколько битов информации несет сообщение о том, что из колоды в 32 карты достали даму пик?*

Решение этой задачи следует описывать так: при случайном вытаскивании карт из перемешанной колоды ни одна из карт не имеет преимущества быть выбранной по сравнению с другими. Следовательно, случайный выбор любой карты, в том числе и дамы пик, — события равновероятные. Отсюда следует, что неопределенность знаний о результате вытаскивания карты равна 32 — числу карт в колоде. Если i — количество информации в сообщении о результате вытаскивания одной карты (дамы пик), то имеем уравнение:

$$2^i = 32.$$

Поскольку $32 = 2^5$, то, следовательно, $i = 5$ битов.

На тему данной задачи учитель может предложить еще несколько заданий. Например: Сколько информации несет сообщение о том, что из колоды карт достали карту красной масти? (1 бит, так как красных и черных карт одинаковое количество.)

Сколько информации несет сообщение о том, что из колоды карт достали карту бубновой масти? (2 бита, так как всего в колоде 4 масти, и количества карт в них равные.)

Пример 2.

[5]: часть II, раздел 1.1, задание № 5. *Проводится две лотереи: «4 из 32» и «5 из 64». Сообщение о результатах какой из лотерей несет больше информации?*

У этой задачи есть «подводный камень», на который может натолкнуться учитель. Первый путь решения тривиальный: вытаскивание номеров из лотерейного барабана — события равновероятные. Поэтому в первой лотерее количество информации в сообщении об одном номере равно 5 битов ($2^5 = 32$), а во второй —

6 битов ($2^6 = 64$). Сообщение о номерах в первой лотерее несет $5 \times 4 = 20$ битов. Сообщение о пяти номерах второй лотереи несет $6 \times 5 = 30$ битов. Следовательно, сообщение о результатах второй лотереи несет больше информации, чем сообщение о результатах первой.

Но возможен и другой путь рассуждения. Представьте себе, что вы наблюдаете за розыгрышем лотереи. Выбор первого шара производится из 32-х шаров в барабане. Результат несет 5 битов информации. Но второй шар будет выбираться уже из 31 номера, третий — из 30 номеров, четвертый — из 29. Значит, количество информации, которое несет второй номер, находится из уравнения: $2^i = 31$. Глядя в таблицу 1.1, находим: $i = 4,95420$ бита. Для третьего номера: $2^i = 30$; $i = 4,90689$ бита. Для четвертого номера: $2^i = 29$; $i = 4,85798$ бита. В сумме получаем: $5 + 4,95420 + 4,90689 + 4,85798 = 19,71907$ бита. Аналогично и для второй лотереи. Конечно, на окончательном выводе такие подсчеты не отразятся. Можно было вообще, ничего не вычисляя, сразу ответить, что второе сообщение несет больше информации, чем первое. Но здесь интересен сам путь вычислений с учетом «выбывания участников».

Последовательные события в этом случае не являются независимыми друг от друга (кроме первого). Это, как мы увидели, отражается в различии информативности сообщений о каждом из них. Первый (тривиальный) вариант решения задачи получен в предположении независимости событий и является в таком случае неточным.

Пример 3.

Задачник-практикум: раздел 1.4, задача № 1. В корзине лежит 8 черных шаров и 24 белых. Сколько информации несет сообщение о том, что достали черный шар?

Эта задача для варианта углубленного изучения базового курса. Для ее решения надо знать о связи между

вероятностью события и информативностью сообщения об этом событии (см. [5], часть II, раздел 1.2).

Из условия задачи следует, что всего в корзине 32 шара. Поскольку черных шаров меньше, чем белых, то вероятность достать черный шар меньше вероятности доставания белого шара. Численная мера вероятности равна доле шаров одного цвета в общем количестве шаров. Вероятность доставания черного шара равна $p_4 = 8/32 = 1/4$; вероятность доставания белого шара равна $p_6 = 24/32 = 3/4$. Тогда количество информации в сообщении о том, что выпал черный шар, нужно вычислять так:

$$I_4 = \log_2(1/p_4) = \log_2 4 = 2 \text{ бита.}$$

Количество информации в сообщении о белом шаре:

$$\begin{aligned} I_6 &= \log_2(1/p_6) = \log_2(4/3) = \log_2 4 - \log_2 3 = \\ &= 2 - 1,58496 = 0,41504 \text{ бита.} \end{aligned}$$

Здесь для вычисления логарифмов можно использовать таблицы, приведенные как в учебнике, так и в задачнике.

Пример 4.

Задачник-практикум: раздел 1.4, задача № 7. В течение четверти ученик получил 100 оценок. Сообщение о том, что он получил четверку, несет 2 бита информации. Сколько четверок ученик получил за четверть?

Обозначим вероятность получения учеником четверки p_4 . Эта величина равна доле числа четверок (n_4) в общем числе оценок: $p_4 = n_4/100$. Связь между этой величиной и количеством информации в сообщении о получении четверки выражается формулой:

$$2^i = 1/p_4 = 100/n_4.$$

Но поскольку $i = 2$ бита, имеем:

$$2^2 = 100/n_4; \quad 4 = 100/n_4; \quad n_4 = 100/4 = 25.$$

Таким образом, получен следующий ответ: в течение четверти ученик получил 25 четверок.

Заметим, что данный результат мог бы быть получен и без вычислений, путем следующих рассуждений: 2 бита информации несет сообщение об одном из четырех равновероятных событий ($2^2=4$). Отсюда следует, что четверки составляют $1/4$ часть от общего числа оценок. Следовательно, ученик получил 25 четверок.

5. История информатики

Разделы учебника: часть I, § 5, 6, 50. Дополнительный материал: часть II, разделы 10.2, 12.9.

Основные цели. Познакомить учащихся с основными событиями, открытиями, изобретениями, связанными с развитием информатики как в период до появления компьютеров, так и в компьютерную эпоху.

Изучаемые вопросы.

- ☐ История развития средств хранения, передачи и обработки информации.
- ☐ История чисел.
- ☐ История ЭВМ.
- ☐ История языков программирования.

Общие методические рекомендации

В любом общеобразовательном школьном предмете присутствует историческая линия. Нет и не должно быть учебной дисциплины без дат, событий, имен. Знание истории предмета помогает сформировать в сознании учеников целостное представление о его содержании. Невозможно, например, изучать физику и не знать об открытиях Ньютона, Фарадея, Резерфорда, Бора, Эйнштейна; об изобретениях Эдисона, Попова, Вильсона и др. История предмета — это «драма идей» и логика его развития.

Школьная информатика пока еще формируется как общеобразовательный предмет. Она ищет свое фундаментальное содержание. И в этом содержании должно быть место истории.

В учебнике, в разных его разделах, присутствуют следующие исторические темы: история средств докомпьютерной информатики (в том числе история чисел); история ЭВМ, история операционных систем для персональных компьютеров, история языков программирования. Логика раскрытия исторической линии курса следующая.

1. Информационная деятельность человека раскладывается на три составляющие, три типа информационных процессов: хранение информации, передача информации и обработка информации. Первоначально орудия и средства информационной деятельности человека развивались отдельно по каждому из этих трех направлений. В области хранения информации раскрывается история информационных носителей: от камня и кости до современных магнитных и оптических дисков. Важнейшими моментами этой истории было изобретение бумаги и письма на бумаге (Китай, II век н. э.), а также изобретение книгопечатания в XV веке в Европе. Эти события способствовали массовому распространению грамотности, а под грамотностью понимается умение читать, писать, считать, то есть работать с информацией. Последствия изобретения книгопечатания можно назвать первой информационной революцией в истории цивилизации. Причиной второй информационной революции стало изобретение компьютера в XX веке.

История развития средств передачи информации прослеживается от первых почтовых контактов до современных средств спутниковой связи. Учителю необходимо обратить внимание учеников на следующий момент: в настоящее время много говорится об информационном обществе, о том, что информация становится важнейшим ресурсом в жизни человечества. В

информационном обществе происходят интенсивные процессы обмена информацией между его членами. Такое общество постепенно складывается в течение последних тысячелетий с момента появления первого средства дальней связи — почты. Обмен информацией на расстоянии сейчас принято называть телекоммуникациями. Здесь тоже можно говорить о двух революционных эпохах в развитии средств связи. Первый этап: XIX век, изобретение средств электросвязи (телефон, телеграф); радиосвязи и телевидения (первая половина XX века). Второй этап: появление и распространение компьютерных телекоммуникаций во второй половине XX века. Многие физические открытия и технические изобретения ориентировались на развитие средств передачи информации. Все это подчеркивает важность для человеческого общества процессов информационного обмена.

Из всех видов обработки информации наиболее сложными являются математические вычисления. Именно в этой области приложено немало усилий для изобретения средств, облегчающих работу человека. В учебнике прослеживаются две линии в истории: история представления чисел и систем счисления (§ 6) и история изобретений средств механизации и автоматизации вычислений (§ 5). Одно с другим тесно связано, поскольку только после распространения десятичной позиционной системы счисления стали активно изобретаться вычислительные механизмы: десятичные счеты, счетная машина Паскаля и арифмометр, современный калькулятор. Следует подчеркнуть значимость работы Ч. Бэббиджа над проектом Аналитической машины. Несмотря на то, что в окончательном виде такая машина не была построена, заслуга Бэббиджа состоит в том, что он первым разработал проект программно управляемого вычислительного автомата и заложил ряд идей, впоследствии использованных изобретателями электронно-вычислительной машины.

2. История электронно-вычислительных машин описана в конце части I учебника, в § 50. По своему усмотрению учитель может использовать этот материал на разных этапах изучения курса. Нередко об истории ЭВМ рассказывается в самом начале, на первых уроках. Такое вполне возможно. Увлекательный рассказ о развитии компьютерной техники может поднять интерес учеников к предмету. Однако есть существенные аргументы за то, чтобы историю ЭВМ изучать в конце курса. Дело в том, что понимание этого материала требует от учеников определенной компьютерной грамотности. Если ученики еще ничего не знают о памяти и процессоре ЭВМ, о программировании, о круге задач, решаемых с помощью ЭВМ, то вряд ли они смогут понять рассказ о том, как развивались эти средства и возможности компьютера. Получив на уроках представление об устройстве и областях применения современного компьютера, ученики получают «точку отсчета», относительно которой они смогут оценивать технику предыдущих поколений.

Возможен и компромиссный вариант. Историческую информацию учитель дает «порциями», вводя ее понемногу в различные разделы курса. Например, на первом вводном уроке можно рассказать о том, что первые ЭВМ появились в конце 40-х годов XX века, применялись исключительно для математических расчетов и их производительность составляла около 1000 арифметических операций в секунду. Современные ЭВМ умеют работать с любой информацией: числами, текстами, графическими изображениями, звуком; а их быстродействие составляет миллионы и даже миллиарды операций в секунду. Рассказывая об устройстве персонального компьютера, о микропроцессоре, который уместается на ладони, можно сообщить, что на первых ламповых машинах подобное устройство было очень громоздким и занимало несколько шкафов. Знакомя учеников с современными устройствами ввода/вывода

информации (клавиатурой, монитором), можно рассказать о том, что раньше для ввода информации использовались перфорационные носители — перфоленты, перфокарты. Если есть возможность, то хорошо показать образцы таких перфоносителей.

Изучая тему «Первое знакомство с компьютером», ученики обязательно должны услышать имя Джона фон Неймана, узнать о значении его работ для развития компьютерной техники. Подытожить тему истории ЭВМ можно в конце курса, ознакомив учеников с содержанием § 50, в котором описывается историческое деление компьютерной техники на поколения, дается целостная картина ее эволюции. Вопросы истории компьютерной техники могут быть предметом реферативных работ учащихся. На эту тему существует разнообразная литература. Одна из таких книг: [21].

3. Современный компьютер — это единство аппаратных средств и программного обеспечения. Такое представление должно формироваться в сознании учеников в ходе изучения базового курса. Историческая линия курса должна отражать эволюцию обеих этих составляющих. Основой программного обеспечения современного компьютера является операционная система. Операционная система играет роль платформы, на которой функционируют все виды программ. Смена этой платформы ведет к качественному изменению прикладных программ, технологий программирования, пользовательского интерфейса. В разделе 10.2 части II учебника рассказывается о развитии операционных систем для персональных компьютеров.

Тема «Введение в программирование» может также изучаться в историческом ключе. Фактически эта тема начинается с главы 10. В этой главе объясняется, что такое команда процессора, программа на языке машинных команд. Необходимый состав системы команд процессора был описан Дж. фон Нейманом в 1946 году. Следует рассказать ученикам, что программирование

на первых ЭВМ производилось исключительно в машинных командах.

В главе 12 рассказывается об историческом развитии средств программирования: языкам машинных команд пришли на смену автокоды (ассемблеры), затем появились языки высокого уровня. В разделе 12.9 описывается история языков программирования высокого уровня. Ученикам полезно знать, что понятие «программное обеспечение» берет начало от трансляторов с языков программирования.

6. Системы счисления

Разделы учебника: часть I, § 6, 45, 46.

Основные цели. Раскрыть понятие системы счисления. Познакомить учеников со способами представления чисел в позиционных системах счисления. Дать представление об использовании двоичной системы в компьютере.

Изучаемые вопросы

- Позиционные и непозиционные системы счисления.
- ▷ Основные понятия позиционных систем: основание, алфавит.
- Развернутая форма представления чисел в позиционных системах.
- Перевод чисел из одной системы в другую.
- Особенности двоичной арифметики.
- Связь между двоичной и шестнадцатеричной системами.

Методические рекомендации по изложению теоретического материала

Тема «Системы счисления» имеет прямое отношение к математической теории чисел. Однако в школь-

ном курсе математики она, как правило, не изучается. Необходимость изучения этой темы в курсе информатики связана с тем фактом, что числа в памяти компьютера представлены в двоичной системе счисления, а для внешнего представления содержимого памяти, адресов памяти используют шестнадцатеричную или восьмеричную системы. Это одна из традиционных тем, которая присутствует в любом варианте курса информатики или программирования. Являясь смежной с математикой, данная тема вносит вклад также и в фундаментальное математическое образование школьников. Подробный и систематизированный материал по этой теме содержится в пособии [22].

1. Ученики 7 или 8 класса, безусловно, знакомы с записью чисел как римскими, так и арабскими цифрами. Они привыкли видеть римские цифры в обозначении глав в книге, в указании столетий (XX век) и в некоторых других нумерациях. Математические расчеты они всегда производили в арабской системе чисел. В данной теме учителю предстоит раскрыть перед учениками эти, казалось бы знакомые вещи, с новой стороны. С методической точки зрения бывает очень эффективным прием, когда учитель подводит учеников к самостоятельному, пусть маленькому, открытию. В данном случае желательно, чтобы ученики сами подошли к формулировке различия между позиционным и непозиционным принципами записи чисел. Сделать это можно, отталкиваясь от конкретного примера. Напишите на доске два числа:

XXX

333

Первое — римское «тридцать», второе — арабское «триста тридцать три». И задайте вопрос: «Чем отличаются принципы записи многозначных чисел римскими и арабскими цифрами?» Скорее всего, вы сразу не услышите тот ответ, который бы хотели получить. Тогда, указывая на отдельные цифры римского числа, спрашивайте:

— Что (какое количество) обозначает эта цифра?

Получите ответ:

— Десять.

— А эта цифра?

— Десять.

— А эта?

— Десять.

— Как получается значение данного трехзначного числа?

— Десять прибавить десять, прибавить десять, получается тридцать.

А теперь переходим к числу 333. Снова задаем вопросы:

— Какое количество в записи числа обозначает первая цифра справа?

— Три единицы.

— А вторая цифра?

— Три десятка.

— А третья цифра?

— Три сотни.

— А как получается общее значение числа?

— К трем единицам прибавить три десятка и прибавить три сотни — получится триста тридцать три.

Из этого диалога следуют все правила, которые учитель должен сообщить детям. В римском способе записи чисел значение, которое несет каждая цифра в числе, не зависит от позиции этой цифры. В арабском же способе значение, которое несет каждая цифра в записи числа, зависит не только от того, какая это цифра, но и от позиции, которую она занимает в числе. Сделав ударение на слове «позиция», учитель сообщает, что римский способ записи чисел называется непозиционным, а арабский — позиционным.

После этого можно ввести термин «система счисления». Система счисления — это определенный способ представления чисел и соответствующие ему правила действия над числами. Римский способ записи чисел

является примером *непозиционной системы счисления*, а арабский — это *позиционная система счисления*.

Следует подчеркнуть связь между способом записи чисел и приемами арифметических вычислений в соответствующей системе счисления. Предложите ученикам выполнить умножение, например, числа сто тридцать четыре на семьдесят шесть, используя римскую и арабскую системы счисления! С арабскими числами они легко справятся, а также смогут убедиться, что римские цифры — не помощники в вычислениях. В римской системе нет простых и понятных правил выполнения вычислений с многозначными числами. Для арабской системы такие правила известны еще с IX века. В этой теме полезно рассказать ученикам, что правила выполнения вычислений с многозначными числами были разработаны выдающимся математиком средневекового Востока Мухамедом аль-Хорезми и в Европе были названы алгоритмами (от латинского написания имени аль-Хорезми — *Algorithmi*). Этот факт следует напомнить позже, при изучении алгоритмизации. Итак, именно позиционные системы счисления стали основой современной математики. Далее, как и в математике, в информатике мы будем иметь дело только с числами в позиционных системах счисления.

2. Теперь нужно дать понять ученикам, что позиционных систем счисления существует множество, и отличаются они друг от друга *алфавитом* — множеством используемых цифр. Размер алфавита (число цифр) называется *основанием системы счисления*. Задайте вопрос: «Почему арабская система называется *десятичной системой счисления*?» Наверняка услышите в ответ про десять цифр в алфавите. Делаем вывод: *основание арабской системы счисления равно десяти, поэтому она называется десятичной*.

Следует показать алфавиты различных позиционных систем счисления. В системах с основаниями, не большими 10 используются только арабские цифры.

Если же основание больше 10, то в роли цифр выступают также латинские буквы в алфавитном порядке. Из таких систем в дальнейшем будет рассматриваться лишь шестнадцатеричная система.

Далее нужно научить учеников записывать натуральный ряд чисел в различных позиционных системах. Объяснение следует проводить на примере десятичной системы, для которой вид натурального ряда чисел им хорошо известен:

1 2 3 4 5 6 7 8 9 10 11 ... 19 20 ... 99 100 101 ...

Принцип построения ряда такой: сначала в порядке возрастания значений записываются все однозначные числа; первое двузначное число — всегда 10 (у многозначных целых чисел 0 впереди не является значащей цифрой и обычно не пишется). Далее следуют все двузначные сочетания единицы с другими цифрами; затем — двузначные числа, начинающиеся с 2, затем — с 3 и так далее. Самое большое двузначное число — 99. Затем идут трехзначные числа, начиная от 100 до 999 и так далее.

По такому же принципу строится натуральный ряд и в других системах счисления. Например, в четверичной системе (с основанием 4):

1 2 3 10 11 12 13 20 21 22 23 30 31 32 33
100 101 102 103 110 111 ... 333 1000 ...

Аналогично и для других систем. Наибольший интерес представляет натуральный ряд двоичных чисел. Вот как он выглядит:

1 10 11 100 101 110 111 1000 1001 1010 1011
1100 1101 1110 1111 10000

и так далее. Следует обратить внимание учеников на быстрый рост количества цифр в числе.

Для указания на основание системы, к которой относится число, вводим индексное обозначение. Например, 36_8 указывает на то, что это число в восьмеричной

системе счисления, $1A6_{16}$ — шестнадцатеричное число, 1011_2 — число в двоичной системе. Индекс всегда записывается десятичным числом. Следует подчеркнуть то, что в любой системе счисления ее основание записывается как 10.

Еще одно важное замечание: ни в коем случае нельзя называть недесятичные числа так же, как десятичные. Например, нельзя называть восьмеричное число 36_8 как тридцать шесть! Надо говорить: «Три—шесть». Или, нельзя читать 101_2 как «сто один». Надо говорить «один—ноль—один». Следует также понимать, что, например, $0,1_2$ — это не одна десятая, а одна вторая, или $0,1_8$ — это одна восьмая и т. п.

3. Сущность позиционного представления чисел отражается в *развернутой форме записи чисел*. Снова для объяснения привлекаем десятичную систему. Пример:

$$\begin{aligned} 5319,12 &= 5000 + 300 + 10 + 9 + 0,1 + 0,02 = \\ &= 5 \times 10^3 + 3 \times 10^2 + 1 \times 10^1 + 9 + 1 \times 10^{-1} + \\ &\quad + 2 \times 10^{-2}. \end{aligned}$$

Последнее выражение и называется *развернутой формой записи числа*. Слагаемые в этом выражении являются произведениями значащих цифр числа на степени десятки (основания системы счисления), зависящие от позиции цифры в числе — разряда. Цифры в целой части умножаются на положительные степени числа 10, а цифры в дробной части — на отрицательные степени. Показатель степени является номером соответствующего разряда. Аналогично можно получить развернутую форму чисел в других системах счисления. Например, для восьмеричного числа:

$$1753_8 = 1 \times 10^3 + 7 \times 10^2 + 5 \times 10^1 + 3.$$

Здесь $10_8 = 8_{10}$.

На этом заканчивается материал главы 1, посвященный системам счисления.

4. Способы перевода чисел из одной системы счисления в другую рассматриваются в главе 10 учебника. Основная идея заключается в следующем: перевод чисел неизбежно связан с выполнением вычислений. *Поскольку нам хорошо знакомы лишь десятичная арифметика, то любой перевод следует свести к выполнению вычислений над десятичными числами.*

Объяснение способов перевода следует начать с перевода в десятичную систему чисел из других систем счисления. Делается это просто: нужно перейти к записи развернутой формы числа в десятичной системе. Вот пример такого перехода для приведенного выше восьмеричного числа:

$$\begin{aligned} 1753_8 &= (1 \times 10^3 + 7 \times 10^2 + 5 \times 10^1 + 3)_8 = \\ &= (1 \times 8^3 + 7 \times 8^2 + 5 \times 8^1 + 3)_{10}. \end{aligned}$$

Теперь нужно вычислить полученное выражение по правилам десятичной арифметики и получить окончательный результат:

$$1753_8 = (512 + 448 + 40 + 3)_{10} = 1003_{10}.$$

Чаще всего развернутую форму числа сразу записывают в десятичной системе. Вот еще пример с двоичным числом:

$$\begin{aligned} 101101,1_2 &= (1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + \\ &+ 0 \times 2^1 + 1 + 1 \times 2^{-1})_{10} = 32 + 8 + 4 + 1 + \\ &+ 0,5 = 45,5_{10}. \end{aligned}$$

Для вычисления значения числа по его развернутой форме записи существует удобный прием, который называется *вычислительной схемой Горнера*. Суть его состоит в том, что *развернутая запись числа преобразуется в эквивалентную форму с вложенными скобками*. Например, для рассмотренного выше восьмеричного числа это выглядит так:

$$\begin{aligned} 1753_8 &= (1 \times 8^3 + 7 \times 8^2 + 5 \times 8^1 + 3)_{10} = \\ &= ((1 \times 8 + 7) \times 8 + 5) \times 8 + 3. \end{aligned}$$

Нетрудно понять, что если раскрыть скобки, то получится то же самое выражение. В чем же удобство скобочной структуры? А в том, что ее вычисление производится путем выполнения последовательной цепочки операций умножения и сложения в порядке их записи слева направо. Для этого можно использовать самый простой калькулятор (без памяти), поскольку не требуется сохранять промежуточные результаты. *Схема Горнера сводит вычисление таких выражений к минимальному числу операций.*

Перевод десятичных чисел в другие системы счисления — задача более сложная. В принципе, все происходит через ту же самую развернутую форму записи числа. Только теперь нужно суметь десятичное число разложить в сумму по степеням нового основания $n \neq 10$. Например, число 85_{10} по степеням двойки раскладывается так:

$$85_{10} = 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 = 1010101_2.$$

Однако проделать это в уме довольно сложно. В § 6 описывается формальная процедура такого перевода. Здесь нужно обратить внимание на следующее обстоятельство: десятичные дроби с конечным числом цифр при переводе в другие системы могут превратиться в бесконечные дроби. Если удастся найти период, тогда его следует выделить. Если же период не обнаруживается, то нужно договориться о точности (то есть о количестве цифр), с которой производится перевод.

Если ставится цель получения при переводе дробного числа наиболее близкого значения, то, ограничивая число знаков, нужно производить округления. Для этого в процессе перевода следует вычислять на одну цифру больше, а затем, применяя правила округления, сокращать эту цифру. Выполняя округление, нужно соблюдать следующее правило: если первая отбрасываемая цифра больше или равна $n/2$ (n — основание системы), то к сохраняемому младшему разряду числа

прибавляется единица. Например, округление восьмеричного числа $32,32471_8$ до одного знака после запятой даст в результате $32,3$; а округление до двух знаков после запятой — $32,33$.

Математическая суть отмеченной выше проблемы связана со следующим фактом: *многие дробные рациональные десятичные числа в других системах счисления оказываются иррациональными.*

5. Применение двоичной системы счисления в ЭВМ может рассматриваться в двух аспектах: 1) двоичная нумерация; 2) двоичная арифметика, то есть выполнение арифметических вычислений над двоичными числами. С двоичной нумерацией ученики встретятся в § 14 «Тексты в компьютерной памяти». Рассказывая о таблице кодировки ASCII, учитель должен сообщить ученикам, что внутренний двоичный код символа — это его порядковый номер в двоичной системе счисления. О двоичной нумерации можно говорить и при изучении § 19 «Как кодируется изображение». Коды цветов восьмицветной палитры (табл. 4.1) и шестнадцатичной палитры (табл. 4.2) можно перевести в их десятичные номера. Такая система нумерации цветов используется в компьютерной графике. Например, красный цвет имеет номер 4 (двоичный код 100); коричневый — номер 6 (код 110); ярко-розовый — номер 13 (код 1101).

Практическая потребность знакомства с двоичной арифметикой возникает при изучении главы 10 «Как работает процессор ЭВМ». В этом разделе рассказывается, как процессор ЭВМ выполняет арифметические вычисления. Согласно принципу Дж. фон Неймана, компьютер производит вычисления в двоичной системе счисления. В рамках базового курса достаточно ограничиться рассмотрением вычислений с целыми двоичными числами.

Для выполнения вычислений с многозначными числами необходимо знать правила сложения и правила умножения однозначных чисел. Вот эти правила:

$$0 + 0 = 0 \quad 0 \times 0 = 0$$

$$1 + 0 = 1 \quad 1 \times 0 = 0$$

$$1 + 1 = 10 \quad 1 \times 1 = 1$$

Перестановочный закон сложения и умножения работает во всех системах счисления. Далее следует сообщить, что приемы выполнения вычислений с многозначными числами в двоичной системе аналогичны таковым в десятичной. Иначе говоря, процедуры сложения, вычитания и умножения «столбиком» и деления «уголком» в двоичной системе производятся так же, как и в десятичной. В § 46 приведены примеры сложения и умножения многозначных двоичных чисел. Рассмотрим правила вычитания и деления двоичных чисел.

Операция вычитания является обратной по отношению к сложению. Из приведенной выше таблицы сложения следуют правила вычитания:

$$0 - 0 = 0; \quad 1 - 0 = 1; \quad 10 - 1 = 1.$$

А вот пример вычитания многозначных чисел:

$$\begin{array}{r} 1001101101 \\ - 100110111 \\ \hline 100110110 \end{array}$$

Полученный результат можно проверить сложением разности с вычитаемым. Должно получиться уменьшаемое.

Деление — операция, обратная умножению. В любой системе счисления делить на 0 нельзя. Результат деления на 1 равен делимому. Деление двоичного числа на 10_2 ведет к перемещению запятой на один разряд влево, подобно десятичному делению на десять. Например:

$$10010/10 = 1001;$$

$$1011/10 = 101,1;$$

$$101100/10 = 10110.$$

Деление на 100 смещает запятую на 2 разряда влево и так далее. В базовом курсе можно не рассматривать сложные примеры деления многозначных двоичных чисел. Хотя способные ученики могут справиться и с ними, поняв общие принципы. В практическом задании 4 к § 46, варианты 7 и 8, результаты деления будут следующими:

$$1010/101 = 10; \quad 1110/100 = 11,1.$$

6. Представление информации, хранящейся в компьютерной памяти, в ее истинном двоичном виде весьма громоздко из-за большого количества цифр. Имеется в виду запись такой информации на бумаге или вывод ее на экран. Для этих целей принято использовать восьмеричную или шестнадцатеричную системы счисления. В § 46 рассматривается только шестнадцатеричная система. В современных ПК чаще всего используется именно эта система.

Существует простая связь между двоичным и шестнадцатеричным представлениями числа. При переводе числа из одной системы в другую одной шестнадцатеричной цифре соответствует четырехразрядный двоичный код. Это соответствие отражено в двоичношестнадцатеричной таблице, приведенной в учебнике (табл. 10.3). Такая связь основана на том, что $16 = 2^4$, и число различных 4-хразрядных комбинаций из цифр 0 и 1 равно 16: от 0000 до 1111. Поэтому перевод чисел из шестнадцатеричной системы в двоичную и обратно производится путем формальной перекодировки. Принято считать, что если дано шестнадцатеричное представление внутренней информации, то это равносильно наличию двоичного представления. Преимущество шестнадцатеричного представления состоит в том, что оно в 4 раза короче двоичного. Желательно, чтобы ученики запомнили двоично/шестнадцатеричную таб-

лицу. Тогда действительно для них шестнадцатеричное представление станет эквивалентным двоичному.

В шестнадцатеричном виде записываются адреса оперативной памяти компьютера. Например, для учебного компьютера «Нейман», который описывается в главе 10, диапазон адресации байтов памяти — от 00 до FF. Значит, в десятичной системе этот диапазон — от 0 до 255. Рассматривая структуру памяти компьютера, принципы адресации байтов памяти, можно обсудить с учениками следующий вопрос: как связан диапазон адресов с разрядностью адреса. В учебном компьютере «УК Нейман» адреса памяти представляются восьмиразрядными двоичными числами (двухразрядными шестнадцатеричными). Поэтому число различных адресов равно 2^8 , а диапазон значений — от 0 до $2^8 - 1 = 255$ (FF_{16}). Если адрес 16-разрядный, что часто имеет место для реальных ЭВМ, то размер адресуемой памяти равен

$$2^{16} \text{ байт} = 2^6 \text{ Кбайт} = 64 \text{ Кбайт}.$$

Диапазон шестнадцатеричных адресов в таком случае: 0000 до FFFF.

В современных компьютерах существуют приемы, позволяющие адресовать гораздо большие размеры памяти без увеличения разрядности адреса. Для этого используется многоуровневая структура организации памяти. Данный вопрос выходит за рамки содержания базового курса. Однако тема «Адресация памяти в современных ЭВМ» может быть предметом реферативной работы учащихся. Материал можно найти в специальной литературе, посвященной архитектуре современных ЭВМ.

Методические рекомендации по решению задач

Задачи на системы счисления содержатся в разделе 1.5 задачника-практикума. Первые семь задач относятся к непозиционным системам счисления. Во всех случаях предполагается справедливость принципа, используемого в римской системе записи чисел: если меньшая цифра записана после большей по величине

цифры, то ее значение прибавляется к общему значению числа, если меньшая цифра предшествует большей, то ее значение вычитается. Пример:

$$XVI = 10 + 5 + 1 = 16; XIV = 10 + 5 - 1 = 14.$$

Пример 1.

Задачник-практикум: раздел 1.5, задача № 6. В некоторой непозиционной системе счисления цифры обозначаются геометрическими фигурами. На основании представленных на рисунке примеров требуется определить, какую цифру обозначает каждая фигура, и расшифровать данное многозначное число.

Учитывая сделанное выше замечание, легко понять, что круг обозначает 1, треугольник углом вверх — 5, квадрат — 10, прямоугольник — 100, треугольник углом вниз — тысячу. Тогда значение записанного многозначного числа определится так:

$$1000 - 100 + 1000 - 10 + 100 + 5 + 1 + 1 = 1997.$$

Задачи №8–22 раздела 1.5 требуют от учеников знания алфавитов позиционных систем, умения представлять числа в развернутой форме, переводить недесятичные числа в десятичную систему счисления. При переводе рекомендуется использовать вычислительную схему Горнера.

Пример 2.

Задачник-практикум, раздел 1.5, задача № 16. Записать в десятичной системе данные недесятичные числа.

$$221_3 = (2 \times 3 + 2) \times 3 + 1 = 25_{10};$$

$$\begin{aligned} E41A_{16} &= ((14 \times 16 + 4) \times 16 + 1) \times 16 + 10 + \\ &+ (2/16 + 1)/16 = 58394 + 0,0703125 = \\ &= 58394,0703125_{10}. \end{aligned}$$

Обратите внимание учеников на то, что дробная часть числа переводится отдельно, и на то, как применение схемы Горнера модифицируется для дробной ча-

сти: умножение заменяется на деление, а значащие цифры подставляются в обратном порядке — справа налево.

Задачи №23–32 на перевод десятичных чисел в другие системы счисления не должны вызвать принципиальных затруднений. Однако практика работы в школе показывает, что нередко возникает парадоксальная проблема: многие ученики не умеют выполнять деление многозначных чисел без калькулятора.

В этих же заданиях делить приходится вручную, поскольку нужно выделять остатки, чего обычно не делают калькуляторы. Иногда затруднения вызывает и перемножение многозначных чисел при переводе дробей, особенно в шестнадцатеричную систему. В таком случае не вредно восстановить навыки ручных вычислений, забытые многими учениками со времен младшей школы. Задайте детям вопрос: «Кто умнее, человек или калькулятор?» Пусть это стимулирует их поднять свой интеллект выше уровня пластмассовой коробочки.

В задачах №33–45 используется связь между системами с основаниями 2^n . Требуется умение производить переводы между числами в системах с основаниями 2, 8, 16. Как отмечалось выше, такой перевод происходит путем формального перекодирования с помощью двоично/шестнадцатеричного или двоично/восьмеричного кода. Таблицы такой перекодировки приведены в задачнике.

Пример 3.

Задачник-практикум, раздел 1.5, № 38. *Перевести шестнадцатеричные числа в восьмеричную систему.*

Конечно, такой перевод можно производить и через десятичную систему по схеме $16 \Rightarrow 10 \Rightarrow 8$. Но это долго и неудобно. Лучше выполнять такой перевод по схеме $16 \Rightarrow 2 \Rightarrow 8$. В этом случае ничего не требуется вычислять, все сводится к формальной перекодиров-

ке. На втором шаге следует сгруппировать двоичные цифры тройками:

$$\begin{aligned} F_{12,0457}_{16} &= 1111\ 0001\ 0010,0000\ 0100\ 0101\ 0111_2 = \\ &= 111\ 100\ 010\ 010,000\ 001\ 000\ 101\ 011\ 100 = \\ &= 7422,010534_8. \end{aligned}$$

В задачах № 46–62 от учеников требуется умение выполнять арифметические вычисления в различных позиционных системах счисления. Следует обратить внимание учеников на то, что в основе таких вычислений лежат правила сложения и умножения однозначных чисел, а вычислительные алгоритмы везде одинаковы. Наибольшую сложность могут представить задачи с определением неизвестных цифр.

Пример 4.

Задачник-практикум, раздел 1.5, задача № 53. *Найти основание p системы счисления и цифру n , если верно равенство: $33m5n + 2n443 = 55424$. Пример выполнен в системе счисления с основанием p , m — максимальная цифра в этой системе.*

Запишем столбиком данное сложение:

$$\begin{array}{r} 33m5n \\ + 2n443 \\ \hline 55424 \end{array}$$

Очевидно, основание системы $p \geq 6$, так как присутствует цифра 5. Сложение в младшем разряде дает: $n + 3 = 4$. Отсюда $n = 1$. Сложение во втором справа разряде дает:

$$5 + 4 = 12_p = (1 \times p + 2)_{10} = 9_{10}.$$

Отсюда следует, что $p = 9 - 2 = 7$. Наибольшая цифра в семеричной системе — 6. Значит, $m = 6$. Если теперь подставить в данное выражение вместо букв соответствующие им цифры: $n = 1$, $m = 6$ и выполнить сложение в семеричной системе счисления, то получится сумма, данная в условии задачи.

Пример 5.

[2], раздел 1.5, задача № 57. В какой системе счисления выполнено следующее сложение?

$$\begin{array}{r}
 756 \\
 + 307 \\
 \hline
 2456 \\
 \hline
 24 \\
 \hline
 3767
 \end{array}$$

Решение этой задачи рекомендуется искать методом гипотез. Очевидно, что основание системы $p \geq 8$. Оно меньше 10, поскольку нет цифр, обозначенных буквами, а правилам десятичной арифметики данный пример не удовлетворяет. Итак, p равно 8 или 9. Выполним сложение младших разрядов в десятичной системе:

$$6 + 7 + 6 + 4 = 23_{10} = x7_p.$$

В системе с основанием p это двузначное число с младшей цифрой 7 и неизвестной первой слева цифрой. Переведем число 23_{10} в восьмеричную и девятиричную системы. Получим:

$$23_{10} = 27_8 = 25_9.$$

Очевидно, подходит вариант $p = 8$. Проверяя выполнение сложения других разрядов в восьмеричной системе, убеждаемся, что решение верно.

7. Архитектура ЭВМ

Разделы учебника: часть I, § 7–9.

Основные цели. Дать начальные представления о назначении компьютера, его устройстве, функциях основных узлов. Заложить основу для будущего более подробного изучения аппаратных средств компьютера.

Изучаемые вопросы.

- Из каких устройств состоит компьютер (ЭВМ).
- Принцип программного управления ЭВМ.
- Организация информации во внутренней и внешней памяти.
- Устройство персонального компьютера.

Общие методические рекомендации

Одна из содержательных линий базового курса информатики — линия компьютера. Эта линия делится на четыре ветви: устройство компьютера; программное обеспечение; представление данных в ЭВМ; история и перспективы развития ЭВМ (рис. 7.1).

Линия компьютера проходит через весь курс. В большинстве последующих тем ученики будут иметь дело с компьютером, углубляя свои представления о его устройстве, возможностях; развивая собственные навыки работы на компьютере. Освоение содержательной линии «Компьютер» происходит по двум целевым направлениям:

- 1) теоретическое изучение устройства, принципов функционирования и организации данных в ЭВМ;
- 2) практическое освоение компьютера; получение навыков применения компьютера для выполнения различных видов работы с информацией.

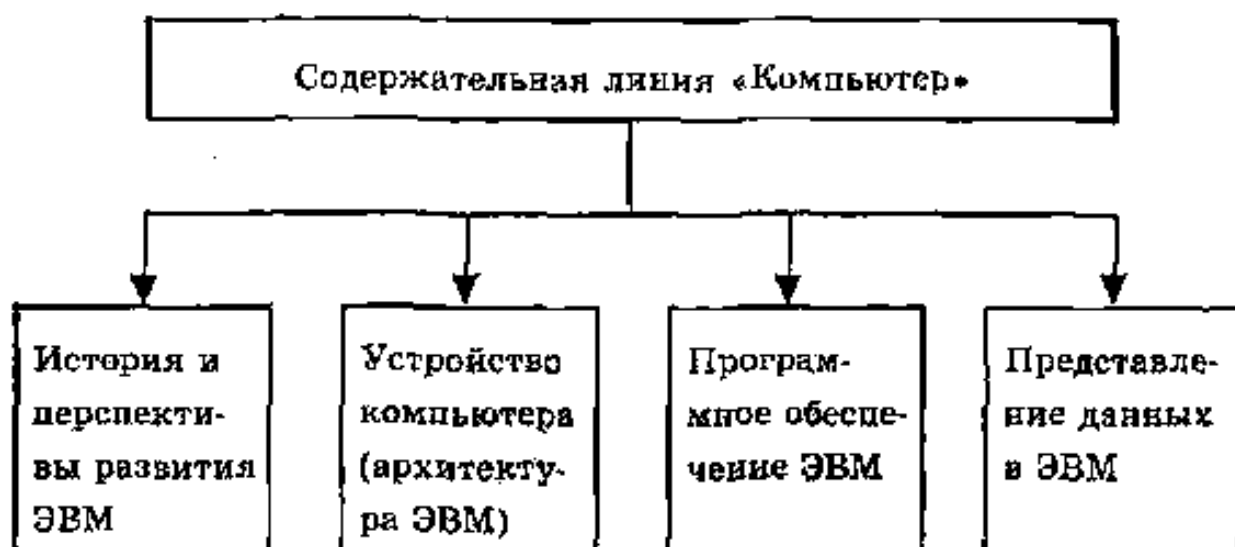


Рис. 7.1. Структура содержательной линии «Компьютер»

В курсе информатики устройство компьютера изучается на уровне его *архитектуры*. Под архитектурой понимают описание устройства и принципов работы ЭВМ без подробностей технического характера (электронных схем, конструктивных деталей и пр.). Описание архитектуры — это представление о компьютере, достаточное для человека, работающего за компьютером, но не конструирующего или ремонтирующего его, то есть для пользователя (в том числе и программиста).

Различным пользователям, в зависимости от уровня использования ими ЭВМ, требуется различный уровень знаний об архитектуре. Наиболее глубокие знания архитектуры компьютера требуются программистам, особенно системным программистам. Как же можно обрисовать диапазон понятий, подходящих под определение архитектуры ЭВМ? Самый поверхностный уровень — это понятия об основных устройствах, входящих в состав ЭВМ, и их назначении. Самый глубокий уровень — это описание системы команд процессора (языка машинных команд), правил работы процессора при выполнении программы. В ходе изучения базового курса ученики будут постепенно углублять свои знания об архитектуре компьютера. В главе 10 им предстоит небольшое знакомство с языком машинных команд, с процессором компьютера. Необходимость таких знаний следует из основной концепции курса: направленности на фундаментальное, базовое образование.

В учебнике разъясняются общие понятия архитектуры без привязки к конкретным маркам ЭВМ. Практическая же работа на уроках происходит на определенных моделях компьютеров, то есть вторая из отмеченных выше целей связана с освоением работы на том типе компьютера, который имеется в классе. В связи с этим возникают две проблемы: проблема увязки общетеоретических знаний с практикой и проблема подбора источников информации по данному типу ЭВМ. Такого

же рода проблемы возникают и при изучении средств программного обеспечения.

Проблему увязки теории с практикой должен решать сам учитель. Вводя общие понятия, например, такие как объем памяти, разрядность процессора, тактовая частота и другие, следует сообщить ученикам, какие конкретно значения этих параметров имеются у школьных компьютеров. Рассказывая о назначении устройств ввода и вывода, о носителях информации, учитель должен продемонстрировать эти устройства, познакомить учеников с их характеристиками, с правилами обращения. Безусловно, нужно рассказывать и о возможностях и характеристиках более совершенной и современной техники, чем та, что есть в школе, раскрывать перспективы ее развития. Однако, прежде всего ученики должны хорошо узнать свой компьютер.

Помимо учебника по базовому курсу ученикам необходима справочная литература по работе с аппаратными и программными средствами школьного компьютера. Очень хорошо, если такая литература ориентирована на использование в школе. В настоящее время издается и продается очень много компьютерной литературы, главным образом по IBM PC. Далеко не всякая такая книжка, как по цене, так и по методическим качествам подходит для использования в школе. Подходящую для школы справочную литературу часто издают региональные методические центры, сами учителя. Много полезных публикаций такого плана можно найти в педагогической периодике: журнале «Информатика и образование», газете «Информатика» и др. Используя множительную технику, учитель может подготовить для учеников справочный раздаточный материал. Важной педагогической целью является развитие у детей навыков самостоятельной работы со справочной литературой. Эти навыки будут необходимы в случае, если их дальнейшая учеба и работа окажутся связанными

с компьютерной техникой, вероятность чего со временем возрастает.

Методические рекомендации по изложению теоретического материала

1. Основные понятия, которые вводятся в § 7–9 части I учебника: *архитектура ЭВМ, память ЭВМ (оперативная, внешняя), процессор, устройства ввода, устройства вывода, программа, данные, программное управление.*

О смысле понятия «архитектура ЭВМ» говорилось выше. В тексте § 7 заложен методический прием, который предлагается использовать при объяснении данного материала. Это прием аналогии. Суть его сводится к следующему. По своему назначению компьютер — это универсальная машина для работы с информацией. Но в природе уже есть такая «биологическая машина», это человек! Информационная функция человека рассматривалась раньше. Она сводится к умению осуществлять три типа информационных процессов: хранение информации, обработку информации, прием/передачу информации, то есть информационную связь с внешним миром. Значит, в состав устройств компьютера должны входить технические средства для осуществления этих процессов. Они называются: память, процессор, устройства ввода и вывода (табл. 7.1).

Таблица 7.1.

Функция	Человек	Компьютер
Хранение информации	Память	Устройства памяти
Обработка информации	Мышление	Процессор
Прием информации	Органы чувств	Устройства ввода
Передача информации	Речь, двигательная система	Устройства вывода

Деление памяти компьютера на внутреннюю и внешнюю также поясняется через аналогию с человеком. Внутренняя память — это собственная (биологическая) память человека; внешняя память — это разнообразные средства записи информации — бумажные, магнитные и пр.

Различные устройства компьютера связаны между собой каналами передачи информации. Из внешнего мира информация поступает в компьютер через устройства ввода; поступившая информация попадает во внутреннюю память. Если требуется длительное ее хранение, то из внутренней памяти она переписывается во внешнюю. Обработка информации осуществляется процессором при непрерывной двусторонней связи с внутренней памятью: оттуда извлекаются исходные данные, туда же помещаются результаты обработки. Информация из внутренней памяти может быть передана во внешний мир (человеку или другим компьютерам) через устройства вывода. Сказанное схематически отображено в учебнике на рис. 2.1.

2. Необходимо объяснить ученикам разницу между понятиями «данные» и «программа». Как уже было сказано, между различными устройствами компьютера циркулирует информация. Всю эту информацию можно разделить на две разновидности: *данные* и *программы*. Здесь следует вспомнить утверждение, которое было сделано в главе 1 «Человек и информация» о том, что все многообразие информации, с которой человек имеет дело, можно поделить на декларативную (я знаю, что...) и процедурную (я знаю, как...). Для компьютера *декларативная информация — это данные*, а *процедурная — это программы*. Данные — это информация, которая обрабатывается компьютером автоматически в соответствии с программой.

Полезно еще раз использовать прием аналогии между человеком и компьютером. Следует привести какой-нибудь знакомый ученикам пример решения ин-

формационной задачи. Вот, например, задача вычисления объема прямоугольного параллелепипеда:

Исходные данные:
Три числа: a , b , c —
длины ребер парал-
лелепипеда.

Программа:
1. Вычислить площадь
основания: $S = a \times b$,
2. Вычислить объем:
 $V = S \times c$.

Здесь данными являются пять чисел: a , b , c , S , V . Они делятся на исходные данные — a , b , c ; промежуточные — S и окончательные (результаты) — V .

*Программа состоит из двух команд, которые должен выполнить человек, чтобы решить задачу. И если человек умеет выполнять операцию умножения, то он решит эту задачу, даже, может быть, и не понимая, почему ее надо решать таким образом; то есть человек будет действовать формально. В таком случае его можно назвать **формальным исполнителем программы**.*

Ситуация с компьютером аналогична. Для решения рассмотренной задачи компьютеру нужно сообщить исходные данные и программу работы. И данные и программа представляются в определенной форме, «понятной» компьютеру, заносятся во *внутреннюю память компьютера* и затем компьютер переходит к выполнению программы, то есть решению задачи. Компьютер является *формальным исполнителем программы*.

Необходимо подчеркнуть, что любая работа выполняется компьютером по программе, будь то решение математической задачи, перевод текста с иностранного языка, получение рисунков на экране, игра с пользователем и пр. Подводя итог теме, следует сказать, что *суть принципа программного управления компьютером сводится к следующим трем положениям:*

- 1) любая работа выполняется компьютером по программе;
- 2) исполняемая программа находится в оперативной памяти;
- 3) программа выполняется автоматически.

3. Знакомство учеников с архитектурой ЭВМ должно происходить путем постепенного высвечивания того «черного ящика», каким первоначально был для них компьютер. Этот процесс начинается с главы 2 учебника и продолжается в некоторых последующих главах. В § 8 «высвечивается» компьютерная память.

О делении памяти на внутреннюю и внешнюю ученики уже знают. Какие свойства каждого из этих видов памяти им нужно усвоить на данном этапе курса? Следует говорить о двух типах свойств: о физических свойствах и о принципах организации информации.

Внутренняя память. К физическим относятся следующие свойства:

- это память, построенная на электронных элементах (микросхемах), которая хранит информацию только при наличии электропитания; по этой причине внутреннюю память можно назвать *энергозависимой*;
- это *быстрая* память; время занесения (записи) в нее информации и извлечения (чтения) очень маленькое — микросекунды;
- это память *небольшая по объему* (по сравнению с внешней памятью).

Быструю энергозависимую внутреннюю память называют *оперативной памятью*, или ОЗУ — оперативное запоминающее устройство.

В качестве дополнительной информации ученикам можно сообщить, что в компьютере имеется еще один вид внутренней памяти — постоянное запоминающее устройство (ПЗУ). Основное его отличие от ОЗУ — энергонезависимость, то есть при отключении компьютера от электросети в ПЗУ информация не исчезает. Кроме того, однажды записанная в ПЗУ информация не меняется. ПЗУ — это память, предназначенная только для чтения, в то время как ОЗУ — для чтения и для записи. Обычно ПЗУ по объему существенно меньше ОЗУ.

Внешняя память. Есть две разновидности носителей информации, используемых в устройствах внешней памяти: магнитные и оптические. Существуют магнитные ленты и магнитные диски. CD-ROM (Compact Disk — Read Only Memory) — оптический диск, предназначенный только для чтения. Существуют оптические диски с возможностью записи. По аналогии с отмеченными выше физическими свойствами внутренней памяти, свойства внешней памяти описываются так:

- внешняя память *энергонезависима*, то есть информация в ней сохраняется независимо от того, включен или выключен компьютер, вставлен носитель в компьютер или лежит на столе;
- внешняя память — *медленная* по сравнению с оперативной; в порядке возрастания скорости чтения/записи информации, устройства внешней памяти располагаются так: магнитные ленты — магнитные диски — оптические диски;
- объем информации, помещающейся во внешней памяти больше, чем во внутренней; а с учетом возможности смены носителей — неограничен.

Необходимо обращать внимание учеников на точность в используемой терминологии. Ленты, диски — это носители информации. Устройство компьютера, которое работает с магнитной лентой, записывает и считывает с нее информацию, называется *накопителем на магнитной ленте* (НМЛ). Употребляется также английское название этого устройства — *стример*. Устройство чтения/записи на магнитный диск называется *накопителем на магнитном диске* (НМД) или *дисководом*. С оптическими дисками работает *оптический дисковод*.

Теперь — о принципах организации информации. Изучив базовый курс, ученики должны будут узнать, что:

1) компьютер работает со следующими видами данных (обрабатываемой информации): символьным, числовым, графическим, звуковым;

2) любая информация в памяти компьютера (в том числе и программы) представляется в двоичном виде.

Сформулированные положения следует сообщить ученикам в данной теме и в последующих темах к ним возвращаться.

Двоичный вид обозначает то, что любая информация в памяти компьютера представляется с помощью всего двух символов: нуля и единицы. Как известно, один символ из двухсимвольного алфавита несет 1 бит информации. Поэтому двоичную форму представления информации еще называют битовой формой. В электронных элементах компьютера происходит передача и преобразование электрических сигналов. Двоичные символы распознаются так: есть сигнал — единица, нет сигнала — ноль. На магнитных носителях единице соответствует намагниченный участок поверхности, нулю — ненамагниченный.

Информационную структуру внутренней памяти следует представлять как последовательность двоичных ячеек — битов. В учебнике она изображена на рис. 2.2. Битовая структура внутренней памяти определяет ее первое свойство: *дискретность*. Каждый бит памяти в данный момент хранит одно из двух значений: 0 или 1, то есть один бит информации. В процессе работы компьютера эти нули и единички «мигают» в ячейках. Можно предложить ученикам такой зрительный образ: представьте себе память компьютера в виде фасада многоэтажного дома вечером. В одних окнах горит свет, в других — нет. Окно — это бит памяти. Окно светится — единица, не светится — ноль. И если все жильцы начнут щелкать выключателями, то фасад будет подобен памяти работающего компьютера, в которой перемигиваются единички и нули.

Второе свойство внутренней памяти называется *адресуемостью*. Но адресуются не биты, а *байты* — 8 расположенных подряд битов памяти. Адрес байта — это его порядковый номер в памяти. Здесь снова можно предложить аналогию с домом: квартиры в доме пронумерованы; порядковый номер квартиры — ее адрес. Только в отличие от квартир, нумерация которых начинается с единицы, номера байтов памяти начинаются с нуля. Доступ к информации в оперативной памяти происходит по адресам: чтобы записать данные в память, нужно указать, в какие байты ее следует занести. Точно так же и чтение из памяти производится по адресам. Таким способом процессор общается с оперативной памятью. Можно продолжить аналогию с домом: чтобы попасть в нужную квартиру или переслать туда письмо, нужно знать адрес.

Итак, информационная структура внутренней памяти — битово-байтовая. Ее размер (объем) обычно выражают в килобайтах, мегабайтах.

Информационная структура внешней памяти — файловая. Наименьшей именуемой единицей во внешней памяти является файл. Для объяснения этого понятия в учебнике предлагается книжная аналогия: файл — это аналог наименьшего поименованного раздела книги (параграфа, рассказа). Конечно, информация, хранящаяся в файлах, тоже состоит из битов и байтов. Но, в отличие от внутренней памяти, байты на дисках не адресуются. При поиске нужной информации на внешнем носителе должно указываться имя файла, в котором она содержится; сохранение информации производится в файле с конкретным именем.

Надо сказать, что понятие файла усваивается детьми постепенно, с накоплением опыта практической работы на компьютере. При изучении первой прикладной темы — работы с текстом, им предстоит самим сохранять файлы, открывать файлы. И только после этого

представление о файлах из абстрактного превратится в конкретное.

На магнитные носители информация записывается (и считывается) с помощью *магнитной головки* накопителя, подобно бытовому магнитофону. Линия, по которой магнитная головка контактирует с магнитной поверхностью носителя, называется *дорожкой*. На ленте дорожки продольные (прямые), на диске — круговые. Магнитная головка дисководов подвижная. Она может перемещаться вдоль радиуса диска. При таком *перемещении* происходит переход с одной дорожки на другую.

Книжная аналогия помогает понять ученикам назначение *корневого каталога диска* — его своеобразного оглавления. Это список, в котором содержатся сведения о файлах на диске; иногда его называют *директорией* диска. В каталоге содержатся сведения о файле (имя, размер в байтах, дата и время создания или последнего изменения). Эта информация всегда хранится на определенных дорожках. Если список файлов вывести на экран, то, подобно просмотру оглавления книги, из него можно получить представление о содержимом диска.

4. Существуют различные классы электронно-вычислительных машин: суперЭВМ, большие ЭВМ, мини-ЭВМ, микроЭВМ. *Персональные компьютеры (ПК)* относятся к классу микроЭВМ. В абсолютном большинстве учебных заведений используются ПК. По этой причине ученики, прежде всего, должны получить представление об устройстве персонального компьютера.

В § 7 и 9 присутствуют три схемы, имеющие отношение к устройству компьютера. Давайте в них разберемся и определим разницу в назначении этих схем.

Рис. 2.1 представляет собой *схему информационного взаимодействия основных устройств компьютера любого класса*. Это схема, которая была предложена Джоном фон Нейманом еще в 1946 году, для однопро-

цессорной ЭВМ. Заметим, что в настоящее время существуют компьютеры с многопроцессорной архитектурой (суперЭВМ), для которых эта схема не применима. Большинство же типов персональных компьютеров имеют один центральный процессор, и их устройство соответствует рис. 2.1. Данная схема полностью лишена каких-либо конструктивных деталей. Ее назначение — отразить пути информационного обмена между устройствами компьютера в процессе его работы.

Рис. 2.4 тоже отражает информационное взаимодействие между устройствами, но применительно к персональному компьютеру. Этот рисунок содержит в себе некоторые конструктивные детали, характерные для ПК. В нем присутствует следующая информация: роль центрального процессора в ПК выполняет микропроцессор; в качестве устройства ввода используется клавиатура; устройства вывода — монитор и принтер; устройство внешней памяти — дисковод. Информационная связь между устройствами осуществляется через общую многопроводную магистраль (шину); внешние устройства подсоединены к магистрали через контроллеры. Необходимо обратить внимание учеников на то, что принципы информационного взаимодействия, отраженные на рис. 2.1, справедливы и для ПК. Таким образом, эти две схемы дополняют друг друга.

Структуру ПК, изображенную на рис. 2.4, принято называть *архитектурой с общей шиной* (другое название — *магистральная архитектура*). Впервые она была применена на мини-ЭВМ третьего поколения, затем перенесена на микроЭВМ и ПК. Ее главное достоинство — простота, возможность легко изменять конфигурацию компьютера путем добавления новых устройств или замены старых. Отмеченные возможности принято называть *принципом открытой архитектуры ПК*.

В архитектуре различных типов ПК имеются свои особенности. Например, в IBM PC между микропроцессором и внутренней памятью имеется линия прямой связи, помимо общей шины. Клавиатура с микропроцессором

также связана отдельным каналом. Схема на рис. 2.4 носит упрощенный, обобщенный характер. В качестве дополнительной информации учитель может рассказать об особенностях архитектуры модели школьного ПК, используя дополнительную литературу (см., например, [23]). Однако следует соблюдать меру и не «сваливать сразу на головы детей» множество технических подробностей. Такую информацию можно давать постепенно, небольшими порциями в течение всего курса.

Схему на рис. 2.3 можно назвать *компоновочной схемой*. Она отражает состав комплекта блоков (отдельных корпусов) настольного варианта персонального компьютера, а также содержимое системного блока. Компоновка устройств портативных ПК другая. В них все основные устройства объединены в одном корпусе, используется жидкокристаллический монитор.

Можно говорить о том, что основным устройством ПК является микропроцессор (МП). Это «мозг» машины. В первую очередь возможности МП определяют возможности компьютера в целом. Для пользователя наиболее важным свойством ЭВМ является ее быстродействие, то есть скорость обработки информации. Для ЭВМ первых поколений было принято выражать быстродействие компьютера в количестве операций, выполняемых за одну секунду (опер./с). В те времена компьютеры использовались, главным образом, для математических расчетов, поэтому имелись в виду арифметические и логические операции. Такая характеристика быстродействия позволяла спрогнозировать время решения математической задачи. На современных компьютерах гораздо более разнообразны типы решаемых задач, виды обрабатываемой информации. Единица «опер./с» сейчас не употребляется. Скорость работы компьютера зависит от целого ряда его характеристик. Важнейшими из них являются две характеристики процессора: *тактовая частота* и *разрядность*. В учебнике приводится аналогия понятия тактовой частоты и частоты ударов метронома, задающего темп исполнения музыкального

произведения. Кстати, эту музыкальную аналогию можно усилить, если сказать о том, что различные устройства компьютера подобны музыкантам ансамбля, исполняющим одно произведение. Своеобразной партитурой здесь является программа, а генератор тактовой частоты задает темп исполнению. И чем быстрее он «стучит», тем быстрее работает компьютер, решается задача.

Разрядность процессора — это размер той порции информации, которую процессор может обработать за одну операцию (одну команду). Такими порциями процессор обменивается данными с оперативной памятью. На современных компьютерах чаще всего используются 32- и 64-разрядные процессоры. Фактически разрядность тоже влияет на быстродействие, поскольку, чем больше разрядность, тем больший объем информации процессор может обработать за единицу времени.

Дополнительный материал для углубленного изучения базового курса

При наличии дополнительного учебного времени, полезно обсудить с учениками понятие «машинное слово», а также рассмотреть информационную структуру магнитных дисков. Практические задания по этим вопросам имеются в пособии [2].

Машинное слово — это еще одна информационная единица оперативной памяти. Но если понятия бита и байта инвариантны, то есть не зависят от типа компьютера, то машинное слово у разных ЭВМ бывает разным. *Размер машинного слова (в битах) равен разрядности процессора.* Следовательно, у компьютера с 8-разрядным процессором размер машинного слова равен 1 байту, с 16-разрядным процессором — 2 байтам, с 32-разрядным процессором — 4 байтам и так далее. Данное ранее правило можно перефразировать теперь так: обмен информацией между процессором и оперативной памятью происходит порциями,

объем каждой из которых равен размеру машинного слова.

В ОЗУ слово — это адресуемая часть памяти. Адрес слова памяти равен адресу входящего в него младшего байта. Если размер слова равен 1 байту, то адреса слов, как и адреса байтов, изменяются через единицу; если размер слова — 2 байта, то адреса слов меняются через двойку: 0, 2, 4, 6, ..., то есть являются четными числами. На рис. 7.2 показан принцип деления памяти на слова для 32-разрядного компьютера.

Адреса слов	БАЙТЫ			
	0	1	2	3
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11
12	12	13	14	15
...	...	...	...	...

Рис. 7.2. Адресация оперативной памяти

Из всех устройств внешней памяти наиболее активно используемыми являются магнитные диски. Они позволяют считывать и записывать информацию, переносить информацию с одного компьютера на другой, длительно хранить информацию вне компьютера. Информационный объем магнитного диска — величина конечная. Поэтому пользователь должен уметь сопоставлять эту величину с объемом информации, которую собирается сохранить.

Учитель, рассказывая на уроке об устройстве персонального компьютера, имеющегося в компьютерном классе, обязательно должен уделить внимание типам используемых дисковых устройств. Следует рассказать ученикам о том, что существуют жесткие, встроенные в системный блок магнитные диски большого объема — винчестеры. Гибкие диски — дискеты пользова-

тель может сам вставлять в дисковод, это сменные носители. Надо сообщить ученикам информационную емкость используемых на ПК носителей.

В качестве дополнительного материала можно рассказать об информационной структуре диска более подробно, чем об этом написано в учебнике. Представление о магнитной дорожке уже было введено. Теперь можно ввести понятие *магнитной поверхности (стороны)* дискового накопителя. Дискета — однодисковое устройство, поэтому у нее может быть одна или две магнитных поверхности. С каждой поверхностью контактирует отдельная магнитная головка. Винчестер представляет собой пакет дисков, закрепленных на общей оси. Соответственно, число магнитных поверхностей может быть до $2n$, где n — число дисков в пакете. Дорожки на магнитных поверхностях расположены концентрично, их количество на каждой поверхности одинаково. Например, если диск 4-сторонний и на каждой стороне расположено по 10 дорожек, то на всем диске — 40 магнитных дорожек. На каждой магнитной поверхности дорожки пронумерованы.

Еще одно новое понятие — *сектор*. Каждая дорожка поделена на целое число одинаковых секторов. Все секторы имеют равные информационные объемы. Характерным размером сектора является величина 512 байт = 0,5 Кб. Размер сектора — это объем той наименьшей порции информации, которая передается при обмене между оперативной памятью и магнитным диском. Секторы на дорожке пронумерованы. Таким образом, координаты сектора на магнитной поверхности определяются номером дорожки и номером сектора. По этим координатам и происходит поиск информации на диске.

Процедура разметки магнитного диска на дорожки и секторы называется *форматированием диска*. Форматирование производится с помощью специальной системной программы. Ученикам следует знать, что если на диске ранее была записана какая-то ин-

формация, то вследствие форматирования она будет потеряна.

Методические рекомендации по решению задач

Раздел 3.1.1 задачника-практикума содержит задачи на расчет параметров внутренней памяти компьютера. Для их решения ученики должны быть знакомы с понятиями: объем памяти, размер машинного слова, принципы адресации памяти, связь между единицами памяти бит, байт, килобайт, мегабайт.

Пример 1.

Задачник-практикум: раздел 3.1.1, задача № 2.
Объем оперативной памяти компьютера составляет 1/8 часть мегабайта. Сколько машинных слов составляют оперативную память, если одно слово содержит 64 бита?

Во-первых, нужно перевести объем памяти и размер машинного слова в одинаковые единицы. Удобнее всего — в байты. Обозначим объем памяти буквой M , а размер слова — W . Тогда:

$$M = 1/8 \text{ МБ} = 1024 \times 1024 / 8 = 131\,072 \text{ байта}, \\ W = 64 / 8 = 8 \text{ байтов}.$$

Теперь можно вычислить число слов, составляющих память:

$$N = M / W = 131\,072 / 8 = 16\,384 \text{ слова}.$$

Пример 2.

Задачник-практикум, раздел 3.1.1, задача № 8.
Какой объем имеет оперативная память компьютера, если $3FF$ — шестнадцатеричный адрес последнего байта памяти?

Здесь подразумевается, что объем памяти нужно выразить десятичным числом, равным количеству байтов, составляющих ОЗУ. Адрес последнего байта задан в шестнадцатеричной системе. Поскольку нумерация

байтов памяти начинается с нуля, то значит диапазон адресов от 0 до 3FF. Следовательно, число байтов памяти в шестнадцатеричной системе счисления равно $3FF + 1 = 400_{16}$. Для получения ответа нужно перевести это число в десятичную систему счисления:

$$400_{16} = 4 \times 16^2 = 4 \times 256 = 1024 \text{ байт} = 1 \text{ Кбайт}.$$

Пример 3.

Задачник-практикум: раздел 3.1.1, задача № 14. *Компьютер имеет объем оперативной памяти, равный 0,5 Кбайт и содержит 128 машинных слов. Укажите адрес последнего байта и адрес последнего машинного слова памяти в шестнадцатеричной форме.*

Выразим размер памяти в байтах:

$$0,5 \text{ Кбайт} = 512 \text{ байт}.$$

Размер машинного слова определяется делением объема памяти на число слов в памяти:

$$512/128 = 4 \text{ байта}.$$

Переведем величину объема памяти в шестнадцатеричную систему счисления:

$$512_{10} = 2 \times 256 = 2 \times 16^2 = 200_{16}.$$

Следовательно, диапазон шестнадцатеричных адресов байтов памяти: от 0 до 1FF. Отсюда *адрес последнего байта равен 1FF*.

Последнее машинное слово включает в себя 4 последних байта памяти: 1FC, 1FD, 1FE, 1FF. Значит, *адрес последнего слова равен 1FC*.

Задачи на расчеты, связанные с параметрами информационно-емкости дисков, присутствуют в разделе 3.1.2 задачника-практикума. Для решения этих задач требуется знать связь между объемом диска и числом сторон (магнитных поверхностей), числом доро-

жек, числом секторов на дорожке, числом байтов в секторе. Эта связь выражается следующей формулой:

$$\text{ОБЪЕМ} = \text{СТОРОНЫ} \times \text{ДОРОЖКИ} \times \text{СЕКТОРЫ} \times \text{БАЙТЫ}$$

Пример 4.

Задачник-практикум: раздел 3.1.2, задача № 16. *Двусторонняя дискета имеет объем 1200 Кбайт. Сколько дорожек на одной стороне дискеты, если каждая дорожка содержит 15 секторов по 4096 битов?*

Как это уже делалось раньше, здесь нужно перейти к одной единице измерения информации. Переведем в килобайты размер сектора:

$$4096/8 = 512 \text{ байт} = 0,5 \text{ Кбайт.}$$

Теперь вычислим информационный размер дорожки:

$$0,5 \times 15 = 7,5 \text{ Кбайт.}$$

Поскольку дискета двухсторонняя, то на одной стороне

$$1200/2 = 600 \text{ Кбайт.}$$

Теперь можно получить окончательный ответ. Чтобы найти число дорожек на одной стороне дискеты, нужно информационный объем стороны разделить на информационный размер одной дорожки:

$$600/7,5 = 80 \text{ дорожек.}$$

Пример 5.

Задачник-практикум: раздел 3.1.2, задача № 22. *Односторонняя дискета имеет объем 180 Кбайт. Сколько дорожек на диске, если каждая из них содержит 9 секторов, а в каждом секторе размещается по 1024 символа из 16-символьного алфавита.*

Главная «хитрость» этой задачи состоит в том, что надо сообразить, сколько памяти занимает 1 символ. Этот вопрос относится к теме «Измерение информации». Один символ из 16-символьного алфавита несет

4 бита информации, поскольку $2^4 = 16$. Это значит, что и в памяти компьютера символы такого алфавита будут занимать по 4 бита. Следовательно, в одном байте помещается 2 таких символа.

Теперь можно определить размер сектора. Он равен:

$$1024/2 = 512 \text{ байт} = 0,5 \text{ Кбайт.}$$

Поскольку на одной дорожке размещается 9 секторов, то информационный объем дорожки равен:

$$9 \times 0,5 = 4,5 \text{ Кбайт.}$$

Теперь можно определить число дорожек на всем диске:

$$180/4,5 = 40 \text{ дорожек.}$$

8. Программное обеспечение ЭВМ

Разделы учебника: часть I, § 10–13.

Основные цели. Дать учащимся представление о составе программного обеспечения компьютера. Раскрыть назначение операционной системы; ввести понятие файловой структуры дисков.

Изучаемые вопросы

- ☐ Назначение программного обеспечения (ПО) ЭВМ.
- ☐ Классификация ПО ЭВМ.
- ☐ Что такое прикладное ПО.
- ☐ Назначение систем программирования.
- ☐ Основные функции операционной системы
- ☐ Начальные сведения об организации файлов.
- ☐ Пользовательский интерфейс.

Методические рекомендации по изложению теоретического материала

1. Основная педагогическая задача этого раздела — привести учеников к пониманию того факта, что современный компьютер представляет собой двуединую систему, состоящую из аппаратной части (технических устройств) и информационной части (программного обеспечения). Полезно познакомить учащихся с терминами *hardware & software* — «твердая» компонента и «мягкая» компонента компьютера, поскольку в последнее время они употребляются довольно часто.

КОМПЬЮТЕР = АППАРАТУРА (*hardware*) +
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (*software*)

Логика раскрытия этой темы учителем может быть отражена в следующих рассуждениях. Если компьютер выполняет любую работу только под управлением программы, то значит ли это, что человек, желающий воспользоваться компьютером, должен уметь программировать? Если бы компьютер представлял собой только «голую» аппаратуру, то это было бы так. Причем программировать пришлось бы на языке процессора. Так было на самых первых ЭВМ, на которых могли работать только профессиональные программисты.

Современный компьютер доступен практически каждому. Эта доступность обеспечена тем, что компьютер оснащен богатым программным обеспечением (ПО). ПО — это совокупность программ, хранящихся на устройствах долговременной памяти компьютера и предназначенных для массового использования. И если пользователю требуется выполнить какую-то работу на компьютере, то он должен выбрать подходящую для этих целей программу из ПО и инициализировать ее выполнение. Таким образом, использование компьютера человеком происходит по такой схеме:
ЗАДАЧА⇒ВЫБОР И ИНИЦИАЛИЗАЦИЯ ПРОГРАММЫ⇒РАБОТА

Здесь термин «задача» понимается в самом широком смысле и обозначает любую информационную потребность пользователя, которую можно удовлетворить с помощью компьютера: создать текстовый документ, нарисовать иллюстрацию, выполнить вычисления, получить справку, принять и отправить электронную почту и так далее.

В дальнейшем учителю необходимо будет обращать внимание на то, чтобы ученики отчетливо понимали, с помощью каких программных средств какие информационные задачи можно решать. При этом они должны научиться отделять задачи системного характера от задач прикладного характера, например, понимать, что копирование или удаление файлов осуществляется с помощью операционной системы, а редактирование текстового документа — с помощью текстового редактора, то есть прикладной программы.

Сделаем еще два замечания, суть которых должна быть понятна учителю и, при необходимости, может быть объяснена ученикам.

Во-первых, программное обеспечение — это не только собственно программы, но и данные, с которыми работают эти программы. Данные и программы хранятся на дисках в отдельных файлах. Часто объем данных во много раз превышает размеры программ. Например, нередко размер баз данных, графической информации, мультимедийных продуктов много больше, чем размер программ, позволяющих работать с ними.

В учебнике использована аналогия с человеком: «твердая» компонента — это анатомия и физиология человека, «мягкая» компонента — это интеллект, знания человека, которые бывают декларативными и процедурными. «Знания» компьютера — это данные (декларативная информация) и программы (процедурная информация) в его памяти. Учитывая все вышесказанное, можно сделать вывод, что более адекватным ана-

логом термина «software» является термин «информационное обеспечение компьютера».

Работа в среде современных объектно-ориентированных операционных систем (Windows) может происходить по схеме, отличной от приведенной выше. Пользователь может выбрать хранимые данные, с которыми он собирается работать, а операционная система автоматически инициирует выполнение нужной программы обработки данных. Но и в этой ситуации грамотный пользователь (а именно такими мы хотим воспитать своих учеников) должен понимать, какая программа будет выполняться на компьютере.

Во-вторых, для компьютеров, работающих в составе компьютерных сетей, представление о программном (информационном) обеспечении выходит за границы дискового пространства ПК. Для таких ПК оказываются доступными информационные ресурсы компьютеров-серверов. С подключением компьютера к сети Интернет эти ресурсы становятся практически неограниченными.

2. В учебнике рассматривается классификация ПО, согласно которой все программы делятся на *системные, прикладные и системы программирования* (рис. 8.1). Надо сказать, что это не единственный вариант классификации ПО, который встречается в литературе. Подобные классификации носят, в некотором смысле, субъективный характер. Однако рассмотренное в учебнике деление является одним из наиболее

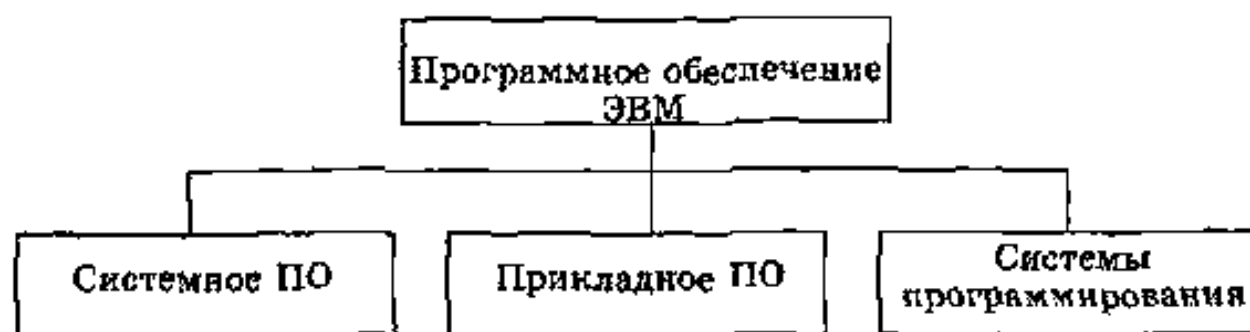


Рис. 8.1. Структура программного обеспечения ЭВМ

распространенных, и в него хорошо вписываются все те программные продукты, с которыми будут знакомиться ученики в процессе изучения информатики. Между программами, относящимися к этим трем группам, есть достаточно четкое различие по их назначению.

3. Проще всего ученикам понять назначение *прикладных программ*. Это те программы, которые непосредственно удовлетворяют информационные потребности пользователя: поиграть в компьютерную игру, напечатать рассказ, нарисовать рисунок на экране и распечатать его на бумаге, найти в компьютерном словаре перевод английского слова; выполнить вычисления с помощью калькулятора, изображенного на экране; изучить с помощью компьютера правила уличного движения и многое другое. Задача учителя — рассказать ученикам, а еще лучше — продемонстрировать разнообразные прикладные возможности современных компьютеров. То, что принято называть современными информационными технологиями, прежде всего, состоит из множества прикладных программ. На первом же уроке по данной теме приведите примеры некоторых прикладных программ, имеющихся на школьных компьютерах. Например, если в вашем классе используются IBM PC с операционной системой Windows, представьте программы из группы *Стандартные*: Калькулятор, Блокнот, Paint, объяснив их назначение.

4. Первоначальное представление о *системах программирования* должно быть дано в самом общем виде. Ученики должны получить представление о том, что:

- программы для компьютера составляют *программисты*;
- программисты пишут программы на *языках программирования*;
- существует множество различных языков программирования (Паскаль, Бейсик, Фортран и др.);

□ системы программирования позволяют программисту вводить программы в компьютер, редактировать, отлаживать, тестировать, исполнять их.

Полезно сообщить ученикам, с каким из языков программирования им предстоит познакомиться в школе.

5. Более сложной задачей является объяснение назначения *системного ПО*. Нужно дать понять ученикам, что системное ПО предназначено, прежде всего, для обслуживания самого компьютера, для управления работой его устройств. Главной частью системного ПО является *операционная система (ОС)*. ОС — это очень сложная программная система. В рамках базового курса невозможно и не нужно детально объяснять ее состав и функции. Но общее назначение, роль ОС в работе компьютера ученики должны представлять.

Начните с того, что сообщите название ОС, используемой в школьных компьютерах. Например: «Наши компьютеры работают под управлением операционной системы MS DOS (или CP/M, или Windows-98)». Фразу нужно построить именно так для того, чтобы подчеркнуть, что ОС *управляет* работой компьютера. Сообщите, что при включении компьютера происходит *загрузка операционной системы* в оперативную память ЭВМ. Точнее говоря, в ОЗУ загружается с магнитного диска *ядро ОС*, то есть та часть системы, которая должна постоянно находиться в оперативной памяти, пока работает компьютер. Диск, на котором хранится ОС и с которого происходит ее загрузка, называется *системным диском*.

Любые операционные системы, независимо от типа, выполняют три основные функции:

- 1) управление устройствами компьютера;
- 2) взаимодействие с пользователем;
- 3) работа с файлами.

Не вдаваясь в излишние подробности, учитель должен дать представление ученикам о функциях операционной системы на примере конкретной ОС, используемой в классе. Эти представления должны остава-

ться в рамках сведений, необходимых для работы пользователя с ПК.

Итак, вы сообщили ученикам, как называется ваша ОС. Далее нужно сказать о том, какой режим работы поддерживает данная система: однозадачный или многозадачный. Например, *MS DOS поддерживает однозадачный режим работы компьютера*. Это значит, что в данный момент на компьютере может выполняться только одна программа, запущенная пользователем на исполнение (инициализированная пользователем). И только после того, как выполнение этой программы будет завершено, пользователь может инициализировать другую программу. А вот операционная система *Windows поддерживает многозадачный режим работы компьютера*^{*}. Пользователь может запустить сразу несколько прикладных программ и работать с ними одновременно. Запущенные программы называются *активными задачами*, и все они отражаются значками на *панели задач Windows*. Например, можно одновременно запустить программы Калькулятор, Блокнот, Paint; на экране появятся *окна* этих программ. Пользователь, переходя из одного окна в другое, может поочередно работать с этими приложениями. При таких переходах предыдущая программа не закрывается, то есть не выгружается из оперативной памяти и в любой момент готова продолжить свою работу, как только пользователь вернется в ее окно.

Нередко встречается такая ситуация, когда одна из запущенных программ требует длительного времени для своей работы. Это, например, большие математические вычисления или проверка дисков на наличие вирусов. В таком случае эта программа выполняется в *фоновом режиме*, то есть ее исполнение не прекращается до полного завершения, а в это же время (на ее фоне) пользователь может выполнять какую-то оперативную

* Здесь и в дальнейшем имеется в виду Windows 95 и последующие версии этой ОС.

работу с другими программами, например, может набирать текст в текстовом редакторе.

В поддержке многозадачного режима работы проявляется одна из сторон управляющей функции операционной системы: управление процессором и оперативной памятью. Для одновременного выполнения нескольких программ операционная система должна *разделять* между ними *время работы процессора*, следить за размещением этих программ и данных в памяти так, чтобы они не мешали друг другу (*разделять память*).

Управление внешними устройствами компьютера — еще одна из сторон первой функции ОС. В состав операционной системы входят специальные программы управления внешними устройствами, которые называются *драйверами внешних устройств*. Для каждого типа и каждой конкретной модели внешнего устройства существует свой драйвер. Иногда ОС автоматически подбирает подходящий драйвер, иногда об этом приходится заботиться пользователю. В учебнике понятие драйвера встречается в главе 3, где речь идет о печати документов на принтере.

Научиться работать на компьютере — это значит, прежде всего, научиться взаимодействовать с операционной системой. Обычно пользователь, привыкший работать в среде какой-то определенной ОС, сталкиваясь с незнакомой ему системой, с непривычной обстановкой на экране, испытывает растерянность. В компьютерной терминологии для обозначения способа взаимодействия программы с пользователем принят термин «*пользовательский интерфейс*». Очень удобно, когда пользовательский интерфейс унифицирован. Примером такого унифицированного интерфейса является среда *диалоговой оболочки Norton Commander (NC)*. Ее аналоги реализованы в различных операционных системах. Не очень грамотный пользователь может и не знать, с какой ОС он работает, но ориентироваться в интерфейсе NC. Однако, как уже отмечалось, одна из целей базового курса — подготовка грамотного поль-

ватель через клавиатуру вводит нужную команду. При этом он должен точно соблюдать синтаксис команды, иначе она не будет воспринята. Например:

```
C:\ > copy file1.txt A:\file2.txt.
```

Перед значком **>** стоит информация, выводимая с приглашением ОС, которая указывает на текущий диск (C) и текущий каталог (корневой). После значка **>** записана команда, которую ввел пользователь. Ее смысл: скопировать файл с именем file1.txt с текущего диска и каталога в корневой каталог диска A под новым именем file2.txt.

Совокупность команд, которые понимает операционная система, составляет язык команд ОС. В таком режиме общения — режиме командной строки пользователь должен знать язык команд со всеми подробностями его синтаксиса. Сейчас весьма редко работают на ПК в режиме командной строки. Основным средством общения являются диалоговые оболочки. Но, независимо от того, используется диалоговая оболочка или нет, у любой ОС существует свой язык команд.

Работая с диалоговой оболочкой, пользователь также формирует команды ОС, только при этом он использует вспомогательные средства, упрощающие его работу. При работе с оболочками *признаком приглашения ОС является появление на экране среды (интерфейса) оболочки: панелей NC или Рабочего стола Windows. Пользователь отдает команды путем выбора из представленных на экране меню, нажатия функциональных клавиш или каких-либо групп клавиш. Использование оболочек освобождает пользователя от необходимости знать подробности синтаксиса языка команд ОС. Однако пользователь должен понимать, какие именно команды он может отдать и как это сделать. Интерфейс современных оболочек ОС достаточно дружелюбный к пользователю. В нем применяются многочисленные формы подсказок и пояснений.*

6. Третья функция операционной системы — работа с файлами. Эта работа осуществляется с помощью раздела ОС, который называется *файловой системой*. Первоначальные понятия, которые должны быть даны ученикам по данной теме, это «*имя файла*», «*тип файла*», «*файловая структура*», «*логический диск*», «*каталог*», «*путь к файлу*», «*дерево каталогов*». Все эти понятия раскрываются в § 12 учебника. Здесь учителю также следует ориентировать учеников на конкретную операционную систему. Допустим, если вы работаете с MS DOS, то говорите ученикам, что имя файла может содержать не более 8 символов — латинских букв и цифр; для Windows сообщаете, что имя файла может быть длинным (до 255 символов) и допускает использование русских букв.

Рассказывая о типах файлов и связи типа с расширением имени файла, в первую очередь разделите файлы на программные (их еще называют исполнимыми файлами) и файлы данных. Снова напомните ученикам, что вся информация в компьютере делится на программы и данные. Программные файлы имеют расширение имени *exe* или *com*. Есть еще один вид исполнимых файлов — это так называемые командные файлы с расширением *bat*. Они представляют собой программы, написанные на командном языке ОС, и выполняют некоторые системные функции. Все прочие типы файлов — это файлы данных. В дальнейшем, при изучении каждого нового приложения, обращайтесь внимание учеников на типы файлов, с которыми оно работает. Так постепенно они узнают, что текстовый редактор Word сохраняет создаваемые документы в файлах типа *doc*; графический редактор Paint создает файлы типа *bmp*; табличный процессор Excel — файлы типа *xls* и пр. Полезно в самом начале обратить внимание учеников на расширения *arj*, *rar*, *zip*. В таких файлах может храниться любая информация (программы и данные) в сжатом виде. Их называют *архивными файлами*. Сжатие (архивацию) информации производят для сокращения

занимаемого места в памяти при длительном хранении информации.

Вводить понятие *логического диска* имеет смысл лишь при наличии на вашем компьютере жесткого диска (винчестера). Только память жесткого диска делится на части, каждой из которых присваивается свое имя (С, D и так далее), и в этом случае различают понятия физического и логического дисков. Если же компьютер обладает только накопителями для дискет, то можно просто говорить о дисках А или В, не употребляя эпитеты «логический» или «физический».

Понятие дерева каталогов характерно для операционных систем, работающих с иерархическими файловыми структурами. На IBM это MS DOS и Windows. На простых учебных ЭВМ типа «Корвет», «Электроника-УКНЦ» используются операционные системы CP/M, RT-11. В них создаются простые одноуровневые файловые структуры, для которых понятие дерева отсутствует. Под каталогом (директорией) диска понимается общий список файлов на диске. Компьютеры и ОС таких типов уходят в прошлое. В подавляющем большинстве случаев пользователю современных ПК приходится иметь дело с иерархическими файловыми структурами.

Знакомство с иерархической файловой структурой дисков имеет не только практическое значение, но и теоретическое, общеобразовательное. Знакомясь с этим вопросом, ученики впервые встречаются с информационной структурой, то есть с системой данных, имеющих определенную взаимосвязь. Эти вопросы будут затронуты впоследствии в теме «Базы данных». Иерархические структуры — это один из распространенных способов организации данных (наряду с сетевым и табличным). Наглядное представление о файловой структуре дает *дерево* — графическое отображение иерархии каталогов (папок) на диске. Учителю рекомендуется активно использовать представление дерева на экране в оболочках NC или Проводник для Windows. Ученики

должны понять принципы навигации по файловому дереву, научиться перемещаться по нему вверх и вниз. Вводя представление о пути к файлу, подчеркните факт единственности пути к каждому файлу от корневого каталога. Полезно выполнить на компьютере упражнения такого рода: учитель записывает на доске полное имя файла (диск-путь-имя), ученики находят этот файл в директории на экране, работая в одной из оболочек ОС. Например, на доске записано:

`C:\WINDOWS\SYSTEM32\DRIVERS\update.sys`

Задание: начиная от корневого каталога диска C, пройти указанный путь и найти файл. В следующем упражнении попросите учеников перейти от файла, найденного в предыдущем задании, к файлу со следующим полным именем:

`C:\WINDOWS\JAVA\PACKAGES\DATA\bndrtz5v.dat`

Обратите внимание учеников, что единственный путь перехода — это возврат (движение вверх по дереву) до каталога WINDOWS, а затем движение вниз по новому направлению.

В учебнике для пояснения представлений об иерархической файловой структуре используется аналогия с системой «шкаф—ящики—папки—документы». Можно предложить и другую аналогию, ассоциируемую со словом «дерево». Файлы — это листья, каталоги (папки) — ветки. Правда, если понятие корня перевести в буквальный «растительный» смысл этого слова, то аналогией файловой структуре скорее будет куст, чем дерево. От корня может идти множество веток-каталогов, но это более свойственно кустарнику. Путь, который надо пройти, чтобы найти файл, можно ассоциировать с маршрутом, по которому должна проползти гусеница, чтобы добраться до определенного листа или перебраться с одной ветки на другую.

Еще один полезный прием, позволяющий усвоить понятие «полное имя файла». Как известно, полное

имя файла включает в себя как его собственное имя, так и его полный адрес во внешней памяти: имя диска и путь к файлу на диске. Попросите учеников по аналогии назвать свое «полное имя». Правильным будет ответ такого типа: «Россия, Москва, Проспект Мира, дом 10, кв. 23, Сыроежкин Сергей Петрович». Полное имя включает в себя подробный адрес. Его записывают на конверте письма, чтобы оно нашло адресата. Так же по полному имени отыскиваются файлы на дисках.

Освоив навигацию по файловой структуре, нужно переходить к основным операциям с файлами и каталогами. Эти операции следует давать в таком порядке: просмотр содержимого файла, копирование и перемещение файла, создание каталога (папки), удаление файла и каталога, переименование файла и каталога. Работая с ОС Windows, приучайте учеников к приемам, характерным для этой объектной среды: методике Drag&Drop, использованию контекстного меню.

Безусловно, что за 1–2 занятия ученики не смогут хорошо закрепить практические навыки работы с файловой системой ОС. Основным результатом этих уроков будет получение представления об организации файлов и понимание возможностей работы с файлами. В последующих прикладных темах, при выполнении практических работ на компьютере в задания необходимо включать пункты, требующие от учеников умения работать с дисками и файлами (сохранять, копировать, переименовывать и пр.). Только в таком случае представления перейдут у них в прочные навыки.

В основе современных компьютерных технологий лежит объектно-ориентированный подход. Начало ему было положено появлением объектно-ориентированной технологии программирования — ООП. Затем появились операционные системы и прикладные программы с объектно-ориентированным пользовательским интерфейсом. С внедрением на IBM PC операционной системы Windows и ее приложений графический объ-

ектно-ориентированный интерфейс стал нормой. В § 13 раскрываются основные понятия объектно-ориентированного интерфейса: объект, свойства объекта, поведение объекта, контекстное меню. Наиболее вероятно, что именно с этой формой интерфейса ученики будут иметь дела, работая за компьютером.

9. Текстовая информация и компьютер

Разделы учебника: часть 1, § 14, 15, 16. **Дополнительный материал:** часть 2, раздел 3.1.

Основные цели. Познакомить учащихся со способами представления и организации текстов в компьютерной памяти. Раскрыть назначение текстовых редакторов. Обучить детей основным приемам работы с текстовым редактором.

Изучаемые вопросы

- ☐ Кодирование текстов.
- ☐ Структура текстового документа.
- ☐ Текстовые файлы.
- ☐ Текстовые редакторы и текстовые процессоры.
- ☐ Практическая работа с текстовым редактором.

Общие методические рекомендации

Настоящая тема является первой в базовом курсе, относящейся к содержательной линии «Компьютерные информационные технологии». В каждом тематическом разделе этой линии учитель должен четко различать теоретическое и технологическое содержание. Теоретическое содержание включает в себя вопросы представления различных видов информации в памяти ЭВМ, структурирования данных, постановки и методов решения информационных задач с помощью технологических средств данного типа. Сюда же следует отнести более подробное изучение принципов работы отдельных

устройств компьютера, расширяющее представления учащихся об архитектуре ЭВМ. Технологическое содержание — это знакомство и освоение приемов работы с конкретными прикладными программными системами: редакторами, СУБД, табличными процессорами и пр. Для этих целей необходимо использовать дополнительную литературу. Желательно, чтобы она была ориентирована на применение в школе. Одной из таких книг является учебное пособие [24].

Знакомство учеников с каждым новым для них видом информационных технологий должно начинаться с рассказа об их областях применения. Желательно, чтобы изучение каждого прикладного программного средства затрагивало следующие его стороны: данные, среду (интерфейс), режим работы, команды управления. Предлагаемая методическая схема отображена на рис. 9.1.

К теоретическим основам компьютерных технологий работы с текстом относятся вопросы кодирования текстовой информации, структуры текстовых документов и текстовых файлов.

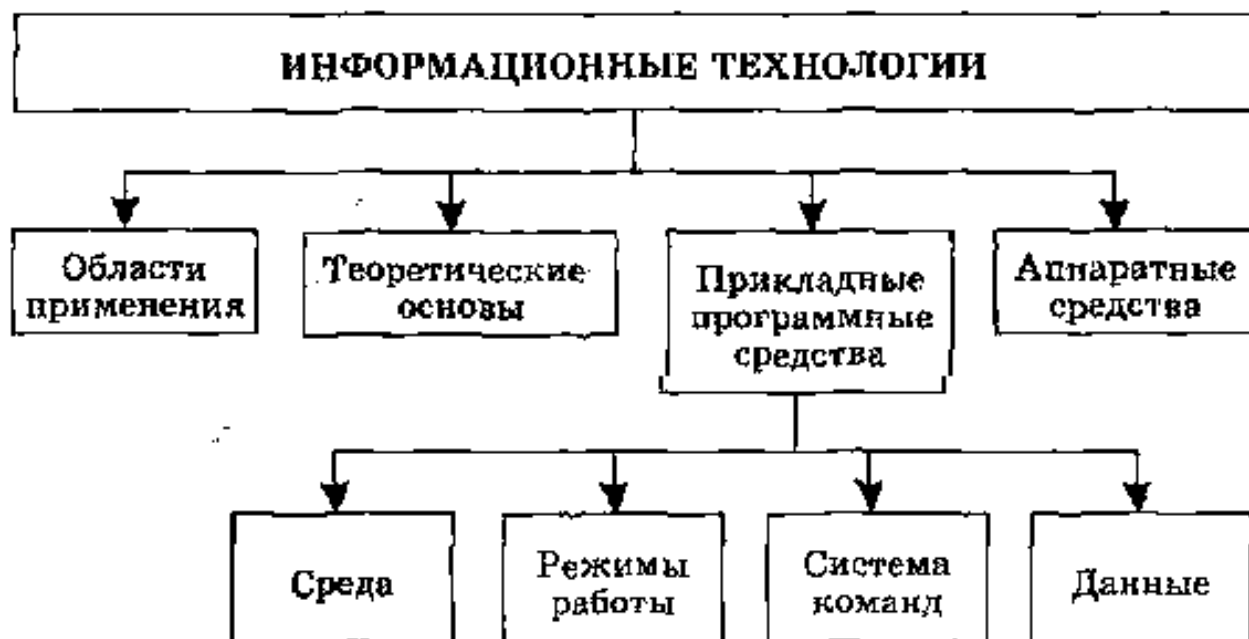


Рис. 9.1.

Методическая схема изучения информационных технологий

Методические рекомендации по изложению теоретического материала

1. Начать объяснение следует с напоминания того факта, что компьютер может работать с четырьмя видами информации: текстовой, графической, числовой и звуковой. Одним из самых массовых приложений ЭВМ является работа с текстами. Термины «текстовая информация» и «символьная информация» используются как синонимы. С точки зрения компьютера текст — это любая последовательность символов из компьютерного алфавита. Совсем не обязательно, чтобы это был текст на одном из естественных языков (русском, английском и др.). Это могут быть математические или химические формулы, номера телефонов, числовые таблицы и пр. Самое главное, чтобы все используемые символы входили в *компьютерный алфавит*.

Первая задача — познакомить учеников с символьным алфавитом компьютера. Они должны знать, что

- алфавит компьютера включает в себя 256 символов;
- каждый символ занимает 1 байт памяти.

Эти свойства символьного алфавита компьютера, в принципе, уже знакомы ученикам. Изучая алфавитный подход к измерению информации, они узнали, что один символ из алфавита мощностью 256, несет 8 битов, или 1 байт информации, потому что $256 = 2^8$. Но поскольку всякая информация представляется в памяти ЭВМ в двоичном виде, то, значит, каждый символ представляется 8-разрядным двоичным кодом. Существует 256 всевозможных 8-разрядных комбинаций, составленных из двух цифр «0» и «1» (в комбинаторике это называется числом размещений из 2 по 8 и равно 2^8): от 00000000 до 11111111. Удобство побайтового кодирования символов очевидно, поскольку байт — наименьшая адресуемая часть памяти и, следовательно, процессор может обратиться к каждому символу отдельно, выполняя обработку текста. С другой стороны, 256 символов — это

вполне достаточное количество для представления самой разнообразной символьной информации.

Далее следует ввести понятие о таблице кодировки. *Таблица кодировки — это стандарт, ставящий в соответствие каждому символу алфавита свой порядковый номер.* Наименьший номер — 0, наибольший — 255. Двоичный код символа — это его порядковый номер в двоичной системе счисления. Международным стандартом для персональных компьютеров стала таблица ASCII. На практике можно встретиться и с другой таблицей — КОИ-8 (Код Обмена Информацией), которая используется в глобальных компьютерных сетях, а также на компьютерах типа PDP. К ним, в частности, относится отечественный школьный компьютер «Электроника-УКНЦ».

От учеников не нужно требовать запоминания кодов символов. Однако некоторые принципы организации кодовых таблиц они должны знать. Рассмотрите вместе с учениками таблицу кода ASCII, приведенную в учебнике. Она делится на две части. Международным стандартом является лишь первая половина таблицы, то есть символы с номерами от 0 до 127. Сюда входят строчные и прописные буквы латинского алфавита, десятичные цифры, знаки препинания, всевозможные скобки, коммерческие и другие символы. Символы с номерами от 0 до 31 принято называть *управляющими*. Их функция — управление процессом вывода текста на экран или печать, подача звукового сигнала, разметка текста и т. п. Символ с номером 32 — пробел, то есть пустая позиция в тексте. Все остальные символы отражаются определенными знаками. Важно обратить внимание учеников на соблюдение *лексикографического порядка* в расположении букв латинского алфавита, а также цифр. На этом принципе основана возможность сортировки символьной информации, с которой ученики впервые встретятся, работая с базами данных.

Могут существовать различные варианты второй половины кодовой таблицы. В первую очередь она ис-

пользуется для размещения национальных алфавитов, отличных от латинского. Поскольку для кодировки русского алфавита — кириллицы применяются разные варианты таблиц, часто возникают проблемы с переносом русского текста с одного компьютера на другой, из одной программной системы в другую. Можно сообщить ученикам, что таблица кодировки символов 128–255 называется *кодовой страницей* и каждый ее вариант имеет свой номер. Так, например, в MS DOS используется кодовая страница номер 866, а в Windows — номер 1251. Последняя приведена в таблице 3.1 учебника.

В качестве дополнительной информации можно рассказать о том, что проблема стандартизации символического кодирования решается введением нового международного стандарта, который называется Unicode. Это 16-разрядная кодировка, то есть в ней на каждый символ отводится 2 байта памяти. Конечно, при этом объем занимаемой памяти увеличивается в 2 раза. Но зато такая кодовая таблица допускает включение до 65 536 символов. Ясно, что в нее можно внести всевозможные национальные алфавиты.

В разделе 3.1.3 «Представление символьной информации» пособия [2] имеется подборка задач по данной теме. Приведенные в этом же разделе примеры поясняют способы решения этих задач. Их решение требует от учеников знания принципов кодирования текстов, умения осуществлять переходы между десятичными, двоичными и шестнадцатеричными кодами.

В разделе 1.3 части II учебника рассматривается иной подход к проблеме кодирования текста в компьютерной памяти. Этот подход основан на применении алгоритма Д. Хаффмена. Суть его состоит в том, что длина двоичного кода у разных символов может быть разной. Чем чаще символ встречается в тексте, тем его код короче, и наоборот — чем символ более «редкий», тем его код длиннее. Перекодирование текста из байтового кода путем применения алгоритма Д. Хаффмена позво-

дает существенно сокращать объем памяти, занимаемый текстом.

2. Текстовым документом будем называть файл, созданный с помощью текстового редактора. Текстовый документ должен иметь определенную структуру. Он должен делиться на строки, абзацы, страницы, разделы; может иметь выделенные заголовки, нумерацию страниц, колонтитулы и пр. Именно в таком виде пользователь желает видеть текст на экране или распечатанным на бумаге. Для компьютера же текст — это непрерывная последовательность байтов. Внешний вид, который приобретает текст при выводе на экран или бумагу, называется *форматом текста*. Информацию о формате текста несут управляющие символы, те самые, которые расположены в начале кодовой таблицы. Среди них есть символы, отмечающие конец строки, конец страницы, конец файла. Есть символы, переводящие к началу новой строки курсор при выводе на экран или печатающую головку при выводе на печать. Имеются символы, обеспечивающие пропуск строки, прогон листа бумаги; управляющие абзацным отступом, табуляцией и др.

Специфическим свойством структуры для компьютерного текста является наличие гиперсвязей. В § 14 объясняется смысл этого понятия. Можно сказать так: гиперструктура документа — это система ссылок по ключевым словам, присутствующим в тексте. Работая в ОС Windows, следует познакомить учеников с гипертекстом на примере встроенной справочной информации как для самой операционной системы, так и для ее приложений.

3. Текстовые файлы — очень важный для компьютера способ организации данных. Большое количество системной информации хранится в текстовых файлах (файл конфигурации, файл автозагрузки и др.). Тексты программ на языках программирования высокого уровня (Паскале, Бейсике и др.) также хранятся в

текстовых файлах. Тексты писем, передаваемые по электронной почте, представляют собой текстовые файлы.

Для работы с текстовыми файлами существуют простые текстовые редакторы. На IBM PC это редакторы MS-DOS Editor, Norton Editor; в операционной системе Windows таким редактором является Блокнот (не забывайте, что MS DOS и Windows используют разные кодовые страницы и поэтому их текстовые файлы не совместимы по русскому алфавиту). Часто в именах текстовых файлов используется расширение txt.

4. Текстовый редактор (ТР) — первый тип прикладной программы, который должны освоить ученики. Выбор ТР, используемого в учебных целях, зависит от технического и программного обеспечения школьного компьютерного класса. При использовании IBM PC и операционной системы MS DOS целесообразно познакомить учащихся с одним из простых текстовых редакторов (например, Norton Editor) и каким-нибудь более совершенным редактором: «Лексикон», Word и др. Если в классе работает операционная система Windows, то начать можно с ТР Блокнот. Использование на следующем этапе текстового процессора MS Word совсем не обязательно. Безусловно, MS Word — это один из наиболее развитых ТР, в котором реализована самая современная технология обработки текста. Тем не менее, есть два аргумента, ставящих под сомнение необходимость использования Word. Во-первых, MS Word входит в состав дорогостоящего офисного программного пакета Microsoft Office, который может оказаться «не по карману» для законопослушных учебных заведений, допускающих использование только лицензионного программного обеспечения (к сожалению, такой аргумент вызовет лишь улыбку у большинства наших читателей). Второй аргумент — методического свойства. MS Word — очень сложный программный продукт. За время, отведенное под данную тему учебной програм-

мой, нельзя освоить даже половину его возможностей. Многие элементы интерфейса Word (пункты меню, инструменты, кнопки) оказываются невостребованными и создают определенный «информационный шум», отвлекающий учеников. Работа с MS Word на компьютерах с относительно медленным процессором и небольшим объемом оперативной памяти, как правило, происходит с заметными задержками. Это слишком громоздкий текстовый процессор для использования в учебных целях.

Отмеченные выше проблемы снимает использование текстового редактора WordPad, который всегда имеется в стандартной поставке Windows. Практически все приемы работы с текстовыми редакторами, которые можно освоить в рамках базового курса, реализованы в WordPad. Интерфейс этого редактора близок к интерфейсу Word, но «без излишеств».

Освоив WordPad, ученики легко смогут перейти к работе с MS Word. При наличии учебного времени такой переход можно произвести в конце данной темы, при необходимости познакомить учащихся с некоторыми возможностями текстовых процессоров, не поддерживаемых WordPad (например, многооконный режим работы, проверка орфографии, работа с таблицами).

Можно говорить о том, что компьютер, на котором запущен на исполнение текстовый редактор, становится специализированным исполнителем для работы с текстовой информацией...

$\text{ЭВМ} + \text{ТР} \Rightarrow \text{исполнитель работы с текстами}$

Этот исполнитель работает под управлением человека (пользователя). Все действия, которые выполняет человек, работая с такой системой, можно рассматривать как *команды управления исполнителем*: ввести символ, удалить символ, занести фрагмент в буфер, сохранить текст в файле и пр. Отсюда следует, что изучение текстового редактора (как и любого другого прикладного средства информационных технологий) мож-

но проводить по традиционной методической схеме, свойственной для изучения всякого исполнителя:

- данные: структурные единицы текста;
- среда ТР;
- режимы работы ТР;
- система команд.

В § 15–16 учебника в такой последовательности приводится инвариантное описание свойств и возможностей текстовых редакторов. Это описание дает некоторое итоговое представление, которое должно сложиться в сознании учащихся в результате изучения темы. Реально же на уроках происходит изучение и практическое освоение конкретных текстовых редакторов. Материал учебника дает опорные понятия, которые должны быть конкретизированы и проиллюстрированы на изучаемых ТР.

Дадим некоторое уточнение понятию «режим работы». *Под режимом работы понимается определенное состояние исполнителя (в данном случае текстового редактора), в котором возможно выполнение определенного вида работы.* Режим ввода/редактирования — основной режимом работы ТР. Признаком того, что ТР находится в этом режиме, является наличие курсора на рабочем поле (для Word — мигающая вертикальная черточка). В этом режиме можно вводить и удалять символы, слова, строки; перемещаться по тексту; разрывать и сливать строки. После выделения фрагмента текста происходит переход в режим работы с фрагментом: его можно переместить, вырезать, копировать, вставить, переформатировать. Но поскольку работа с фрагментами происходит в процессе редактирования текста, то можно считать, что режим работы с фрагментом является подрежимом режима ввода/редактирования. Аналогично форматирование можно рассматривать как подрежим ввода/редактирования. А вот работа с файлами — это самостоятельный режим. В нем можно открывать документы и сохранять файлы на диске. В старых версиях текстовых процессоров орфографический конт-

роль осуществлялся в отдельном режиме работы только по специальной команде. В современных ТР режим орфографического контроля можно совмещать с вводом текста. В таком случае появление ошибки сразу же отмечается подчеркиванием или звуковым сигналом.

Процесс создания текстового документа с помощью ТР носит комплексный характер: в нем задействованы все основные устройства компьютера. В рамках данной темы ученики должны не только развить практические навыки работы с различными аппаратными компонентами ЭВМ, но и углубить свои знания об их устройстве, принципах их работы. В этом проявляется пересечение содержательной линии «Компьютер» с линией информационных технологий.

Клавиатура — основное устройство для ввода текста, управления текстовым редактором. Знакомство с клавиатурой с помощью клавиатурных тренажеров позволяет ученикам освоить только ее центральную часть — символные клавиши. При работе с ТР активно задействованы все остальные группы клавиш:

- функциональные;
- управляющие;
- переключения режимов;
- перемещения курсора;
- прочие клавиши.

Дисплей. Во-первых, ученики должны узнать, что любое изображение на экране получается из совокупности светящихся точек — пикселей. Символы текста формируются не из непрерывных линий, а из отдельных точек. Эту дискретную структуру символа можно разглядеть при внимательном рассмотрении экрана.

Операционная система MS DOS и все ее приложения различают два режима работы дисплея: символный и графический. Все текстовые редакторы, работающие под DOS, используют *символьный режим экрана*. В этом режиме на экране строго определены позиции и размеры выводимых символов. Можно говорить о том,

что экран разделен на клеточки, выстроенные в строки и столбцы, подобно листу из тетради в клетку. Каждая такая клетка экрана — знакоместо для одного символа. В каждой клетке содержится одинаковое число пикселей, совокупность которых называют матрицей пикселей. Изображение символа создается свечением определенного сочетания пикселей в пределах знакоместа. Например, если размер знакоместа 8×10 пикселей (первое число — горизонтальный размер, второе — вертикальный), а размер всей сетки пикселей экрана 640×200 , то на экране помещается 20 символьных строк, а в каждой строке по 80 символов.

Текстовые редакторы, ориентированные на символьный режим экрана, не позволяют изменять размеры символов, использовать различные типы шрифтов. Существует один стандартный шрифт. В наиболее развитых ТР можно лишь менять начертания символов: прямой, курсив, полужирный. Некоторые принтеры предоставляют возможность менять шрифт в печатном документе. Для этого они обладают небольшим набором встроенных шрифтов.

Операционная система Windows и все ее приложения работают с дисплеем только в графическом режиме. Следствием этого является то, что при выводе на экран текста отсутствуют понятия: фиксированное знакоместо, стандартный размер символов, стандартный тип шрифта. Символы могут выводиться на экран в разных позициях, быть разных размеров и форм. Однако дискретная пиксельная структура экрана остается, и как следствие — мозаичный принцип изображения символов.

Память ЭВМ. При работе над текстовым документом задействована как внутренняя (оперативная) так и внешняя память компьютера. Ученики должны получить представление, для каких целей используется тот и другой вид памяти.

Сам текстовый редактор (как и все программное обеспечение ПК) хранится на магнитном диске. Следу-

ет указать ученикам, в каком каталоге (папке) находится ТР, объяснить, что запуск ТР происходит путем переписывания содержимого файла с программой редактора в оперативную память и инициализации его выполнения. В течение всей работы программа ТР занимает определенную часть памяти. Память освобождается только после выхода, закрытия ТР.

Вводимый пользователем текст заносится в специально отведенную для этого область оперативной памяти — *рабочую область ТР*. Еще один раздел оперативной памяти занимает *буфер для копирования* фрагментов текста. В него попадают фрагменты, с которыми выполняются команды [Вырезать-Копировать] (забрать в буфер). Буфер хранит лишь один занесенный в него фрагмент. При выполнении вставки фрагмента в текст содержимое буфера сохраняется. При копировании следующего фрагмента предыдущий из буфера удаляется. В последних версиях MS-Word появилась возможность сохранять в буфере одновременно несколько фрагментов текста.

Сохранение текста, сформированного в рабочей области оперативной памяти, ведет к созданию *файла документа* на магнитном диске. Пользователю предоставляется возможность указать место сохранения файла на диске (имя диска, каталог) и задать имя файла. Некоторые ТР присваивают таким файлам стандартные расширения. Это характерно для всех приложений Windows. Например, MS Word создает файлы с расширением doc. Текст сохраненного на диске документа может быть снова вызван в рабочую область текстового редактора при выполнении команды [Открыть].

Для поиска в тексте грамматических ошибок на магнитном диске хранится *файл с орфографическим словарем*. Причем таких словарей может быть несколько для контроля текстов на разных языках; например, русский словарь и английский словарь. Справочник по работе с ТР также постоянно хранится на магнитном диске.

При работе с принтером следует сообщить ученикам, что у принтера есть своя небольшая буферная память. Вывод документа на печать происходит путем заполнения *буфера принтера*, а затем из буфера текст переносится на бумагу.

Рекомендации по организации практической работы на компьютере

Навыки ввода и редактирования текста нужны пользователю не только для работы с текстовым редактором. Эти навыки являются базовыми при работе с любыми видами программного обеспечения, где используется интерактивный режим. Набирать тексты приходится при вводе команд операционной системы, при ответах на запросы в диалоговых окнах системных и прикладных программ, при вводе информации в базы данных и электронные таблицы и пр. При любом варианте символьного ввода пользователю предоставляется возможность исправления ошибок, то есть редактирования текста. Существует понятие *основного стандарта редактирования*. Это набор операций редактирования, которые выполняются одинаково для всех вариантов символьного ввода. Прежде всего, ученики должны освоить приемы основного стандарта редактирования. К ним относятся:

1. Перемещение по тексту: клавиши {←}, {↑}, {→}, {↓}, {Home}, {End}, {PageUp}, {PageDown}.
2. Удаление символа: клавиши {Backspace}, {Del}.
3. Вставка/замена символа: переключение режима с помощью клавиши {Insert}.
4. Разрыв и слияние строк: клавиша {Enter}; клавиша {Del} в конце первой соединяемой строки или {Backspace} в начале второй строки.

Практическое знакомство с перечисленными приемами редактирования может начаться раньше изучения темы «Текстовые редакторы», параллельно с изучением таких теоретических вопросов, как архитектура ЭВМ и программное обеспечение компьютера. Для

отработки навыков следует использовать простейшие ТР типа MS DOS Editor или Блокнот.

В разделе 5.1 «Работа с текстом» пособия [2] содержится большой материал для организации практических занятий по текстовым информационным технологиям. Порядок расположения материала соответствует рекомендуемой последовательности проведения практических занятий. Предлагаемые в этом разделе упражнения предназначены для отработки отдельных приемов и методов работы в текстовом редакторе. Эти упражнения могут предлагаться ученикам в качестве общих заданий для одновременного выполнения на компьютерах. Индивидуальные работы многовариантны. Учитель по своему усмотрению может либо дать всем один и тот же вариант, либо распределить разные варианты между учениками. Предполагается, что индивидуальная работа будет оцениваться в баллах.

Для некоторых индивидуальных заданий со стороны учителя требуется подготовительная работа. Она состоит в подготовке файлов, с которыми будут работать ученики. Содержание этих файлов приведено в приложении к пособию [2].

В полном объеме материал раздела 5.1 ориентирован на углубленный вариант изучения базового курса информатики, на использование развитых текстовых процессоров типа MS Word. В других режимах обучения этот материал оказывается избыточным. Поэтому учителю следует выбрать из него то подмножество заданий, которое соответствует целям обучения, имеющимся техническим и программным средствам и резерву учебного времени для данного раздела базового курса.

Учитель должен ясно представлять последовательность педагогических целей, которые ставятся перед учащимися в процессе формирования умений и навыков для работы с программными средствами информационных технологий. В рамках темы «информационные технологии работы с текстом» указанные цели могут быть представлены следующим ниже списком

(список составлен «по максимуму»). Порядок пунктов в этом списке отражает рекомендуемую последовательность обучения.

- Поиск и запуск программы ТР; обращение за справкой.
- Набор русского текста; переход на верхний/нижний регистр (строчные/прописные буквы); знаки препинания (в Windows не всегда совпадают с обозначениями на клавишах); перевод строки.
- Переход «рус/лат».
- Простейшие приемы редактирования: клавиши {Del}, {Backspace}; режимы вставки и замены.
- Открытие файла с текстом; просмотр текста, приемы перемещения по тексту.
- Редактирование данного текста: поиск и исправление ошибок; разрыв строки, слияние строк.
- Сохранение документа на диске.
- Установка параметров формата: размеров полей, длины строки, межстрочного интервала, абзацного отступа, выравнивания строк.
- Работа с фрагментами: выделение фрагмента, перемещение, копирование через буфер обмена.
- Работа со шрифтами: установка типа шрифта, размера, начертания; изменение шрифта выделенного фрагмента.
- Поиск и замена.
- Работа в многооконном режиме; перемещение фрагментов между различными документами.
- Создание и редактирование таблиц.
- Формирование гиперссылок.

Тема «Текстовая информация и компьютер» занимает особое место в базовом курсе. Это первая прикладная тема. Здесь ученикам предстоит впервые самостоятельно создать небольшой реальный продукт с помощью компьютера — текстовый документ. Это обстоятельство, безусловно, повышает мотивацию учеников к работе. На данную тему в учебном плане выделяется 7–8 уроков. Это

учебное время следует разумно распределить между теорией и практикой. Учитель не должен поддаваться напору наиболее нетерпеливых учеников, не желающих слушать никаких объяснений и рвущихся скорее начать «нажимать клавиши». Опыт показывает, что большинство детей плохо воспринимают объяснения, сидя за работающим компьютером. Поэтому наиболее подходящий вариант организации занятий такой, когда объяснения учителя дети слушают, сидя за партами без машин, и лишь получив задания и четко уяснив, что требуется сделать, ученики садятся за компьютеры. Далее работа учителя переходит в режим индивидуального общения с учениками, консультаций, приема заданий.

Следует учитывать, что у детей еще довольно слабые навыки работы с клавиатурой, мышью, дисками; нечеткие представления о работе с файлами. Если ученикам предоставляется возможность вывода текста на печать, то для многих из них это будет первый опыт работы с принтером. Таким образом, практические задания данного раздела носят комплексный обучающий характер. Учитель должен обращать внимание на все стороны работы, активно помогать ученикам.

10. Графическая информация „Компьютер“

Разделы учебника: часть I, § 17–20. Дополнительный материал: часть II, раздел 4.1.

Основные цели. Познакомить учащихся с назначением и областями применения компьютерной графики. Дать им представление об устройстве и функционировании видеосистемы компьютера. Раскрыть способы кодирования графического изображения. Обучить школьников основным приемам работы с графическим редактором.

Изучаемые вопросы

- Области применения компьютерной графики.
- Аппаратные компоненты видеосистемы компьютера.
- Кодирование изображения.
- Графические редакторы.
- Практическая работа с графическим редактором.

Методические рекомендации по изложению теоретического материала

1. Данный раздел базового курса относится еще к одной области компьютерных информационных технологий — технологии работы с графической информацией. Необходимо отметить, что компьютерная графика — это сравнительно новая область применения ЭВМ. Машины первых двух поколений работали только с числовой и символьной информацией. В период третьего поколения ЭВМ появились средства машинной графики, но, в основном, они носили специализированный характер, требовали использования специальных технических и программных средств. Компьютерная графика стала массовой, серийной только во времена четвертого поколения ЭВМ, в период распространения персональных компьютеров.

Материал § 17 вводит учащихся в мир компьютерной графики. Безусловно, это область, которую хочется реально увидеть, а не слушать разговоры о ней. Поэтому большое значение имеют демонстрации на компьютере разнообразных продуктов компьютерной графики: красочных рисунков, схем, чертежей, диаграмм, образцов анимационной и трехмерной графики. Следует обратить внимание учеников на то, что любимые многими из них компьютерные игры в большинстве имеют графический интерфейс, причем, достаточно сложный. Программы, с помощью которых на компьютере получается трехмерное реалистическое изображение, изобилуют математическими расчетами. Программирование графики — одна из самых сложных об-

частей в современном программировании. Благодаря существованию *прикладных графических пакетов*, компьютерная графика стала доступна широкому кругу пользователей.

К теоретическому содержанию данного раздела базового курса следует отнести:

- вопросы состава и функционирования технических средств компьютерной графики (продолжение линии компьютера);
- вопросы представления изображения в памяти компьютера.

2. В § 18 дается описание состава и принципов работы технических средств компьютерной графики. Следует напомнить учащимся, что при первом знакомстве с устройством компьютера говорилось о том, что работой каждого внешнего устройства ПК управляет специальный контроллер. Основным устройством вывода графических изображений является *дисплей*. Работой дисплея управляет *видеоконтроллер*. Употребляется также другой термин для обозначения этого устройства — *видеоадаптер*; в комплекте устройств ПК его еще называют *видеокартой*.

Основные представления об устройстве дисплея, которые должны извлечь ученики из этого материала: *дискретная (пиксельная) структура экрана; сетка пикселей (растр); сканирование раstra электронным лучом; частота сканирования; трехцветная структура пикселя цветного монитора*. Данный материал изобилует физическими понятиями: *электронный луч, люминесценция, смешение трех базовых цветов*. Эти понятия относятся к областям электроники и физической оптики, еще не знакомым ученикам из курса физики. Не следует долго и подробно задерживаться на этих вопросах. Вполне достаточно того описательного уровня объяснения, который приведен в учебнике. Впоследствии в старших классах на уроке физики ученики подробно узнают о сути данных физических явле-

ний. Первое же знакомство с ними на уроке информатики станет своеобразной пропедевтикой и, кроме того, хорошей иллюстрацией системности научных знаний.

Материал данного раздела позволяет «заглянуть внутрь» видеоконтроллера. Как и раньше, это знакомство происходит на уровне архитектуры, то есть не изучаются вопросы технической реализации, а дается лишь функциональное описание. С этой точки зрения видеоконтроллер состоит из двух частей: видеопамяти и дисплейного процессора. Ученикам следует дать представление о роли этих устройств в процессе получения изображения на экране.

Основной универсальный для ЭВМ принцип заключается в том, что компьютер работает с информацией, хранящейся в его памяти в двоичном виде. Следовательно, всякое изображение на экране — это отражение информации в памяти ЭВМ — видеоинформации. Первоначально видеоинформация формируется в оперативной памяти (при открытии графического файла, при рисовании в графическом редакторе). Вывод на экран происходит в результате передачи видеоинформации контроллеру монитора: информация записывается в видеопамять и сразу же воспроизводится на экране вследствие непрерывной работы дисплейного процессора, управляющего работой монитора (рис. 10.1). Таким образом, видеопамять является своеобразным

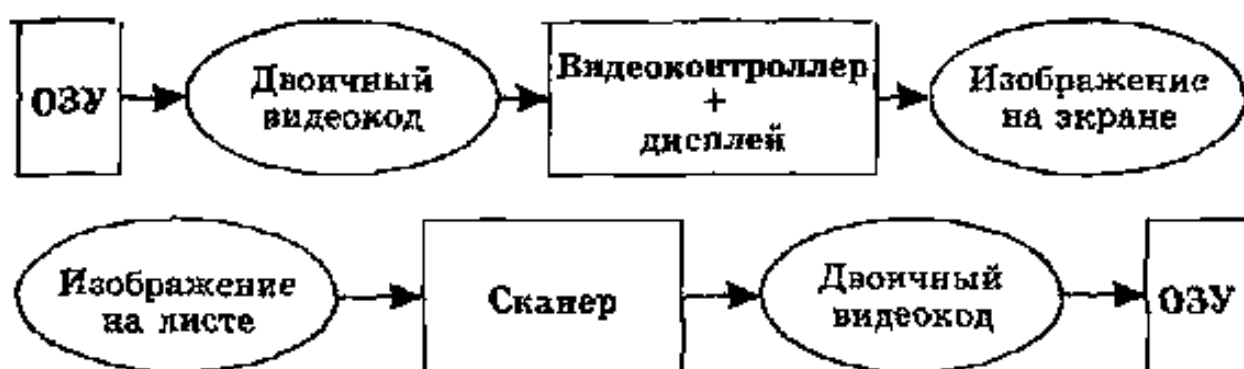


Рис. 10.1.

Преобразование видеоинформации при выводе на экран и при сканировании.

буфером между ОЗУ и дисплеем. Смена «картинки» на экране является следствием смены содержимого видеопамати. Ученики должны понять, что система вывода на экран работает совершенно одинаково, независимо от того, какого рода информация выводится: текст ли это, неподвижный рисунок или анимация.

В качестве устройства ввода изображения с листа в компьютерную память используется *сканер*. Следует подчеркнуть взаимообратную функцию системы вывода изображения на экран и системы ввода изображения с помощью сканера (см. рис. 10.1).

3. Главным теоретическим вопросом данного раздела информатики является вопрос о представлении изображения в памяти компьютера, то есть вопрос о том, что такое видеoinформация. Существуют два подхода к проблеме представления изображения: *растровый* и *векторный*. Суть обоих подходов в декомпозиции, то есть разбиении изображения на части, которые легко описать. *Растровый подход* предполагает разбиение изображения на маленькие одноцветные элементы — *видеопиксели*, которые сливаясь дают общую картину. В таком случае видеoinформация представляет собой перечисление в определенном порядке цветов этих элементов. *Векторный подход* разбивает всякое изображение на геометрические элементы: отрезки прямой, эллиптические дуги, фрагменты прямоугольников, окружностей, области однородной закрашки и пр. При таком подходе видеoinформация — это математическое описание перечисленных элементов в системе координат, связанной с экраном дисплея. Векторное представление более всего подходит для чертежей, схем, штриховых рисунков.

Нетрудно понять, что растровый подход универсальный, то есть он применим всегда, независимо от характера изображения. Из того, что говорилось выше о видеопамати, о дискретной структуре экрана монитора, следует, что в видеопамати любое изображение

представляется в растровом виде. На современных ПК используются только растровые дисплеи, работающие по принципу построчной развертки изображения.

В первой части учебника описывается лишь растровая графика. Сведения о векторной графике даются в дополнительном материале в разделе 4.1. Там же приводится сравнительный анализ растровой и векторной графики.

После того, как ученики узнают, что видеoinформация — это совокупность кодов цвета пикселей экрана, следует рассмотреть способы кодирования цветов. Принципы кодирования цветов описаны в § 19. Как уже было сказано, физический принцип получения разнообразных цветов на экране дисплея заключается в смешивании трех основных цветов: красного, зеленого и синего. Значит, информация, заключенная в коде пикселя, должна содержать сведения о том, какую интенсивность (яркость) имеет каждая составляющая в его цвете.

Необходимо раскрыть перед учениками связь между кодом цвета и составом смеси базовых цветов. Следует начать с рассмотрения варианта восьмицветной палитры. В этом случае используется трехбитовый код и каждый бит такого кода обозначает наличие (1) или отсутствие (0) соответствующего базового цвета. В таблице 4.1 приведены коды восьмицветной палитры. Биты в таком коде распределены по принципу «КЗС», то есть первый бит отвечает за красную составляющую, второй — за зеленую, третий — за синюю. По этой теме ученики должны уметь отвечать на вопросы такого типа:

— *Смешиванием каких цветов получается розовый цвет, если его код 101?*

— *Известно, что коричневый цвет получается смешиванием красного и зеленого цветов. Какой код у коричневого цвета?*

При программировании цветных изображений принято каждому цвету ставить в соответствие десятичный номер. Получить номер цвета очень просто. Для

этого его двоичный код, рассматривая как целое двоичное число, следует перевести в десятичную систему счисления. Тогда, согласно таблице 4.1, номер черного цвета — 0, синего — 1, зеленого — 2 и так далее. Белый цвет имеет номер 7. Полезными, с точки зрения закрепления знаний двоичной системы счисления, являются задания такого рода:

— *Не глядя в таблицу, назвать десятичный номер красного цвета.*

Только после того, как ученики разберутся с восьмицветной палитрой, можно переходить к рассмотрению кодирования большего числа цветов. В таблице 4.2 даются коды 16-цветной палитры. Это те же восемь цветов, но имеющие два уровня яркости. Управляет яркостью дополнительный четвертый бит — бит интенсивности. Структура 16-цветного кода — «ИКЗС». Здесь И — бит интенсивности. Палитры большего размера получаются путем раздельного управления интенсивностью каждого из трех базовых цветов. Для этого в коде цвета под каждый базовый цвет выделяется более одного бита. Например, структура восьмибитового кода для палитры из 256 цветов такая: «КККЗЗЗСС», то есть по 3 бита кодируют красную и зеленую составляющие и 2 бита — синюю. В результате красная и зеленая составляющие имеют по 8 (2^3) уровней интенсивности, а синяя — 4 (2^2). Всего: $8 \times 8 \times 4 = 256$ цветов.

Связь между разрядностью кода цвета — b и количеством цветов — K (размером палитры) выражается формулой: $K = 2^b$. В литературе по компьютерной графике величину b принято называть *битовой глубиной цвета*. Следует раскрыть связь между величинами битовой глубины, разрешающей способностью графической сетки (размером раstra) и объемом видеопамяти. Если обозначить минимальный объем видеопамяти в битах через V_m , разрешающую способность дисплея — $M \times N$, то связь между ними выразится формулой:

$$V_m = b \times M \times N.$$

Для закрепления теоретических знаний по данной теме в разделе 3.1.5 задачника-практикума содержится подборка задач. Задачи могут быть использованы в качестве домашних заданий, в контрольных работах и тестах. В задачнике приведено достаточное количество примеров, демонстрирующих способы решения всех основных типов задач данного раздела.

4. Существует множество прикладных программ, предназначенных для работы с графикой. Для каждого раздела компьютерной графики имеются свои программы. Например, для графической обработки научных данных используется программа Grapher; инженеры-конструкторы для подготовки чертежей пользуются пакетом AutoCAD; существуют специализированные пакеты деловой графики, предназначенные для построения диаграмм, отражающих всевозможные статистические данные.

Название «графический редактор» применяется по отношению к прикладным программам, не имеющим какой-либо специализированной ориентации и используемым для «произвольного рисования» или редактирования сканированных изображений. В соответствии с двумя принципами представления графической информации — растровым и векторным, графические редакторы делятся на растровые и векторные редакторы.

К числу простейших растровых редакторов относятся Paintbrush и Paint (второй стал результатом развития первого). Растровый редактор Adobe Photoshop используется профессиональными дизайнерами. Чаще всего его применяют для редактирования сканированных изображений (фотографий, репродукций картин), создания художественных композиций, коллажей и пр.

Для профессионального рисования на компьютере используются редакторы векторного типа. Наиболее известным из них является CorelDRAW. Это профессиональный редактор с богатыми возможностями и, в то же время, вполне подходящий для «детского художественного творчества».

В стандартной поставке Windows в группе *Стандартные* имеется графический редактор Paint. Paint является прямым «родственником» редактора Paintbrush, который работает в среде MS DOS. В базовом курсе информатики для практической работы с компьютерной графикой обычно используют один из этих редакторов. В профильном курсе компьютерной графики для получения рисованных изображений больше подходит CorelDRAW. Заметим, что профессиональные графические редакторы, такие как CorelDRAW, Adobe Photoshop — довольно дорогие программные продукты и потому не всем доступны.

Использование на уроках информатики редакторов типа Paint вполне достаточно с точки зрения учебных целей, стоящих перед базовым курсом. Растровый редактор позволяет наглядно продемонстрировать ученикам дискретную (пиксельную) структуру рисунка, дает возможность воздействовать на каждый отдельный элемент при увеличении масштаба в режиме прорисовки.

Еще одним полезным учебным элементом является возможность демонстрации механизма смешения цветов. Для этого в главном меню Paint нужно выбрать команду [Палитра-Изменить палитру-Определить цвет]. Появившееся окно является прекрасным инструментом для экспериментов с цветом. Пользователь может изменять оттенки, контрастность, яркость цвета и при этом наблюдать RGB-состав полученной краски. Создав свою краску, пользователь может включить ее в палитру на соответствующей панели редактора.

Принцип растрового рисования — закрашивание каждого отдельного пикселя рисунка. Для растровых редакторов характерно наличие таких инструментов, как Кисть, Карандаш, Ластик. Этих инструментов нет в векторных редакторах. Там рисунок создается только путем манипулирования с графическими примитивами: линиями, дугами, эллипсами и пр. Эти элементы рисунка могут быть в любой момент изменены: сжаты, растянуты, перевернуты, перемещены, удале-

ны. В растровых редакторах тоже используются графические примитивы. Но их применение скорее похоже на использование линеек, лекал, циркуля и др. чертежных инструментов при рисовании на бумаге. Один раз нарисованный такой элемент уже нельзя изменить. Его можно лишь стереть или отредактировать прорисовкой.

Как и в предыдущем разделе, знакомство с графическим редактором следует проводить по методической схеме виртуального исполнителя. Описание среды и режимов работы приведено в § 20 учебника. А что можно рассматривать в качестве данных при работе с графическим редактором? Вспомним, что данными называется обрабатываемая компьютером информация. Итоговой информацией является созданный рисунок. Но, с позиции растровой графики, рисунок — это совокупность разноцветных пикселей. Значит данными являются цвета, которые выбираются из палитры и разносятся по элементам графической сетки с помощью различных инструментов. В памяти ЭВМ они представляются соответствующими двоичными кодами.

Всякое действие пользователя: выбор инструмента, выбор цвета, проведение линии, стирание и пр. можно рассматривать как команду, выполняемую в соответствующем режиме. Из этих команд складываются алгоритмы работы в среде редактора. Правила выполнения определенных действий учитель может сообщать ученикам в форме алгоритмов. Вот несколько примеров таких алгоритмов, ориентированных на использование редактора Paint.

Пример 1. Алгоритм рисования прямой линии с помощью инструмента «Линия»

1. Выбрать цвет линии.
2. На панели инструментов выбрать линию.
3. Выбрать ширину линии.
4. Нарисовать линию (команда состоит из 4 шагов):

- 4.1. Установить курсор на место начальной точки.
- 4.2. Нажать левую кнопку мыши.
- 4.3. Протянуть курсор до конечной точки.
- 4.4. Отпустить кнопку мыши.

Пример 2. Алгоритм переноса фрагмента рисунка на новое место

1. Выбрать инструмент «Выделение произвольной области».
2. Выделить фрагмент:
 - 2.1. Установить курсор около фрагмента.
 - 2.2. Нажать левую кнопку мыши.
 - 2.3. Обвести линией выделяемый фрагмент.
 - 2.4. Отпустить кнопку мыши (вокруг фрагмента образуется пунктирный прямоугольный контур).
3. Перенести фрагмент на новое место:
 - 3.1. Установить курсор внутри контура и нажать левую кнопку мыши.
 - 3.2. Переместить фрагмент на новое место.
 - 3.3. Отпустить кнопку мыши.

Учитель сам может продолжить описание таких алгоритмов. Эффективным обучающим приемом является задание ученикам самим описать в форме алгоритма выполнение некоторых действий в графическом редакторе.

Рекомендации по организации практической работы на компьютере

Абсолютное большинство учеников с удовольствием рисуют в графическом редакторе. При выполнении практических заданий очень заметно проявляются различные способности детей к рисованию. Безусловно, информатика — не уроки рисования и не всякий учитель информатики обладает художественными способностями. Учитель должен считать своей целью раскрытие всех возможностей графического редактора как инструмента для рисования. У редакторов типа Paint

этих возможностей не так уж много, и за 5–6 уроков, выделяемых для данной темы, все их вполне можно раскрыть.

В разделе 5.2.1 задачника-практикума имеется подборка рисунков для изображения их в растровом редакторе. К сожалению, в книге рисунки черно-белые. Ученикам нужно предложить раскрасить их самостоятельно.

Следует объяснить ученикам, что рисование от руки с помощью инструментов «Карандаш» или «Кисть» обычно получается некачественным. Необходимо максимально использовать графические примитивы: прямые, дуги, овалы и пр. В рисунках, где есть симметрия, следует научить детей использовать повороты, отражения. В рисунках с повторяющимися фрагментами они должны научиться применять копирование. Очень эффективным приемом отработки рисунков в растровом редакторе является прорисовка деталей через увеличение масштаба рисунка (использование инструмента «Лупа»). Пример, приведенный в разделе 5.2.1, показывает технологию построения симметричных рисунков.

В разделе 5.2.2 дается практический материал для работы с векторным графическим редактором. Два приведенных примера демонстрируют технологию создания рисунка, в том числе и объемного. Картинки для рисования более интересные, чем в заданиях для растрового редактора. Опыт показывает, что даже дети без особых художественных наклонностей довольно удачно срисовывают непростые картинки, используя редактор CorelDRAW.

11. Передача информации в компьютерных сетях

Разделы учебника: часть I, § 21–24.

Основные цели. Дать ученикам представление о назначении и структуре локальных и глобальных сетей. Познакомить учащихся с основными информационными услугами сетей, с возможностями Интернета. Обучить детей способам обмена файлами в локальной сети компьютерного класса. Познакомить их со способами поиска информации в Иетернете (при наличии технических возможностей).

Изучаемые вопросы

- ☐ Локальная сеть, структура и назначение.
- ☐ Структура и назначение глобальных сетей.
- ☐ Что такое Интернет.
- ☐ Электронная почта; технология «клиент—сервер».
- ☐ Технические средства глобальных сетей.
- ☐ Информационные услуги Интернета и World Wide Web.
- ☐ Практическая работа в компьютерных сетях.

Компьютерные телекоммуникации — одна из наиболее динамично развивающихся областей информационных технологий. Ни у кого не вызывает сомнения то, что эта тема должна быть отражена в базовом курсе. По сравнению с другими разделами информационных технологий ее технологическая составляющая гораздо больше теоретической. Поэтому эффективность изучения данной темы сильно зависит от возможности организовать практическую работу учащихся с компьютерными сетями.

Тема компьютерных сетей обширна по числу понятий и может излагаться с разной степенью подробности. Раскрытие этой темы в главе 5 учебника носит краткий характер, хотя все основные понятия в ней затронуты. Главной целью настоящего раздела методиче-

ского пособия является предоставление учителю информации, дополняющей материал учебника.

Содержание данного раздела базового курса делится на две части по принципу деления компьютерных сетей на два типа:

- локальные сети;
- глобальные сети.

Методические рекомендации по изложению теоретического материала

1. Если компьютеры в вашем кабинете информатики объединены в локальную сеть, то это обстоятельство существенно облегчает изучение данной темы. Именно школьный компьютерный класс должен стать отправной точкой в разговоре о передаче информации в компьютерных сетях.

Определив компьютерную сеть как *систему компьютеров, связанных каналами передачи информации*, учитель демонстрирует такую систему на оборудовании компьютерного класса и сообщает, что такая сеть называется локальной. *Локальные компьютерные сети* — небольшие по масштабам и работают в пределах одного помещения, здания, предприятия. Возможно, что в школе действует локальная сеть, объединяющие компьютеры, установленные в разных помещениях: в учебных кабинетах, в кабинете директора, в бухгалтерии и др. Точно так же в локальную сеть часто объединяются различные отделы предприятий, фирм, учреждений.

Локальные сети (ЛС), в зависимости от назначения и технических решений, могут иметь различные структуры объединения компьютеров. Такую структуру еще называют конфигурацией, архитектурой, топологией сети. На рис. 11.1 показаны четыре типа конфигураций ЛС: кольцевая, радиальная (звезда), шинная и древовидная.

Существуют две основные цели в использовании локальных сетей:

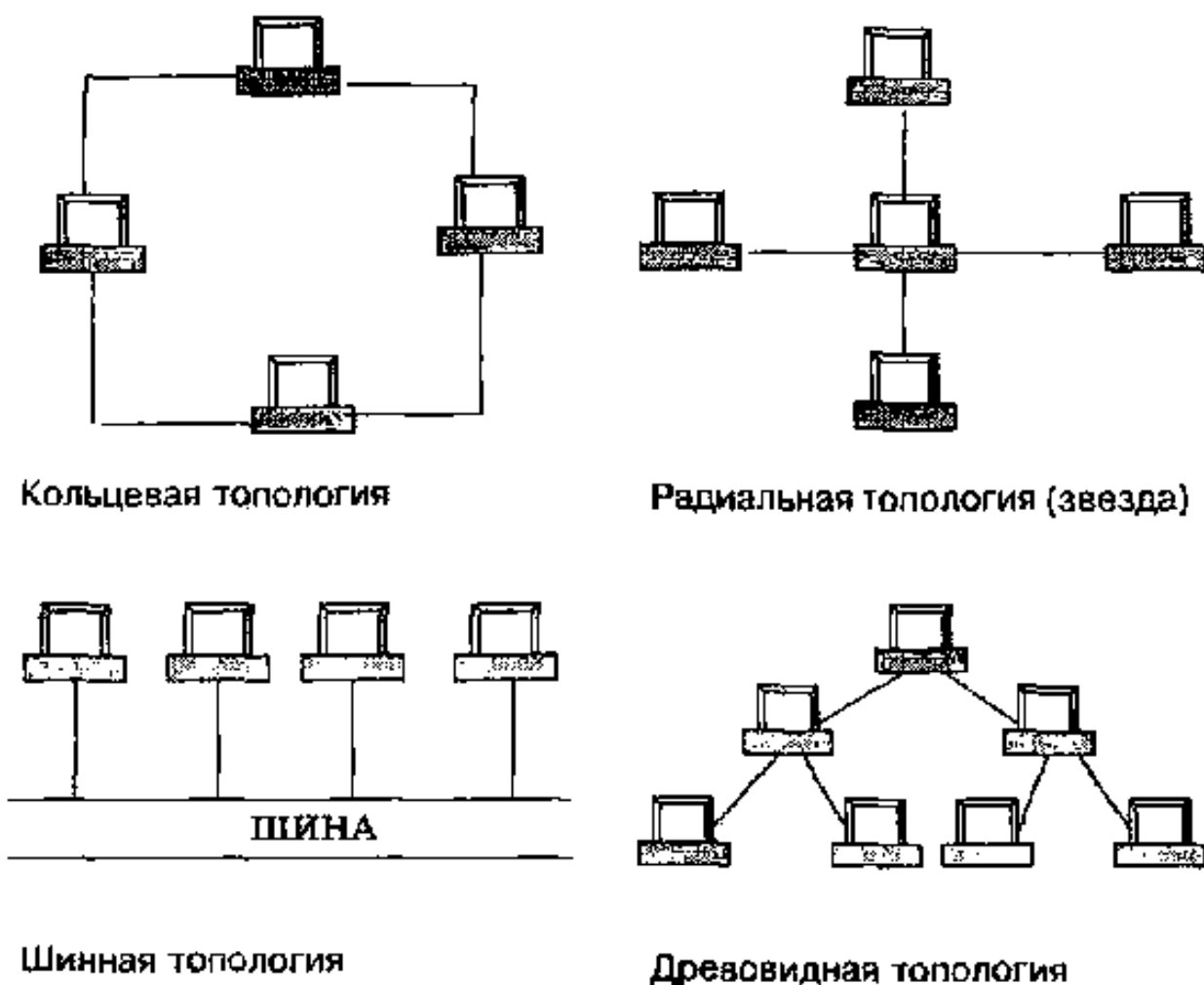


Рис. 11.1
Топологии локальной сети

- 1) обмен файлами между пользователями сети;
- 2) использование общедоступных ресурсов: большого пространства дисковой памяти, принтеров, централизованной базы данных, программного обеспечения и др.

Пользователей общей локальной сети принято называть *рабочей группой*, а компьютеры, за которыми они работают — *рабочими станциями*. Если все компьютеры в сети равноправны, то есть сеть состоит только из рабочих станций пользователей, то ее называют *одноярусной сетью*. Одноярусные сети используются для осуществления первой из отмеченных целей: для обмена файлами. У каждого компьютера в такой сети есть свое имя. Члены рабочей группы могут обращаться

ся по этим именам к дисковой памяти ПК своих коллег и копировать файлы на свой компьютер или копировать свои файлы на другие компьютеры. Возможность такого обмена обеспечивается специальной сетевой операционной системой. Средствами сетевой ОС можно защитить информацию от постороннего доступа. Таким образом, локальная сеть избавляет от необходимости использовать дискеты для переноса информации с одного компьютера на другой.

Другой способ организации локальной сети — сеть с выделенным (главным) компьютером. Его называют *файл-сервером*. Чаще всего в школьных компьютерных классах используется именно такая организация. К файл-серверу имеет доступ учитель, а ученики работают за рабочими станциями. Все рабочие станции соединены с главной машиной (схема соединения «звезда»). Поэтому непосредственный обмен информацией происходит между сервером и каждой рабочей станцией. Конечно, в такой системе ученики тоже могут обмениваться файлами, но «транзитом» через сервер. Обычно сервер — это более мощная машина, чем рабочие станции, с большим жестким диском, с дополнительными внешними устройствами (например, CD-ROM-дисководом, принтером, модемом). При такой организации локальной сети реализуется вторая из отмеченных выше целей: доступ пользователей к общим аппаратным и информационным ресурсам сервера. В частности, программы, хранящиеся на диске сервера, могут загружаться в оперативную память рабочей станции и запускаться на исполнение подобно тому, как это делается с собственного диска ПК. Со своего рабочего места пользователь может создавать и сохранять файлы на жестком диске сервера.

Работой сети управляет *сетевая операционная система*. Операционная система поддерживает стандарты (протоколы) обмена информацией в сети, устанавливает очередность при обращении различных пользователей к одним и тем же ресурсам и пр. Основное назначе-

ние сетевой ОС — дать возможность пользователям работать в локальной сети, не мешая друг другу. Работу одноранговых сетей поддерживает ОС *Windows 9X (95/98)*. Наиболее распространенные ОС для сетей с выделенным сервером: *Novell NetWare, Windows NT, Windows 2000 Server*.

Все реже в отечественных школах можно встретить специализированные классы учебной вычислительной техники — КУВТ, объединенные в локальную сеть. По сравнению с профессиональными IBM или Macintosh это более дешевая техника, предоставляющая минимально необходимые средства для преподавания информатики в школе. В конце 80-х годов было организовано производство и поставки в школы КУВТ на базе отечественных микро-ЭВМ «Электроника-БК0010», УКНЦ, «Корвет». В тот же период в ряд школ были поставлены КУВТ «Ямаха» японского производства. Позднее появились школьные КУВТ на базе IBM PS/2, так называемые «пилотные классы». Для большинства ЛС КУВТ характерно использование архитектуры «звезда», где центральная машина называется рабочим местом преподавателя — РМП, а периферийные машины — рабочими местами учеников — РМУ. На РМП имеется дисковый накопитель (на БК, УКНЦ, «Корвет», «Ямаха» — флоппи-дисковод) и принтер. На РМУ дисковой памяти нет.

На РМП установлена сетевая ОС, которая предоставляет следующий минимальный набор пользовательских возможностей:

- пересылку программ и данных с РМП на каждое из РМУ и обратно;
- исполнение программы как на РМУ, так и на РМП;
- вывод программ и данных с РМУ на внешние накопители и принтер РМП;
- групповую рассылку программ с РМП на все РМУ.

Операционные системы ЛС КУВТ предоставляют учителю ряд дополнительных возможностей, удобных для ведения урока: возможность вмешиваться в работу учащихся, просматривать их экраны, вызывать и редактировать их программы, организовывать коллективные демонстрации и пр.

Узким местом отечественных КУВТ была низкая скорость передачи информации. Поэтому рассылка больших программ на все РМУ занимала значительное время. Ситуация улучшилась с внедрением в классах на базе УКНЦ и «Корвет» сетевой системы, разработанной фирмой «Линтех». Эта система основывается на использовании на РМП компьютера IBM PC. Помимо увеличения скорости передачи данных и повышения надежности работы сети, данная система позволяет на РМУ пользоваться некоторыми средствами программного обеспечения, предназначенного для IBM.

Учебный план не позволяет долго задерживаться на теме локальных сетей. Учитель, прежде всего, должен дать представление ученикам об организации сети, работающей в компьютерном классе, а также обще-школьной сети (если такая имеется). В качестве практической работы на данную тему следует организовать обмен посланиями между учениками в виде текстовых файлов, передаваемых через сеть с одного РМУ на другое (своеобразная электронная почта).

2. Если локальную сеть ученики могут увидеть своими глазами, то знакомство с глобальными сетями будет носить более описательный характер. Здесь, как и во многих других темах, приходит на помощь метод аналогий. Устройство глобальной сети можно сравнить с устройством системы телефонной связи — телефонной сети. Телефоны абонентов связаны с узлами — коммутаторами. В свою очередь, все городские коммутаторы связаны между собой так, что между любыми двумя телефонами абонентов может быть установлена связь. Вся эта система образует телефонную сеть города. Городские

(региональные) сети связаны между собой по междугородним линиям. Выход на телефонные сети других стран происходит по международным линиям связи. Таким образом, весь мир «опутан» телефонными сетями. Два абонента в любой части света, подключенные к этой сети, могут связаться друг с другом.

Рассказав об этом, предложите ученикам представить, что у абонентов вместо телефонных аппаратов установлены персональные компьютеры; вместо коммутаторов — мощные компьютерные узлы, и по такой сети циркулирует самая разнообразная информация: от текстовой до видео и звука. Это и есть современная мировая система глобальных компьютерных сетей. В учебнике на рис. 5.1 изображена характерная структура глобальной сети. Это может быть, например, какая-нибудь региональная сеть, или ведомственная (корпоративная) сеть, объединяющая родственные организации. Сегодня практически все такие сети имеют связь с другими подобными сетями и образуют вместе международную систему компьютерных сетей Интернет.

Первая глобальная компьютерная сеть начала действовать в 1969 году в США, она называлась ARPANET и объединяла в себе всего 4 удаленных компьютера. Примером современной сети научно-образовательного назначения является BITNET. Она охватывает 35 стран Европы, Азии и Америки, объединяет более 800 университетов, колледжей, научных центров. Крупнейшей российской сетью является RELCOM, созданная в 1990 году. RELCOM входит в европейское объединение сетей EUNET, которое, в свою очередь, является участником гигантского мирового сообщества Интернет. Такая иерархичность характерна для организации глобальных сетей.

Из описания глобальной сети, приведенного в §21 учебника, следует, что основными техническими компонентами глобальной сети являются: узловые хост-компьютеры, ПК абонентов сети (терминалы), линии связи.

Обычно узел сети содержит не один, а множество компьютеров. Функции серверов различных сетевых услуг могут выполнять разные компьютеры.

Хост-компьютеры постоянно находятся во включенном состоянии, постоянно готовы к приему/передаче информации. В таком случае говорят, что они работают в режиме *on-line*. Компьютеры абонентов выходят на связь с сетью (в режим *on-line*) лишь на определенное время — сеанс связи. Переслав и получив необходимую информацию, абонент может отключиться от сети и далее работать с полученной информацией автономно — в режиме *off-line*. Маршрут передачи информации пользователю обычно неизвестен. Он может быть уверен лишь в том, что информация проходит через узел подключения и доходит до пункта назначения. Маршрутизацией передаваемых данных занимаются системные средства сети. В разных сеансах связь с одним и тем же корреспондентом может проходить по разным маршрутам.

3. На вопрос, что такое Интернет, в литературе можно прочесть разные варианты ответов. Чаще всего на этот вопрос отвечают так: Интернет — это супер-сеть, охватывающая весь мир, представляющая собой совокупность многих (более 2000) сетей, поддерживающих единый протокол *TCP/IP* (*Transmission Control Protocol/Internet Protocol*).

Сетевой протокол — это стандарт на представление, преобразование и пересылку информации в компьютерной сети. Образно можно сказать так: протокол — это определенный сетевой язык. Пока различные глобальные сети работали автономно, они «разговаривали на разных языках». Для их объединения понадобилось придумать общий язык (своеобразный сетевой Эсперанто), которым стал протокол *TCP/IP*. Этот протокол поддерживается как программными, так и аппаратными средствами сети. Сводится он к стандартизации следующих процедур:

- разбиения передаваемых данных на пакеты (части);
- адресации пакетов и передачи их по определенным маршрутам в пункт назначения;
- сборки пакетов в форму исходных данных.

При этом происходит контроль правильности приема/передачи пакета, правильности сборки всех переданных пакетов в нужном месте.

Основой Интернета является система так называемых IP-адресов. Каждый хост-компьютер, включенный в Интернет, получает уникальный в рамках всей сети адрес. IP-адрес — это последовательность из четырех целых десятичных чисел, разделенных точками. Например: 195.205.31.47. Поскольку Интернет — это сеть сетей, то первое число определяет сеть, к которой принадлежит компьютер, следующие числа уточняют координаты компьютера в этой сети.

Цифровая адресация является «внутренним делом» системы. Для пользователей она неудобна. Поэтому для пользователей используется другая — буквенная — форма записи адресов — доменные адреса. Домены — это символьные имена, разделяемые точками. Пример доменного адреса: `www.rsu.ru`. Адрес читается справа налево. Первый справа домен называется суффиксом. Чаще всего он определяет страну, в которой находится компьютер (таким образом, компьютер является элементом национальной сети). Например, `ru` — Россия, `uk` — Великобритания, `fr` — Франция. Адреса хост-компьютеров США обычно имеют суффикс, обозначающий их принадлежность к корпоративным сетям: `edu` — научные и учебные, `gov` — правительственные, `mil` — военные организации и пр.

Следующие домены (их может быть больше одного) определяют хост-компьютер в данной сети (`PSU` — Интернет-центр Пермского государственного университета). Последний домен — имя сервера (`Web` — сервер).

С помощью специальной серверной программы устанавливается связь между числовыми и доменными адресами.

Все перечисленные выше характеристики Интернета чаще всего пользователю неизвестны. С точки зрения пользователя Интернет — это определенное множество информационных услуг, которые он может получать от сети. В число услуг входят: электронная почта, телеконференции (списки рассылки), архивы файлов, справочники и базы данных, Всемирная паутина — WWW и пр. Интернет — это неограниченные информационные ресурсы. Влияние, которое окажет Интернет на развитие человеческого общества, еще до конца не осознано.

4. В истории глобальных сетей электронная почта (e-mail) появилась как самая первая информационная услуга. Эта услуга остается основной и важнейшей в компьютерных телекоммуникациях. Можно сказать, что происходит процесс вытеснения традиционной бумажной почты электронной почтой. Преимущества последней очевидны: прежде всего, это высокая скорость доставки корреспонденции (минуты, редко — часы), сравнительная дешевизна. Уже сейчас огромные объемы деловой и личной переписки идут через e-mail. Электронная почта в сочетании с факсимильной связью обеспечивают абсолютное большинство потребностей в передаче писем и документов.

Из содержания § 23 учебника следует, что для того, чтобы человек мог воспользоваться услугами электронной почты, он должен:

- иметь аппаратное подключение своего персонального компьютера к почтовому серверу узла компьютерной сети;
- иметь на этом сервере свой почтовый ящик и пароль для обращения к нему;
- иметь личный электронный адрес;
- иметь на своем компьютере клиент-программу электронной почты («мэйлер»).

Аппаратное подключение чаще всего происходит по телефонным линиям, поэтому пользователю необхо-

дим выход в телефонную сеть, то есть свой телефонный номер. Организация — владелец узла глобальной сети, предоставляющая сетевые услуги, называется *провайдером*. В последнее время их становится все больше, и пользователь имеет возможность выбрать того провайдера, условия которого его в большей степени устраивают. Провайдер назначает для пользователя пароль, электронный адрес, создает для него на почтовом сервере почтовый ящик — папку для размещения корреспонденции. Как правило, провайдер помогает пользователю установить и настроить почтовую клиент-программу.

Подготовка электронного письма производится пользователем в режиме *off-line* — отключения от сети. С помощью почтовой клиент-программы он формирует текст письма, указывает адрес получателя, вкладывает в письмо различные вложения. Затем пользователь переходит в режим *on-line*, то есть соединяется с почтовым сервером и отдает команду «*доставить почту*». Подготовленная корреспонденция передается на сервер, а поступившая на адрес пользователя переносится с сервера на его ПК. При этом полученные письма удаляются из почтового ящика, а переданные немедленно отправляются по сети адресату в соответствии с протоколом TCP/IP.

На примере электронной почты хорошо иллюстрируется суть технологии «клиент-сервер», принятой в современных сетях. Эта технология основана на разделении функций программного обеспечения, обслуживающего каждую информационную услугу, между компьютером клиента и сервером. Соответствующее ПО называется клиент-программой и сервер-программой (часто говорят короче: *клиент* и *сервер*). Популярными клиент-программами электронной почты являются: MAI для MS DOS и Outlook Express для Windows.

На базе протокола TCP/IP реализованы другие *прикладные протоколы Интернета*, составляющие основу сервиса в сети. Для поддержки электронной почты

используется протокол SMTP (Simple Mail Transfer Protocol).

В начальный период развития электронной почты передаваемая корреспонденция могла иметь только текстовый формат. Данные другого формата (двоичные файлы) перекодировывались в текстовый формат с помощью специальной программы-перекодировщика UUDECOD. Сейчас в Интернете используется стандарт MIME, позволяющий без такого перекодирования передавать в теле электронного письма самую разнообразную информацию. Согласно этому стандарту передающая машина помещает в заголовке электронного письма описания типов информационных единиц, составляющих письмо. Машина-получатель по этим описаниям правильно интерпретирует полученную информацию. Теперь в электронном письме помимо текста можно помещать графические образы (тип image), аудиоинформацию (audio), видеофильмы (video), любые приложения (application).

5. Обсудим несколько подробнее технические (аппаратные) средства глобальных сетей. Кратко об этом рассказано в § 23 учебника.

Хост-компьютеры (серверы). Хост-компьютер имеет собственный уникальный адрес в сети и выполняет роль узловой машины, обслуживающей абонентов. В качестве хост-компьютеров используются разные типы машин: от мощных ПК до миниЭВМ и даже мэйнфреймов (больших ЭВМ). Основные требования — высокоскоростной процессор и большой объем дисковой памяти (десятки и сотни Гбайт). На хост-компьютерах в сети Интернет используется операционная система Unix. Все сервер-программы, обслуживающие приложения, работают под управлением Unix.

Из того, о чем уже говорилось выше, следует, что понятие «сервер» носит программно-аппаратный смысл. Например, хост-компьютер, на котором в данный момент работает сервер-программа электронной почты,

выполняет роль почтового сервера. Если на этой же машине начинает работать сервер-программа WWW, то она становится Web-сервером. Часто функции серверов различных услуг разделены на узле сети между разными компьютерами.

Линии связи. Основные типы линий связи между компьютерами сети: телефонные линии, электрические кабели, оптоволоконный кабель и радиосвязь. Главными параметрами линий связи являются пропускная способность (максимальная скорость передачи информации), помехоустойчивость, стоимость. По параметру стоимости самыми дорогими являются оптоволоконные линии, самыми дешевыми — телефонные. Однако с уменьшением цены уменьшается и качество работы линии. В табл. 11.1 даны сравнительные характеристики линий по параметрам скорости и помехоустойчивости.

Таблица 11.1

Тип связи	Скорость, Мбит/с	Помехоустойчивость
Витая пара проводов	10–100	Низкая
Коаксиальный кабель	До 10	Высокая
Телефонная линия	1–2	Низкая
Оптоволоконный кабель	10–200	Абсолютная

Чаще всего для связи между хост-компьютерами используются выделенные телефонные линии или радиосвязь. Если узлы сети расположены сравнительно недалеко друг от друга (в пределах города), то связь между ними может быть организована по кабельным линиям — электрическим или оптоволоконным. В последнее время в сети Интернет активно используется спутниковая радиосвязь.

Обычно абоненты (клиенты) подключаются к узлу своего провайдера через телефонную линию. Все чаще для этих целей начинает применяться радиосвязь.

Для передачи информации по каналам связи необходимо преобразовывать ее из той формы, в которой она существует в компьютере, в сигналы, передаваемые по линиям связи. Такие преобразования осуществляют специальные устройства, которые называются *сетевыми адаптерами*. Существуют адаптеры для кабельной, для оптоволоконной связи. Адаптер вставляется в свободное гнездо материнской платы и соединяется кабелем с адаптером другого компьютера. Так обычно делается в локальных сетях.

В глобальных сетях, связанных по телефонным линиям, в качестве устройства сопряжения используются *модемы*. Назначение и функции модема описаны в § 23. Добавим к сказанному то, что помимо преобразования информации из двоичного компьютерного кода в телефонный сигнал и обратно, модем выполняет еще ряд функций. Например, модем клиента сети должен дозваниваться до узла, к которому он подключается.

Основной характеристикой модема является предельная скорость передачи данных. Например, некоторые современные модемы работают со скоростями 28 000 бит/с, 54 000 бит/с. Однако реальная скорость передачи данных зависит не только от модема, но и от качества телефонных линий. С переходом на цифровую телефонную связь возрастет скорость передачи информации, исчезнет потребность в использовании модемов.

6. Наряду с электронной почтой в Интернете существуют и другие виды информационных услуг для пользователей. Каждую из них обслуживает определенный прикладной протокол. Перечислим основные из них.

Telnet. Эта услуга и протокол позволяют пользователю работать в режиме терминала удаленного компьюте-

ра, то есть использовать установленные на нем программы так же, как программы на собственном компьютере.

FTP. Так называются протокол и программы, которые обслуживают работу с каталогами и файлами удаленной машины. Клиент FTP имеет возможность просматривать каталоги FTP-серверов, копировать интересные его файлы.

Archie. Так называются специальные серверы, выполняющие роль поисковых программ в системе FTP-серверов. Они помогают быстро найти нужные вам файлы.

Gopher. Система поиска и извлечения информации из сети с развитыми средствами многоуровневых меню, справочных книг, индексных ссылок и пр.

WAIS. Сетевая информационно-поисковая система, основанная на распределенных базах данных и библиотеках.

Usenet. Система телеконференций. Другое название — группы новостей. Обслуживает подписчиков определенных тематических конференций, рассылая им материалы по электронной почте.

IRC. Internet Relay Chat — «болтовня» в реальном времени. Позволяет вести письменный диалог удаленным собеседникам в режиме on-line.

Интернет-телефония. Развивающаяся услуга, поддерживающая голосовое общение клиентов сети в режиме on-line.

WWW — World Wide Web — Всемирная паутина. Это гипертекстовая информационная система в Интернете. В последнее время WWW и ее программное обеспечение становятся универсальным средством информационных услуг в Интернете. Они обеспечивают пользователям доступ практически ко всем перечисленным выше ресурсам (FTP, E-mail, WAIS, Gopher и др.)

В § 24 учебника раскрыты основные понятия, связанные с WWW: Web-страница, Web-сервер, Web-брау-

зер и др. Не упоминается лишь все чаще используемый в последнее время термин «Web-сайт». Под Web-сайтом понимают раздел данных на Web-сервере, принадлежащий какой-то организации или лицу. В этом разделе его владелец размещает свою информацию в виде множества взаимосвязанных Web-страниц. Обычно сайт имеет титул — главную страницу, от которой по гиперссылкам или указателям «вперед/назад» можно двигаться по страницам сайта.

Наиболее популярными Web-браузерами являются Internet Explorer и Netscape Navigator. Основная задача браузера — обращение к Web-серверу за искомой страницей и вывод страницы на экран. Простейший способ получения нужной информации из Интернета — указание адреса искомого ресурса.

Для хранения и поиска информации в Интернете используется универсальная адресация, которая носит название URL — Uniform Resource Locator. *URL-адрес* содержит информацию не только о том, где находится ресурс, но и по какому протоколу к нему следует обращаться. URL-адрес состоит из двух частей: первая (слева) указывает используемый протокол, а вторая (справа) — где именно в сети расположен данный ресурс (имя соответствующего сервера). Например:

`http://имя сервера/путь/файл.`

- `ftp://` — используется протокол `ftp` при обращении к `ftp`-серверам;
- `gopher://` — подключение к серверам `Gopher`;
- `http://` — использование протокола работы с гипертекстом (`Hyper Text Transfer Protocol`), который лежит в основе `WWW`. Этот тип связи надо указывать при обращении к любому `WWW`-серверу.

Вот пример адреса файла, содержащего дистанционный курс немецкого языка:

`http://www.scholar.urs.ac.ru/Teacher/
German/main.html.`

Кроме прямой адресации, поиск информации в Интернете может осуществляться по гиперссылкам.

В помощь пользователю в Интернете действует ряд специальных поисковых программ. Еще их называют *поисковыми серверами, поисковыми машинами, поисковыми системами*. Такая система работает постоянно. С помощью специальных программ-роботов она производит периодический обход всех Web-серверов в сети и собирает сводную информацию об их содержании. По результатам таких просмотров организуются справочники, индексные списки с указанием документов, где встречаются определенные ключевые слова. Затем по этим спискам обслуживаются запросы пользователей на поиск информации. Поисковая система выдает пользователю список адресов документов, в которых встречаются указанные пользователем ключевые слова.

Ниже приведены адреса наиболее популярных российских поисковых серверов:

<http://russia.agama.com/Aport/>

<http://www.rambler.ru/>

<http://yandex.ru/>

<http://www.list.ru/>

Поиск информации по ключевым словам требует от пользователя определенных навыков. Алгоритмы поиска в сети, подобно поиску информации в базах данных, основаны на логике. Рассмотрим этот вопрос на примере организации поиска по нескольким ключевым словам, принятого в поисковой системе Alta Vista.

1. Несколько ключевых слов, разделенных пробелами, соответствуют операции логического сложения ИЛИ (OR). Например, указав ключ Школьная информатика, мы получим список всех документов, в которых встречается слово «Школьная» или слово «информатика». Очевидно, таких документов окажется

слишком много и большинство из них не нужны пользователю.

2. Несколько слов, заключенных в кавычки, воспринимаются как единое целое. Указав в запросе *Школьная информатика* мы получим документы, содержащие такую строку.

3. Знак «+» между словами равносильна операции логического умножения: И (AND). Указав в запросе ключ *Школьная + информатика*, получим все документы, в которых имеются эти два слова одновременно, но они могут быть расположены в любом порядке и вразброс.

Очевидно, второй вариант запроса в большей степени соответствует цели. Однако ключевых слов в таком сочетании в списках поисковой программы может не оказаться.

В разделе 5.3 задачника-практикума приведены задания для выполнения учащимися практических работ в сети Интернет.

12. Введение в информационное моделирование

Разделы учебника: часть I, § 25, 26. Дополнительный материал: часть II, раздела 6.1, 6.2.

Основные цели. Ввести понятие модели. Познакомить учащихся с основными типами информационных моделей. Рассмотреть различные варианты использования таблиц для представления информации. В углубленном варианте познакомить детей с понятиями «система», «структура», «граф», дать представление об объектно-информационном типе моделей.

Изучаемые вопросы

- Модели натурные и информационные.
- Типы информационных моделей.
- Графические информационные модели.

- Таблицы типа «объект—свойство» и «объект—объект».
- Двоичные матрицы.
- Системы и структуры.
- Графы.
- Начала системологии.
- Объектно-информационные модели.

Общие методические рекомендации

В обязательном минимуме содержания образования по информатике присутствует линия «Моделирование и формализация». Содержание этой линии определено следующим перечнем понятий: моделирование как метод познания; формализация; материальные и информационные модели; информационное моделирование; основные типы информационных моделей. Линия моделирования, наряду с линией информации и информационных процессов является теоретической основой базового курса информатики. Дальнейшее развитие общеобразовательного курса информатики должно быть связано, прежде всего, с углублением этих содержательных линий. Основными проблемами для разработчиков базового курса являются, во-первых, выделение из обширной научной области информационного моделирования тех базовых знаний и понятий, которые должны войти в общеобразовательный школьный предмет; во-вторых — разработка методики преподавания этих вопросов.

Не следует считать, что тема моделирования носит чисто теоретический характер и автономна от всех других тем. Большинство последующих разделов базового курса имеют прямое отношение к моделированию, в том числе и темы, относящиеся к технологической линии курса. Изучавшиеся ранее текстовые и графические редакторы, программное обеспечение телекоммуникаций можно отнести к средствам, предназначенным для рутинной работы с информацией: позволяющим на-

брать текст, построить чертеж, передать или принять информацию по сети. Программные средства информационных технологий, которые предстоит изучать дальше — СУБД, табличные процессоры, следует рассматривать как инструменты для работы с информационными моделями. Алгоритмизация и программирование также имеют прямое отношение к моделированию. Следовательно, линия моделирования является сквозной для многих разделов базового курса.

Однако прежде чем перейти к прикладным вопросам моделирования, необходим вводный разговор, обсуждение некоторых общих понятий, в частности тех, которые обозначены в обязательном минимуме. Здесь возникают проблемы как содержательного, так и методического характера, связанные с глубоким научным уровнем понятий, относящихся к этой теме. Методика информационного моделирования связана с вопросами системологии, системного анализа. Степень глубины изучения этих вопросов существенно зависит от уровня подготовленности школьников. В возрасте 14–15 лет дети еще с трудом воспринимают абстрактные, обобщенные понятия. Поэтому раскрытие таких понятий должно опираться на простые, доступные ученикам примеры.

Методические рекомендации по изложению теоретического материала

В учебнике и задачнике-практикуме представлен материал, который позволяет изучать вопросы формализации и моделирования с разной степенью подробности. Ниже будут рассмотрены три уровня изучения: первый — минимальный, второй — дополненный, третий — углубленный.

Первый уровень

Минимальный уровень содержания темы «Введение в информационное моделирование» соответствует материалу учебника, изложенному в § 25, 26. Здесь раскрывается система понятий, отраженная на следующей схеме.

Разговор с учениками по данной теме можно вести в форме беседы. Сам термин «модель» большинству из них знаком. Попросив учеников привести примеры каких-нибудь известных им моделей, учитель наверняка услышит в ответ: «модель автомобиля», «модель самолета» и другие технические примеры. Хотя технические модели не являются предметом изучения информатики, все же стоит остановиться на их обсуждении. Информатика занимается информационными моделями. Однако между понятиями материальной (натурной) и информационной модели есть аналогии. Примеры материальных моделей для учеников более понятны и наглядны. Обсудив на таких примерах некоторые общие свойства моделей, затем их можно будет перенести на модели информационные.

Расширив список натуральных моделей (записав на доске), следует обсудить их общие свойства. Все эти модели воспроизводят объект-оригинал в каком-то упрощенном виде. Часто модель воспроизводит только форму реального объекта в уменьшенном масштабе. Могут быть модели, воспроизводящие какие-то функции объекта. Например, заводной автомобильчик может ездить, модель корабля может плавать. Но, в любом случае, модель не повторяет всех свойств реального объекта, а лишь только те, которые требуются для ее будущего применения. Поэтому важнейшим понятием в моделировании является понятие цели. *Цель моделирования — это назначение будущей модели.* Цель определяет те свойства объекта-оригинала, которые должны быть воспроизведены в модели.

Полезно отметить, что моделироваться могут не только материальные объекты, но и процессы. Напри-

мер, конструкторы авиационной техники используют аэродинамическую трубу для воспроизведения на земле условий полета самолета. В такой трубе корпус самолета обдувается воздушным потоком. Создается модель полета самолета, то есть условия, подобные тем, что происходят в реальном полете. На такой модели измеряются нагрузки на корпус, исследуется прочность самолета и пр. С моделями физических процессов работают физики-экспериментаторы. Например, в лабораторных условиях они моделируют процессы, происходящие в океане, в недрах Земли и др.

Условимся в дальнейшем термин «объект моделирования» понимать в широком смысле: это может быть и некоторый вещественный объект (предмет, система) и реальный процесс.

Закрепив в сознании учеников понимание смысла цепочки «объект моделирования — цель моделирования — модель» (рис. 12.1), можно перейти к разговору об информационных моделях. Самое общее определение: *информационная модель — это описание объекта моделирования*. Иначе можно сказать, что это информация об объекте моделирования. А, как известно, информация может быть представлена в разных формах, поэтому существуют различные формы информационных моделей. В их числе: словесные, или *вербальные*

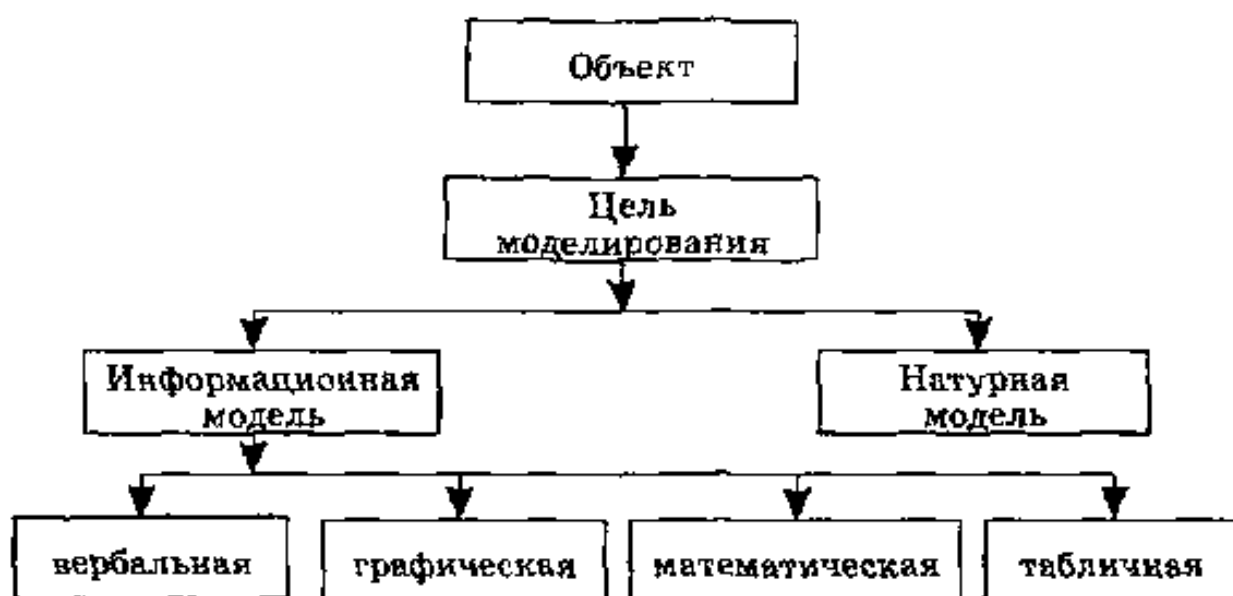


Рис. 12.1. Структура основных понятий

модели, *графические, математические, табличные* (см. рис. 12.1).

Следует иметь в виду, что нельзя считать этот список полным и окончательным. В научной и учебной литературе встречаются разные варианты классификаций информационных моделей. Например, еще рассматривают *алгоритмические модели, имитационные модели* и др. Естественно, что в рамках базового курса мы вынуждены ограничить эту тему. В старших классах при изучении профильных курсов могут быть рассмотрены и другие виды информационных моделей.

Построение информационной модели, так же как и натурной, должно быть связано с целью моделирования. Всякий реальный объект обладает бесконечным числом свойств, поэтому для моделирования должны быть выделены только те свойства, которые соответствуют цели. Процесс выделения существенных для моделирования свойств объекта, связей между ними с целью их описания называется *системным анализом*. В минимальном содержании данной темы, изложенном в § 25–26, сущность системного анализа практически не раскрывается. Присутствующее определение дает лишь самое начальное представление, но далее не детализируется.

Форма информационной модели также зависит от цели ее создания. Если важным требованием к модели является ее наглядность, то обычно выбирают графическую форму. В учебнике приведены различные примеры графических моделей: карта местности, чертеж, электрическая схема, график изменения температуры. Следует обратить внимание учеников на различные назначения этих графических моделей. На примере графика температуры можно обсудить то обстоятельство, что та же самая информация могла бы быть представлена и в другой форме. Зависимость температуры от времени можно отразить в числовой таблице — табличная модель, можно описать в виде математической функции — математическая модель. Для разных целей могут оказаться удобными разные формы модели. С

точки зрения наглядности наиболее подходящей является графическая форма.

А что обозначает слово «формализация»? Это все то, о чем говорилось выше. *Формализация — это замена реального объекта или процесса его формальным описанием, то есть его информационной моделью.* Построив информационную модель, человек использует ее вместо объекта-оригинала для изучения свойств этого объекта, прогнозирования его поведения и пр. Прежде чем строить какое-то сложное сооружение, например мост, конструкторы делают его чертежи, проводят расчеты прочности, допустимых нагрузок. Таким образом, вместо реального моста они имеют дело с его модельным описанием в виде чертежей, математических формул.

Рассмотрение табличных моделей в § 26 носит подготовительный характер к будущему изучению реляционных баз данных и электронных таблиц. Однако данная в учебнике классификация таблиц полезна не только применительно к информационным технологиям. Очень часто в табличной форме представляется информация в различных документах, справочниках, учебниках. Табличная форма придает лаконичность и наглядность данным, структурирует данные, позволяет увидеть закономерности в характере данных. Нередко табличная форма представления информации применяется наряду с графической. Большинство табличных процессоров дают возможность совмещать таблицы с диаграммами и графиками.

Умение представлять данные в табличной форме — очень полезный общеметодический навык. Практически все школьные предметы используют таблицы, но никакой из них не учит школьников строить таблицы. Приведение данных к табличной форме является одним из приемов систематизации информации. В § 26 описываются два типа таблиц: таблицы типа «объект—свойство» и «объект—объект». Это наиболее простые и наиболее часто встречающиеся типы таблиц. Кроме того, даны примеры применения двоичных матриц.

Двоичные матрицы используются в тех случаях, когда нужно отразить наличие или отсутствие связей между отдельными элементами некоторой системы. Например, между населенными пунктами (табл. 6.4), между учениками и факультативами (табл. 6.5). С помощью двоичных матриц удобно представлять сетевые структуры. Задание № 6 в конце параграфа содержит двоичную матрицу, отражающую связи между различными серверами компьютерной сети:

	С 1	С 2	С 3	С 4	С 5
С 1	1	0	0	1	0
С 2	0	1	0	1	0
С 3	0	0	1	1	0
С 4	1	1	1	1	1
С 5	0	0	0	1	1

Глядя на таблицу, ученики должны определить, какой из пяти серверов является узловым. Решение очень простое: поскольку по данному определению узловым называется тот сервер, с которым непосредственно связаны все другие серверы, то в матрице нужно искать строку, состоящую только из единиц. Это строка С4. Значит, сервер С4 является узловым. Предложите ученикам, в качестве дополнительного задания, нарисовать эту компьютерную сеть, изобразив серверы кружками, а связи между ними линиями.

Задание № 5 также содержит двоичную матрицу, хотя в ней нет нулей и единиц:

	9-а	9-б	10-а	10-б	11-а	11-б
урок 1						
урок 2						
урок 3						
урок 4						
урок 5						
урок 6						

Это расписание уроков по физкультуре. Ее можно привести к традиционному виду двоичной матрицы, поставив в заштрихованные клетки единицы, в другие — нули (можно этого и не делать). Глядя на таблицу, легко ответить на поставленные вопросы. Например, требуется определить, какое минимальное количество учителей физкультуры требуется при таком расписании. Из таблицы видно, что на втором уроке физкультура стоит в трех классах. Следовательно, требуется три учителя физкультуры. Подобные задачи приходится решать завучу школы при составлении расписания уроков.

Второй уровень

Дополнительный материал для изучения темы «Введение в информационное моделирование» содержится в разделе 6.1 «Информационные модели на графах». Здесь обсуждаются такие понятия, как «система», «структура», «граф», «деревья», «сети». Необходимо отметить, что эти понятия постепенно начинают проникать в перечень обязательных для изучения в рамках базового курса. В период подготовки учебника в содержании образовательного минимума они не упоминались, поэтому и были помещены в дополнительный раздел. Тем не менее, авторы считают необходимым хотя бы краткое знакомство учащихся с элементами системологии. Они придают цельность и некоторую понятийную полноту данной теме.

Понятие «система» очень часто употребляется как в научных дисциплинах, так и в повседневной жизни. Примеров тому достаточно много: Солнечная система, периодическая система химических элементов, системы растений и животных, система образования, система транспорта, файловая система, операционная система и многое другое. Во многих случаях понятие системы считается интуитивно ясным. Однако для информатики оно является одним из фундаментальных и требует разъяснения.

Под системой понимается любой объект, состоящий из множества взаимосвязанных частей и существующий как единое целое. В информатике понятие «система» употребляется достаточно часто. Совокупность взаимосвязанных данных, предназначенных для обработки на компьютере — система данных. Совокупность взаимосвязанных программ определенного назначения — программные системы (ОС, системы программирования, пакеты прикладных программ и др.). Информационные системы — одно из важнейших приложений компьютерных технологий.

Основным методическим принципом информационного моделирования является *системный подход*, согласно которому всякий объект моделирования рассматривается как система. Из всего множества элементов, свойств и связей выделяются лишь те, которые являются существенными для целей моделирования. В этом и заключается сущность системного анализа. Задача системного анализа, который проводит исследователь — упорядочить свои представления об изучаемом объекте для того, чтобы в дальнейшем отразить их в информационной модели.

Сама информационная модель представляет собой также некоторую систему параметров и отношений между ними. Эти параметры и отношения могут быть представлены в разных формах: графической, математической, табличной и др. Таким образом, просматривается следующий порядок этапов перехода от реальному объекту к информационной модели:

реальный объект $\Rightarrow$ системный анализ $\Rightarrow$ система данных, существенных для моделирования $\Rightarrow$ информационная модель

Важной характеристикой всякой системы является ее структура. *Структура — это определенный порядок объединения элементов, составляющих систему.* Наиболее удобным и наглядным способом представления структуры систем являются *графы*. В

разделе 6.1 описываются основные правила представления графов, вводятся понятия «вершина», «дуга», «ребро», «ориентированный граф», «дерево», «сеть». Обычно у учащихся не вызывает проблем понимание схем, представленных в форме графа: графа родственных связей, графа системы связанных между собой населенных пунктов и др. Важной разновидностью графов являются деревья. *Дерево — это графическое представление иерархической структуры системы.* Обычно это системы, между элементами которых установлены отношения подчиненности или вхождения друг в друга: системы власти, административные системы, системы классификации в природе и др. Ученики уже знакомы с понятием дерева применительно к системе файлов на дисках компьютера. Многим из них известен смысл понятия «родословное дерево».

В качестве дополнительного материала для работы с таблицами следует использовать раздел 2.3 «Табличные информационные модели» из задачника-практикума. В этом разделе достаточно подробно описывается методика построения таблиц различных типов. Задачи содержат большое количество данных, которые следует организовать в табличной форме. Содержание этого раздела выходит за рамки материала учебника. Кроме таблиц типа «объект—свойство» и «объект—объект» рассматриваются некоторые другие виды таблиц.

В разделе 6.2 даются начальные представления об одном из современных подходов к информационному моделированию — объектно-ориентированном подходе. Создаваемые в соответствии с этим подходом модели названы объектно-информационными моделями. Как известно, данный подход берет свое начало в парадигме объектно-ориентированного программирования (ООП), возникшей еще в 1970-х годах. ООП базируется на трех основных понятиях: инкапсуляция, наследование и полиморфизм. Примеры, рассмотренные в разделе 6.2, иллюстрируют первые два свойства, связанные с объектно-информационными моделями: инкапсуля-

цию и наследование. Инкапсуляция — это объединение в перечне характеристик объекта его свойств и поведения. Наследование связано с иерархией классов объектов, с передачей свойств и поведения, присущих для класса верхнего уровня классам нижних уровней иерархии. Основная педагогическая задача при изучении этого раздела: разобраться в приводимых примерах, выполнить задания, данные в конце раздела.

Подводя итог, можно сказать, что второй уровень изучения темы «Введение в информационное моделирование» более подробно раскрывает суть системного анализа, знакомит учащихся с таким важным инструментом формализации, как графы, дает начальные представления об объектно-информационном моделировании.

Третий уровень

Наиболее полный и последовательный материал по вопросам системологии содержится в разделе 2 задачника-практикума. Этот материал может быть использован как для углубленного варианта преподавания базового курса информатики, так и для профильных курсов, ориентирующихся на информационное моделирование. Содержание данного раздела позволяет реализовать на уроках следующий перечень дидактических целей:

- Научить учеников рассматривать окружающие объекты как системы взаимосвязанных элементов, осознавать, в чем проявляется системный эффект (принцип эмерджентности) в результате объединения отдельных элементов в единое целое.
- Раскрыть смысл модели «черного ящика». Этот подход характерен для кибернетики и применяется он в тех случаях, когда внутреннее устройство системы не раскрывается, а система рассматривается лишь с точки зрения ее взаимодействия с окружающей средой. В таком случае основными понятиями, характеризующими систему, являются:

ся не ее состав и структура, а ее «входы» и «выходы».

- Дать детям представление о некоторых методах системного анализа, в частности декомпозиции, классификации.
- Научить учеников читать информационные модели, представленные в виде графов, и строить граф-модели.
- Научить учеников разбираться в различных типах таблиц, подбирать наиболее подходящий тип таблицы для организации данных, грамотно оформлять таблицы.

В задачнике-практикуме содержится значительное число заданий следующего плана: имеется множество несистематизированных данных, приведенных в вербальной форме. Задача заключается в том, чтобы систематизировать эту информацию, перейдя к другой форме ее представления: таблице или графу. Это очень важный для практики тип информационных задач. Они наглядно показывают, что несистематизированная информация оказывается во многом обесцененной. Систематизация данных имеет особенно важное значение для информационного моделирования тогда, когда строятся модели сложных систем: экономических, социальных, производственных с большим числом разнообразных параметров. От исследователя требуется умение классифицировать данные по некоторым признакам, отражать иерархические связи и пр.

Содержательная линия формализации и моделирования выполняет в базовом курсе информатики важную педагогическую задачу: *развитие системного мышления учащихся*. Эффективная работа с большими объемами информации невозможна без навыков ее систематизации. Компьютер предоставляет пользователю удобные инструменты для этой работы, но систематизацию данных пользователь должен выполнять сам.

Можно выделить три типа задач из области информационного моделирования, которые по возрастанию степени сложности для восприятия учащимися располагаются в таком порядке:

1) дана информационная модель объекта; научиться ее понимать, делать выводы, использовать для решения задач;

2) дано множество несистематизированных данных о реальном объекте (системе, процессе); систематизировать их и, таким образом, получить информационную модель;

3) дан реальный объект (процесс, система); получить его информационную модель.

Информационное моделирование — это прикладной раздел информатики, связанный с самыми разнообразными предметными областями: техникой, экономикой, естественными и общественными науками и пр. Поэтому решением задач третьего типа занимаются специалисты в соответствующих областях. В рамках школьного курса информатики информационное моделирование может быть предметом профильного курса, смежного с другими школьными дисциплинами: физикой, биологией, экономикой и др. Базовый курс информатики дает лишь начальные понятия о моделировании, систематизации данных, знакомит с компьютерными технологиями, применяемыми для информационного моделирования.

13. Базы данных

Разделы учебника: часть I, § 27–32. Дополнительный материал: часть II, разделы 7.1–7.2.

Основные цели. Дать ученикам представление о назначении информационных систем и баз данных. Познакомить детей с основами реляционных баз дан-

ных. Обучить их основным приемам работы с одной из реляционных СУБД. Обучить учеников организации поиска, сортировки, редактирования данных. В углубленном варианте: дать детям представление о проблемах проектирования реляционных баз данных.

Изучаемые вопросы

- ☐ Назначение информационных систем и баз данных (БД).
- ☐ Классификация БД.
- ☐ Структура реляционной базы данных (РБД).
- ☐ Элементы РБД: главный ключ; имя, значение и тип поля.
- ☐ Назначение СУБД; режимы работы СУБД.
- ☐ Поиск информации в базе данных.
- ☐ Логические выражения в условиях поиска и удаления записей.
- ☐ Сортировка; ключи сортировки.
- ☐ Элементы проектирования РБД; нормализация данных.

Методические рекомендации по изложению теоретического материала

1. Изучение баз данных следует начать с обоснования актуальности данного приложения компьютерной техники. Задачу можно сформулировать следующим образом: имеется большой объем данных о какой-то реальной системе объектов или событий, например, о книгах в библиотеке, о работниках предприятия, о товарах на складе, о дорожно-транспортных происшествиях за длительный период времени и т. п. Необходимо организовать хранение этой информации таким образом, чтобы ее было удобно просматривать, пополнять, изменять, искать нужные сведения, делать любые выборки, осуществлять сортировку в любом порядке. Такой работой людям приходилось заниматься и задолго до появления компьютеров. Основным средством хранения данных была бумага. Данные хранились в виде

списков в толстых журналах, папках, на картонных карточках. Последний способ используется, например, в библиотечных каталогах. Большинству учеников он хорошо знаком: на каждой карточке записаны сведения об отдельной книге. В алфавитном каталоге карточки систематизированы по фамилиям авторов в алфавитном порядке, в предметном каталоге — по тематике книг. Подобные систематизированные картотеки используются в отделах кадров предприятий. Они удобны тем, что легко можно извлечь нужную карточку, заменить ее, добавить новые карточки, сохраняя установленный порядок. Тем не менее, если такая картотека содержит тысячи карточек, то, как бы совершенна ни была ее организация, обработка данных в ней — дело длительное и трудоемкое.

Другой пример — архивы различных документов. Например, существуют исторические архивы, архивы судебных дел, архивы патентов на изобретения и многие другие. Порой такие архивы занимают целые здания. Поиск в них нужных документов требует значительных усилий. Кроме того, существуют киноархивы, фотоархивы, архивы звуковых записей.

В наше время решению описанных проблем помогают компьютеры. *Компьютерные информационные системы позволяют хранить большие объемы данных, осуществлять в них быстрый поиск, вносить изменения, выполнять всевозможные манипуляции с данными (группировать, сортировать и пр.).* Следует привести примеры таких информационных систем. Это, например, система продажи железнодорожных и авиационных билетов. Другой знакомый ученикам пример: во время телерепортажей с крупных международных соревнований, олимпийских игр на экран мгновенно выводится досье любого спортсмена, о котором говорит комментатор — это работает компьютерная информационная система.

Основой всякой информационной системы является база данных — организованная совокупность дан-

ных на магнитных дисках. Ученики уже хорошо знают, что информация на дисках хранится в виде файлов. Поэтому первый вывод, который можно сделать относительно организации больших баз данных, — это то, что они требуют больших объемов дисковой памяти.

2. Следующий вопрос — классификация баз данных. Базы данных классифицируются по разным признакам. *По характеру хранимой информации* БД делятся на фактографические и документальные. Если проводить аналогию с описанными выше примерами информационных хранилищ, то фактографические БД — это картотеки, а документальные — это архивы. В фактографических БД хранится краткая информация в строго определенном формате. В документальных БД — всевозможные документы. Причем это могут быть не только текстовые документы, но и графика, видео и звук (мультимедиа).

Классификация *по способу хранения данных* делит БД на *централизованные и распределенные*. Вся информация в централизованной БД хранится на одном компьютере. Это может быть автономный ПК или сервер сети, к которому имеют доступ пользователи-клиенты. Распределенные БД используются в локальных и глобальных компьютерных сетях. В таком случае разные части базы хранятся на разных компьютерах.

Третий признак классификации баз данных — *по структуре организации данных*. В разделе «Введение в информационное моделирование» говорилось о трех способах организации данных: табличном, иерархическом и сетевом. Базы данных, использующие эти способы организации информации, называются соответственно *реляционными* (табличные БД), *иерархическими* и *сетевыми* БД.

3. В базовом курсе информатики рассматриваются лишь фактографические реляционные базы данных. Это связано не только с ограниченностью школьного курса, но и с тем фактом, что реляционный тип БД ис-

пользуется сегодня наиболее часто и является универсальным. Теоретически доказано, что любая система данных может быть отражена с помощью таблиц. Простейшая реляционная БД содержит одну таблицу, более сложная может состоять из множества взаимосвязанных таблиц.

Структура таблицы обсуждалась в предыдущей теме и поэтому ученикам уже знакома: в разных строчках содержится информация о разных объектах описываемой системы, а столбцы соответствуют различным атрибутам этих объектов. В терминологии реляционных баз данных *строки таблицы называются записями, столбцы — полями*. Само название «реляционная БД» происходит от английского слова «relation», что переводится как «отношение». Здесь термин «отношение» понимается как взаимосвязь между полями таблицы. В реляционном подходе таблица называется отношением.

Объяснение данного материала следует проводить на конкретных примерах. В качестве примеров в учебнике используются таблицы, рассмотренные в § 25: «Домашняя библиотека», «Погода», «Успеваемость», «Факультативы». Следует подчеркнуть, что в базах данных каждая таблица должна иметь свое имя. Учителю будет удобно работать, если эти таблицы вынести на плакаты.

Основные представления, которые должны быть закреплены учениками:

- всякая таблица содержит в себе информацию о некоторой реальной системе (процессе) и, следовательно, является ее информационной моделью;
- всякая запись в таблице — информация о конкретном объекте (событии) данной системы;
- значение поля в каждой записи — это определенная характеристика (свойство, атрибут) объекта.

Вопросы, задаваемые ученикам для закрепления этих понятий, должны быть следующего характера:

— Информация о какой системе содержится в таблице «Успеваемость»?

- Об определенном классе школы.
- Информация о каком объекте содержится в каждой записи этой таблицы?
- О конкретном ученике.
- Что обозначают значения полей в записи?
- Оценки, полученные данным учеником по школьным предметам.

4. Основные понятия, связанные с записями и полями: *главный ключ записи, имя поля, значение поля, тип поля.*

Главный ключ — это поле или совокупность полей, которое однозначно определяет запись в таблице. Можно еще сказать так: *главный ключ* — это идентификатор записи. В учебнике приведены примеры простых и составных ключей. В базах данных слово «ключ» имеет несколько употреблений: *ключ поиска* — поле, по значению которого ищется запись в БД, *ключ сортировки* — поле, по значению которого происходит упорядочение записей. Поэтому идентификатор записей приходится называть *главным ключом*.

Учителю важно понимать, что тема «Базы данных» содержит в себе ряд узловых вопросов, имеющих фундаментальное значение для курса информатики в целом. В этой теме ученики впервые встречаются с понятием *величины*. В дальнейшем это понятие будет широко использоваться в электронных таблицах, в алгоритмах и программах. *Величина* — это отдельный информационный объект, имеющий собственное имя и занимающий место в памяти компьютера. С этой точки зрения поля являются величинами. Каждое поле в таблице имеет имя, для каждого поля определен тип. Понятие типа величины связано с тремя ее свойствами:

- множеством значений, которые может принимать величина;
- множеством операций, которые можно выполнять с этой величиной;
- формой внутреннего представления в памяти ЭВМ.

В большинстве случаев в базах данных используются 4 основных типа: символьный, числовой, «дата» и логический. Поле символьного типа может хранить значение любой последовательности символов; числовые поля могут содержать целые или дробные десятичные числа; дата — день/месяц/год; логические поля — значения логических величин (да/нет, истина/ложь, true/false).

Необходимо обратить внимание учеников на следующее обстоятельство: символьное поле может состоять из цифр. По этой причине иногда возникает путаница с символьным и числовым типами. Чтобы этого не происходило, нужно придерживаться следующего правила: *если поле обозначает порядковый номер или цифровой код, то ему нужно назначить символьный тип*. Если же поле обозначает количество чего-то или какую-то размерную величину, то это поле должно иметь числовой тип. Например, полями символьного типа должны быть: инвентарный номер книги в библиотеке, номер дома, номер телефона. Числовой тип имеют, например, такие поля: возраст человека, вес объекта, расстояние, количество учеников в классе.

Значения числовых полей могут быть использованы в вычислениях. Иногда в условиях поиска информации присутствуют арифметические выражения, операндами в которых могут быть только значения числовых полей. Вычислительные операции с номерами и кодами обычно не имеют смысла. В то же время к символьным полям можно применять операции отношений (меньше, больше, равно и др.). Поскольку цифры в кодовой таблице расположены в порядке возрастания, то отношения между строками, состоящими из цифр, сохраняют математический смысл. Например, истинным будет отношение $"58" > "24"$. Однако, если число цифр разное, то результат может оказаться неверным. Например, отношение $"2" < "11"$ окажется ложным, хотя с математической точки зрения оно верное. Поэтому желательно выравнивать число знаков в цифро-

вых строках добавлением нулей слева. Тогда отношение: "02" < "11" будет истинным.

В данном разделе впервые в курсе информатики ученики встречаются с логическим типом данных, с логическими величинами. Первое понятие о логической величине можно дать такое: это ответ на альтернативный вопрос. Например: «Имеется ли данная книга в библиотеке?», или «Поступил ли абитуриент в университет?», или «На улице идет дождь?» и т. п. Ответами на такие вопросы могут быть только «да» или «нет». Синонимами являются «истина», «ложь»; «true», «false». Если поле таблицы будет принимать только такие значения, то ему назначается логический тип.

Необходимо закрепить понятия «главный ключ», «имя поля», «тип поля» на серии заданий.

Задания первого типа: дано имя таблицы и перечень полей, требуется указать главный ключ и определить типы всех полей.

Приведем примеры из учебника.

БИБЛИОТЕКА (НОМЕР, АВТОР, НАЗВАНИЕ, ГОД, ПОЛКА)

Здесь использована традиционная форма для представления таблиц (отношений) в теории реляционных баз данных. Перед скобками записано имя таблицы, в скобках через запятую перечислены имена полей. Ключевые поля подчеркиваются. В данном отношении все поля *символьного типа*. *Выше было объяснено, почему полям «НОМЕР», «ГОД» и «ПОЛКА» не имеет смысла задавать числовой тип.*

ПОГОДА (ДЕНЬ, ОСАДКИ, ТЕМПЕРАТУРА, ДАВЛЕНИЕ, ВЛАЖНОСТЬ)

Здесь поле «ДЕНЬ» имеет тип «дата», «ОСАДКИ» — символьный тип; «ТЕМПЕРАТУРА», «ДАВЛЕНИЕ» и «ВЛАЖНОСТЬ» — числовой тип.

УСПЕВАЕМОСТЬ (УЧЕНИК, РУССКИЙ, АЛГЕБРА, ХИМИЯ, ФИЗИКА, ИСТОРИЯ, МУЗЫКА)

Здесь поле «УЧЕНИК» — символьного типа, все остальные — числового. Оценкам имеет смысл определить числовой тип, так как в запросах могут использоваться вычисления, например, поиск по среднему баллу.

ФАКУЛЬТАТИВЫ (УЧЕНИК, ГЕОЛОГИЯ, ЦВЕТОВОДСТВО, ТАНЦЫ)

Здесь поле «УЧЕНИК» имеет символьный тип, остальные поля логического типа. Значения этих полей — ответы на вопросы, посещает ли ученик данный факультатив.

Задания второго типа: определена предметная область базы данных; требуется озаглавить таблицу, определить имена полей и их типы, назначить главный ключ. Конечно, эта задача имеет отношение к области проектирования БД и уже по этой причине сложная. Поэтому требовать от учеников какой-то полноты ее решения не следует. Достаточно, чтобы они указали несколько полей, имеющих отношение к данной теме, и правильно определили типы полей. Например (задание № 12 к § 27):

СТРАНЫ МИРА	ОДНОКЛАССНИКИ	КИНОФИЛЬМЫ
<u>СТРАНА</u> — символьный	<u>ФАМИЛИЯ</u> — символьный	<u>НАЗВАНИЕ</u> — символьный
СТОЛИЦА — символьный	<u>ИМЯ</u> — символьный	СТРАНА — символьный
ПЛОЩАДЬ — числовой	АДРЕС — символьный	ЖАНР — символьный
НАСЕЛЕНИЕ — числовой	ТЕЛЕФОН — символьный	РЕЖИССЕР — символьный
ПОСЕЩАЛ — логический	РОДИЛСЯ — дата	ДУБЛИРОВАН — логический

5. После знакомства с основными понятиями, относящимися к организации информации в реляционных

БД, следует перейти к изучению программного обеспечения, предназначенного для работы с базами данных. Такое программное обеспечение называется СУБД — *система управления базами данных*.

Для персональных компьютеров существует целый ряд СУБД реляционного типа. Исторически одной из первых таких систем была dBASE. Используемые в этой системе форматы представления данных, язык обработки данных стали стандартом для ряда последующих СУБД. К ним относятся FoxPro, Paradox; СУБД с русифицированными оболочками «Карат» и «Ребус», работающие в операционной системе MS DOS. В состав пакета Microsoft Office входит реляционная СУБД Access, все чаще используемая в школе.

Существуют два способа работы пользователя с базой данных: работа с помощью прикладных программ, заранее составленных программистом в среде СУБД, и работа путем непосредственного взаимодействия с СУБД. В начальный период появления и распространения баз данных использовался только первый способ. Позже стали появляться СУБД, ориентированные на работу с пользователем. Для них были созданы удобные диалоговые оболочки, позволяющие пользователю легко реализовывать свои потребности в работе с базой данных: пополнять и изменять базу, осуществлять поиск данных по любым условиям, создавать отчетные документы. Наиболее совершенной СУБД такого типа является MS Access.

В разных школах используется разная техника и программное обеспечение. Варианты доступных СУБД могут быть самыми разнообразными: от учебной РБД для УКНЦ до профессиональной MS Access для IBM PC. В различных системах различается способ взаимодействия пользователя с СУБД — пользовательский интерфейс. Есть два принципиально различных режима такого взаимодействия: посимвольный ввод команд и работа с диалоговой оболочкой. Эта ситуация подобна взаимодействию пользователя с операционной систе-

мой MS DOS: работа в режиме командной строки или через оболочку Norton Commander. У всякой СУБД существует свой *язык описания данных и язык манипулирования данными*. Если пользователю приходится работать в режиме посимвольного ввода команд, то он должен знать все подробности синтаксиса командного языка. Диалоговая оболочка — более высокоуровневое средство интерфейса. Однако следует понимать, что, работая с оболочкой, пользователь формирует те же самые команды, привлекая вспомогательные средства: меню, панели инструментов, подсказки и пр.

Любая команда — это информация, управляющая выполнением определенного вида работы. Она должна содержать все необходимые данные для этого. Обычно это *имя команды и набор параметров*. С методической точки зрения важно, чтобы ученики, выполняя любые действия с БД, понимали, какая отдается команда, какие параметры для ее выполнения должны быть сообщены системе. В учебнике применен следующий методический прием: рассматривается некоторая гипотетическая реляционная СУБД. Для нее определяется язык команд, в котором используются служебные слова на русском языке. Система взаимодействует с пользователем в диалоговом режиме: на экран выводится приглашение в виде точки (так принято в dBASE и ей подобных СУБД). В ответ на приглашение пользователь вводит с клавиатуры очередную команду. После выполнения команды повторяется приглашение.

В учебнике не дается описания среды какой-либо конкретной СУБД. В отличие от интерфейса текстовых или графических редакторов, интерфейсы различных СУБД менее унифицированы. Поэтому здесь трудно давать обобщенное описание. Адаптацию к конкретной системе должен выполнить учитель. Как всегда в таких случаях, нужно привлекать дополнительную литературу: пособия, справочники, методические пособия.

Изучение конкретной СУБД следует проводить по стандартной методической схеме: «среда-режимы ра-

боты—система команд—данные». Например, для СУБД Access различаются следующие основные режимы работы:

- режим работы с таблицей: «Таблица»;
- режим работы с запросами: «Запрос»;
- режим работы с отчетами: «Отчет»;
- режим работы с формами: «Форма».

(Режимы работы с макросами и модулями в базовом курсе не рассматриваются.) В свою очередь, в каждом из отмеченных режимов есть подрежимы: «Просмотр», «Конструктор» и «Создать». Например, работая в режиме «Таблица—Просмотр», пользователь может просмотреть содержание таблицы, а также отредактировать некоторые ее поля. В режиме «Таблица—Конструктор» можно просмотреть описание структуры таблицы и внести в нее изменения. В режиме «Таблица—Создать» описывается и создается структура новой таблицы.

6. Материал учебника ориентирован на два уровня изучения темы «Базы данных». Задача первого уровня: дать детям общие представления о базах данных, научить работать с готовой БД: осуществлять поиск информации; сортировку, удаление и добавление записей; научить их создавать структуру и заполнять однотабличную базу данных. Дополнительная задача второго уровня: познакомить детей с основами проектирования БД.

Работа с СУБД начинается с запуска соответствующей программы, поэтому ученикам необходимо показать, где хранится программа и как ее запустить на исполнение.

Первое понятие, которое должны усвоить ученики: база данных хранится в файле; чтобы начать с ней работать, необходимо *открыть файл* с БД. Затем учитель должен показать, как можно просмотреть на экране записи таблицы. Для этого должна быть заранее подготовлена демонстрационная база данных. В приложении к разделу 5.4 задачника-практикума содержатся

ся разнообразные таблицы, которые можно использовать в этих целях.

Представляя ученикам демонстрационную базу данных, необходимо обратить внимание на то, что наряду с самой таблицей в памяти компьютера хранится описание ее структуры, откуда пользователь может узнать параметры полей: имя, тип, формат и др. В СУБД Access это делается в режиме «Таблица-Конструктор».

Основная задача любой информационной системы — поиск информации в базе данных. Поиск происходит по запросу пользователя. На языке гипотетической СУБД команда запроса имеет следующую структуру:

.выбрать <список выводимых полей>
для <условие поиска>

В результате выполнения запроса получается таблица, состоящая из полей, указанных в команде. В эту таблицу включается информация из тех записей, которые удовлетворяют условию поиска. Условие поиска представляет собой *логическое выражение*. Здесь мы встречаемся еще с одной фундаментальной составляющей данной темы курса — основами математической логики. Помимо прикладного применения в информатике эта тема имеет большое общеобразовательное значение.

В учебнике дано следующее определение: логическое выражение — это некоторое высказывание, по поводу которого можно заключить, истинно оно или ложно. Применительно к базам данных, определение можно перефразировать так: *логическое выражение — это некоторое высказывание по поводу значений полей базы данных; это высказывание по отношению к разным записям может быть истинным или ложным.*

Логические выражения разделяются на простые и сложные. В простых выражениях всегда используется лишь одно поле таблицы, и не применяются логические операции. В сложных логических выражениях

используются логические операции. Простое логическое выражение представляет собой либо имя поля логического типа либо отношение (в математике говорят «неравенство»). В § 30 подробно объясняются правила вычисления отношений для полей разных типов. Отношения для числовых величин сохраняют смысл математических неравенств; при вычислении отношений для символьных величин учитывается лексикографический порядок; даты сравниваются в порядке их календарной последовательности.

Основная проблема — научить учеников формальному представлению условий поиска в виде логических выражений. Например, от фразы «найти все книги, лежащие выше пятой полки» нужно перейти к логическому выражению $\text{ПОЛКА} > 5$; или условие «выбрать всех неуспевающих по физике» следует представить в виде $\text{ФИЗИКА} < 3$; а условие «выбрать все дни, когда шел дождь», — в виде $\text{ОСАДКИ} = \text{дождь}$.

Особое внимание надо обратить на использование полей логического типа в условиях поиска. Обычно к ним не применяются отношения. Логическое поле само несет логическое значение «истина» или «ложь». Например, условие «выбрать всех учеников, посещающих танцы» представится одним именем логического поля: ТАНЦЫ . Закреплению этих понятий помогут задания к § 30.

Сложные логические выражения содержат в себе логические операции. Рассматриваются три основные операции математической логики: конъюнкция (И), дизъюнкция (ИЛИ), отрицание (НЕ).

Обычно при объяснении этого вопроса учитель отталкивается от семантического смысла высказываний на русском языке, содержащих союзы «и», «или», частицу «не». Например, высказывание «Сегодня будет контрольная по алгебре И по физике» справедливо, если состоятся обе контрольные и ложно, если хотя бы одна не состоится. Другое высказывание: «Сегодня будет контрольная по алгебре ИЛИ по физике» будет истинным, если состоится хотя бы одна контрольная ра-

бота. И, наконец, высказывание: «Сегодня НЕ будет контрольной» истинно, если контрольная не состоится. Из подобных примеров учитель делает выводы о правилах выполнения логических операций: если A и B — логические величины, то:

- A И B истинно в том и только в том случае, если истинны оба операнда;
- A ИЛИ B ложно в том и только в том случае, если ложны оба операнда;
- НЕ A меняет значение логической величины на противоположное: НЕ истина — ложь; НЕ ложь — истина.

Эти правила сводятся в таблицы истинности.

При переходе к составлению условий поиска в базах данных ученики нередко попадают в «ловушки естественного смысла». Например, задача формулируется так: из БД «Библиотека» получить сведения обо всех книгах Толстого и Тургенева. Абсолютное большинство учеников записывают это условие следующим образом:

АВТОР = "Толстой" И АВТОР = "Тургенев"

Прозвучавший в задании союз «и» машинально переносится в логическое выражение. После этого учителю приходится объяснять, что автором книги не может быть одновременно и Толстой и Тургенев. Поэтому в библиотеке нет ни одной книги, удовлетворяющей такому условию. Здесь следует применить логическую операцию ИЛИ:

АВТОР = "Толстой" ИЛИ АВТОР = "Тургенев"

Тогда будет получена искомая выборка книг обоих авторов.

Операция ИЛИ объединяет в одну выборку записи, удовлетворяющие либо одному, либо другому условию. Операция И работает иначе: сначала выбираются все записи, удовлетворяющие первому условию, затем из отобранных записей выбираются те, которые удовлетворяют второму условию. В учебнике используется

представление о порядке вырезания и склеивания записей при отборе по заданному условию поиска.

Полезно выполнять с учениками несколько формальных заданий на обработку сложных условий поиска. Например, нарисуйте на доске следующую таблицу:

Записи	Поля		
	A	B	C
R1	1	2	3
R2	1	3	1
R3	2	2	2
R4	3	3	3
R5	3	2	3

Предложите серию заданий такого содержания: задано условие поиска в форме логического выражения; определить, какие записи ему удовлетворяют.

Условие:

Ответ:

- | | |
|------------------------------|----------------------|
| 1) $A=1$ И $B=2$ | : R1 |
| 2) $A=1$ ИЛИ $A=3$ | : R1, R2, R4, R5 |
| 3) $A=1$ ИЛИ $B=2$ | : R1, R2, R3, R5 |
| 4) $A=1$ ИЛИ $B=2$ ИЛИ $C=3$ | : R1, R2, R3, R4, R5 |
| 5) $A=1$ И $B=2$ И $C=3$ | : R1 |
| 6) НЕ $A=1$ | : R3, R4, R5 |

На примере этой же таблицы отрабатывается вопрос о старшинстве операций и порядке их выполнения. Сообщив, что логические операции по убыванию старшинства расположены так: НЕ, И, ИЛИ, приведите примеры логических выражений, содержащих разные операции:

- | | |
|-------------------------------|------------------|
| 7) $A=1$ И $B=2$ ИЛИ $C=3$ | : R1, R4, R5 |
| 8) $A=1$ ИЛИ $B=2$ И $C=3$ | : R1, R2, R5 |
| 9) НЕ $A=1$ ИЛИ $B=2$ И $C=3$ | : R1, R3, R4, R5 |
| 10) $(A=1$ ИЛИ $B=2)$ И $C=3$ | : R1, R5 |

После решения таких формальных задач следует снова вернуться к содержательным задачам. Теперь ученики будут гораздо успешнее формализовать сложные условия поиска в логические выражения. Например, требуется выбрать все книги Беляева и Толстого, расположенные от 2-й до 5-й полки. Логическое выражение запишется так:

(АВТОР="Толстой Л.Н." ИЛИ
АВТОР="Беляев А.Р.") И ПОЛКА>=2 И ПОЛКА<=5

Переходя к работе с конкретной СУБД, учитель знакомит учеников с правилами формирования в ней команды запроса. Если нет диалоговой оболочки и команды задаются посимвольным вводом, то нужно описывать синтаксис команд, обращая внимание на все детали. Например, на языке dBASE команда вывода на экран записей с одним из рассмотренных выше условий поиска будет выглядеть так:

.DISPLAY FOR A=1 .AND. B=2 .OR. C=3

При использовании диалоговой оболочки (например, работа в режиме ассистента в СУБД «Ребус») команда формируется в диалоге. Система подсказывает пользователю каждый шаг и предоставляет меню для выбора очередного элемента команды. При этом пользователь видит на экране формируемую команду.

В СУБД Access для создания запросов используется *Конструктор запросов*. Формируемая команда указывается *скрытый* от пользователя. Работа с конструктором требует определенных навыков, которые следует отрабатывать на упражнениях. Целесообразно поступать следующим образом: выполнение любого задания на поиск данных в БД начинать с записи в тетради команды на языке гипотетической СУБД, а затем, перейдя в режим создания запроса на выборку, соответствующим образом заполнить поля Конструктора.

В СУБД Access используется своеобразный табличный способ представления условий поиска. В ячейках таблицы Конструктора запросов записываются условия, накладываемые на значения соответствующих полей. Условия, стоящие в одной строке, должны выполняться одновременно, то есть они соединены между собой операцией И; условия в разных строках соединены операцией ИЛИ. Таблица играет роль фильтра при отборе записей из БД: сначала отбираются записи, удовлетворяющие условиям первой строки, затем к ним добавляются записи, удовлетворяющие условиям второй строки и так далее.

В следующей таблице приведены примеры реализации логических выражений табличным методом, применяемым в конструкторе запросов СУБД Access. Используются условия поиска из рассмотренного выше формального примера.

Условие	A	B	C
1) A=1 И B=2	=1	=2	
2) A=1 ИЛИ A=3	=1		
	=3		
3) A=1 ИЛИ B=2	=1	.	
		=2	
4) A=1 ИЛИ B=2 ИЛИ C=3	=1		
		=2	
			=3
5) A=1 И B=2 И C=3	=1	=2	=3

Условие	A	B	C
6) НЕ A=1	<>1		
7) A=1 И B=1 ИЛИ C=3	=1	=2	
			=3
8) A=1 ИЛИ B=2 И C=3	=1		
		=2	=3
9) НЕ A=1 ИЛИ B=2 И C=3	<>1		
		=2	=3
10) (A=1 ИЛИ B=2) И C=3	=1		=3
		=2	=3

Обратите внимание на последнее условие. При записи в таблицу фактически произошло раскрытие скобок, и данное логическое выражение заменилось эквивалентным выражением:

$$A = 1 \text{ И } C = 3 \text{ ИЛИ } B = 2 \text{ И } C = 3$$

В ячейках конструктора условий отбора можно непосредственно записывать логические операции. Например, условие поиска в БД «Библиотека», представленное логическим выражением:

(АВТОР="Толстой Л.Н." ИЛИ
АВТОР="Беляев А.Р.") И ПОЛКА>=2 И ПОЛКА<=5

в конструкторе запроса на выборку может быть реализовано следующими двумя способами:

АВТОР	ПОЛКА
= "Толстой Л.Н."	>=2 and <=5
= "Беляев А.Р."	>=2 and <=5

АВТОР	ПОЛКА
= "Толстой Л.Н." or = "Беляев А.Р."	>=2 and <=5

Условия поиска в форме логических выражений используются и в командах удаления записей из таблицы. Команда имеет следующую структуру:

.удалить для <логическое выражение>

В Access с помощью конструктора формируется запрос на удаление, и условие для отбора удаляемых записей задается в табличной форме так же, как и в запросе на выборку.

Еще один важный вид манипулирования информацией в базе данных — *сортировка записей*. Здесь основными понятиями, которые должны усвоить ученики, являются «ключ сортировки» и «порядок сортировки». Ключ сортировки — это поле, по значению которого происходит упорядочение записей в таблице. Порядок сортировки имеет два варианта: по возрастанию значений ключа и по убыванию значений. Если ключей несколько, то среди них устанавливается иерархия: первичный ключ, вторичный ключ и так далее. В первую очередь записи сортируются по значению первичного ключа; внутри группы записей с одинаковыми значениями первичного ключа происходит сортировка по вторичному ключу и так далее. СУБД Access позволяет сортировать записи как во всей исходной таблице, так и в таблицах, получаемых в результате выполнения запроса на выборку.

7. При углубленном изучении раздела «Базы данных» ученики знакомятся с элементами проектирования БД. Материал на эту тему содержится в разделе 7.2 второй части учебника. Проектирование БД — сложная задача. Лишь на первый взгляд она может показаться простой. Для небольших учебных БД ошибки при проектировании не столь существенны. Но если создается большая база, в которой будут сохраняться многие тысячи записей, то ошибки при проектировании могут стоить очень дорого. Основные последствия неправильного проектирования — избыточность информации, ее противоречивость, потеря целостности, то есть взаимосвязи между данными. В результате БД может оказаться неработоспособной и потребовать дорогостоящей переделки.

Теория реляционных баз данных была разработана в 70-х годах Е. Коддом. Кодд предложил технологию проектирования баз данных, в результате применения которой в полученной БД не возникает отмеченных выше недостатков (см., например, книгу [25]). Сущность этой технологии сводится к приведению таблиц, составляющих базу данных, к третьей нормальной форме. Этот процесс называется *нормализацией данных*: сначала все данные, которые планируется включить в БД, представляются в первой нормальной форме, затем преобразуются ко второй и на последнем шаге — к третьей нормальной форме. Проиллюстрируем процесс нормализации данных на примере, который использован в разделе 5.4 задачника-практикума. Ставится задача: создать базу данных, содержащую сведения о посещениях пациентами поликлиники своего участкового врача. Сначала строится одна таблица, в которую заносятся фамилия пациента, его дата рождения, номер участка, к которому приписан пациент, фамилия участкового врача, дата посещения врача и установленный диагноз болезни. Ниже приведен экземпляр такой таблицы:

БД «Поликлиника»

ФАМИЛИЯ	ДАТА РОЖДЕНИЯ	УЧАСТОК	ВРАЧ	ДАТА ПОСЕЩЕНИЯ	ДИАГНОЗ
Лосев О.И.	20.04.65	2	Петрова О.И.	11.04.98	грипп
Орлова Е.Ю.	25.01.47	1	Андреева И.В.	05.05.98	ОРЗ
Лосев О.И.	20.04.65	2	Петрова О.И.	26.07.98	бронхит
Дуров М.Т.	05.03.30	2	Петрова О.И.	14.03.98	стено- кардия
Жукова Л.Г.	30.01.70	2	Петрова О.И.	11.04.98	ангина
Орлова Е.Ю.	25.01.47	1	Андреева И.В.	11.07.98	гастрит
Быкова А.А.	01.04.75	1	Андреева И.В.	15.06.98	ОРЗ
Дуров М.Т.	05.03.30	2	Петрова О.И.	26.07.98	ОРЗ

Нетрудно понять недостатки такой организации данных. Во-первых, очевидна избыточность информации: повторение даты рождения одного и того же человека, повторение фамилии врача одного и того же участка. В такой БД велика вероятность иметь недостоверные, противоречивые данные. Например, если на втором участке сменится врач, то придется просматривать всю базу и вносить изменения во все записи, относящиеся к этому участку. При этом велика вероятность что-то пропустить. После каждого нового посещения пациентом больницы потребуется снова вводить его дату рождения, номер участка, фамилию врача, то есть информацию, уже существующую в БД.

Полученная таблица соответствует первой нормальной форме. Для устранения отмеченных недостатков требуется ее дальнейшая нормализация. Структура такой таблицы (отношения) описывается следующим образом:

ПОЛИКЛИНИКА (ФАМИЛИЯ, ДАТА_РОЖДЕНИЯ, УЧАСТОК,
ВРАЧ, ДАТА_ПОСЕЩЕНИЯ, ДИАГНОЗ)

Необходимо установить ключ записей. Здесь ключ составной, который включает в себя два поля: «ФАМИЛИЯ» и «ДАТА_ПОСЕЩЕНИЯ». Каждая запись — это информация о конкретном посещении пациентом больницы. Если допустить, что в течение одного дня данный пациент может сделать только один визит к участковому врачу, то в разных записях не будет повторяться комбинация двух полей: фамилии пациента и даты посещения врача.

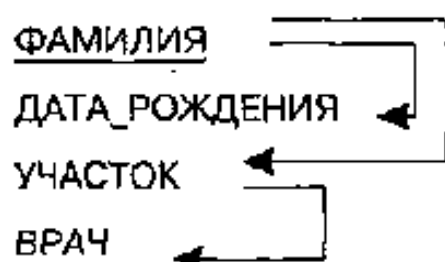
Согласно определению второй нормальной формы, все неключевые поля должны функционально зависеть от полного ключа. В данной таблице лишь «ДИАГНОЗ» определяется одновременно фамилией пациента и датой посещения. Остальные поля связаны лишь с фамилией, т. е. от даты посещения они не зависят. Для преобразования ко второй нормальной форме таблицу нужно разбить на две следующие:

ПОСЕЩЕНИЯ (ФАМИЛИЯ, ДАТА_ПОСЕЩЕНИЯ, ДИАГНОЗ)

ПАЦИЕНТЫ (ФАМИЛИЯ, ДАТА_РОЖДЕНИЯ, УЧАСТОК, ВРАЧ)

В отношении таблицы «ПОСЕЩЕНИЯ» по-прежнему действует составной ключ из двух полей, а в отношении таблицы «ПАЦИЕНТЫ» — одно ключевое поле «ФАМИЛИЯ».

Во втором отношении имеется так называемая транзитивная зависимость. Она отображается следующим образом:



Значение поля ВРАЧ связано с фамилией пациента транзитивно через поле УЧАСТОК. В самом деле, вся-

кий участковый врач приписан к своему участку и обслуживает больных, относящихся к данному участку.

Согласно определению третьей нормальной формы, в отношении не должно быть транзитивных зависимостей. Значит, требуется еще одно разбиение — отношения «ПАЦИЕНТЫ» на два отношения.

В итоге получаем базу данных, состоящую из трех отношений:

ПОСЕЩЕНИЯ (ФАМИЛИЯ, ДАТА ПОСЕЩЕНИЯ, ДИАГНОЗ)

ПАЦИЕНТЫ (ФАМИЛИЯ, ДАТА РОЖДЕНИЯ, УЧАСТОК)

ВРАЧИ (УЧАСТОК, ВРАЧ)

В третьем отношении ключом является номер участка, поскольку он повторяться не может. В то же время возможна ситуация, когда один врач обслуживает больше одного участка. Полученная структура БД удовлетворяет требованиям третьей нормальной формы: в таблицах все неключевые поля полностью функционально зависят от своих ключей и отсутствуют транзитивные зависимости.

Еще одним важным свойством полученной БД является то, что между тремя отношениями существует взаимосвязь через общие поля. Отношения «ПОСЕЩЕНИЯ» и «ПАЦИЕНТЫ» связаны общим полем «ФАМИЛИЯ». Отношения «ПАЦИЕНТЫ» и «ВРАЧИ» связаны через поле «УЧАСТОК». Для связанных таблиц существует еще одно понятие: тип связи. Возможны три варианта типа связей: «один к одному», «один ко многим», «многие ко многим». В нашем примере между связанными таблицами существуют связи типа «один ко многим», и схематически они отображаются так:

ВРАЧИ $\longleftrightarrow$ ПАЦИЕНТЫ $\longleftrightarrow$ ПОСЕЩЕНИЯ

Смысл следующий: у каждого врача (на каждом участке) много пациентов; каждый пациент посещает врача множество раз.

В приведенном примере показана процедура нормализации в строгом соответствии с теорией реляцион-

ных баз данных. Понимание смысла этой процедуры очень полезно для учителя.

В учебнике не дается подробного описания процедуры нормализации, не приводится строгого определения трех нормальных форм. Используется нетрадиционный термин «хорошо нормализованная база данных». В этом понятии фактически заложены свойства третьей нормальной формы. Эти свойства сформулированы так: «Все поля таблицы должны отражать непосредственные характеристики (атрибуты) объекта, к которому относится запись». Ученикам предлагается следующая, в некотором смысле, интуитивная методика получения хорошо нормализованной БД. *Все множество данных нужно разделить между различными объектами, к которым они относятся.* На примере приведенной выше таблицы «ПОЛИКЛИНИКА» нужно увидеть три различных типа объектов, к которым относится данная информация: это пациенты поликлиники, врачи и посещения пациентами врачей. Соответственно строятся три таблицы, содержащие атрибуты, относящиеся к этим трем типам объектов.

Рекомендации по организации практической работы

Практические задания делятся на три типа:

- 1) *задачи*: теоретические задания для закрепления основных понятий;
- 2) *упражнения*: практические задания для работы в среде СУБД с целью отработки отдельных навыков;
- 3) *индивидуальные работы*: зачетные задания, требующие от учеников комплексного владения теоретическими знаниями и практическими навыками.

Задания трех таких типов приведены в разделе 5.4 задачника-практикума. Задачи могут быть предметом коллективного разбора на уроке, использоваться в контрольных и домашних заданиях.

Упражнения выполняются на компьютере. Учитель может использовать материал упражнений в ходе

объяснений для демонстрации приемов работы с базой данных средствами изучаемой СУБД. После такого объяснения следует предложить ученикам коллективно выполнить другие упражнения того же типа. Рабочий материал для упражнений (файлы с таблицами) учитель должен подготовить заранее.

Варианты индивидуальных работ учитель распределяет среди учеников по своему усмотрению. Возможен вариант, когда все делают одно и то же задание. В таком случае учителю легче осуществлять контроль, оказывать помощь ученикам. По результатам выполнения работы каждый ученик получает оценку.

Подбор практического материала в разделе 5.4 позволяет организовать изучение темы «Базы данных» на разных уровнях, использовать различные СУБД. Весь комплект заданий ориентирован на углубленное изучение темы, на использование развитых версий реляционных СУБД, в том числе MS Access. Задача учителя — сделать выборку заданий для организации практической работы в соответствии с поставленными целями обучения, резервом учебного времени, имеющимися техническими и программными средствами.

14. Табличные вычисления на компьютере

Разделы учебника: часть I, § 33–37. Дополнительный материал: часть II, раздел 8.1.

Основные цели. Познакомить учащихся с назначением и структурой электронных таблиц. Обучить их основным приемам работы с табличным процессором. Научить детей организации простых табличных расчетов с помощью электронных таблиц.

Изучаемые вопросы

- Назначение и структура электронных таблиц (ЭТ).
- Табличный процессор: среда, режимы работы.

- Информация в ячейках таблицы.
- Правила записи формул.
- Принцип относительной адресации.
- Организация табличных вычислений.
- Сортировка данных в ЭТ.
- Графическая обработка данных.
- Абсолютная адресация.
- Логические функции.

Предметом изучения данного раздела являются *электронные таблицы (ЭТ)*. Появление электронных таблиц исторически совпадает с началом распространения персональных компьютеров. Первая программа для работы с электронными таблицами — табличный процессор, была создана в 1979 году, предназначалась для компьютеров типа Apple II и называлась VisiCalc. В 1982 году появляется знаменитый табличный процессор Lotus 1-2-3, предназначенный для IBM PC. Lotus объединял в себе вычислительные возможности электронных таблиц, деловую графику и функции реляционной СУБД. Популярность табличных процессоров росла очень быстро. Появлялись новые программные продукты этого класса: Multiplan, Quattro Pro, SuperCalc и другие. Одним из самых популярных табличных процессоров сегодня является MS Excel, входящий в состав пакета Microsoft Office.

Что же такое электронные таблицы? Это средство информационных технологий, позволяющее решать целый комплекс задач:

1) прежде всего, выполнение вычислений. Издавна многие расчеты выполняются в табличной форме, особенно в области делопроизводства: многочисленные расчетные ведомости, табуляграммы, сметы расходов и т. п. Кроме того, решение численными методами целого ряда математических задач удобно выполнять в табличной форме. Электронные таблицы представляют собой удобный инструмент для автоматизации таких вычислений. Решения многих вычислительных задач на ЭВМ, которые раньше можно было осуществить то-

лько путем программирования, стало возможно реализовать на электронных таблицах;

2) математическое моделирование. Использование математических формул в ЭТ позволяет представить взаимосвязь между различными параметрами некоторой реальной системы. Основное свойство ЭТ — мгновенный пересчет формул при изменении значений входящих в них операндов. Благодаря этому свойству, таблица представляет собой удобный инструмент для организации численного эксперимента: подбор параметров, прогноз поведения моделируемой системы, анализ зависимостей, планирование. Дополнительные удобства для моделирования дает возможность графического представления данных;

3) использование электронной таблицы в качестве базы данных. Конечно, по сравнению с СУБД электронные таблицы имеют меньшие возможности в этой области. Однако некоторые операции манипулирования данными, свойственные реляционным СУБД, в них реализованы. Это поиск информации по заданным условиям и сортировка информации.

При первоначальном знакомстве с электронными таблицами удобно оттолкнуться от известных ученикам представлений о реляционных базах данных. Объяснение нужно строить на конкретном примере. В учебнике использован пример таблицы учета продажи молочных продуктов в магазине. Рассматривая такую таблицу как базу данных, ее поля можно разделить на два типа: независимые, то есть содержащие некоторую исходную информацию, и вычисляемые по определенным формулам. К независимым полям относятся цена единицы товара, количество поставленного товара, количество проданного товара. К вычисляемым полям относятся остаток, равный разности между количествами поставленного и проданного товаров, выручка от продажи, равная произведению количества проданного товара на цену. Основным свойством электронной таблицы является возможность мгновенного пересче-

та вычисляемых полей при изменении значений операндов, входящих в формулы.

Возможность размещения в ячейках таблицы формул — это первая базовая идея электронных таблиц. Вторая базовая идея — это принцип относительной адресации. Но об этом немного позже.

На уроках ученикам предстоит освоить конкретный табличный процессор. Как и в предыдущих темах курса, рассматривающих информационные технологии, рекомендуется придерживаться методической схемы виртуального исполнителя, элементами которой является изучение среды, режимов работы, системы команд, данных.

Методические рекомендации по изложению теоретического материала

При работе с табличным процессором (ТП) создается документ, который называется *электронной таблицей*. Электронная таблица формируется в оперативной памяти компьютера. В дальнейшем ее можно просматривать, изменять, записывать на магнитный диск для хранения, печатать на принтере.

Среда ТП

При работе с табличным процессором на экран выводится рабочее поле таблицы и панель диалога. Электронная таблица представляется в виде матрицы, состоящей из строк и столбцов. Строки нумеруются сверху вниз, начиная с 1. Столбцы именуются латинскими буквами (одно- и двухбуквенными именами) в алфавитном порядке в направлении слева направо. Число строк и столбцов зависит от конкретного типа ТП.

На пересечении строк и столбцов образуются ячейки, каждая из которых имеет свое обозначение (имя, адрес), состоящее из имени столбца и номера строки: A1, C5, AB356 и т. п.

На экране дисплея видна не вся электронная таблица (документ), а только ее часть. Документ в полном

объеме хранится в оперативной памяти, а экран можно считать «окном», через которое пользователь имеет возможность просматривать его. Кроме основного раздела памяти, где хранится электронная таблица, используются еще дополнительные разделы:

- буфер для хранения копируемых фрагментов таблицы;
- раздел памяти для размещения справочной информации.

Важным элементом электронной таблицы является *табличный курсор* — прямоугольник, выделенный цветом или рамкой. Ячейка таблицы, которую в данный момент занимает курсор, называется *текущей ячейкой*. При перемещении курсора по таблице происходит перемещение «окна» по документу, в результате чего становятся видны различные его части. Часть таблицы, заполненная информацией, называется *активной таблицей*.

	А	В	С	Д		П	Ю
1							
2		КУРСОР					
3							
...			ЭКРАН				
	АКТИВНАЯ ТАБЛИЦА						
9999	ПОЛНАЯ ТАБЛИЦА						

Рис. 14.1. Структура электронной таблицы

У разных табличных процессоров может различаться расположение на экране рабочего поля и панели диалога. На рис. 14.2 изображена среда ТП SuperCalc, работающего в операционной системе MS DOS. На панели диалога меню команд постоянно не отражается и вызывается нажатием клавиши «\» — слэш. Для таб-

личных процессоров, работающих в ОС Windows (в т. ч. Excel), свойственно наличие панели инструментов, линеек прокрутки. На рис. 14.3 показана характерная структура среды для таких ТП.

	A	B	C	D				
1								
2		КУРСОР						
3				РАБОЧЕЕ	ПОЛЕ			
...								
B2 (строка состояния)								
Width: 9 Memory: 137 Last Col/Row: (строка запроса)								
B2								
1> (строка ввода)								
READY F1: Help F3: Names (строка подсказки)								

Рис. 14.2. Среда электронной таблицы SuperCalc

Главное меню								
Вспомогательная область управления								
B2 = Строка ввода								
	A	B	C	D	E	F	G	H
1								
2		КУРСОР						
3			РАБОЧЕЕ ПОЛЕ					
4								
5								
...								
Строка подсказки								

Рис. 14.3. Среда электронной таблицы Excel

Строка ввода предназначена для отражения вводимых в текущую ячейку данных. В этой строке мож-

но просмотреть и отредактировать хранимую в ячейке формулу; в самой ячейке пользователь видит результат вычисления по формуле.

Главное меню содержит основные команды управления электронной таблицей; представляет собой иерархическую систему команд. Команды, вызывающие открытие подменю, можно назвать режимами. Исполняемые команды инициируют определенные действия над электронной таблицей.

Строка подсказки используется для вывода сообщений, подсказывающих пользователю возможные действия при данном состоянии таблицы.

Вспомогательная область управления включает в себя панель инструментов, линейки прокрутки, строку состояния.

Основные режимы работы

Режим готовности. В этом режиме происходит выбор текущей ячейки или выделение блока ячеек.

Режим ввода данных. Происходит посимвольный ввод данных с клавиатуры в текущую ячейку.

Режим редактирования. Используется при необходимости отредактировать содержимое ячейки без полной его замены.

Командный режим. Режим выбора и исполнения команд из иерархической системы меню. После выполнения команды происходит возврат к режиму готовности.

Кроме перечисленных основных режимов работы ТП можно говорить о режимах отображения таблицы и режимах управления вычислениями.

Режимы отображения таблицы. В ячейках, хранящих формулы, могут отображаться результаты вычисления по формулам или сами формулы. Первый режим называется *режимом отображения значений*, второй — *режимом отображения формул*. Рабочим состоянием таблицы является режим отображения значений. Ре-

жим отображения формул может использоваться при формировании и отладке таблицы.

Режим управления вычислениями. Табличный процессор производит вычисления по формулам, сканируя таблицу в определенном порядке. Такое сканирование всегда начинается с клетки A1. Порядок вычислений может быть установлен *по строкам* или *по столбцам*. Некоторые ТП позволяют устанавливать этот порядок по желанию пользователя.

При каждом вводе новых данных в ячейку вся таблица автоматически пересчитывается заново (режим *автоматического пересчета*). В некоторых ТП существует возможность установки *режима ручного пересчета*, то есть таблица заново пересчитывается только после подачи специальной команды.

Система команд

Команды ТП организованы в иерархическую систему, верхним уровнем которой является главное меню. Кроме того, выполнение команд может инициироваться через панель инструментов, контекстное меню, «горячие клавиши».

Команды редактирования таблицы позволяют манипулировать с фрагментами таблицы: *удалять, копировать, перемещать, вставлять*. Вставки и удаления столбцов или строк приводят к сдвигу других строк или столбцов таблицы. При этом действующая в таблице относительная адресация автоматически модифицирует формулы в соответствии с их изменившимися адресами. Прием копирования позволяет быстро строить большие таблицы, содержащие однотипные элементы.

Команды форматирования позволяют изменять внешний вид таблицы, ее оформление. К элементам формата относятся:

- направления выравнивания данных относительно границ ячейки;
- высота строки и ширина столбца;

- тип, начертание и размер шрифта;
- формат представления чисел (обычный, экспоненциальный, разрядность);
- вид разлиновки таблицы;
- цвет фона и пр.

В электронной таблице действует некоторый набор стандартных параметров формата «по умолчанию». Командами форматирования его можно изменять как по отношению ко всей таблице, так и в отдельных ее фрагментах.

Команды работы с файлами включают в себя стандартный набор команд, позволяющих открывать и сохранять файлы, организовывать вывод на печать полученного документа.

Команды работы с таблицей как с базой данных. Способность ТП искать и выбирать данные из таблицы позволяет использовать электронную таблицу в качестве несложной базы данных. При работе с базами данных имеют дело с *записями и полями*. В электронных таблицах базой данных является сама таблица, записями — строки таблицы, полями — ячейки таблицы. В ТП реализованы команды *поиска и сортировки*.

Чтобы организовать *поиск и извлечение данных*, необходимо задать:

- *входной блок*, то есть диапазон ячеек, в котором хранятся данные (записи и поля); важное требование: все строки в этом блоке должны быть однородны;
- *блок критериев*, то есть диапазон, содержащий условие, в соответствии с которым осуществляется поиск и выборка данных из входного блока;
- *выходной блок*, то есть диапазон, в который будут извлечены данные из входного блока, в соответствии с условием, содержащимся в блоке критериев. Задание этих блоков осуществляется специальными командами.

Сортировка строк таблицы производится по значениям определенного столбца. В команде указывается

порядок сортировки: по возрастанию или убыванию значений (в том же смысле, что и в БД).

Команды графической обработки данных дают возможность отображать числовую информацию в графическом виде, чаще всего — в виде диаграмм. Команды графического режима можно разбить на две группы:

- команды описания диаграмм (задают данные, которые будут выведены в графическом виде, задают тип диаграмм и так далее.);
- команды вывода диаграмм.

Данные в ячейках таблицы

Данные для табличных процессоров — это информация, содержащаяся в ячейках таблицы, представленная в определенной символьной форме.

Здесь мы снова встречаемся с такими свойствами данных, как тип и структура, знакомыми из темы о базах данных. Кроме того, появляются новые темы в разговоре о данных. Это понятия константы и переменной, арифметического и логического выражений, адресации.

Содержимым ячейки электронной таблицы может быть формула или текст. Частным случаем формулы является числовая константа или переменная. Более общим — арифметическое или логическое выражение.

Типы данных. Табличный процессор должен «знать», какого типа данное хранится в конкретной ячейке таблицы для того, чтобы правильно интерпретировать ее содержимое. Так, например, признаком текстовых данных является символ " (двойные кавычки). Тип данных определяется множеством значений, принимаемых величиной, и совокупностью операций, применимых к величинам этого типа. Отсюда, например, следует, что нельзя применять арифметические операции к ячейкам таблицы, в которых хранится текстовая информация. Основной набор типов данных в электронных таблицах практически тот же, что и в базах данных: числовой, символьный, логический типы. В некоторых ТП (Excel) существует тип «дата».

Структуры данных. Минимальным структурным элементом данных, представленных в электронной таблице, является ячейка. Основная работа производится с ячейками: их заполняют, редактируют, очищают.

Ячейки объединяются в структуры данных — столбцы и строки. Табличные процессоры позволяют оперировать со строками или столбцами как единым целым. Например, можно удалять или вставлять строки (столбцы), менять их местами.

Базовым структурным понятием в электронных таблицах является понятие *диапазона ячеек* (блока). Оно используется во многих командах табличных процессоров и в некоторых функциях. Диапазон — это множество ячеек, образующих в таблице область прямоугольной формы (матрицу). Минимальный диапазон — это ячейка, строка и столбец также являются блоком, максимальный диапазон — вся таблица. Некоторые табличные процессоры позволяют задавать имя для диапазона ячеек, что дает возможность работать с блоком как единым целым.

Числовые константы разделяются на целые и вещественные. Вещественные константы можно записывать двумя способами: в форме с фиксированной точкой и в экспоненциальной форме (в форме с плавающей точкой).

Запись числовой константы с фиксированной точкой предполагает, что число содержит целую и дробную части, разделенные десятичной точкой. Например, числовая константа $-3,1415$ записывается как -3.1415 (в ТП Excel может употребляться запятая). При записи числовой константы в экспоненциальной форме сначала записывается мантисса, затем — латинская буква E (прописная или строчная), после нее — порядок. Мантисса может быть записана, как целая константа или константа с фиксированной точкой, а порядок — только как целая двузначная константа. Числовая константа в экспоненциальной форме трактуется, как мантисса, умноженная на 10 в степени, равной порядку. На-

пример, числа 0,0001 и 1 000 000 могут быть записаны следующим образом: $1E-4$ или $0.1e-3$ и $1e6$ или $1E+6$.

Переменные. Каждую ячейку таблицы будем интерпретировать как ячейку памяти табличного процессора. Каждая ячейка имеет свое имя, состоящее из имени столбца и номера строки. В каждой ячейке может храниться информация того или иного вида. Здесь прослеживается прямая аналогия с понятием переменной в языках программирования. Переменная — это поименованное место в памяти (ячейка), куда можно записать значение. Переменная принимает различные значения определенного типа. Каждая переменная обозначается символическим именем (идентификатором). Ячейку таблицы можно рассматривать как переменную. Следовательно, A1, C5, G10 и пр. — имена переменных.

Выражения (формулы). В электронных таблицах используются два вида выражений: арифметические и логические. Выражение, определяющее способ вычисления некоторого числового значения по математической формуле, называется *арифметическим выражением*. Существуют определенные правила записи арифметических выражений. Эти правила аналогичны тем, что используются в языках программирования.

Выражения состояются из констант, переменных, знаков операций, функций, круглых скобок. Примеры выражений:

$2.5*(G5+G2)$ $SQRT(B4^2-4*B3*B5)$ $SUM(C10:C20)$

Здесь символ «*» — знак умножения; «^» — возведение в степень. В TI SuperCalc $SQRT$ — функция вычисления квадратного корня; SUM — функция суммирования. В русифицированных версиях Excel функция квадратного корня — **КОРЕНЬ**, функция суммирования — **СУММ**. Порядок вычисления выражений зависит от приоритетов выполнения арифметических операций, расстановки скобок. Приобретение учениками навыков записи формул требует практических

упражнений. Задачи такого рода имеются в задачнике-практикуме.

Логические выражения (логические формулы) строятся с помощью операций отношения ($<$, $>$, $=$, $<=$, $>=$, $<>$) и логических операций (логическое И, логическое ИЛИ, логическое отрицание НЕ). Результатом вычисления логического выражения является логическая величина «истина» или «ложь». С логическими выражениями ученики уже встречались при работе с базами данных.

Особенность логических выражений для электронных таблиц заключается в том, что логические операции используются как функции: сначала записывается имя логической операции: И, ИЛИ, НЕ (AND, OR, NOT), а затем в круглых скобках перечисляются логические операнды. Например, логическое выражение

$$\text{И}(A1>0, A1<1)$$

соответствует математическому неравенству $0 < A1 < 1$.

С помощью логических выражений, в частности, задаются условия, проверку которых осуществляет условная функция. Форма записи условной функции в значительной мере зависит от типа табличного процессора. Если в ячейку заносится условная функция, то на экране отображается результат ее вычисления, то есть то или иное значение в зависимости от условия, заданного логическим выражением. Обычно условная функция имеет такую структуру:

$$\text{IF}(\text{условие}, \text{действие1}, \text{действие2}).$$

Если условие истинно, то выполняется действие1 иначе — действие2. Условная функция может иметь вложенную структуру. Используя ее, можно строить простейшие алгоритмы ветвления, например, вычисление разрывных функций:

$$\text{sign } x = \begin{cases} 1, & \text{если } x > 0, \\ 0, & \text{если } x = 0, \\ -1, & \text{если } x < 0; \end{cases}$$

$$F(x) = \begin{cases} |x|, & \text{если } -1 < x < 1, \\ 1 & \text{в противном случае.} \end{cases}$$

Если значение аргумента x хранится в ячейке A1, то вычисление функции $\text{sign}(x)$ произойдет по следующему выражению (приводится вариант для ТП Super-Calc):

$$\text{IF}(A1>0, 1, \text{IF}(A1<0, -1, 0))$$

Вычисление функции $F(x)$:

$$\text{IF}(\text{AND}(A1>-1, A1<1), \text{ABS}(A1), 1).$$

Адресация

Следует обратить внимание учеников на определенную родственность структуры электронной таблицы и оперативной памяти ЭВМ. В обоих случаях используется принцип адресации для хранения и поиска информации. Разница состоит в том, что в ОЗУ наименьшей адресуемой единицей является байт, а в таблице — ячейка.

Создавая формулы, мы указываем на ячейки, в которые будем обращаться. В то же время, их адресами в таблице. В таблице может быть установлен режим относительной адресации или режим абсолютной адресации. В режиме относительной адресации всякие изменения в местоположении формулы путем копирования блока, переноса блока, вставки или удаления строк или столбцов приводят к автоматическому изменению адресов переменных в формулах, находящихся в смещенных ячейках. Иначе говоря, формулы модифицируются в соответствии со своим новым положением.

При отмене режима относительной адресации устанавливается режим абсолютной адресации. В этом слу-

чае при смещении ячеек модификации формул не происходит.

Обычно режим относительной адресации работает в таблице по умолчанию. Абсолютная адресация применяется к отдельным ссылкам на ячейки в формулах. Для этого используется символ «замораживания» адреса — «\$». С помощью этого символа можно «заморозить» как весь адрес, например $\$B\2 , так и отдельные его части, например $\$B2$, $B\$2$.

Рекомендации по организации практической работы

Главная задача учащихся на минимальном уровне изучения данной темы: научиться основным методам организации расчетов с помощью электронных таблиц. Для этого они должны освоить следующие практические приемы работы в среде электронных таблиц:

- перемещение табличного курсора; установка курсора в нужную ячейку;
- ввод данных: чисел, текстов, формул;
- редактирование данных в ячейках;
- копирование информации в ячейках;
- вставка и удаление строк и столбцов.

Теоретические вопросы, которые на первом этапе вызывают наибольшие затруднения — это правила записи формул и понимание принципа относительной адресации. Их отработку следует проводить на задачах и упражнениях. Набор таких заданий имеется в разделе 5.5 задачника-практикума.

Основные *правила записи формул* сводятся к следующему:

- все символы в формуле записываются в одну строку;
- проставляются все знаки операций (в отличие от алгебры, где знак умножения часто пропускается);
- используются круглые скобки для влияния на последовательность выполнения операций;

- учитываются приоритеты операций, расположенные в таком порядке: $\wedge$ — возведение в степень; $*$, $/$ — умножение и деление; $+$, $-$ — сложение и вычитание;
- приоритет стандартных функций выше приоритета арифметических операций; аргумент записывается в круглых скобках после имени функции;
- последовательно записанные операции одинакового старшинства выполняются в порядке записи, то есть слева направо.

Все эти правила совпадают с правилами записи выражений в языках программирования. Поэтому при более позднем изучении программирования этот вопрос ученикам будет уже знаком.

Для упражнений на данную тему следует давать задачи как прямые (дано математическое выражение, записать формулу для электронной таблицы) так и обратные (дана формула, записать математическое выражение).

Например, дано математическое выражение:

$$\frac{40y^3 + \frac{4}{9}x}{6x^2 - 18xy}$$

Нужно записать формулу для электронной таблицы. Во-первых, переменным величинам в этой формуле нужно поставить в соответствие адреса ячеек. Например, так: x — A1, y — B1. Теперь следует записать формулу:

$$(40*B1^3 + 4/9*A1)/(6*A1^2 - 18*A1*B1)$$

Для обоснования правильности формулы следует пронумеровать операции в порядке их выполнения так, как это сделано в примере.

Пример обратной задачи. Дана формула для электронных таблиц:

$$C5*B5/A5^2 + B2^2$$

Нужно записать соответствующее математическое выражение. От адресов ячеек перейдем к математическим переменным: $C5 — x$, $B5 — y$, $A5 — z$, $B2 — k$. Следующим шагом нужно, как это делали раньше, указать последовательность вычислений.

$$\begin{matrix} 3 & 4 & 1 & 5 & 2 \\ C5*B5/A5^2 + B2^2 \end{matrix}$$

Теперь становится ясно, что эта формула соответствует следующему математическому выражению:

$$\frac{xy}{z^2} + k^2$$

Теперь о *принципе относительной адресации*. Как уже отмечалось, это один из базовых принципов функционирования электронной таблицы. Адреса ячеек, используемых в формулах, определяются относительно места расположения формулы. Смысл этого принципа следует пояснить на примерах. В данной ниже таблице формулу в ячейке C1 табличный процессор воспринимает так: сложить значение из ячейки, расположенной на две ячейки левее данной формулы со значением из ячейки, расположенной на одну ячейку левее данной формулы.

	A	B	C
1	5	3	$A1 + B1$

При переносе этой формулы в любую другую ячейку любым способом (копированием, вследствие вставки или удаления фрагментов), сохранится сформулированный выше смысл формулы. Вследствие этого изменятся ссылки на ячейки. Например, при копировании формулы из ячейки C1 в ячейку C2 формула примет вид $A2 + B2$, при копировании в ячейку F6 — $F4 + F5$.

Для закрепления понимания принципа относительной адресации следует выполнить несколько заданий. Условия предлагаемых задач должны быть следующего типа: дан фрагмент электронной таблицы (например такой, как приведен ниже). Какие формулы занесутся в ячейки блока D1:F1, если в них скопировать формулу из ячейки C2? Результат выполнения этого задания в режиме отображения формул и в режиме отображения значений будет следующим:

	A	B	C	D	E	F
1	5	3	A1+B1	B1+C1	C1+D1	D1+E1

	A	B	C	D	E	F
1	5	3	8	11	19	30

Здесь заливкой отмечены ячейки, в которые произведено копирование.

Наиболее эффективной будет такая постановка задания: сначала решить задачу теоретически, а затем проверить полученное решение на компьютере. В этом случае происходит как закрепление понимания теоретического вопроса, так и отработка навыков копирования данных в электронной таблице.

При знакомстве с приемом «замораживания» адресов ячеек в формулах, полезно выполнить аналогичные упражнения. Например, в той же таблице в ячейке C1 записана формула A\$1+\$B\$1. Какой вид примет формула, если ее скопировать в блок D1:F1 и в блок C2:F2? Результаты решения этой задачи будут следующими:

	A	B	C	D	E	F
1	5	3	A\$1+\$B\$1	B\$1+\$B\$1	C\$1+\$B\$1	D\$1+\$B\$1
2			A\$1+\$B\$1	B\$1+\$B\$1	C\$1+\$B\$1	D\$1+\$B\$1

	A	B	C	D	E	F
1	5	3	8	5	11	9
2			8	6	11	9

Основные типы расчетных задач, которые учащиеся должны научиться решать в электронных таблицах:

- 1) получение несложных расчетных ведомостей;
- 2) статистическая обработка числовых таблиц;
- 3) построение диаграмм по табличным данным;
- 4) сортировка таблицы по значениям параметра (столбца);
- 5) табулирование функций.

Примером решения задачи первого типа является построение таблицы учета продажи молочных продуктов, рассмотренное в § 33–35 учебника в качестве основного примера. Другая задача этого же типа, предлагаемая в списке заданий к § 34: построить таблицу расчета ежемесячной оплаты за расход электроэнергии по данным показаниям счетчика и стоимости 1 кВт·ч. Обсудим ход решения этой задачи.

Начать решение задачи следует с проектирования таблицы. Это наиболее сложный этап для учеников. Во-первых, необходимо пояснить алгоритм подсчета оплаты за электричество (не всем детям это очевидно). Показания счетчика снимаются в конце каждого месяца. Расход электроэнергии за месяц определяется как разность показания счетчика в данном месяце и в предыдущем. Затем оплата подсчитывается как произведение расхода на цену 1 кВт·ч. Для расчета платы за январь необходимо знать показания счетчика в декабре прошлого года. Из всего сказанного нужно сделать вывод: какие величины являются исходными данными, какие — вычисляемыми по формулам. Исходные данные — показания счетчика каждый месяц и стоимость 1 кВт·ч, а вычисляемые — ежемесячный расход электроэнергии и сумма оплаты.

Теперь можно переходить к построению таблицы. Следует сразу же сформулировать ученикам некоторые правила оформления таблицы. Во-первых, у таблицы обязательно должен быть заголовок. Строки и столбцы таблицы должны быть поименованы. Постепенно учитель должен познакомить учеников с основными средствами форматирования таблицы для получения оформленного документа. Начальная часть таблицы будет выглядеть так (предполагается, что цена 1 кВт·ч равна 1 руб. 05 коп., а плата подсчитывается в рублях):

	А	В	С	Д
1		Ведомость оплаты электроэнергии		
2	Месяц	Счетчик	Расход в кВт·ч	Оплата в руб.
3	Декабрь	4000		
4	Январь	4200	B4-B3	C4 * 1,05
5	Февраль	4350	B5-B4	C5 * 1,05
6	Март	4475	B6-B5	C6 * 1,05

Во время работы с табличным процессором обязательно нужно использовать прием копирования формул. Формулы в ячейках C4, D4 являются исходными. Все формулы, расположенные ниже их, получены путем копирования. Из этого примера ученикам станет ясен практический смысл принципа относительной адресации: он позволяет быстро создавать большие таблицы без переписывания формул.

Разбор этой задачи приводит к идее использования абсолютного («замороженного») адреса. Удобно цену 1 кВт·ч хранить в отдельной ячейке (например, в B16), а в ячейке D4 записать формулу C4*\$B\$16. При копировании формулы абсолютный адрес меняться не будет. Теперь, если произойдет изменение цены 1 кВт·ч,

то будет достаточно внести изменения лишь в одну ячейку B16.

Очень часто в числовых таблицах подсчитываются различные итоговые данные: суммы, средние значения, наибольшие и наименьшие значения. Получение таких данных называется *статистической обработкой таблицы*. Во всех табличных процессорах имеются для этого соответствующие функции. В задачах того же типа, что рассмотрен выше, следует добавить задание на подобную статистическую обработку данных. Например, к таблице оплаты потребления электроэнергии надо добавить вычисление общей суммы денег, выплаченных за год, среднемесячного расхода электроэнергии, наибольшей и наименьшей месячной платы.

Представление табличных данных в графической форме часто используется на практике. Графическая обработка придает наглядность, обозримость результатам расчетов. Табличные процессоры предоставляют пользователю на выбор множество типов диаграмм (гистограмм, графиков). Такие графические средства принято называть *деловой графикой*.

Для построения диаграммы пользователь должен указать ее тип и сообщить табличному процессору, из каких блоков таблицы нужно выбирать всю необходимую информацию. Достаточно подробно основные типы диаграмм описаны в задачнике-практикуме в разделе 5.5.5, даны рекомендации по их применению.

Большие возможности придает использование в электронной таблице *условной и логических функций*. В учебнике эти средства рассматриваются в § 36—37. Таблица без использования условной функции реализует в себе линейный вычислительный алгоритм. Использование условной функции вносит в таблицу структуру ветвления. Потребность в ветвлении появляется при усложнении условия задачи. Например, если бы правило оплаты за электроэнергию звучало так: за первые 100 кВт·ч нужно платить по 1 руб. 05 коп.; за каждый кВт·ч, израсходованный выше этого, нужно платить по

1 руб. 10 коп., то расчетная формула в ячейке D4 выглядела бы так:

$$IF(C4 \leq 100, C4 * 1.05, 1.05 + (C4 - 100) * 1.1).$$

Возможность *сортировки данных* в таблице существует у табличных процессоров благодаря наличию режима базы данных. Обычно сортировка организуется в выделенном блоке таблицы. Как и в базах данных, указывается столбец, по которому производится сортировка, и порядок сортировки: по возрастанию или по убыванию значений в столбце. Например, в рассмотренной таблице можно отсортировать строки, начиная с четвертой, по убыванию размера оплаты. Тогда вначале расположится самый «дорогой» месяц, далее — по убыванию.

Табулирование функции — одна из часто решаемых прикладных задач математики. Табулирование означает построение таблицы значений функции для значений аргумента, изменяющихся в определенном интервале с данным шагом. Табулирование позволяет исследовать функцию: проследить характер изменения, выделить области корней, определить экстремальные значения. Применяя прием копирования, в электронной таблице можно быстро построить таблицу значений функции большого размера. Для этого достаточно ввести две начальные строки таблицы и затем скопировать их вниз на нужное число строк. Например, таблица функции

$$F(x) = x^3 + 0,5\sqrt{x},$$

вычисленной с шагом 0,2 начиная от значения $x = 0$, строится следующим образом:

	A	B
1	Шаг =	0,2
2	x	F(x)
3	0	A3^3+SQRT(A3)*0.5

4	A3+\$B\$1	A4^3+SQRT(A4)*0.5
5	A4+\$B\$1	A5^3+SQRT(A5)*0.5
6	A5+\$B\$1	A6^3+SQRT(A6)*0.5

Если нужно изменить начальное значение x в таблице, то достаточно исправить ячейку A3, а для изменения шага нужно отредактировать содержимое ячейки B1. Формулы записываются по правилам, принятым в используемом табличном процессоре.

Применение электронной таблицы в качестве инструмента для математического моделирования рассматривается в углубленном варианте изучения базового курса.

Что же такое *математическая модель*? Это описание состояния или поведения некоторой реальной системы (объекта, процесса) на языке математики, то есть с помощью формул, уравнений и др. математических соотношений. Характерная конфигурация всякой математической модели показана на рис. 14.4.

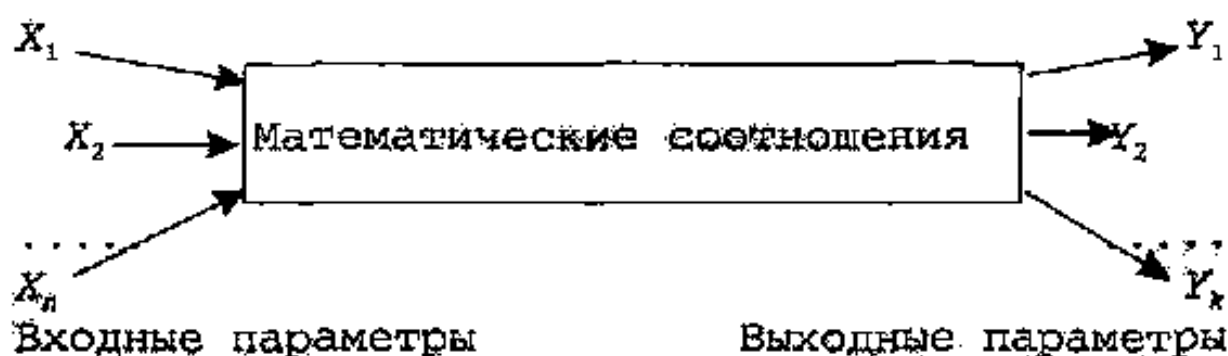


Рис. 14.4. Обобщенная структура математической модели

Здесь X и Y — некоторые количественные характеристики моделируемой системы.

Реализация математической модели — это применение определенного метода расчетов значений выходных параметров по значениям входных параметров. Технология электронных таблиц — один из возможных методов реализации математической модели. Дру-

гими методами реализации математической модели может быть составление программ на языках программирования, применение математических пакетов (MathCAD, «Математика» и др.), специализированных программных систем для моделирования. Реализованные такими средствами математические модели будем называть компьютерными математическими моделями.

Цель создания компьютерной математической модели — проведение численного эксперимента, позволяющего исследовать моделируемую систему, спрогнозировать ее поведение, подобрать оптимальные параметры и пр.

Итак, характерные признаки компьютерной математической модели следующие:

- наличие реального объекта моделирования;
- наличие количественных характеристик: входных и выходных параметров;
- наличие математической связи между входными и выходными параметрами;
- реализация модели с помощью определенных компьютерных средств.

В дополнении к главе 8 части II учебника приведен пример реализации в электронных таблицах математической модели эволюционного типа: исследуется изменение со временем числа рыб в пруду исходя из закона Мальтуса.

Серия заданий для индивидуальных работ, приведенная в разделе 5.5 задачника-практикума, содержит материал для решения всех упомянутых выше пяти типов задач. Использование электронных таблиц в математическом моделировании отражено в разделе 6 задачника-практикума «Компьютерное математическое моделирование».

15. Информация и управление

Разделы учебника: часть I, § 38—44. **Дополнительный материал:** часть II, раздел 9.1.

Основные цели. Познакомить учащихся с основами кибернетики, кибернетической моделью процессов управления. Дать им представление о применении ЭВМ для автоматизации процессов управления. Ввести понятие алгоритма управления, определить свойства алгоритма. Познакомить учащихся со способами описания алгоритмов; обучить их составлению алгоритмов для управления учебным исполнителем.

Изучаемые вопросы

- ☐ Что такое кибернетика.
- ☐ Кибернетическая модель управления.
- ☐ Алгоритм управления.
- ☐ Автоматизированные системы управления.
- ☐ Системы автоматического управления.
- ☐ Свойства алгоритма.
- ☐ Учебный исполнитель.
- ☐ Алгоритмы управления учебным исполнителем.

Методические рекомендации по изложению теоретического материала

Традиционной темой для любой версии школьного курса информатики является алгоритмизация. В некоторых учебниках эта тема составляет наибольшую часть содержания. Чаще всего алгоритмизация связывается с программированием для ЭВМ на каком либо языке.

В базовом курсе применен иной подход к теме алгоритмизации. Его можно назвать кибернетическим подходом. Алгоритм рассматривается как информационный компонент системы управления. Такой подход дает возможность ввести в содержание базового курса новую содержательную линию — линию управления. Это многоплановая линия, которая позволяет затронуть следующие вопросы:

- элементы теоретической кибернетики: кибернетическая модель управления с обратной связью;
- элементы прикладной кибернетики: структура компьютерных систем автоматического управления (систем с программным управлением); назначение автоматизированных систем управления;
- основы теории алгоритмов.

О теоретической кибернетике. Кибернетика — слово греческого происхождения. В буквальном переводе оно обозначает «искусство управления». Впервые этот термин встречается в научных трудах древнегреческого философа Платона (IV век до н.э.). В XIX веке А. Ампер предложил назвать кибернетикой науку об управлении человеческим обществом. В XX веке вновь этот термин использует Н. Винер (1948), назвав им науку об управлении и связи в живом организме и машине.

В разделе 1.4 части II учебника рассказывается о сущности кибернетического подхода к решению задачи управления сложными системами. Она сводится к так называемой модели черного ящика: по отношению к исследуемой системе определяются лишь входные и выходные сигналы, описывается взаимосвязь между ними. Входные и выходные сигналы, независимо от их физической природы, интерпретируются как информация. Поэтому управление системой рассматривается как информационное взаимодействие с ней некоторого управляющего объекта.

Основным открытием кибернетической науки является принцип универсальности *схемы управления с обратной связью*. Эта модель управления распространяется на технические устройства, биологические и социальные системы. Наряду с общей теорией управления, разработанной кибернетикой, существуют многочисленные прикладные ветви этой науки: техническая кибернетика, экономическая кибернетика, биологическая кибернетика, медицинская кибернетика, социальная кибернетика и др.

Рассказывая о кибернетической модели управления, учитель должен проиллюстрировать ее примерами, знакомыми и понятными ученикам. В любом процессе управления присутствуют следующие составляющие:

- управляющий объект;
- управляемый объект;
- канал прямой связи для передачи управляющего воздействия (команд управления);
- канал обратной связи для передачи информации о состоянии объекта управления, окружающей среды.

Следует начать с очевидных примеров, таких, как шофер и автомобиль. Шофер — управляющий, автомобиль — управляемый объект. Канал прямой связи — система управления автомобилем: педали, руль, рычаги, клавиши и пр. Каналы обратной связи: приборы на панели управления, вид из окон, слух шофера. Всякое воздействие на средства управления можно рассматривать как передаваемую информацию: «увеличить скорость», «затормозить», «повернуть направо» и так далее. Информация, передаваемая по каналам обратной связи, также является необходимой для успешного управления. Предложите ученикам задание: что произойдет, если отключить один из каналов прямой или обратной связи? Обсуждение таких ситуаций обычно бывает очень оживленным.

Управление с обратной связью называют *адаптивным управлением*. Действия управляющего адаптируются (то есть подстраиваются) к состоянию объекта управления, окружающей среды.

Затем следует обсудить более сложные примеры, например, полет радиоуправляемой модели самолета; дистанционное управление полетом космического корабля и т. п. Во всех случаях нужно выделить четыре отмеченных выше элемента процесса управления.

После технических примеров перейдите к рассмотрению различных ситуаций управления в биологических,

общественных системах. Самый близкий ученикам пример: учитель, управляющий процессом обучения на уроке. Обсудите различные формы управляющего воздействия учителя на учеников: речь, жесты, мимика, записи на доске. Предложите ученикам перечислить различные формы обратной связи; объяснить, как адаптирует учитель ход урока по результатам обратной связи, привести примеры такой адаптации. Например: ученики не справились с предложенным заданием — учитель вынужден повторить объяснение.

Далее можно перейти к обсуждению управления в более крупных социальных системах: управление предприятием его администрацией, страны правительством и т. п. Анализируя механизмы прямой и обратной связи, обратите внимание учеников на тот факт, что в большинстве случаев существует множество каналов прямой и обратной связи. Часто они дублируются, что повышает надежность работы системы управления.

Следующий вопрос для обсуждения — понятие алгоритма управления. Управление — это целенаправленный процесс. Он должен обеспечить определенное поведение объекта управления, достижение определенной цели. А для этого должен существовать план управления. Этот план реализуется через последовательность управляющих команд, передаваемых по прямой связи. Такая последовательность команд и называется алгоритмом управления.

Под системой управления будем понимать всю совокупность управляющих средств: управляющий объект, а также каналы прямой и обратной связи. *Алгоритм управления является информационной компонентой системы управления.* Например, учитель ведет урок согласно заранее составленному плану. Шофер ведет автомобиль по заранее продуманному маршруту. В системах управления, где роль управляющего выполняет человек, алгоритм управления может изменяться, уточняться в процессе работы. Шофер не мо-

жет спланировать заранее каждое свое действие во время движения; учитель корректирует план урока по его ходу. Если же процессом управляет автоматическое устройство, то детальный алгоритм управления должен быть в него заложен заранее в некотором формализованном виде. В таком случае его называют *программой управления*. Для хранения программы автоматическое устройство управления должно обладать *программной памятью*.

Наконец, существует понятие *самоуправляемой системы*. Это некоторый единый объект, организм, в котором присутствуют все отмеченные выше компоненты систем управления: управляющие и управляемые части (органы), прямая и обратная информационная связь, управляющая информация — алгоритмы, программы и память для ее хранения. Примерами таких систем являются живые организмы. Наиболее совершенный из них — человек. Человек управляет сам собой. Основным управляющим органом является мозг человека, управляемыми — все части организма. Есть управление осознанное (я делаю, что хочу) и есть подсознательное (управление физиологическими процессами). Подобные процессы происходят и у животных. Однако доля осознанного управления у животных меньше, чем у человека в силу более высокого уровня интеллектуального развития человека.

Создание искусственных самоуправляемых систем — одна из сложнейших задач науки и техники. Робототехника — пример такого научно-технического направления. В нем объединяются многие области науки: кибернетика, искусственный интеллект, медицина, математическое моделирование и пр.

О прикладной кибернетике. К задачам *технической кибернетики* относится разработка и создание технических систем управления. Современные системы этого типа основаны на использовании ЭВМ, микропроцессорной техники.

В § 39 учебника приводится характерная схема системы автоматического управления (САУ) на базе компьютера. Это замкнутая техническая система, которая работает без участия человека. Человек (программист) подготовил программу управления, занес ее в память компьютера. Дальше система работает автоматически.

Рассматривая этот вопрос, следует обратить внимание учеников на то, что с преобразованием информации из аналоговой формы в цифровую и обратно мы уже встречались в предыдущих темах. По такому же принципу работает модем в компьютерных сетях, звуковая карта при вводе/выводе звука.

Сущность аналого-цифрового преобразования в дискретизации непрерывно меняющейся величины (например, силы тока), которая называется аналоговым сигналом. Значения этой величины измеряются прибором с определенным шагом по времени и результаты этих измерений сохраняются в памяти компьютера в виде числовой таблицы. Этот процесс отображен на рис. 15.1. Здесь F — измеряемая величина. В память компьютера заносится таблица из последовательности чисел: $F_1, F_2, F_3, F_4, \dots$. Очевидно, чем выше частота измерений (меньше шаг по времени), тем точнее снимается зависимость.

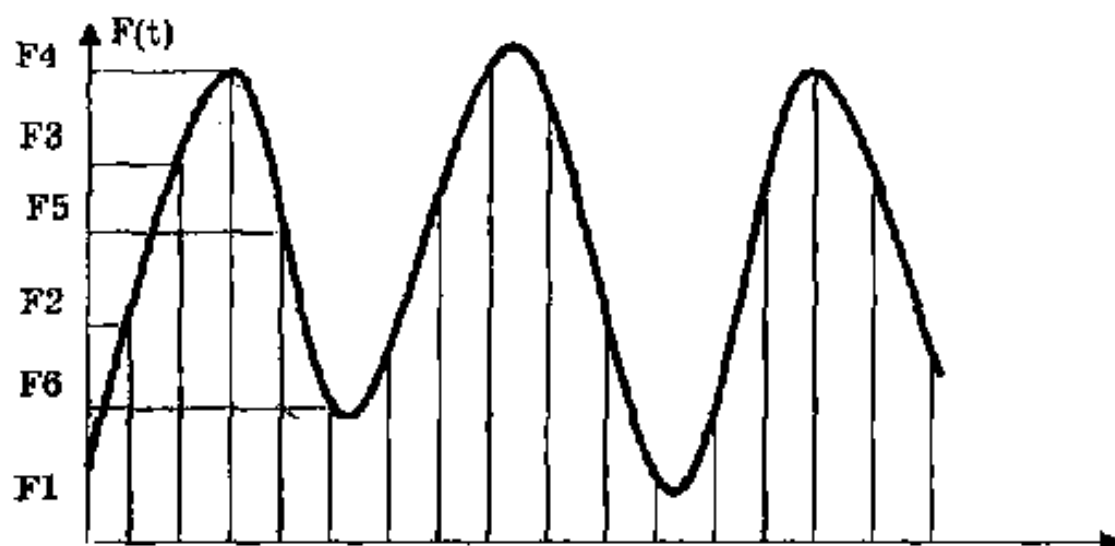


Рис. 15.1. Процесс аналого-цифрового преобразования

В § 39 обсуждается и другой вариант систем управления с применением ЭВМ — автоматизированные системы управления (АСУ). Можно сказать, что АСУ — это человеко-машинная система. Как правило, АСУ ориентированы на управление деятельностью производственных коллективов, предприятий. Это системы компьютерного сбора, хранения, обработки разнообразной информации, необходимой для работы предприятия. Например, это данные о финансовых потоках, наличии сырья, объемах готовой продукции, кадровая информация и так далее. Основная цель таких систем — быстро и точно предоставлять руководителям предприятия необходимую информацию для принятия управляющих решений.

Задачи, решаемы средствами АСУ, относятся к области *экономической кибернетики*. Как правило, технической базой таких систем являются локальные компьютерные сети. В АСУ используются разнообразные информационные технологии: базы данных, машинная графика, компьютерное моделирование, экспертные системы и пр.

Об алгоритмах управления

Традиционная методическая последовательность, используемая в преподавании алгоритмизации:

1. Определение алгоритма.
2. Свойства алгоритма.
3. Примеры исполнителей алгоритмов.
4. Программирование работы исполнителя.

Определение и свойства алгоритма. В § 40 учебника дано следующее определение алгоритма: «Алгоритм — понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящих от исходных данных к искомому результату.»

В этом определении содержатся основные понятия, связанные с алгоритмом и его главные свойства. Взаимосвязь понятий отражена на рис. 15.2.

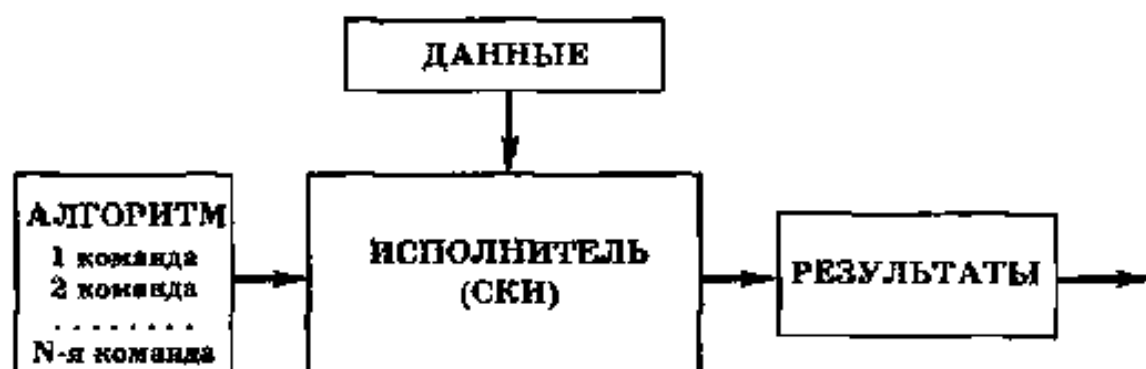


Рис. 15.2

Схема функционирования исполнителя алгоритмов

Центральным объектом в этой системе является *исполнитель* алгоритмов. Исполнитель — это тот объект (или субъект), для управления которым составляется алгоритм. Выше мы его называли объектом управления. Основной характеристикой исполнителя, с точки зрения управления, является его система команд — СКИ. Это конечное множество команд, которые исполнитель понимает, то есть умеет выполнять.

Для выполнения всякой работы, решения поставленной задачи исполнитель на входе получает алгоритм и исходные данные, а на выходе выдает требуемые результаты. *Алгоритм может включать в себя только команды, входящие в СКИ.* Это требование к алгоритму называется *свойством понятности*.

Другое свойство алгоритма — *точность*. Всякая команда должна быть сформулирована так, чтобы определить однозначное действие исполнителя. Например, кулинарный рецепт можно рассматривать как алгоритм для исполнителя-повара по приготовлению блюда. Но если одним из пунктов в нем будет написано: «Положить несколько ложек сахара», то это пример неточной команды. Сколько ложек? Каких ложек (чайных, столовых)? Каждый повар может это понимать по-своему, и результаты будут разными. Пример точной команды: «Положить 2 столовые ложки сахара».

Работа исполнителя состоит в последовательном формальном выполнении команд алгоритма. От

сюда следует вывод о возможности создания автоматических исполнителей. В частности, таким автоматическим исполнителем алгоритмов по обработке информации является компьютер.

Еще одно свойство, которое отражено в определении алгоритма — *конечность*. Оно формулируется так: исполнение алгоритма и, следовательно, получение искомого результата должно завершиться за конечное число шагов. Здесь под шагом подразумевается выполнение отдельной команды. Это свойство является предупреждением от ситуации, которую программисты называют заикливанием. Бесконечно исполняемый алгоритм безрезультатен. Поэтому свойство конечности называют еще *результативностью* алгоритма.

Методические рекомендации по решению задач

Для закрепления основных понятий, связанных с определением алгоритма, полезно выполнить несколько заданий следующего содержания:

- 1) выполнить роль исполнителя: дан алгоритм, формально исполнить его;
- 2) определить исполнителя и систему команд для данного вида работы;
- 3) в рамках данной системы команд построить алгоритм;
- 4) определить необходимый набор исходных данных для решения задачи.

В качестве примера задачи первого типа можно использовать алгоритм игры Баше, приведенный в § 40. После того, как ученики поиграют в эту игру по тем правилам, что описаны в учебнике, можно предложить им несколько заданий аналитического характера на тему игры Баше. Задания могут быть предложены в качестве домашней работы.

Задача 1

«Разгадать загадку» алгоритма, то есть объяснить, почему второй игрок всегда выигрывает.

По данным правилам второй игрок будет всегда выигрывать, если общее число камней определяется формулой $N = 5k + 1$. Здесь k — любое натуральное число.

Задача 2

Составить алгоритм, по которому игрок, делающий первый ход, может выиграть в том случае, если соперник не знает выигрышной тактики.

Необходимо перехватить инициативу, то есть оказаться в положении второго игрока, который дополняет предыдущий ход соперника до 5 камней. Это возможно лишь в случае ошибки соперника. Начать игру можно так:

1. Взять 1 камень.
2. Предоставить ход сопернику; соперник взял n камней.
3. Если $n + 1 < 5$, то взять $5 - (n + 1)$ камней.
4. Предоставить ход сопернику.

И далее играть по выигрышному алгоритму для второго игрока.

Следующая задача требует от учеников незаурядных математических навыков.

Задача 3

Попробуйте провести математический анализ игры. Билия Чобизем, играя для N камней. Определите правила игры (то есть сколько камней можно брать за один ход), при которых имеется выигрышный алгоритм. Опишите этот алгоритм в виде последовательности команд.

Выигрышный алгоритм для второго игрока можно построить только в тех случаях, когда исходное число камней (n) представимо в виде: $N = x \times k + 1$, где x и k — натуральные числа. По правилам игры за один ход можно брать от 1 до $x - 1$ камней. Второй игрок будет всегда выигрывать, если своим ходом он будет дополнять число камней, взятых соперником, до X . Напри-

мер, пусть $N = 25$. Это значение можно представить так: $25 = 4 \times 6 + 1$. Следовательно, правило игры должно быть таким: за один ход можно брать от 1 до 3 камней. А для того, чтобы второй игрок всегда выигрывал, на своем ходе он должен дополнять ход соперника до 4 камней.

Задачи типа 2–4 имеются в списке заданий к § 40 и в разделе 4.1 задачника-практикума.

Задача 4

Назвать исполнителя следующего вида работы — выдача заработной платы; определить СКИ исполнителя.

Очевидно, исполнителя можно назвать «Кассир». Система команд, которые он должен уметь выполнять, следующая:

- ☐ найти в ведомости запись о получателе;
- ☐ посчитать деньги;
- ☐ выдать деньги.

В задачах такого типа нужно учить учеников разбивать работу исполнителя на сравнительно простые действия, которые требуют формального исполнения. Команда «выдать зарплату» не удовлетворяет таким требованиям.

При построении СКИ решаются две проблемы: проблема элементарности команд и проблема полноты системы команд. Система команд исполнителя называется полной, если она содержит весь минимально необходимый набор команд, позволяющий построить любой алгоритм в том классе задач, на который ориентирован исполнитель.

Рассмотрим еще один пример (задача № 4 из раздела 4.1 задачника-практикума).

Задача 5

Описать систему команд исполнителя «Геометр», который мог бы выполнять геометрические построения с помощью циркуля и линейки.

Ученикам знаком класс задач, которые в геометрии называются задачами на построение с помощью линейки, циркуля и карандаша. Полной системой команд для исполнителя «Геометр» является следующий список:

1. Провести отрезок прямой между двумя данными точками.
2. Установить раствор циркуля, равный длине данного отрезка.
3. Установить ножку циркуля в данную точку.
4. Провести окружность.
5. Выделить общие точки двух линий (пересечения или касания).

Обратите внимание на элементарность каждой команды. Делить их на более простые не имеет смысла.

Задача 6

Записать алгоритм для Геометра решения следующей задачи: дан отрезок AB ; построить окружность, для которой отрезок AB является диаметром.

Алгоритм ОКРУЖНОСТЬ ДАННОГО ДИАМЕТРА

начало

установить ножку циркуля в т. A
установить раствор циркуля, равный AB
провести окружность
установить ножку циркуля в т. B
провести окружность
выделить точки пересечения
окружностей: т. C и т. D
провести отрезок CD
выделить точку пересечения
 AB и CD : т. O
установить ножку циркуля в т. O
установить раствор циркуля, равный OB
провести окружность

конец

Анализируя этот пример, следует подчеркнуть то, что данный алгоритм удовлетворяет всем основным свойствам: понятности, точности, конечности, благодаря чему может исполняться формально.

Задания 4-го типа относятся к проблеме постановки задач на построение алгоритмов. Для выполнения требуемой работы, решения данной задачи необходим не только алгоритм, но и полный набор исходных данных, что отражено на рис. 15.2. Это могут быть какие-то материальные объекты (например, детали для сборки устройства; продукты для приготовления блюда и пр.) или информация (числовые данные для расчетов). Вот некоторые задачи на определение полного набора данных.

1. Определить полный набор данных для вычисления стоимости покупок в магазине. *Ответ:* количество каждого вида купленных товаров и цены единиц товаров (1 кг или 1 штуки).

2. Определить полный набор данных для вычисления месячной платы за расход электроэнергии. *Ответ:* показания счетчика в конце предыдущего и в конце настоящего месяца, стоимость 1 кВт · ч.

3. Определить полный набор данных для вычисления времени падения кирпича с крыши дома. *Ответ:* высота дома, ускорение свободного падения (без учета сопротивления воздуха).

Обучение методике алгоритмизации

Обучение методам построения алгоритмов — один из наиболее отработанных разделов школьной информатики. Традиционно применяемым дидактическим средством в этом разделе являются учебные исполнители алгоритмов.

Большие возможности при изучении алгоритмизации предоставляют исполнители, работающие по принципу «черепашьей графики». Идея черепашьей графики происходит из учебного языка программирования

ЛОГО, созданного в конце 60-х годов американским психологом С. Пейпертом для работы с младшими школьниками [26]. В ЛОГО включен графический исполнитель «Черепашка», которая, выполняя команды типа «вперед», «назад», «поворот» и пр., якобы ползая по песку, рисует на нем своим хвостом.

Конечно, учитель может взять непосредственно ЛОГО и обучать программированию на этом языке. Методики обучения информатике на базе ЛОГО известны. В нашем курсе мы предлагаем использовать другие исполнители, работающие по аналогичному принципу. Их главное достоинство — русский язык общения. Форма записи алгоритмов для таких исполнителей совпадает с формой записи на учебном алгоритмическом языке и, таким образом, соблюдается единая форма описания алгоритмов во всем курсе.

В пакете учебных программ для курса А. Г. Гейна и др. имеется графический исполнитель «Чертежник». Аналогичный исполнитель, который называется «Кенгуренок», реализован фирмой КУДИЦ для IBM PC.

В этом разделе курса учитель может использовать и других исполнителей. Например, это может быть Робот из пакета КуМИР, Кукарана из Роботландии, Муравей Г. Н. Гутмана и др., хотя они не являются аналогом черепашьей графики. Вообще говоря, подходит любой исполнитель, который удовлетворяет следующим условиям:

- это должен быть исполнитель, работающий в некоторой обстановке;
- этот исполнитель должен имитировать процесс управления некоторым реальным объектом (черепашкой, роботом и др.);
- в системе команд исполнителя должны быть все структурные команды управления (ветвления, циклы);
- исполнитель должен позволять использовать вспомогательные алгоритмы (процедуры).

Последние два пункта означают, что на данном исполнителе можно обучать структурной методике алгоритмизации. Всякое педагогическое средство должно соответствовать поставленной учебной цели. *Главной целью раздела алгоритмизации является овладение учащимися структурной методикой построения алгоритмов.*

Для того чтобы соблюсти принцип инвариантности содержания по отношению к конкретным версиям программного обеспечения, в учебнике описывается гипотетический учебный исполнитель, которому дано имя ГРИС — графический исполнитель. Это исполнитель, работающий в обстановке (то есть без использования величин). Наиболее близкими к нему являются Кенгуренок и Чертежник (к учебнику А. Г. Гейна). На примере ГРИС вводятся основные понятия алгоритмизации. Предлагаемая последовательность заданий способствует эффективному достижению основной цели раздела — освоения структурной методики построения алгоритмов.

Однако на уроке ученикам предстоит работать с конкретным учебным исполнителем и учителю придется адаптировать материал учебника к данному педагогическому средству. Каким бы исполнителем ни пользовался учитель, рекомендуем следовать методической схеме, излагаемой ниже. Изложения будут частями на примере исполнителя «Кенгуренок». Авторы программы «Кенгуренок» назвали своего героя Ру (Roo) — по имени персонажа мультфильма. Мы также будем пользоваться этим именем.

Будем рассматривать данное учебное программное средство, как имитатора автоматической системы управления. Как отмечалось выше, в процессе управления участвуют три «персонажа»: человек, система управления (будем в дальнейшем говорить короче — система) и сам объект управления (образ кенгуренка Ру на экране). В методической литературе под словами «учебный

исполнитель» понимается вся совокупность программ, реализующих систему управления и имитацию на экране управляемого объекта.

В разделе «Информационные технологии» мы трактовали средства компьютерных информационных технологий (редакторы, СУБД и пр.) как исполнителей определенного вида работы с информацией. При их описании выделялись следующие составляющие: *среда, режимы работы, система команд, данные*. По этой же схеме будем описывать и алгоритмических учебных исполнителей.

Среда исполнителя. На экране присутствуют три основных элемента среды учебного исполнителя: строка меню, поле программы и поле рисунка, на котором находится Кенгуренок. На поле рисунка неявно (то есть ее не видно) нанесена прямоугольная сетка. Длину стороны одной квадратной ячейки этой сетки назовем шагом. Размер всего поля — 15 шагов по горизонтали и 19 шагов по вертикали.

Режимы работы. Это режимы, которые предоставляет пользователю (человеку) система управления. Для исполнителя «Кенгуренок» вся система режимов работы отражена в схеме на рис. 15.3.

Кенгуренок может работать в режиме прямого управления: «команда—исполнение» (в схеме это названо ручным управлением) и в режиме программного управления: «программирование—автоматическое исполнение программы». Программный режим устанавливается тогда, когда текстовый курсор находится на поле программы. Если его вывести за границу поля программы, то установится режим прямого управления.

Реализация этих двух режимов очень полезна с методической точки зрения, в контексте темы управления. Работа Кенгуренка в режиме прямого управления имитирует ситуацию «ручного» управления объектом со стороны человека. Человек отдает только простые коман-



Рис. 15.3.

Система режимов работы исполнителя «Кенгуренок»

ды (*шаг, поворот, прыжок*), а управляющие решения принимает сам, анализируя обстановку на поле исполнителя.

Режим программного управления следует трактовать как имитацию ситуации, когда объектом управляет компьютер. Роль человека — составление алгоритма, ввод программы и инициализация ее исполнения компьютером.

В свою очередь, в режиме программного управления имеются следующие режимы (подрежимы) работы:

- *установка исходного состояния*: стирается рисунок с поля Кенгуренка, устанавливается исходное положение и направление Кенгуренка (этот режим работает и при «ручном» управлении);
- *программирование*: набор программы на программном поле;
- *исполнение*: работа Ру по заданной программе.

И, наконец, исполнение может проходить в трех режимах:

- *в автоматическом режиме* (на экране сразу появляется результат выполнения программы);
- *в автоматическом пошаговом режиме* (Кенгуренок демонстрирует выполнение каждой команды);
- *в отладочном режиме* (исполнение каждой команды запускается отдельно и подробно комментируется мультипликационными средствами).

Режим работы с файлами. В этом режиме можно сохранить программу в файле, прочитать программу из файла в поле программы, сохранить рисунок в файле, распечатать на принтере рисунок и программу.

Режим справки. Данный режим позволяет получить справку о работе системы и о способах редактирования программы.

Система команд исполнителя (СКИ). Команды делятся на команды установки (изменения) режимов и команды управления Кенгуренком (команды программы). Все команды могут быть выбраны из меню. Меню команд первого типа имеет желтый цвет; меню команд второго типа — белый цвет. Будем называть их кратко: «желтое меню» и «белое меню». Переход от одного меню к другому производится нажатием клавиши {Tab}.

Каждую команду можно вызвать нажатием определенной функциональной клавиши. «Желтое меню» является многоуровневым. Команды первого уровня «желтого меню» и соответствующие им функциональные клавиши следующие:

- | | |
|--------------------|---|
| {F1} — [Пуск] | — запуск на исполнение готовой программы в пошаговом автоматическом режиме; |
| {F2} — [Отладка] | — выполнение программы в отладочном режиме с остановкой после каждой команды; |
| {F3} — [Установка] | — очистка поля и установка положения Ру с помощью клавиш перемещения курсора; |
| {F4} — [Разное] | — содержит подменю с дополнительными командами работы с файлами; |
| {F5} — [Результат] | — мгновенное получение результата работы программы (автоматический режим исполнения). |

В свою очередь, команда [Разное] имеет два уровня подменю, состоящих из команд файлового типа. Это дерево меню можно изобразить, как показано на рис. 15.4.

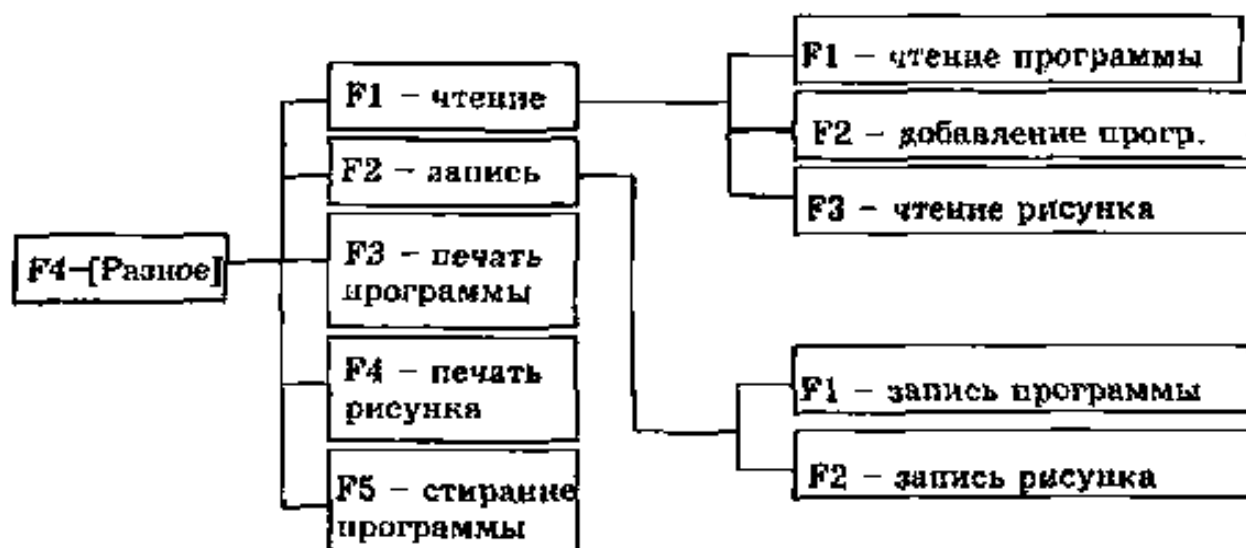


Рис. 15.4.

Система файловых команд исполнителя «Кенгуренок»

В режим справки можно попасть, нажав одновременно клавиши {Ctrl}+{H}.

Систему команд управления Кенгуренком следует сообщать ученикам не всю сразу, а постепенно, по мере необходимости для решения задач.

В режиме ручного управления (курсор за программным полем) используются всего три простые команды «белого меню»:

- 1) шаг — перемещение Ру на один шаг вперед с рисованием линии;
- 2) поворот — поворот Ру на 90° против часовой стрелки;
- 3) прыжок — перемещение Ру на один шаг вперед без рисования линии.

При переходе в режим программирования «белое меню» меняется. К нему, кроме перечисленных, добавляются следующие команды:

- 4) пока <условие>, повторять
 <тело цикла>
 конец цикла — цикл с предусловием;
- 5) если <условие>
 то <серия 1>
 иначе <серия2>
 конец ветвления — полное ветвление;
- 6) если <условие>
 то <серия >
 конец ветвления — неполное ветвление;
- 7) сделай <имя процедуры> — обращение к процедуре.

Данными при работе с этим исполнителем является обстановка на поле Кенгуренка. Она определяется имеющимся рисунком и состоянием самого Кенгуренка: его позицией на поле и направлением (куда смотрит). Обычно исходной обстановкой является отсутствие рисунка (чистое поле) и определенное состояние Ру. Такая обстановка устанавливается в режиме [Установка]. Результатом работы является рисунок.

Приведенное выше описание можно назвать *описанием архитектуры исполнителя «Кенгуренок»*. По такой же схеме рекомендуется описывать архитектуру любого другого алгоритмического исполнителя.

Работа с Кенгуренком. Первыми заданиями на управление Кенгуренком должны быть задачи на работу в режиме прямого управления с целью получить определенный рисунок. Этими рисунками могут быть фигуры, узоры, буквы, построенные из вертикальных и горизонтальных отрезков. Следует обратить внимание учеников на необходимость первоначальной установки Ру на поле.

Обратите внимание учеников на реакцию Ру в случае, если его пытаются переместить за границу поля. С Кенгуренком происходит «авария». Следовательно, выход за край поля допускать нельзя.

Наибольший интерес представляет работа в программном режиме. Для перехода в этот режим доста-

точно переместить курсор в поле программы. После этого сразу увеличится «белое меню».

Обучение программированию для исполнителя нужно строить на последовательности решаемых задач. Эта последовательность должна определяться следующими принципами:

- *от простого к сложному*: постепенное усложнение задач;
- *новизна*: каждая задача вносит какой-то новый элемент знаний (новая команда, новый прием программирования);
- *наследование*: следующая задача требует использования знаний, полученных при решении предыдущих задач.

Порядок рассматриваемых в учебнике задач позволяет ученикам осваивать приемы алгоритмизации в такой последовательности:

1. Составление линейных алгоритмов.
2. Описание и использование вспомогательных алгоритмов.
3. Составление циклических алгоритмов.
4. Использование ветвлений в алгоритмах.
5. Использование метода последовательной детализации при составлении сложных алгоритмов.

Задача 1

Составить алгоритм рисования буквы «Т» в центре поля рисунка. Длина горизонтального и вертикального отрезков — 4 шага. Кенгуренок находится в крайней левой точке горизонтального отрезка и смотрит на восток (направо).

При разборе этой задачи необходимо обратить внимание учеников на два обстоятельства. Первое: управление Кенгуренком для достижения поставленной цели будет происходить без обратной связи. Считаем, что рисование производится вдали от края и выход на границу поля произойти не может. В этом случае алгоритм управления будет иметь линейную структуру.

Второе: алгоритм зависит не только от сформулированной цели (искомого результата), но и от исходного состояния исполнителя. Если бы исходное состояние было другим (например, Ру находится в нижней точке вертикального отрезка и направлен на север), то был бы другим и алгоритм, несмотря на то, что в результате получается один и тот же рисунок. Для алгоритмов работы в обстановке начальное состояние исполнителя является исходным данным задачи. Состояние Кенгуренка определяется местом его расположения на поле и ориентацией. Результатом же выполнения алгоритма становится не только рисунок (главная цель), но и конечное состояние исполнителя.

Задача 2

Составить алгоритм рисования числа 1919.

Эта задача естественным образом выводит на идею вспомогательного алгоритма.

Учитель может поступить следующим образом: предложить ученикам написать алгоритм прежними средствами, то есть используя только три основные команды: шаг, поворот, прыжок. Такое задание, очевидно, не вызовет энтузиазма учеников, поскольку принцип им уже понятен, а писать длинный линейный алгоритм довольно скучно. В этой ситуации вполне возможно самостоятельное «открытие» учениками идеи вспомогательного алгоритма. Обратив внимание на то, что в рисунке дважды присутствуют цифры «1» и «9», ученики могут прийти к идее отдельного описания алгоритмов рисования этих цифр, а затем использования их для получения четырехзначного числа «1919». После обсуждения этой идеи учитель вводит понятие вспомогательного алгоритма и объясняет, как производится его описание и использование. На языке Кенгуренка вспомогательный алгоритм называется процедурой.

Умение использовать вспомогательные алгоритмы необходимо вырабатывать у учеников как можно раньше, уже на примерах линейных алгоритмов. Важней-

ний прием алгоритмизации и программирования — декомпозиция задачи, то есть выделение в исходной задаче некоторых более простых подзадач. Алгоритмы решения таких подзадач называются вспомогательными алгоритмами, а реализующие их программы — подпрограммами (процедурами). Таким образом, решение исходной задачи разбивается на несколько алгоритмов: основной алгоритм и вспомогательные алгоритмы. Как правило, в основном алгоритме происходит многократное обращение к вспомогательному алгоритму.

В § 42 рассказывается о двух методах программирования с использованием подпрограмм: методе последовательной детализации, или методе программирования «сверху вниз», и сборочном методе, или методе «снизу вверх». При построении сложных алгоритмов наиболее эффективным является метод последовательной детализации. В этом случае программирование начинается с составления основного алгоритма. В нем записываются обращения к вспомогательным алгоритмам, которые пока еще не составлены. Но уже на этом этапе необходимо решить вопрос о спецификациях подпрограмм, то есть об их входах и выходах. Для графического исполнителя спецификации заключаются в ответах на вопросы: какой рисунок создает подпрограмма, каким должно быть исходное состояние исполнителя, каким будет конечное состояние исполнителя.

Задача 3

Составить алгоритм рисования горизонтальной линии, проведенной от одного края поля до другого.

Эта задача вносит в данную тему следующие новые элементы: управление с обратной связью; структурная команда цикла. Обратная связь между объектом управления (Кенгуренком) и управляющей системой заключается в том, что перед выполнением каждого шага проверяется условие «впереди не край?». Если оно истинно, то есть ответ положительный, то делается шаг, в противном случае выполнение цикла прекращается.

Механизм обратной связи наглядно иллюстрируется в отладочном режиме исполнения программы. В этом режиме анимационными средствами на экране изображается мальчик Кристоффер, который отдает команды управления Кенгуренку. В начале каждого цикла Кристоффер спрашивает РУ: «Впереди не край?». И получает в ответ: «да» или «нет». Дальнейшие действия зависят от ответа.

Команда цикла является *структурной командой*, в отличие от *простых команд* «шаг», «поворот», «прыжок». Структурная команда включает в себя несколько действий: проверка условия, выполнение тела цикла, которое, в свою очередь, может состоять из нескольких команд.

В теории алгоритмов существуют два типа циклов: цикл с предусловием и цикл с постусловием. Цикл с предусловием (другое его название — «цикл-пока») является основным видом циклической структуры и достаточен для построения любого циклического алгоритма. В языке исполнителя ГРИС — «Кенгуренок» имеется только команда «цикл-пока».

Задача 4

Построить прямоугольную рамку по краю поля.

Решение этой задачи требует объединения умений, полученных учениками при решении предыдущих задач. Циклическая программа рисования линии оформляется в виде процедуры ЛИНИЯ. А в основной программе происходит четырехкратное обращение к этой процедуре.

Задача 5

Расчертить экран горизонтальными линиями.

Эта задача является прямым продолжением предыдущей задачи. Усложнение заключается в том, что используются две процедуры: ЛИНИЯ и ВОЗВРАТ. Кроме того, основной алгоритм сам становится циклическим. Здесь снова нужно отметить то важное обстоятельство,

что при использовании процедур в основной программе необходимо учитывать начальное и конечное состояния исполнителя при их выполнении.

Задача 6

Нарисовать орнамент, состоящий из квадратов, расположенных по краю поля.

Здесь вводится еще одна структурная команда — ветвление. На примере этой задачи еще раз демонстрируется методика последовательной детализации. Причем, в отличие от предыдущих программ, здесь используются два шага детализации, поскольку в процедуре РЯД содержится обращение к процедуре следующего уровня — КВАДРАТ.

Разобравшись в рассмотренных задачах, выполнив самостоятельные задания аналогичного типа, ученики должны усвоить два основных принципа структурной методики алгоритмизации (структурного программирования):

- всякий алгоритм может быть построен с использованием трех типов управляющих структур: следования, ветвления, цикла;
- при построении сложных алгоритмов следует применять метод последовательной детализации.

О способах описания алгоритмов. Традиционно в школьной информатике используются два способа описания алгоритмов: блок-схемы и учебный Алгоритмический язык. В базовом курсе информатики необходимо использовать обе эти формы. Основное достоинство блок-схем — наглядность алгоритмической структуры. Однако это качество проявляется лишь в том случае, если изображение блок-схемы происходит стандартным способом. Эта тема затрагивается в разделе 9.1 «Что такое структурное программирование» в части II учебника. Основным следствием освоения учениками структурной методики должно стать умение при построении алгоритмов «мыслить структурами», например, исходя из условия задачи, делать вывод, подобный следующему:

«алгоритм решения данной задачи будет представлять собой два вложенных цикла». Структурно изображенные блок-схемы помогают такому видению алгоритма. Рассмотрим, например, блок-схемы на рис. 15.5.

На первый взгляд, трудно понять, что на них изображен один и тот же алгоритм. Из схемы а) четко видна его структура: цикл-пока с вложенным ветвлением. В схеме б) довольно сложно усмотреть эту же структуру. Блок-схема а) нарисована стандартно, блок-схема б) — произвольно.

Алгоритмический язык (АЯ) — это текстовая форма описания алгоритма. Она ближе к языкам программирования, чем блок-схемы. Однако это еще не язык программирования. Поэтому строгого синтаксиса в АЯ нет. Для структурирования текста алгоритма на АЯ используются строчные отступы. При этом соблюдается следующий принцип: все конструкции одного уровня вложенности записываются на одном вертикальном уровне; вложенные конструкции смещаются относительно внешней вправо. Соблюдение этих правил улучшает наглядность структуры алгоритма, однако не дает такой степени наглядности, как блок-схемы. Приемы структурного изображения блок-схем и описания алгоритмов на АЯ проиллюстрированы в разделе 9.1.

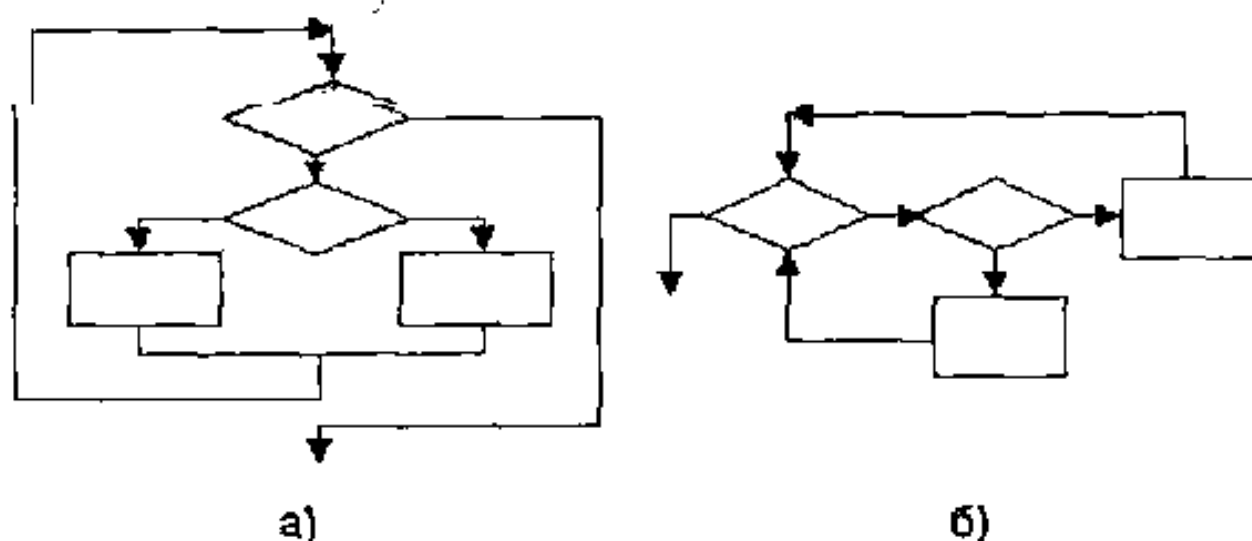


Рис. 15.5

- а) Пример структурного изображения алгоритма
б) Пример неструктурного изображения алгоритма

16. Как работает процессор ЭВМ

Разделы учебника: часть I, § 43–49.

Основные цели. Продолжение изучения архитектуры ЭВМ на уровне знакомства с устройством и работой процессора. Получение представления о программе на машинном языке, машинной команде, автоматическом исполнении программы процессором.

Изучаемые вопросы

- Двоичная машинная арифметика.
- Представление целых чисел в памяти ЭВМ.
- Язык машинных команд.
- Структура программы на машинном языке.
- Цикл работы процессора.

Общие методические рекомендации

Вспомним определение архитектуры ЭВМ, данное в § 7 учебника и в разделе 7 части 2 настоящего пособия: *под архитектурой ЭВМ понимают описание устройства и работы компьютера, достаточное для пользователя или программиста.*

В базовом курсе вплоть до данной темы изучение компьютера происходило с позиции пользователя, то есть человека, работающего с прикладным программным обеспечением. Программирование — профессиональный род деятельности, требующий от специалиста владения языками программирования. Профессиональными программистами созданы все существующие средства системного и прикладного программного обеспечения ЭВМ.

Знания программиста об архитектуре ЭВМ должны быть глубже, чем знания пользователя. В данной теме ученики знакомятся с новыми для них подробностями архитектуры ЭВМ, которые должны быть известны программисту. Это вопросы об устройстве и работе процессора ЭВМ, о представлении чисел в памяти ЭВМ, о языке команд процессора.

Основной задачей раздела является продолжение знакомства учащихся с принципами архитектуры ЭВМ, сформулированными в 1946 году Джоном фон Нейманом (принципами Неймана). В литературе по программированию устоялось словосочетание «машина Неймана». Под этим понятием подразумеваются определенные принципы организации архитектуры ЭВМ. К ним относятся:

1) состав устройств и структура однопроцессорной ЭВМ;

2) использование двоичной системы счисления в машинной арифметике;

3) адресуемость памяти ЭВМ;

4) хранение данных и программ в общей памяти ЭВМ;

5) структура машинной команды и состав системы команд процессора;

6) цикл работы процессора (алгоритм выполнения процессором программы).

Некоторые из этих принципов уже были затронуты в предыдущих разделах курса. Вопросы о составе устройств ЭВМ, о дискретной (битовой) структуре памяти, об адресуемости памяти рассматривались при изучении темы «Первое знакомство с компьютером». Там же рассказывалось о принципе совместного хранения в оперативной памяти обрабатываемых данных и исполняемой программы. В данном разделе, в соответствии с методом дидактической спирали, происходит возврат к этим же вопросам, но на другом, более высоком уровне. Кроме того, изучаются совершенно новые для учеников элементы архитектуры ЭВМ: язык машинных команд, устройство процессора ЭВМ, алгоритм работы процессора при выполнении программы.

Было бы слишком сложно в рамках базового курса изучать эти вопросы в полном объеме на примере реальной ЭВМ. Поэтому в учебнике используется методический прием, который встречается в ряде других учебных пособий по информатике и программированию: рассматривается *учебный компьютер (УК)*, то есть упрощенная модель реального компьютера.

Здесь он называется «УК Нейман». Это название подчеркивает тот факт, что архитектура учебного компьютера соответствует принципам Дж. фон Неймана.

В школьной информатике известны и другие модели учебных компьютеров: это «Кроха» из учебника А. Г. Гейна и др. [12], «Малютка» из учебника [27]; ToyCom из пакета учебного ПО фирмы БИТ. В учебниках А. П. Ершова, А. Г. Кушниренко также рассматривается упрощенная модель работы процессора с архитектурой типа PDP-11. Учебный компьютер с архитектурой процессора типа Intel описан и реализован Е. А. Ереминым [32].

По поводу учебного компьютера «УК Нейман» возможны возражения в том плане, что его архитектура не соответствует архитектуре современных ЭВМ. Действительно, она ближе к архитектуре ЭВМ первого и второго поколений. Однако, по мнению авторов, использование «УК Нейман» в базовом курсе вполне целесообразно. Данная модель способствует достижению главной цели изучения темы «Как работает компьютер»: получения учащимися представления о механизме программного управления работой компьютера. Основное преимущество этой модели — относительная простота. Трехадресная структура команды выглядит более естественной для арифметических расчетов. Из множества существующих способов адресации данных используется лишь один: прямая адресация данных в ОЗУ. В условиях ограниченного учебного времени данное педагогическое средство позволяет учащимся освоить основные понятия темы.

При желании в данном разделе курса учитель может использовать и другие модели учебных компьютеров. Дальнейшее изложение ведется в форме общей методической схемы, которая может быть реализована при работе с любым типом УК. Мы будем показывать реализацию этой схемы на примере «УК Нейман».

Назначение «УК Нейман» — вычисления с целыми числами. Следует отметить, что большинство учебных компьютеров, описываемых в других источниках, так-

же ориентированы на работу с целыми числами. В следующей таблице представлены общие характеристики «УК Нейман»*:

Учебный компьютер «УК НЕЙМАН»

Назначение «УК НЕЙМАН»:	арифметические расчеты с целыми числами.
Интерфейс с пользователем:	ввод с клавиатуры, вывод на экран.
Размер оперативной памяти:	256 байтов.
Размер машинного слова:	4 байта.
Представление чисел в памяти:	в форме с фиксированной точкой.
Система счисления:	двоично-шестнадцатеричная.
Язык программирования:	язык машинных команд (ЯМК).
Структура команд:	трехадресная.
Система команд:	команды целочисленной арифметики и минимальный набор команд управления.
Режимы работы:	режим ввода программы; режим исполнения программы.

Методические рекомендации по изложению теоретического материала

1. Методические указания по теме «Двоичная система счисления и двоичная арифметика» присутствуют в разделе 6 части 2 настоящего пособия. В зависимости от учебной программы, по которой работает учитель, данный вопрос может изучаться как в разделе «Системы счисления», так и в данном разделе. Еще раз перечислим основные утверждения, раскрываемые в этой теме:

* Исполнитель «УК Нейман» для IBM PC реализован Т. С. Белозеровой. Программа может быть получена через Интернет по адресу: <http://emc.km.ru/neumann/>

- согласно принципу Неймана, арифметические вычисления на компьютере выполняются в двоичной системе счисления;
- преимущество двоичной системы счисления в простоте двоичной арифметики, в относительной простоте и надежности ее аппаратной реализации;
- шестнадцатеричная система счисления используется в компьютере для компактного представления внутренней двоичной информации, адресов внутренней памяти.

2. Вопрос о внутреннем (машинном) представлении чисел рассмотрим несколько подробнее, чем это делается в учебнике. Структурные единицы памяти компьютера — бит, байт и машинное слово. Причем, понятия бита и байта универсальны и не зависят от модели компьютера, а размер машинного слова зависит от типа процессора ЭВМ. Если машинное слово для данного компьютера равно одному байту, то такую машину называют восьмиразрядной (8 битов); если машинное слово состоит из двух байтов, то это 16-разрядный компьютер; 4-х байтовое слово у 32-разрядных ЭВМ. Обсуждение вопроса о том, как представляются числа в памяти ЭВМ, будем вести на примере 16-разрядной машины.

Числа в памяти ЭВМ хранятся в двух форматах: в формате с *фиксированной точкой* и в формате с *плавающей точкой*. Под точкой здесь и в дальнейшем подразумевается знак разделения целой и дробной частей числа. Формат с фиксированной точкой используется для хранения в памяти целых чисел. В этом случае число занимает одно машинное слово памяти (16 битов). Чтобы получить внутреннее представление целого положительного числа N в форме с фиксированной точкой, нужно:

- 1) перевести число N в двоичную систему счисления;
- 2) полученный результат дополнить слева незначащими нулями до 16 разрядов.

Например: $N = 1607_{10} = 11001000111_2$. Внутреннее представление этого числа в машинном слове будет следующим:

0000	0110	0100	0111
------	------	------	------

В сжатой шестнадцатеричной форме этот код запишется так: 0647.

Двоичные разряды в машинном слове нумеруются от 0 до 15 справа налево. Старший, 15-й разряд в машинном представлении любого положительного числа равен нулю. Поэтому максимальное целое число в такой форме равно:

$$\begin{aligned} 0111 \ 1111 \ 1111 \ 1111_2 &= 7FFF_{16} = \\ &= (2^{16} - 1) = 32 \ 767_{10}. \end{aligned}$$

Для записи внутреннего представления целого отрицательного числа ($-N$) нужно:

- 1) получить внутреннее представление положительного числа N ;
- 2) получить обратный код этого числа заменой 0 на 1 и 1 на 0;
- 3) к полученному числу прибавить 1.

Определим по этим правилам внутреннее представление числа -1607_{10} :

$$\begin{array}{rcl} 1) & 0000 & 0110 \ 0100 \ 0111 \\ 2) & 1111 & 1001 \ 1011 \ 1000 \\ 3) & & \quad \quad \quad +1 \\ & & \hline & 1111 & 1001 \ 1011 \ 1001 - \text{результат.} \end{array}$$

Шестнадцатеричная форма результата: F9B9.

Описанный способ представления целого отрицательного числа называют *дополнительным кодом*. Старший разряд в представлении любого отрицательного числа равен 1. Следовательно, он указывает на знак числа и поэтому называется *знаковым разрядом*.

Применение дополнительного кода для внутреннего представления отрицательных чисел дает возможность

заменить операцию вычитания операцией сложения с отрицательным числом: $N - M = N + (-M)$. Очевидно, должно выполняться следующее равенство: $N + (-N) = 0$. Выполним такое сложение для полученных выше чисел 1607 и -1607:

$$\begin{array}{r}
 0000\ 0110\ 0100\ 0111\quad 1607 \\
 +1111\ 1001\ 1011\ 1001\quad -1607 \\
 \hline
 1\ 0000\ 0000\ 0000\ 0000\quad 0
 \end{array}$$

Единица в старшем разряде, получаемая при сложении, выходит за границу разрядной сетки машинного слова и исчезает, а в памяти остается ноль.

Выход двоичных знаков за границу ячейки памяти, отведенной под число, называется *переполнением*. Для вещественных чисел такая ситуация является аварийной. Процессор ее обнаруживает и прекращает работу (прерывание по переполнению). Однако при вычислениях с целыми числами переполнение не фиксируется как аварийная ситуация и прерывания не происходит.

Двоичное 16-разрядное число 1000 0000 0000 0000 = 2^{15} является «отрицательным самому себе»:

$$\begin{array}{r}
 1) \quad 1000\ 0000\ 0000\ 0000\quad 2^{15} \\
 2) \quad 0111\ 1111\ 1111\ 1111 \\
 3) \quad \quad \quad \quad \quad \quad +1 \\
 \hline
 \quad 1000\ 0000\ 0000\ 0000\quad -2^{15}
 \end{array}$$

Этот код используется для представления значения $-2^{15} = -32\ 768$. Следовательно, диапазон представления целых чисел в 16-разрядном машинном слове:

$$-32768 \leq N \leq 32767.$$

В общем случае для k -разрядного машинного слова этот диапазон следующий:

$$-2^{k-1} \leq N \leq 2^{k-1} - 1.$$

Формат с плавающей точкой используется для представления как целочисленных значений, так и значений с дробной частью. В математике такие числа называют действительными, в программировании — вещественными.

Формат с плавающей точкой предполагает представление вещественного числа (R) в форме произведения мантиссы (m) на основание системы счисления (n) в некоторой целой степени, которую называют порядком (p):

$$R = \pm m \times n^p.$$

Порядок указывает, на какое количество позиций и в каком направлении должна сместиться («переплыть») точка в мантиссе. Пример: $25,324_{10} = 0,25324 \times 10^2$. Однако справедливы и следующие равенства:

$$25,324 = 2,5324 \times 10^1 = 0,0025324 \times 10^4 \text{ и т. д.}$$

Следовательно, представление числа в форме с плавающей точкой неоднозначно. Чтобы не было неоднозначности, в ЭВМ используют нормализованную форму с плавающей точкой. Мантисса в нормализованной форме должна удовлетворять условию:

$$0,1_n \leq m < 1.$$

Для рассмотренного числа нормализованной формой будет: $0,25324 \times 10^2$.

В памяти ЭВМ мантисса представляется как целое число, содержащее только ее значащие цифры (0 целых и запятая не хранятся). Следовательно, задача внутреннего представления вещественного числа сводится к представлению пары целых чисел: мантиссы (m) и порядка (p). В рассмотренном нами примере $m = 25\,324$, $p = 2$.

В разных типах ЭВМ используются разные варианты организации формата с плавающей точкой. Вот пример одного из вариантов представления вещественного числа в 4-байтовой ячейке памяти:

	Машинный порядок	М	А	Н	Т	И	С	С	А
	1 байт	2 байт		3 байт			4 байт		

В старшем бите 1-го байта хранится знак числа: 0 — плюс, 1 — минус; 7 оставшихся битов первого байта содержат машинный порядок; в следующих трех байтах хранятся значащие цифры мантиссы.

В рамках базового курса информатики вопрос о представлении вещественных чисел может рассматриваться лишь на углубленном уровне. Теоретический материал и практические задания на эту тему имеются в разделе 3.1.4 задачника-практикума.

3. Вопрос о представлении чисел для учебного компьютера «УК Нейман» разбирается в § 47 учебника. Размер машинного слова «УК Нейман» — 4 байта, то есть 32 бита. Учебный компьютер работает только с целыми числами. Из сформулированного выше правила следует, что диапазон целых чисел, представимых в памяти «УК Нейман»: -2^{31} до $2^{31}-1$. Шестнадцатеричный код наибольшего положительного числа следующий:

7F FF FF FF

В этом же параграфе описывается структура памяти учебного компьютера. Структурной единицей памяти является ячейка, помещающая в себе машинное слово. Адрес ячейки равен адресу входящего в нее младшего байта и кратен четырем. Объем памяти — 256 байтов, следовательно, память делится на 64 ячейки. Диапазон шестнадцатеричных адресов байтов памяти — от 0 до FF. Диапазон адресов ячеек памяти — от 0 до FC (с шагом 4).

4. *Принцип совместного хранения программы и данных.* Кратко его называют принципом хранимой программы. Этот принцип принято считать главной идеей архитектуры ЭВМ, предложенной Дж. фон Нейманом. На первых, «донецких» ЭВМ хранение программы и данных было раздельным. Идея Неймана позволила сделать управление компьютером более гибким. Появилась возможность интерпретировать команды как данные и, следовательно, выполнять операции над командами (модифицировать команды в ходе выполнения программы). Еще одно важное следствие этого принципа — возможность изменять порядок выполнения программы с помощью команд условного и безусловного переходов.

Общий принцип хранимой программы	Реализация принципа хранимой программы в УК Нейман
ОЗУ КОМАНДЫ ПРОГРАММЫ ДАННЫЕ Свободная часть памяти	ОЗУ яч. 00 1-я команда программы яч. 04 2-я команда программы яч. 08 3-я команда программы яч. N последняя команда программы ЧИСЛА (константы и переменные) Свободная часть памяти яч. FF.....

5. Система команд ЭВМ. Существуют четыре основных типа машинных команд, которые представлены в следующей таблице.

Основные типы команд процессора	Типы команд, реализованные в «УК Нейман»
Арифметические и логические команды: выполняют четыре действия арифметики и некоторые логические операции	Арифметические команды (сложения, вычитания, умножения, целого деления и деления по модулю)
Команды внутренней пересылки данных: позволяют перемещать полностью машинные слова или байты из одного места памяти в другое	Команда пересылки машинного слова
Команды передачи управления: служат для определения порядка выполнения команд программы (условные и безусловные переходы)	Команды условного и безусловного перехода

Основные типы команд процессора	Типы команд, реализованные в «УК Нейман»
команды ввода/вывода: служат для организации обмена данными между оперативной памятью и внешними устройствами	Отдельных команд ввода и вывода нет. Ввод и вывод организуются с помощью команды пересылки через буферную ячейку памяти с адресом FC

Общая структура машинной команды	Общая структура машинной команды в «УК Нейман»						
Всякая машинная команда состоит из двух частей: кода операции (КОП) и адресной части. <table><tr><td>КОП</td><td>Адресная часть</td></tr></table> В машинной команде может содержаться следующая информация: 1) какая выполняется операция; 2) какие используются операнды; 3) куда поместить результат операции; 4) какую команду выполнять следующей. Ответ на вопрос 1 задается кодом операции — КОП. Ответы на вопросы 2–4 чаще всего определяются указанием адресов памяти, где хранятся операнды, куда помещается результат, где хранится следующая исполняемая команда	КОП	Адресная часть	Команда имеет <i>трехадресную структуру</i>: <table><tr><td>КОП</td><td>A1</td><td>A2</td><td>A3</td></tr></table> Во всех командах используется <i>прямая адресация памяти</i>: A1, A2, A3 — адреса ячеек ОЗУ. В арифметических командах A1, A2 — адреса операндов, A3 — адрес результата. В команде пересылки A1 — адрес исходного слова, A3 — адрес пересылки. Затем выполняется команда из следующей ячейки. В командах управления A3 — адрес следующей исполняемой команды	КОП	A1	A2	A3
КОП	Адресная часть						
КОП	A1	A2	A3				

Система команд «УК Нейман»

КОП	Операция	Пояснение
00	Пересылка	$(A1) \Rightarrow A3$
01	Сложение	$(A1) + (A2) \Rightarrow A3$
02	Вычитание	$(A1) - (A2) \Rightarrow A3$
03	Умножение	$(A1) \times (A2) \Rightarrow A3$
04	Деление нацело	$(A1) \text{ div } (A2) \Rightarrow A3$ — целая часть от деления
05	Получение остатка от деления нацело	$(A1) \text{ mod } (A2) \Rightarrow A3$ — оста- ток от деления нацело
0A	Условный переход	Переход к ячейке A3 по регистру-признаку результата $W = 1$
0B	Безусловный переход	Безусловный переход к ячейке A3
77	Останов	Прекращение выполнения программы
FF	Пустая команда	При выполнении пропускается

Запись $(A1) + (A2) \Rightarrow A3$ надо понимать так: содержимое ячейки с адресом A1 сложить с содержимым ячейки с адресом A2 и результат записать в ячейку A3; $(A1) \Rightarrow A3$ — содержимое ячейки A1 занести (скопировать) в ячейку A3.

В таблице серым фоном выделен минимальный набор команд, приведенный в учебнике.

Как уже отмечалось выше, ввод и вывод данных в «УК Нейман» реализован с помощью команды пересылки через буферную ячейку памяти FC. Схематически это представлено на рис. 16.1.



Рис. 16.1. Выполнение ввода и вывода в «УК Нейман»

Данные, набираемые на клавиатуре, заносятся в ячейку FC. Содержимое ячейки FC всегда отражается на экране.

6. Устройство и работа процессора. В § 49 учебника даются самые общие понятия на эту тему. Прочитав его, ученики узнают, что

- **назначение процессора:** управлять работой компьютера по заданной программе; выполнять команды обработки информации;
- **состав процессора:** АЛУ — арифметико-логическое устройство, УУ — устройство управления, регистры процессорной памяти;
- **алгоритм работы процессора:** последовательное выполнение команд программы, начиная с первой до команды «останов».

Если ограничиться теоретическим изучением данной темы на минимальном уровне, то этого вполне достаточно. Но уже использование программы-исполнителя «УК Нейман» требует более детальных знаний о структуре процессора учебного компьютера, об алгоритме работы процессора при выполнении программы. Изложим эти вопросы более подробно.

Прежде всего, уточним состав регистров и их назначение. В «УК Нейман» всего шесть регистров. Каждый из них служит своего рода черновиком, используя который, процессор выполняет расчеты и записывает промежуточные результаты. У каждого регистра есть определенное назначение (рис. 16.2).

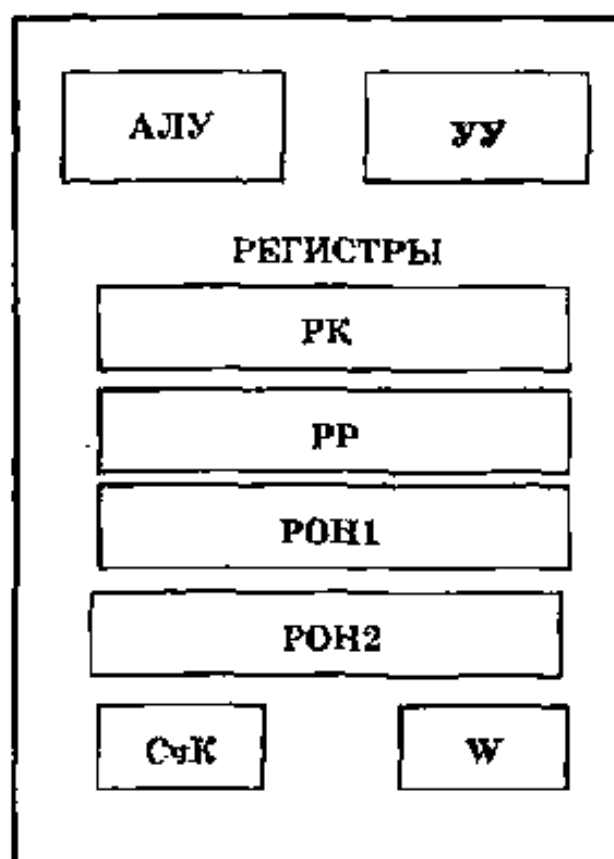


Рис. 16.2. Структура процессора «УК Нейман»

- СчК** — счетчик команд; в нем последовательно меняются адреса исполняемых команд;
- РК** — регистр команд, в который при выполнении программы помещается текущая выполняемая команда;
- РР** — регистр результата, в него первоначально помещается результат выполненной операции;
- РОН1 и РОН2** — регистры общего назначения, в которые при выполнении команды помещаются операнды (слагаемые, множители и так далее);
- W** — регистр-признак знака результата, фиксирует знак результата очередной операции: если результат операции положительный (> 0), то $W = 1$, иначе $W = 0$.

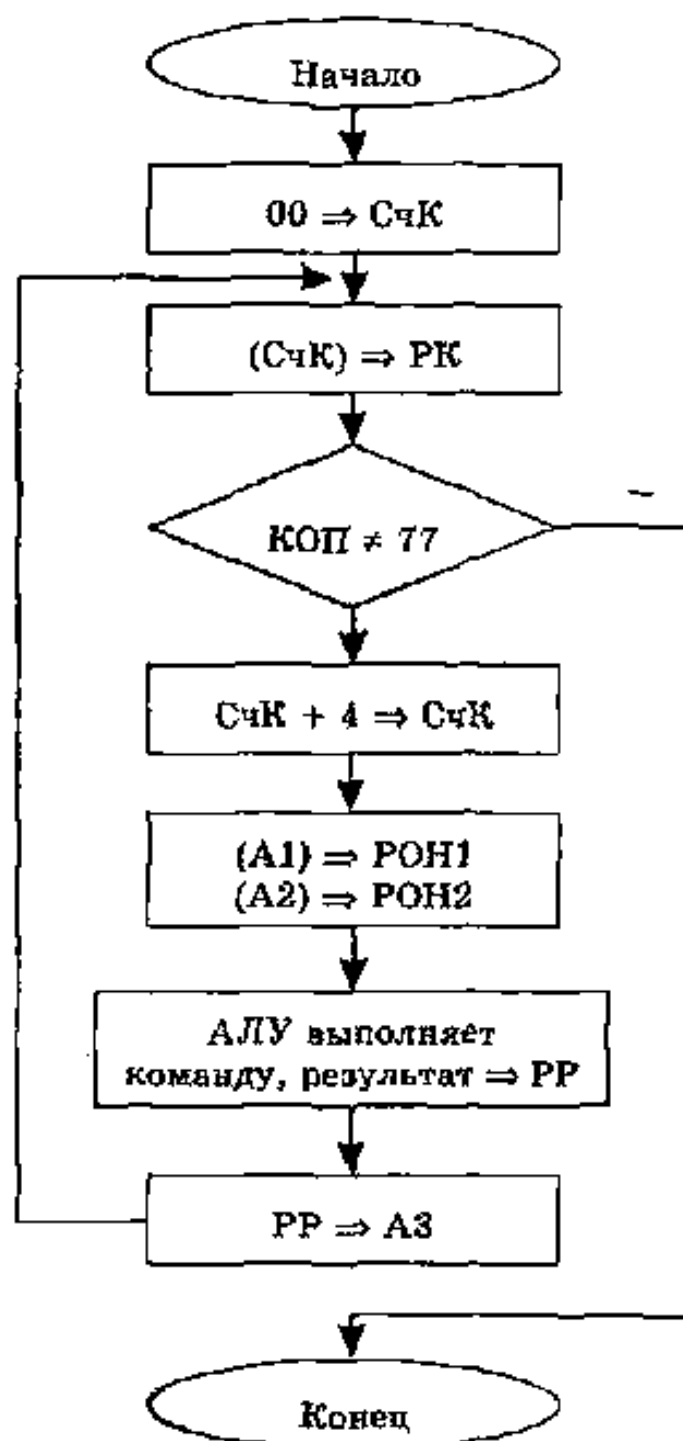


Рис. 16.3. Алгоритм работы процессора

Регистры РК, РР, РОН1 и РОН2 — 32-разрядные. Их размер равен размеру машинного слова и совпадает с размером ячейки. Счетчик команд (СчК) и регистр-признак знака результата (W) имеют размер 1 байт.

Исполнение программы начинается с того, что в СчК (счетчик команд) заносится 00 — адрес первой команды. Затем вступает в работу УУ (управляющее

устройство). Оно организует запись в РК содержимого ячейки, адрес которой указан в СчК. По коду операции УУ определяет, не останов ли это. Если «да», то работа прекращается, иначе содержимое СчК увеличивается на 4, тем самым подготавливается выполнение следующей команды.

Управляющее устройство продолжает анализировать код операции, в зависимости от которого выбирается тот или иной вариант действия. АЛУ (арифметико-логическое устройство) работает при выполнении арифметических операций, после занесения операндов в регистры общего назначения РОН1 и РОН2. Результат операции заносится в РР (регистр результата). Затем УУ переписывает результат в ячейку памяти и переходит к выполнению следующей команды.

Следует учесть, что приведенный алгоритм работы процессора является упрощенным вариантом полного алгоритма, поскольку здесь не учтены команды управления (условный и безусловный переходы) и, следовательно, он описывает только исполнение линейных программ.

Рассмотрим этот процесс на примере выполнения арифметических операций в программе 11.1, приведенной в § 48 учебника. Проанализируем по шагам выполнение процессором команды сложения двух чисел ($A + B$), которая хранится в ячейке 0С. Переменная A хранится в ячейке 20 и ее значение равно $200_{10} = C8_{16}$; переменная B — в ячейке 24 и ее значение равно $437_{10} = 1B5_{16}$.

К моменту выполнения команды сложения в счетчике команд (СчК) находится адрес ячейки 0С.

Шаг 1. Устройство управления (УУ) помещает в регистр команд (РК) содержимое ячейки, адрес которой равен значению СчК:

Адрес	Память			
0С	01	20	24	2С
10	03	2С	28	2С
...	...	...	...	...
20	00	00	00	С8
24	00	00	01	В5
28	00	00	00	02

Регистры				
СчК	0С			
РК	01	20	24	2С
РОН1				
РОН2				
РР				
W				

Анализируя КОП, УУ устанавливает, что это не команда останова и продолжает работу.

Шаг 2. Значение СчК увеличивается на 4. В РОН1 заносится содержимое ячейки с адресом 20, в РОН2 — содержимое ячейки с адресом 24:

Адрес	Память			
...	...	...	...	...
20	00	00	00	С8
24	00	00	01	В5
28	00	00	00	02
2С				

Регистры				
СчК	10			
РК	01	20	24	2С
РОН1	00	00	00	С8
РОН2	00	00	01	В5
РР				
W				

Шаг 3. В работу вступает АЛУ, которое выполняет операцию в соответствии с КОП (01 — сложить). Результат помещается в регистр результата (РР). В регистре W вырабатывается признак результата в зависимости от его знака:

Адрес	Память			
...	...	...	...	...
20	00	00	00	С8
24	00	00	01	В5
28	00	00	00	02
2С				

Регистры				
СчК	10			
РК	01	20	24	2С
РОН1	00	00	00	С8
РОН2	00	00	01	В5
РР	00	00	02	7D
W	01			

Шаг 4. Устройство управления помещает значение, полученное в РР, в ячейку, указанную в АЗ команды с адресом (2С):

Адрес	Память			
...	...	...	...	...
20	00	00	00	С8
24	00	00	01	В5
28	00	00	00	02
2С	00	00	02	7D

Регистры				
СЧК	10			
РК	01	20	24	2С
РОН1	00	00	00	С8
РОН2	00	00	01	В5
РР	00	00	02	7D
W	01			

После этого процессор возвращается к выполнению шага 1.

Описанная последовательность действий называется *циклом работы процессора*. Этот цикл закончится тогда, когда УУ обнаружит код останова 77.

Методические рекомендации по решению задач

Изучение данной темы на уровне, соответствующем изложению материала в главе 11 учебника, не предполагает полноценное обучение учеников программированию на ЯМК. Как уже было отмечено, основная цель состоит в том, чтобы *дать учащимся представление* о том, что такое машинная программа, язык машинных команд. В таком объеме эта тема может изучаться теоретически, *без использования исполнителя «УК Нейман»*. Достаточно добиться того, чтобы ученики понимали смысл команд, сумели разобраться в небольшой линейной программе на ЯМК. В этом разделе ученикам можно предлагать задачи следующего содержания:

1) Дана программа (последовательность команд); что получится в результате ее выполнения?

2) Составить по аналогии программу для решения заданной задачи.

3) Как изменится результат, если в программу внести заданное изменение?

4) Что нужно изменить в данной программе, чтобы получить заданный результат?

Пример 1.

[5]: часть I, § 48, задание № 10. Дана машинная команда

02 20 24 28

и содержимое двух ячеек памяти:

яч. 20 00 00 00 5C

яч. 24 00 00 00 28

Какое действие определяет эта команда? Каким будет содержимое ячейки 28 после выполнения команды?

02 — код операции вычитания. Следовательно, данная команда вычитает из содержимого ячейки 20 содержимое ячейки 24 и записывает результат в ячейку 28.

Поскольку вычитание чисел в шестнадцатеричной системе для учеников может оказаться затруднительным, то ответ на второй вопрос удобнее искать через десятичную систему счисления:

$$5C_{16} = 5 \times 16 + 12 = 92_{10}$$

$$28_{16} = 2 \times 16 + 8 = 40_{10}$$

$$92 - 40 = 52_{10} = 34_{16}$$

Ответ: в ячейке 28 будет получен следующий код:

00 00 00 34

Пример 2.

[5]: часть I, § 48, задание № 10. Пусть значения ячеек 20 и 24 такие же, как в предыдущем задании. Определите их значения после выполнения трех следующих команд (в скобках указаны адреса команд):

(10) 00 20 00 28

(14) 00 24 00 20

(18) 00 28 00 24

Сделайте вывод о том, что делает эта программа.

В этом задании рассматривается одна из базовых задач программирования: обмен значениями между двумя переменными величинами. Ее понимание имеет большое значение для дальнейшего изучения программирования.

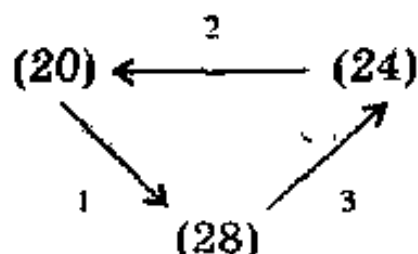
Рекомендуется разобрать эту задачу в режиме трассировки, то есть проследить за изменением значений переменных (ячеек памяти) после выполнения каждой команды. Трассировочную таблицу можно представить в таком виде:

Команда	Яч. 20	Яч. 24	Яч. 28
	00 00 00 5C	00 00 00 28	
00 20 00 28	00 00 00 5C	00 00 00 28	00 00 00 5C
00 24 00 20	00 00 00 28	00 00 00 28	00 00 00 5C
00 28 00 24	00 00 00 28	00 00 00 5C	00 00 00 5C

Основным результатом выполнения этой программы является то, что ячейки 20 и 24 обменялись значениями.

При разборе этой задачи необходимо обсудить с учениками следующие моменты. Программа состоит из трех команд пересылки. В результате выполнения команды пересылки в ячейку с адресом A3 копируется значение из ячейки с адресом A1. При этом содержимое ячейки A1 не изменяется. Такое действие принято называть присваиванием: *значение ячейки A1 присваивается ячейке A3.*

И, наконец, главный вывод этой задачи: для обмена значений между двумя ячейками требуется третья ячейка. В рассмотренной программе роль такой вспомогательной ячейки выполняет ячейка 28. Обмен значениями происходит по схеме:



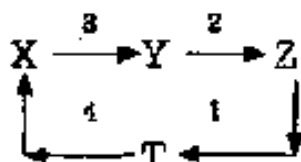
Здесь стрелки указывают направление пересылки, а цифры — порядок выполнения команд пересылки.

Пример 3

Задачник-практикум: раздел 3.2, задача № 11. Составить программу выполнения циклического обмена содержимого трех ячеек памяти: $X \rightarrow Y \rightarrow Z$. Например, если исходные значения были такими: $X = 1$, $Y = 2$, $Z = 3$, то после обмена они станут следующими: $X = 2$, $Y = 3$, $Z = 1$.

Программа для решения этой задачи составляется по аналогии с программой из предыдущей задачи. Для ее получения следует:

- 1) понять, что для решения этой задачи также потребуется дополнительная ячейка памяти; обозначим ее T ;
- 2) представить схему пересылок значений между ячейками:



- 3) выделить четыре ячейки памяти под величины X , Y , Z , T ; например: $X = 20$, $Y = 24$, $Z = 28$, $T = 2C$;
- 4) написать программу из четырех команд пересылки:

00 28 00 2C	(Z $\Rightarrow$ T)
00 24 00 28	(Y $\Rightarrow$ Z)
00 20 00 24	(X $\Rightarrow$ Y)
00 2C 00 20	(T $\Rightarrow$ X)

Пример 4

[5]: часть I, § 49, задание № 3. Составить программу вычисления по формуле $(A+B)^2$ для данных значений A и B .

Программу для решения этой задачи следует составлять по аналогии с программой 11.1 из § 48 учеб-

ника. Разница состоит в том, что вводить нужно две величины, после вычисления суммы $A + B$ полученное значение нужно умножить само на себя, то есть возвести в квадрат.

Следует обратить внимание учеников на то, что составление программы нужно начинать с распределения памяти под данные. При этом адреса данных и адреса команд программы не должны пересекаться. По аналогии с программой 11.1 память распределим так: A — ячейка 20, B — ячейка 24. С целью экономии памяти результат также можно получить в одной из этих ячеек.

Окончательно программа будет иметь следующий вид:

Адрес	КОП	A1	A2	A3	Пояснения
00	00	FC	00	20	Ввод A
04	00	FC	00	24	Ввод B
08	01	20	24	20	$A + B$
0C	03	20	20	20	$(A + B) \times (A + B)$
10	00	20	00	FC	Вывод результата
14	77	00	00	00	Стоп

При наличии исполнителя «УК Нейман» эту программу можно выполнить на компьютере. Аналогично решаются и другие задачи из данного раздела.

Дополнительный материал для учителя

Архитектура ЭВМ, отличная от архитектуры Неймана. Согласно фон Нейману, в состав компьютера входит один процессор, который осуществляет обработку информации, управляет работой всех устройств компьютера. Отход от однопроцессорной архитектуры начался с третьего поколения ЭВМ. На машинах серии IBM 360/370 появились специальные процессоры, управляющие работой внешних устройств (внешней

памятью, устройствами ввода/вывода), а центральный процессор выполнял лишь вычислительную работу.

В конце 70-х годов начинается производство особого класса компьютеров, которые получили название «супер-ЭВМ». На этих машинах был преодолен рубеж быстродействия в 1 млрд оп./с. В различных моделях супер-ЭВМ используются два основных принципа организации вычислений: *матричный* и *конвейерный*.

Матричная система — это множество взаимосвязанных процессоров, способных одновременно (параллельно) выполнять одинаковые вычислительные операции. Поэтому эффективность применения матричных систем зависит от возможности распараллеливания вычислений, то есть выделения таких операций в вычислительном алгоритме, которые могут выполняться одновременно, независимо друг от друга. Такие операции называются *векторными*, в отличие от *скалярных* операций, которые нельзя выполнять параллельно.

В конвейерных системах выполнение каждой команды разбивается на последовательные шаги (этапы). Конвейер — это система блоков процессора, выполняющих эти шаги друг за другом. Ускорение вычислений происходит за счет того, что каждый блок конвейера, выполнив свою часть работы с одними данными, переходит к обработке других данных. Применение конвейера также основано на выделении в алгоритме векторных операций.

Вот названия некоторых зарубежных моделей супер-ЭВМ: Cyber 205, Cray-3, SX-344. Например, производительность последней составляет 680 млн скалярных и 22 млрд векторных операций в секунду.

Подробно о супер-ЭВМ можно прочитать в книге [33].

17. Искусственный интеллект и базы знаний

Разделы учебника: часть 2, глава 11.

Основные цели. Познакомить учащихся с разделом информатики под названием «Искусственный интеллект». Раскрыть назначение систем искусственного интеллекта. Дать детям представление о базах знаний, логической модели знаний. Познакомить их с основами языка логического программирования Пролог; научить строить простые базы знаний на Прологе, формулировать к ним запросы.

Изучаемые вопросы

- ☐ Искусственный интеллект: цели и задачи.
- ☐ Что такое база знаний.
- ☐ Логическая модель знаний.
- ☐ База знаний на Прологе: факты, правила.
- ☐ Два типа целей.
- ☐ Механизм вывода.

Методические рекомендации по преподаванию материала

Тема «Искусственный интеллект и базы знаний» в школьном курсе информатики имеет несколько аспектов. Во-первых, ее цель — познакомить учащихся с одним из самых сложных и актуальных направлений в информатике — искусственным интеллектом. Это направление является своеобразной вершиной развития компьютерной техники и технологий. Дальнейшее развитие фундаментальной составляющей школьного курса информатики должно быть связано с увеличением объема раздела искусственного интеллекта в общем объеме курса. Этому будет способствовать появление простых и удобных инструментальных средств для разработки систем искусственного интеллекта, распространение общедоступных экспертных систем.

Другая сторона данного раздела — продолжение линии информационного моделирования в базовом курсе. В этой линии можно выделить два направления: моделирование объектов и процессов и моделирование знаний (рис. 17.1).

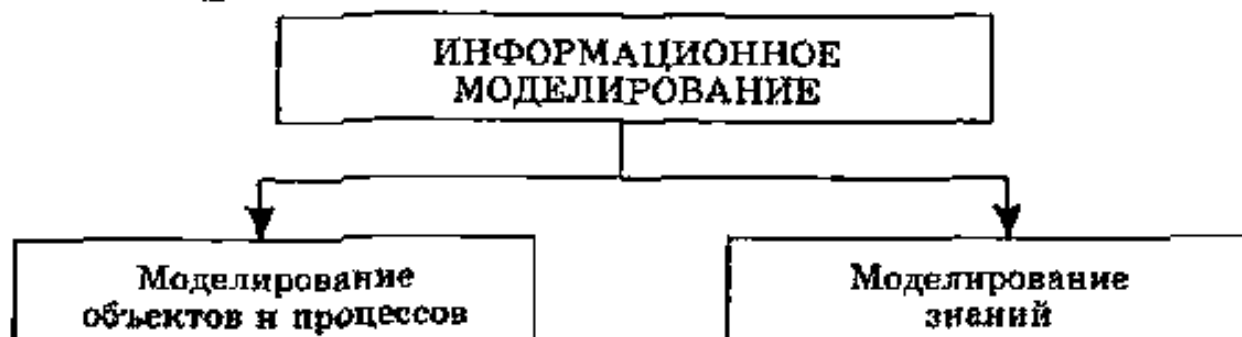


Рис. 17.1

Разделы 11.1 и 11.2 содержат материал общего характера на тему «Искусственный интеллект и базы знаний». Здесь рассказывается о целях и задачах исследований и разработок в области искусственного интеллекта, вводится представление о компьютерной базе знаний. Кроме того, дается общее представление о логической модели знаний, построенной на совокупности фактов, правил и аппарате формальной логики. Содержание данных разделов носит чисто теоретический характер. Дадим некоторые дополнительные сведения, которые позволят учителю иметь более широкое представление на данную тему, чем это можно почерпнуть из содержания учебника.

1. *Искусственный интеллект (ИИ)* как самостоятельное научное направление оформился в конце 60-х годов. Первый международный конгресс по искусственному интеллекту состоялся в США в 1969 году.

Задача ИИ — разработка интеллектуальных систем на базе компьютерной техники. Системы ИИ относятся к трудно формализуемым областям деятельности человека, для которых свойственна нестандартность принятия решений, элементы творчества. Для задач, решаемых методами ИИ, характерно наличие большо-

го количества степеней свободы, с числом вариантов поиска решения, приближающимся к бесконечности. Система ИИ — неформальный исполнитель. В отличие от жестко детерминированных программ она сама ищет путь решения поставленной задачи.

Сегодня искусственный интеллект — это обширная область исследований и разработок интеллектуальных систем. Основные направления ИИ представлены в схеме на рис. 17.2.

Как следует из схемы, проблемы ИИ решаются как техническими так и программными средствами. Существуют системы, сочетающие в себе те и другие средства, например в робототехнике.

Нейрокомпьютеры — это ЭВМ, которые принято считать машинами шестого поколения. В них с помо-



Рис. 17.2

Основные направления Искусственного интеллекта

пью электронных элементов воспроизводится система взаимодействующих нейронов — нервных клеток, составляющих человеческий мозг. Иначе говоря, нейрокомпьютер — это электронная модель человеческого мозга. С помощью нейрокомпьютеров удастся, например, решать проблему распознавания образов.

Интеллектуальные роботы. Как известно, роботы — это технические устройства, предназначенные для автоматизации человеческого труда. В отличие от роботов, работающих по жесткой схеме управления, интеллектуальные роботы обладают способностью адаптации к внешней обстановке, самоорганизации и самообучения.

Игры и творчество. Характерным примером этого направления являются шахматные программы. Самые развитые шахматные программы играют уже на гроссмейстерском уровне. Существуют программы, ориентированные на художественное творчество: сочинение музыки, стихов и др.

Компьютерная лингвистика. К этому направлению относятся решения двух основных проблем. Первая — это перевод текста с одного языка на другой (например, с английского на русский). Простой пословный перевод никого не удовлетворит. Программы должны учитывать сложные связи между структурой фразы и ее смыслом. Вторая проблема — это разработка интерфейса между ЭВМ и человеком на естественном языке. Проекты интеллектуальных компьютеров пятого-шестого поколений требуют решения этой проблемы.

Распознавание образов. К задачам этого направления относится, например, чтение компьютером рукописного текста вне зависимости от почерка автора. Еще одна задача — восприятие естественной речи.

Адаптивные обучающие системы позволяют для каждого обучаемого строить свой маршрут обучения, учитывать его интеллектуальные и психологические особенности.

Экспертные системы. Их еще называют «системы, основанные на знаниях». Это системы-решатели задач, системы-консультанты в определенной предметной области, заключающие в себе знания специалиста-эксперта.

2. Далее речь будет идти лишь о *системах, основанных на знаниях*. Специалисты по ИИ пытаются понять, как мыслит человек, решая какую-то проблему в определенной предметной области, например, решая математическую задачу, доказывая теорему, устанавливая диагноз больному, размышляя над конструкторской проблемой и т. п. Во всех этих ситуациях, во-первых, требуются определенные знания в данной области: знания математики, медицины, технические знания и пр. Во-вторых, требуется умение логически рассуждать, анализировать имеющуюся информацию, делать выводы. Если мы хотим научить компьютер решению таких проблем, значит, во-первых, в него нужно заложить необходимые знания, во-вторых, научить его логически мыслить.

Представленные в компьютерной памяти формализованные знания из определенной предметной области называются *базой знаний*. Компьютерная модель логических рассуждений на основе базы знаний называется *механизмом вывода*. Компьютерные системы ИИ, основанные на знаниях, должны обладать еще двумя важными свойствами. Во-первых, они должны общаться с пользователем на привычном для него естественном или профессиональном языке данной предметной области. Такая форма общения называется *интеллектуальным интерфейсом*. Во-вторых, эти системы должны быть *открыты для дополнения базы знаний, самообучаемы*, то есть должны уметь учитывать полученный опыт в решении проблем.

На рис. 17.3 показан состав и структура типичной экспертной системы, схема взаимодействия между отдельными ее блоками, а также общения системы и

человека. О назначении базы знаний, логического блока и блока интеллектуального интерфейса было сказано выше.

На схеме присутствует еще и блок объяснений, которого нет в других видах интеллектуальных систем. Благодаря этому блоку пользователь всегда может запросить объяснение, каким образом получено предлагаемое решение. Система сообщит всю цепочку логических рассуждений. При необходимости это может послужить доказательством правильности решения, может иметь обучающее значение, а может и помочь обнаружить недостатки в базе знаний системы или в логике рассуждений. Как было сказано выше, экспертные системы обычно открыты для дополнения и совершенствования.

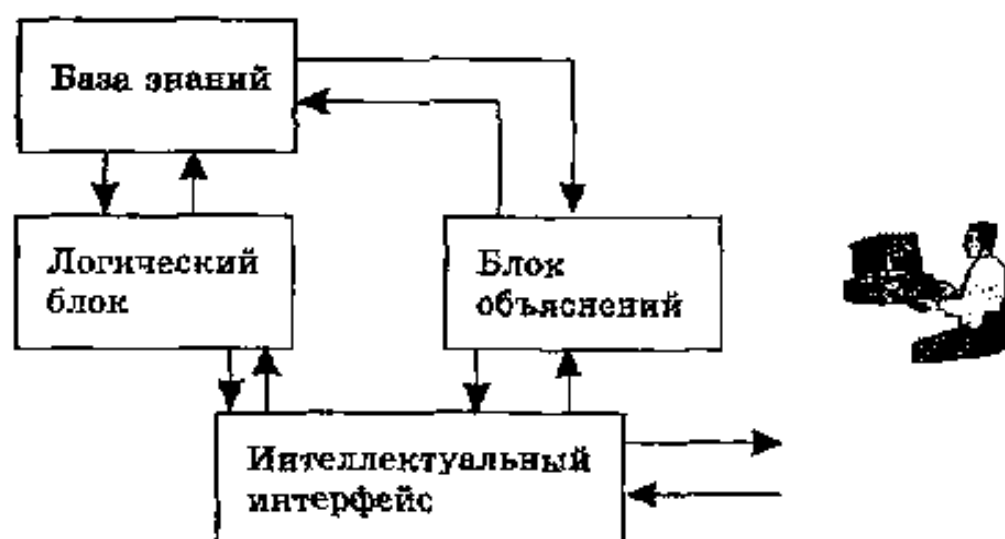


Рис. 17.3. Состав и структура экспертной системы

В этой схеме присутствует *пользователь*, то есть человек, на удовлетворение информационных потребностей которого настроена система. Существует требование к экспертной системе, согласно которому пользователь не должен видеть разницы между общением с компьютерной экспертной системой и общением со специалистом-экспертом.

В создании экспертной системы участвует *специалист-эксперт* (или группа экспертов) — основной ис-

точник знаний, а также *инженер по знаниям*. Задача последнего — принять от эксперта его знания, формализовать их и построить базу знаний, а также все остальные компоненты системы.

3. Информационной основой экспертной системы является *база знаний* — *компьютерная модель знаний в данной предметной области*. Существуют различные типы моделей знаний. Наиболее известные из них — продукционная модель, семантическая сеть, фреймы, логическая модель.

Продукционная модель знаний построена на правилах (они называются *продукциями*), представляемыми в форме:

ЕСЛИ выполняется некоторое условие
ТО выполняется некоторое действие

На основе поступающих данных экспертная система, анализируя имеющиеся правила, принимает решение о необходимых действиях. Например:

ЕСЛИ температура в помещении 15
ТО включить отопление

Продукционные модели часто используются в промышленных экспертных системах.

Семантические сети. Семантической сетью называется система понятий и отношений между ними, представленная в форме ориентированного графа. Это одна из наиболее гибких и универсальных форм моделей знаний. Подробная информация о семантических сетях дана в разделе 7.2 задачника-практикума. На рис. 17.4. приведен небольшой пример из этого раздела, представляющий в форме графа сведения, заключенные в предложении «Петух Петя является птицей, и он умеет кукарекать».

Фреймы. Фрейм — это некоторый абстрактный образ, относящийся к определенному типу объектов, понятий. Фрейм объединяет в себе атрибуты (характеристики), свойственные данному объекту. Фрейм име-

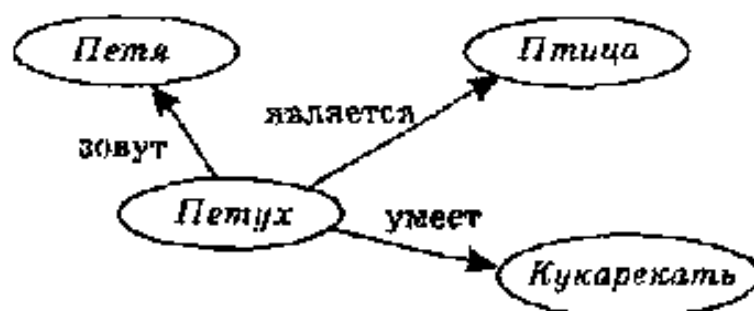


Рис. 17.4. Пример семантической сети

ет имя и состоит из частей, которые называются слотами. Изображается фрейм в виде цепочки:

Имя фрейма = <слот 1><слот 2>...<слот N>

Вот пример фрейма под названием «Битва»:

Битва = <кто?><с кем?><когда?><где?><результат>

Такое представление называется прототипом фрейма. Если же в слоты подставить конкретные значения, то получится экземпляр фрейма. Например:

Битва = <Царевич><Кожей Бессмертный>
<утром><в чистом поле><победил>

Слоты сами могут быть фреймами. Таким образом, возможны иерархии фреймов, сети фреймов. К фреймам применимо понятие наследования свойств. Для реализации моделей знаний с использованием фреймов хорошо подходит технология объектно-ориентированного программирования.

Логическая модель знаний представляет собой совокупность утверждений. О каждом утверждении можно сказать истинно оно или ложно. Утверждения делятся на факты и правила. Совокупность фактов представляет собой базу данных, лежащую в основе базы знаний. Правила имеют форму «ЕСЛИ А, ТО Б» (здесь есть сходство с продукционной моделью). Механизм вывода основан на аппарате математической логики (он называется исчислением предикатов первого порядка). Прикладные возможности этой модели весьма

ограничены. Логическая модель знаний лежит в основе языка Пролог.

4. *Пролог* является языком логического программирования. Как известно, в программировании для ЭВМ существует несколько различных направлений (парадигм): процедурное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. В языке Пролог реализована логическая парадигма. Однако в рамках базового курса информатики использование Пролога очень ограничено и о программировании, в полном смысле этого слова, здесь речи не идет. Пролог рассматривается лишь как средство построения несложных баз знаний логического типа. При таком подходе систему Пролог можно рассматривать как своеобразную *систему управления базами знаний* (по аналогии с СУБД). Она позволяет создавать базу знаний и обращаться к ней с запросами.

Реализации Пролога существуют для большинства компьютеров, доступных школам. Поэтому представляется возможным предлагать Пролог в качестве средства для практической работы по теме «Искусственный интеллект и моделирование знаний». Кроме того, имеется определенный методический опыт и традиции преподавания Пролога в школьной информатике. В качестве литературы по Прологу можно рекомендовать пособия [28—30].

5. В разделах 11.3—11.4 учебника раскрываются основные понятия Пролога: факт, правило, цель. На традиционном примере базы знаний о родственных связях показываются способы описания фактов и правил, способы формулировки запросов (целей). Однако за рамками учебника оказывается механизм вывода, используемый при поиске ответов на запросы (доказательстве целей). Практический опыт преподавания данной темы показывает, что если ученики не имеют никакого представления о механизме вывода Пролога,

то серьезные проблемы вызывает у них построение даже самой простой базы знаний. Рассмотрим методику объяснения работы механизма вывода Пролога на конкретных примерах.

Механизм вывода представляет собой алгоритмическую модель рассуждений человека на основе имеющихся знаний, человеческой логики. Навыки логического мышления частично присущи человеку от рождения, но в основном вырабатываются в процессе жизнедеятельности. Разгадка тайны человеческой логики — одна из древнейших научных проблем. Первый удачный опыт формального описания логики принадлежит древнегреческому философу Аристотелю (более 2,2 тысячи лет назад).

До XIX века формальная логика Аристотеля считалась непререкаемой. В середине XIX века в эту науку внесли существенный вклад английские математики Огастес Морган (правила Моргана) и Джордж Буль (Булева алгебра). Формальная логика активно развивается в XX веке.

Обсудим на конкретном примере работу механизма вывода Пролога. Рассмотрим базу знаний, которой дадим название «Знатоки». В ней содержатся факты о том, что Мария знает английский язык, информатику и музыку. Кроме того задано правило: «Иван знает то же, что знает Мария». На Прологе это записывается следующим образом:

Факты: знает(мария, английский)
 знает(мария, информатика)
 знает(мария, музыка)

Правила: знает(иван, X) :- знает(мария, X)

Здесь используются предикаты — конструкции вида: имя (аргументы). Предикат обозначает некоторое отношение между аргументами или свойство (если аргумент один).

Сформулируем цель: знает ли Иван английский язык?

?знает (иван, английский)

Это цель первого типа, на которую может быть получен альтернативный ответ «да» или «нет».

Поиск ответа заключается в доказательстве цели. Это значит, что Пролог-система на основании базы знаний пытается доказать справедливость факта, записанного в формулировке цели. Если факт доказывается, то выдается ответ «да», в противном случае — «нет».

Пролог ищет в БЗ факты и заголовки правил, сопоставимые с целью. Два предиката сопоставимы (или соответствуют друг другу), если выполняются следующие три условия:

1. Предикаты имеют одинаковые имена (побуквенное совпадение).
2. Предикаты имеют одинаковое число аргументов.
3. Аргументы, расположенные в одинаковых позициях, сопоставимы. Сопоставление аргументов осуществляется по следующим правилам:
 - 3.1. Имена конкретных объектов сопоставимы, если они совпадают;
 - 3.2. Переменная сопоставима с именем конкретного объекта;
 - 3.3. Переменная сопоставима с другой переменной;
 - 3.4. Переменная и константа сопоставимы с пустым аргументом ($\square$).

Пролог просматривает утверждения в том порядке, в каком они вводились в базу знаний (на печатной странице это соответствует просмотру сверху вниз). Поэтому сначала сравнивается цель

?знает (иван, английский)

и факт

знает (мария, английский)

Первые аргументы несопоставимы (см. правило 3.1). Следовательно, попытка сопоставить цель и факт неуспешна. Такой же результат дают сопоставления и с другими фактами БЗ. Последовательно просматривая базу, система выходит на правило:

знает(иван, X) :- знает(мария, X)

Цель и заголовок правила сопоставимы, так как выполнены условия 1, 2 и 3.2. В результате сопоставления переменная *X* *означивается*, то есть принимает значение «английский». Теперь каждому появлению переменной *X* в правиле соответствует значение «английский»:

знает(иван, английский) :
знает(мария, английский)

На следующем этапе Пролог должен проверить истинность тела правила для того, чтобы доказать или опровергнуть его голову. Следовательно, теперь нужно доказывать дополнительную цель (подцель)

?знает(мария, английский)

которая создается самим Прологом. Для доказательства подцели Пролог снова просматривает базу знаний и находит факт

знает(мария, английский)

который сопоставим с подцелью. Из доказательства подцели следует, что исходная цель доказана. Значит, на первоначально поставленный вопрос получен положительный ответ «да».

Теперь обсудим работу механизма вывода Пролога при поиске ответа на вопросы второго типа. Требуется узнать, какие дисциплины знает Мария. Соответствующая цель записывается так:

?знает(мария, X)

Пролог-система ищет все возможные значения переменной *X*, при которых доказывается цель.

Сначала у переменной цели X нет значения. В этом случае говорят, что переменная не означена. Неозначенные переменные называют свободными.

Пролог сопоставляет цель с первым фактом. Согласно правилам сопоставление дает положительный результат. После этого переменная X перестает быть свободной, она связывается с объектом «английский». Пролог выводит на экран имя этого объекта как первый результат.

Теперь необходимо найти другой объект, который могла бы обозначать переменная X . Это значит, что Пролог должен «забыть» о том, что переменная X связана с объектом «английский» и снова продолжить поиск с неозначенной переменной X . Таким образом, переменная вновь становится свободной, если цель успешно доказана.

Следующий факт, соответствующий цели:

знает (мария, информатика)

Переменная X означивается объектом «информатика». Пролог выводит новое значение переменной, «забывает» о том, что X обозначает «информатика» и снова продолжает поиск со свободной переменной X . В результате сопоставления цели и третьего факта переменная X означивается объектом «музыка».

В нашем примере нет больше информации о том, что знает Мария, поэтому Пролог завершает поиск.

Теперь рассмотрим алгоритм механизма вывода при получении ответа на вопрос к базе знаний, содержащей сложные правила с конъюнкцией.

Дополним базу знаний о Марии и Иване новой информацией. Будем считать, что Мария учится в университете и музыкальном училище. Для изучаемых курсов введем предикаты:

университет_курс(X)

музыка_курс(Y)

смысла которых заключается в следующем: курс X изучается в университете, а курс Y — в музыкальном училище.

Новая база знаний теперь выглядит так:

- знает (мария, хор)
- (1) знает (мария, сольфеджио)
- (2) знает (мария, информатика)
- (3) знает (мария, алгебра)
- знает (мария, геометрия)
- знает (мария, английский)
- знает (иван, X) : - знает
- (мария, X) , университет_курс
- (X) .
- университет_курс (информатика)
- университет_курс (алгебра)
- университет_курс (геометрия)
- университет_курс (английский)
- музыка_курс (хор)
- музыка_курс (сольфеджио)

И задана цель

? знает (иван, X)

Пролог сопоставляет цель со всеми фактами для факта «знает». Результаты всех этих сопоставлений неуспешны, так как имя объекта «Мария» несопоставимо с именем «Иван». Пролог переходит к правилу. Цель и голова правила сопоставимы, так как их переменные свободны. Теперь факты правой части правила становятся подцелями. Первой подцелью является

? знает (мария, X)

Это новая подцель, поэтому Пролог снова начинает просмотр базы знаний с первого факта для факта «знает» и находит

знает (мария, хор)

Этот факт сопоставим с подцелью. Значением переменной X становится «хор». Однако существуют дру-

гие утверждения, которые могли быть использованы для доказательства первой подцели. Поэтому Пролог устанавливает указатель отката в точку 1. С этого указателя Пролог сделает попытку найти другое решение, если вся цель окажется неуспешной.

Так как область действия переменной — это все правило, то вторая подцель есть

? университет_курс (хор)

так как переменная *X* имеет значение «хор». Все сопоставления этой подцели с фактами базы знаний неуспешны. Поэтому первая попытка доказать цель завершилась неудачей.

Пролог выполняет откат к указателю 1. Переменная *X* становится свободной из-за неуспешного вычисления цели. В точке, определяемой указателем отката, Пролог находит утверждение

знает (мария, сольфеджио)

и устанавливает указатель отката 2 на следующий факт для факта «знает». Переменная *X* принимает значение «сольфеджио». Подцель

?университет_курс (сольфеджио)

не может быть доказана, следовательно, доказательство цели снова привело к неудаче. Переменная *X* освобождается, а для доказательства цели будет сделана следующая попытка. Пролог выполняет откат в точку 2. Теперь первая подцель сопоставляется с фактом

знает (мария, информатика)

X получает значение «информатика», а указатель отката устанавливается в точку 3. Вторая подцель принимает вид:

?университет_курс (информатика)

Успешное сопоставление этой подцели доказывает цель. Следовательно, ответ на поставленный вопрос формулируется так: «Иван знает информатику». Цель

успешно доказана, поэтому переменная X становится свободной и может быть вновь означена при поиске других решений.

6. В каждой конкретной реализации системы программирования на Прологе имеются свои особенности в синтаксисе языка, в правилах построения программы. Одна из распространенных версий — система Турбо Пролог, используемая на IBM PC. Кратко опишем основные правила Турбо Пролога.

Турбо Пролог допускает использование следующих символов (алфавит):

- прописные буквы латинского алфавита от A до Z;
- строчные буквы латинского алфавита от a до z;
- цифры от 0 до 9;
- специальные символы ? - : , () " и др. (остальные специальные символы используются в особых случаях, которые здесь не рассматриваются).

Правила образования имен. Имя — последовательность букв и/или цифр, начинающаяся с буквы. Имена предикатов и объектов выбираются произвольно. Однако имена должны быть подобраны так, чтобы наилучшим образом отражать смысл тех понятий, которые они представляют. *Имена предикатов и объектов должны начинаться со строчных букв.*

Описание базы знаний на Турбо Прологе состоит из двух разделов:

- описания предикатов;
- описания утверждений: фактов и правил.

Ключевые слова `predicates`, `clauses` отмечают начало соответствующего раздела:

```
predicates
    <описание предикатов>
clauses
    <факты и правила>
```

В разделе `predicates` необходимо указать имена предикатов, количество и типы аргументов (объектов).

Объекты могут иметь числовые или символьные типы. В разделе предикатов факт know (знает) должен быть представлен следующим описанием:

```
predicates
  know(symbol, symbol)
```

Это описание означает, что оба аргумента предиката know относятся к символьному типу — symbol. Этот тип является одним из стандартных типов Пролога (табл. 17.1).

Таблица 17.1

Стандартные типы Турбо Пролога

Тип данных	Ключевое слово	Диапазон значений	Примеры использования
Символы	char	Любые символы	'a', 'b', 'c'
Целые числа	integer	от -32768 до 32767	-63 89
Действительные числа	real	от -1E-307 до 1E308	1.25E27 -56478
Строки	String	Последовательность символов	"Home"
Символические имена	Symbol	1. Имя: первый символ — строчная буква 2. Последовательность символов, заключенная в кавычки	flower "window"

Опишем на Турбо Прологе базу знаний «Знатоки» и сформулируем цель.

```
predicates
  know(symbol, symbol)
clauses
  know(mary, informatics)
  know(mary, algebra)
```

```
know(mary, geometry)
know(mary, english)
know(john, what) :- know
(mary, what)
```

Данное описание БЗ формируется вводом с клавиатуры в окне редактора. Для обозначения цели используется служебное слово `goal`. Цель может быть внутренней и внешней. Внутренние цели записываются непосредственно в текст БЗ между разделами `predicates` и `clauses`. В таком случае программа выполняется автоматически при запуске на исполнение, доказываются все цели и выполнение программы завершается.

Возможен диалоговый режим исполнения программы. В этом случае цели называются внешними. Они вводятся пользователем последовательно с клавиатуры в окно диалога. Например:

```
goal: know(X, geometry)
```

После ввода цели она обслуживается, выводится результат:

```
X=mary
```

```
X=john
```

Затем в окно диалога вводится новое приглашение: `goal:`, и система ждет ввода новой цели.

18. Введение в программирование

Разделы учебника: часть II, глава 12.

Основные цели. Познакомить учащихся с разделом информатики «Программирование». Обучить их приемам построения простых вычислительных алгоритмов и программированию алгоритмов на языке высокого уровня (Паскале). Обучить детей начальным навыкам работы с системой программирования.

Изучаемые вопросы

- Программирование как раздел информатики.
- Алгоритмы работы с величинами.
- Построение вычислительных алгоритмов разной структуры.
- Знакомство с языком программирования Паскаль.
- Назначение систем программирования.
- Освоение приемов работы с системой программирования на Паскале.

Общие методические рекомендации

Программирование — одна из самых традиционных тем в курсе информатики. Место и вес этой темы в программе курса со временем меняются. Наиболее существенный пересмотр этого вопроса произошел с переносом информатики из старших классов в базовую школу. В первых версиях обязательного минимума раздел программирования практически отсутствовал. В последних версиях — снова занял значительное место. Глава 12 «Введение в программирование» вынесена в часть II учебника, то есть предназначена для варианта углубленного изучения базового курса. Однако в связи с отмеченными выше тенденциями, изучение этой темы возможно и на минимальном уровне. В тематическом плане курса, приведенном в разделе 1.5, под нее выделено учебное время.

Обсудим вопрос о том, что такое программирование как предметная область. *Программирование — это раздел информатики, задача которого — разработка программного обеспечения ЭВМ.* В узком смысле слово «программирование» обозначает процесс разработки программы на определенном языке программирования. Разработку средств системного ПО и систем программирования принято называть *системным программированием*; разработку прикладных программ называют *прикладным программированием*. По этому принципу делят программистов на системных и прикладных, в зависимости от типа создаваемых ими программ.

В период существования ЭВМ первого, второго и третьего поколений для использования компьютера в любой области специалист должен был уметь программировать. В 1985 году, когда в школу вводился курс информатики, авторами первого школьного учебника был провозглашен лозунг «Программирование — вторая грамотность». Средства прикладного программного обеспечения стали появляться на ЭВМ третьего поколения. Их называли пакетами прикладных программ. Но лишь на ЭВМ четвертого поколения, на персональных компьютерах, прикладное ПО стало массовым и общедоступным. Теперь уже совсем не обязательно уметь программировать, знать языки программирования для того, чтобы воспользоваться компьютером. В наше время к программированию (прикладному) специалисты обращаются лишь в том случае, если отсутствует подходящее средство прикладного ПО для решения нужной задачи. Со временем такая ситуация становится все более редкой.

Возникает вопрос: для чего же тогда нужно изучать программирование в школе? Есть два ответа на этот вопрос. Во-первых, с целью фундаментализации курса информатики. Одним из фундаментальных принципов информатики является принцип программного управления работой компьютера. Понять его невозможно, не зная, что такое программа для ЭВМ. Таким образом, знакомство с программированием является элементом общеобразовательного содержания информатики. Во-вторых, с точки зрения профориентационной функции предмета. Любой школьный предмет должен выполнять и эту функцию наряду с образовательной, развивающей и воспитательной функциями. Программирование является профессиональной областью деятельности, весьма важной, современной, престижной. В большей степени профориентационная задача достигается при изучении профильного курса программирования.

Методика преподавания программирования в высшей и средней школе отработывалась многие годы. Авторы настоящего пособия имеют большой практический опыт в этой области и могут предложить некоторую универсальную методическую схему изучения программирования. В зависимости от цели и объема курса ее наполнение может быть разным. Наиболее полное — для вузовского курса программирования, самое краткое — для раздела школьного базового курса информатики. Школьные учителя могут использовать эту схему и при изучении профильного курса программирования в старших классах.

Существуют различные парадигмы программирования, и преподавание каждой из них имеет свои особенности. К основным парадигмам программирования относятся:

- *процедурное программирование* (Паскаль, Бейсик, Фортран, Си, Ассемблеры);
- *логическое программирование* (Пролог);
- *функциональное программирование* (Лисп);
- *объектно-ориентированное программирование* (Смол ток, Си++, Delphi).

В скобках приведены примеры языков программирования, в которых реализована соответствующая парадигма.

Классической, универсальной и наиболее распространенной является процедурная парадигма. Наибольшее количество существующих языков программирования относятся к этой линии. Поэтому чаще всего в учебных заведениях изучается процедурное программирование. А наиболее часто изучаемыми в школе языками программирования являются Паскаль и Бейсик. В дальнейшем под словом «программирование» мы будем подразумевать именно процедурную парадигму.

Процесс изучения и практического освоения программирования можно разделить на три части (рис. 18.1):



Рис. 18.1. Основные разделы изучения программирования

- изучение методов построения вычислительных алгоритмов;
- изучение языка программирования;
- изучение и практическое освоение определенной системы программирования.

Здесь и в дальнейшем термин «вычислительные алгоритмы» будем понимать в самом широком смысле как алгоритмы работы с величинами любых типов, ориентированные на исполнителя — ЭВМ.

Методические рекомендации по изучению алгоритмизации

Основное качество, необходимое для программиста — развитое алгоритмическое мышление. Развитие этого качества имеет не только профессиональное, но и общепедагогическое значение. Известны варианты школьного курса информатики, в которых эта задача рассматривается как главная цель предмета.

Есть две стороны в обучении алгоритмизации для ЭВМ:

- обучение структурной методике построения алгоритмов;
- обучение методам работы с величинами.

Первая задача уже ставилась и решалась в алгоритмическом разделе темы «Информация и управление». Знакомясь с программным управлением исполнителями, работающими «в обстановке», ученики осваивали методику структурного программирования. При этом никак не затрагивалось понятие величины. Однако с величинами ученики уже встречались в других те-

мах базового курса: в частности при изучении баз данных, электронных таблиц. Теперь им предстоит объединить навыки структурной алгоритмизации и навыки работы с величинами.

ЭВМ — исполнитель алгоритмов. Как известно, всякий алгоритм (программа) составляется для конкретного исполнителя, в рамках его системы команд. О каком же исполнителе идет речь в теме «программирование для ЭВМ»? Ответ очевиден: исполнителем является компьютер. Точнее говоря, исполнителем является комплекс «ЭВМ + система программирования (СП)». Программист составляет программу на том языке, на который ориентирована СП. Иногда в литературе по программированию такой комплекс называют «виртуальной ЭВМ». Например, компьютер с работающей системой программирования на Бейсике называют «Бейсик-машина»; компьютер с работающей системой программирования на Паскале называют «Паскаль-машина» и т. п. Схематически это изображено на рис. 18.2.

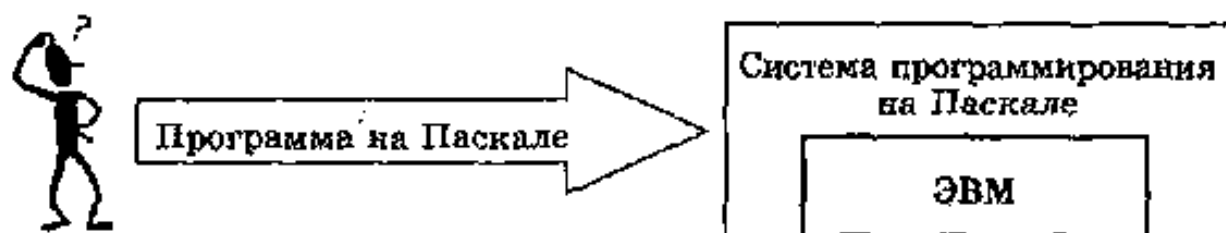


Рис. 18.2. Взаимодействие программиста с компьютером

Входным языком такого исполнителя является язык программирования Паскаль.

В разделе базового курса «Введение в программирование» необходимо продолжать ту же структурную линию, которая была заложена в алгоритмическом разделе. Поэтому при выборе языка программирования

следует отдавать предпочтение языкам структурного программирования. Наиболее подходящим из них для обучения является Паскаль.

Процесс программирования делится на три этапа:

- 1) составление алгоритма решения задачи;
- 2) составление программы на языке программирования;
- 3) отладка и тестирование программы.

Для описания алгоритмов работы с величинами следует, как и раньше, использовать блок-схемы и учебный Алгоритмический язык. Описание алгоритмов должно быть ориентировано на исполнителя со структурным входным языком независимо от того, какой язык программирования будет использоваться на следующем этапе.

Свойства величин. Теперь обсудим специфику понятия величины и методические проблемы раскрытия этого понятия.

Компьютер работает с информацией. Информация, обрабатываемая компьютерной программой, называется *данными*. *Величина* — это отдельный информационный объект, отдельная единица данных. Команды в компьютерной программе определяют действия, выполняемые над величинами.

По отношению к программе данные делятся на исходные, результаты (окончательные данные) и промежуточные данные, которые получаются в процессе вычислений (рис. 18.3).



Рис. 18.3. Уровни данных относительно программы

Например, при решении квадратного уравнения $ax^2 + bx + c = 0$ исходными данными являются коэффи-

циенты: a , b , c ; результатами — корни уравнения: x_1 , x_2 ; промежуточным данным — дискриминант уравнения: $D = b^2 - 4ac$.

Важнейшим понятием, которое должны усвоить ученики, является следующее: *всякая величина занимает свое определенное место в памяти ЭВМ*. Усвоению этого понятия эффективно способствует изучение предыдущей темы «Как работает компьютер». В результате в сознании учеников должен закрепиться образ ячейки памяти, сохраняющей величину. Термин «ячейка памяти» рекомендуется употреблять и в дальнейшем для обозначения места хранения величины.

У всякой величины имеются три основных свойства: *имя, значение и тип*. На уровне машинных команд всякая величина идентифицируется адресом ячейки памяти, в которой она хранится, а ее значение — двоичный код в этой ячейке. В алгоритмах и языках программирования величины делятся на *константы* и *переменные*. Константа — неизменная величина и в алгоритме она представляется собственным значением, например: 15, 34.7, 'k', true и пр. Переменные величины могут изменять свои значения в ходе выполнения программы и представляются символическими именами — идентификаторами, например: X, S2, cod15 и пр. Однако ученики должны знать, что и константа, и переменная занимают ячейку памяти, а значения этих величин определяются двоичным кодом в этой ячейке.

Теперь о типах величин — *типах данных*. С понятием типа данных ученики уже встречались, изучая базы данных и электронные таблицы. Это понятие является фундаментальным для программирования. Поэтому в данном разделе базового курса происходит возврат к знакомому разговору о типах, но на новом уровне.

В каждом языке программирования существует своя концепция типов данных, своя система типов. Однако в любой язык входит минимально необходимый набор основных типов данных, к которому относятся: *целый, вещественный, логический и символьный* типы.

С типом величины связаны три ее свойства: множество допустимых значений, множество допустимых операций, форма внутреннего представления. В табл. 18.1 представлены эти свойства основных типов данных.

Таблица 18.1

Тип	Значения	Операции	Внутреннее представление
Целый	Целые положительные и отрицательные числа в некотором диапазоне. Примеры: 23, -12, 387	Арифметические операции с целыми числами: +, -, ×, целочисленное деление и получение остатка от деления; операции отношения (<, >, = и др.)	Формат с фиксированной точкой
Вещественный	Любые (целые и дробные) числа в некотором диапазоне. Примеры: 2,5; -0,01; 45,0; $3,6 \times 10^9$	Арифметические операции: +, -, ×, /; операции отношения	Формат с плавающей точкой
Логический	true (истина); false (ложь)	Логические операции: И (and), ИЛИ (or), НЕ (not); операции отношения	1 бит; 1 — true; 0 — false
Символьный	Любые символы компьютерного алфавита. Примеры: 'a', '5', '+', '\$'	Операции отношения	Коды таблицы символьной кодировки. 1 символ — 1 байт

Типы констант определяются по контексту (то есть по форме записи в тексте), а типы переменных устанавливаются в описании переменных.

Есть еще один вариант классификации данных: классификация по структуре. Данные делятся на простые и структурированные. Для простых величин (их еще называют скалярными) справедливо утверждение: одна величина — одно значение. Для структурированных: одна величина — множество значений. К структурированным величинам относятся массивы, строки, множества и др. В разделе базового курса «Введение в программирование» рассматриваются только простые величины.

Действия над величинами, определяемые алгоритмом (программой), основываются на следующей иерархии понятий: операция — выражение — команда или оператор — система команд (рис. 18.4).

Операция — простейшее законченное действие над данными. Операции для основных типов данных перечислены в приведенной выше таблице.

Выражение — запись в алгоритме (программе), определяющая последовательность операций для вычисления некоторой величины.

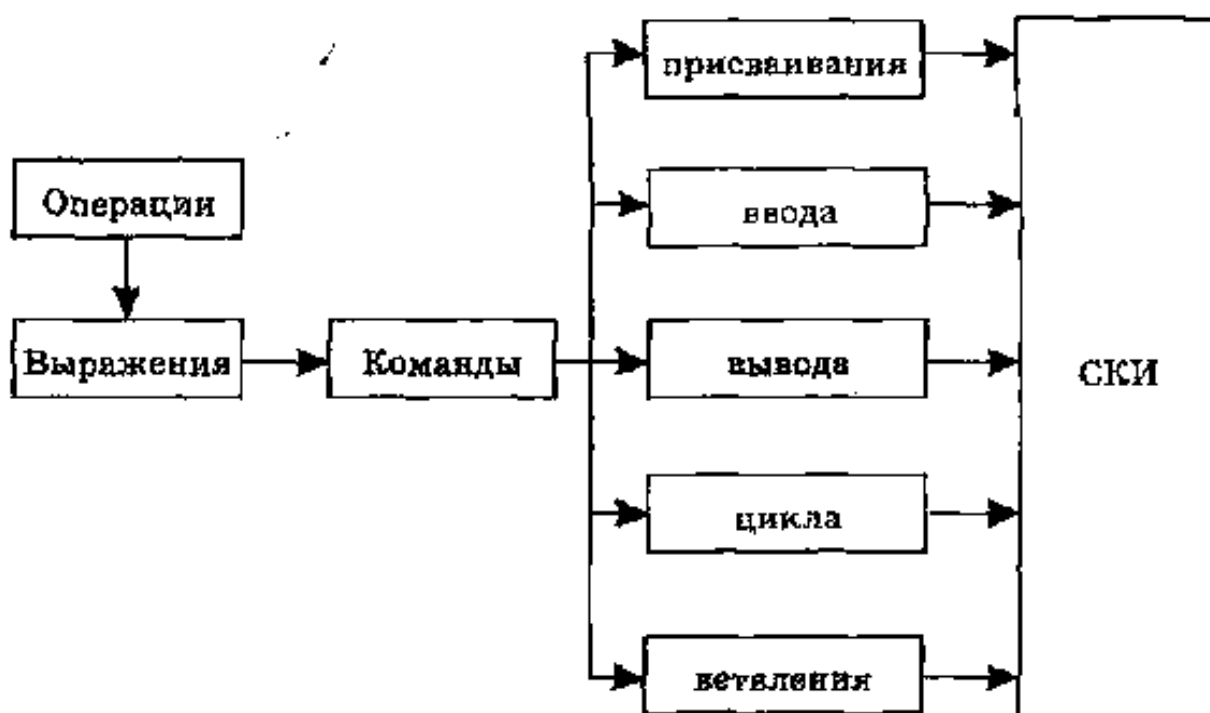


Рис. 18.4. Средства выполнения действий над величинами

Команда — входящее в запись алгоритма типовое предписание исполнителю выполнить некоторое законченное действие. Команды присваивания, ввода, вывода называются простыми командами; команды цикла и ветвления — составными или структурными командами.

В языках программирования строго определены правила записи операций, выражений, команд. Эти правила составляют синтаксис языка. При описании алгоритма в виде блок-схемы или на учебном Алгоритмическом языке строгое соблюдение синтаксических правил не является обязательным. Алгоритм программист пишет для себя, как предварительный этап работы перед последующим составлением программы. Поэтому достаточно, чтобы смысл алгоритма был понятен его автору. В то же время, в учебном процессе требуется некоторая унификация способа описания алгоритма для взаимопонимания. Однако еще раз подчеркнем, что эта унификация не так жестко формализована, как в языках программирования.

Не следует требовать от учеников строгости в описаниях алгоритмов с точностью до точки или запятой. Например, если каждая команда в алгоритме на АЯ записывается в отдельной строке, то совсем не обязательно в конце ставить точку с запятой. В качестве знака умножения можно употреблять привычные из математики точку или крестик, но можно и звездочку — характерную для языков программирования. Но следует иметь в виду, что и в описаниях алгоритмов нужно ориентироваться только на тот набор операций и команд, который имеется у исполнителя. Проще говоря, не нужно употреблять операции или функции, которых нет в используемом языке программирования. Например, если составляется алгоритм для дальнейшего программирования на Бейсике, то в нем можно использовать операцию возведения в степень в виде: X^5 или $X\wedge 5$, потому что в языке программирования есть эта операция (пишется $X\wedge 5$). Если же программа будет записываться на Паскале, в котором отсутствует опера-

ция возведения в степень, то и в алгоритме не следует ее употреблять; нужно писать так: $X * X * X * X * X$. Возведение в большую целую степень, например в 20-ю, 30-ю, следует производить циклическим умножением. Возведение в вещественную степень организуется через функции $\exp$ и $\ln$:

$$x^y = e^{y \ln x} \Rightarrow \exp(y * \ln(x)).$$

Узловыми понятиями в программировании являются понятия переменной и присваивания. О переменной уже говорилось выше. Процесс решения вычислительной задачи — это процесс последовательного изменения значений переменных. В итоге в определенных переменных получается искомый результат. *Переменная получает определенное значение в результате присваивания.* Ученикам уже известно, что на уровне машинных команд присваивание — это занесение в ячейку, отведенную под переменную, определенного значения в результате выполнения команды. Из числа команд, входящих в представленную выше СКИ, присваивание выполняют команда ввода и команда присваивания. Есть еще третий способ присваивания — передача значений через параметры подпрограмм. Но о нем мы здесь говорить не будем.

Педагогический опыт показывает, что в большинстве случаев непонимание некоторыми учениками программирования происходит от непонимания смысла присваивания. Поэтому учителям рекомендуется обратить особое внимание на этот вопрос.

Команда присваивания имеет следующий вид:

(переменная) := (выражение)

Знак «:=» надо читать как «присвоить». Это инструкция, которая обозначает такую последовательность действий:

- 1) вычислить выражение;
- 2) присвоить полученное значение переменной.

Обратите внимание учеников на то, что команда выполняется справа налево. Нельзя путать команду присваивания с математическим равенством! Особенно часто путаница возникает в тех случаях, когда в качестве знака присваивания используется знак « $=$ » и учитель читает его как «равно». В некоторых языках программирования знак « $=$ » используется как присваивание, например, в Бейсике и Си. В любом случае надо говорить «присвоить».

Ученикам, отождествляющим присваивание с равенством, совершенно непонятна такая команда: $X := X + 1$. Такого математического равенства не может быть! Смысл этой команды следует объяснять так: к значению переменной X прибавляется единица и результат присваивается этой же переменной X . Иначе говоря, данная команда увеличивает значение переменной X на единицу.

Под вводом в программировании понимается процесс передачи данных с любого внешнего устройства в оперативную память. В рамках введения в программирование можно ограничиться узким пониманием ввода как передачи данных с устройства ввода — клавиатуры в ОЗУ. В таком случае ввод выполняется компьютером совместно с человеком. По команде ввода работа процессора прерывается и происходит ожидание действий пользователя; пользователь набирает на клавиатуре вводимые данные и нажимает на клавишу <ВВОД>; значения присваиваются вводимым переменным.

Снова вернемся к вопросу об архитектуре ЭВМ — исполнителя вычислительных алгоритмов. Как известно, одним из важнейших дидактических принципов в методике обучения является принцип наглядности. За каждым изучаемым понятием в сознании ученика должен закрепиться какой-то визуальный образ. Успешность обучения алгоритмизации при использовании учебных исполнителей объясняется именно наличием таких образов (Черепашки, Робота, Кенгуренка и др.). Можно еще сказать так: архитектура учебных испол-

нителей является наглядной, понятной ученикам. Исполнителем вычислительных алгоритмов (алгоритмов работы с величинами) является компьютер. Успешность освоения программирования для ЭВМ во многом зависит от того, удастся ли учителю создать в сознании учеников наглядный образ архитектуры компьютера-исполнителя. Работа с реализованными в виде исполнителей учебными компьютерами (например, с «УК Нейман») помогает решению этой задачи. Составляя вычислительные алгоритмы, программы на языках высокого уровня, ученики в своем понимании архитектуры могут отойти от деталей адресации ячеек памяти, типов регистров процессора и т. п. подробностей, но представление об общих принципах работы ЭВМ по выполнению программы у них должно остаться.

На рис. 18.5 показано, как должен представлять себе ученик выполнение алгоритма сложения двух чисел.

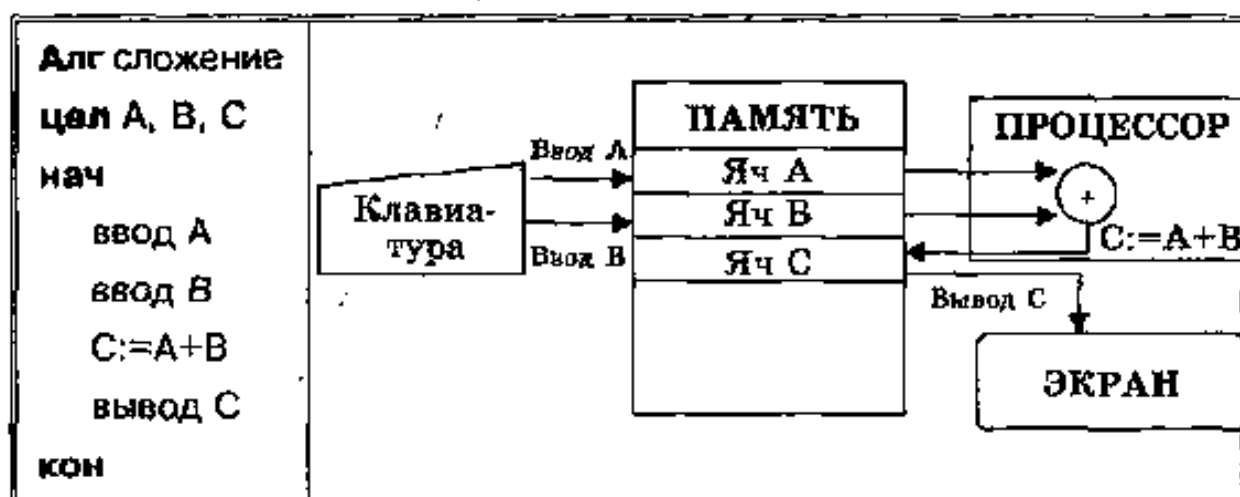


Рис. 18.5.

Исполнение компьютером вычислительного алгоритма

Эффективным методическим средством, позволяющим достичь понимания программирования, является ручная трассировка алгоритмов, которая производится путем заполнения трассировочной таблицы. В разделе 12.2 даны правила оформления трассировки алгоритма. В этом же разделе рассматривается задача обмена

значениями двух переменных величин. Это одна из базовых задач программирования, поэтому к ней стоит вернуться еще раз. Первый раз эта задача рассматривалась в главе «Как работает компьютер» на уровне ячеек, теперь — на уровне переменных.

Закреплению понятий «переменная», «присваивание» способствует выполнение заданий к разделам 12.1, 12.2.

Методические рекомендации по изучению языков программирования

Достаточно хорошо известна методика изучения языков программирования с целью практического их освоения. Эта методика опирается на структуру самого объекта изучения — языка программирования, которую можно представить схемой на рис. 18.6.

Языки программирования делятся на:

- машинно-ориентированные: автокоды, ассемблеры;
- языки программирования высокого уровня (ЯПВУ).

В настоящее время практически все программисты пользуются языками высокого уровня. Даже та-

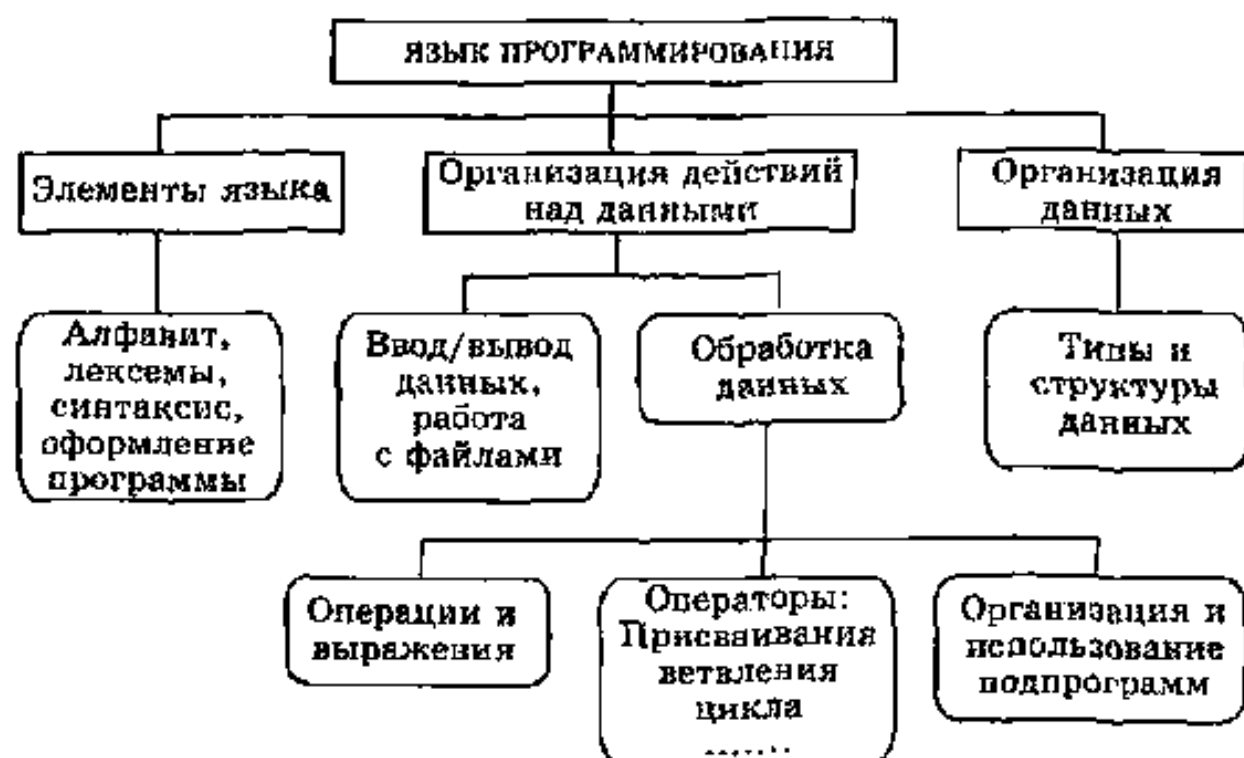


Рис. 18.6.

Структура языка программирования высокого уровня

кие системные программные продукты, как трансляторы, операционные системы и пр. составляются на языках высокого уровня (обычно для этого используют язык Си).

На любом языке программирования алгоритм решения задачи представляется через совокупность команд. Что такое команда на машинном языке, было показано на примере учебного компьютера. В языках высокого уровня одна команда определяет уже не одну операцию процессора, а, в общем случае, множество. Поэтому к командам ЯПВУ более подходит термин «оператор».

Важнейшим оператором является *оператор присваивания*. С ним ученики уже знакомились в предыдущих разделах. В ЯПВУ оператор присваивания записывается практически так же, как в Алгоритмическом языке.

В ЯПВУ одним оператором записываются целые алгоритмические структуры: ветвление, цикл. Правда, такое есть не во всех языках (например, нет в стандартном Бейсике). Языки, в которых имеются структурные операторы, принято называть *структурными языками*. К их числу относятся Паскаль и СИ.

Изучение языков программирования высокого уровня в базовом курсе должно носить только ознакомительный характер. Но использовать для этого какой-то учебный язык, учебную систему программирования совсем не обязательно. Реальные ЯПВУ можно изучать с разной степенью подробности. Освоение же работы в современных системах программирования на таких языках не вызывает больших затруднений.

Наиболее целесообразно для начального знакомства с языками программирования использовать язык Паскаль. Язык Паскаль был создан в 1971 году Никлаусом Виртом как учебный язык. Основной принцип, заложенный в нем, — поддержка структурной методики программирования. Этот же принцип лежит в основе псевдокода, который мы здесь называем Алгоритмическим языком (АЯ). По сути дела, расхождение между АЯ и Паскалем состоит в двух вещах: АЯ —

русскоязычный, Паскаль — англоязычный; синтаксис Паскаля определен строго и однозначно в отличие от сравнительно свободного синтаксиса АЯ.

Конечно, учитель может выбрать и язык Бейсик из-за привычки к нему или при отсутствии системы программирования на Паскале. Но в этом случае возникают серьезные методические проблемы: как аккуратно отразить концепцию типов данных и структурную методику программирования на Бейсике? В принципе, известно, как это делать, но для неопытного учителя это может оказаться проблемой.

Поскольку в базовом курсе в разделе «Введение в программирование» ставится только лишь цель первоначального знакомства, то строгого описания языка программирования не требуется. Основной метод, используемый в этом разделе — демонстрация языка на примерах простых программ с краткими комментариями. Некоторые понятия ученикам достаточно воспринять на интуитивном уровне. Наглядность такого языка, как Паскаль, облегчает это восприятие. Кроме того, пониманию помогает аналогия между Паскалем и русскоязычным Алгоритмическим языком. Для выполнения учениками несложных самостоятельных заданий достаточно действовать методом «по образцу».

Учитель может задуматься над проблемой: как лучше связать изучение методов построения алгоритмов работы с величинами и языка программирования.

Здесь возможны два варианта:

- 1) сначала рассматриваются всевозможные алгоритмы, для описания которых используются блок-схемы и АЯ, а затем — правила языка программирования, способы перевода уже построенных алгоритмов в программу на этом языке;

- 2) алгоритмизация и язык программирования осваиваются параллельно.

В учебнике использован второй подход. Дело в том, что теоретическое изучение алгоритмизации и программирования, оторванное от практики, малоэффек-

тивно. Желательно, чтобы ученики как можно раньше получили возможность проверять правильность своих алгоритмов, работая на компьютере. А для этого им нужно знакомиться с языком программирования, осваивать приемы работы в системе программирования. Метод последовательного изучения алгоритмизации и языка программирования приемлем лишь в безмашинном варианте.

Даже при использовании компьютера на первом этапе рекомендуется не отказываться от ручной трассировки алгоритма. Этот прием помогает ученикам «почувствовать» процесс исполнения, увидеть свои ошибки, допущенные в алгоритме. Когда же они станут более опытными программистами, например, осваивая профильный курс программирования в старших классах, тогда можно будет отказаться от ручной трассировки.

Методические рекомендации по изучению систем программирования

В данной теме, как и в предыдущих темах, связанных с информационными технологиями, следует придерживаться основного методического принципа изучения — принципа исполнителя. Выше уже говорилось о том, что теперь в качестве исполнителя рассматривается компьютер, оснащенный определенной системой программирования. Всякий исполнитель предназначен для выполнения определенного вида работы. Что это за работа? Она состоит из трех составляющих: *создание программы, отладка программы, исполнение программы*.

Точно так же, как для рассмотренных нами ранее исполнителей, в каждой конкретной системе программирования можно выделить следующие компоненты: *среда, режимы работы, система команд, данные*.

Под *средой* системы программирования (СП), как и раньше, будем понимать «обстановку» на экране (системную оболочку), в которой работает пользователь

СП. Здесь учитель должен описать и показать ученикам оболочку именно той системы программирования, с которой намерен работать в данной теме.

Определенная стандартизация оболочек систем программирования произошла с появлением турбо-систем фирмы Borland. Для таких сред характерно наличие на экране двух компонент: окна редактора, в котором формируется текст программы и меню команд переключения режимов работы системы.

Характерными *режимами работы систем* программирования являются:

- режим редактирования текста программы;
- режим компиляции;
- режим исполнения;
- режим работы с файлами;
- режим помощи;
- режим отладки программы.

Режим редактирования обычно устанавливается автоматически при инициализации работы системы. При этом работает встроенный в систему текстовый редактор. В принципе, текст программы можно подготовить и с помощью другого редактора, формирующего текстовые файлы. Но обычно программисты предпочитают пользоваться собственным редактором системы.

Освоить работу с текстовым редактором СП ученикам помогут навыки, полученные при изучении темы «Текстовая информация и компьютер».

Режим компиляции существует у систем, обслуживающих компилируемые языки (Паскаль, СИ, Фортран и др.). Результатом компиляции является исполнимая программа, то есть программа на языке машинных команд.

В некоторых случаях получение исполнимой программы происходит в два этапа: собственно компиляции и редактирования связей. Хотя учитель должен понимать смысл этих процедур, в базовом курсе, при объяснении ученикам эти вопросы можно подробно не комментировать.

Режим исполнения. В компилирующих системах в этом режиме выполняется полученная после трансляции программа в машинных командах. Интерпретатор непосредственно сам исполняет программу на ЯПВУ. Так, например, работает Бейсик-система. Обычно в том и в другом случае исполнение программы начинается по команде RUN.

Режим работы с файлами. В файлах на внешних носителях система хранит тексты программ на исходном языке; программы, полученные в результате трансляции; исходные данные и конечные результаты. В файловом режиме выполняются традиционные операции: сохранить информацию в файле, прочитать информацию из файла в оперативную память, именовать файл и др. К этому же режиму относится команда вывода содержимого окна редактора на печать, поскольку печать трактуется как вывод информации в файл, связанный с принтером.

Режим помощи позволяет программисту получить подсказку на экране, помогающую ему в работе как с системой, так и с языком программирования.

Режим отладки. Этот режим чаще всего реализован в развитых СП на профессиональных компьютерах. В режиме отладки можно производить трассировку, пошаговое исполнение программы; можно следить за изменением определенных величин; назначать остановку исполнения программы в определенном месте или при определенном условии. Режим отладки предоставляет программисту удобные средства для поиска алгоритмических ошибок в программе.

Следуя уже знакомой методической схеме, рассказывая про режимы работы СП, учитель должен рассказать о системе команд, используемой в каждом из режимов.

Для системы программирования *данными* являются файлы с текстами программ, с исходной и конечной информацией, связанной с решаемой задачей.

Тема базового курса «Введение в программирование» завершается изучением вопросов, изложенных в разделе 12.8 «О языках программирования и трансляторах». Основные вопросы, которые затрагиваются в этом разделе: понятие об уровнях языков программирования, понятие о трансляторах и способах трансляции. В учебнике этот материал носит теоретический характер. Однако при наличии дополнительного учебного времени, а также программы-исполнителя «УК Нейман», изучение вопросов об уровнях языков программирования, о трансляции могут быть поддержаны практикой.

Заметим в конце, что концепции базисности настоящего курса в разделе программирования в наибольшей степени соответствовало бы изучение того материала, который приведен ниже в дополнительном разделе. Этот материал можно использовать как в рамках углубленного базового курса, так и в профильном курсе программирования. Его изучение должно предшествовать освоению программирования на каком-либо языке высокого уровня. Знания программирования в таком комплексе действительно можно будет назвать базовыми, фундаментальными.

Методические рекомендации по решению задач

Пример 1

[5]: часть II, раздел 12.1, задание 9. *Вместо многократно вписать в алгоритм несколько команд присваивания, в результате чего должен получиться алгоритм возведения в 4-ю степень введенного числа (дополнительных переменных кроме A не использовать):*

ввод A . . . вывод A

Эти команды следующие:

Ввод A

$A := A \times A$ A^2

$A := A \times A$ A^4

Вывод A

Пример 2

[5]: часть II, раздел 12.2, задание 7. *Написать на АЯ алгоритм вычисления y по формуле*

$$Y = (1 - X^2 + 5X^4)^2,$$

где X — данное целое число. Учесть следующие ограничения: 1) в арифметических выражениях можно использовать только операции сложения, вычитания и умножения; 2) выражение может содержать только одну арифметическую операцию. Выполнить трассировку алгоритма при $X = 2$.

Учениками могут быть построены разные варианты алгоритма решения этой задачи. Для сохранения промежуточных результатов можно использовать дополнительные переменные. Обсуждая задачу, полезно рассмотреть вариант алгоритма, при котором не понадобится дополнительных переменных, то есть можно обойтись всего двумя переменными x и y . Тем самым достигается экономия памяти компьютера. Вот как выглядит этот алгоритм и его трассировка:

Команда	X	Y
Ввод X	2	
$X := X \times X$	4	
$Y := 1 - X$		-3
$X := X \times X$	16	
$X := 5 \times X$	80	
$Y := Y + X$		77
$Y := Y \times Y$	5929	
Вывод Y	5929	

Пример 3

[5]: часть II, раздел 12.2, задание 8. *Пользуясь ограничениями предыдущей задачи, написать наиболее короткие алгоритмы вычисления:*

а) $Y = X^3$; б) $Y = X^{10}$; в) $Y = X^{15}$; г) $Y = X^{19}$.

Постараться использовать минимальное количество дополнительных переменных. Выполнить трассировку алгоритмов.

Выполним вариант г) задания. Построим алгоритм, учитывая равенство $X^{19} = X^{16} \cdot X^3$. Значения X^k , где $k = 2^n$, быстро вычисляются n -кратным умножением переменной самой на себя. Алгоритм:

Ввод X	
$Y := X \times X$	X^2
$Z := Y \times Y$	X^4
$Z := Z \times Z$	X^8
$Z := Z \times Z$	X^{16}
$Z := Z \times Y$	X^{18}
$Z := Z \times X$	X^{19}
Вывод Z	

Пример 4

[5], часть II, раздел 12.2, задание 9. Записать алгоритм циклического обмена значениями трех переменных A, B, C . Схема циклического обмена: $A \rightarrow B \rightarrow C \rightarrow A$

Например, если до обмена было: $A = 1, B = 2, C = 3$, то после обмена должно стать: $A = 3, B = 1, C = 2$. Выполнить трассировку.

Эта задача также рассматривалась в предыдущем разделе на уровне языка машинных команд. Идея прежняя: для обмена значениями требуется дополнительная переменная, обозначим ее X . Вот как будет выглядеть алгоритм и его трассировка:

Команда	A	B	C	X
Ввод A, B, C	1	2	3	
$X := C$				3
$C := B$			2	
$B := A$		1		
$A := X$	3			
Вывод A, B, C	3	1	2	

Обучение составлению алгоритмов работы с величинами должно проводиться на примерах типовых задач с постепенным усложнением структуры алгоритмов. Такая последовательность задач рассматривается в теоретическом материале главы 12 части II учебника и предлагается в списке заданий для самостоятельного выполнения. По признаку алгоритмической структуры их можно классифицировать так:

- *линейные алгоритмы*: вычисления по формулам, всевозможные пересылки значений переменных;
- *ветвящиеся алгоритмы*: поиск наибольшего или наименьшего значений из нескольких данных; сортировка двух-трех значений; диалог с ветвлениями;
- *циклические алгоритмы*: вычисление сумм и произведений числовых последовательностей; циклический ввод данных с последовательной обработкой.

Изучение языка программирования Паскаль происходит в контексте решаемых задач, то есть новые средства языка вводятся по мере необходимости в них для решения очередного типа задач.

Рассмотрим решение некоторых задач, предлагаемых в учебнике.

Пример 5

[5]: часть II, глава 12, раздел 12.4, задание 8; раздел 12.5, задание 7. *Составить алгоритм, по которому на компьютере будет происходить следующее: в переменную S вводится возраст Саши, в переменную M вводится возраст Маши. В качестве результата на экран выводится фраза «Саша старше Маши» или «Маша старше Саши», или: «Саша и Маша ровесники». Написать программу на Паскале по этому алгоритму.*

Алгоритм имеет структуру двух вложенных полных ветвлений. При переходе от Алгоритмического

языка к Паскалю обратите особое внимание на необходимость точного соблюдения правил синтаксиса языка: точки с запятой в конце операторов, параметры ввода и вывода в круглых скобках и пр.

алг САША-МАША	Program Sasha&Masha;
вещ S, M	var S, M : real;
нач вывод 'Возраст Саши:'	Begin write ('Возраст Саши:');
ввод S	readln (S);
вывод 'Возраст Маши:'	write ('Возраст Маши:');
ввод M	readln (M);
если S>M	if S>M
то вывод 'Саша старше Маши'	then write ('Саша старше Маши')
иначе если S=M	else if S=M
то вывод 'Саша и Мама ровесники'	then write ('Саша и Мама
иначе вывод 'Мама старше Саши'	ровесники') else write ('Мама
кв	старше
кв	Саши')
кон	End.

Пример 6

[5]: часть II, глава 12, раздел 12.4, задание 9; раздел 12.5, задание 7. Составить алгоритм упорядочения значений трех переменных по возрастанию, то есть при любых исходных значениях A, B, C отсортировать их так, чтобы стало: $A \leq B \leq C$. Написать программу на Паскале по этому алгоритму.

В разделе 12.4 рассмотрен алгоритм упорядочения значений двух переменных. Для решения данной задачи требуется трижды применить этот алгоритм: упорядочить A и B , упорядочить B и C (после этого максимальное значение попадет в C), еще раз упорядочить A и B . Таким образом, алгоритм будет иметь структуру трех последовательных неполных ветвлений.

Алг СОРТИРОВКА-3	Program SORT_3;
вещ A, B, C, X	var A, B, C, X: real;
нач	begin
ввод A, B, C	readln(A, B, C);
если A>B	if A>B
то	then begin
X:=A; A:=B; B:=X	X:=A; A:=B; B:=X
кв	end;
если B>C	if B>C
то	then begin
X:=B; B:=C; C:=X	X:=B; B:=C; C:=X
кв	end;
если A>B	if A>B
то	then begin
X:=A; A:=B; B:=X	X:=A; A:=B; B:=X
кв	end;
вывод A, B, C	write(A,B,C)
кон	end.

По поводу рассмотренной задачи сделаем следующее замечание. Это задача, которая выводит на идею использования вспомогательных алгоритмов. Упорядочение трех переменных производится трехкратным применением алгоритма упорядочения двух переменных, поэтому разумно этот алгоритм оформить как вспомогательный. Программы, реализующие вспомогательные алгоритмы, называются подпрограммами. В Паскале есть два вида подпрограмм: подпрограммы-процедуры и подпрограммы-функции. В данной задаче можно использовать процедуру.

В главе 12 части II учебника подпрограммы не рассматриваются. Это объясняется ограниченностью учебного времени, которое можно выделить в базовом курсе на тему «Введение в программирование». При наличии дополнительного времени рекомендуется дать ученикам представление о подпрограммах. Рассмотренная здесь задача может быть использована как опорная для раскрытия этой темы. Приведем пример программы сортировки значений трех переменных с использовани-

ем процедуры (правила работы с процедурами смотрите в учебниках по Паскалю).

```

Program SOR_3;
var A, B, C : real;
Procedure SOR2(var X, Y : real);
var Z : real;
begin
    Z:=X; X:=Y; Y:=Z
end;
begin
    readln(A,B,C);
    SOR2(A,B);
    SOR2(B,C);
    SOR2(A,B);
    writeln(A,B,C)
end.

```

Пример 7

[5]: часть II, глава 12, раздел 12.7, задание 8. Дано вещественное число X и натуральное N . Составить алгоритм вычисления X^N . Написать программу на Паскале.

В Паскале нет операции возведения в степень. Если показатель степени — целое положительное число, то возводить в степень нужно путем N -кратного умножения основания самого на себя. Реализуется это циклическим алгоритмом.

алг Степень

цел N , i ; вещ X

нач ввод N

Ввод X

$I:=1$; $Y:=1$;

пока $i \leq N$, повторять

нц

$Y:=Y*X$

$I:=i+1$

кц

Вывод Y

кон

Program Power;

var N, i : integer; X : real;

begin readln(N);

readln(X);

$i:=1$; $Y:=1$;

while $i \leq N$ do

begin

$Y:=Y*X$;

$i:=i+1$

end;

Write(Y)

end.

При выполнении трассировки этого алгоритма обязательно проверьте правильность его работы при $N = 0$. Как известно из математики, $X^0 = 1$. Трассировка доказывает, что и в этом случае алгоритм будет давать правильный результат.

Замечания по поводу заданий 10, 11 к разделу 12.7. Для выполнения этих заданий используется алгоритм Евклида, описанный на АЯ и реализованный на Паскале в данном разделе. Как и в задаче с сортировкой трех переменных, здесь удобно использовать подпрограммы. Рекомендуем, при углубленном изучении темы, сделать это. Алгоритм Евклида может быть реализован как подпрограммой-процедурой, так и подпрограммой-функцией.

В конце рассмотрим еще один тип задач, связанный с последовательным вводом значений и их обработкой в одном цикле. Такие задачи приводятся в задачнике-практикуме.

Пример 8

Задачник-практикум: раздел 4.3, задача 33. *Последовательно вводятся N целых чисел. Найти максимальное из них.*

В программировании часто приходится решать задачу поиска максимального (минимального) значения в числовом массиве. В разделе «Введение в программирование» не рассматриваются структурированные данные, в том числе массивы. Однако эту задачу можно решить и без использования массива, ограничившись простыми переменными. Для этого ввод данных и обработку надо совместить в одном цикле. Вот как это делается.

Алг максимум

цел N, i, X, MAX

нач

вывод 'введите N'; ввод N

вывод 'введите X'; ввод X

MAX:=X; i:=1

пока i<N, повторять

нц

вывод 'Введите X';

ввод X

если X>MAX

то MAX:=X

кв

i:=i+1

кц

вывод MAX

кон

Program maximum

var N, i, X, MAX: integer;

begin

write('введите N');

readln(N);

write('введите X');

readln(X);

MAX:=X; i:=1;

while i<N do

begin

write('Введите X');

readln(X);

if X>MAX

then MAX:=X;

i:=i+1

end;

writeln(MAX)

end.

Дополнительный материал

Программирование для учебного компьютера «УК Нейман»

Работа с исполнителем «УК Нейман» может происходить на трех языках: языке машинных команд, автокоде и Паскале. Правда, Паскаль сокращен до минимального уровня, позволяющего работать только с целыми числами, поэтому ему дано название «мини-Паскаль». Рассмотрим приемы программирования на языке машинных команд (ЯМК) и на автокоде «УК Нейман». Делать это будем на примерах конкретных задач.

Программирование на ЯМК

Задача 1

Определить, к какому веку относится данный год новой эры.

На примере этой задачи продемонстрируем все этапы работы, выполняемые при разработке программы решения содержательной задачи средствами программирования. Таких этапов пять:

- 1) постановка задачи;
- 2) математическая формализация задачи;
- 3) построение алгоритма;
- 4) составление программы;
- 5) отладка и тестирование.

Первый этап: постановка задачи. Исходным данным является натуральное число, обозначающее год новой эры (например 1262 или 1997). Результат — натуральное число, равное веку, к которому относится этот год (например, 13, 20).

Введем следующие обозначения для переменных: G — год, V — век. Следовательно:

Дано: G

Найти: V

Второй этап: математическая формализация. В данном случае нужно получить формулу для вычисления величины V по известному значению G . Формула такая:

$$V = (G + 99) \operatorname{div} 100.$$

В ней используется операция целочисленного деления, знак которой задан словом div (от английского *division* — деление). Эта операция применяется только к целым операндам. Результат $X \operatorname{div} Y$ равен целой части частного от деления X на Y . Например:

$$\begin{aligned} 5 \operatorname{div} 2 &= 2 \\ 2 \operatorname{div} 5 &= 0 \\ 20 \operatorname{div} 4 &= 5 \\ 20 \operatorname{div} 3 &= 6 \end{aligned}$$

Проверим данную формулу. Пусть $G = 1262$. Тогда:
 $V = (1262 + 99) \operatorname{div} 100 = 1361 \operatorname{div} 100 = 13.$

Получилось что 1262 год относится к 13 веку, что, очевидно, верно.

Второй тест: $G = 1997$. Тогда:

$$V = (1997 + 99) \operatorname{div} 100 = 2096 \operatorname{div} 100 = 20.$$

Получился 20 век. Тоже верно. Проверим, правильно ли по этой формуле происходит смена века. Как известно, 2000 год относится к 20 веку, а 2001 — уже к 21-му.

$$V = (2000 + 99) \operatorname{div} 100 = 2099 \operatorname{div} 100 = 20$$

$$V = (2001 + 99) \operatorname{div} 100 = 2100 \operatorname{div} 100 = 21.$$

Все правильно!

Третий этап: составление алгоритма. Опишем алгоритм решения задачи. Вот как выглядит его описание на Алгоритмическом языке.

```

алг ЗАДАЧА 1
цел G, V
нач  ввод G
    G:=G+99
    V:=G div 100
    вывод V
кон

```

Составление программы на ЯМК начинается с **распределения памяти под данные**. В нашем алгоритме имеются две переменные величины: G , V и две константы: 99, 100. Для хранения каждой величины нужно выделить ячейку, которую можно выбрать произвольно, но с таким расчетом, чтобы она не перекрывалась программой. Например, распределим память так:

величина	99	100	G	V
ячейка	14	18	1C	20.

Следующий шаг работы — **запись программы на ЯМК**, то есть непосредственное программирование. Поскольку у нас уже есть детальный алгоритм, то программирование превращается в формальную кодировку, когда каждой команде алгоритма ставится в соответствие команда на ЯМК. Система команд «УК Нейман» приведена в разделе 16. В результате ее использования получается следующая программа:

Адрес	КОП	A1	A2	A3	Пояснения
00	00	FC	00	1C	ввод G
04	01	1C	14	1C	$G := G + 99$
08	04	1C	18	20	$V := G \text{ div } 100$
0C	00	20	00	FC	вывод V
10	77	00	00	00	стоп
14	00	00	00	63	константа 99
18	00	00	00	64	константа 100

Программа записана на бланке. Теперь ее можно вводить в компьютер. Войдем в режим ввода программы. Ввод всегда автоматически начинается с ячейки 00. Обратите внимание на то, что константы вводятся в память наряду с командами программы. Значения кон-

стант определяются во время ввода программы. Значения переменных определяются в ходе выполнения программы.

Последние этапы работы над программой — *отладка и тестирование*. Отладкой в программировании называют поиск и исправление ошибок в программе. Тестирование — это проверка правильности работы программы на тестовых вариантах исходных данных.

При программировании на ЯМК эти две задачи объединяются. Программист должен придумать тест, то есть такой вариант исходных данных, для которого ему известен результат. И если при выполнении программы на ЭВМ с этими данными будет получен ожидаемый результат, то, значит, программа правильная. Если же получен другой результат или вообще исполнение программы не доходит до конца, значит, в ней есть ошибки. Программист должен их найти, исправить и снова повторить тестирование.

Выберем следующий вариант исходных данных:

$$G = 1262.$$

Перейдем в режим исполнения программы и отдадим команду «исполнить». Первая команда требует ввода данных с клавиатуры. Введем выбранное тестовое значение в шестнадцатеричной форме ($1262_{10} = 4EE_{16}$):

00 00 04 EE <ВВОД>

После вычислений, которые произойдут мгновенно, на индикаторе появится:

00 00 00 0D

Перевод в десятичную систему дает:

$$D_{16} = 13_{10}.$$

Следовательно, программа работает правильно.

Следует заметить, что один тест не всегда доказывает правильность программы. Для исчерпывающего тестирования должна быть придумана система тестов. И чем сложнее программа, тем сложнее система тестов,

ее проверяющая. Проверка данной программы на других значениях (например, 1997, 2000, 2001) доказывает ее правильность.

Задача 2

Сколько различных N -буквенных слов можно составить путем перестановок данных N букв?

Алгоритм решения этой задачи рассмотрен в учебнике в разделе 12.6. Задача сводится к вычислению факториала целого числа N .

Дано: N Расчетная формула:

Найти: F $F = N!$

Алгоритм решения этой задачи с использованием структуры «цикл-пока» имеет вид:

```

алг ЗАДАЧА 3
цел F, M, R
нач ввод N
  F:=1
  R:=1
  пока R ≤ N
  ил
    F:=F * R
    R:=R + 1
кц
вывод F
кон

```

Распределение памяти. В программе будут использованы одна константа (1), три переменные (N , F , R) и еще одна дополнительная переменная (Q), которая в алгоритме не отражена:

Величина	1	N	F	R	Q
Ячейка	28	2С	30	34	38

Составление программы

АДРЕС	КОП	A1	A2	A3	Пояснения
00	00	FC	00	2C	ввод N
04	00	28	00	30	$F := 1$
08	00	28	00	34	$R := 1$
0C	02	34	2C	38	$Q := R - N$
10	0A	00	00	20	при $Q > 0$ идти к 20
14	03	30	34	30	$F := F * R$
18	01	34	28	34	$R := R + 1$
1C	0B	00	00	0C	идти к 0C
20	00	30	00	FC	вывод F
24	77	00	00	00	останов
28	00	00	00	01	константа 1

В этой программе кроме уже знакомых команд используются две команды управления.

Команда безусловного перехода имеет вид:

0B — — A3

Эта команда производит передачу управления на команду программы по адресу A3. Содержимое A1 и A2 значения не имеют.

В нашей программе команда безусловного перехода расположена в ячейке 1C. Ее выполнение приводит к тому, что следующей будет выполняться команда в ячейке 0C. Таким образом происходит возврат управления к началу цикла.

Команда условного перехода:

0A — — A3

Эта команда передает управление к ячейке программы A3, если значение регистра-признака W равно 1. Если $W = 0$, то управление перейдет к следующей команде.

Рассмотрим другой вариант алгоритма решения этой же задачи, в котором используется циклическая структура «цикл-до», или цикл с постусловием. Общий вид структуры такой:

повторять <тело цикла >
до <условие окончания цикла>

Повторное выполнение тела цикла заканчивается, когда условие окончания станет истинным. Следующий алгоритм вычисляет произведение в обратном порядке сомножителей: $N! = N \times N - 1 \times \dots \times 3 \times 2 \times 1$.

```

алг ЗАДАЧА 3
цел F, N
нач  ввод N
    F:=1
        повторять
            F:=F * N
            N:=N - 1
        до N=0
    вывод F
кон
  
```

В программе будут использоваться всего две переменных и константа 1. Выделим под них память:

Величина	1	F	N
Ячейка	1C	20	24

Запишем программу на ЯМК:

Адрес	КОП	A1	A2	A3	Пояснения
00	00	FC	00	24	ввод N
04	00	1C	00	20	F:=1
08	03	20	24	20	F:=F * N
0C	02	24	1C	24	N:=N - 1
10	0A	00	00	08	при N > 0 идти к 08
14	0	20	00	FC	вывод F
18	77	00	00	00	останов
1C	00	00	00	01	константа 1

Обратите внимание на то, что в этой программе не понадобилась команда безусловного перехода. Не понадобилось также введения дополнительной переменной Q . Управлением цикла занимается команда условного перехода в ячейке с адресом 10. Пока значение $N > 0$, эта команда передает управление на ячейку 08. Когда в результате выполнения команды в ячейке 08 получится нуль, признак W станет равным нулю и цикл закончит работу. Управление передастся на ячейку 14.

Второй вариант программы вместе с данными занимает 10 ячеек, тогда как первый вариант — 15 ячеек.

В качестве последнего примера программирования на ЯМК рассмотрим задачу нахождения наибольшего общего делителя двух чисел по алгоритму Евклида. Математическая суть метода и алгоритм описаны в разделе 12.7 учебника.

Задача 3

Определить наибольший общий делитель двух натуральных чисел.

Дано: M, N

Найти: $\text{НОД}(M, N)$

Составим алгоритм, отличный от того варианта, который приведен в учебнике. В новом алгоритме будет использоваться структура цикл с постусловием. На предыдущем примере было показано, что цикл с постусловием программировать на ЯМК удобнее.

```

алг ЗАДАЧА 4
цел M, N, X
нач   ввод M
      ввод N
      повторять
        если N > M
        то X := M
           M := N
           N := X
      кв
      M := M - N
до M = 0
вывод N

```

кон

Идея состоит в том, что в переменной M все время должно находиться большее значение из двух. Реализуется это с помощью ветвления: если $M \geq N$, то значения переменных не меняются, иначе переменные обмениваются значениями.

Распределение памяти. В программе используются переменные M , N и вспомогательная переменная X :

Величина	M	N	X
Ячейка	40	44	48

Программа на ЯМК:

Адрес	КОП	A1	A2	A3	Пояснения
00	00	FC	00	40	ввод M
04	00	FC	00	44	ввод N
08	02	40	44	48	$X := M - N$
0C	0A	00	00	1C	при $M = N$ идти к 1C
10	00	40	00	48	$X := M$
14	00	44	00	40	$M := N$
18	00	48	00	44	$N := X$
1C	02	40	44	40	$M := M - N$
20	0A	00	00	08	при $M > 0$ идти к 08
24	00	44	00	FC	вывод N
28	77	00	00	00	останов

Автокод для учебного компьютера

Одним из средств программного обеспечения «УК Нейман» является система программирования на Автокоде. Опишем входной язык этой системы — язык Автокод. *Описать язык — это значит определить правила представления данных и правила представления операций над данными — команд программы.*

Представление данных. На языке машинных команд данные (переменные и константы) обозначаются адресами ячеек. На автокоде константы записываются в виде десятичных чисел, заключенных в угловые скобки.

Например:

<5>, <-12>, <3245>.

Переменные обозначаются символическими именами — *идентификаторами*. Идентификатор может содержать от 1 до 6 символов. Этими символами могут быть латинские буквы и цифры. Первым символом обязательно должна быть буква. Вот несколько примеров идентификаторов:

A, G, X5, BIS12, SIZE.

Следующие примеры содержат ошибочные идентификаторы и пояснения ошибок:

Ф — русская буква,

4D — первый символ — цифра,

dimension — больше шести символов,

S+5 — недопустимый символ «+»,

mov — зарезервированное в автокоде слово (команда пересылки).

Команды автокода. Система команд Автокода ориентирована на систему команд процессора «УК Нейман». Иначе говоря, *каждой команде ЯМК соответствует команда автокода.*

Формат любой трехместной (арифметической) команды следующий:

<мнемокод> <1-й операнд>, <2-й операнд>, <результат>

Операнды могут быть переменными или константами, результат — всегда переменная. Мнемокод — это мнемоническое (буквенное) отображение кода операции. Соответствие между кодами операций на языке машинных команд и мнемокодами автокода показано в табл. 18.1.

Таблица 18.1

КОП	Мнемокод	Операция
00	MOV	Пересылка
01	ADD	Сложение
02	SUB	Вычитание
03	MUL	Умножение
04	DIV	Деление нацело
05	MOD	Остаток от деления
0A	IFGO	Условный переход
0B	GO	Безусловный переход
77	STOP	Останов программы
FF	NOP	Пустая операция

Мнемокоды представляют собой английские слова или сокращения английских слов.

MOV: move — двигать, перемещать;

ADD: add — прибавлять, складывать;

SUB: subtract — вычитать;

MUL: multiply — умножать;

DIV: divide — делить;

MOD: modulo operation — операция по модулю;

GO: go — идти;

IFGO: if ... go — если ... идти;

STOP: stop — стоять;

NOP: no operation — нет операции;

Вот несколько примеров команд на автокоде (справа записаны соответствующие команды на Алгоритмическом языке):

ADD A, B, C C := A + B;

MUL C, <3>, D D := 3 * C;

MOV D, X X := D;

Программирование на автокоде. Линейные программы для «УК Нейман» состоят из команд арифме-

тических операций, пересылки, ввода и вывода. На языке машинных команд ввод и вывод программируются командой пересылки значений через буферную ячейку FC. Однако в программах на автокоде прямые адреса ячеек памяти не используются. Чтобы не нарушать этот принцип, определим специальные команды ввода и вывода:

INPUT < переменная> — команда ввода;
OUTPUT <переменная> — команда вывода.

Например:

INPUT A — ввод значения переменной A;
OUTPUT X — вывод на экран значения переменной X.

Этапы работы над программой на автокоде следующие:

1. Составление программы.
2. Трансляция и синтаксическая отладка.
3. Тестирование и отладка алгоритма.

По сравнению с ЯМК, исчез этап распределения памяти. Зато появились своеобразные этапы 2 и 3, с которыми подробно разберемся ниже.

Для освоения программирования на автокоде рассмотрим те же задачи, с которыми мы уже работали на ЯМК.

Задача 1

Определить, к какому веку новой эры (V) относится данный год (G).

Дано: G

Расчетная формула:

Найти: V

$V = (G + 99) \text{ div } 100$

Воспользуемся готовым алгоритмом и напишем программу на автокоде. Программа расположена в левом столбце таблицы. О содержании правой половины таблицы речь пойдет позже.

Автокод		Результат трансляции				
Команды	Пояснения	Адрес	КОП	A1	A2	A3
INPUT G	ввод G	00	00	FC	00	14
ADD G, <99>, G	$G := G + 99$	04	01	14	18	14
DIV G, <100>, V	$V := G \text{ div } 100$	08	04	14	1C	20
OUTPUT V	вывод V	0C	00	20	00	FC
STOP	останов	10	77	00	00	00
		14	переменная G			
		18	константа 99			
		1C	константа 100			
		20	переменная V			

Команды записываются друг за другом в том порядке, в каком они будут выполняться. В каждой команде мнемокод отделяется от операндов пробелами (минимум один пробел). Операнды отделяются друг от друга запятыми. Запись каждой команды начинается с новой строки.

Безусловно, такую программу легче и писать, и читать, чем программу на ЯМК. На автокоде меньше вероятность допустить ошибку, чем на ЯМК. И все-таки ошибки возможны. Можно, например, неправильно записать мнемокод, идентификатор, пропустить запятую, использовать недопустимый символ и пр. Ошибки такого рода, связанные с нарушением правил записи программ на языке программирования, называются *синтаксическими ошибками* (подробнее об этом позже).

Однако возможны ошибки и в алгоритме. Алгоритмические ошибки транслятор обнаружить не может. Их должен найти и исправить программист на стадии тестирования. О тестировании мы уже говорили раньше.

При программировании циклов и ветвлений используются команды условного (IFGO) и безусловного (GO) переходов. Вспомним, что в соответствующих им командах на ЯМК (0A и 0B) указывается адрес ячейки, к которой производится переход. Но в программе на автокоде не должно быть адресов ячеек.

Для организации переходов в автокоде применяются метки. Метка представляет собой идентификатор, начинающийся с буквы L (l). Метка записывается перед мнемокодом команды и отделяется от него двоеточием.

Например:

```
L1: ADD X,Y,Z  
L15: MUL <2>,F,S
```

Следует ставить метки лишь в тех командах, к которым в программе производится переход. Форматы команд перехода следующие:

```
IFGO метка>  
GO <метка>
```

Например:

```
IFGO L1  
GO L15
```

Их исполнение аналогично соответствующим командам ЯМК.

Задача 2

Сколько различных N -буквенных слов можно составить путем перестановок данных N букв?

Дано: N Расчетная формула:

Найти: F $F = N!$

Составим программу в соответствии с приведенным выше алгоритмом, использующим структуру «цикл с постусловием»:

Программа	Пояснения
INPUT D	ввод N
MOV <1>, F	$F := 1$
L1: MUL F, N, F	$F := F \times N$
SUB N, <1>, N	$N := N - 1$
IFGO L1	при $N > 0$ идти к L1
OUTPUT F	вывод F
STOP	останов

Метка L1 отмечает начало цикла. Команда IFGO L1 производит переход к помеченной команде в тех случаях, когда цикл следует повторять.

Задача 3

Определить наибольший общий делитель двух натуральных чисел.

Дано: M, N

Найти: НОД(M, N)

Программа решения задачи реализует алгоритм Евклида. Составим программу в соответствии с приведенным выше алгоритмом, использующим цикл с постусловием:

Программа	Пояснения	Результат трансляции	
INPUT M	ввод M	00	00 FC 00 2C
INPUT N	ввод N	04	00 FC 00 30
L10: SUB M, N, X	$X := M - N$	08	02 2C 30 34
IFGO L20	при $X > 0$ идти к L20	0C	0A 00 00 1C
MOV M, X	$X := M$	10	00 2C 00 34
MOV N, M	$M := N$	14	00 30 00 2C
MOV X, N	$N := X$	18	00 34 00 30
L20: SUB M, N, M	$M := M - N$	1C	02 2C 30 2C
IFGO L10	при $M > 0$ идти к L10	20	0A 00 00 08
OUTPUT N	вывод N	24	00 30 00 FC
STOP	останов	28	77 00 00 00

Здесь уже две команды имеют метки: L10 отмечает начало цикла, L20 — конец ветвления.

Трансляция с автокода. Разработать язык программирования для ЭВМ — это значит:

- 1) описать язык;
- 2) создать транслятор с этого языка для данной ЭВМ.

Транслятор с автокода — это программа, для которой исходной информацией является текст на Автокоде. Процедура трансляции начинается с поиска синтаксических ошибок в тексте программы. Обнаружив ошибку, транслятор выводит сообщение, указывающее на место ошибки в программе и ее характер. Получив такое сообщение, программист должен исправить ошибку и снова повторить трансляцию. Так продолжается до тех пор, пока не будут исправлены все синтаксические ошибки.

Описанные действия называются *синтаксической отладкой программы*. Синтаксическая отладка выполняется программистом совместно с транслятором.

Если синтаксических ошибок не обнаружено, то начинается второй этап работы транслятора — перекодировка программы в машинные команды. Результатом этой работы является машинный код (программа на ЯМК).

Сказанное выше о работе транслятора и синтаксической отладке программы на автокоде отражено блок-схемой на рис. 18.7.

А теперь опишем алгоритм, по которому работает транслятор с автокода после завершения синтаксической отладки.

Распределение памяти под программу. Поскольку каждая команда автокода переводится в одну команду машинного языка и программа располагается, начиная с нулевой ячейки, то легко определить размер программы (адрес ее последней команды).

Распределение памяти под данные. Сразу вслед за последней командой программы помещаются пере-

менные и константы в той последовательности, в которой они встречаются в программе на автокоде.

Перекодировка команд. Каждая команда на автокоде переводится в соответствующую команду на машинном языке. При этом мнемонический код заменяется на код операции, а переменные и константы — на их машинные адреса. Константы переводятся в двоичную систему и записываются в отведенные для них ячейки.

Посмотрите на результат трансляции программы из задачи 1. Напротив каждой команды Автокода написан результат ее перевода в команду ЯМК.

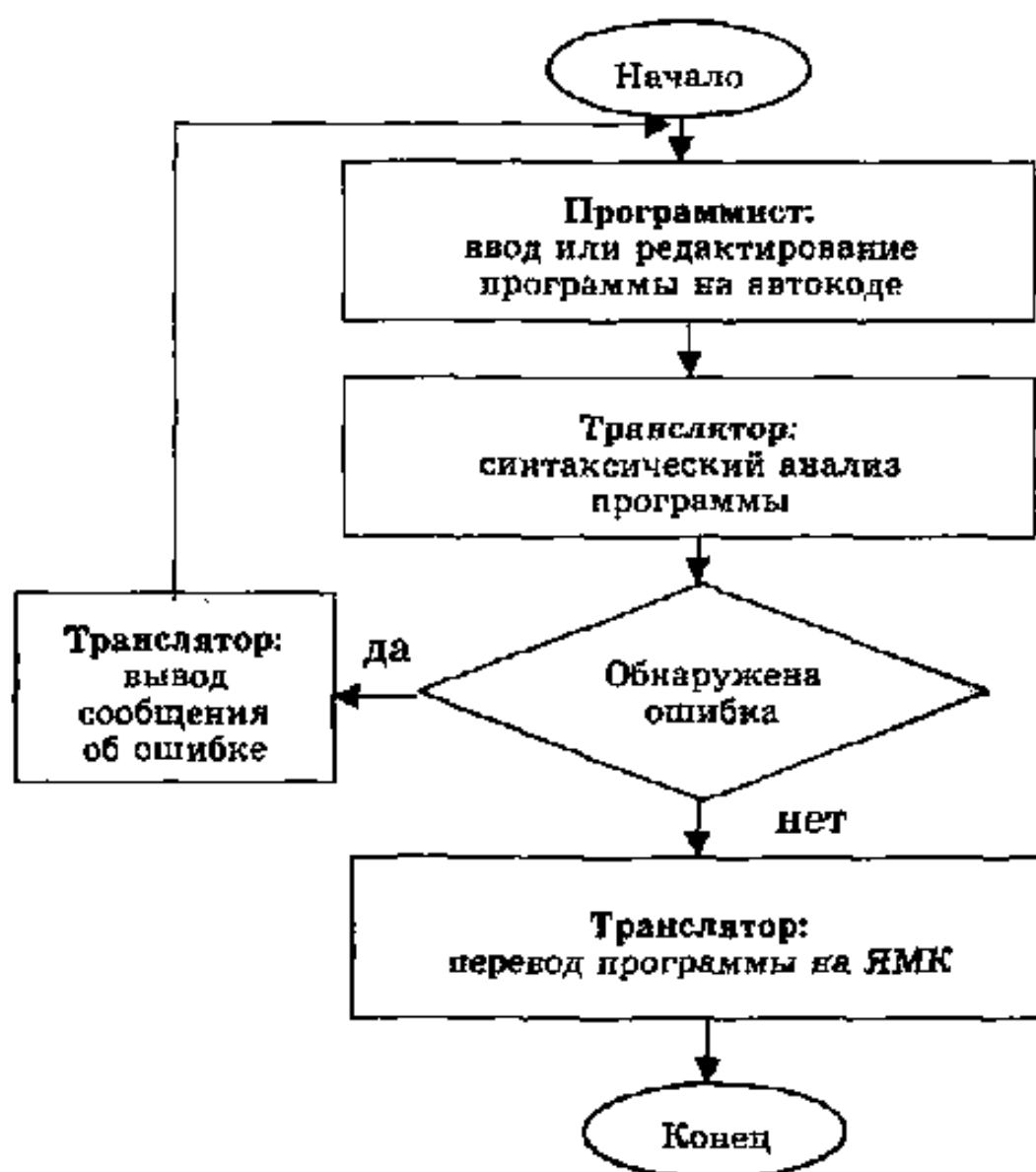


Рис. 18.7. Алгоритм трансляции

Обратите внимание на то, как транслируются команды ввода и вывода. По-прежнему это пересылка через буферную ячейку FC.

Исполнение программы, полученной в результате трансляции, будет происходить так же, как и раньше при работе на ЯМК. *Ввод исходных данных и вывод результатов происходит в шестнадцатеричной форме.*

Для того чтобы при работе на автокоде числа вводились и выводились в десятичной системе счисления, нужно иначе организовать трансляцию команд INPUT, OUTPUT. Команда INPUT должна транслироваться в процедуру перевода введенного числа из десятичной системы в двоичную; команда OUTPUT — в процедуру перевода из двоичной системы в десятичную и вывода результата. В принципе, это сделать можно. С целью упрощения материала такая возможность на «УК Нейман» не реализована. Не надо забывать, что наш компьютер учебный, и он не предназначен для широкого применения.

Память распределяется транслятором «плотно». Ячейки под данные выделяются сразу же после последней команды программы. Таким образом, между данными и программой не остается пустых ячеек. Программа, полученная транслятором в задаче 1, отличается от ранее написанной нами программы адресами ячеек, выделенных под данные.

В правом столбце таблицы из задачи 3 также показан результат трансляции программы на автокоде. Память под переменные распределена следующим образом:

2C — переменная M;

30 — переменная N;

34 — переменная X.

Работа с мини-Паскалем. Как уже было сказано выше, мини-Паскаль — это подмножество языка Паскаль, ограниченное тем, что на нем можно работать с целыми числами. Знакомство с ним полезно с методи-

ческой точки зрения. Оно дает возможность продемонстрировать ученикам процедуру трансляции с языка высокого уровня. Трансляция с мини-Паскаля происходит в два этапа:

ПАСКАЛЬ $\Rightarrow$ АВТОКОД $\Rightarrow$ ЯМК

В некотором упрощенном приближении эта процедура отражает процедуру трансляции в реальных системах программирования: первый этап — получение объектного кода, второй этап — получение исполнимой программы на ЯМК. Исполнитель «УК Нейман» дает возможность проследить за этими преобразованиями программы на экране.

Дидактические материалы к организации учебных занятий

В различных регионах РФ и отдельных школах нередко действуют различные учебные планы изучения информатики. Поэтому в каждом конкретном случае план изучения базового курса информатики должен быть поставлен в соответствие с учебным планом, принятым в школе. Минимальный вариант учебного плана — 68 часов: 2 года по 1 часу в неделю (например, в 8—9 классах). Углубленное изучение может происходить по учебному плану объемом 136 часов. Его планирование возможно на 3 учебных года: с 7 по 9 класс (например, 7 кл. — 2 ч/нед., 8 кл. — 1 ч/нед., 9 кл. — 1 ч/нед.), или на 2 года: 8—9 классы по 2 ч/нед.. Промежуточный вариант учебного плана — 102 часа: 3 года по 1 часу в неделю. Учебно-методическое обеспечение базового курса построено таким образом, что он может преподаваться на разном уровне глубины изучения, по разным учебным планам.

Далее приводятся три варианта поурочного планирования занятий. 1-й вариант — 68 часов: 2 года по 1 часу в неделю; 2-й вариант — 136 часов: 2 года по 2 часа в неделю; 3-й вариант — 102 часа: 3 года по 1 часу в неделю. Весь материал разделен на содержательные модули, изучение которых заканчивается контрольным тестированием. Для вариантов 1 и 2 даны лишь темы уроков. Для варианта 3 помимо тем приводятся разработки уроков, домашние задания, практические задания для выполнения на компьютере и контрольные тесты по содержательным модулям.

Поскольку учебный курс ориентирован на массовую школу, то для организации обучения используется

наиболее распространенная классно-урочная форма обучения. При этом для организации практических занятий с использованием компьютеров класс делится на две группы.

Теоретическая часть материала преподается в режиме фронтального урока. Практическая часть занятий, связанная с работой на компьютерах, происходит в компьютерном классе (1 компьютер на 1—2 ученика). Контроль теоретических знаний учащихся происходит в форме фронтального опроса, проверки домашних заданий, тестирования по тематическим разделам курса. Контроль практических умений и навыков происходит путем приема индивидуальных заданий, выполняемых учащимися на компьютерах.

Для организации обучения базовому курсу информатики требуется компьютерный класс, укомплектованный 7—15 персональными компьютерами. В комплект требуемого программного обеспечения входят:

- ☐ клавиатурный тренажер;
- ☐ текстовый редактор;
- ☐ графический редактор растрового типа;
- ☐ простая реляционная СУБД;
- ☐ табличный процессор;
- ☐ почтовая клиент-программа и браузер WWW (для работы в Интернете или Intranet), или учебные имитаторы этих программ;
- ☐ учебные исполнители алгоритмов (Кенгуренок, Чертежник и т. п.)
- ☐ имитатор учебного компьютера «УК Нейман» (можно найти на сайте <http://www.emc.km.ru>)

Для изучения углубленного варианта базового курса нужны:

- ☐ программа подготовки презентаций;
- ☐ система программирования на Прологе;
- ☐ система программирования на Паскале.

Вариант 1

**Два года обучения (8—9 классы)
по 1 уроку в неделю (68 часов)**

План уроков**8 класс**

№ урока	Тема урока	Вид занятия
	Модуль 1. Введение в предмет. Понятие информации: виды, действия, измерение (алфавитный подход). Понятие систем счисления	
1	Введение в предмет. Понятие информации. Информативность сообщения. Знакомство учеников с компьютерным классом	Теория + практика
2	Виды информации. Информационные процессы	Теория
3	Предыстория информатики. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика
4	Измерение информации (алфавитный подход). Единицы измерения информации	Теория
5	История чисел и понятие системы счисления. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика
6	Знакомство с клавиатурой. Основные приемы редактирования текста. Набор текста по образцу	Практика
7	Тестирование 1	
	Модуль 2. Архитектура компьютера. Основные устройства персонального компьютера. Программное обеспечение. Операционные системы	
8	Начальные сведения об архитектуре компьютера. Виды компьютерной памяти и их сравнение. Свойства оперативной памяти	Теория

№ урока	Тема урока	Вид занятия
9	Состав программного обеспечения. Функции операционной системы	Теория
10	Пользовательский интерфейс ОС Windows. Знакомство с ОС Windows: работа с окнами, запуск и завершение работы программ	Теория + практика
11	Состав и назначение основных устройств персонального компьютера	Теория
12	Файлы и файловые структуры. Понятие файлового менеджера	Теория
13	Работа с файлами и папками в ОС Windows (использование файловых менеджеров)	Практика
14	Шаблоны имен файлов. Понятие ярлыков. Работа с группами файлов. Поиск файлов на диске	Теория + практика
15	Тестирование 2	
	Модуль 3. Тексты в компьютерной памяти. Текстовые редакторы. Компьютерная графика. Графические редакторы	
16	Тексты в компьютерной памяти	Теория
17	Понятие текстового файла и текстового документа. Назначение текстовых редакторов. Среда текстового редактора. Редактирование готового текста	Теория + практика
18	Работа с фрагментами текста. Многооконный режим работы текстового редактора	Практика
19	Форматирование текста. Использование списков	Практика
20	Работа со шрифтами. Поиск и замена	Практика
21	Компьютерная графика и области её применения. Типы графических пакетов. Понятие графических примитивов	Теория

№ урока	Тема урока	Вид занятия
22	Интерфейс графического редактора. Построение изображения с использованием различных графических примитивов. Работа с фрагментами изображения	Практика
23	Технические средства компьютерной графики. Принципы работы растровых дисплеев	Теория
24	Поворот и отображение рисунка. Работа с текстом	Практика
25	Кодирование изображения	Теория
26	Тестирование 3	
	Модуль 4. Информационные структуры и базы данных. Системы управления базами данных	
27	Понятие модели. Виды информационных моделей. Табличная организация информации	Теория
28	Создание таблицы типа «объект—свойства»	Практика
29	Понятие информационной структуры. Базы данных и информационные системы. Основные понятия реляционных БД. Типы данных. Представление о СУБД	Теория
30	Знакомство с СУБД. Создание и редактирование базы данных	Практика
31	Логические выражения. Поиск информации. Формирование простых запросов к БД	Теория + практика
32	Логические операции. Сложные логические выражения. Сортировка записей	Теория
33	Формирование запросов на выборку и удаление записей с использованием логических операций. Сортировка записей	Практика
34	Итоговое тестирование	

9 класс

№ урока	Тема урока	Вид за- нятия
	<i>Модуль 1. Компьютерные сети и их типы. Табличные расчеты и электронные таблицы</i>	
1	Компьютерные сети и их типы. Аппаратное и программное обеспечение компьютерных сетей	Теория
2	Услуги глобальных компьютерных сетей (электронная почта, телеконференции, файловые архивы).	Теория
3	Услуги глобальных компьютерных сетей (World Wide Web). Работа с электронной почтой (альтернатива: работа с локальной сетью)	Теория + практика
4	Работа с WWW. Поиск информации в Интернете	Практика
5	Знакомство с электронными таблицами. Просмотр и редактирование электронной таблицы	Теория + практика
6	Ввод информации в электронные таблицы: тексты, числа, формулы	Теория
7	Разработка электронной таблицы	Практика
8	Понятие диапазона. Относительная и абсолютная адресация. Статистические функции СУММ, МИН, МАКС, СРЗНАЧ	Теория
9	Использование абсолютной и относительной адресации. Сортировка данных	Практика
10	Логические операции и условная функция	Теория
11	Использование логических операций и условной функции	Практика
12	Графическая обработка табличных данных. Табулирование функции и построение графика. Построение диаграммы	Теория + практика
13	Тестирование 1	

N урока	Тема урока	Вид за- нятия
	Модуль 2. Управление и кибернетика. Алгоритм и его свойства. Алгоритмические структуры. Графический учебный исполнитель	
14	Кибернетика. Кибернетическая модель управления. Системы управления с использованием компьютеров (АСУ и САУ)	Теория
15	Определение и свойства алгоритма. Линейные алгоритмы. Использование алгоритмического языка и блок-схем для записи алгоритмов	Теория
16	Графический учебный исполнитель алгоритмов. Разработка линейных алгоритмов	Теория + практика
17	Вспомогательные алгоритмы. Последовательная детализация	Теория
18	Учебный исполнитель алгоритмов. Последовательная детализация	Практика
19	Команда цикла. Циклические алгоритмы	Теория
20	Учебный исполнитель алгоритмов. Команда цикла. Циклические алгоритмы	Практика
21	Ветвления. Циклы в сочетании с ветвлениями	Теория
22	Учебный исполнитель алгоритмов. Ветвления. Циклы в сочетании с ветвлениями	Практика
23	Зачетное занятие по алгоритмизации	Практика
24	Тестирование 2	
	Модуль 3. Продолжение темы системы счисления. Понятие о программном управлении компьютером. Язык машинных команд. История ЭВМ.	
25	Перевод целых чисел из одной системы счисления в другую	Теория

№ урока	Тема урока	Вид за- нятия
26	Двоичная система счисления и двоичная арифметика. Связь между двоичной, восьмеричной и шестнадцатеричной системами счисления	Теория
27	Контрольная работа по системам счисления	Практика
28	Структура памяти учебного компьютера «УК Нейман». Представление целых чисел в памяти компьютера	Теория
29	Язык машинных команд	Теория
30	Учебный компьютер «УК Нейман». Язык машинных команд	Практика
31	Устройство и работа процессора	Теория
32	Язык машинных команд	Практика
33	История развития ЭВМ	Теория
34	Итоговое тестирование	

Вариант 2

**Два года обучения (8—9 классы)
по 2 урока в неделю (136 часов)**

План уроков

8 класс

№ урока	Тема урока	Вид за- нятия
	<i>Модуль 1. Введение в предмет. Понятие информации: виды, действия, измерение (алфавитный подход). Понятие систем счисления</i>	
1	Введение в предмет. Знакомство учеников с компьютерным классом	Теория + практика

№ урока	Тема урока	Вид занятия
2	Информация как знания человека. Информативность сообщения. Виды информации. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика
3	Информационные процессы	Теория
4	Предыстория информатики. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика
5	Измерение информации (алфавитный подход). Единицы измерения информации	Теория
6	История чисел и понятие системы счисления. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика
7	Знакомство с клавиатурой. Основные приемы редактирования текста. Набор текста по образцу	Практика
8	Тестирование 1	
	<i>Модуль 2. Архитектура компьютера. Основные устройства персонального компьютера. Программное обеспечение. Операционные системы. Архиваторы и антивирусные программы.</i>	
9	Начальные сведения об архитектуре компьютера	Теория
10	Состав программного обеспечения. Функции операционной системы	Теория
11	Пользовательский интерфейс ОС Windows. Знакомство с ОС Windows: работа с окнами, запуск и завершение работы программ	Теория + практика
12	Виды компьютерной памяти и их сравнение. Свойства оперативной памяти	Теория
13	Состав и назначение основных устройств персонального компьютера	Теория
14	Файлы и файловые структуры. Понятие файлового менеджера	Теория

№ урока	Тема урока	Вид занятия
15	Работа с файлами и папками в ОС Windows (использование файловых менеджеров)	Практика
16	Шаблоны имен файлов. Понятие ярлыков. Работа с группами файлов. Поиск файлов на диске	Теория + практика
17	Понятие буфера обмена. Использование буфера обмена для копирования и перемещения информации. Сохранение и загрузка файлов	Теория + практика
18	История развития ЭВМ	Теория
19	Зачетное занятие по ОС Windows.	Практика
20	Архиваторы. Компьютерные вирусы и антивирусные программы	Теория
21	Работа с программой-архиватором	Практика
22	Тестирование 2.	
	Модуль 3. Тексты в компьютерной памяти. Текстовые редакторы	
23	Тексты в компьютерной памяти	Теория
24	Понятие текстового файла и текстового документа. Назначение текстовых редакторов и их типы. Знакомство с интерфейсом текстового редактора и изменение его настроек	Теория + практика
25	Орфографическая проверка готового текста и исправление ошибок. Работа со строками	Практика
26	Режимы работа текстового редактора. Основные типы данных (символ, слово, абзац, строка). Понятие фрагмента данных	Теория
27	Работа с фрагментами текста. Многооконный режим работы текстового редактора	Практика
28	Типы и стили шрифтов	Теория
29	Работа со шрифтами. Поиск и замена	Практика
30	Форматирование текста. Использование списков	Практика

№ урока	Тема урока	Вид занятия
31	Использование таблиц. Вставка графического изображения. Использование объектов WordArt	Практика
32	Зачетное занятие по текстовому редактору	Практика
33	Тестирование 3	
	Модуль 4. Компьютерная графика. Графические редакторы. Понятие мультимедиа	
34	Компьютерная графика и области её применения. Типы графических пакетов. Понятие графических примитивов	Теория
35	Интерфейс растрового графического редактора. Построение изображения с использованием различных графических примитивов. Работа с фрагментами изображения	Практика
36	Технические средства компьютерной графики. Принципы работы растровых дисплеев	Теория
37	Работа с растровым графическим редактором. Поворот и отображение рисунка. Работа с текстом	Практика
38	Растровая и векторная графика. Форматы графических файлов	Теория
39	Знакомство с векторным графическим редактором. Построение изображения с использованием простейших графических примитивов	Практика
40	Кодирование изображения	Теория
41	Работа с векторным графическим редактором. Типы заливки	Практика
42	Понятие мультимедиа. Кодирование звука. Мультимедийные устройства	Теория
43	Зачетное занятие по графическому редактору	Практика
44	Тестирование 4	

№ урока	Тема урока	Вид занятия
	Модуль 5. Элементы математической логики. Содержательный подход к измерению информации. Вероятность и информация. Алгоритмы сжатия текстовой информации. Пакет создания презентаций PowerPoint	
45	Элементы математической логики. Логические высказывания, логические величины, логические операции. Логические схемы	Теория
46	Решение логических задач методом рассуждений и табличным методом	Теория
47	Содержательный подход к измерению информации	Теория
48	Знакомство с пакетом создания презентаций. Создание простейшей презентации	Практика
49	Вероятность и информация (содержательный подход)	Теория
50	Разработка заданной презентации. Использование организационной диаграммы	Практика
51	Формула Шеннона. Решение задач	Теория
52	Использование гиперссылок и кнопок перехода при разработке презентации	Практика
53	Алгоритмы сжатия текстовой информации	Теория
54	Тестирование 5	
	Модуль 6. Компьютерные сети и их типы. Знакомство с языком разметки гипертекста HTML	
55	Компьютерные сети и их типы. Основные типы услуг, предоставляемых клиентам глобальных компьютерных сетей (электронная почта, телеконференции, файловые архивы)	Теория
56	Работа с электронной почтой	Практика
57	Всемирная паутина (World Wide Web). Поисковые серверы. Формирование простых запросов для поисковых серверов	Теория

№ урока	Тема урока	Вид занятия
58	Работа с WWW: использование URL-адресов и гиперссылок	Практика
59	Работа с WWW. Поиск информации в Интернете	Практика
60	Основные компоненты компьютерных сетей	Теория
61	Работа в локальной сети	Практика
62	Основные понятия языка HTML. Использование цветов. Форматирование текста. Гиперссылки	Теория
63	Разработка Web-страниц с использованием гиперссылок и форматирования	Практика
64	Работа с графикой. Шрифты	Теория
65	Разработка Web-страниц с использованием графики и различных шрифтов	Практика
66	Разработка таблиц в HTML	Теория
67	Создание Web-страницы с использованием таблицы	Практика
68	Итоговое тестирование	

9 класс

№ урока	Тема урока	Вид занятия
	Модуль 1. Информационные модели и информационные структуры. Базы данных и СУБД	
1	Понятие модели. Виды информационных моделей	Теория
2	Табличная организация информации. Создание таблицы типа «объект—свойства»	Теория + практика

№ урока	Тема урока	Вид за- нятия
3	Понятие информационной структуры. Базы данных и информационные системы. Представление о СУБД	Теория
4	Основные понятия. Типы данных. Понятие структуры таблицы. Первичный ключ	Теория
5	Знакомство с СУБД. Создание и редактирование базы данных	Практика
6	Формирование запросов к БД. Сортировка записей	Теория
7	Формирование запросов на выборку, замену и удаление записей с использованием логических операций. Сортировка записей	Практика
8	Вычисляемые поля. Формирование отчетов и форм с использованием Мастеров	Теория + практика
9	Проектирование многотабличных БД и установка связей между ними	Теория
10	Проектирование структуры многотабличной БД и ее создание с помощью СУБД	Практика
11	Зачетное занятие по СУБД	Практика
12	Тестирование 1	
	Модуль 2. Базы знаний. Язык логического программирования Пролог	
13	Понятие искусственного интеллекта и баз знаний	Теория
14	Язык логического программирования Пролог. Основные понятия Пролога: факт, правило, цель. Синтаксис Турбо Пролога	теория
15	Создание простейших баз знаний на Турбо Прологе и формирование к ним целей	Практика
16	Конъюнкция и дизъюнкция в Прологе	Теория
17	Использование конъюнкции и дизъюнкции в Турбо Прологе для решения задач	Практика

№ урока	Тема урока	Вид занятия
18	Операция отрицания. Решение задач с использованием логических операций	Теория
19	Использование операции отрицания при решении задач на Турбо Прологе	Практика
20	Тестирование 2	
	Модуль 3. Табличные расчеты и электронные таблицы	
21	Знакомство с электронными таблицами. Просмотр и редактирование электронной таблицы	Теория + практика
22	Ввод информации в электронные таблицы: тексты, числа, формулы	Теория
23	Разработка электронной таблицы	Практика
24	Понятие диапазона. Относительная и абсолютная адресация. Статистические функции СУММ, МИН, МАКС, СРЗНАЧ	Теория
25	Использование абсолютной и относительной адресации. Сортировка данных	Практика
26	Логические операции и условная функция	Теория
27	Использование логических операций и условной функции	Практика
28	Графическая обработка табличных данных. Табулирование функции и построение графика. Построение диаграммы	Теория + практика
29	Зачетное занятие по ЭТ	Практика
30	Тестирование 3	
	Модуль 4. Управление и кибернетика. Алгоритм и его свойства. Алгоритмические структуры. Графический учебный исполнитель	
31	Кибернетика. Кибернетическая модель управления. Системы управления с использованием компьютеров (АСУ и САУ)	Теория

№ урока	Тема урока	Вид занятия
32	Определение и свойства алгоритма. Линейные алгоритмы. Использование алгоритмического языка и блок-схем для записи алгоритмов	Теория
33	Графический учебный исполнитель алгоритмов. Разработка линейных алгоритмов	Теория + практика
34	Вспомогательные алгоритмы. Последовательная детализация	Теория
35	Учебный исполнитель алгоритмов. Последовательная детализация	Практика
36	Команда цикла. Циклические алгоритмы	Теория
37	Учебный исполнитель алгоритмов. Команда цикла. Циклические алгоритмы	Практика
38	Ветвления. Циклы в сочетании с ветвлениями	Теория
39	Учебный исполнитель алгоритмов. Ветвления. Циклы в сочетании с ветвлениями	Практика
40	Алгоритмы работы с величинами: ввод, вывод, присваивание, ветвление	Теория
41	Технология решения задач с использованием компьютера. Циклы в алгоритмах работы с величинами	Теория
42	Зачетное занятие по алгоритмизации	Практика
43	Тестирование 4	
	Модуль 5. Перевод чисел из одной системы счисления в другую. Понятие о программном управлении компьютером. Язык машинных команд и автокод. Алгоритмы работы с величинами	
44	Перевод целых чисел из одной системы счисления в другую	Теория
45	Двоичная система счисления и двоичная арифметика. Связь между двоичной, восьмеричной и шестнадцатеричной системами счисления	Теория

№ урока	Тема урока	Вид занятия
46	Контрольная работа по системам счисления (перевод целых чисел)	Практика
47	Перевод вещественных чисел. Внутреннее машинное представление целого числа в двухбайтовом машинном слове	Теория
48	Контрольная работа по системам счисления (перевод вещественных чисел и внутреннее машинное представление целых чисел)	Практика
49	Структура памяти учебного компьютера «УК Нейман». Устройство и работа процессора	Теория
50	Учебный компьютер «УК Нейман». Язык машинных команд. Линейные алгоритмы	Теория
51	Язык машинных команд. Разработка линейных алгоритмов	Практика
52	Язык машинных команд. Ветвления и циклы	Теория
53	Язык машинных команд. Использование ветвлений и циклов при решении задач	Практика
54	Автокод. Понятие трансляторов. Алгоритм трансляции с автокода на ЯМК. Линейные алгоритмы на автокоде	Теория
55	Автокод. Разработка линейных алгоритмов	Практика
56	Автокод. Ветвления и циклы	Теория
57	Автокод. Использование ветвлений и циклов при решении задач	Практика
58	Зачетное занятие по ЯМК и автокоду	Практика
59	Тестирование 5	
	<i>Модуль 6. Языки программирования высокого уровня. Понятие объектно-ориентированного проектирования. Знакомство с языком программирования Паскаль</i>	
60	Языки программирования высокого уровня. Компиляторы и интерпретаторы	Теория

№ урока	Тема урока	Вид занятия
61	Объектно-ориентированное проектирование. Объекты, их свойства и поведение	Теория
62	Структура Паскаль-программы. Операторы ввода, вывода и присваивания	Теория
63	Разработка простейших программ на языке Паскаль. Линейные алгоритмы	Практика
64	Ветвления на языке Паскаль. Решение задач	Теория
65	Использование ветвлений на языке Паскаль при решении задач	Практика
66	Циклы с предусловием и с постусловием на языке Паскаль	Теория
67	Использование циклов с предусловием и с постусловием на языке Паскаль при решении задач	Практика
68	Итоговое тестирование	

Вариант 3

Три года обучения (7—9 классы)
по 1 уроку в неделю (102 часа)

План уроков

7 класс

№ урока	Тема	Вид занятий	Вид отчета
	Модуль 1. Введение в предмет. Понятие информации: виды, действия, измерение (алфавитный подход). Понятие систем счисления		
1	Введение в предмет. Знакомство учеников с компьютерным классом	Теория + практика	Д/з № 1

№ урока	Тема	Вид занятий	Вид отчета
2	Информация как знания человека. Информативность сообщения. Виды информации. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика	Д/з № 2
3	Информационные процессы	Теория	Д/з № 3
4	Предыстория информатики. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика	Д/з № 4
5	Измерение информации (алфавитный подход). Единицы измерения информации	Теория	Д/з № 5
6	История чисел и понятие системы счисления. Знакомство с клавиатурой. Работа с клавиатурным тренажером	Теория + практика	Д/з № 6
7	Знакомство с клавиатурой. Основные приемы редактирования текста. Набор текста по образцу	Практика	П/з № 1
8	Тестирование		Тест № 1
	Модуль 2. Архитектура компьютера. Основные устройства персонального компьютера. Программное обеспечение. Операционные системы		
9	Начальные сведения об архитектуре компьютера	Теория	Д/з № 7
10	Состав программного обеспечения. Функции операционной системы	Теория	Д/з № 8
11	Пользовательский интерфейс ОС Windows. Знакомство с ОС Windows: работа с окнами, запуск и завершение работы программ	Теория + практика	Д/з № 9, П/з № 2
12	Виды компьютерной памяти и их сравнение. Свойства оперативной памяти	Теория	Д/з № 10

№ урока	Тема	Вид занятий	Вид отчета
13	Состав и назначение основных устройств персонального компьютера	Теория	Д/з № 11
14	Файлы и файловые структуры. Понятие файлового менеджера	Теория	Д/з № 12
15	Работа с файлами и папками в ОС Windows (использование файловых менеджеров)	Практика	П/з № 3
16	Шаблоны имен файлов. Понятие ярлыков. Работа с группами файлов. Поиск файлов на диске	Теория + практика	Д/з № 13, П/з № 4
17	Понятие буфера обмена. Использование буфера обмена для копирования и перемещения информации. Сохранение и загрузка файлов	Теория + практика	Д/з № 14, П/з № 5
18	Зачетное занятие по ОС Windows.	Практика	П/з № 6
19	Тестирование		Тест № 2
	<i>Модуль 3. Тексты в компьютерной памяти. Текстовые редакторы. Компьютерная графика. Графические редакторы</i>		
20	Тексты в компьютерной памяти	Теория	Д/з № 15
21	Понятие текстового файла и текстового документа. Назначение текстовых редакторов и их типы. Знакомство с интерфейсом текстового редактора и изменение его настроек	Теория + практика	Д/з № 16, П/з № 7
22	Орфографическая проверка готового текста и исправление ошибок. Работа со строками	Практика	П/з № 8
23	Режимы работа текстового редактора. Основные типы данных (символ, слово, абзац, строка). Понятие фрагмента данных	Теория	Д/з № 17

№ урока	Тема	Вид за- нятий	Вид от- чета
24	Работа с фрагментами текста. Много- скользящий режим работы текстового редактора	Практика	П/з № 9
25	Типы и стили шрифтов	Теория	Д/з № 18
26	Работа со шрифтами. Поиск и заме- на	Практика	П/з № 10
27	Форматирование текста. Использо- вание списков	Практика	П/з № 11
28	Зачетное занятие по текстовому ре- дактору	Практика	П/з № 12
29	Компьютерная графика и области ее применения. Типы графических паке- тов. Понятие графических примити- вов	Теория	Д/з № 19
30	Интерфейс графического редактора. Построение изображения с использо- ванием различных графических примитивов. Работа с фрагментами изображения	Практика	П/з № 13
31	Технические средства компьютерной графики. Принципы работы растро- вых дисплеев	Теория	Д/з № 20
32	Поворот и отображение рисунка. Ра- бота с текстом	Практика	П/з № 14
33	Кодирование изображения	Теория	Д/з № 21
34	Итоговое тестирование		Тест № 3

Урок 1

Теория. Введение в предмет. (35 мин)

Изучаемые вопросы:

- ☐ Понятия вещества, энергии, информации.
- ☐ Информатика как наука.
- ☐ Компьютер — универсальное средство для работы с информацией.

Практика. Знакомство учеников с компьютерным классом. Техника безопасности и правила поведения в компьютерном классе. (10 мин)

Домашнее задание:

- ☐ Читать: [5] — Введение.
- ☐ Выполнить: домашнее задание № 1.

Урок 2

Теория. Информация как знания человека. Информативность сообщения. Виды информации. (30 мин)

Изучаемые вопросы:

- ☐ Информация как знания человека.
- ☐ Декларативные и процедурные знания.
- ☐ Информативность сообщения.
- ☐ Образная и знаковая формы восприятия информации.
- ☐ Разговорные (естественные) и формальные языки.

Практика. Знакомство с клавиатурой. Работа с клавиатурным тренажером. (15 мин)

Групповые задания и упражнения.

Набор русских букв на клавиатуре.

Домашнее задание:

- ☐ Читать: [5] — § 1, 2.
- ☐ Выполнить: домашнее задание № 2.

Урок 3

Теория. Информационные процессы. (45 мин)

Изучаемые вопросы:

- ☐ Виды информационных процессов: хранение, передача, обработка.
- ☐ Информационные носители.

- Каналы передачи информации.
- Виды обработки информации.

Домашнее задание:

- Читать: [5] — § 3.
- Выполнить: домашнее задание № 3.

Урок 4

Теория. Предыстория информатики. (30 мин)

Изучаемые вопросы:

- Докомпьютерные средства хранения, обработки и передачи информации.
- Аналитическая машина Чарльза Бэббиджа.

Практика. Знакомство с клавиатурой. Работа с клавиатурным тренажером. (15 мин)

Приобретаемые умения и навыки:

набор русских букв на клавиатуре.

Домашнее задание:

- Читать: [5] — § 5.
- Выполнить: домашнее задание № 4.

Урок 5

Теория. Измерение информации (алфавитный подход). Единицы измерения информации. (45 мин)

Изучаемые вопросы:

- Понятие алфавита и его мощности.
- Формула определения количества информации.
- Единицы измерения количества информации (байт, килобайт, мегабайт, гигабайт).
- Скорость информационного потока.
- Каналы связи и их пропускная способность.

Домашнее задание:

- Читать: [5] — § 4.
- Выполнить: домашнее задание № 5.

Урок 6

Теория. История чисел и понятие системы счисления. (30 мин)

Изучаемые вопросы:

- Понятие системы счисления.

- Позиционные и непозиционные системы счисления.
- Алфавит и основание системы счисления.

Практика. Знакомство с клавиатурой. Работа с клавиатурным тренажером. (15 мин)

Приобретаемые умения и навыки:

набор латинских букв на клавиатуре.

Домашнее задание:

- Читать: [5] — §6.
- Выполнить: домашнее задание № 6.

Урок 7

Практика. Знакомство с клавиатурой. Основные приемы редактирования текста. Набор текста по образцу. (Выполнение практического задания № 1.) (45 мин)

Приобретаемые умения и навыки:

- 1) ввод русских и латинских букв при наборе текста;
- 2) ввод строчных и прописных букв;
- 3) ввод знаков препинания;
- 4) редактирование набранного текста: вставка и удаление символов, слияние и разрыв строк.

Урок 8

Тестирование. Тест № 1 по теме «Понятие информации: виды, действия, измерение (алфавитный подход). Понятие систем счисления». (45 мин)

Урок 9

Теория. Начальные сведения об архитектуре компьютера. (45 мин)

Изучаемые вопросы:

- Понятие архитектуры ЭВМ.
- Основные типы устройств компьютера (устройства ввода, устройства вывода, устройства хранения, устройство обработки информации).
- Схема устройства компьютера.
- Понятие программы и данных.

Домашнее задание:

- Читать: [5] — §7.
- Выполнить: домашнее задание № 7.

Урок 10

Теория. Состав программного обеспечения. Функции операционной системы. (45 мин)

Изучаемые вопросы:

- Понятие программного обеспечения.
- Назначение прикладного программного обеспечения.
- Системное ПО и функции операционной системы.
- Понятие системы программирования.

Домашнее задание:

- Читать: [5] — §10, 11.
- Выполнить: домашнее задание № 8.

Урок 11

Теория. Пользовательский интерфейс ОС Windows. (20 мин)

Изучаемые вопросы:

- Понятия объектно-ориентированного интерфейса и объекта.
- Виды объектов в ОС Windows.
- Свойства и поведение объектов.

Практика. Знакомство с ОС Windows: работа с окнами, запуск и завершение работы программ. Выполнение практического задания № 2. (25 мин)

Приобретаемые умения и навыки:

- 1) запуск приложений;
- 2) изменение размеров окна;
- 3) перемещение окна;
- 4) переключение между запущенными приложениями;
- 5) сворачивание окна и его восстановление;
- 6) закрытие окна и завершение работы приложения.

Домашнее задание:

- Читать: [5] — § 13.
- Выполнить: домашнее задание № 9.

Урок 12

Теория. Виды компьютерной памяти и их сравнение. Свойства оперативной памяти. (45 мин)

Изучаемые вопросы:

- ☐ Двоичное кодирование информации в памяти компьютера.
- ☐ Дискретность и адресуемость — свойства оперативной памяти.
- ☐ Виды внешней памяти (жесткие и гибкие магнитные диски, лазерные диски).
- ☐ Файловая организация информации на внешних носителях.
- ☐ Сравнение оперативной и внешней памяти.

Домашнее задание:

- ☐ Читать: [5] — § 8.
- ☐ Выполнить: домашнее задание № 10.

Урок 13

Теория. Состав и назначение основных устройств персонального компьютера. (45 мин)

Изучаемые вопросы:

- ☐ Состав персонального компьютера и назначение основных устройств (микропроцессор, оперативная память, магнитные диски, дисководы, монитор, клавиатура, мышь, принтер, контроллеры и адаптеры).
- ☐ Назначение системной шины и ее составляющие.
- ☐ Основные характеристики персонального компьютера.

Домашнее задание:

- ☐ Читать: [5] — § 9.
- ☐ Выполнить: домашнее задание № 11.

Урок 14

Теория. Файлы и файловые структуры. Понятие файлового менеджера. (45 мин)

Изучаемые вопросы:

- ☐ Файловая система как часть ОС.

- Понятия файла и папки (каталога).
- Имена файлов.
- Файловая структура диска.
- Путь к файлу на диске.
- Понятие файлового менеджера.

Домашнее задание:

- Читать: [5] — § 12.
- Выполнить: домашнее задание № 12.

Урок 15

Практика. Работа с файлами и папками в ОС Windows (использование файловых менеджеров). (Выполнение практического задания № 3.) (45 мин)

Приобретаемые умения и навыки:

- 1) смена устройства (логического диска);
- 2) смена папки;
- 3) создание папок;
- 4) копирование файлов и папок;
- 5) перемещение файлов и папок;
- 6) переименование файлов и папок;
- 7) удаление файлов и папок;
- 8) изменение вида содержимого папки;
- 9) сортировка файлов и папок;
- 10) использование корзины для удаления файлов и ее очистка.

Урок 16

Теория. Шаблоны имен файлов. Понятие ярлыков. (20 мин)

Изучаемые вопросы:

- Понятие шаблонов имен файлов и области их применения.
- Понятие ярлыка и его функции.

Практика. Работа с группами файлов. Поиск файлов на диске. (Выполнение практического задания № 4.) (25 мин)

Приобретаемые умения и навыки:

- 1) выделение группы файлов;

2) выполнение действий с группой файлов: копирование, удаление, перемещение;

3) создание ярлыков;

4) поиск файлов на диске.

Домашнее задание:

□ Читать: [24] — раздел 4.5.

□ Выполнить: домашнее задание № 13.

Урок 17

Теория. Понятие буфера обмена. (15 мин)

Изучаемые вопросы:

□ Назначение буфера обмена.

□ Принципы работы с буфером обмена.

Практика. Использование буфера обмена для копирования и перемещения информации. Сохранение и загрузка файлов. Выполнение практического задания № 5. (30 мин)

Приобретаемые умения и навыки:

1) Использование буфера обмена для копирования и перемещения информации между приложениями;

2) сохранение файлов на диске;

3) загрузка файлов с диска.

Домашнее задание:

□ Читать: [24] — раздел 4.5.

□ Выполнить: домашнее задание № 14.

Урок 18

Практика. Зачетное занятие по ОС Windows. Выполнение практического задания № 6. (45 мин)

Урок 19

Тестирование. Тест № 2 по теме «Архитектура ЭВМ. Основные устройства персонального компьютера. Программное обеспечение. Операционные системы». (45 мин)

Урок 20

Теория. Тексты в компьютерной памяти. (45 мин)

Изучаемые вопросы:

- Преимущества файлового хранения текстов.
- Кодировочная таблица, международный стандарт ASCII.
- Понятие гипертекста.

Домашнее задание:

- Читать: [5] — § 14.
- Выполнить: домашнее задание № 15.

Урок 21

Теория. Понятие текстового файла и текстового документа. Назначение текстовых редакторов. (30 мин)

Изучаемые вопросы:

- Понятия текстового файла и текстового документа.
- Назначение текстовых редакторов.
- Типы текстовых редакторов (общего назначения, научных документов, встроенные редакторы систем программирования).

Практика. Знакомство с интерфейсом текстового редактора и изменение его настроек. (Выполнение практического задания № 7.) (15 мин)

Приобретаемые умения и навыки:

- 1) Запуск программы ТР и завершение ее работы;
- 2) вывод и скрытие панелей инструментов;
- 3) добавление и удаление кнопок с панелей инструментов.

Домашнее задание:

- Читать: [5] — § 15.
- Выполнить: домашнее задание № 16.

Урок 22

Практика. Орфографическая проверка готового текста и исправление ошибок. Работа со строками. (Выполнение практического задания № 8.) (45 мин)

Приобретаемые умения и навыки:

- 1) исправление орфографических ошибок в тексте вручную и с использованием словаря;

- 2) разбиение строки на несколько строк;
- 3) соединение нескольких строк в одну.

Урок 23

Теория. Режимы работы текстового редактора. Основные типы данных (символ, слово, абзац, строка). Понятие фрагмента данных. (45 мин)

Изучаемые вопросы:

- Основные режимы работы текстового редактора: ввод/редактирование, форматирование, орфографический контроль, работа с файлами, печать, помощь.
- Типы данных при работе с текстовым редактором: символ, слово, абзац, строка.
- Понятие фрагмента данных и действия, выполняемые над фрагментами.

Домашнее задание:

- Читать: [5] — § 16.
- Выполнить: домашнее задание № 17.

Урок 24

Практика. Работа с фрагментами текста. Многооконный режим работы текстового редактора. (Выполнение практического задания № 9.) (45 мин)

Приобретаемые умения и навыки:

- 1) выделение фрагмента текста;
- 2) копирование, перемещение и удаление фрагментов текста;
- 3) создание нового документа;
- 4) работа в многооконном режиме;
- 5) сохранение документа в файле.

Урок 25

Теория. Типы и стили шрифтов. (45 мин)

Изучаемые вопросы:

- Отличие векторных шрифтов от растровых, преимущества и недостатки каждого из этих типов шрифтов.

- Отличие пропорциональных шрифтов от моноширинных.
- Рубленые прифты и прифты с засечками.
- Виды начертания шрифтов.
- Размеры шрифтов.

Домашнее задание:

- Читать: [24] — разделы 7.3.2—7.3.3.
- Выполнить: домашнее задание № 18.

Урок 26

Практика. Работа со шрифтами. Поиск и замена. (Выполнение практического задания № 10.) (45 мин)

Приобретаемые умения и навыки:

- 1) изменение вида, размера, начертания и цвета шрифта;
- 2) поиск заданного фрагмента в тексте;
- 3) замена одного фрагмента текста на другой.

Урок 27

Практика. Форматирование текста. Использование списков. (Выполнение практического задания № 11.) (45 мин)

Приобретаемые умения и навыки:

- 1) установка параметров страницы;
- 2) установка параметров абзаца;
- 3) переформатирование абзаца;
- 4) выравнивание текста (по центру, по левому краю, по правому краю, по ширине);
- 5) использование списков.

Урок 28

Практика. Зачетное занятие по текстовому редактору. Выполнение практического задания № 12. (45 мин)

Урок 29

Теория. Компьютерная графика и области ее применения. Типы графических пакетов. Понятие графических примитивов. (45 мин)

Изучаемые вопросы:

- Компьютерная графика как раздел информатики.
- Области применения компьютерной графики и типы графических пакетов.
- Понятие графических примитивов.

Домашнее задание:

- Читать: [5] — § 16, 19.
- Выполнить: домашнее задание № 19.

Урок 30

Практика. Интерфейс графического редактора. Построение изображения с использованием различных графических примитивов. Работа с фрагментами изображения. (Выполнение практического задания № 13.) (45 мин)

Приобретаемые умения и навыки:

- 1) запуск программы графического редактора и завершение ее работы;
- 2) использование базовых инструментов;
- 3) изменение рабочего и фоновых цветов;
- 4) стирание части изображения с помощью Ластика;
- 5) сохранение рисунка в файле;
- 6) использование отката;
- 7) масштабирование изображения;
- 8) выделение фрагмента изображения;
- 9) копирование фрагмента;
- 10) перемещение фрагмента;
- 11) удаление фрагмента.

Урок 31

Теория. Технические средства компьютерной графики. Принципы работы растровых дисплеев. (45 мин)

Изучаемые вопросы:

- Графическая система компьютера: графический дисплей и видеоадаптер.
- Принципы работы растровых дисплеев.
- Понятие разрешающей способности монитора в графическом режиме.

- Сканер — устройство ввода графической информации.

Домашнее задание:

- Читать: [5] — § 17.
- Выполнить: домашнее задание № 20.

Урок 32

Практика. Поворот и отображение рисунка. Работа с текстом. (Выполнение практического задания № 14.) (45 мин)

Приобретаемые умения и навыки:

- 1) поворот и отображение рисунка;
- 2) использование инструмента «Текст».

Урок 33

Теория. Кодирование изображения. (45 мин)

Изучаемые вопросы:

- Принцип формирования цвета пикселя на экране.
- Формула определения объема видеопамати для хранения изображения заданного размера.

Домашнее задание:

- Читать: [5] — § 18.
- Выполнить: домашнее задание № 21.

Урок 34

Тестирование. Итоговое тестирование. (45 мин)

Домашние задания

Домашнее задание № 1

1. Выберите из предложенного списка профессии, которые в наибольшей степени связаны с информационной сферой деятельности человека: бухгалтер, повар, водитель, программист, учитель, портной, продавец, журналист, актер.
2. В чем отличие пользователя от программиста?
3. Для каких целей могут использовать компьютер люди следующих профессий:
 - ☐ конструктор автомобилей;
 - ☐ писатель;
 - ☐ бухгалтер;
 - ☐ врач;
 - ☐ физик?
4. Расскажите, для чего вы сами используете (если у вас есть) или могли бы использовать (если у вас нет) свой домашний компьютер.

Домашнее задание № 2

1. Из предложенного списка выберите *декларативные и процедурные* знания:
 - а) чтобы определить кислотность раствора, нужно опустить в него лакмусовую бумажку и посмотреть на ее цвет: если цвет красный, то раствор кислотный; если цвет синий, то раствор щелочной; если цвет не изменился, то раствор нейтральный;
 - б) Вашингтон — столица США;
 - в) чтобы найти площадь прямоугольника, нужно его ширину умножить на длину;
 - г) для определения числа решений квадратного уравнения, нужно вычислить его дискриминант: если он меньше 0, то решений нет; если равен 0, то решение одно; если больше 0, то решений — 2;
 - д) А. С. Пушкин — автор поэмы «Полтава»;

- е) Н. Винер — основатель кибернетики;
 - д) чтобы сократить дробь, нужно найти наибольший общий делитель числителя и знаменателя и разделить на него и числитель, и знаменатель;
 - е) для удаления текущего файла с помощью Norton Commander, нужно нажать клавишу F8;
 - ж) Юрий Гагарин — первый космонавт.
2. В какой форме человек воспринимает информацию (образной или знаковой), если он:
- а) слушает оперу;
 - б) смотрит футбольный матч;
 - в) читает орфографический словарь;
 - г) выбирает букет в магазине;
 - д) ест мороженое;
 - е) разговаривает по телефону;
 - ж) гладит кошку;
 - з) слушает по радио сообщение о курсе доллара?
3. Составьте таблицу способов символьного представления информации в перечисленных областях. Если знаков много, то запишите или нарисуйте 5 из них:

Область применения	Используемые символы
Математика	
Музыка	
Медицина	
География	
Литература	

4. [2]: раздел 1.3, задача № 1.

Домашнее задание № 3

1. Опишите, какие действия с информацией (информационные процессы) будут выполнены учеником, если он:
- а) учит правило;

- б) решает у доски задачу;
 - в) слушает музыку;
 - г) пишет письмо;
 - д) переводит текст с английского языка на русский;
 - е) пишет диктант.
2. Опишите, кто или что будет являться *источником* и кто или что — *приемником* информации в следующих ситуациях:
- а) Андрей собирается переходить перекресток, регулируемый светофором,;
 - б) Петя беседует с Колей по телефону;
 - в) Аня слушает прогноз погоды по радио;
 - г) Женя учит стихотворение А. С. Пушкина «Анчар»;
 - д) Света едет в автобусе до остановки «Парк культуры» в незнакомом городе.
3. Какие информационные носители вы используете в своей жизни?
4. [2]: раздел 1.2, задача № 8.

Домашнее задание № 4

1. Какие средства хранения информации были изобретены в XIX—XX веках?
2. Перечислите основные вычислительные средства с указанием времени их изобретения.
3. В чем основная заслуга Чарльза Бэббиджа?

Домашнее задание № 5

1. Подсчитать в килобайтах количество информации в тексте, если текст состоит из 800 символов, а мощность используемого алфавита — 128 символов.
2. Имеется два текста на разных языках. Первый текст использует 32-символьный алфавит и содержит 200 символов, второй — 16-символьный алфавит и содержит 250 символов. Какой из текстов содержит большее количество информации и на сколько битов?

3. Сколько символов в тексте, если мощность алфавита — 64 символа, а объем информации, содержащейся в нем, — 1,5 килобайта?
4. Скорость информационного потока — 20 бит/сек. Сколько времени потребуется для передачи информации объемом в 10 килобайт?
5. Сравните (поставьте знак отношения):
 - а) 200 байт и 0,25 килобайта;
 - б) 3 байта и 24 бита;
 - в) 1536 битов и 1,5 килобайта;
 - г) 1000 битов и 1 килобайт;
 - д) 8192 байта и 1 Кбайт.

Домашнее задание № 6

1. Какие из приведенных ниже чисел не являются правильной записью в пятеричной системе счисления? Обоснуйте свой ответ.
304, 100002, 253, 1118, 300, 4444.
2. [5]: § 6, вопрос 6.
3. [5]: § 6, вопрос 7.
4. [5]: § 6, вопрос 8.

Домашнее задание № 7

1. Саша набирает текст на компьютере. Куда первоначально заносится набираемый текст?
2. Заполните таблицу:

Устройство	Действие с информацией (хранение, ввод, вывод, обработка)
Процессор	
Оперативная память	
Жесткий диск	
Дискета	
Клавиатура	
Монитор	
Принтер	

3. Какая информация хранится в оперативной памяти?
4. Почему оперативная память не может использоваться для длительного хранения данных?

Домашнее задание № 8

1. Выберите из предложенного списка программы, относящиеся к: а) системному ПО; б) прикладному ПО:
 - ☐ программа обучения английскому языку;
 - ☐ текстовый редактор Word 97;
 - ☐ операционная система MS DOS;
 - ☐ антивирусная программа DrWeb 4.05;
 - ☐ операционная система Windows'95;
 - ☐ клавиатурный тренажер;
 - ☐ программа-архиватор WinZip;
 - ☐ компьютерная игра DOOM II;
 - ☐ графический редактор CorelDRAW;
2. Перечислите основные функции операционной системы.
3. Без какого типа программного обеспечения работа компьютера невозможна?

Домашнее задание № 9

1. Что такое пользовательский интерфейс?
2. Перечислите все известные вам значения поведения объекта *«окно»*.
3. Назовите известные вам типы объектов ОС Windows (не менее пяти).

Домашнее задание № 10

1. Впишите в пустые ячейки таблицы следующие характеристики внешней (жесткого диска) и внутренней памяти компьютера: *быстрый доступ, энергонезависимая, медленный доступ, средний объем от 4 до 64 Мбайт, энергозависимая, средний объем от 1 до 12 Гбайт.*

Примерный вид таблицы:

Характеристика	Внутренняя память	Внешняя память
Скорость доступа		
Энергозависимость		
Средний объем		

2. Сколько Кбайт памяти требуется для хранения двоичного кода, состоящего из 6000 нулей и 2192 единиц?
3. Что может храниться в одном бите памяти компьютера?

Домашнее задание № 11

1. Какие из перечисленных ниже устройств входят в состав минимальной конфигурации компьютера: принтер, процессор, клавиатура, звуковая карта, монитор, оперативная память, устройство CD-ROM.
2. Для чего нужны контроллеры внешних устройств?
3. У Андрея на жестком диске компьютера есть игра Tetris. У его друга Коли такой игры нет. Что нужно сделать Андрею, чтобы и Коля смог на своем домашнем компьютере играть в эту игру?

Домашнее задание № 12

1. В каталоге зарегистрированы файлы: ball.exe, ball.txt, ball.bmp, ball.mid.
 - а) Какой файл нужно запустить на выполнение, чтобы начать игру BALL?
 - б) В каком файле может храниться инструкция к игре?
 - в) В каком файле может храниться заставка к игре?
 - г) В каком файле хранится музыкальное сопровождение к игре?

2. На экран выведено содержимое каталога BALL:

ball.exe	35600	10.05.93	10:30
ball.txt	2050	12.05.93	18.50
ball.rez	1700	25.09.96	14.25

Какую информацию о файле ball.exe можно получить из этого списка?

3. [2]: раздел 3.1, задача № 29.

4. [2]: раздел 3.1, задача № 25.

Домашнее задание № 13

1. Напишите шаблоны, с помощью которых можно выделить:

- а) все файлы, имена которых начинаются на букву «t», а расширения состоят из двух символов;
- б) все файлы, имена которых состоят из трех символов и заканчиваются цифрой «3».

2. Какие файлы будут выделены при использовании шаблонов (опишите имена и расширения файлов в общем виде аналогично описанию из задачи № 1):

- а) a?b.ах б) *.?b в) cb.*

3. В чем преимущество использования ярлыков?

Домашнее задание № 14

1. Опишите последовательность действий, которые необходимо выполнить, чтобы скопировать информацию из одного документа в другой.

2. Оля набрала следующий текст:

*В лесу родилась елочка,
В лесу она росла.*

Затем она выполнила следующие действия:

- 1) выделила слово «елочка»;
- 2) скопировала его в буфер обмена;
- 3) выделила слово «росла»;
- 4) скопировала его в буфер обмена;
- 5) перевела курсор в конец второй строки;
- 6) выбрала пункт меню [Правка-Вставить].

Что будет отображено на экране после выполнения этих действий?

3. Петя открыл файл RABBIT, в котором содержится следующий текст:

Раз, два, три, четыре, пять.

Петя добавил к этому тексту строку:

Вышел зайчик погулять.

Затем он выбрал команду [Сохранить как...] и ввел с клавиатуры имя файла MY_RABBIT. Какие тексты будут храниться в файлах RABBIT и MY_RABBIT?

Домашнее задание № 15

1. [2]: раздел 3.1, задача № 32.
2. [2]: раздел 3.1, задача № 35.
3. [2]: раздел 3.1, задача № 36.
4. Сколько букв и какого алфавита содержит зашифрованный десятичными кодами текст: 128 32 139 32 148 32 128 32 130 32 136 32 146? Какой объем памяти занимает данный текст?

Решите данную задачу, не заглядывая в таблицу кодировки.

Домашнее задание № 16

1. Даны три файла со следующим содержимым:

а) файл № 1:

$$Y = \sqrt{\frac{x^2 + 1}{4}}$$

б) файл № 2:

Текстовый файл — простейший способ организации данных в компьютере.

в) файл № 3:



Какие из этих файлов относятся к текстовым, а какие — нет? Поясните свой ответ.

2. Выберите из предложенного списка *минимальный* набор функций, которые должен выполнять любой текстовый редактор:

- ☐ сохранение и загрузка файлов;
- ☐ работа с графикой;
- ☐ форматирование текста;
- ☐ редактирование текста;
- ☐ работа с математическими формулами;
- ☐ печать текста;
- ☐ проверка правописания;
- ☐ создание таблиц

Домашнее задание № 17

1. Сколько слов (с точки зрения ТР) содержится в следующем тексте, содержащем ряд ошибок:

Белый снег пушистый в воздухе; кружится
И на землю тихо... падает, ложится.

2. После ввода слова текстовый редактор выделил его как ошибочное. Какие действия нужно выполнить, чтобы данное слово ошибочным не считалось (перечислите все возможные варианты ответа)?
3. Какие действия можно выполнить с фрагментом данных?
4. Какие из перечисленных ниже действий выполняются в режиме форматирования:
- ☐ сохранение файла;
 - ☐ центрирование заголовка;
 - ☐ орфографическая проверка;
 - ☐ выравнивание абзаца по ширине;
 - ☐ поиск слова в тексте;
 - ☐ установка границы «красной строки» абзаца?

Домашнее задание № 18

1. Перечислите основные характеристики (растровый или векторный, пропорциональный или непропор-

циональный, рубленый или с засечками) шрифта, которым набран следующий текст:

Microsoft Office'2000

2. Вырежьте из газеты или журнала и приклейте в тетрадь образцы текстов, набранных: а) рубленым шрифтом; б) шрифтом с засечками.
3. Когда лучше использовать растровые шрифты?
4. Назовите известные вам векторные шрифты (2-3 примера).

Домашнее задание № 19

1. Специалисты каких профессий используют графические пакеты (назовите не менее 5-6 профессий)?
2. К какой области компьютерной графики относятся графические пакеты для:
 - а) получения движущихся изображений;
 - б) подготовки чертежей;
 - в) построения графиков;
 - г) графического представления результатов научных экспериментов;
 - д) построения диаграмм;
 - е) создания иллюстраций к книгам;
 - ж) создания видеопрезентаций?
3. Какие графические примитивы будут использоваться для получения приведенного ниже изображения?



Домашнее задание № 20

1. У Саши есть друг в Германии, с которым он общается с помощью электронной почты. Они решили

- обмениваться фотографиями. Какое устройство необходимо Саше, чтобы он мог послать свою фотографию в виде файла?
2. Какие устройства имеют доступ к видеопамяти?
 3. Из каких основных цветов формируется вся палитра?
 4. Что хранится в видеопамяти?
 5. По какому принципу работают *растровые дисплеи*?

Домашнее задание № 21

1. Какой объем видеопамяти в килобайтах нужен для хранения изображения размером 600×350 пикселей и использующего 8-цветную палитру?
2. Придумайте способ кодировки цветов для 64-цветной палитры.
3. Сколько цветов можно максимально использовать для хранения изображения размером 350×200 пикселей, если объем страницы видеопамяти — 65 Кбайт?
4. Сколько цветов будет в палитре, если каждый базовый цвет кодировать в 3 битах?

Практические задания для выполнения на компьютере

Практическое задание № 1

1. Набрать текст:

С 1980 года «законодателем мод» на рынке ПК становится американская фирма IBM. Конструкторам IBM удалось создать такую архитектуру, которая стала фактически международным стандартом. В конце 80-х — начале 90-х годов большую популярность приобрели компьютеры фирмы Apple Corporation марки Macintosh.

2. Набрать свою визитную карточку (фамилия, имя, отчество, адрес, телефон, школа, класс).
3. Набрать текст:
*Yesterday all my troubles seemed
so far away. Now it looks as though they're
here to stay, Oh, I believe in yesterday.*
4. Привести набранный в задании 3 текст к следующему виду:
*Yesterday all my troubles seemed so far away.
Now it looks as though they're here to stay,
Oh, I believe in yesterday.*

Практическое задание № 2

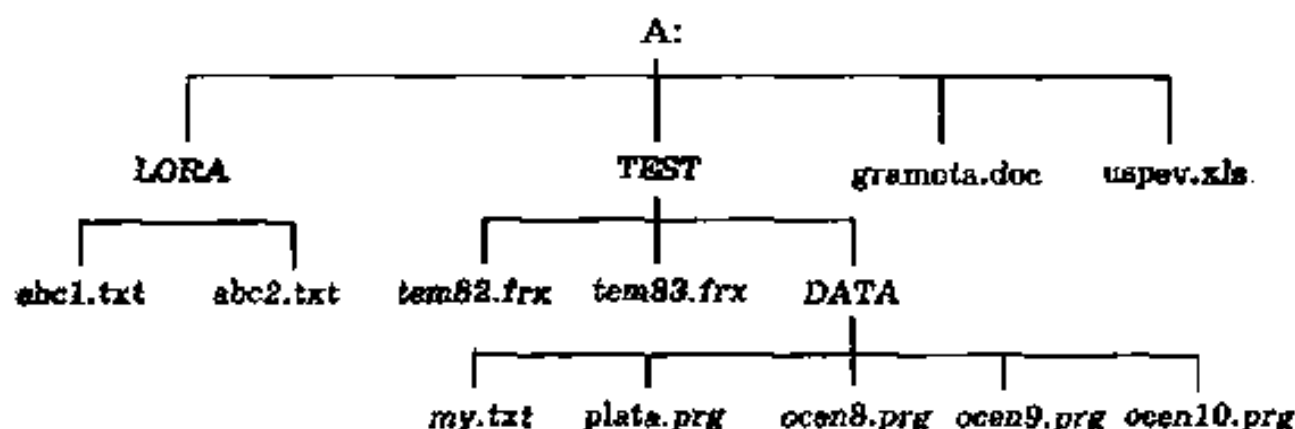
1. Запустить программу WordPad.
2. Установить размеры окна Word Pad в 1/4 часть экрана (1/2 высоты и 1/2 ширины экрана) и разместить его в левом верхнем углу рабочего стола.
3. Запустить программу Калькулятор и разместить окно этой программы в правом верхнем углу Рабочего стола.
4. Запустить программу Paint и разместить окно этой программы в левом нижнем углу рабочего стола.
5. Переключиться в программу WordPad и развернуть окно программы на полный экран.
6. Переключиться в программу Калькулятор и свернуть окно этой программы.
7. Завершить работу программы Paint.
8. Восстановить прежние размеры окна программы WordPad.
9. Завершить работу программы Калькулятор.
10. Завершить выполнение программы WordPad.

Практическое задание № 3

1. Создать в папке SCHOOL диска C папку с именем NEW.
2. Скопировать в папку NEW с дискеты папку LORA и файлы gramota.doc и uspev.xls.

3. Переименовать папку LORA на диске C в папку STAR.
4. Переместить папку STAR в корневой каталог диска A.
5. Удалить с диска C папку NEW, а с диска A — папку STAR.
6. Очистить Корзину.
7. Открыть на диске A папку TEST\DATA и вывести ее содержимое в полной форме (Таблица), отсортировав по размеру.

Примечание. Перед занятием необходимо создать на диске C папку SCHOOL и подготовить дискету со следующей файловой структурой:



Практическое задание № 4

1. Создать в папке SCHOOL диска C папку MAIN.
2. Вывести на экран содержимое папки A:\TEST\DATA и выделить файлы ocen8.prg, ocen10.prg и my.txt.
3. Скопировать выделенные файлы в каталог MAIN на диске C.
4. Заменить у скопированных файлов расширение prg на расширение doc.
5. Создать в корневом каталоге диска A папку DOC.
6. Переместить все файлы из папки C:\SCHOOL\MAIN в папку A:\DOC.

7. Удалить из папки C:\SCHOOL каталог MAIN, а с диска A — папку DOC.
8. Найти на диске C все файлы с расширением, начинающимся на букву «s».
9. Найти на диске C все файлы, имя которых начинается с символа «п», а расширение — с символа «е».
10. Создать на Рабочем столе ярлык программы write.exe и переименовать его в «WordPad».
11. Запустить с помощью созданного ярлыка программу WordPad.
12. Закрыть все окна.

Примечание. Перед занятием необходимо создать на диске C папку SCHOOL и подготовить дискету с файловой структурой, приведенной в п/з № 3.

Практическое задание № 5

1. Запустить программу Блокнот.
2. Набрать приведенный ниже текст (для ускорения набора использовать копирование):
*Маленькие ножки,
Маленькие ножки!
За водой ходили
Маленькие ножки
И домой спешили
Маленькие ножки,
Дома танцевали
Маленькие ножки,
Ой, как танцевали
Маленькие ножки!*
3. Сохранить набранный текст в заданной преподавателем папке под именем роеша.
4. Закрыть Блокнот.
5. Запустить программу Калькулятор.
6. Вычислить значение произведения 45·33.
7. Скопировать полученный результат в буфер обмена.
8. Запустить программу Paint.

9. Вставить содержимое буфера обмена в окно графического редактора.
10. Заккрыть окно графического редактора без сохранения.
11. Запустить программу Блокнот и открыть файл роета.
12. Вставить в конец текста содержимое буфера обмена.
13. Сохранить файл под новым именем роета1.
14. Удалить с диска оба созданных файла (роета и роета1).

Практическое задание № 6

1. Создать в папке SCHOOL диска C папку NEW.
2. Скопировать в папку NEW с диска A папку TEST\DATA.
3. Переименовать скопированную папку DATA в папку PUPIL.
4. Переместить папку PUPIL в папку C:\SCHOOL.
5. Вывести на экран содержимое папки PUPIL в виде таблицы (полная форма), отсортировав по времени создания.
6. Открыть окно программы Калькулятор и вычислить значение произведения $564 \cdot 35$.
7. Скопировать полученное значение в буфер обмена.
8. Открыть окно программы Paint и вставить в него содержимое буфера обмена.
9. Сохранить данные в папке NEW под именем otvet.
10. Создать на Рабочем столе ярлык папки NEW.
11. Заккрыть все окна.
12. Открыть с помощью ярлыка папку NEW.
13. Открыть файл otvet.bmp.
14. Сохранить открытый документ под новым именем answer и закрыть окно документа.
15. Удалить из папки SCHOOL папку NEW и папку PUPIL, с Рабочего стола — ярлык папки NEW.

16. Найти на диске С все файлы, созданные за последние два месяца.
17. Заккрыть все окна.

Примечание. Перед занятием необходимо создать на диске С папку SCHOOL и подготовить дискету с файловой структурой, приведенной в п/з № 3.

Практическое задание № 7

1. Запустить программу текстового редактора.
2. Вывести на экран панели «Стандартная», «Рисование», «Таблицы и границы» и «Форматирование», остальные панели убрать.
3. На панель «Форматирование» добавить кнопки «Верхний индекс» и «Нижний индекс».
4. На панель «Стандартная» добавить кнопки «Разрыв страницы» и «Номер страницы».
5. Скрыть панели «Рисование» и «Таблицы и границы».
6. Удалить кнопки «Верхний индекс», «Нижний индекс», «Разрыв страницы», «Номер страницы».
7. Завершить работу текстового редактора.

Практическое задание № 8

1. Запустить программу текстового редактора.
2. Открыть файл ROMEO.
3. Привести текст в соответствие с образцом, исправив орфографические ошибки и выполнив построчную разбивку (см. [2], раздел 5.1, индивидуальная работа № 2, вариант 1).
4. Завершить работу текстового редактора.

Примечание. Перед началом работы необходимо подготовить файл ROMEO из приложения к [2].

5. [2]: раздел 5.1, индивидуальная работа № 1, вариант 1.

Практическое задание № 9

1. [2]: раздел 5.1, индивидуальная работа № 4, вариант 1.

Примечание. Перед началом работы необходимо подготовить файл AZBUKA из приложения [2].

Практическое задание № 10

1. Запустить текстовый редактор и открыть файл CSAR.
2. оформить текст согласно следующим указаниям (см. ниже образец):
 - заголовок — шрифт TimesNewRoman, размер — 26, цвет — красный, использовать центрирование и подчеркивание;
 - первый абзац — шрифт TimesNewRoman, размер — 14; использовать курсив, полужирный шрифт и подчеркивание;
 - имя царя (Алексей Михайлович) — шрифт Arial, размер 18, цвет — синий;
 - годы жизни царя (Алексей Михайлович) — шрифт Arial, размер 12;
 - имена цариц (Мария Ильинична Милославская и Наталья Кирилловна Нарышкина) — шрифт Arial, размер 18, цвет — голубой;
 - годы жизни цариц — шрифт Arial, размер 12;
 - имена и годы жизни детей — шрифт Courier New, размер 12;
 - во всем тексте в соответствии с образцом использовать курсив, подчеркивание и полужирный шрифт.

Династическая ветвь

27 апреля 1682 г. умер царь Федор Алексеевич. Ему было всего 20 лет. Слабый и больной, он вступил на престол после своего отца — царя Алексея Михайловича в 1676 г. и правил всего 6 лет. И хотя Федор же-

нился дважды, детей у него не было. Боярская дума, собравшаяся в Кремле после смерти царя, должна была решать, кому стать русским самодержцем. Кандидатов было двое — 16-летний царевич Иван и 10-летний царевич Петр. Оба они были детьми царя Алексея, но от разных матерей. Вот здесь обратимся к *династической ветви* — фрагменту династического древа Романовых, чтобы разобраться во всех хитросплетениях проблемы наследия престола.

Алексей Михайлович (1629–1676, царь с 1645)

Мария Ильинична

Милославская

(1626–1669)

Дмитрий (1648–1649)

Евдокия (1650–1712)

Марфа (1652–1707)

Алексей (1654–1670)

Анна (1655–1659)

Софья (1657–1704)

Екатерина (1658–1718)

Мария (1660–1723)

Федор (1661–1682, царь с 1676)

Симеон (1665–1669)

Иван (1666–1696, царь с 1682)

Евдокия (1669–1669)

Наталья Кирилловна

Нарышкина

(1651–1694)

Наталья (1673–1716)

Феодора (1674–1678)

Петр (1672–1725, царь с 1682)

3. Заменить во всем тексте слово «царь» на «csar».

Примечание. Перед занятием необходимо подготовить файл CSAR (см. приложение к [2]).

Практическое задание № 11

1. Запустить текстовый редактор и открыть файл RECEPT.
2. Установить следующие параметры страницы: отступ слева — 1,5 см, отступ справа — 2 см, отступ сверху — 2 см, отступ снизу — 2,5 см.
3. С помощью различных параметров форматирования привести текст согласно указаниям в соответствии с образцом (см. в конце задания):
 - заголовок («Винегрет овощной») отцентрировать;
 - список продуктов выровнять по левому краю;
 - в первом после списка продуктов абзаце установить абзацный отступ — 1,5 см, левую границу — 0 см, правую границу — 13 см;
 - во втором после списка продуктов абзаце установить абзацный отступ — 9 см, левую границу — 8 см, правую границу — 16 см.
4. Скопировать набранный текст ниже 2 раза (всего должно получиться 3 рецепта).
5. Сохранить текст в файле VINEGRET в текущем каталоге.

Винегрет овощной

- ☐ Картофель — 3 шт
- ☐ Морковь — 2 шт
- ☐ Свекла — 1 шт
- ☐ Соленые огурцы — 2 шт
- ☐ Лук зеленый — 50 г
- ☐ Масло растительное — 2 ст. ложки
- ☐ Перец молотый, горчица, укроп — по вкусу
- ☐ Листья салата

Огурцы, вареный картофель, свеклу, морковь нарезать тонкими ломтиками, лук нашинковать. Овощи выложить в посуду, перемешать, заправить маслом с добавлением перца, соли, горчицы. Готовый винегрет поставить в холодильник.

При подаче на стол винегрет уложить горкой в салатник, украсить зеленым салатом, посыпать укропом.

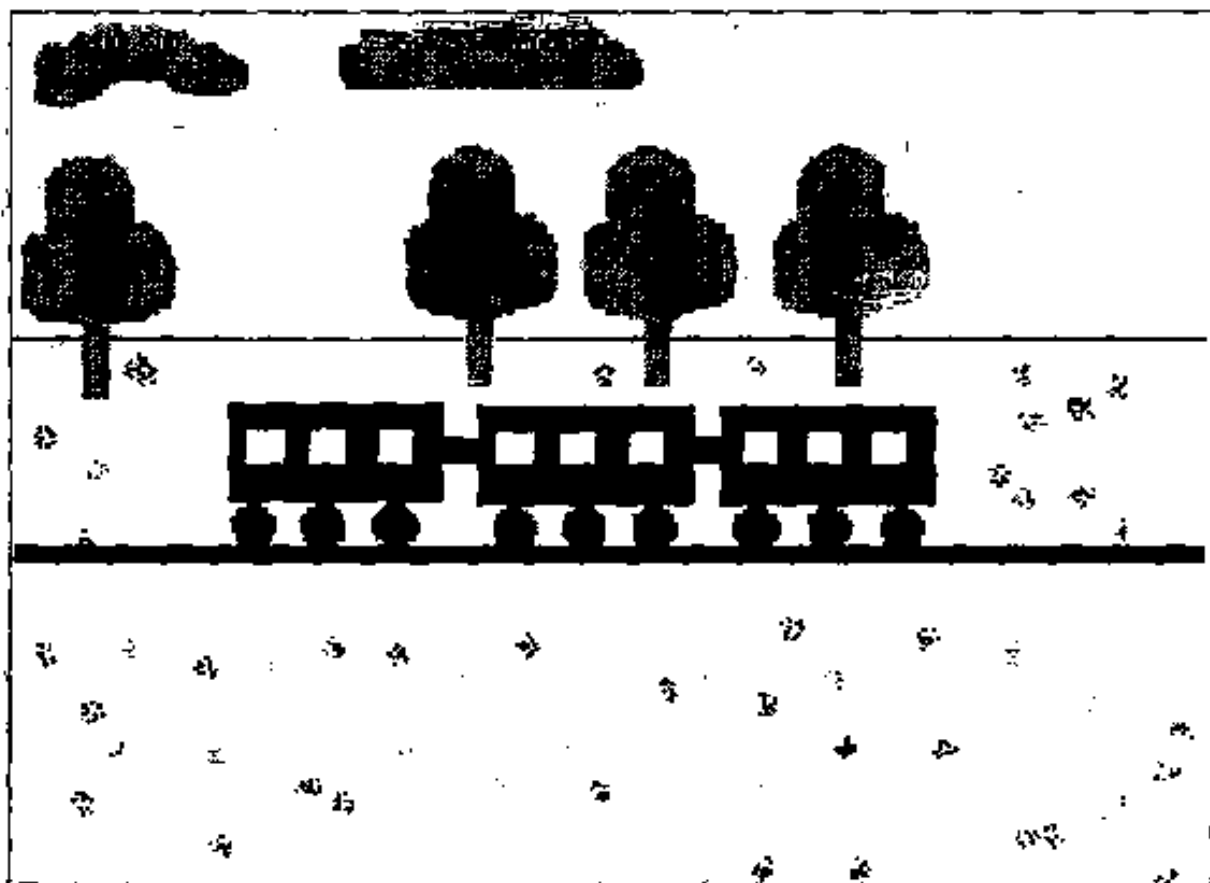
Примечание. Перед занятием необходимо подготовить файл РЕСЕРТ (см. приложение к [2]).

Практическое задание № 12

1. [2]: раздел 5.1, упражнение № 40.
2. Провести орфографическую проверку набранного текста.
3. Заменить слово «артист» на слово «artist».
4. Сохранить набранный текст в указанном преподавателем каталоге под именем алпа.
5. Скопировать набранный текст в новое окно текстового редактора.
6. Удалить вторую половину текста (со слова «Дирижер»).
7. Сохранить оставшийся текст в указанном преподавателем каталоге под именем геслама.
8. Удалить с диска оба созданных файла.

Практическое задание № 13

1. Изобразить следующий рисунок:



Практическое задание № 14

1. Изобразить следующий рисунок:



flower

Тесты**Тест № 1**

1. Информация, хранимая в книгах, на магнитных носителях, грампластинках называется:
А) архивной; С) полезной; Е) оперативной.
В) внутренней; D) внешней.
2. Образная информация, которую можно хранить на внешних носителях, — это:
А. звуковые образы;
В. изображение и звук;
С. осязательные образы;
D. текст, записанный на каком-либо языке;
Е. только изображение.
3. Ученик пишет изложение. Какие информационные процессы выполняются учеником?
А) Прием и отправление;
В) прием и обработка;
С) прием, обработка и хранение;
D) прием, хранение, отправление;
Е) прием, обработка, хранение, отправление.

4. Ваня учится в 1 классе и хорошо знает таблицу умножения, но не знает английского языка. Какое из сообщений будет для Вани информативным?
- А) $2 \times 8 = 16$;
В) MY FRIEND IS SCHOOLBOY;
С) Ваня учится в школе;
D) в английском алфавите 26 букв;
Е) 6 MULTIPLY 8 EQUAL 48.
5. Из предложенных сообщений выбрать декларативную информацию:
- А) бит — единица измерения информации;
В) сумма углов в треугольнике равна 180 градусам;
С) А. С. Пушкин — великий русский поэт;
D) окружность — множество точек, равноудаленных от центра;
Е) при умножении простых дробей их числители и знаменатели перемножаются.
6. В течение 5 минут со скоростью 20 байт/с вождь племени Мульти передавал информационное сообщение. Сколько символов оно содержало, если алфавит племени состоит из 32 символов?
- А) 9600; В) 3200; С) 1200; D) 1500; Е) 150.
7. 1,5 мегабайта равны:
- А) 1536 Кбайт; С) 1536 байт; Е) 0,015 Гбайт;
В) 1500 Кбайт; D) 1500 байт.
8. Информационное сообщение объемом 1,5 Кбайт содержит 3072 символа. Сколько символов содержит алфавит, с помощью которого записано это сообщение?
- А) 16; В) 128; С) 32; D) 64; Е) 256.
9. Результат сложения двух чисел, записанных римскими цифрами MCM + LXIV, равен:
- А) 1964; В) 2164; С) 154; D) 2014; Е) 1164.
10. Число 30111011 может существовать в системах счисления с основаниями:
- А) 2 и 3; В) 4 и 3; С) 3 и 10; D) 4 и 8; Е) 2 и 4.

Ответы на тест № 1

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	D	B	E	D	E	A	A	A	A	D

Тест № 2

1. Перед отключением компьютера информацию можно сохранить:
A) в оперативной памяти;
B) во внешней памяти;
C) в регистрах процессора;
D) на дисковом; *Е) в контроллере магнитного диска.*
2. Электронный блок, управляющий работой внешнего устройства, называется:
A) адаптером (контроллером);
B) драйвером;
C) регистром процессора;
D) общей шиной;
E) интерфейсом.
3. Наименьшая адресуемая часть памяти компьютера:
A) бит;
B) файл;
C) килобайт;
D) байт;
E) ячейка.
4. «Каталог содержит информацию о ..., хранящихся в ...». Вместо многоточий надо вставить слова:
A) программах, оперативной памяти;
B) файлах, оперативной памяти;
C) программах, внешней памяти;
D) файлах, внешней памяти;
E) программах, процессоре.
5. Драйвер — это:
A) устройство длительного хранения информации;

- В) программа, управляющая конкретным внешним устройством;
 - С) устройство ввода;
 - Д) устройство, позволяющее подсоединить к компьютеру новое внешнее устройство;
 - Е. устройство вывод.
6. Во время работы компьютера в оперативной памяти постоянно находится:
- А) ядро операционной системы;
 - В) вся операционная система;
 - С) прикладное программное обеспечение;
 - Д) система программирования;
 - Е) программа-архиватор.
7. Информацию из оперативной памяти можно сохранить на внешнем запоминающем устройстве в виде:
- А) блока;
 - В) каталога;
 - С) директории;
 - Д) программы;
 - Е) файла.
8. Шаблон, который используется для поиска файлов, имена которых начинаются с символа «F»:
- А) F?.exe;
 - В) F?.???;
 - С) F*.*;
 - Д) F*.exe;
 - Е) *F.bat.
9. Приложение выгружается из памяти и прекращает свою работу, если:
- А) запустить другое приложение;
 - В) свернуть окно приложения;
 - С) переключиться в другое окно;
 - Д) переместить окно приложения;
 - Е) закрыть окно приложения.

10. Панель задач служит для:
- А) переключения между запущенными приложениями;
 - В) завершения работы Windows;
 - С) обмена данными между приложениями;
 - Д) запуска программ MS DOS;
 - Е) просмотра каталогов.
11. Файл tetris.com находится на диске С в каталоге GAMES, который является подкаталогом каталога DAY. Полное имя файла:
- А) C:\tetris.com\GAMES\DAY;
 - В) C:\GAMES\tetris.com;
 - С) C:\DAY\GAMES\tetris.com;
 - Д) C:\GAMES\DAY\tetris.com;
 - Е) C:\GAMES\tetris.com.
12. Предложен набор команд:
1. создать файл home.txt;
 2. создать каталог TOWN;
 3. создать каталог STREET;
 4. войти в созданный каталог;
 5. сделать диск А текущим.
- Чтобы был получен алгоритм, с помощью которого на пустой дискете создается файл с полным именем A:\TOWN\STREET\home.txt, эти команды следует расположить так:
- А) 5; 2; 4; 3; 4; 1;
 - В) 5; 2; 3; 1;
 - С) 5; 1; 3; 4; 2;
 - Д) 5; 1; 2; 3; 4;
 - Е) 1; 3; 2; 5.
13. «... памяти означает, что любая информация заносится в память и извлекается из нее по ...». Вместо многоточий надо вставить слова:
- А) дискретность; адресам;
 - В) адресуемость; значениям;
 - С) дискретность; битам;
 - Д) адресуемость; байтам;
 - Е) адресуемость; адресам;

14. В прикладное ПО входят:
- А) языки программирования;
 - В) операционные системы;
 - С) диалоговая оболочка;
 - Д) совокупность всех программ, установленных на компьютере;
 - Е) текстовые редакторы.
15. «Программа, хранящаяся во внешней памяти, после вызова на выполнение попадает в ... и обрабатывается ...». Вместо многоточий надо вставить слова:
- А) устройство ввода; процессором;
 - В) процессор; регистрами процессора;
 - С) процессор; процессором;
 - Д) оперативную память; процессором;
 - Е) файл, процессором.

Ответы на тест № 2

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	В	А	Д	Д	В	А	Е	С	Е	А
№ вопроса	11	12	13	14	15					
Ответ	С	А	Е	Е	Д					

Тест № 3

1. Координаты курсора текстового редактора фиксируются:
 - А) в меню текстового редактора;
 - В) в словаре текстового редактора;
 - С) в строке состояния текстового редактора;
 - Д) в окне текстового редактора;
 - Е) в буфере для копирования.
2. Во время работы текстового редактора орфографический словарь:
 - А) по мере необходимости загружается во внешнюю память;
 - В) постоянно находится в устройстве ввода;
 - С) постоянно находится в оперативной памяти;

- D) по мере необходимости загружается в оперативную память;
 - E) постоянно находится в устройстве вывода.
3. Таблица кодировки символов устанавливает соответствие между:
- A) символами, их десятичными номерами и двоичными кодами;
 - B) символами разных алфавитов;
 - C) символами и количеством байтов, которые они занимают;
 - D) символами и номерами ячеек памяти, которые они занимают;
 - E) символами и клавишами.
4. К устройствам вывода графической информации относится:
- A) дисплей;
 - B) мышь;
 - C) клавиатура;
 - D) сканер;
 - E) графический редактор.
5. К форматированию текста относятся следующие действия:
- A) копирование фрагментов текста;
 - B) исправление опечаток;
 - C) проверка орфографии;
 - D) изменение абзацных отступов;
 - E) перемещение фрагментов текста.
6. При выполнении следующей(их) команды(команд) выделенный фрагмент текста попадает в буфер обмена:
- A) вставить;
 - B) заменить;
 - C) проверить орфографию;
 - D) выравнивать по центру;
 - E) вырезать и копировать.
7. Текст занимает 0,25 Кбайт памяти компьютера. Сколько символов содержит текст?

- A) 256;
 - B) 2048;
 - C) 32;
 - D) 250;
 - E) 2000.
8. Номера символов «Z» и «Б» в таблице кодировки ASCII соответственно равны:
- A) 80 и 90;
 - B) 130 и 140;
 - C) 90 и 129;
 - D) 160 и 50;
 - E) 10 и 250;
9. Объем страницы видеопамати — 125 Кбайт. Графический дисплей работает с 16-цветной палитрой. Какова разрешающая способность графического дисплея?
- A) 640×200;
 - B) 320×400;
 - C) 640×400;
 - D) 640×800;
 - E) 512×400.
10. Для кодирования зеленого цвета служит код 010. Сколько цветов содержит палитра?
- A) 4; B) 3; C) 8; D) 16; E) 6.
11. Код (номер) буквы «j» в таблице кодировки символов равен 106. Какая последовательность кодов будет соответствовать слову «file»? Алфавит латинских букв: a b c d e f g h i j k l m n o p ...
- A) 110 107 104 111;
 - B) 74 98 120 66;
 - C) 132 112 90 140;
 - D) 102 105 108 101;
 - E) 90 102 114 86.
12. Последовательность кодов 105 162 109 таблицы кодировки шифрует некоторые символы. Среди этих символов:
- A) не может быть букв русского алфавита;

- В) не может быть букв латинского алфавита;
 - С) могут быть буквы русского и латинского алфавитов;
 - Д) не может быть букв русского и латинского алфавитов;
 - Е) может быть только одна буква латинского алфавита.
13. Для хранения текста требуется 84 000 битов. Сколько страниц займет этот текст, если на странице размещается 30 строк по 70 символов в строке?
А) 40; В) 5; С) 50; Д) 400; Е) 350.
14. Объем видеопамати составляет 62,5 Кбайт. Графический дисплей работает в режиме 640×400 пикселей. Какое максимальное количество цветов может содержать палитра?
А) 4; В) 2; С) 8; Д) 16; Е) 32.
15. Для удаления неверно набранного символа используется клавиша:
А) Ctrl;
В) Esc;
С) Num Lock;
Д) Insert;
Е) Delete.
16. Окно текстового редактора содержит 25 строк по 60 символов в строке. Окно графического редактора, работающего в двухцветном режиме, — 80 × 25 пикселей. Сравнить объемы памяти, необходимой для хранения текста, занимающего все окно текстового редактора, и для кодирования картинки, занимающей все поле графического редактора:
А) для текста памяти требуется больше, чем для картинки;
В) для текста памяти требуется меньше, чем для картинки;
С) для текста памяти требуется столько же, сколько для картинки;

- Д) недостаточно информации, чтобы ответить на вопрос.
17. Какие символы могут быть зашифрованы кодами таблицы ASCII 119 и 235?
- А) «д» и «ш»;
 - В) «j» и «s»;
 - С) «d» и «D»;
 - Д) «я» и «t»;
 - Е) «w» и «ы».
18. Часть книги представлена цветными изображениями в 16-цветной палитре в формате 320×640 точек. Страницы, содержащие текст, имеют формат — 64 строки по 48 символов в строке. Определить максимально возможное количество страниц, которые можно сохранить на жестком магнитном диске объемом 40 Мбайт, если количество страниц, представленных цветными изображениями, на 80 больше количества страниц, содержащих только текст.
- А) 400; В) 720; С) 320; Д) 100; Е) 530.
19. «Содержимое видеопамати непрерывно просматривается ... и выводится ... на экран». Вместо многоточий надо вставить слова:
- А) дисплейным процессором, центральным процессором;
 - В) центральным процессором, дисплейным процессором;
 - С) дисплейным процессором, дисплейным процессором;
 - Д) центральным процессором, графическим адаптером;
 - Е) графическим адаптером, центральным процессором.
20. В текстовом редакторе можно использовать только один шрифт и две кнопки, служащие для установки таких эффектов, как полужирное начертание и курсив. Сколько различных начертаний символов можно получить?

- A) 1; B) 2; C) 3; D) 4; E) 6.
21. В течение 5 минут со скоростью 10 байт/сек вождь племени Мульти передавал информационное сообщение. Сколько символов оно содержало, если алфавит племени состоит из 32 символов?
A) 4800; B) 2400; C) 600; D) 60; E) 80.
22. Информационное сообщение объемом 3 килобайта содержит 1024 символа. Сколько символов содержит алфавит, с помощью которого записано это сообщение?
A) 16; B) 128; C) 8; D) 64; E) 256.
23. *Каталог содержит информацию о..., хранящихся в...». Вместо многоточий надо вставить слова:
A) программах, оперативной памяти;
B) файлах, оперативной памяти;
C) программах, внешней памяти;
D) файлах, внешней памяти;
E) программах, процессоре.
24. Программа, управляющая работой конкретного внешнего устройства, называется:
A) адаптером;
B) драйвером;
C) командным процессором;
D) архиватором;
E) контроллером.
25. Файл line.exe находится на диске C в каталоге GAMES, который является подкаталогом каталога MY. Полное имя файла:
A) C:\line.exe\GAMES\MY;
B) C:\GAMES\lines.exe;
C) C:\MY\GAMES\lines.exe;
D) C:\GAMES\MY\lines.exe;
E) C:\GAMES\lines.exe.
26. С помощью рубленого векторного шрифта набрано слово:

A) **КРАБ** ;

- В) **?????**;
С) **КРАБ**;
D) **КРАБ**;
Е) **КРАБ**.

27. Информативность сообщения, принимаемого человеком, определяется:
А) способом передачи сообщения;
В) способом обработки принимаемого сообщения;
С) способом приема сообщения;
D) временем приема сообщения;
Е) наличием новых знаний и понятностью.
28. Дисковод магнитного диска — это устройство для:
А) только записи информации;
В) хранения информации;
С) обработки информации;
D) только чтения информации;
Е) чтения и записи информации.
29. С помощью какого(их) элемента(ов) управления Рабочего стола можно переключаться между открытыми окнами папок и приложений в среде Windows?
А) Кнопки *Пуск*;
В) панели задач;
С) индикатора системных часов;
D) индикатора раскладки клавиатуры;
Е) меню открытых окон и приложений.
30. В прикладное программное обеспечение входит(ят):
А) системы программирования;
В) операционная система;
С) текстовые редакторы;

- D) операционная система с диалоговой оболочкой;
 E) устройства совместного пользования.

Ответы на тест № 3

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	C	D	A	A	D	E	A	C	C	C
№ вопроса	11	12	13	14	15	16	17	18	19	20
Ответ	D	C	B	A	E	A	E	B	C	D
№ вопроса	21	22	23	24	25	26	27	28	29	30
Ответ	A	C	D	B	C	C	E	E	B	C

8 класс

План уроков

№ урока	Тема	Вид занятий	Вид отчета
	Модуль 1. Компьютерные сети и их типы. Знакомство с языком разметки гипертекста HTML		
1	Компьютерные сети и их типы. Основные типы услуг, предоставляемых клиентам глобальных компьютерных сетей (электронная почта, телеконференции, файловые архивы)	Теория	Д/з № 1
2	Работа с электронной почтой	Практика	П/з № 1
3	Всемирная паутина (World Wide Web). Поисковые серверы. Формирование простых запросов для поисковых серверов	Теория	Д/з № 2

№ урока	Тема	Вид занятий	Вид отчета
4	Работа с WWW. Поиск информации в Интернете	Практика	П/з № 2
5	Основные компоненты компьютерных сетей	Теория	Д/з № 3
6	Работа в локальной сети	Практика	П/з № 3
7	Основные понятия языка HTML. Использование цветов. Форматирование текста. Гиперссылки	Теория	Д/з № 4
8	Разработка Web-страниц с использованием гиперссылок и форматирования	Практика	П/з № 4
9	Работа с графикой. Шрифты. Основные правила разработки Web-страниц	Теория	Д/з № 5
10	Разработка Web-страниц с использованием графики и различных шрифтов	Практика	П/з № 5
11	Тестирование		Тест № 1
	Модуль 2. Информационные модели и информационные структуры. Базы данных и СУБД		
12	Понятие модели. Виды информационных моделей	Теория	Д/з №6
13	Табличная организация информации. Создание таблицы типа «объект—свойство»	Теория + практика	Д/з №7 П/з №6
14	Понятие информационной структуры. Базы данных и информационные системы. Представление о СУБД	Теория	Д/з №8
15	Понятие реляционной БД. Типы данных. Понятие структуры таблицы. Первичный ключ	Теория	Д/з №9
16	Знакомство с СУБД. Создание и редактирование базы данных	Практика	П/з №7

№ урока	Тема	Вид занятий	Вид отчета
17	Элементы математической логики. Логические высказывания, логические величины, логические операции	Теория	Д/з № 10
18	Формирование запросов к БД. Сортировка записей	Теория	Д/з № 11
19	Формирование запросов на выборку, замену и удаление записей с использованием логических операций. Сортировка записей	Практика	П/з № 8
20	Вычисляемые поля. Формирование отчетов и форм с использованием Мастеров	Теория + практика	Д/з № 12, П/з № 9
21	Проектирование многотабличных БД и установка связей между ними	Теория	Д/з № 13
22	Проектирование структуры многотабличной БД и ее создание с помощью СУБД	Практика	П/з № 10
23	Зачетное занятие по СУБД	Практика	П/з № 11
24	Тестирование		Тест № 2
	Модуль 3. Табличные расчеты и электронные таблицы		
25	Знакомство с электронными таблицами. Просмотр и редактирование электронной таблицы	Теория + практика	Д/з № 14, П/з № 12
26	Ввод информации в электронные таблицы: тексты, числа, формулы	теория	Д/з № 15
27	Разработка электронной таблицы	практика	П/з № 13

№ урока	Тема	Вид занятий	Вид отчета
28	Понятие диапазона. Относительная и абсолютная адресация. Статистические функции СУММ, МИН, МАКС, СРЗНАЧ	Теория	Д/з № 16
29	Использование абсолютной и относительной адресации. Сортировка данных	Практика	П/з № 14
30	Логические операции и условная функция	Теория	Д/з № 17
31	Использование логических операций и условной функции	Практика	П/з № 15
32	Графическая обработка табличных данных. Табулирование функции и построение графика. Построение диаграммы	Теория + практика	Д/з № 18, П/з № 16
33	Зачетное занятие по ЭТ	Практика	П/з № 17
34	Итоговое тестирование		

Урок 1

Теория. Компьютерные сети и их типы. Основные типы услуг, предоставляемых клиентам глобальных компьютерных сетей (электронная почта, телеконференции, файловые архивы). (45 мин)

Изучаемые вопросы:

- Понятие компьютерной сети.
- Различие локальных и глобальных компьютерных сетей.
- Электронная почта как одна из услуг, предоставляемых клиентам глобальных сетей.
- Электронный почтовый ящик, имя почтового ящика.
- Структура электронного письма.

- ☐ Представление о телеконференции.
- ☐ Файловые архивы.

Домашнее задание:

- ☐ Читать: [5] — § 21, 22.
- ☐ Выполнить: домашнее задание № 1.

Урок 2

Практика. Работа с электронной почтой. (Выполнение практического задания № 1.) (45 мин)

Приобретаемые умения и навыки:

- 1) создание сообщения;
- 2) присоединение файла к письму;
- 3) отправка и получение сообщений;
- 4) работа с адресной книгой;
- 5) сохранение присоединенного файла на диске;
- 6) просмотр свойств письма;
- 7) удаление корреспонденции.

Урок 3

Теория. Всемирная паутина (World Wide Web). Поисковые серверы. Формирование простых запросов для поисковых серверов. (45 мин)

Изучаемые вопросы:

- ☐ Интернет — всемирная глобальная компьютерная сеть.
- ☐ World Wide Web и ее основные понятия (гипертекст, гиперссылка, URL-адрес, Web-страница, Web-сервер).
- ☐ Способы поиска информации в Интернете.

Домашнее задание:

- ☐ Читать: [5] — § 24.
- ☐ Выполнить: домашнее задание № 2.

Урок 4

Практика. Работа с WWW. Поиск информации в Интернете. (Выполнение практического задания № 2.) (45 мин)

Приобретаемые умения и навыки:

1) поиск заданной Web-страницы с использованием URL-адреса;

2) навигация по Web-страницам с помощью гиперссылок;

3) поиск информации с помощью поискового сервера с использованием тематических каталогов;

4) поиск информации с помощью поискового сервера по ключевому слову.

Урок 5

Теория. Основные компоненты компьютерных сетей. (45 мин)

Изучаемые вопросы:

- ☐ Линии связи и их типы.
- ☐ Модемы их характеристики.
- ☐ Понятие сервера и рабочей станции.
- ☐ Сетевые карты.
- ☐ Протоколы и их основные виды (TCP/IP, HTTP, FTP).
- ☐ Сетевое ПО.

Домашнее задание:

- ☐ Читать: [5] — § 23.
- ☐ Выполнить: домашнее задание № 3.

Урок 6

Практика. Работа в локальной сети. (Выполнение практического задания № 3.) (45 мин)

Приобретаемые умения и навыки:

1) получение доступа к ресурсам других рабочих станций и сервера;

2) создание сетевых дисков;

3) пересылка данных по локальной сети на другую рабочую станцию.

Урок 7

Теория. Основные понятия языка HTML. Использование цветов. Форматирование текста. Гиперссылки. (45 мин)

Изучаемые вопросы:

- ☐ Язык HTML — язык разметки гипертекста.
- ☐ Структура HTML-документа.
- ☐ Использование цветов в HTML.
- ☐ Основные тэги форматирования текста (<P>, <CENTER>,
, <H1>...<H6>).
- ☐ Создание гиперссылок.

Домашнее задание:

- ☐ Читать: [4] — глава 16.
- ☐ Выполнить: домашнее задание № 4.

Урок 8

Практика. Разработка Web-страниц с использованием гиперссылок и форматирования текста. (Выполнение практического задания № 4.) (45 мин)

Приобретаемые умения и навыки:

- 1) Создание HTML-документа с помощью простейшего текстового редактора (Блокнота) и его просмотр с помощью браузера;
- 2) использование гиперссылок;
- 3) использование основных тэгов форматирования текста;
- 4) использование фонового цвета страницы.

Урок 9

Теория. Работа с графикой. Шрифты. Основные правила разработки Web-страниц. (45 минут)

Изучаемые вопросы.

- ☐ Добавление графического изображения на страницу (тэг и его атрибуты).
- ☐ Форматы графических файлов JPEG и GIF.
- ☐ Графические гиперссылки.
- ☐ Тэги изменения шрифта (, <I>, , <U>).
- ☐ Основные правила разработки Web-страниц.

Домашнее задание:

- ☐ Читать: [31] — глава 16.
- ☐ Выполнить: домашнее задание № 5.

Урок 10

Практика. Разработка Web-страниц с использованием графики и различных шрифтов. (Выполнение практического задания № 5.) (45 мин)

Приобретаемые умения и навыки:

- 1) добавление графического изображения на страницу;
- 2) использование различных шрифтов при оформлении текста страницы.

Урок 11

Тестирование. Тест № 1 по теме «Компьютерные сети и их типы. Знакомство с языком разметки гипертекста HTML». (45 мин)

Урок 12

Теория. Понятие модели. Виды информационных моделей. (45 мин)

Изучаемые вопросы:

- ☐ Понятие модели.
- ☐ Материальные и информационные модели.
- ☐ Представление о системном анализе и его задачах.

Домашнее задание:

- ☐ Читать: [5] — § 25.
- ☐ Выполнить: домашнее задание № 6.

Урок 13

Теория. Табличная организация информации. (20 мин)

Изучаемые вопросы:

- ☐ Таблица как способ представления информационной модели.
- ☐ Типы таблиц.

Практика. Создание таблицы типа «объект—свойство» с помощью текстового редактора. (Выполнение практического задания № 6.) (25 мин)

Приобретаемые умения и навыки:

- 1) систематизация данных в табличную структуру;

2) поиск информации в табличной структуре данных.

Домашнее задание:

- ☐ Читать: [5] — § 26
- ☐ Выполнить: домашнее задание № 7.

Урок 14

Теория. Понятие информационной структуры. Базы данных и информационные системы. Представление о СУБД. (45 мин)

Изучаемые вопросы:

- ☐ Понятие информационной структуры.
- ☐ Понятие базы данных.
- ☐ Понятие информационных систем и сферы их применения.
- ☐ Понятие СУБД.

Домашнее задание:

- ☐ Читать: [5] — § 27, 28.
- ☐ Выполнить: домашнее задание № 8.

Урок 15

Теория. Понятие реляционной БД. Типы данных. Понятие структуры таблицы. Первичный ключ. (45 мин)

Изучаемые вопросы:

- ☐ Реляционная база данных и ее основные понятия (поле, запись).
- ☐ Понятие первичного ключа таблицы.
- ☐ Основные типы данных: числовой, текстовый, дата, логический.

Домашнее задание:

- ☐ Читать: [5] — § 27, 29.
- ☐ Выполнить: домашнее задание № 9.

Урок 16

Практика. Знакомство с СУБД. Создание и редактирование базы данных. (Выполнение практического задания № 7.) (45 мин)

Приобретаемые умения и навыки:

- 1) создание структуры таблицы базы данных;
- 2) редактирование структуры таблицы базы данных (добавление и удаление полей, изменение имени, типа и ширины поля);
- 3) определение первичного ключа таблицы;
- 4) добавление, удаление и редактирование записей базы данных.

Урок 17

Теория. Элементы математической логики. Логические высказывания, логические величины, логические операции. (45 мин)

Изучаемые вопросы:

- ☐ Понятие логического высказывания.
- ☐ Логические величины.
- ☐ Логические операции: умножение, сложение, отрицание.
- ☐ Таблица истинности.
- ☐ Приоритеты выполнения логических операций.

Домашнее задание:

- ☐ Читать: [5] — § 30, 31.
- ☐ Выполнить: домашнее задание № 10.

Урок 18

Теория. Формирование запросов к БД. Сортировка записей. (45 мин)

Изучаемые вопросы:

- ☐ Простые и сложные условия поиска.
- ☐ Понятие запроса и его виды (выборка, замена, удаление).
- ☐ Сортировка записей по простому и составному ключам.

Домашнее задание:

- ☐ Читать: [5] — § 30, 31, 32
- ☐ Выполнить: домашнее задание № 11.

Урок 19

Практика. Формирование запросов на выборку, замену и удаление записей с использованием логиче-

ских операций. Сортировка записей. (Выполнение практического задания № 8.) (45 мин)

Приобретаемые умения и навыки:

- 1) формирование запросов на выборку с включением в них только заданных полей;
- 2) формирование запросов на замену;
- 3) формирование запросов на удаление;
- 4) просмотр результатов запроса на выборку;
- 5) сортировка записей в запросах на выборку.

Урок 20

Теория. Вычисляемые поля. Понятие формы и отчета. (15 мин)

Изучаемые вопросы:

- Понятие вычисляемого поля.
- Форма и отчет — объекты СУБД.

Практика. Вычисляемые поля. Формирование отчетов и форм. (Выполнение практического задания № 9.) (30 мин)

Приобретаемые умения и навыки:

- 1) формирование запросов на выборку с использованием вычисляемых полей;
- 2) формирование формы с использованием Мастера ее использование для ввода и редактирования информации;
- 3) формирование отчетов с помощью Мастера и их просмотр.

Домашнее задание:

- Читать: [24] — разделы 14.2.3, 14.2.4, 14.5.
- Выполнить: домашнее задание № 12.

Урок 21

Теория. Проектирование многотабличных БД и установка связей между ними. (45 мин)

Изучаемые вопросы:

- Понятие многотабличной БД.
- Понятие нормализованной БД и правила нормализации.

- ☐ Установка связей между таблицами.
Домашнее задание:
- ☐ Читать: [5] — § 7.2.
- ☐ Выполнить: домашнее задание № 13.

Урок 22

Практика. Проектирование структуры многотабличной БД и ее создание с помощью СУБД. (Выполнение практического задания № 10.) (45 мин)

Приобретаемые умения и навыки:

- 1) проектирование структуры многотабличной БД на основе заданного списка полей;
- 2) формирование структуры многотабличной БД с помощью СУБД;
- 3) установка связей между таблицами.

Урок 23

Практика. Зачетное занятие по СУБД. (Выполнение практического задания № 11.) (45 мин)

Урок 24

Тестирование. Тест № 2 по теме «Информационные модели и информационные структуры. Базы данных и СУБД». (45 мин)

Урок 25

Теория. Знакомство с электронными таблицами. (30 мин)

Изучаемые вопросы:

- ☐ Назначение электронных таблиц.
- ☐ Структура электронной таблицы.
- ☐ Понятие адреса ячейки ЭТ.

Практика. Просмотр и редактирование электронной таблицы. (Выполнение практического задания № 12.) (15 мин)

Приобретаемые умения и навыки:

- 1) редактирование ячеек электронной таблицы;
- 2) изменение ширины столбца таблицы;
- 3) удаление строк и столбцов таблицы;
- 4) очистка содержимого ячеек таблицы.

Домашнее задание:

- Читать: [5] — § 33.
- Выполнить: домашнее задание № 14.

Урок 26

Теория. Ввод информации в электронные таблицы: тексты, числа, формулы. (45 мин)

Изучаемые вопросы:

- Правила заполнения ЭТ.
- Представление чисел в ЭТ (с фиксированной и с плавающей точкой).
- Правила записи формул и знаки арифметических операций.
- Понятие режима отображения формул.

Домашнее задание:

- Читать: [5] — § 34.
- Выполнить: домашнее задание № 15.

Урок 27

Практика. Разработка электронной таблицы. (Выполнение практического задания № 13.) (45 мин)

Приобретаемые умения и навыки:

- 1) ввод и редактирование формул ЭТ;
- 2) изменение оформления ЭТ (границы, заливка, шрифты).

Урок 28

Теория. Понятие диапазона. Относительная и абсолютная адресация, статистические функции СУММ, МИН, МАКС, СРЗНАЧ. (45 мин)

Изучаемые вопросы:

- Понятие диапазона ячеек.
- Различие абсолютной и относительной адресации.
- Функция суммирования.
- Функции вычисления минимума, максимума и среднего арифметического.

Домашнее задание:

- Читать: [5] — § 35.
- Выполнить: домашнее задание № 16.

Урок 29

Практика. Использование абсолютной и относительной адресации. Сортировка данных. (Выполнение практического задания № 14.) (45 мин)

Приобретаемые умения и навыки:

- 1) создание формул с использованием абсолютной адресации;
- 2) копирование содержимого ячеек ЭТ;
- 3) использование функции суммирования;
- 4) нахождение минимального и максимального значений диапазона;
- 5) вычисление среднего арифметического;
- 6) сортировка таблицы по значению одного столбца.

Урок 30

Теория. Логические операции и условная функция. (45 мин)

Изучаемые вопросы:

- Логические операции конъюнкции, дизъюнкции и отрицания.
- Условная функция.

Домашнее задание:

- Читать: [5] — § 36, 37.
- Выполнить: домашнее задание № 17.

Урок 31

Практика. Использование логических операций и условной функции. (Выполнение практического задания № 15.) (45 мин)

Приобретаемые умения и навыки:

- 1) создание формул с использованием логических операций;
- 2) создание формул с использованием условной функции.

Урок 32

Теория. Графическая обработка табличных данных. (20 мин)

Изучаемые вопросы:

- Графический режим работы табличного процессора.
- Типы диаграмм.

Практика. Табулирование функций и построение графика. (Выполнение практического задания № 16.) (25 мин)

Приобретаемые умения и навыки:

- 1) табулирование функции на заданном отрезке с указанным шагом табуляции;
- 2) построение графика заданной функции;
- 3) построение диаграммы заданного типа;
- 4) использование легенды, меток и заголовков осей при создании диаграммы.

Домашнее задание:

- Читать: [5] — § 36.
- Выполнить: домашнее задание № 18.

Урок 33

Практика. Зачетное занятие по ЭТ. (Выполнение практического задания № 17.) (45 мин)

Урок 34

Тестирование. Итоговое тестирование. (45 мин)

Домашние задания

Домашнее задание № 1

1. В чем различие локальных и глобальных сетей?
2. [2]: раздел 5.3, задача № 2.
3. Какие преимущества имеет электронная почта по сравнению с обычной?
4. Саша увлекается компьютерными играми и хочет найти друзей по своему хобби. Какой услугой глобальной вычислительной сети он может воспользоваться?

5. Какие действия может выполнить абонент электронной почты с полученным сообщением?

Домашнее задание № 2

1. В чем заключается гипертекстовый способ представления информации?
2. Вам нужно найти информацию об архитекторе Исаакиевского собора. Перечислите несколько возможных способов поиска данной информации, которые вы смогли бы использовать.
3. Какие объекты в данном документе могут выполнять роль ключа для связи с другими гипертекстовыми документами?
4. Какое программное обеспечение используется для просмотра Web-страниц?
5. Какие функции выполняет Web-сервер?

Домашнее задание № 3

1. Что такое шлюз?
2. Саша собирается подключить свой компьютер к глобальной сети через телефонную линию. Какое устройство ему необходимо для этого и почему?
3. Сколько символов текста можно передать по сети за 5 секунд, используя модем, работающий со скоростью 28800 бит/с?
4. Организация собирается объединить свои компьютеры в локальную сеть. Какое оборудование им будет необходимо?
5. Для чего используются протоколы TCP/IP?

Домашнее задание № 4

1. Какие функции выполняют тэги?
2. Какой тэг нужно использовать, чтобы вывести название Web-страницы в заголовке окна?
3. Передпишите фрагмент HTML-документа, исправив ошибки:

```
<HTML>  
<TITLE>
```

Кинотеатры Перми

```
<BODY><COLOR=yellow>Кинотеатры  
ми</COLOR>
```

Пер-

```
<A HREF=triumf.htm> Триумф</A>
```

```
<A HREF=kristall.htm> Кристалл</HREF>
```

```
<A HREF=october.htm> Октябрь
```

```
<BODY>
```

```
</HTML>
```

4. Зарисуйте внешний вид страницы, описанной в вопросе № 3.

Домашнее задание № 5

1. Зарисуйте как можно более точно вид Web-страницы, имеющей следующее описание (в файле rose.jpg хранится изображение розы):

```
<TITLE>Поздравительная открытка</TITLE>
```

```
<BODY BGCOLOR=yellow>
```

```
<CENTER>
```

```
<FONT COLOR=red size=+4>
```

```
<I>Поздравляю, с днем рождения!!!</I>
```

```
</FONT>
```

```
<BR><BR>
```

```
<IMG BORDER=0 SRC="rose.jpg">
```

```
<FONT COLOR=blue SIZE=+3>
```

```
<p>Дорогая Оленька!</p>
```

```
<p>Поздравляю тебя с днем твоего рождения.  
Желаю счастья, здоровья, успехов в уче-  
бе</p>
```

```
</FONT>
```

```
<FONT COLOR=blue SIZE=+2>
```

```
<p>Маша Иванова</p>
```

```
</FONT>
```

```
</CENTER>
```

```
</BODY>
```

2. Какие правила необходимо соблюдать при разработке Web-страниц (перечислите 5 наиболее важных)? Поясните, почему это необходимо.

3. Какие тэги используются для изменения расположения текста на странице?
4. Написать тэг или последовательность тэгов для вставки в документ рисунка cat.jpg, который является ссылкой на документ cat.htm.

Домашнее задание № 6

1. В чем состоит задача системного анализа?
2. Приведите примеры материальной и информационной моделей дома.
3. Какие свойства реальных объектов воспроизводят следующие модели:
 - муляжи продуктов в витрине магазина;
 - чучело птицы;
 - заводной игрушечный автомобиль.
4. Приведите примеры различных типов графических моделей.

Домашнее задание № 7

1. [2]: раздел 2.3, задача № 11.
2. [5]: § 26, задача 5.
3. [5]: § 26, задача 7.

Домашнее задание № 8

1. Преобразуйте приведенную ниже информацию к табличному виду, определив имя таблицы и название каждого столбца: Оля, Петя, 13, пение, 14, баскетбол, Вася, Катя, 13, хоккей, баскетбол, футбол, 15, 11, Коля, 11, танцы, Сережа.
2. Что такое СУБД? Приведите 2-3 примера СУБД.
3. Приведите 3-4 примера информационных систем.

Домашнее задание № 9

1. [2]: раздел 5.4, задача № 3.
2. [2]: раздел 5.4, задача № 5.
3. [2]: раздел 5.4, задач № 6.

Домашнее задание № 10

1. Установите, какие из следующих предложений являются логическими высказываниями, а какие — нет (объясните, почему):
 - а) Солнце — спутник Земли;
 - б) $2+3 \times 4$;
 - в) Сегодня отличная погода;
 - г) У каждой лошади есть хвост.
2. У Пети есть следующие животные: кошка Мурка (серая, 3 года), кошка Машка (белая, 4 года), кошка Соня (белая, 2 года), кошка Лиза (рыжая, 3 года), собака Шарик (белая, 3 года), собака Дружок (рыжая, 4 года), собака Дик (серая, 4 года), попугай Кеша (белый, 4 года), попугай Вася (серый, 2 года). Петя пошел на прогулку и взял с собой несколько животных. Для каждого из взятых им животных истинно утверждение: это собака младше 4 лет или кошка белого цвета. Для всех оставшихся дома животных это утверждение будет ложным. Назовите имена животных, взятых Петей на прогулку.
3. Для описанных выше животных выберите истинные логические утверждения:
 - а) Шарик или Дружок рыжего цвета и Кеше 2 года;
 - б) Лиза — собака или Шарик — кошка или Вася — попугай;
 - в) Кеша и Вася серого цвета или Лиза — кошка.
4. Что получится в результате следующих логических операций (опишите весь ход решения по действиям): (НЕ истина ИЛИ ложь) И (ложь ИЛИ истина)?

Домашнее задание № 12

1. Дана БД «Спортивная гимнастика»:

№	Фамилия	Страна	Перекладина	Кольца	Конь
302	Робсон Джон	США	9,225	9,000	8,875
303	Леонидов Сергей	Россия	9,500	9,225	9,875
305	Чугайнов Андрей	Россия	9,225	9,775	9,925
301	Пьолин Франсуа	Франция	8,500	8,900	8,225
304	Штольц Андреас	Германия	9,775	9,225	9,000
308	Морозов Олег	Украина	9,885	9,500	9,625
309	Джонсон Фрэнк	США	8,885	9,000	9,325
310	Ли Грег	США	9,500	9,500	9,225
307	Овсиенко Геннадий	Украина	9,975	9,000	9,225
306	Куэртен Пьер	Франция	9,925	8,775	9,500

Сформируйте к данной БД условия запросов, после которых на экран будут выведены сведения о спортсменах:

- набравших в сумме более 28 баллов;
- имеющих средний балл за все снаряды выше 9,225.

2. Сведения о каких спортсменах из БД «Спортивная гимнастика» (вопрос № 1) будут выведены на экран после применения запроса:

(страна=Россия ИЛИ страна=США) И перекладина+кольца>=19

3. Сформируйте для БД из вопроса № 1 отчет, сгруппировав данные по полю «страна».

Домашнее задание № 13

1. Дан набор полей: «фамилия», «имя», «дата рождения», «пол», «вес», «рост», «школа», «класс», «дата прохождения медосмотра», «оценка по математике», «оценка по физкультуре», «оценка по рисованию», «домашний адрес», «домашний телефон». Какие из перечисленных полей необходимо будет включить в базу данных «Городская библиотека»?
2. Опишите структуру БД «Коллекция», в которой можно будет хранить сведения о какой-либо отдельной коллекции (марок, значков, монет и так далее). База должна содержать не менее 5 полей.
3. Спроектируйте и опишите структуру базы данных «Программа передач на неделю», с помощью которой можно будет получить ответы на следующие вопросы:
 - ☐ Какие фильмы можно посмотреть в четверг?
 - ☐ В какое время транслируются мультфильмы в пятницу?
 - ☐ Какие программы будут показаны с 12.00 до 16.00 в воскресенье?
 - ☐ В какое время будут показаны программы новостей в понедельник по каналам ОРТ и НТВ?
 - ☐ Какие развлекательные программы можно посмотреть в субботу по РТР?
4. Нормализуйте структуру БД «Подписка» (опишите структуру каждой таблицы и определите связи между ними).

Фамилия	Адрес	Индекс издания	Название издания	Тип издания	С какого месяца	По какому месяцу
Михайлов	ул. Солнечная 5-1	40532	Звезда	газета	1	12
Шолохова	ул. Звездная 9-53	33234	Семья	журнал	7	12

Фамилия	Адрес	Индекс издания	Название издания	Тип издания	С какого месяца	По какой месяц
Андреев	ул. Лесная 17-56	22312	Гудок	газета	1	6
Григорьева	ул. Лесная 10-6	33234	Семья	журнал	4	6
Зорин	ул. Звездная 32-7	22312	Гудок	газета	4	12
Рыкова	ул. Звездная 15-34	33234	Семья	журнал	1	6

Домашнее задание № 14

1. В чем основное отличие ЭТ от базы данных?
2. Даны ячейки электронной таблицы с адресами DA15, D10, AB3000, R5, AH102, KA200, B17000. Какие из перечисленных ячеек электронной таблицы не существуют и почему?
3. [5]: раздел 5.5, задача № 7.

В какие ячейки следующих таблиц заносятся числа, а в какие — формулы? Если есть несколько вариантов ответа, то рассмотрите их все.

а)

	А	В	С
Г	цена единицы товара	количество товара	стоимость
2			

б)

	А	В	С
1	длина пути	скорость	время в пути
2			

Домашнее задание № 15

1. В ячейки ЭТ введены следующие формулы:

$$\begin{aligned} A1: 8; \quad A2: A1*10; \quad A3: A2-A1; \\ B1: A2/4; \quad B2: (B1-A1)/6. \end{aligned}$$

Запишите значения, которые будут отображены на экране в ячейках ЭТ.

2. Запишите арифметическое выражение в виде формулы для ЭТ (вместо x и y в формуле используйте адреса ячеек):

$$\frac{15x^2 - \frac{7}{12}y}{5(x^3 - 6y)}.$$

3. [5]: раздел 5.5, задача № 14.

Запишите в традиционной математической форме следующие формулы из электронных таблиц:

- а) $C2+A5/3$; в) $C2/(A5+3)$; д) $A1*A2/D12/D3$;
б) $(C2+A5)/3$; г) $A1*A2/D12*D3$.

4. Переведите числа из экспоненциальной формы в вещественную:

- а) $-0,567E4$; б) $25645E-7$; в) $15,3E-2$; г) $0,3E1$.

5. Придумайте и сформулируйте условие задачи, для решения которой можно использовать следующую ЭТ:

	A	B
1	15	=A1/A4*100
2	30	=A2/A4*100
3	10	=A3/A4*100
4	=A1+A2+A3	

Домашнее задание № 16

1. Сколько ячеек ЭТ включают в себя следующие диапазоны:

- а) A2:B10; б) C13:E20; в) Z100:AB109 ?

2. Дана таблица:

	A	B	C
1	5	10	18
2	=A1+6		
3	=B1*A1		
4			

Какие формулы будут записаны в ячейки A4, B2, B3, B4, C2, если в ячейки B2 и B3 скопировать содержимое ячейки A2, а в ячейки A4, B4, и C2 — содержимое ячейки A3?

- После копирования содержимого ячейки A2 в ячейки B2 и A3 в них оказались формулы $=B1+C1$ и $=A2+B2$ соответственно. Что было записано в ячейке A2?
- Запишите приведенные ниже выражения с помощью функции суммирования:
 - $A5+A6+A7+B5+B6+B7$
 - $C4+C5+D4+D5+E4+E5+F4+F5$.
- Придумайте и опишите таблицу, с помощью которой можно вести учет расхода электроэнергии в течение года и определять сумму оплаты за каждый месяц с января по декабрь. Сумма оплаты определяется на основе разницы показаний счетчика за текущий и предыдущий месяцы. Учитывать, что стоимость 1 кВт·ч в течение года не меняется. Таблицу опишите в режиме отображения формул.

Домашнее задание № 17

- В ячейках ЭТ хранятся следующие значения и формулы:

A1: 10;

B1: ЕСЛИ(A2*A1>1000;5;10);

A2: 20;

B2: 5;

A3: A2/A1;

B3: СУММ(A1:B2).

Какой результат будут иметь следующие логические выражения (ИСТИНА или ЛОЖЬ):

а) ИЛИ(A1<5;B3=45);

б) НЕ(A3=2);

в) ИЛИ(И(НЕ(A3>2);A1=10);И(B2<=5;B3=50))?

Примечание. Опишите весь ход решения по действиям.

2. В январе 1 кг картофеля, 1 л молока, 1 десяток яиц стоили по A руб. (A — произвольное и может меняться). Ежемесячно цена картофеля увеличивается на 3% по отношению к предыдущему месяцу, цена молока — на 4%, цена яиц — на 1%. Опишите таблицу, в которой будет прослеживаться ежемесячное изменение цены каждого продукта в течение квартала с января по март (в режиме отображения формул).
3. Сформируйте в режиме отображения формул таблицу, содержащую сведения о пяти учениках школы (фамилия, возраст, рост). Сколько учеников могут заниматься в баскетбольной секции, если туда принимают детей с ростом не менее 160 см и возрастом не старше 13 лет?
4. Дана таблица в режиме отображения формул:

	A	B
1	2	=A1+2
2	=МИН(A1:B1)	=ЕСЛИ(A1=A2;8;СУММ(A1:A2))
3	=И(A2<B1;B2>8)	=ИЛИ(A2=1;B1=1;B1=4)
4	=НЕ(A1=B1)	=СРЗНАЧ(A1:B2)

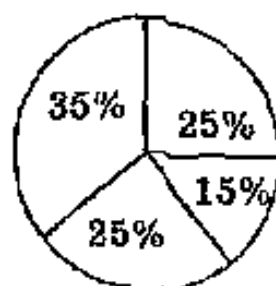
Запишите эту же таблицу в режиме отображения значений.

Домашнее задание № 18

1. Изобразите диаграммы двух различных типов для следующей таблицы:

дата	доллар	марка
1.02.98	5,9	3,35
1.03.98	6,07	3,53
1.04.98	6,15	3,68

2. Придумайте и заполните таблицу, для которой может быть построена следующая круговая диаграмма:



Примечание. Таблица должна нести смысловую нагрузку, а не просто содержать одни числовые данные.

Практические задания для выполнения на компьютере

Практическое задание № 1

1. Запустить почтовую программу.
2. Создать сообщение, содержащее информацию о вас (фамилия, имя) и один-два любых вопроса адресату.
3. Отправить сообщение по указанным преподавателем двум адресам.
4. Получить отправленные вам сообщения.
5. Занести адреса отправителей в адресную книгу.
6. Определить время отправки сообщений и их размер.
7. Прочитать полученные сообщения и ответить на них. К своим сообщениям присоединить графический файл из папки FOTO.
8. Получить и прочитать вновь отправленные вам сообщения.
9. Сохранить полученные фотографии в папке FOTO.
10. Очистить папки «Входящие» и «Отправленные».

Примечание. Перед началом занятия необходимо на всех компьютерах создать папку FOTO, в которую

записать один графический файл с произвольной фотографией (на всех компьютерах фотографии должны быть различны и файлы должны иметь разные имена).

Практическое задание № 2

1. Зайти на поисковый сервер Rambler (<http://www.rambler.ru>).
2. Найти информацию о репертуаре театра ЛЕНКОМ на текущий месяц (раздел «Театр»).
3. Зайти на сервер Пермского государственного университета (<http://www.psu.ru>) и узнать об условиях приема на механико-математический факультет.
4. Узнать курс доллара на сегодняшний день.
5. Узнать настоящую фамилию писателя Кира Булычева.
6. Узнать прогноз погоды на сегодняшний день.

Практическое задание № 3

1. Создать в вашем рабочем каталоге папку SETI.
2. Запустить программу Блокнот и набрать в нем следующую информацию о себе: фамилия, имя, школа, класс, фамилия и инициалы преподавателя информатики.
3. Сохранить набранную информацию в папке SETI под именем aboutN.txt (N — номер вашего компьютера).
4. Зайти на соседний справа компьютер в указанную преподавателем папку и сделать ее сетевым диском.
5. Скопировать файл aboutN.txt на созданный сетевой диск в папку SETI.
6. Отключить сетевой диск.
7. Скопировать свой файл aboutN.txt на все остальные компьютеры (без использования сетевого диска).
8. Просмотреть в вашей папке SETI скопированные в нее другими учениками файлы.
9. Удалить папку SETI со своего компьютера.

Примечания.

1) Перед началом занятия необходимо на всех компьютерах создать рабочую папку с полным сетевым доступом.

2) В пункте 4 данного задания преподаватель должен указать ученикам рабочую папку, для которой был создан полный сетевой доступ.

Практическое задание № 4

Разработать Web-сайт на тему «Мой город» (следующий ниже пример относится к г. Перми)

Предварительная работа:

1) создать в вашем рабочем каталоге свой каталог (имя произвольное);

2) Скопировать во вновь созданный каталог файлы `history.txt` и `new.htm` из указанного преподавателем каталога.

Описание содержания сайта:

Главная страница (файл `perm.htm`)

Title: Город Пермь

Заголовок: Город Пермь

Гиперссылки: История (ссылка на файл `history.htm`)

Современная Пермь (ссылка на файл `new.htm`)

Способ размещения: в центре страницы

История (файл `history.htm`)

Title: История

Заголовок: История Пермского края

Текст: из файла `history.txt`

Способ размещения: заголовок — в центре, первый и третий абзацы выравнивать по левому краю, второй — по правому.

Примечание. Для работы необходимо подготовить файлы `history.txt` и `new.htm`.

Практическое задание № 5

Разработать Web-сайт на тему «Группа Spice Girls».

Предварительная работа:

1) создать в вашем рабочем каталоге свой каталог (имя произвольное);

2) скопировать во вновь созданный каталог файлы `группа.jpg`, `emma.jpg`, `gery.jpg`, `melb.jpg`, `melc.jpg` из указанного преподавателем каталога.

Описание содержания сайта:

Главная страница (файл `main.htm`)

Title: Spice Girls

Графика: `группа.jpg`

Гиперссылки: `Emma` (ссылка на файл `emma.htm`)

`Gery` (ссылка на файл `gery.htm`)

`MelaB` (ссылка на файл `melb.jpg`)

`MelaC` (ссылка на файл `melc.jpg`)

Способ размещения: в центре страницы.

`Emma` (файл `emma.htm`)

Title: Emma

Заголовок: Emma Bentone

Графика: `emma.jpg`

Способ размещения: в центре страницы.

`Gery` (файл `gery.htm`)

Title: Gery

Заголовок: *Geraldine Estelle Haliwell*

Графика: `gery.jpg`

Способ размещения: в центре страницы.

Обязательно использовать различные шрифты (не менее двух различных типов и двух различных начертаний). По окончании работы удалить созданный каталог (с разрешения преподавателя).

Примечание. Для работы необходимо подготовить файлы `группа.jpg`, `emma.jpg`, `gery.jpg`, `melb.jpg`, `melc.jpg`.

Практическое задание № 6

1. На основе следующих данных спроектировать таблицу и создать ее, пользуясь средствами текстового редактора:

Полярная звезда находится в созвездии Малая Медведица. Бетельгайзе находится в созвездии Орион. Расстояние до Спика — 260 световых лет. Денеб находится в созвездии Лебедь. Акрукс ярче Солнца в 2200 раз. Расстояние до Бетельгайзе — 650 световых лет. Ригель ярче Солнца в 55000 раз. Канопус находится в созвездии Стрекоза. Ахернар ярче Солнца в 660 раз. Расстояние до Капеллы — 46 световых лет. Спика находится в созвездии Дева. Антарес находится в созвездии Скорпион. Расстояние до Арктура — 36 световых лет. Альдебаран ярче Солнца в 165 раз. Бетельгайзе ярче Солнца в 22000 раз. Расстояние до Акрукса — 260 световых лет. Денеб ярче Солнца в 72500 раз. Расстояние до Антареса — 425 световых лет. Альдебаран находится в созвездии Телец. Антарес ярче Солнца в 6600 раз. Расстояние до Канопуса — 181 световой год. Арктур находится в созвездии Волопас. Капелла ярче Солнца в 150 раз. Ахернар находится в созвездии Эридан. Расстояние до Полярной звезды — 780 световых лет. Ригель находится в созвездии Орион. Спика ярче Солнца в 2200 раз. Акрукс находится в созвездии Южный Крест. Расстояние до Альдебарана — 70 световых лет. Арктур ярче Солнца в 105 раз. Расстояние до Ахернара — 127 световых лет. Расстояние до Денеба — 1600 световых лет. Канопус ярче Солнца в 6600 раз. Капелла находится в созвездии Возничий. Полярная звезда ярче Солнца в 6000 раз. Расстояние до Ригеля — 820 световых лет.

2. Дать название таблице, определить названия столбцов.

3. Анализируя таблицу, определить самую яркую звезду и звезду, максимально удаленную от Земли.

Практическое задание № 7

1. Создать структуру таблицы базы данных «Ученик», содержащую следующие поля: фамилия, имя, школа, класс, дата рождения, вес.
2. Определить главный ключ таблицы.
3. В режиме таблицы ввести в базу данных 5 записей об учениках с фамилиями: Бобров, Семенова, Петров, Лукьянова, Филимонов (остальные поля заполнить произвольно).
4. Добавить в структуру таблицы после поля «дата рождения» поле «рост».
5. Удалить из структуры поле «дата рождения».
6. Заполнить в таблице поле «рост».
7. Удалить из таблицы сведения об учениках Петрове и Филимонове.

Практическое задание № 8

1. Открыть файл VIDEO.
2. Открыть таблицу «Фильмы» и ознакомиться с ее содержимым.
3. Вывести на экран поля «фильм», «страна» и «время» для всех боевиков.
4. Вывести на экран поля «номер», «фильм», «страна» и «жанр» для боевиков, снятых в России и фантастических фильмов, снятых в США, отсортировав их по ключу «страна (возр)+фильм (убыв)».
5. Вывести на экран поля «фамилия», «фильм», «дата выдачи» и «страна» для фильмов, выданных позднее 01.01.97 и снятых не в США, отсортировав их по ключу «дата приобретения (убыв)».
6. Удалить из БД все фильмы с жанром «комедия» продолжительностью более 90 минут.
7. Заменить поле «дата приобретения» на 15.03.98 для всех записей с датами выдачи от 01.01.98 до 01.04.98.

Примечание. Перед началом занятия необходимо создать БД VIDEO (см. приложение к [2]).

Практическое задание №9

1. Открыть БД STUDENT.
2. Открыть таблицу «Экзамены» и ознакомиться с ее содержимым.
3. Создать с помощью Мастера форму (произвольного дизайна) и добавить в таблицу три новых записи (любых).
4. Сформировать запрос, с помощью которого вывести на экран фамилии, группу, все оценки и сумму баллов для студентов, не имеющих «двоек».
5. Создать отчет, с помощью которого вывести полную информацию о студентах, средний балл которых выше 3,5, сгруппировав данные по полю «группа» и отсортировав их в алфавитном порядке.

Примечание. Перед началом занятия необходимо создать БД STUDENT (см. приложение к [2]).

Практическое задание № 10

1. Нормализовать структуру базы данных VIDEOPROKAT (на бумаге).
Структура базы данных VIDEOPROKAT:
номер — номер кассеты;
фильм — название фильма;
жанр — жанр фильма;
время — продолжительность фильма;
страна — страна-производитель;
фамилия — фамилия клиента;
адрес — адрес клиента;
дата выдачи — дата выдачи кассеты.
2. Создать нормализованную структуру базы данных VIDEOPROKAT.
3. Установить связи между таблицами.

Практическое задание № 11

1. Открыть файл АВІТ.
2. Ознакомиться с содержимым таблиц «Абитуриент», «Специальность», «Факультет».
3. Создать запрос для вывода на экран полей «фамилия», «факультет», «название специальности» для абитуриентов, поступающих на математический и физический факультеты, окончивших подготовительные курсы.
4. Создать запрос для вывода на экран фамилии, факультета, оценки за первый экзамен, оценки за второй экзамен, оценки за третий экзамен для всех абитуриентов, у которых оценка за третий экзамен выше среднего балла за все экзамены.
5. Создать отчет для вывода на экран фамилии, факультета и суммы баллов для абитуриентов, набравших более 12 баллов за три экзамена. Данные сгруппировать по полю «факультет» и отсортировать по сумме баллов.
6. Удалить информацию обо всех абитуриентах, имеющих хотя бы одну «двойку».

Примечание. Перед началом занятия необходимо создать БД АВІТ (см. приложение к [2]).

Практическое задание № 12

1. Открыть таблицу АВІТ.
2. Установить ширину ячеек таким образом, чтобы вся информация целиком отображалась на экране.
3. Очистить содержимое столбца В (кроме заголовков).
4. Ввести новые оценки по математике для всех абитуриентов и проследить за изменением среднего балла и общей суммы баллов.
5. Удалить из таблицы столбец, содержащий сведения о сумме баллов.
6. Удалить из таблицы строки, в которых содержатся сведения об абитуриентах, имеющих «двойки».

Примечание. Перед началом занятия необходимо создать файл АВІТ (см. приложение к [2]).

Практическое задание № 13

1. Создать ЭТ «Успеваемость девятиклассников по информатике», в которой хранятся следующие сведения об учениках 9А, 9Б, 9Г и 9Д классов:
 - класс;
 - общее количество учеников в классе;
 - количество учеников, имеющих по информатике оценку «5»;
 - количество учеников, имеющих по информатике «4»;
 - количество учеников, имеющих по информатике «3»;
 - количество учеников, имеющих по информатике «2» (использовать формулу).
2. Добавить к ЭТ столбцы, в которых вычисляется процент «пятерок», «четверок», «троек» и «двоек» от общего количества учеников в каждом классе.
3. Изменить общее количество учеников в каждом классе и проследить за изменением процентного соотношения.
4. Добавить в таблицу после сведений о 9Б классе строку со сведениями об успеваемости 9В класса и заполнить ее данными (формулы копировать).
5. Оформить таблицу, выделив особым шрифтом, заливкой и границами первый столбец и первую строку таблицы.
6. Вывести таблицу в режиме отображения формул.
7. Сохранить созданную таблицу в файле KLASS9.

Практическое задание № 14

Валя, Юра, Костя, Марина и Света пошли в магазин за фруктами. Валя купила 2 кг яблок, 3 кг бананов, 1 кг апельсинов и 4 кг груш. Юра купил 1 кг яблок, 2 кг бананов, 4 кг апельсинов и 2 кг груш. Костя купил 3 кг яблок, 5 кг бананов, 2 кг апельсинов, 1 кг

груш. Марина купила 3 кг яблок, 2 кг бананов, 2 кг апельсинов и 1 кг груш. Света купила всех фруктов по 2 кг.

1 кг яблок стоит 3 монеты, 1 кг бананов — 5 монет, 1 кг апельсинов — 4 монеты и 1 кг груш — 4,5 монеты.

1. Построить на основе вышеперечисленных данных электронную таблицу.
2. Добавить к таблице столбец, в котором будет вычисляться общая сумма, которую заплатил каждый из ребят (формула для вычислений должна вводиться только в одну ячейку, а в остальные — копироваться).
3. Вычислить общую сумму покупок всех ребят.
4. Определить максимальную и минимальную стоимости покупок.
5. Вычислить среднюю арифметическую сумму покупок ребят.
6. Отсортировать таблицу по убыванию суммы покупок.
7. Изменить стоимость бананов на 4 монеты и выделить другим цветом ячейки, в которых произошли изменения.

Практическое задание № 15

1. Разработать таблицу, содержащую следующие сведения о восьми абитуриентах университета: фамилия, оценка за экзамен по математике, оценка за экзамен по физике, оценка за экзамен по русскому языку, сумма баллов за три экзамена.
2. Добавить в таблицу столбец, в котором будет выводиться ИСТИНА, если абитуриент имеет только «4» и «5» и ЛОЖЬ — в противном случае (использовать логическое умножение).
3. Добавить в таблицу столбец, в котором будет выводиться ИСТИНА, если абитуриент имеет хотя бы одну «пятерку», и ЛОЖЬ в противном случае (использовать логическое сложение).

4. Добавить в таблицу столбец, в котором будет выводиться «зачислен», если сумма баллов абитуриента больше или равна проходному баллу 12, и «нет», если сумма баллов меньше проходного балла (использовать условную функцию).

Практическое задание № 16

1. На отрезке $[0; 2]$ с шагом 0,2 протабулировать функцию $0,25x^3 + x - 1,2502$.
Построить график этой функции.

Практическое задание № 17

Компания по снабжению электроэнергией взимает плату с клиентов по тарифу:

- k_1 рублей за 1 кВт·ч за первые 200 кВт·ч;
- k_2 рублей за 1 кВт·ч за каждый кВт·ч свыше 200.

Начальные значения: $k_1=0,1$ руб, $k_2=0,15$ руб.

Например, если расход электроэнергии равен 550 кВт·ч, то клиент должен заплатить сумму $200 \times 0,1 + (550 - 200) \times 0,15$ руб.

1. Построить таблицу, отображающую сведения о расходе электроэнергии и внесенной плате пятью клиентами компании и оформить ее (использовать различные шрифты, заливку, обрамление).
2. Посчитать общую суммарную плату, внесенную клиентами.
3. Выяснить, сколько клиентов израсходовали свыше 200 кВт·ч.
4. Выяснить, какая минимальная и какая максимальная платы были внесены клиентами.
5. Построить график, отображающий расход электроэнергии каждым клиентом.
6. Отсортировать таблицу по убыванию расхода электроэнергии.

Тесты

Тест № 1

1. HTTP — это:

- А) средство организации Web-серверов;
- В) протокол передачи гипертекста;
- С) протокол эмуляции терминала;
- Д) организация адресов в Интернете;
- Е) сервер WWW.

2. Гипертекст — это:

- А) очень большой текст;
- В) структурированный текст, в котором могут осуществляться переходы по выделенным меткам;
- С) текст, набранный на компьютере;
- Д) текст, в котором используется шрифт большого размера;
- Е) текст, занимающий большой объем памяти компьютера.

3. Браузеры (например, Internet Explorer) являются

- А) серверами Интернета;
- В) почтовыми программами;
- С) средством создания Web-страниц;
- Д) средством просмотра Web-страниц;
- Е) средством ускорения работы коммуникационной сети.

4. Каждый компьютер, подключенный к глобальной сети Интернет имеет свой уникальный:

- А) Web-адрес;
- В) TCP/IP-адрес;
- С) IP-адрес;
- Д) TCP-адрес;
- Е) URL-адрес.

5. Какие ресурсы может предоставить рабочая станция другим пользователям одноранговой локальной сети?

- А) Никакие;
- В) оперативную память;

- С) жесткий диск;
D) оперативную память и жесткий диск;
E) сетевую плату.
6. Какие из устройств: 1) сетевая плата; 2) сетевой адаптер; 3) модем; 4) телефон; 5) сетевое программное обеспечение необходимы для подключения домашнего компьютера к глобальной сети Интернет?
A) 3, 4, 5; B) 1, 3, 4;
C) 2, 3, 4, 5; D) 1, 4, 5.
E) 2, 3, 5;
7. Что не повлияет на скорость вашей работы в Интернете?
A) Возможности вашего модема;
B) пропускная способность информационного канала, исходящего от провайдера;
C) количество пользователей у вашего провайдера, работающих одновременно с вами;
D) количество гиперссылок у считываемого документа;
E) быстродействие вашего компьютера.
8. Имеющийся у вас модем обеспечивает прием/передачу сообщений со скоростью 12 Кбит·с. Скорость передачи сообщений по телефонной линии — 23 Кбит·с. Пропускная способность системы равна:
A) 23 Кбит·сек;
B) 12 Кбит·сек;
C) 15 Кбит·сек;
D) 11 Кбит·сек;
E) не менее 12 Кбит·сек.
9. Какой(ие) тэг(и) способен(ны) изменить цвет фона документа?
A) <HTML>...</HTML>;
B) <BODY>...</BODY>;
C) <BODY>...</BODY> и ...;
D) <BODY>...</BODY> и ...</P>;
E) </COLOR>...</COLOR>.

10. Сколько разных по размеру слов «шрифт» покажет на экране фрагмент кода HTML?

```
<FONT size=3>
шрифт
<FONT size=+1>
шрифт
<FONT size=+1>
шрифт
<FONT size=+3>
шрифт
<FONT size=+1>
шрифт
<FONT size=+2>
шрифт
<FONT size=+2>
шрифт
</FONT></FONT></FONT></FONT></FONT></FONT>
></FONT>
</P>
```

A) 5; B) 1; C) 7; D) 4; E) 3.

Ответы на тест № 1

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	В	В	Д	С	С	А	Д	В	В	Д

Тест № 2

- В реляционной базе данных связь между таблицами организована через:
 - запросы;
 - общие строки;
 - условия поиска;
 - поля, связанные по смыслу;
 - условия сортировки.
- Структура реляционной базы данных изменяется при:
 - удалении любой записи из базы;
 - удалении любого поля;

- С) изменении любой записи;
 - Д) добавлении записи в базу;
 - Е) удалении всех записей.
3. Полем реляционной БД является:
- А) строка таблицы;
 - В) корень дерева;
 - С) дерево;
 - Д) столбец таблицы;
 - Е) ветви дерева.
4. Одним из основных типов информационных структур является:
- А) логическая;
 - В) база данных;
 - С) строковая;
 - Д) дерево;
 - Е) числовая.
5. Какие атрибуты (признаки) объекта должны быть отражены в информационной модели, описывающей клуб любителей плавания, если эта модель позволяет получить ответы на следующие вопросы: выяснить фамилии и возраст всех участников клуба, плавающих стилем «кроль»; выяснить общее количество соревнований, в которых участвовал спортсмен Иванов; выяснить фамилии женщин, одержавших более 10 побед в соревнованиях?
- А) Фамилия, пол, стиль плавания, количество соревнований, количество побед;
 - В) фамилия, пол, кроль, количество соревнований, количество побед;
 - С) фамилия, женщина, возраст, кроль, количество соревнований;
 - Д) Фамилия, женщина, мужчина, кроль, количество соревнований, количество побед;
 - Е) Фамилия, пол, возраст, стиль плавания, количество соревнований, количество побед.
6. В какой из перечисленных пар данные относятся к одному типу?

- A) 12.04.98 и 123;
 - B) 123 и 189;
 - C) Иванов и 1313;
 - D) ДА и TRUE (ИСТИНА);
 - E) $45 < 999$ и 54.
7. БД содержит информацию об учениках компьютерной школы: имя; номер группы; балл за тест; балл за задание; общее количество баллов. Какого типа должно быть поле «Общее количество баллов»?
- A) Символьное;
 - B) логическое;
 - C) числовое;
 - D) любого типа;
 - E) числовое или логическое.
8. Какие характеристики объекта «Склад» должны быть отражены в структуре реляционной базы данных, если необходимо получить следующую информацию: наименование и количество товара с истекшим сроком хранения (дата окончания срока хранения превысила текущую дату); наименование товара с ценой менее 70 руб.; наименование всех товаров на общую сумму более 2 000 руб. Построенная модель не должна содержать избыточную информацию.
- A) наименование, количество, цена, дата окончания срока хранения, общая сумма;
 - B) наименование, количество, цена, дата окончания срока хранения, текущая дата, общая сумма;
 - C) наименование, количество, цена, дата окончания срока хранения;
 - D) наименование, количество, цена, дата окончания срока хранения, текущая дата;
 - E) наименование, количество, цена, текущая дата, общая сумма.

9. Реляционная база данных задана таблицей:

	Код спортсмена	Код дистанции	Дата соревнования	Время (сек)
1	101	д02	11.12.98	56,6
2	104	д01	12.10.99	37
3	102	д02	11.12.98	56,1
4	103	д05	11.12.98	242,8
5	101	д04	13.01.99	181,1
6	102	д01	12.10.98	35,45

Сформулировать условие поиска, дающее сведения о спортсменах, принимавших участие в соревнованиях на дистанциях с кодами д01 и д03 не позднее 10.12.98.

- А) Код_дистанции=д01 И Код_дистанции=д03 И Дата_соревнования>10.12.98;
 В) (Код_дистанции=д01 ИЛИ Код_дистанции=д03) И Дата_соревнования>10.12.98;
 С) Код_дистанции=д01 И (Код_дистанции=д03 ИЛИ Дата_соревнования<=10.12.98);
 D) Код_дистанции=д01 И Код_дистанции=д03 И Дата_соревнования<=10.12.98;
 Е) (Код_дистанции=д01 ИЛИ Код_дистанции=д03) И Дата_соревнования<=10.12.98.

10. Реляционная база данных задана таблицами:

Таблица 1

	Регистр_номер	Фамилия	Пол	Возраст	Школа
1	100	Иванов	М	15	1
2	123	Сидоренко	Ж	16	27
3	133	Журавлев	М	16	77
4	199	Сергеев	М	15	98
5	121	Грач	Ж	17	303
6	145	Яценко	М	17	77

Таблица 2

	Регистр_номер	Балл_за_тест
1	100	
2	133	
3	121	
4	145	
5	199	
6	123	

Таблица 3

	Школа	Директор	Телефон
1	1	Петрова	33-55-77
2	77	Павлов	14-19-33
3	27	Кузнецова	11-56-89
4	98	Павлов	33-57-28
5	203	Кирюхина	14-61-90

Через какое поле и какие таблицы должны быть связаны между собой, чтобы можно было дать ответ на вопрос, сколько баллов за тест набрал ученик Сергеев?

- А) Таблицы 1 и 2; через поле «Фамилия»;
- В) таблицы 1 и 2; через поле «Регистр_номер»;
- С) таблицы 1 и 3; через поле «Школа»;
- Д) таблицы 1 и 2; через поле «Балл_за_тест»;
- Е) ответ можно дать, не организовывая связи между таблицами.

11. По некоторому условию в таблице 1 БД из задания 10 были выбраны записи 3 и 6. Выбрать условие поиска.

- А) Пол=М И Возраст<=17;
- В) Пол=М ИЛИ Возраст>15;
- С) Пол=М И Возраст>15;
- Д) НЕ(Пол=Ж И Возраст<=15);
- Е) Пол=М ИЛИ Возраст <=17.

12. Через какие поля и какие таблицы БД из задания 10 должны быть связаны между собой, чтобы можно было ответить на вопрос, кто является директорами школ, в которых учатся школьники, набравшие при тестировании более 20 баллов?

- А) Таблицы 1 и 2 через поле «Регистр_номер»; таблицы 1 и 3 через поле «Директор»;

- В) Таблицы 1 и 2 через поле «Балл_за_тест»; таблицы 1 и 3 через поле «Директор»;
- С) Таблицы 2 и 3 через поле «Балл_за_тест» и поле «Директор»;
- Д) ответ можно дать, не организовывая связи между таблицами;
- Е) Таблицы 1 и 2 через поле «Регистр_номер»; таблицы 1 и 3 через поле «Школа».
13. Сформулировать условие поиска для таблицы 1 БД из задания 10, дающее сведения об ученицах школ 1 и 77 в возрасте 15 лет.
- А) Пол=Ж ИЛИ (Школа=1 ИЛИ Школа=77) И Возраст=15;
- В) Пол=Ж И Школа=1 ИЛИ Школа=77 И Возраст=15;
- С) Пол=Ж И Школа=1 И Школа=77 И Возраст=15;
- Д) Пол=Ж И (Школа=1 ИЛИ Школа=77) И Возраст=15;
- Е) Пол=Ж И Школа=1 И Школа=77 ИЛИ Возраст=15.
14. Какие записи в таблице 1 БД из задания 10 будут выбраны по условию
Пол=М И Возраст>15 ИЛИ Пол=Ж?
- А) 1, 2; С) 5, 6; Е) 1, 3, 4, 6.
- В) 1, 2, 5, 6; Д) 2, 3, 5, 6;
15. В каком порядке будут располагаться записи таблицы 1 БД из задания 10 после сортировки по двум полям: «Пол (убыв.)+Возраст (возр.)»?
- А) 2, 5, 1, 3, 4, 6; Д) 2, 5, 1, 3, 6, 4;
- В) 1, 4, 3, 6, 2, 5; Е) 1, 2, 3, 6, 4, 5.
- С) 1, 4, 3, 2, 6, 5;
16. Сформулировать условие поиска для таблицы 1 БД из задания 10, дающее сведения об участниках тестирования (мальчиках и девочках), регистрационные номера которых меньше 120 или больше 150.
- А) Регистр_номер<120 ИЛИ Регистр_номер>150;

- В) Регистр_номер<120 И Регистр_номер>150 И (Пол=Ж ИЛИ Пол=М);
 С) Регистр_номер<120 И Регистр_номер>150 И (Пол=Ж И Пол=М);
 D) Регистр_номер<120 И Регистр_номер>150;
 Е) (Регистр_номер<120 ИЛИ Регистр_номер>150) И (Пол=Ж И Пол=М).

17. Реляционная база данных задана таблицей:

	Ф.И.О.	Пол	Возраст	Клуб	Спорт
1	Панько Л.П.	жен	22	Спарта	Футбол
2	Арбузов А.А.	муж	20	Динамо	Лыжи
3	Жиганова П.Н.	жен	19	Ротор	Футбол
4	Иванов О.Г.	муж	21	Звезда	Лыжи
5	Седова О.Л.	жен	18	Спарта	Биатлон
6	Бегаета С.И.	жен	23	Звезда	Лыжи

Сформулировать условие поиска, дающее сведения о всех лыжниках и биатлонистах в возрасте от 18 до 22 лет.

- А) (Спорт="Лыжи" ИЛИ Спорт="Биатлон") И (Возраст>=18) ИЛИ (Возраст<=22);
 В) (Спорт="Лыжи" И Спорт="Биатлон") ИЛИ (Возраст>=18) И (Возраст<=22);
 С) (Спорт="Лыжи" И Спорт="Биатлон") И (Возраст>=18) ИЛИ (Возраст<=22);
 D) (Спорт="Лыжи" И Спорт="Биатлон") И (Возраст>=18) И (Возраст<=22);
 Е) (Спорт="Лыжи" ИЛИ Спорт="Биатлон") И (Возраст>=18 И Возраст<=22).

18. Какие записи БД из задания 17 будут выбраны по условию

(Клуб = "Спарта" И Клуб = "Ротор") И НЕ (Пол = "жен") ?

- А) 3, 5; С) 2, 3, 4, 5; Е) таких записей нет.
 В) 1, 3, 5; D) 24.

19. В каком порядке будут идти записи БД из задания 17, если их отсортировать по двум полям: «Клуб (возр.)+Ф.И.О. (возр.)»?
 А) 2, 4, 6, 3, 1, 5;
 В) 1, 5, 3, 6, 4, 2;
 С) 5, 1, 3, 4, 6, 2;
 D) 2, 6, 4, 3, 1, 5;
 Е) 2, 6, 3, 4, 1, 5.
20. Даны фрагменты текста:
 54disk;disk22;5disk4;52disk.
 Расположить их в порядке убывания.
 А. 52disk;54disk;5disk4;disk22
 В. disk22;5disk4;54disk;52disk
 С. disk22;54disk;52disk;5disk4
 D. 52disk;5disk4;54disk;disk22
 Е. 52disk;54disk;disk22;5disk4

Ответы на тест № 2

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	D	B	D	D	E	C	C	C	E	B
№ вопроса	11	12	13	14	15	16	17	18	19	20
Ответ	C	E	D	D	B	A	E	E	D	B

Тест № 3

1. Сколько ячеек войдет в диапазон A5:D8?
 А) 2; В) 5; С) 8; D) 16; Е) 13.
2. Модем — это ..., согласующее работу ... и телефонной сети. Вместо многоточий надо вставить слова:
 А) устройство, программы;
 В) программа, компьютера;
 С) программное обеспечение, компьютера;
 D) устройство, дисковода;
 Е) устройство, компьютера.
3. Почтовый ящик абонента электронной почты — это:
 А) часть оперативной памяти на сервере;

- В) часть внешней памяти на сервере;
 С) часть ОП на рабочей станции;
 D) часть внешней памяти на рабочей станции;
 Е) номер телефона, с которым связан модем.
4. Адрес ячейки электронной таблицы — это
- А) имя, состоящее из любой последовательности символов;
 В) имя, состоящее из имени столбца и номера строки;
 С) адрес байта оперативной памяти, отведенного под ячейку;
 D) адрес машинного слова оперативной памяти, отведенного под ячейку;
 Е) имя, состоящее из номера столбца и номера строки.
5. Чтобы соединить два компьютера по телефонным линиям, необходимо иметь:
- А) модем на одном из компьютеров;
 В) модем и специальное программное обеспечение на одном из компьютеров;
 С) по модему на каждом компьютере;
 D) по модему на каждом компьютере и специальное программное обеспечение;
 Е) по два модема на каждом компьютере (настроенных, соответственно, на прием и передачу) и специальное программное обеспечение.
6. Арифметическое выражение $\frac{4^{\frac{5+3}{6}} \cdot 7}{2} / \frac{14 \cdot 23}{8}$ может быть записано в электронной таблице в следующем виде:
- А) $(4^{5+3}/6*7)/2/14*23/8$;
 В) $4^{(5+3)}/6*7/2/14*23/8$;
 С) $4^{(5+3)}/6*7/2*8/14*23$;
 D) $4^{(5+3)}/6*7/2*8/14/23$;
 Е) $4^{((5+3)/6)*7/2*8/14/23}$.

7. Числовая константа 300 000 может быть записана в виде:

- A) 3,0E+6;
- B) 3,0E+5;
- C) 0,3E+7;
- D) 30,E+5;
- E) 0,3E+5.

8. В ЭТ записано арифметическое выражение $2/3^2 - (13-6)/2/4$. Этому выражению соответствует математическая запись:

A) $\left(\frac{2}{3}\right)^2 - \frac{13-6}{2 \cdot 4};$

D) $\frac{2}{3^2} - \frac{13-6}{2 \cdot 4};$

B) $\frac{2}{3^2} - \frac{13-6}{2 / 4};$

E) $\left(\frac{2}{3}\right)^2 - \frac{13-6}{2 / 4};$

C) $\frac{2}{3^2} - \frac{13-6}{2};$

9. Числовая константа 12,3E+4 может быть записана в виде:

- A) 1230;
- B) 123;
- C) 123000;
- D) 12,3000;
- E) 0,00123.

10. Дан фрагмент ЭТ:

	A	B	C	D
1	1	8	5	9
2	9	5	3	4
3	0	0	9	5
4	3	24	3	6

Какое из утверждений истинно для этого фрагмента таблицы?

- A) В ячейку D4 введена формула $= (A1+B2+C3)/3$;
- B) в ячейку D1 введена формула $= \text{СУММ}(A2:B3)$;

С) в ячейку D2 введена формула

$=\text{СУММ}(B3:C4)/\text{СУММ}(A3:C3);$

Д) в ячейку D3 введена формула $=C3*C4-(C1-C2)/5;$

Е) в ячейку A4 введена формула $=\text{СУММ}(A1:C2)-1.$

11. Дан фрагмент ЭТ в режиме отображения формул:

	А	В
1	10	$=A1+A2$
2	20	
3	30	
4	40	

Чему будут равны значения в ячейках В2 и В3 (отображаемые после выхода из режима отображения формул), если в них было скопировано содержимое ячейки В1?

А) 30 и 30;

В) 50 и 70;

С) 30 и 50;

Д) 50 и 30;

Е) будет выдано сообщение об ошибке.

12—16. Дан фрагмент электронной таблицы:

	А	В	С	Д	Е
1	1		2		
2	3	9	3	24	
3	0,5				
4					

Для заданий 12—16 даны значения выражений x и y и дополнительная информация:

№	X	Y	Дополнительная информация
12	D3	B3	В ячейку D3 занесено выражение $=(A2+A1)/C1$, а в ячейку B3 — выражение $=A2+A1/C1$
13	C3	D3	В ячейку C3 занесено выражение $=C1*D2/A1$, а в ячейку D3 — выражение $=C1/A1*D2$

№	X	Y	Дополнительная информация
14	C3	D3	В ячейку C3 занесено выражение $=\text{СУММ}(A1:D2)$, а в ячейку D3 — выражение $=\text{СУММ}(A2:D2)$
15	E1	3	В ячейку E1 занесено выражение $=\text{СУММ}(A1:A3)/5*0,3$
16	E2	E3	В ячейку E2 занесено выражение $=A2*B2/C2$, а в ячейку E3 — выражение $=A2/C2*B2$

Для каждого из заданий 12—16 выбрать верное утверждение:

- А) значение X больше значения Y;
- В) значение X меньше значения Y;
- С) значения X и Y равны;
- Д) сравнение недопустимо.

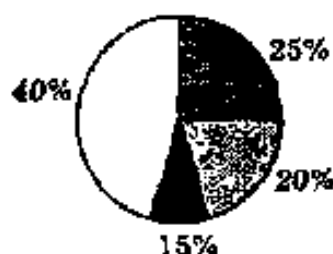
17. Дан фрагмент ЭТ в режиме отображения формул:

	A	B
1	1	5
2	$=A1+1$	$=2*B1$
3		

Какие формулы будут отображены в ячейках A3 и B3, если в ячейку A3 скопировав содержимое ячейки B2, а в ячейку B3 — содержимое ячейки A2?

- А) A3: $=A2+1$ B3: $=2*B2$;
- В) A3: $=2*A2$ B3: $=B2+1$;
- С) A3: $=2*A1$ B3: $=B1+1$;
- Д) A3: $=2*B1$ B3: $=A1+1$;
- Е) A3: $=2*B2$ B3: $=A2+1$.

18. Для какой из приведенных ниже таблиц может быть построена следующая круговая диаграмма?



A)

	A
1	100
2	$=A1*0,2$
3	$=A1*0,15$
4	$=A1*0,4$

D)

	A
1	100
2	$=A1*0,8$
3	$=A2*0,6$
4	$=A3*1,6$

B)

	A
1	100
2	$=A1*0,8$
3	$=A1*0,6$
4	$=A1*1,6$

E)

	A
1	100
2	$=A1*0,2$
3	$=A1*0,15$
4	$=A1*0,4$

C)

	A
1	25
2	$=A1*0,2$
3	$=A1*0,15$
4	$=A1*0,4$

19. Каждый компьютер, подключенный к глобальной сети Интернет, имеет свой уникальный:

- A) Web-адрес;
- B) TCP (IP-адрес;
- C) IP-адрес;
- D) TCP-адрес;
- E) URL-адрес.

20. Для просмотра Web-страниц предназначены:

- A) поисковые серверы;
- B) программы-браузеры;
- C) телеконференции;
- D) почтовые программы;
- E) провайдеры.

21. Базы данных — это:

- А) информационные структуры, хранящиеся во внешней памяти;
В) программные средства, позволяющие организовывать информацию в виде таблиц;
С) программные средства, обрабатывающие табличные данные;
D) программные средства, осуществляющие поиск информации;
Е) информационные структуры, хранящиеся в ОП.
22. В реляционной БД информация организована в виде:
А) сети;
В) иерархической структуры;
С) файла;
D) дерева;
Е) прямоугольной таблицы.
23. В коробке меньше 9 шаров, но больше 3 шаров. Сколько шаров может быть в коробке?
А) 3; В) 9; С) 2; D) 5; Е) 10.
24. БД содержит информацию об учениках школы: фамилиях, классах, баллах за тест, баллах за практическое задание, общем количестве баллов. Какого типа должно быть поле «общее количество баллов»?
А) символьное; С) числовое; Е) дата.
В) логическое; D) любого типа;
25. Какие атрибуты (признаки) объекта должны быть отражены в информационной модели, описывающей хобби ваших одноклассников, если эта модель позволяет получить ответы на следующие вопросы:
— указать возраст всех детей, увлекающихся компьютером;
— указать имена девочек, увлекающихся пением;
— указать фамилии мальчиков, увлекающихся хоккеем.
А) имя, пол, хобби;
В) фамилия, пол, хоккей, пение, возраст;

- С) имя, пол, хобби, возраст;
 D) имя, возраст, хобби;
 E) фамилия, имя, пол, возраст, хобби.

26. Реляционная база данных задана таблицей:

	Ф.И.О.	Пол	Возраст	Клуб	Спорт
1	Панько Л.П.	жен	22	Спарта	Футбол
2	Арбузов А.А.	муж	20	Динамо	Лыжи
3	Жиганова П.Н.	жен	19	Ротор	Футбол
4	Иванов О.Г.	муж	21	Звезда	Лыжи
5	Седова О.Л.	жен	18	Спарта	Биатлон
6	Бегаяева С.И.	жен	23	Звезда	Лыжи

Какие записи будут выбраны по условию
 (клуб= "Спарта" И клуб= "Ротор") И НЕ (пол=
 "жен")?

- A) 3, 5; B) 1, 3, 5; C) 2, 3, 4, 5; D) 2, 4.
 E. таких записей нет

27. Какие записи будут выбраны по условию
 Спорт= "Лыжи" И Пол= "жен" ИЛИ Возраст<20
 A) 2, 3, 4, 5, 6; C) 1, 3, 5, 6; E) 2, 3, 5, 6
 B) 3, 5, 6 D) таких записей нет.

28. БД содержит информацию о собаках из клуба собаководства: кличках, породах, датах рождения, поле, количестве медалей. Какого типа должны быть поля?

- A) Текстовое, текстовое, числовое, текстовое, числовое;
 B) текстовое, текстовое, дата, текстовое, числовое;
 C) текстовое, текстовое, дата, логическое, числовое;
 D) текстовое, текстовое, числовое, логическое, числовое;
 E) текстовое, текстовое, дата, логическое, текстовое.

29. Реляционная БД задана таблицей:

	Название	Категория	Кинотеатр	Начало сеанса
1	Буратино	х/ф	Рубин	14
2	Кортик	х/ф	Искра	12
3	Винни-Пух	м/ф	Экран	9
4	Дюймовочка	м/ф	Россия	10
5	Буратино	х/ф	Искра	14
6	Ну, погоди	м/ф	Экран	14
7	Два капитана	х/ф	Россия	16

Выбрать главный ключ для таблицы (допуская, что в кинотеатре один зал):

- А. название+кинотеатр
- В. кинотеатр+начало_сеанса
- С. название+начало_сеанса
- Д. кинотеатр
- Е. начало_сеанса

30. В каком порядке будут идти записи в БД из задания 28, если их отсортировать по двум ключам: «название (возр.) + кинотеатр (возр.)»?

- А) 1, 5, 3, 4, 7, 2, 6;
- В) 5, 1, 3, 7, 4, 2, 6;
- С) 6, 2, 4, 7, 3, 1, 5;
- Д) 6, 2, 7, 4, 3, 1, 5;
- Е) 2, 5, 4, 7, 1, 3, 6.

Ответы на тест № 3

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	Д	Е	В	В	Д	Е	В	Д	С	С
№ вопроса	11	12	13	14	15	16	17	18	19	20
Ответ	В	В	С	А	В	С	В	В	С	В
№ вопроса	21	22	23	24	25	26	27	28	29	30
Ответ	А	Е	Д	С	Е	Е	В	В	В	В

План уроков

9 класс

№ урока	Тема	Вид занятий	Вид отчета
	Модуль 1. Содержательный подход к измерению информации. Вероятность и информация. Пакет создания презентаций		
1	Содержательный подход к измерению информации	Теория	Д/з № 1
2	Знакомство с пакетом создания презентаций. Создание простейшей презентации	Практика	П/з № 1
3	Вероятность и информация (содержательный подход)	Теория	Д/з № 2
4	Разработка заданной презентации. Использование организационной диаграммы	Практика	П/з № 2
5	Формула Шеннона. Решение задач	Теория	Д/з № 3
6	Использование гиперссылок и кнопок перехода при разработке презентации	Практика	П/з № 3
7	Тестирование		Тест №1
	Модуль 2. Базы знаний. Язык логического программирования Пролог		
8	Понятие искусственного интеллекта и базы знаний	Теория	Д/з № 4
9	Язык логического программирования Пролог. Основные понятия Пролога: факт, правило, цель. Синтаксис Турбо Пролога	Теория	Д/з № 5
10	Создание простейших баз знаний на Турбо Прологе и формирование к ним целей	Практика	П/з № 4
11	Конъюнкция и дизъюнкция в Прологе	Теория	Д/з № 6

№ урока	Тема	Вид занятий	Вид отчета
12	Использование конъюнкции и дизъюнкции в Турбо Прологе для решения задач	Практика	П/з № 5
13	Тестирование		Тест № 2
	<i>Модуль 3. Управление и кибернетика. Алгоритм и его свойства. Алгоритмические структуры. Графический учебный исполнитель</i>		
14	Кибернетика. Кибернетическая модель управления. Системы управления с использованием компьютеров (АСУ и САУ)	Теория	Д/з № 7
15	Определение и свойства алгоритма. Линейные алгоритмы. Использование алгоритмического языка и блок-схем для записи алгоритмов	Теория	Д/з № 8
16	Графический учебный исполнитель алгоритмов. Разработка линейных алгоритмов	Теория + практика	Д/з № 9, П/з № 6
17	Вспомогательные алгоритмы. Последовательная детализация	Теория	Д/з № 10
18	Учебный исполнитель алгоритмов. Последовательная детализация	Практика	П/з № 7
19	Команда цикла. Циклические алгоритмы	Теория	Д/з № 11
20	Учебный исполнитель алгоритмов. Команда цикла. Циклические алгоритмы	Практика	П/з № 8
21	Ветвления. Циклы в сочетании с ветвлениями	Теория	Д/з № 12
22	Учебный исполнитель алгоритмов. Ветвления. Циклы в сочетании с ветвлениями	Практика	П/з № 9

№ урока	Тема	Вид занятий	Вид отчета
23	Зачетное занятие по алгоритмизации	Практика	П/з № 10
24	Тестирование		Тест № 3
	<i>Модуль 4. Перевод чисел из одной системы счисления в другую. Понятие о программном управлении компьютером. Язык машинных команд. История развития ЭВМ</i>		
25	Перевод целых чисел из одной системы счисления в другую	Теория	Д/з № 13
26	Двоичная система счисления и двоичная арифметика. Связь между двоичной, восьмеричной и шестнадцатеричной системами счисления	Теория	Д/з № 14
27	Контрольная работа по системам счисления	Практика	
28	Структура памяти учебного компьютера «УК Нейман». Представление целых чисел в памяти компьютера	Теория	Д/з № 15
29	Язык машинных команд	Теория	Д/з № 16
30	Учебный компьютер «УК Нейман». Язык машинных команд	Практика	П/з № 11
31	Устройство и работа процессора	Теория	Д/з № 17
32	Язык машинных команд	Практика	П/з № 12
33	История развития ЭВМ	Теория	Д/з № 18
34	Итоговое тестирование		

Урок 1

Теория. Содержательный подход к измерению информации. (45 мин)

Изучаемые вопросы:

- ☐ Неопределенность знаний.
- ☐ Равновероятные события.
- ☐ Метод половинного деления.
- ☐ Формула измерения количества информации в сообщении об одном из нескольких равновероятных событий.
- ☐ Измерение информации в сообщении о нескольких равновероятных событиях.

Домашнее задание:

- ☐ Читать: [5] — § 1.1.
- ☐ Выполнить: домашнее задание № 1.

Урок 2

Практика. Знакомство с пакетом создания презентаций. Создание простейшей презентации. (Выполнение практического задания № 1.) (45 мин)

Приобретаемые умения и навыки:

- 1) выбор дизайна оформления презентации;
- 2) выбор шаблона кадра;
- 3) добавление текстовых и графических объектов в кадр и их редактирование;
- 4) создание эффектов анимации;
- 5) установка эффектов при смене кадров;
- 6) установка времени задержки кадров;
- 7) просмотр презентации.

Урок 3

Теория. Вероятность и информация (содержательный подход). (45 мин)

Изучаемые вопросы:

- ☐ Понятие логарифма.
- ☐ Формула Хартли.
- ☐ Понятие вероятности.

- Формула измерения количества информации в сообщении об одном из нескольких неравновероятных событий.

Домашнее задание:

- Читать: [5] — § 1.2.
- Выполнить: домашнее задание № 2.

Урок 4

Практика. Разработка заданной презентации. Использование организационной диаграммы. (Выполнение практического задания № 2.) (45 мин)

Приобретаемые умения и навыки:

- 1) добавление организационной диаграммы в кадр,
- 2) добавление, редактирование и удаление элементов диаграммы.

Урок 5

Теория. Формула Шеннона. Решение задач. (45 мин)

Изучаемые вопросы:

- Понятие частотной характеристики символа в заданном алфавите.
- Определение информационного веса символа.
- Формула измерения объема информации в тексте при использовании вероятностного подхода.
- Вычисление средней информативности символа в алфавите (формула Шеннона).
- Сравнение информационных объемов текста при использовании формул Шеннона и Хартли.

Домашнее задание:

- Читать: [5] — § 1.3, [2] — раздел 1.4.
- Выполнить: домашнее задание № 3.

Урок 6

Практика. Использование гиперссылок и кнопок перехода при разработке презентации. (Выполнение практического задания № 3.) (45 мин)

Приобретаемые умения и навыки:

- 1) преобразование текстовой и графической информации в гиперссылку;
- 2) использование автофигур для создания кнопок перехода.

Урок 7

Тестирование. Тест № 1 по теме «Содержательный подход к измерению информации. Вероятность и информация». (45 мин)

Урок 8

Теория. Понятие искусственного интеллекта и базы знаний. (45 мин)

Изучаемые вопросы:

- ☐ Искусственный интеллект как раздел информатики и его цели.
- ☐ Понятие базы знаний.
- ☐ Экспертные системы и их назначение.
- ☐ Понятие факта и правила.

Домашнее задание:

- ☐ Читать: [5] — § 11.1, 11.2.
- ☐ Выполнить: домашнее задание № 4.

Урок 9

Теория. Язык логического программирования Пролог. Основные понятия Пролога: факт, правило, цель. Синтаксис Турбо Пролога. (45 мин)

Изучаемые вопросы:

- ☐ Пролог — язык логического программирования.
- ☐ Описание фактов на языке Пролог.
- ☐ Описание простейших правил на языке Пролог.
- ☐ Понятие цели и их типы.
- ☐ Структура простейшей базы знаний на языке Турбо Пролог.
- ☐ Синтаксис Турбо Пролога.

Домашнее задание:

- ☐ Читать: [5] — § 11.3, 11.4.
- ☐ Выполнить: домашнее задание № 5.

Урок 10

Практика. Создание простейших баз знаний на Турбо Прологе и формирование к ним целей. (Выполнение практического задания № 4.) (45 мин)

Приобретаемые умения и навыки:

- 1) создание новой базы знаний на Турбо Прологе;
- 2) загрузка существующей базы знаний;
- 3) формирование целей к базе знаний.

Урок 11

Теория. Конъюнкция и дизъюнкция в Прологе. (45 мин)

Изучаемые вопросы:

- Использование конъюнкции и дизъюнкции при формировании правил.
- Использование конъюнкции и дизъюнкции при формировании целей.
- Использование пустого аргумента при формировании правил и целей.
- Правила ответов на цели при использовании конъюнкции и дизъюнкции
- Понятие механизма вывода.

Домашнее задание:

- Читать: [5] — § 11.3, 11.4.
- Выполнить: домашнее задание № 6.

Урок 12

Практика. Использование конъюнкции и дизъюнкции в Турбо Прологе для решения задач. (Выполнение практического задания № 5.) (45 мин)

Приобретаемые умения и навыки:

- 1) формирование правил с использованием логических операций умножения и сложения;
- 2) формирование целей с использованием логических операций умножения и сложения;
- 3) использование пустого аргумента при формировании правил и целей.

Урок 13

Тестирование. Тест № 2 по теме «Базы знаний. Язык логического программирования Пролог». (45 минут)

Урок 14

Теория. Кибернетика. Кибернетическая модель управления. Системы управления с использованием компьютеров (АСУ и САУ). (45 мин)

Изучаемые вопросы:

- ☐ Кибернетика как раздел информатики.
- ☐ Понятия управляющего и управляемого объектов.
- ☐ Схема управления без обратной связи.
- ☐ Схема управления с обратной связью.
- ☐ Автоматизированные системы управления (АСУ).
- ☐ Системы автоматического управления (САУ).
- ☐ Цифро-аналоговый и аналого-цифровой преобразователи (ЦАП и АЦП), их назначение.

Домашнее задание:

- ☐ Читать: [1] — § 38, 39.
- ☐ Выполнить: домашнее задание № 7.

Урок 15

Теория. Определение и свойства алгоритма. Линейные алгоритмы. Использование алгоритмического языка и блок-схем для записи алгоритмов. (45 мин)

Изучаемые вопросы:

- ☐ Понятие алгоритма.
- ☐ Свойства алгоритма (точность, корректность, понятность, конечность).
- ☐ Понятие линейного алгоритма.
- ☐ Учебный Алгоритмический язык как способ записи алгоритмов.
- ☐ Графическое описание алгоритмов (блок-схемы).

Домашнее задание:

- ☐ Читать: [1] — § 40.
- ☐ Выполнить: домашнее задание № 8.

Урок 16

Теория. Графический учебный исполнитель алгоритмов (рассматриваются исполнители типа ГРИС, реализованные на компьютере: Кенгуренок, Чертежник, Робот и т. п.). (15 мин)

Изучаемые вопросы:

- ☐ Графический исполнитель алгоритмов (ГРИС).
- ☐ Режимы управления ГРИС.
- ☐ Команды ГРИС.

Практика. Графический учебный исполнитель алгоритмов. Режимы работы. Простые команды. Линейные алгоритмы. (Выполнение практического задания № 6.) (30 мин)

Приобретаемые умения и навыки:

- 1) запуск программы ГРИС;
- 2) смена командного и программного режимов работы;
- 3) разработка линейного алгоритма для поставленной задачи;
- 4) набор и редактирование текста программы;
- 5) начальная установка исполнителя;
- 6) запуск программы на исполнение;
- 7) работа в режиме отладки.

Домашнее задание:

- ☐ Читать: [5] — § 41.
- ☐ Выполнить: домашнее задание № 9.

Урок 17

Теория. Вспомогательные алгоритмы и подпрограммы. Последовательная детализация. (45 мин)

Изучаемые вопросы:

- ☐ Понятие вспомогательного алгоритма и причины его использования.
- ☐ Описание и вызов вспомогательного алгоритма на учебном Алгоритмическом языке.
- ☐ Описание и вызов вспомогательного алгоритма на языке блок-схем.
- ☐ Метод последовательной детализации.

Домашнее задание:

- Читать: [5] — § 42.
- Выполнить: домашнее задание № 10.

Урок 18

Практика. Учебный исполнитель алгоритмов. Последовательная детализация. (Выполнение практического задания № 7.) (45 мин)

Приобретаемые умения и навыки:

использование метода пошаговой детализации при написании алгоритма.

Урок 19

Теория. Команда цикла. Циклические алгоритмы. (45 мин)

Изучаемые вопросы:

- Команда цикла и ее запись на учебном Алгоритмическом языке.
- Блок-схемы алгоритмов с использованием циклов.

Домашнее задание:

- Читать: [5] — § 43.
- Выполнить: домашнее задание № 11.

Урок 20

Практика. Учебный исполнитель алгоритмов. Команда цикла. Циклические алгоритмы. (Выполнение практического задания № 8.) (45 мин)

Приобретаемые умения и навыки:

использование циклов при написании алгоритма.

Урок 21

Теория. Ветвления. Циклы в сочетании с ветвлениями. (45 мин)

Изучаемые вопросы:

- Команды ветвления (полного и неполного).

Домашнее задание:

- Читать: [5] — § 44.
- Выполнить: домашнее задание № 12.

Урок 22

Практика. Учебный исполнитель алгоритмов. Ветвления. Циклы в сочетании с ветвлениями. (Выполнение практического задания № 9.) (45 мин)

Приобретаемые умения и навыки:

использование ветвлений в сочетании с циклами при написании алгоритма.

Урок 23

Практика. Зачетное занятие по алгоритмизации. (Выполнение практического задания № 10.) (45 мин)

Урок 24

Тестирование. Тест № 3 по теме «Управление и кибернетика. Алгоритм и его свойства. Алгоритмические структуры. Графический учебный исполнитель». (45 мин)

Урок 25

Теория. Перевод целых чисел из одной системы счисления в другую. (45 мин)

Изучаемые вопросы:

- ☐ Перевод целых чисел из десятичной системы счисления в любую другую.
- ☐ Перевод целых чисел из любой системы счисления в десятичную.

Домашнее задание:

- ☐ Читать: [5] — § 45.
- ☐ Выполнить: домашнее задание № 13.

Урок 26

Теория. Двоичная система счисления и двоичная арифметика. Связь между двоичной, восьмеричной и шестнадцатеричной системами счисления. (45 мин)

Изучаемые вопросы:

- ☐ Связь двоичной, восьмеричной и шестнадцатеричной систем счисления.
- ☐ Двоичная арифметика.

Домашнее задание:

- ☐ Читать: [5] — § 46.
- ☐ Выполнить: домашнее задание № 14.

Урок 27

Практика. Контрольная работа по системам счисления. (45 мин)

Приобретаемые умения и навыки:

- ☐ перевод целых чисел из десятичной системы счисления в любую другую.
- ☐ Перевод целых чисел из любой системы счисления в десятичную.
- ☐ Двоичная арифметика.

Урок 28

Теория. Структура памяти учебного компьютера «УК Нейман». Представление целых чисел в памяти компьютера. (45 мин)

Изучаемые вопросы:

- ☐ Ячейка как структурная единица памяти, адрес ячейки.
- ☐ Структура памяти «УК Нейман».
- ☐ Понятие машинного слова.
- ☐ Представление положительных целых чисел в памяти компьютера.

Домашнее задание:

- ☐ Читать: [5] — § 47.
- ☐ Выполнить: домашнее задание № 15.

Урок 29

Теория. Язык машинных команд. (45 мин)

Изучаемые вопросы:

- ☐ Понятие языка машинных команд.
- ☐ Структура машинной команды.
- ☐ Присваивание, ввод и вывод на языке машинных команд.

Домашнее задание:

- ☐ Читать: [5] — § 48.
- ☐ Выполнить: домашнее задание № 16.

Урок 30

Практика. Учебный компьютер «УК Нейман». Язык машинных команд. (Выполнение практического задания № 11.) (45 мин)

Приобретаемые умения и навыки:

- 1) запуск программы УК «Нейман»;
- 2) редактирование программы;
- 3) запуск программы на исполнение;
- 4) разработка линейного алгоритмы на ЯМК.

Урок 31

Теория. Устройство и работа процессора. (45 мин)

Изучаемые вопросы:

- ☐ Назначение процессора.
- ☐ Состав процессора.
- ☐ Цикл работы процессора.

Домашнее задание:

- ☐ Читать: [5] — § 49.
- ☐ Выполнить: домашнее задание № 17.

Урок 32

Практика. Язык машинных команд. (Выполнение практического задания № 12.) (45 мин)

Приобретаемые умения и навыки:

- 1) пошаговая отладка программы;
- 2) объяснение правильности полученного результата.

Урок 33

Теория. История развития ЭВМ. (45 мин)

Изучаемые вопросы:

- ☐ История развития вычислительной техники в докомпьютерный период.
- ☐ Поколения ЭВМ.
- ☐ Микропроцессор и персональные компьютеры.
- ☐ Перспективы развития компьютерной техники.

Домашнее задание:

- ☐ Читать: [5] — § 50.

□ Выполнить: домашнее задание № 18.

Урок 34

Тестирование. Итоговое тестирование № 4.
(45 мин)

Домашние задания

Домашнее задание № 1

1. В коробке лежали 64 фломастера разных цветов. Какое количество информации содержит сообщение о том, что из коробки достали красный фломастер?
2. В доме 16 этажей. На каждом этаже по несколько квартир. Сообщение о том, что Саша живет в квартире № 40, содержит 6 битов информации. Сколько квартир на каждом этаже?
3. [2]: раздел 1.3, задача № 12.
При угадывании целого числа в некотором диапазоне было получено 6 битов информации. Сколько чисел содержит этот диапазон?
4. Какое количество информации содержит сообщение о том, что из колоды в 32 карты достали даму треф и короля пик? Решите задачу для двух случаев:
а) карты достаются одновременно;
б) карты достаются по очереди.

Домашнее задание № 2

1. [2]: раздел 1.4, задача № 1.
2. [2]: раздел 1.4, задача № 4.
3. [2]: раздел 1.4, задача № 5.
4. [2]: раздел 1.4, задача № 15.

Домашнее задание № 3

1. Для приведенного ниже текста определите частотные характеристики всех букв алфавита и оформить результат в виде таблицы.

Через некоторое время (может, минут, а может, лет) забвение стало менее беспросветным, и Ричард Авери понял, что видит сон. Во тьме заискрились смутные, полусформировавшиеся образы.

Он увидел звезды. Он по-настоящему увидел звезды. Целые хороводы звезд — ярких и ослепительных, застывших в полном пустоты величии огромной звездной туманности. Он плыл вдаль по космической реке. Она вынесла его к самому краю космоса, и островки вселенных — невообразимые чаши света и пыли — понеслись мимо ледяными водопадами творения.

Было слишком холодно. Нет, не физически холодно. Духовно холодно. Его полупроснувшийся разум отвергал картины страшного в своей грандиозности великолепия. Он жадно пытался найти вокруг смысл, облегчение, точку отсчета. Вот он подплыл к солнцу, и солнце родило планеты. Одна из них была белой от облаков, голубой и зеленой от океанов, красной, и коричневой, и желтой от островов.

2. Определите полный информационный объем текста из задачи № 1 по формуле Шеннона.
3. Используя частотный словарь русского языка (см. [5]: § 1.3), определите количество информации во фразе «Повторение — мать учения».

Домашнее задание №4

1. [5]: § 11.1, вопрос 7.
2. [5]: § 11.2, вопрос 2.
3. [5]: § 11.2, вопрос 3.
4. [5]: § 11.2, вопрос 4.

Домашнее задание № 5

1. Опишите базу знаний, содержащую следующие факты и правила:

- ☐ Петр учится в университете;
- ☐ Мария учится в музыкальном училище;
- ☐ Евгений учится там же, где и Петр;
- ☐ Елена учится там же, где и Мария.

Сформулируйте на Прологе вопросы и ответьте на них:

- ☐ Учится ли Мария в музыкальном училище?
- ☐ Учится ли Евгений в музыкальном училище?
- ☐ Кто учится в университете?
- ☐ Кто учится в музыкальном училище?

2. Опишите на Турбо Прологе базу знаний, содержащую следующие факты и правила:

- ☐ Саша — ученик Петрова;
- ☐ Петя — ученик Сергеева;
- ☐ Алеша — ученик Иванова;
- ☐ Сергей — ученик того же учителя, что и Петя;
- ☐ Роман — ученик того же учителя, что и Алеша;
- ☐ Андрей — ученик того же учителя, что и Сергей.

Сформулируйте правило, пользуясь которым можно ответить на вопросы:

- ☐ Кто учитель Саши?
- ☐ Кто ученик Иванова?
- ☐ Является ли Сергеев учителем Андрея?
- ☐ Является ли Роман учеником Петрова?
- ☐ Кто какому учителю приходится учеником?

Запишите эти вопросы на Турбо Прологе и ответьте на них.

Словарь: ученик — pupil; учитель — teacher.

3. Дано описание базы знаний на Турбо Прологе. Перепишите, исправив ошибки.

```
predicates
    father(symbol)
clauses
    father(peter; oleg) .
```

```

father(sasha, SERGEY).
mother(olga, lena).
mother(olga, natasha).
mother(lena, irina).
daughter(X, Y) :- mother(X, X).
goal: daughter(irina, X)
           no mother(olga, X)           X=Lena

```

4. Опишите на Турбо Прологе одну из приведенных ниже баз знаний (на выбор):

- а) «Столицы государств»;
- б) «Победители чемпионатов мира по футболу»;
- в) «Авторы книг»;
- г) «Столицы олимпиад».

Опишите не менее 6 фактов. Сформулируйте к данной базе по две цели первого и второго типов и ответьте на них.

Домашнее задание № 6

1. Даны факты:

- ☐ мама(лена, оля);
- ☐ мама(лена, тая);
- ☐ мама(оля, юля);
- ☐ мама(оля, маша);
- ☐ мама(тая, катя).

Сформулируйте правила, позволяющие ответить на вопросы:

- ☐ Кто кому приходится бабушкой?
- ☐ Кто сестра Оли?
- ☐ Кто тетя Кати?

Задайте эти вопросы на Прологе и ответьте на них.

2. Для базы знаний из задания № 1 ответьте на вопросы:

- а) ?тетя(X, юля)
- б) ?тетя(тая, катя)
- в) ?бабушка(X, катя); сестра(X, тая)
- г) ?сестра(юля, маша); сестра(X, тая)
- д) ?тетя(тая, маша); тетя(тая, катя)

3. Опишите базу знаний, содержащую сведения об успеваемости трех учеников вашего класса по любым четырем предметам. Запишите правило, определяющее понятие «отличник» (все оценки — «5»).
4. Дана система родственных связей, описанная в виде следующих фактов:
 - отец (андрей, вера)
 - отец (андрей, алексей)
 - жена (ольга, андрей)
 - пол (андрей, муж)
 - пол (алексей, муж)
 - пол (вера, жен)
 - пол (ольга, жен)Для данной базы знаний опишите правила, определяющие следующие понятия: мать, сын, дочь, муж, сестра, брат.

Домашнее задание № 7

1. Дополните приведенный ниже список и укажите, какой объект будет управляющим, а какой — управляемым:
 - а) оркестр — ...;
 - б) автобус — ...;
 - в) наездник — ...;
 - г) тренер — ...;
 - д) режиссер —
2. Приведите по 2-3 примера кибернетических систем:
 - а) без обратной связи;
 - б) с обратной связью.
4. Какую структуру может иметь управляющий алгоритм:
 - а) без обратной связи;
 - б) при наличии обратной связи?
3. В чем различие между автоматизированными системами управления (АСУ) и системами автоматического управления (САУ)?

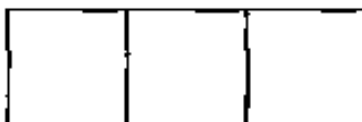
4. Придумайте 2-3 примера ситуаций, в которых можно было бы использовать САУ.

Домашнее задание № 8

1. Назовите исполнителей следующих видов работы: уборка мусора во дворе, перевозка пассажиров, выдача заработной платы, прием экзаменов, сдача экзаменов, обучение детей в школе.
2. [2]: раздел 4.1, задача № 6.
3. Перечислите основные свойства алгоритма.
4. Есть исполнитель Перевозчик, который перевозит через реку волка, козу и капусту. Напишите алгоритм перевоз через реку волка, козы и капусты, если СКИ Перевозчика содержит 5 команд: ВЗЯТЬ КОЗУ, ВЗЯТЬ ВОЛКА, ВЗЯТЬ КАПУСТУ, ВЫСАДИТЬ, ПЕРЕПЛЫТЬ. В лодку может поместиться только один предмет или животное. Нельзя оставлять на берегу одних волка с козой и козу с капустой.

Домашнее задание № 9

1. В чем различие программного и командного режимов управления исполнителем?
2. [2]: раздел 4.2, задача № 39.
3. Напишите на языке ГРИС алгоритм для изображения следующего ниже рисунка. Используйте не более 15 команд. Ширина фигуры — 3 шага, высота — 1 шаг.



4. Нарисуйте фигуру, которая получится после выполнения алгоритма (на языке ГРИС, исходное направление — вправо): шаг шаг поворот шаг шаг поворот шаг шаг поворот шаг шаг поворот поворот поворот шаг шаг.

Домашнее задание № 10

1. Используя язык ГРИС напишите программу изображения слова «НАГАН». Для написания программы используйте метод пошаговой детализации. Высота букв — 4 шага, ширина — 2 шага. Укажите начальное направление движения ГРИС.

2. Дан алгоритм на языке ГРИС:

шаг шаг поворот поворот поворот шаг шаг прыжок прыжок прыжок шаг шаг поворот поворот поворот шаг шаг прыжок прыжок прыжок шаг шаг поворот поворот поворот шаг шаг прыжок прыжок прыжок шаг шаг поворот поворот поворот шаг шаг прыжок прыжок прыжок

Запишите этот же алгоритм с использованием метода пошаговой детализации.

3. Дан алгоритм на языке ГРИС (исходное положение — вверх):

сделай А
сделай В
сделай А
сделай В
сделай А
сделай В
сделай А

После выполнения этого алгоритма получается следующий рисунок (расстояние между квадратами в ширину и в высоту — 1 шаг):



Напишите тексты процедур А и Б, используя минимально возможное число команд.

4. Решить задачу № 4 из домашнего задания № 9 с использованием вспомогательного алгоритма (используйте язык блок-схем).

Домашнее задание № 11

1. [2]: раздел 4.2, задача № 44.
2. Запишите блок-схемы алгоритмов из задачи № 1.
3. [2]: раздел 4.2, задача № 45.
4. Придумайте задачу для ГРИС, для решения которой необходимо использовать циклы.

Домашнее задание № 12

1. ГРИС находится в одном из углов поля. Направление произвольное. Что будет изображено на экране после выполнения следующего алгоритма (опишите все возможные варианты)?
если впереди край то
 поворот
иначе
 прыжок
 прыжок
конец ветвления
пока впереди не край
 шаг
конец цикла
2. Запишите блок-схему алгоритма из задачи № 1.
3. В чем отличие полного и неполного ветвлений?
4. Напишите для ГРИС программу, по которой вдоль границ поля будет нарисована пунктирная рамка.

Домашнее задание № 13

1. Переведите десятичное число 415 в двоичную, восьмеричную и шестнадцатеричную системы счисления.
2. Переведите числа 10011_2 , 314_8 , $A8_{16}$ в десятичную систему счисления.

3. [2]: раздел 1.5, задача № 15.

4. [2]: раздел 1.5, задача № 20.

Домашнее задание № 14

1. [2]: раздел 1.5, задача № 40.

2. [2]: раздел 1.5, задача № 39.

3. [2]: раздел 1.5, задача № 47.

4. [2]: раздел 1.5, задача № 48.

Домашнее задание №15

1. [5]: § 44, задача 3.

2. [5]: § 44, задача 4.

3. [5]: § 44, задача 7.

4. Объем оперативной памяти компьютера равен 1 Мбайту, а адрес последнего машинного слова — 1 048 574. Чему равен размер машинного слова?

Домашнее задание №16

1. [5]: § 48, задача 8.

2. [5]: § 48, задача 10.

3. Напишите программу на ЯМК для вычисления значения X по формуле $X = A^2 + 3B^2 - 18AC$.

4. Что будет выведено на экран после выполнения программы (ответ дайте в шестнадцатеричном виде)?

адрес	КОП	A1	A2	A3
00	01	24	28	40
04	01	28	2C	44
08	03	40	44	44
0C	02	40	2C	48
10	01	44	48	48
14	02	48	40	48
18	00	48	00	FC
20	77	-	-	-
24	00	00	00	0A
28	00	00	00	05
2C	00	00	00	10

Домашнее задание № 17

1. Перечислите устройства, входящие в состав процессора «УК Нейман» и назовите их функции.
2. Что такое цикл работы процессора и по какому условию он завершается?

Домашнее задание № 18

1. Когда и где была создана первая ЭВМ? Как она называлась?
2. Что такое микропроцессор?
3. Когда и кем была создана первая отечественная ЭВМ? Как она называлась?
4. Назовите два основных направления в развитии ЭВМ четвертого поколения.

Практические задания для выполнения на компьютере**Практическое задание № 1**

Создать презентацию вашей школы, состоящую из следующих слайдов (шаблон дизайна и эффекты подобрать самостоятельно, время перехода между слайдами — 3 секунды):

Слайд 1

Разметка слайда — Титульный слайд.

Заголовок: Наша школа.

Подзаголовок: указать номер школы и название населенного пункта.

Слайд 2

Разметка слайда — Маркированный список.

Заголовок: Структура школы.

Список:

- ☐ младшие классы (1—4 классы)
- ☐ средние классы (5—9 классы)
- ☐ старшие классы (10—11 классы)

Добавить к слайду в нижнем правом углу произвольный рисунок.

Слайд 3

Разметка слайда — Только заголовок.

Заголовок: Наши преподаватели.

Добавить к слайду в центре фотографию из файла, указанного преподавателем.

Примечание. Перед началом занятий необходимо подготовить фотографию для слайда 3. В случае отсутствия подобной фотографии, тему слайда можно изменить.

Практическое задание № 2

Создайте презентацию «Животный мир», состоящую из следующих слайдов.

Слайд 1

Заголовок: Животный мир.

Подзаголовок: Фотоальбом.

Слайд 2

Заголовок: Царство животных.

Диаграмма:



Слайд 3

Заголовок: Млекопитающие.

Текст слайда:

Существует около 4500 видов млекопитающих.

Характерные признаки:

- теплокровные
- вскармливают детенышей молоком
- дышат воздухом через легкие

Слайд 4

Заголовок: Волк.

Текст слайда:

Хищное млекопитающее семейства псовых.

- длина тела 1–1,6 м
- обитает в Евразии, Сев. Америке.

Рисунок слайда: volk.jpg

Слайд 5

Заголовок: Рысь.

Текст слайда:

Млекопитающее семейства кошек.

- длина тела — до 109 см
- обитает в лесах Евразии и Сев. Америке

Рисунок слайда: rys.jpg

Рекомендации и требования к презентации:

- шаблон дизайна, разметки слайдов, оформление и эффекты подобрать самостоятельно;
- время перехода между слайдами — 1 секунда;
- обязательно использовать объект WordArt (хотя бы в одном слайде);
- обязательно использовать организационную диаграмму (в слайде 2);
- обязательно использовать эффекты анимации (не менее 3 типов).

Примечание. Перед началом занятия необходимо подготовить файлы volk.jpg и rys.jpg.

Практическое задание № 3

Создать презентацию «Музеи России», состоящую из следующих слайдов.

Слайд 1

Заголовок: Музеи России.

Рисунок слайда: museum.jpg.

Переход к следующему слайду: автоматически через 1 секунду.

Слайд 2

Заголовок: Музеи Москвы и Санкт-Петербурга.

Гиперссылки:

Третьяковская галерея (переход к слайду 3);

Эрмитаж (переход к слайду 5).

Кнопка «Выход».

Переход к следующему слайду: после выбора одной из гиперссылок.

Слайд 3

Заголовок: Васнецов Виктор Михайлович.

Рисунок слайда: [vasnesov.jpg](#).

Переход к предыдущему и следующему слайдам: кнопки перехода.

Слайд 4

Заголовок: Левитан Исаак Ильич.

Рисунок слайда: [levitan.jpg](#).

Переход к предыдущему слайду: кнопка перехода.

Слайд 5

Заголовок: Рембрандт Харменс ван Рейн.

Рисунок слайда: [rembrand.jpg](#).

Переход к следующему слайду и слайду 2: кнопки перехода.

Слайд 6

Заголовок: Рафаэль.

Рисунок слайда: [rafael.jpg](#).

Переход к предыдущему слайду: кнопка перехода.

Рекомендации и требования к презентации:

- шаблон дизайна, разметки слайдов, оформление и эффекты подобрать самостоятельно;
- обязательно использовать объекта WordArt (хотя бы в одном слайде);
- обязательно использовать эффекты анимации (не менее 3 типов);

- использовать одинаковый дизайн для кнопок перехода (размер и цвет).

Примечание. Перед началом занятия необходимо подготовить файлы museum.jpg, vaspesov.jpg, levitan.jpg, rembrand.jpg, rafael.jpg.

Практическое задание № 4

1. Загрузить базу знаний FAMILY. Сформировать цели к базе:

- ☐ Приходится ли Ольга дочерью Нине?
- ☐ Кто мать Тани?
- ☐ Кто дочь Ольги?
- ☐ Кто кому приходится матерью?

Примечание. Перед занятием необходимо подготовить базу знаний FAMILY.PRO.

2. Описать базу знаний, содержащую следующие факты и правила:

- ☐ Петр учится в школе № 2.
- ☐ Ольга учится в школе № 7.
- ☐ Иван учится там же, где и Петр.
- ☐ Елена учится там же, где и Ольга.

Сформулировать на Турбо Прологе вопросы и ответить на них:

- ☐ Кто учится в школе № 2?
- ☐ Учится ли Иван в школе № 7?
- ☐ Кто учится в школе № 7?

3. Описать базу знаний, содержащую факты:

- ☐ Петр — тренер Марии и Дмитрия.
- ☐ Игорь — тренер Елены.
- ☐ Сергей — тренер Олега и Ивана.

Сформулировать правило, с помощью которого можно получить ответы на вопросы:

- ☐ Кто ученики Игоря?
- ☐ Чья ученица Елена?

Сформулировать эти вопросы на Турбо Прологе и ответить на них.

Практическое задание № 5

Даны результаты сдачи экзаменов для группы из пяти учеников:

Фамилия	Алгебра	Геометрия	История
Антонов	5	5	5
Бобров	4	3	2
Вяткин	4	5	5
Кротов	2	3	3
Соснин	4	4	4

- Используя систему Пролог, построить базу знаний о результатах экзаменов, содержащую правила по определению следующих понятий:
 - отличник (человек, у которого по всем предметам «пятерки»);
 - двоечник (есть хотя бы одна «двойка»);
 - математик (по алгебре и геометрии учится на «4» и «5»).
- Получить ответы на следующие запросы:
 - ☐ Является ли Вяткин отличником?
 - ☐ Вывести список всех отличников.
 - ☐ Вывести список двоечников.
 - ☐ Является ли Соснин математиком?
 - ☐ Получить список неуспевающих по истории.
 - ☐ Получить список всех математиков.

Практическое задание № 6

- Написать алгоритм для изображения лестницы из 5 ступеней (высота и ширина ступеней — 1 шаг).
- Написать алгоритм для изображения буквы Р (высота буквы — 6 шагов, ширина — 3 шага):

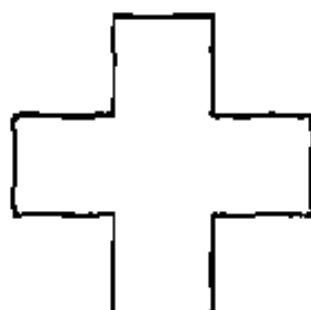


Практическое задание № 7

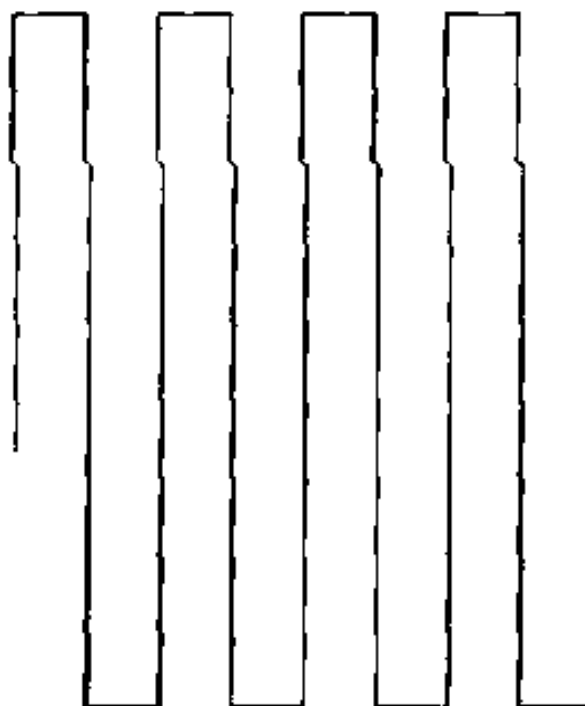
Разработать алгоритм для написания слова «БАОБАБ» (для изображения каждой отдельной буквы использовать процедуру). Высота каждой буквы — 2 шага, ширина — 1 шаг.

Практическое задание № 8

Составить алгоритм перехода из любой позиции экрана в угол рамки и изображения во всех четырёх углах рамки креста размером 3х3

**Практическое задание № 9**

Написать программу, изображающую зигзаг с любого места экрана до границы поля. Высота зигзага — полный размер экрана:



Практическое задание № 10

1. [2]: раздел 4.2, задача № 52.

Практическое задание № 11

1. Написать программу вычисления выражения $A = X^3 + X^2Y + Y^3$ на языке машинных команд (коэффициенты даны в десятичной системе). Провести тестирование программы для $X = 1, Y = 2$ и для $X = 2, Y = 1$.
2. Написать на ЯМК программу, запрашивающую три целых числа и выводящую их кубы.

Практическое задание № 12

1. Написать на языке машинных команд программу вычисления выражения $A = X^3 + 5 \times X^2 \times Y + 10 \times X \times Y^2 + Y^3$. Провести тестирование программы для $X = 1, Y = 2$ и для $X = 2, Y = 1$. Доказать правильность полученных результатов.
2. Написать на языке машинных команд программу, запрашивающую три целых числа X, Y, Z и осуществляющую их перестановку по следующей схеме: $X \rightarrow Z, Z \rightarrow Y, Y \rightarrow X$. Доказать правильность полученного результата.

Тесты

Тест № 1

1. На остановке останавливаются автобусы № 5, № 15 и № 10, маршрутное такси № 40, троллейбусы № 2 и № 7 и несколько трамваев. Появление транспорта каждого маршрута равновероятно. Сообщение о том, что на остановку подошел трамвай № 5, содержит 3 бита информации. Сколько маршрутов трамваев останавливается на данной остановке?

А) 8; В) 2; С) 1; D) 4; Е) 3.

2. Вы подошли к светофору, когда горел желтый свет. После этого загорелся зеленый. Какое количество информации вы при этом получили?

А) 1 бит; С) 0 битов; Е) 2 байта.
В) 2 бита; D) 1 байт;

3. Группа школьников пришла в бассейн, в котором 4 дорожки для плавания. Тренер сообщил, что группа будет плавать на дорожке номер 3. Сколько информации получили школьники из этого сообщения?

А) 1 бит; С) 2 байта; Е) 2 бита.
В) 4 бита; D) 1 байт;

4. В корзине лежат 32 шара. Среди них несколько красных. Сообщение о том, что достали красный шар, несет 3 бита информации. Сколько красных шаров было в корзине?

А) 3; С) 8 Е) 4;
В) 32; D) 16.

5. В корзине лежат 20 шаров. Из них 10 черных и 5 желтых. Какое количество информации несут сообщения о том, что из корзины случайным образом достали соответственно черный и желтый шары?

А) 1 байт; 2 байта; D) 2 бита; 4 бита;
В) 8 битов; 4 бита; Е) 2 байта; 4 байта.
С) 1 бит; 2 бита;

6. В классе 32 человека. За контрольную работу по математике получено 4 пятерки, 16 четверок, 11 троек и 1 двойка. Какое количество информации (в битах) в сообщении о том, что Иванов получил «четверку»?

А) 2; С) 5; Е) 1.
В) 4; D) 8;

7. В лотерее выпало 5 шаров из N возможных и сообщение об этом несет 35 битов информации. Чему равно N ?

- В) совокупность возможных вопросов к БЗ;
С) вопрос к БЗ и ответы на него;
D) описание БЗ;
Е) вопрос к БЗ.
2. Дана база знаний, содержащая сведения о времени начала и окончания работы магазина, продавцах:
магазин(эльдорадо, 8, 20) .
магазин(сатурн, 9, 19) .
магазин(весна, 10, 19) .
продавец(юля, сатурн) .
продавец(сергей, эльдорадо) .
продавец(ваня, весна) .
продавец(катя, сатурн) .
день(X, A, B) : -продавец(X, Y) , магазин(Y, A, B) .
- Каким будет ответ на цель ?день(X, Y, 19)?
- A) X=сатурн Y=9
X=весна Y=10;
B) X=эльдорадо;
C) нет решения;
D) X=юля Y=9
X=ваня Y=10
X=катя Y=9;
Е) X=юля X=ваня X=катя.
3. Сколько решений будет выдано в ответ на цель:
?день(X, Y, Z) ?
A) 16; B) 8; C) 4; D) 3; E) 1.
4. Каким будет ответ на цель к БЗ из задания 2
?день(юля, 11, 20) ?
A) нет решения; C) сатурн; E) юля.
B) нет; D) катя;
5. Каким будет ответ на цель к БЗ из задания 2
?продавец(X, сатурн) ?
A) нет решения;
B) X=юля;
C) X=юля X=катя;

D) X=катя;

E) X=юля X=сергей X=ваня X=катя.

6. БЗ содержит информацию о группе людей и имеющихся у них вещах:

имеет (иван, автомобиль) . имеет (иван, телевизор) .

имеет (иван, видео) .

имеет (петр, холодильник) . имеет (петр, телевизор) .

имеет (николай, пылесос) .

наличные (иван, 10000) . наличные (николай, 15000) . наличные (петр, 5000) .

жена (анна, николай) . жена (мария, иван) .

цена (автомобиль, 17000) . цена (видео, 1000) .

цена (телевизор, 2100) . цена (пылесос, 850) .

Каким будет ответ на цель

? жена (мария, петр) ?

A) нет решения; C) иван; E) мария, иван.

B) нет; D) да;

7. Каким будет ответ на цель к БЗ из задания 6

? имеет (X, телевизор) ; имеет (X, пылесос) ?

A) иван, петр; D) нет решения;

B) николай; E) нет.

C) иван, петр, николай;

8. Каким будет ответ на цель к БЗ из задания 6

? наличные (X, 15000) , имеет (X, автомобиль) ?

A) петр; D) иван, николай;

B) иван; E) нет решения.

C) николай;

9. Для ответа на некоторый вопрос к БЗ из задания 6 было сформулировано правило:

$\text{пр}(X, Y, Z) :- \text{жена}(X, Q), \text{наличные}(Q, Y),$
 $\text{имеет}(Q, Z) .$

Каким будет ответ на цель ? $\text{пр}(X, 15000,$
 $\text{пылесос}) ?$

A) николай; C) нет решения; E) мария.

B) анна; D) петр;

10. В БЗ из задания 6 имеется правило:
может_купить(X, Y) : -наличные(X, Z) ,
цена(Y, Q) , $Q < Z$.
Каким будет ответ на цель
?может_купить(николай, X) , $X \neq$ "телевизор"?
А) нет решения;
В) видео, пылесос;
С) видео, телевизор, пылесос;
D) видео;
Е) пылесос.
11. База знаний из задания 6 дополнена правилом
имеет(X, Y) : - жена(X, Z) , имеет(Z, Y) .
Каким будет ответ на цель ?имеет(X , видео)?
А) иван; С) иван, maria; Е) да.
В) maria; D) нет решения;
12. Каким будет ответ на цель к БЗ из задания 6
?имеет(иван, X) , not (имеет(николай, X)) ?
А) пылесос;
В) автомобиль, телевизор, видео;
С) нет решения;
D) нет;
Е) да.
13. База знаний содержит информацию о членах клуба любителей плавания:
умеет(антонов, кроль) . умеет(антонов, брасс) .
умеет(антонов, баттерфляй) .
умеет(кузнецов, кроль) . умеет(кузнецов, брасс) .
умеет(бобров, брасс) .
Каким будет ответ на цель
? умеет(X , брасс) , умеет(X , кроль) ;
умеет(X , брасс) , умеет(X , баттерфляй) ;
умеет(X , кроль) , умеет(X , баттерфляй)?
А. антонов, кузнецов, антонов, антонов
В. антонов, кузнецов, бобров
С. нет решения
D. кузнецов
Е. нет

14. Сформулировать цель к БЗ из задания 13, позволяющую получить список всех пловцов, владеющих стилем брасс.
- А) ?умеет(брасс, X).
 - В) ?умеет(антонов, кузнецов, бобров, брасс).
 - С) ?умеет(антонов, брасс), умеет(кузнецов, брасс), умеет(бобров, брасс).
 - Д) ?умеет(X, брасс).
 - Е) ?умеет(антонов, брасс); умеет(кузнецов, брасс).
умеет(бобров, брасс).
15. Каким будет ответ на цель к БЗ из задания 13 ?умеет(X, баттерфляй); умеет(X, брасс)?
- А) нет решения;
 - В) антонов, кузнецов;
 - С) антонов;
 - Д) нет;
 - Е) антонов, антонов, кузнецов, бобров.
16. Каким будет ответ на цель к БЗ из задания 13 ?умеет(кузнецов, X), умеет(бобров, X)?
- А) да;
 - В) нет;
 - С) кроль;
 - Д) брасс;
 - Е) кроль, брасс.
17. Сформулировать правило БЗ из задания 13: «Кузнецов владеет теми же видами плавания, что и Антонов, кроме брасса».
- А) что_умеет(кузнецов, X) :- умеет(антонов, X), X<>брасс.
 - В) что_умеет(кузнецов, X) :- умеет(антонов, X); X<>брасс.
 - С) что_умеет(кузнецов, X) :- умеет(X, антонов), X<>брасс.
 - Д) что_умеет(кузнецов, X) :- умеет(антонов, брасс).
 - Е) что_умеет(кузнецов, X) :-
18. Отношения «Мастер и Новичок» задаются в БЗ 3 из задания 13 правилами:

мастер(X) :- умеет(X,басс),умеет(X,кроль),умеет(X,бадтерфляй).
 новичок(X) :- умеет(X,басс),
 not(умеет(X,кроль)),not(умеет(X,бадтерфляй));

умеет(X,кроль),not(умеет(X,басс)),
 not(умеет(X,бадтерфляй));

умеет(X,бадтерфляй),not(умеет(X,басс)),
 not(умеет(X,кроль)).

Каким будет ответ на цель ?мастер(бобров);новичок(бобров)?

А) да; С) нет решения; Е) антонов, бобров.
 В) нет; D) бобров;

19. Известно, что Митя, Сергей, Толя, Костя и Юра пришли в музей до открытия и встали в очередь. Митя пришел позже Сергея, Толя — раньше Кости, Митя — раньше Толи, Юра — позже Кости. Порядок прихода мальчиков в музей можно описать с помощью базы знаний, содержащей следующие факты:

раньше(толя, костя). раньше(митя, толя). раньше(костя, юра). раньше(сергей, митя).

Сформулировать вопрос, позволяющий определить, между какими мальчиками стоял Костя.

А) раньше(A, костя);раньше(костя,В).

В) раньше(A, костя),раньше(костя,А).

С) раньше(A, костя),раньше(костя,В).

D) раньше(A, костя);раньше(костя,А).

Е) раньше(A, костя,В).

20. Сформулировать вопрос к БЗ из задания 19, позволяющий определить, в каком порядке мальчики стояли в очереди.

А) раньше(X, Y);раньше(Y, Z);раньше(Z, N);
 раньше(N, Q);

В) раньше(X, Y),раньше(Y, Z),раньше(Z, N),
 раньше(N, Q);

С) раньше(X, Y, Z, N, Q), $X < Y, Y < Z, Z < N, N < Q$;

- D) раньше (X, Y, Z, N, Q) ;
 E) раньше (X, A) , раньше (Y, B) , раньше (Z, C) ,
 раньше (N, D) , раньше (Q, E) .

Ответы на тест № 2

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	Е	Д	С	В	С	В	С	Е	В	В
№ вопроса	11	12	13	14	15	16	17	18	19	20
Ответ	С	В	А	Д	Е	Д	А	А	С	В

Тест № 3

- В системах автоматического управления техническими устройствами аналого-цифровой преобразователь является частью:
 - линии обратной связи;
 - объекта управления;
 - линии прямой связи;
 - линии прямой и обратной связи;
 - объекта управления и линии прямой связи.
- В системах управления по линии прямой связи передаются:
 - команды управления;
 - информация о состоянии объекта управления;
 - информация о состоянии управляющей системы;
 - команды управления и информация об объекте управления;
 - команды управления и информация об управляющей системе.
- Без учета обратной связи алгоритм управления:
 - может содержать только циклы и ветвления;
 - может содержать циклы;
 - может быть только циклическим;
 - может быть только линейным;
 - может содержать ветвления.
- Понятность алгоритма обозначает то, что он должен быть записан с помощью:

- А) команд, понятных создателю алгоритма;
 - В) команд из системы команд исполнителя;
 - С) команд, понятных пользователю алгоритма;
 - Д) команд, понятных для компьютера;
 - Е) операторов языков программирования.
5. Цифро-аналоговый преобразователь осуществляет преобразование:
- А) аналогового сигнала в двоичный код;
 - В) двоичного кода в аналоговый сигнал;
 - С) управляющих команд в двоичный код;
 - Д) числовой информации в двоичный код;
 - Е) двоичного кода в числовую информацию.
6. Структура алгоритма, все команды которого выполняются по очереди только один раз, называется:
- А) циклической;
 - В) линейной;
 - С) ветвящейся;
 - Д) основной;
 - Е) вспомогательной.
7. Алгоритмы, которые решают некоторую подзадачу главной задачи и, как правило, выполняются многократно, называются:
- А) циклическими;
 - В) вспомогательными;
 - С) линейными;
 - Д) основными;
 - Е) ветвящимися.
8. От любого исполнителя не требуется:
- А) умения точно выполнять команды;
 - В) соблюдения последовательности действий алгоритма;
 - С) понимания смысла алгоритма;
 - Д) формального выполнения команд алгоритма;
 - Е) выполнения вспомогательных алгоритмов.

Задания 10—14 посвящены работе с исполнителем «Кенгуренок».

9. Система команд графического исполнителя «Кенгуренок РУ»:
- ШГ (ШАГ) — перемещение на шаг вперед с рисованием линии.

ПР (ПРЫЖОК) — перемещение на шаг вперед без рисования.

ПТ (ПОВОРОТ) — поворот на 90° против часовой стрелки

СД (СДЕЛАЙ) — выполнение вспомогательного алгоритма.

ПРОЦЕДУРА — заголовок вспомогательного алгоритма.

КН ПРОЦ — конец описания тела вспомогательного алгоритма.

ПОКА <УСЛ> ПОВТОРЯТЬ — заголовок цикла с предусловием.

КН ЦИКЛА — конец цикла.

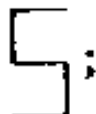
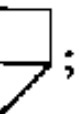
ЕСЛИ <УСЛ> ТО — команда ветвления.

КН ВЕТВЛЕНИЯ — конец ветвления.

<УСЛ> — ВПЕРЕДИ НЕ КРАЙ (ВПЕРЕДИ КРАЙ).

При записи программы одна команда от другой отделяется пробелом. Положение Кенгуренка характеризуется направлениями: ВВЕРХ, ВНИЗ, ВПРАВО, ВЛЕВО. Перед работой РУ находится в ЛЕВОМ ВЕРХНЕМ УГЛУ, направление — ВНИЗ.

Какую из представленных фигур Ру в принципе не сможет начертить?

А) ; В) ; С) ; D) ; Е) .

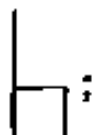
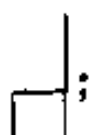
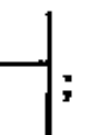
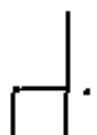
10. Что будет изображено на экране после выполнения Кенгуренком следующей программы?

СД Символ

ПРОЦЕДУРА Символ

СД Линия ПТ ПР ПР ПТ СД Линия ПТ ПР ПТ ШГ
ПТ ШГ КН ПРОЦ

ПРОЦЕДУРА Линия ШГ ШГ КН ПРОЦ

А) ; В) ; С) ; D) ; Е) .

11. Программа формирует изображение четырех квадратов со стороной 1 шаг на расстоянии 1 шаг по горизонтали друг от друга:

СД Два ПТ ПТ ПР ПР ПТ ПТ ПТ ПТ СД Два

Выбрать описание тела процедуры Два, в которой используется процедура Квадрат:

ПРОЦЕДУРА Квадрат ШГ ПТ ШГ ПТ ШГ ПТ ШГ
КН ПРОЦ

А) СД Квадрат ПТ ПТ ПР ПР ПТ ПР ПТ СД Квадрат

В) СД Квадрат ПТ ПТ ПР ПР ПТ ПТ ПТ СД Квадрат

С) СД Квадрат ПТ ПТ ШГ ШГ ПТ ПР ПТ СД Квадрат

Д) СД Квадрат ПТ ПТ ПР ШГ ПТ ПР ПТ СД Квадрат

Е) СД Квадрат ПТ ПР ШГ ПР ПТ ПР ПТ СД Квадрат

12. Кенгуренок рисует горизонтальную линию, отстоящую от правой и левой границ на два шага. В программу вместо многоточия вставить подходящий фрагмент:

ПТ ПР ПР ПОКА ВПЕРЕДИ НЕ КРАЙ ПОВТОРЯТЬ
ШГ ПР ПР

ЕСЛИ ВПЕРЕДИ НЕ КРАЙ ТО ... КН
ВЕТВЛЕНИЯ КН ЦИКЛА

А) ПТ ШГ ПР ПР ПТ ПТ

В) ПТ ПТ ПР ПР ПТ

С) ПТ ПТ ПР ПР

Д) ПТ ПР ПР ПТ ПТ

Е) ПТ ПР ПР ПТ ПТ

13. Что будет изображено на экране после выполнения программы?

СД РИС СД РИС ПРОЦЕДУРА РИС ПТ ШГ ПТ ПТ
ПТ ШГ ШГ КН ПРОЦ

А) Прямоугольник с горизонтальной стороной длиной 2 шага и вертикальной стороной длиной 1 шаг;

В) прямоугольник с горизонтальной стороной длиной 1 шаг и вертикальной стороной длиной 2 шага;

- С) квадрат со стороной длиной 2 шага;
- Д) две ступени лестницы высотой длиной 2 шага и шириной 1 шаг;
- Е) две ступени лестницы с высотой и шириной 2 шага.

14. Что сделает Кенгуренок в результате выполнения программы?

```
ПОКА ВПЕРЕДИ НЕ КРАЙ ПОВТОРЯТЬ  
ШГ ЕСЛИ ВПЕРЕДИ КРАЙ ТО ПТ ПТ КН  
ВЕТВЛЕНИЯ КН ЦИКЛА
```

- А) Начертит рамку вдоль границ и остановится;
- В) будет чертить рамку вдоль границ, не останавливаясь;
- С) расчертит все поле вертикальными линиями и остановится;
- Д) будет чертить вертикальную линию вдоль левой границы, не останавливаясь;
- Е) начертит вертикальную линию вдоль левой границы и остановится.

Задания 15—20 посвящены работе с исполнителем «Линейка».

15. Имеется линейка с делениями длиной 14 см, вдоль которой прыгает «умный» мячик. Если мы хотим, чтобы мячик двигался вправо, надо задать число знаком «+», а влево — число со знаком «-». Например: +2 означает продвинуться на 2 см вправо, а -3, на 3 см влево. Над каждым числом на линейке можно расставить любые символы, тогда мячик, передвигаясь, может собирать слова.

Команда «!» означает взять символ, на который указывает мячик.

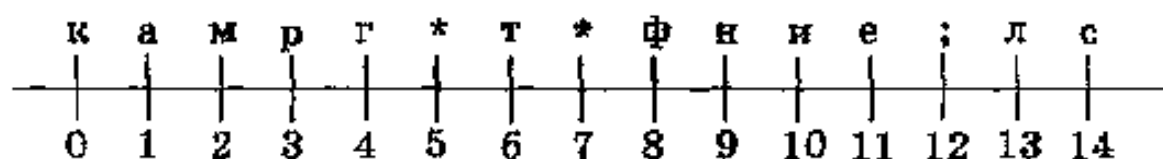
Команда ?символ (действия1, действия2) означает проверку символа. Например: ?а(+3,!) означает, что, если символ, на который указывает мячик — «а», то продвинуться на 3 см вправо, иначе — оставаться на месте и взять символ.

Команда ПОКА НЕ символ(действия) означает выполнение действий до тех пор, пока текущая позиция мячика не будет совпадать с указанным в команде символом.

Мячик может исполнять вспомогательные алгоритмы. ПРОЦ <имя процедуры> — это заголовок вспомогательного алгоритма. КН ПРОЦ — конец описания тела вспомогательного алгоритма. Обращение к вспомогательному алгоритму происходит путем указания его имени в основной программе.

Команда «.» означает конец собирания слова.

Рассмотрим следующий вариант Линейки. Исходное положение мячика — 4 см. Знак «*» («звездочка») означает невидимый нам символ. Известно, что, по крайней мере под одной из «звездочек» находится буква «о».



Какой из предложенных алгоритмов не будет понятен исполнителю «Линейка»?

А) $-2-1+2+2$.

В) ПОКА НЕ а(-1).

С) $!+1?o(!,+2!) ПОКА НЕ л(+1)!$.

Д) ПОКА НЕ с(+1?) $!-3!+2! ПОКА НЕ o(-1?)!$

Е) $!+1?o(!,+2!) ПОКА НЕ л(+1)!!$.

16. Какое слово соберет «мячик» после исполнения алгоритма

$-4! ПОКА НЕ o(+1)! ПОКА НЕ р(-1)!+3!$.

А) крот; В) кот; С) корт; Д) кт; Е) крт.

17. Какое слово соберет «мячик» после исполнения следующего алгоритма?

$-4! АЛГ ПОКА НЕ к(-1)! АЛГ ПОКА НЕ с(+1)!,$

ПРОЦ АЛГ $+5? o(!,+2!) КН ПРОЦ$

А) кок; В) кокос; С) кокс; Д) скок; Е) сок.

18. Выбрать алгоритм, с помощью которого мячик соберет слово «алгоритм».

А) $-3!+12!-9!+1?o(!-2!,+2!-4!+7!-4!-4!)$.

В) $-3!+12!-9!+1?o(!-2!+7!-4!-4!,+3-4!)+7!-4!-4!$.

С) $-3!+12!-9!+!1?o(-2!,+3!-4!)+7!-4!-4!$.

Д) $-3!+12!-9!+1?o(!-2!,+2!-4!)+7!-4!-4!$.

Е) $-3!+12!-9!+1?o(!-2!,+3!-4!)+7!-4!-4!$.

19. Указать начальное положение мячика (в см), если после исполнения алгоритма

$-3?o(!-1!,+2!-3)+5!+1$.

будет собрано слово «огни»:

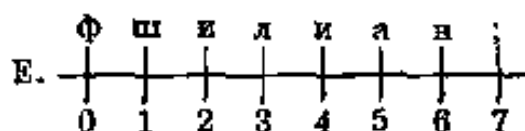
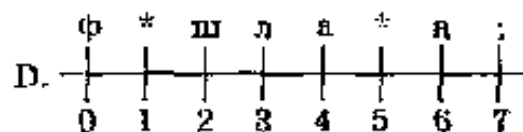
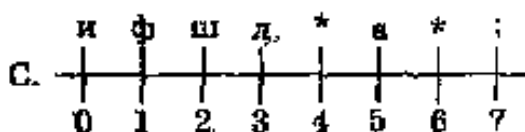
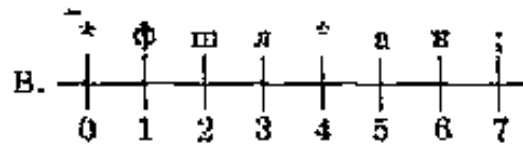
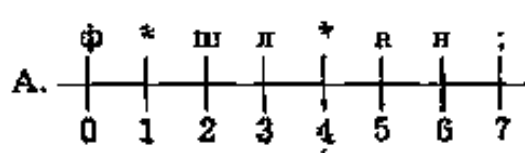
А) 4; В) 10; С) 8; Д) 9; Е) 2.

20. После выполнения алгоритма

$-5 \text{ АЛГ } -2? \text{ и } (!-2!,-3!+1!)+5!-7 \text{ АЛГ } -1!-2!$

$\text{ПРОЦ АЛГ } !+1 ? \text{ и } (!+5!,+3+2!) \text{ КН ПРОЦ}$

будет собрано: «финиш;финал». Выбрать соответствующий вариант Линейки, если начальное положение мячика — 5 см и под одной из «звездочек» находится буква «а», а под другой — «и».



Ответы на тест № 3

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ	А	А	Д	В	В	В	В	С	Д	В
№ вопроса	11	12	13	14	15	16	17	18	19	20
Ответ	В	Д	Д	Д	Д	С	В	Д	С	А

Тест № 4

1. Группа школьников пришла в бассейн, в котором 4 дорожки для плавания. Тренер сообщил, что группа будет плавать на дорожке номер 3. Сколько информации получили школьники из этого сообщения?
A) 1 бит; C) 2 байта; E) 2 бита.
B) 4 бита; D) 1 байт;
2. В корзине лежат 32 шара. Среди них несколько красных. Сообщение о том, что достали красный шар, несет 3 бита информации. Сколько красных шаров было в корзине?
A) 3; C) 8; E) 4.
B) 32; D) 16;
3. БЗ содержит информацию о группе людей и имеющихся у них вещах:
имеет (иван, автомобиль) .
имеет (иван, телевизор) .
имеет (иван, видео) .
имеет (петр, холодильник) .
имеет (петр, телевизор) .
имеет (николай, пылесос) .
наличные (иван, 10000) .
наличные (николай, 15000) .
наличные (петр, 5000) .
жена (анна, николай) . жена (мария, иван) .
цена (автомобиль, 17000) . цена (видео, 1000) .
цена (телевизор, 2100) . цена (пылесос, 850) .
Каким будет ответ на цель ? жена (мария, петр) ?
A) нет решения; C) иван; E) мария, иван.
B) нет; D) да;
4. Каким будет ответ на цель к БЗ из задания 3 ? имеет (X, телевизор) ; имеет (X, пылесос) ?
A) иван, петр; D) нет решения;
B) николай; E) нет.
C) иван, петр, николай;

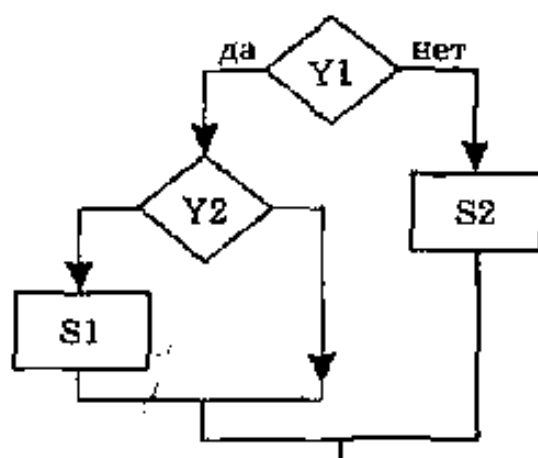
5. Каким будет ответ на цель к БЗ из задания 6
?наличные (X, 15000) , имеет (X, автомобиль) ?
А) петр; С) николай; Е) нет решения.
В) иван; D) иван, николай;
6. Для ответа на некоторый вопрос к БЗ из задания 6
было сформулировано правило:
пр (X, Y, Z) : - жена (X, Q) , наличные (Q, Y) , имеет (Q, Z) .
Каким будет ответ на цель
? пр (X, 15000, пылесос) ?
А) николай; С) нет решения; Е) мария.
В) анна; D) петр;
7. В БЗ из задания 6 имеется правило:
может_купить (X, Y) : -наличные (X, Z) ,
цена (Y, Q) , $Q < Z$.
Каким будет ответ на цель
?может_купить (николай, X) , Xтелевизор ?
А) нет решения;
В) видео, пылесос;
С) видео, телевизор, пылесос;
D) видео;
Е) пылесос.
8. Алгоритмы, которые решают некоторую подзадачу
главной задачи и, как правило, выполняются многократно, называются:
А) циклическими; D) основными;
В) вспомогательными; Е) ветвящимися.
С) линейными;

Задания 9—11 посвящены работе с исполнителем Кузнечик.

9. Имеется исполнитель Кузнечик, который «живет»
на числовой оси.
Система команд Кузнечика:
ВПЕРЕД N — прыжок по числовой оси на N единиц вперед.
НАЗАД N — прыжок по числовой оси на N единиц назад.

- Стартовав из точки (-217) и выполнив 58 команд ВПЕРЕД 3, Кузнечик попал в точку
А) (-43) ; В) 43; С) 44; D) 42; E) (-44) .
10. Кузнечик выполнил программу из 50 команд, в которой команд НАЗАД 2 на 10 больше, чем команд ВПЕРЕД 3 (других команд нет). На сколько единиц и в какую сторону сместился Кузнечик?
А) Такой программы не может быть;
В) на 20 единиц вперед;
С) на 30 единиц вперед;
D) вернулся в исходное положение;
E) на 30 единиц назад.
11. Стартовав из точки 0 и выполнив некоторую программу, Кузнечик побывал последовательно в точках: 0; 3; 1; 4; 2; 0; -2. В каких точках побывает Кузнечик, выполняя эту же программу, стартовав из точки -5?
А) 5; 8; 6; 9; 7; 5; 3;
В) 5; 8; 6; 9; 7; 5; -3;
С) -5; -2; -4; -1; -3; -5; -7;
D) -5; -2; -4; -1; -3; -5; -3;
E) -6; -3; -5; -2; -4; -6; -8.
12. Шестнадцатеричное число 16A перевести в двоичную систему счисления:
А) 11101010; D) 100001101010;
В) 101101010; E) 1001101010.
С) 101101010000;
13. Шестнадцатеричное число AB перевести в восьмеричную систему счисления:
А) 523; В) 526; С) 253; D) 1011; E) 1763.
14. Двоичное число 10111101 перевести в восьмеричную систему счисления:
А) 571; В) 572; С) 1075; D) 275; E) 5701.
15. Двоичное число 1011011101 перевести в шестнадцатеричную систему счисления:
А) 1335; В) B71; С) 5561; D) 2CC; E) 2DD.

16. Выполнить сложение двоичных чисел 110011 и 1111:
A) 1000110; C) 111122; E) 1001110.
B) 1000010; D) 1001111;
17. Восьмеричное число 27,76 перевести в шестнадцатеричную систему счисления:
A) 17,F8; C) 17,F2; E) 5C,F2.
B) 53,72; D) 17,76;
18. Выбрать фрагмент программы на алгоритмическом языке, соответствующий блок-схеме



- A) ЕСЛИ Y1 ТО ЕСЛИ Y2 ТО S1 КОНЕЦ ВЕТВЛЕНИЯ
ИНАЧЕ S2 КОНЕЦ ВЕТВЛЕНИЯ
- B) ЕСЛИ Y1 ТО ЕСЛИ Y2 ТО S1 КОНЕЦ ВЕТВЛЕНИЯ
КОНЕЦ ВЕТВЛЕНИЯ S2
- C) ЕСЛИ Y1 ТО ЕСЛИ Y2 ТО S1 ИНАЧЕ S2 КОНЕЦ
ВЕТВЛЕНИЯ. КОНЕЦ ВЕТВЛЕНИЯ
- D) ЕСЛИ Y1 ТО S1 ИНАЧЕ ЕСЛИ Y2 ТО S1 КОНЕЦ
ВЕТВЛЕНИЯ КОНЕЦ ВЕТВЛЕНИЯ
- E) ЕСЛИ Y1 ТО ПОВТОРЯТЬ S2 ЕСЛИ Y2 ТО S1
КОНЕЦ ВЕТВЛЕНИЯ КОНЕЦ ЦИКЛА
19. Для блок-схемы из задания 18 выбрать условие,
при котором будет выполняться команда S1.
A) (Y1=истина) И (Y2=истина)
B) Y1=ложь
C) (Y1=истина) ИЛИ (Y2=ложь)
D) (Y1=истина) И (Y2=ложь)
E) Y1=истина

Итоговая система знаний и умений учащихся по базовому курсу информатики

В концепции Федеральных компонент Государственного образовательного стандарта о требованиях к минимально необходимому уровню подготовки учащихся сказано следующее: *«требования должны отражать всю совокупность базового содержания и заданы в предметно-деятельностной форме; ... объем требований не может быть тождественным объему содержания образования, предъявляемого в процессе обучения»*.

Рассматривая информатику как фундаментальную научную область, подходя к школьной дисциплине «Информатика» как общеобразовательному предмету, целесообразно говорить о понятийно-деятельностном подходе к определению уровня подготовки учащихся. Грамотность — это не просто набор навыков практических действий (что и как делать), но и понимание смысла этих действий (почему делаем именно так). Концепция знаний, умений и навыков как раз и основана на взаимосвязи этих двух сторон результата обучения.

Далее приводится перечень знаний и умений учащихся по базовому курсу, ориентированный на два уровня подготовки: основной (минимальный) уровень и углубленный уровень. Основной уровень знаний и умений учащихся может быть обеспечен в результате обучения по первому варианту программы: 2 года по 1 часу в неделю. Углубленный уровень — по второму и третьему вариантам программы обучения.

1. Основной уровень знаний и умений учащихся по итогам освоения базового курса информатики

По линии «Информация и информационные процессы»

Учащиеся должны знать:

- ☐ определение информации в соответствии с содержательным подходом и кибернетическим (алфавитным) подходом;
- ☐ что такое информационные процессы;
- ☐ какие существуют носители информации;
- ☐ как определяется единица измерения информации — бит;
- ☐ что такое байт, килобайт, мегабайт, гигабайт;
- ☐ функции языка как способа представления информации; что такое естественные и формальные языки;
- ☐ что такое система счисления; в чем различие между позиционными и непозиционными системами счисления;
- ☐ что такое логическая величина, логическое выражение;
- ☐ что такое логические операции, как они выполняются.
- ☐ правила записи и вычисления логических выражений.

Учащиеся должны уметь:

- ☐ приводить примеры информации и информационных процессов из области человеческой деятельности, живой природы и техники;
- ☐ определять в конкретном процессе передачи информации источник, приемник, канал;
- ☐ приводить примеры информативных и неинформативных сообщений;
- ☐ приводить примеры сообщений, несущих 1 бит информации;

- измерять информационный объем текста в байтах (при использовании компьютерного алфавита);
- пересчитывать количество информации в различных единицах (битах, байтах, Кб, Мб, Гб);
- переводить целые числа из десятичной системы счисления в другие системы и обратно;
- выполнять простейшие арифметические операции с двоичными числами;
- определять истинность высказываний (логических выражений);
- записывать логические выражения с использованием основных логических операций: И, ИЛИ, НЕ.
- использовать логические выражения при работе с базами данных, электронными таблицами, языками программирования.

По линии «Компьютер (ЭВМ)»

Учащиеся должны знать:

- правила техники безопасности при работе на компьютере;
- состав основных устройств компьютера, их назначение и информационное взаимодействие;
- основные характеристики компьютера в целом и его узлов (различных накопителей, устройств ввода и вывода информации);
- структуру внутренней памяти компьютера (биты, байты); понятие адреса памяти;
- типы и свойства устройств внешней памяти;
- типы и назначение устройств ввода/вывода;
- сущность программного управления работой компьютера;
- архитектуру персонального компьютера;
- принципы организации информации на дисках: что такое файл, каталог (папка), файловая структура;
- назначение программного обеспечения и его состав;
- основные этапы развития информационно-вычислительной техники, программного обеспечения ЭВМ и информационных технологий.

Учащиеся должны уметь:

- ☐ включать и выключать компьютер;
- ☐ пользоваться клавиатурой;
- ☐ вставлять дискеты в накопители;
- ☐ ориентироваться в типовом интерфейсе: пользоваться меню, обращаться за справкой, работать с окнами;
- ☐ инициализировать выполнение программ из программных файлов;
- ☐ просматривать на экране директорию диска;
- ☐ выполнять основные операции с файлами и каталогами (папками): копирование, перемещение, удаление, переименование, поиск.

По линии «Компьютерные информационные технологии»

Учащиеся должны знать:

- ☐ способы представления символьной информации в памяти ЭВМ (таблицы кодировки, текстовые файлы);
- ☐ назначение текстовых редакторов (текстовых процессоров);
- ☐ основные режимы работы текстовых редакторов (ввод/редактирование, печать, орфографический контроль, поиск и замена, работа с файлами);
- ☐ способы представления изображений в памяти ЭВМ; понятия о пикселе, растре, кодировке цвета, видеопамяти;
- ☐ какие существуют области применения компьютерной графики;
- ☐ назначение графических редакторов;
- ☐ назначение основных компонентов среды графического редактора: рабочего поля, меню инструментов, графических примитивов, палитры, Ножниц, Ластика и пр;
- ☐ что такое компьютерная сеть; в чем различие между локальными и глобальными сетями;

- назначение основных технических и программных средств функционирования сетей: каналов связи, модемов, серверов, клиентов, протоколов;
- назначение основных видов услуг глобальных сетей: электронной почты, телеконференций, распределенных баз данных и др;
- что такое Интернет; какие возможности предоставляет пользователю Всемирная паутина — WWW;
- что такое база данных, СУБД, информационная система;
- что такое реляционная база данных, ее элементы (записи, поля, ключи); типы и форматы полей;
- структуру команд поиска и сортировки информации в базах данных;
- что такое электронная таблица и табличный процессор;
- основные информационные единицы электронной таблицы: ячейки, строки, столбцы, блоки и способы их идентификации;
- какие типы данных заносятся в электронную таблицу; как табличный процессор работает с формулами;
- основные функции (математические, статистические), используемые при записи формул в ЭТ;
- графические возможности табличного процессора;

Учащиеся должны уметь:

- набирать и редактировать текст в одном из текстовых редакторов;
- выполнять основные операции над текстом, допускаемые этим редактором;
- сохранять текст на диске, загружать его с диска, выводить на печать;
- строить несложные изображения с помощью одного из графических редакторов;
- сохранять рисунки на диске и загружать с диска; выводить на печать;
- осуществлять обмен информацией с файл-сервером локальной сети;

- открывать готовую БД в одной из СУБД реляционного типа;
- организовывать поиск информации в БД;
- редактировать содержимое полей БД;
- сортировать записи в БД по ключу;
- добавлять и удалять записи в БД;
- открывать готовую электронную таблицу в одном из табличных процессоров;
- редактировать содержимое ячеек; осуществлять расчеты по готовой электронной таблице;
- выполнять основные операции манипулирования с фрагментами ЭТ: копирование, удаление, вставку, сортировку;
- получать диаграммы с помощью графических средств табличного процессора;
- создавать электронную таблицу для несложных расчетов.

По линии «Процессы управления (информационные основы)»

Учащиеся должны знать:

- что такое наука «Кибернетика»; предмет и задачи этой науки;
- сущность кибернетической схемы управления с обратной связью; назначение прямой и обратной связи в этой схеме;
- что такое алгоритм; какова роль алгоритма в системах управления;
- в чем состоят основные свойства алгоритма;
- способы записи алгоритмов: блок-схемы, учебный алгоритмический язык;
- основные алгоритмические конструкции: следование, ветвление, цикл; структуры алгоритмов;
- назначение вспомогательных алгоритмов; технологии построения сложных алгоритмов: метод последовательной детализации и сборочный (библиотечный) метод.

- основные свойства величин в алгоритмах обработки информации: что такое имя, тип, значение величины; смысл присваивания;
- назначение языков программирования.

Учащиеся должны уметь:

- при анализе простых ситуаций управления определять механизм прямой и обратной связи;
- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- выполнить трассировку алгоритма для известного исполнителя;
- составлять несложные линейные, ветвящиеся и циклические алгоритмы управления одним из учебных исполнителей;
- выделять подзадачи; определять и использовать вспомогательные алгоритмы.

По линии «Информационное моделирование»

Учащиеся должны знать:

- что такое модель; в чем разница между натурной и информационной моделью;
- какие существуют формы представления информационных моделей (графические, табличные, вербальные, математические);
- что такое реляционная модель данных; основные элементы реляционной модели: запись, поле, ключ записи.

Учащиеся должны уметь:

- приводить примеры натурных и информационных моделей;
- проводить в несложных случаях системный анализ объекта (формализацию) с целью построения его информационной модели;
- ориентироваться в таблично организованной информации;
- описывать объект (процесс) в табличной форме для простых случаев;

2. Дополнительные знания и умения учащихся соответствующие углубленному уровню освоения базового курса информатики

По линии «Информация и информационные процессы»

Учащиеся должны знать:

- ☐ связь между количеством информации в сообщении о некотором событии и вероятностью этого события (в приближении равной вероятности и в общем случае);
- ☐ схему К. Шеннона процесса передачи информации по техническим каналам связи; смысл и назначение ее отдельных элементов.

Учащиеся должны уметь:

- ☐ вычислять количество информации в сообщении о событии с известной вероятностью (в приближении равной вероятности и в общем случае);
- ☐ осуществлять перевод целых и дробных десятичных чисел в другие позиционные системы счисления и обратный перевод;
- ☐ переходить от записи двоичной информации к восьмеричной и шестнадцатеричной формам и осуществлять обратный переход.

По линии «Компьютер (ЭВМ)»

Учащиеся должны знать:

- ☐ принципы архитектуры ЭВМ Джона фон Неймана;
- ☐ структуру машинной команды;
- ☐ состав процессора и назначение входящих в него элементов (арифметико-логического устройства, устройства управления, регистров);
- ☐ как процессор выполняет программу (цикл работы процессора);

- состав и функции операционной системы;

Учащиеся должны уметь:

- составлять простую линейную программу на языке машинных команд одного из учебных компьютеров («УК Нейман», «Кроха», «Малютка» и др.);
- работать с сервисными программами: архиваторами, антивирусными программами и др.;
- с помощью системных средств управлять диалоговой средой операционной системы (оболочкой NC для MS DOS, Рабочим столом для Windows).

По линии «Компьютерные информационные технологии»

Учащиеся должны знать:

- в чем различие между растровым и векторным способами представления изображения в компьютере;
- что такое мультимедиа;
- что такое численный эксперимент и как его организовать в электронных таблицах;
- в чем состоит задача проектирования базы данных; что такое нормализация данных;
- назначение основных средств Интернета: Web-сервера, Web-страницы, гиперссылок, программы-браузера, поисковой программы.

Учащиеся должны уметь:

- создавать на компьютере документы, совмещающие объекты разного типа: тексты, таблицы, рисунки и др. (на примерах школьного учебного материала);
- работать с мультимедийными обучающими программами;
- производить на электронных таблицах несложные расчеты учебно-исследовательского характера (на примерах школьного учебного материала);
- проектировать реляционную базу данных для простых информационных систем (на примерах

школьного учебного материала или организационной информации из деятельности школы);

- осуществлять просмотр и поиск информации в Интернете с помощью браузеров и поисковых программ (на примерах материала учебного и развивающего характера).

По линии «Процессы управления (информационные основы)»

Учащиеся должны знать:

- в чем различие между языками программирования высокого уровня и машинно-ориентированными языками;
- правила представления данных на одном из языков программирования высокого уровня (например, на Паскале);
- правила записи основных операторов: ввода, вывода, присваивания, цикла, ветвления;
- правила записи программы;
- что такое трансляция;
- назначение систем программирования;
- содержание этапов разработки программы: алгоритмизация — кодирование — отладка — тестирование.

Учащиеся должны уметь:

- составлять несложные программы решения вычислительных задач с целыми числами;
- программировать простой диалог;
- работать в среде одной из систем программирования (например, Турбо Паскаль);
- осуществлять отладку и тестирование программы.

По линии «Информационное моделирование»

Учащиеся должны знать:

- что такое система, системный анализ, системный подход;
- что такое граф, элементы графа;

- что такое иерархическая система и дерево;
- какие проблемы решает раздел информатики «Искусственный интеллект»;
- что такое модель знаний, база знаний;
- состав базы знаний на Прологе;
- как в Прологе представляются факты и правила;
- как в Прологе формулируются запросы (цели).

Учащиеся должны уметь:

- проводить вычислительный эксперимент над простейшей математической моделью;
- ориентироваться в информационных моделях на языке графов;
- описывать несложную иерархическую систему в виде дерева;
- строить базу знаний на Прологе для простой предметной области (типа родственных связей);
- формулировать на Прологе запросы к данной базе знаний;
- работать на компьютере в среде системы программирования Пролог.

3. Базовый курс в вопросах и ответах

В этом разделе в табличной форме приведена система знаний (основных понятий, правил, определений), которая заключается в содержании базового курса информатики. Материал разделен по темам, в соответствии со структурой учебника. Первая графа таблицы содержит понятия, которые ученики должны уметь раскрывать, вопросы, на которые они должны уметь отвечать. Во второй графе в краткой форме приводятся определения понятий, ответы на вопросы, требования к умениям учащихся, связанным с данным понятием или вопросом. Данная таблица может быть использована учителем для контроля знаний учащихся, а также учениками для подготовки к контрольным мероприятиям: тестам, зачетам, экзаменам.

Глава 1. Человек и информация

Информация и знания	
1. Чем является информация для человека?	Информация для человека — это знания
2. Декларативные и процедурные знания	1) «Я знаю, что...» 2) «Я знаю, как...» Приводить примеры.
3. Три типа информационных процессов	Хранение, передача, обработка. Приводить примеры
4. Роль органов чувств в процессе восприятия информации	Информация к человеку поступает через все органы чувств: слух, зрение, вкус, обоняние, осязание (внутренние информационные каналы человека)
5. Языки естественные и формальные: символьная форма представления информации	Примеры: естественные языки — русский, английский и др.; формальные языки — языки математики, химии, музыки, информатики, логики, программирования и др.

6. Что такое образная информация?	Информация, воспринимаемая человеком и не сводимая к символической форме: вкусы, запахи, картины природы и др.
Содержательный подход к измерению информации	
1. От чего зависит информативность сообщения, принимаемого человеком?	Сообщение информативно, если оно содержит новые и понятные сведения для принимающего его человека
2. Единица измерения информации	1 бит — количество информации в сообщении, уменьшающем неопределенность знаний человека в 2 раза
3. Количество информации в сообщении об одном из N равновероятных событий	Находится из решения показательного уравнения $2^I = N$.
Алфавитный подход к измерению информации	
1. Что такое алфавит, мощность алфавита?	Алфавит — множество символов, используемых для представления информации в некотором языке. Мощность алфавита — число символов в алфавите
2. Что такое информационный вес символа в алфавите?	Количество информации, которое несет один символ в тексте. Находится из решения уравнения: $2^I = N$, где N — мощность алфавита
3. Как измерить информационный объем текста с алфавитной точки зрения?	$I = K \cdot i$ битов, где K — число символов в тексте, i — информационный вес символа в данном алфавите (в равновероятном приближении)
4. Что такое байт, Кбайт, Мбайт, Гбайт?	1 байт = 8 битов; 1 Кбайт = 1024 байтов; 1 Мбайт = 1024 Кбайт и т.д.

5. Скорость информационного потока и пропускная способность канала	Скорость информационного потока измеряется в бит/с. Пропускная способность канала — максимальная скорость информационного потока
Системы счисления	
1. Определение системы счисления	Способ представления чисел и соответствующие ему правила действия над числами
2. Непозиционные системы счисления	Значение, которое несет цифра в записи числа, не зависит от ее позиции в числе (пример — римские цифры)
3. Позиционные системы счисления	Значение, которое несет цифра в записи числа, зависит от ее позиции в числе (пример — арабские цифры)
4. Алфавит позиционной с. с.; основание позиционной с. с	Алфавит — множество цифр; основание — количество цифр в алфавите (мощность алфавита)
5. Развернутая форма представления чисел в позиционной с. с.	$a_n a_{n-1} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-k} = a_n q^n + a_{n-1} q^{n-1} + \dots + a_2 q^2 + a_1 q^1 + a_0 + a_{-1} q^{-1} + \dots + a_{-k} q^{-k}$
6. Перевод чисел из одной с. с. в другую	Перевод $n \rightarrow 10$ — через развернутую форму числа; перевод $10 \rightarrow n$ — через последовательные деления на основание с. с. (целые числа)

Глава 2. Первое знакомство с компьютером

Архитектура ЭВМ	
1. Что такое компьютер (ЭВМ)?	Универсальное программно-управляемое автоматическое устройство для работы с информацией
2. С какой информацией (какими данными) работает компьютер?	Компьютер работает с числовой, символьной (текстовой), графической (видео), звуковой информацией
3. Из каких устройств состоит компьютер?	Основные устройства ЭВМ: память, процессор, устройства ввода/вывода. Память делится на внутреннюю и внешнюю. Знать схему информационного взаимодействия устройств (схема Неймана)
4. Принцип программного управления	Работа компьютера происходит автоматически по программе, хранящейся во внутренней памяти (ОЗУ). Там же хранятся оперативные данные
5. Организация информации во внутренней памяти	Принцип дискретности: память состоит из битов; в одном бите хранится 0 или 1; любая информация имеет двоичную форму. Принцип адресуемости: 8 подряд расположенных битов образуют байт памяти. Порядковый номер байта называется адресом. Информация заносится и извлекается из памяти по адресам
6. Организация информации во внешней памяти	Информация во внешней памяти имеет файловую организацию. Файл — поименованная однотипная информация, хранящаяся в устройстве внешней памяти

7. Устройство персонального компьютера	ПК имеет магистральную архитектуру. Основным элементом ПК — микропроцессор. Внешние устройства подключаются к магистрали через контроллеры. Основные характеристики ПК: тактовая частота, объем внутренней памяти, разрядность процессора Знать характеристики школьного ПК, типы устройств
Программное обеспечение ЭВМ	
1. Что такое ПО и его назначение?	ПО — вся совокупность программ в долговременной памяти компьютера, предназначенных для массового использования
2. Классификация ПО ЭВМ	Три типа ПО: системное ПО, прикладное ПО, системы программирования
3. Что такое прикладное ПО?	Прикладное ПО — программы, предназначенные для решения информационных задач пользователя. Прикладное ПО бывает общего назначения и специализированное
4. Назначение систем программирования	Предоставление программисту средств для составления, отладки и исполнения программ на языках программирования
5. Состав системного ПО	Основная часть системного ПО — операционная система. Кроме того существуют сервисные программы: обслуживание дисков, архиваторы, антивирусные программы и др.

6. Основные функции операционной системы	1) Управление ресурсами компьютера; 2) взаимодействие (диалог) с пользователем; 3) управление файлами (файловая система)
7. Начальные сведения об организации файлов	Имя файла: <имя>.<тип> . Файловая структура — совокупность файлов и взаимосвязей между ними. Одноуровневые и многоуровневые (иерархические) файловые структуры. Дерево — графическое отображение иерархической файловой структуры. Путь к файлу, полное имя файла

Глава 3. Текстовая информация и компьютер

Представление текстовой информации	
1. Размер символьного компьютерного алфавита; объем памяти, занимаемый символом	256 символов; 1 символ — 1 байт
2. Что такое таблица кодировки; что такое код символа?	Таблица кодировки — пронумерованный символьный алфавит. Код символа — порядковый номер символа (от 0 до 255). Внутренний код — порядковый номер в двоичной с. с.
3. Примеры таблиц кодировки; структура таблицы кодировки	ASCII, КОИ-8. Коды от 0 до 127 — стандартная часть; от 128 до 255 — альтернативная часть; от 0 до 31 — управляющие символы
4. Лексикографический порядок (принцип последовательного кодирования)	Буквы латинского алфавита и десятичные цифры в кодовой таблице всегда расположены в порядке возрастания номеров. Русский алфавит — не всегда

5. Что такое гипертекст?	Способ организации текста с внутренними смысловыми связями (гиперссылками)
6. Структура текстового файла	Последовательность символов (кодов), разделенная на строки
<i>Текстовые редакторы</i>	
7. Что такое текстовый редактор (текстовый процессор)? Примеры	Прикладная программа для создания и обработки текстовых документов. Примеры: Norton Editor, Лексикон, Блокнот, WordPad, MS Word
8. Что такое текстовый документ?	Файл, созданный с помощью ТР
9. Основные элементы среды ТР	Рабочее поле, текстовый курсор, строка меню, строка состояния
10. Основные параметры формата текста	Длина строки, межстрочное расстояние, выравнивание строк, отступы
11. Основные характеристики шрифтов	Тип, начертание, размер (кегель), эффекты
12. Для чего используется буфер обмена?	Для переноса и копирования фрагментов текста
13. Работа ТР с внешними устройствами	Сохранение текста на диске, открытие файла (вызов текста с диска в ОЗУ), распечатка текста на принтере
14. Дополнительные возможности ТР	Поиск и замена фрагментов текста; орфографический контроль; многооконный режим работы; вставка рисунков; подсказка и пр.

Глава 4. Графическая информация и компьютер

1. Что такое компьютерная графика?	Раздел информатики, занимающийся проблемами получения графических изображений (рисунков) на ЭВМ
2. Основные разделы компьютерной графики	Научная графика, конструкторская графика, деловая графика, иллюстративная графика, художественная и рекламная графика
3. Что такое анимация?	Движущиеся графические изображения на экране компьютера (мультфильмы)
4. Состав устройств компьютера для вывода изображения	Дисплей, видеоконтроллер, видеопамять, дисплейный процессор
5. Структура изображения на экране дисплея	Составлено из светящихся точек — пикселей
6. Структура пикселя цветного дисплея	Состоит из трех зерен, светящихся красным, синим и зеленым цветами. Все другие цвета получаются смешением трех основных цветов
7. Что такое растр, разрешающая способность экрана?	Размер сетки растра — сетки пикселей экрана: $M \times N$.
8. Назначение видеопамяти	Хранение видеоинформации: информации о цвете каждого пикселя экрана
9. Назначение дисплейного процессора	Управление работой дисплея в соответствии с информацией в видеопамяти
10. Что такое сканер?	Устройство для ввода изображения с листа
11. Связь между длиной кода пикселя — b и числом цветов — K (размером палитры)	$K = 2^b$

12. Как определить минимальный размер видеопамати?	$Vm = b \times M \times N$, где: b — длина кода пикселя; $M \times N$ — разрешающая способность экрана
13. Что такое графический редактор? Примеры	Прикладная программа для получения графических изображений (для рисования и черчения) Paintbrush, Paint, CorelDraw! и др.
14. Основные элементы среды ГР	Рабочее поле, графический курсор, меню команд, панель инструментов, палитра
15. Основные инструменты для рисования (в растровых редакторах)	Карандаш, Кисть, Ластик, Заливка, графические примитивы: линии (прямые, ломаные, дуги), Прямоугольник, Овал
16. Основные возможности манипулирования с рисунком	Копирование фрагментов, повороты, отражение, масштабирование, прорисовка деталей
17. Работа ГР с внешними устройствами	Сохранение рисунка в графическом файле, открытие файла, распечатка рисунка

Глава 5. Передача информации в компьютерных сетях

1. Что такое компьютерная сеть?	Множество компьютеров, связанных каналами передачи информации
2. Что такое локальная компьютерная сеть?	Сеть в пределах одного помещения, здания, предприятия
3. Что такое глобальная компьютерная сеть?	Объединение локальных сетей и отдельных компьютеров, расположенных на больших расстояниях
4. Какие виды линий связи используются в компьютерных сетях?	Телефонная сеть, кабельные линии (электрические и оптические), радиосвязь

5. Состав локальной сети	Файл-сервер, рабочие станции
6. Программное обеспечение локальной сети	Сетевая операционная система
7. Назначение файл-сервера локальной сети	Хранение файлов (программ и данных), предназначенных для общего доступа пользователей
8. Аппаратные средства глобальной сети	Хост-компьютер, ПК абонентов — терминалы, линии связи; шлюзы — компьютеры, связывающие разные сети
9. Что такое модем?	Модем (модулятор-демодулятор) — устройство для подключения компьютера к сети через телефонную линию связи. Преобразует двоичную информацию в аналоговый сигнал телефонной связи и обратно
10. Что такое сетевой протокол?	Стандарт на представление, интерпретацию и передачу информации
11. Что такое Интернет?	Всемирная глобальная компьютерная сеть
12. Что такое технология клиент—сервер?	Принцип организации программного обеспечения в глобальных сетях. Каждая информационная услуга обслуживается сервер-программой и клиент-программой
13. Основные виды информационных услуг в глобальных сетях	Электронная почта, телеконференции, передача файлов, базы данных, WWW и др.
14. Что такое электронная почта?	Обмен письмами в глобальных сетях
15. Что такое почтовый ящик?	Раздел дисковой памяти на почтовом сервере, отведенный для пользователя
16. Из чего состоит электронное письмо?	Конверт (адресная часть), текст письма, приложение

17. Структура электронного адреса	<имя почтового ящика> @ <адрес почтового сервера>
18. Что такое телеконференция?	Система обмена информацией на определенную тему (темую конференции) между абонентами сети (подписчиками конференции)
19. Что такое World Wide Web?	Всемирная паутина. Информационная система с гиперсвязями в Интернете
20. Что такое Web-страница?	Отдельный документ в WWW. Основная информационная единица WWW
21. Что такое Web-сервер?	Компьютер в сети Интернет, хранящий Web-страницы и соответствующее программное обеспечение для работы с ними
22. Клиент-программа WWW	Web-браузер. Позволяет извлекать и просматривать Web-страницы

Глава 6. Модели и таблицы

1. Что такое модель?	Упрощенное подобие реального объекта (процесса)
2. Какие бывают модели?	Материальные (натурные), информационные
3. Что такое информационная модель?	Описание объекта моделирования
4. Что такое формализация?	Замена реального объекта его информационной моделью
5. Формы информационных моделей	Вербальная, графическая, табличная, математическая
6. Примеры графических моделей	Карты, схемы, чертежи, графики

7. Структурные части таблицы	Строки, столбцы, ячейки
8. Какая информация содержится в таблице типа «объект»–«свойство»?	В строке — информация об отдельном объекте, в столбце — свойство (характеристика) объекта; в ячейке — значение конкретного свойства конкретного объекта
9. Какая информация содержится в таблице типа «объект–объект»?	В строках и в столбцах — разные объекты; в ячейках — данные о связи (отношении) между соответствующими объектами
10. Что такое двоичная матрица?	Таблица типа «объект–объект», в ячейках которой помещены нули и единицы (1 — связь есть, 0 — связи нет)
Вопросы для углубленного уровня	
11. Что такое система?	Сложный объект, состоящий из множества взаимосвязанных элементов
12. Что такое структура?	Определенный порядок объединения элементов в системе
13. Что такое граф?	Графическое отображение состава и структуры системы
14. Что такое дерево?	Граф иерархической системы. От корня дерева к любому элементу существует единственный путь

Глава 7. Базы данных

1. Что такое база данных?	Организованная совокупность данных во внешней памяти ЭВМ, предназначенная для постоянного применения
2. В чем разница между фактографической и документальной БД?	В фактографической БД — краткая информация в фиксированном формате, в документальной — документы разного типа и объема
3. Что такое распределенная БД?	БД, работающая в компьютерной сети и хранящаяся на разных компьютерах
4. Что такое реляционная БД?	БД с табличной организацией данных
5. Что такое запись, поле?	Запись — строка таблицы; поле — столбец таблицы
6. Что такое главный ключ записи?	Поле, отличающее записи друг от друга. Составной ключ — совокупность полей
7. Какие существуют основные типы полей?	Числовой, символьный, логический, дата/время
8. Что такое СУБД?	Система управления базами данных — программное обеспечение для работы с БД
9. Примеры реляционных СУБД	dBase, FoxPro, Ребус, Карат, Access
10. Что можно делать с информацией в БД средствами СУБД?	Заносить, изменять, удалять, искать, сортировать
11. Что указывается в запросе на получение справки из БД?	Выводимые поля и условия поиска
12. Что такое условие поиска?	Логическое выражение
13. Что представляет собой простое логическое выражение?	Одна величина логического типа или одно отношение

14. Что такое сложное логическое выражение?	Логическое выражения, содержащее логические операции
15. Какие существуют основные логические операции?	И — логическое умножение (конъюнкция); ИЛИ — логическое сложение (дизъюнкция); НЕ — отрицание (инверсия)
16. Правила выполнения логических операций	Таблица истинности (уметь представлять)
17. Старшинство логических операций	По убыванию: НЕ, И, ИЛИ
18. Что указывается в запросе на удаление записей?	Условие поиска удаляемых записей (логическое выражение)
19. Что указывается в запросе на сортировку?	Ключи сортировки и порядок сортировки (по возрастанию или убыванию значений ключа)
20. Что такое структура таблицы?	Перечень имен полей, типов полей и форматов полей таблицы
21. Этапы разработки БД	Проектирование, создание структуры, заполнение (ввод данных)

Глава 8. Табличные вычисления на компьютере

1. Для чего используются электронные таблицы?	Для организации табличных расчетов на ПК
2. Что такое табличный процессор? Примеры	Прикладная программа для работы с электронными таблицами. Supercalc, Excel
3. Наименьший структурный элемент таблицы	Ячейка
4. Как идентифицируется ячейка?	Имя столбца и номер строки: A1, C5 и т. п.
5. Какая информация заносится в ячейки таблицы?	Тексты, числа, формулы

6. В чем состоит основное свойство электронной таблицы	Мгновенный пересчет формул при изменении значений величин, входящих в формулы
7. Правила записи символьных и числовых данных (для конкретного ТП)	Уметь организовывать ввод текстов и чисел
8. Правила записи формул	Запись в строку; начинаются со знака «=». Уметь проставлять все операции, использовать круглые скобки, знать старшинство операций: функции, * , * и $/$, $+$ и $-$
9. Что такое блок таблицы?	Прямоугольный фрагмент. Обозначается: <лев.верх.яч.>:<прав.нижн.яч.>
10. Какие вычисления можно выполнять над блоком таблицы?	Сумма чисел, среднее значение, выбор максимального и минимального значения и др.
11. Что такое принцип относительной адресации?	Адреса ячеек в формулах определены относительно места расположения формулы
12. Какие манипуляции можно выполнять с таблицей?	Копировать и переносить фрагменты; вставлять и удалять строки и столбцы; сортировать строки по значению столбца
13. Что такое графическая обработка данных в электронной таблице?	Построение диаграмм и графиков по табличным данным (деловая графика)
14. Как можно отменить принцип относительной адресации?	Путем «замораживания» адреса ячейки, используя значок \$
15. Что такое условная функция?	Функция вида: ЕСЛИ (условие, выражение 1, выражение 2). Условие — логическое выражение. Если условие истинно, то выполнится выражение 1, иначе — выражение 2
16. Как реализованы логические операции в ЭТ?	В виде логических функций: И (условие 1, условие 2), ИЛИ (условие 1, условие 2), НЕ (условие)

Глава 9. Информация и управление

1. Что такое кибернетика?	Наука об управлении в живых и неживых системах
2. Кто основал кибернетику?	Американский математик Норберт Винер
3. Из каких элементов с точки зрения кибернетики состоит всякая система управления?	Объект управления, управляющий объект, канал прямой связи, канал обратной связи
4. Для чего используется канал прямой связи?	Для передачи команд управления
5. Для чего используется канал обратной связи?	Для передачи данных о состоянии объекта управления
6. Что такое алгоритм управления?	Последовательность команд управления
7. Какую структуру может иметь алгоритм управления в системе без обратной связи?	Линейную (последовательную) структуру
8. Какую структуру может иметь алгоритм в системах с обратной связью?	Циклическую и ветвящуюся
9. Какую роль выполняют АСУ?	Автоматизированные системы управления применяются в производстве для сбора и анализа информации, для помощи в принятии управляющих решений
10. Для чего используются системы автоматического управления?	САУ используются для программного управления техническими устройствами
11. Что такое исполнитель алгоритма?	Объект, для управления которым составлен алгоритм

12. Что такое система команд исполнителя (СКИ)?	Конечное множество команд, которые исполнитель умеет выполнять
13. Что обозначает свойство понятности алгоритма?	В алгоритм должны входить только те команды, которые включены в СКИ исполнителя
14. Что означает свойство точности алгоритма?	Каждая команда алгоритма должна определять однозначное действие исполнителя
15. Что такое конечность алгоритма?	За конечное число шагов (выполненных команд) должен быть получен результат
16. Что такое полный набор данных?	Начальные данные, необходимые для получения однозначного результата
17. В чем различие между программой и алгоритмом?	Различие может быть только в форме описания. Программа записывается в строгом соответствии с правилами языка исполнителя
18. Что значит формальное исполнение алгоритма?	Исполнителю не требуется принимать самостоятельных решений во время исполнения алгоритма
19. Примеры учебных исполнителей	Робот, Чертежник, Кенгуренок, Черепашка и др.
20. Описать систему команд одного из учебных исполнителей	Уметь описывать СКИ
21. Что такое линейный алгоритм?	Алгоритм, в котором команды выполняются последовательно, каждая один раз

22. Что такое вспомогательный алгоритм?	Алгоритм решения некоторой подзадачи для исходной задачи. Обычно его исполнение повторяется
23. Что такое цикл?	Структурная команда, обозначающая повторное выполнение серии команд по некоторому условию
24. Что такое ветвление?	Структурная команда, обозначающая выбор одного из двух путей продолжения алгоритма в зависимости от условия. В конце — выход на общее продолжение
25. Какие существуют способы описания алгоритмов?	Блок-схемы, учебный алгоритмический язык
26. Изображение в блок-схемах простых и структурных команд алгоритма	Уметь изображать следования, ветвления, циклы
27. Как могут соединяться между собой структурные команды?	Последовательно и вложением
28. Что такое метод последовательной детализации?	Метод построения сложных алгоритмов. Сначала составляется основной алгоритм, затем вспомогательные алгоритмы первого уровня, затем второго и т. д.

Глава 10. Как работает процессор ЭВМ

1. В какой системе счисления представляются числа в памяти ЭВМ?	В двоичной системе счисления
2. Для чего используется шестнадцатеричная система счисления?	Для компактного внешнего представления (на экране или в распечатке) внутренней информации
3. Что такое ячейка памяти?	Часть памяти, доступная для обработки одной командой процессора. Содержимое ячейки — машинное слово
4. Чему равен адрес ячейки?	Адрес ячейки равен адресу первого (младшего) байта, входящего в ячейку
5. Какая информация может храниться в ячейке памяти?	Команда программы или одна величина (например, число)
6. В чем заключается принцип хранимой программы Дж. фон Неймана?	В оперативной памяти компьютера наряду с данными помещается программа управления его работой
7. Какая информация может содержаться в одной команде программы?	Код операции — какую операцию надо выполнить, адресная часть — адреса операндов и адрес результата, адрес следующей выполняемой команды (в командах перехода)
8. Что такое ЯМК?	Язык машинных команд — система команд процессора ЭВМ
9. Из чего состоит процессор?	1) устройство управления; 2) арифметико-логическое устройство; 3) регистры — внутренняя память процессора.
10. Назначение устройства управления	Управление работой всех других устройств ЭВМ в соответствии с программой

11. Назначение процессора	Выполнение команд программы по обработке данных
12. Назначение регистров процессора	Хранение очередной выполняемой команды (регистр команд), адреса очередной команды (счетчик команд), операндов, результата операции и пр.
13. В какой последовательности процессор выполняет программу?	Начиная с первой команды в порядке возрастания адресов
14. Что такое присваивание?	Занесение в ячейку некоторого значения в результате выполнения команды
15. Какие команды обеспечивают связь между компьютером и человеком?	Команды ввода и вывода

Глава 11. Искусственный интеллект и базы знаний

1. Что такое искусственный интеллект?	Раздел информатики, целью которого является разработка компьютерных интеллектуальных систем
2. Примеры компьютерных интеллектуальных систем	Шахматные программы, перевод с одного языка на другой, сочинение музыки, распознавание рукописного текста, доказательство теорем, экспертные системы и др.
3. Что такое экспертная система?	Система ИИ, заключающая в себе знания специалиста-эксперта в определенной предметной области
4. Для чего используются экспертные системы?	Для консультации пользователя; для помощи в принятии сложных решений
5. Что такое база знаний?	Компьютерная модель знаний специалиста в определенной предметной области

6. Что такое механизм вывода?	Модель логических рассуждений на основе базы знаний
7. Из чего состоит логическая модель знаний?	Из фактов и правил
8. Что такое факт?	Сведения частного характера. Приводить примеры
9. Что такое правило?	Утверждение общего характера, справедливое для многих объектов. Приводить примеры
Вопросы для углубленного изучения	
10. Что такое Пролог?	Язык логического программирования
11. Что такое предикат?	Конструкция вида: имя(аргументы). Обозначает некоторое отношение между аргументами или свойство (если аргумент один)
12. Из чего состоит база знаний на Прологе?	Из фактов и правил
13. Что такое факт на Прологе?	Предикат, у которого аргументы являются константами
14. Что такое правило?	Предложение вида <голова правила>:—<тело правила>. Значок :— читается «если». Если истинно тело правила, то истинна голова правила. Приводить примеры
15. Что обозначает запрос (цель) первого типа?	Подтвердить справедливость факта. Ответ: да или нет
16. Что обозначает запрос второго типа?	Указать все значения переменных, присутствующие в запросе, удовлетворяющие базе знаний
17. В каких случаях в Прологе употребляется конъюнкция?	В теле правила и в сложных запросах

Глава 12. Введение в программирование

Алгоритмы работы с величинами	
1. Что такое величина?	Отдельный информационный объект, занимающий определенное место в памяти (ячейку памяти)
2. Что такое константа?	Величина с постоянным значением
3. Что такое переменная?	Символически обозначаемая величина, значение которой может меняться
4. Основные типы величин	Числовые (целый, вещественный), символьный, логический
5. Чем различаются величины различных типов?	Допустимыми значениями, допустимыми операциями, формой внутреннего представления
6. Из каких команд составляется любой алгоритм работы с величинами?	Присваивания, ввода, вывода, обращения к вспомогательному алгоритму, цикла, ветвления
7. Как представляется и выполняется команда присваивания?	<переменная>:=<выражение> 1) Вычисляется выражение; 2) полученное значение присваивается переменной
8. Как представляется и выполняется команда ввода?	Ввод <список переменных>. Значения переменных задаются через устройства ввода
9. Как представляется и выполняется команда вывода?	Вывод <список вывода>. Значения элементов списка вывода выносятся на устройства вывода (сообщаются пользователю)
10. Как представляется и выполняется команда цикла?	Пока <условие повторения> повторять <тело цикла> Условие повторения — простое или сложное логическое выражение Пока истинно условие повторения, повторяются команды тела цикла

11. Как представляется и выполняется команда ветвления?	Если <условие> то <серия 1> иначе <серия 2> Если условие истинно, выполняется первая серия команд, иначе — вторая
12. Что такое параметры вспомогательного алгоритма?	Это переменные-аргументы и переменные-результаты
Системы и языки программирования	
13. Что такое уровень языка программирования?	Степень удаленности от языка машинных команд
14. Какие бывают уровни ЯП?	Машинноориентированные (ЯМК, автокод, ассемблер); машиннонезависимые (ЯПВУ: Паскаль, Бейсик, Си)
15. Какой язык вы изучали и каково его назначение?	Паскаль — универсальный язык программирования
16. Основные типы данных, используемые в Паскале	Целый, вещественный, символьный, логический
17. Структура программы на Паскале	Заголовок, разделы описаний, раздел операторов
18. Идентификация констант и переменных	Уметь записывать константы разных типов и описывать переменные
19. Правила записи арифметических выражений	Знать основные правила, уметь записывать выражения
20. Оператор присваивания	<переменная>:=<выражение>; Типы правой и левой частей должны быть согласованы
21. Операторы ввода и вывода	Read(список ввода), write(список вывода), а также readln и writeln
22. Операторы цикла	В минимальном варианте достаточно while <логическое выражение> do <тело цикла>
23. Условный оператор	If <логическое выражение> then <оператор 1> else <оператор 2>

24. Что такое трансляция? Что такое транслятор?	Трансляция — перевод с языка высокого уровня на ЯМК. Транслятор — программа-переводчик
25. В чем разница между компиляцией и интерпретацией?	Компиляция — полный перевод программы перед ее выполнением; интерпретация — перевод, осуществляемый параллельно с выполнением программы
26. Основные компоненты системы программирования	Транслятор с входного языка, текстовый редактор, библиотеки подпрограмм, отладчик и др.

Литература

1. *Семакин И. Г., Залогова Л. А., Русаков С. В., Шестакова Л. В.* Информатика. Базовый курс. 7—9: М.: Лаборатория Базовых Знаний, 1998.

2. *Задачник-практикум по информатике. Учеб. пособие для сред. школы / Под ред. И. Г. Семакина, Е. К. Хеннера.* М.: Лаборатория Базовых Знаний, 2003.

3. *Семакин И. Г., Шеина Т.* Преподавание базового курса информатики в средней школе: Методическое пособие. М.: Лаборатория Базовых Знаний, 2002.

4. *Семакин И. Г., Вараксин Г. С.* Информатика. Структурированный конспект базового курса. М.: Лаборатория Базовых Знаний, 2001.

5. *Семакин И. Г., Залогова Л. А., Русаков С. В., Шестакова Л. В.* Информатика. Базовый курс. 7—9. Второе издание: М.: БИНОМ. Лаборатория Знаний, 2003.

6. *Залогова Л. А., Русаков С. В., Семакин И. Г., Хеннер Е. К., Шестакова Л. В.* Основы информатики и вычислительной техники в базовой школе: Пособие для учителя. Пермь, 1995.

7. Базовый курс ОИВТ: «Пермская версия» // Информатика и образование. 1994. № 5, 6; 1995, № 1, 2, 3, 5.

8. О минимальном содержании образовательных программ основной общеобразовательной школы // Информатика. 1997. № 79.

9. Основы информатики и вычислительной техники: Пробн. учеб. пособие для сред. учеб. завед. В 2-х ч. / Под ред. А. П. Ершова и В. М. Монахова. М.: Просвещение, 1985 (ч. 1), 1986 (ч. 2).

10. Кушниренко А. Г., Лебедев Г. В., Сковорень Р. А. Основы информатики и вычислительной техники. М.: Просвещение, 1991.
11. Каймин В. А., Щеголев А. Г., Ерохина Е. А., Федюшин Д. П. Основы информатики и вычислительной техники. М.: Просвещение, 1989.
12. Гейн А. Г., Линецкий Е. В., Сапир М. В., Шолохович В. Ф. Информатика. М.: Просвещение, 1994.
13. Программы средней общеобразовательной школы. Основы информатики и вычислительной техники. М.: Просвещение, 1991.
14. Базисный учебный план средней общеобразовательной школы // Вестник образования. 1993. № 9.
15. Проект федерального компонента государственного образовательного стандарта начального общего, основного общего и среднего (полного) образования. Образовательная область «Информатика». Авторский коллектив под рук. А. А. Кузнецова // Информатика и образование. 1997. № 1.
16. Основные компоненты содержания информатики в общеобразовательных учреждениях. // Информатика и образование. 1995. № 4.
17. Информатика. Энциклопедический словарь для начинающих / Под. ред. Д. А. Поспелова. М.: Педагогика-Пресс, 1994.
18. Мозилев А. В., Пак Н. И., Хеннер Е. К. Информатика: Учеб. пособие для студ. пед. вузов. М.: Академия, 1999.
19. Лалчик М. П., Семакин И. Г., Хеннер Е. К. Методика преподавания информатики: Учеб. пособие для студ. пед. вузов. М.: Академия, 2001.
20. Винер Н. Кибернетика. М., 1983.

21. *Частиков А. П.* История компьютера. М.: Информатика и образование, 1996.

22. *Андреева Е., Фалина И.* Системы счисления и компьютерная арифметика. М.: Лаборатория Базовых Знаний, 1999.

23. *Леонтьев В.* Новейшая энциклопедия персонального компьютера. М.: Олма-ПРЕСС, 1999.

24. *Шафрин Ю. А.* Информационные технологии. В 2-х ч. М.: Лаборатория Базовых Знаний, 2000.

25. *Мартин Дж.* Организация баз данных в вычислительных системах. М.: Мир, 1980.

26. *Пейперт С.* Дети, компьютер и плодотворные идеи / Пер. с англ. М.: Мир, 1990.

27. *Гейн А. Г., Сенокосов А. И., Шолохович В. Ф.* Информатика 7—9. М.: Дрофа 1998.

28. *Братко И.* Программирование на языке Пролог для искусственного интеллекта. М.: Мир, 1990.

29. *Ин Ц., Соломон Д.* Использование Турбо Пролога. М.: Мир, 1993.

30. *Пугач В. И., Добудько Т. В.* Элементы логики и программирования в системе Турбо Пролог. Самара.: СамГПИ. 1993.

31. *Угринович Н. Д.* Информатика и информационные технологии. 10—11. М.: Лаборатория Базовых Знаний, 2000.

32. *Еремин Е. А.* Как работает современный компьютер. Пермь: изд-во ПРИПИТ, 1997.

33. *Супер-ЭВМ.* Аппаратная и программная организация / Под ред. С. Фернбаха. М.: Радио и связь, 1991.

Учебное издание

**Семакин Игорь Геннадьевич,
Шеина Татьяна Юрьевна**

**ПРЕПОДАВАНИЕ БАЗОВОГО КУРСА ИНФОРМАТИКИ
В СРЕДНЕЙ ШКОЛЕ**
Методическое пособие

Редактор О. Полежаева

Художественный редактор Н. Лозинская

Компьютерная верстка Л. Катуркина, Е. Голубова

Подписано в печать 17.10.03. Формат 84x108 $\frac{1}{32}$.
Бумага газетная. Гарнитура Школьная. Печать офсетная.
Усл. печ. л. 28,56. Тираж 3000 экз. Заказ 2280

Издательство «БИНОМ. Лаборатория знаний»

Адрес для переписки: 103473, Москва, а/я 9

Телефон: (095)955-0398, E-mail: lbz@aha.ru

**Отпечатано с готовых диапозитивов
в полиграфической фирме «Полиграфист»
160001, г. Вологда, ул. Челюскинцев, 3**

ИНФОРМАТИКА

для преподавателя

со 2-го по 11-й классы

Преподавание базового курса информатики в средней школе

Структурированный комплекс базового курса

Системы счисления и компьютерная арифметика

ИНФОРМАТИКА: Процессы, системы, ресурсы

ЭКЗАМЕНАЦИОННЫЕ БИЛЕТЫ И ОТВЕТЫ. 11 класс

Построение тестовых заданий по информатике

Формализация и моделирование

Язык в информатике

Тестовые задания

Кабинет информатики

Информационное моделирование

ЗАДАНИЯ И ПРАКТИКУМ

Задачи и практикум по информатике (в 2-х частях)

Практикум по информационным технологиям (7-11 классы)

Практикум по информатике и информационным технологиям

Практикум по компьютерной графике

Уроки Web-мастера

Основы программирования

Программирование в алгоритмах

ISBN 5-94774-093-1



9 795947 740935