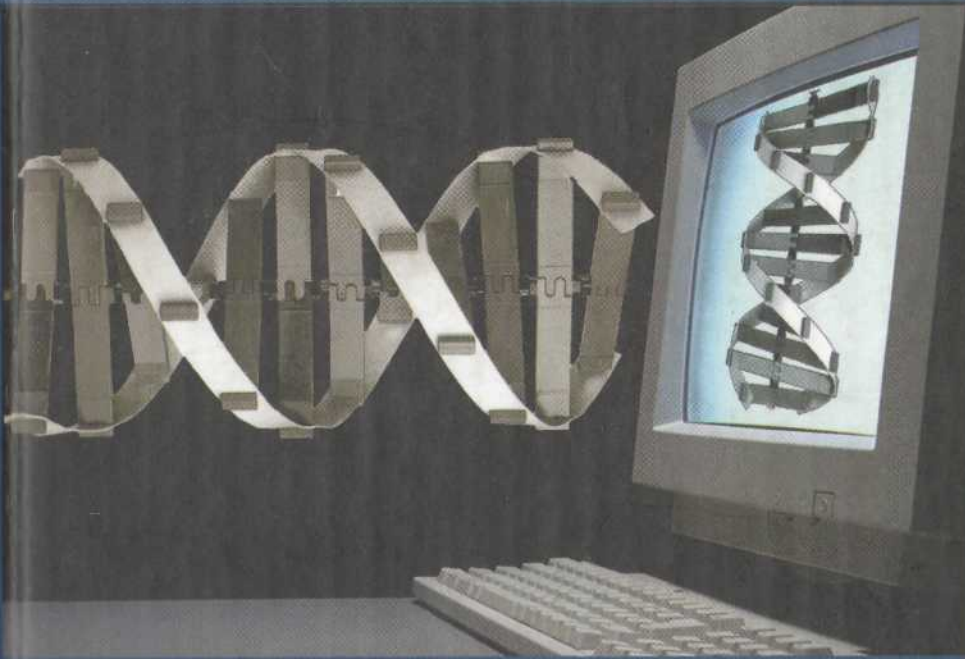


11

Н. Д. Угринович



ИНФОРМАТИКА И ИКТ



БИНОМ

Н. Д. Угринович

ИНФОРМАТИКА и ИКТ

БАЗОВЫЙ УРОВЕНЬ

Учебник для 11 класса



Москва
БИНОМ. Лаборатория знаний
2008

УДК 004.9

БК 32.97

У27

Угринович Н. Д.

У27 Информатика и ИКТ. Базовый уровень : учебник для 11 класса / Н. Д. Угринович. — М. : БИНОМ. Лаборатория знаний, 2008. — 188 с. : ил.

ISBN 978-5-94774-861-1

Учебник по курсу «Информатика и ИКТ-11. Базовый уровень» ориентирован на преподавание в общеобразовательных учреждениях профильного курса на базовом уровне в 11 классе. Учебник полностью соответствует образовательному стандарту, утвержденному Министерством образования и науки РФ. В учебнике рассматриваются архитектура компьютера и методы защиты информации, системный подход к моделированию, формализация и ее визуализация с использованием интерактивных компьютерных моделей, базы данных и СУБД. Большое внимание уделяется формированию практических умений и навыков в процессе выполнения практических компьютерных работ. Учебник мультисистемный, так как практические работы могут выполняться в операционных системах Windows и Linux. В учебник помещены тесты для подготовки к ЕГЭ по курсу «Информатика и ИКТ».

УДК 004.9

БК 32.97

По вопросам приобретения обращаться:

«БИНОМ. Лаборатория знаний»
(499) 157-52-72, e-mail: Lbz@aha.ru
<http://www.Lbz.ru>

ISBN 978-5-94774-861-1

© Угринович Н. Д., 2008

© БИНОМ. Лаборатория знаний, 2008

Оглавление

Рекомендации по использованию учебника	7
Глава 1. Компьютер как средство автоматизации информационных процессов	9
1.1. История развития вычислительной техники.	10
<i>Практическая работа 1.1. Виртуальные компьютерные музеи</i>	<i>15</i>
1.2. Архитектура персонального компьютера	19
<i>Практическая работа 1.2. Сведения об архитектуре компьютера</i>	<i>23</i>
1.3. Операционные системы!	25
1.3.1. Основные характеристики операционных систем	25
<i>Практическая работа 1.3. Сведения о логических разделах дисков</i>	<i>28</i>
1.3.2. Операционная система Windows	30
<i>Практическая работа 1.4. Значки и ярлыки на Рабочем столе</i>	<i>34</i>
1.3.3. Операционная система Linux.	36
<i>Практическая работа 1.5. Настройка графического интерфейса для операционной системы Linux</i>	<i>40</i>
<i>Практическая работа 1.6. Установка пакетов в операционной системе Linux</i>	<i>41</i>
1.4. Защита от несанкционированного доступа к информации.	43
1.4.1. Защита с использованием паролей.	43
1.4.2. Биометрические системы защиты	45

<i>Практическая работа 1.7. Биометрическая защита: идентификация по характеристикам речи</i>	48
1.5. Физическая защита данных на дисках	49
1.6. Защита от вредоносных программ	51
1.6.1. Вредоносные и антивирусные программы	51
1.6.2. Компьютерные вирусы и защита от них	53
<i>Практическая работа 1.8. Защита от компьютерных вирусов</i>	56
1.6.3. Сетевые черви и защита от них	63
<i>Практическая работа 1.9. Защита от сетевых червей</i>	66
1.6.4. Троянские программы и защита от них	71
<i>Практическая работа 1.10. Защита от троянских программ</i>	72
1.6.5. Хакерские утилиты и защита от них	75
<i>Практическая работа 1.11. Защита от хакерских атак</i> ...	76
Глава 2. Моделирование и формализация	79
2.1. Моделирование как метод познания	80
2.2. Системный подход в моделировании.	82
2.3. Формы представления моделей.	84
2.4. Формализация	86
2.5. Основные этапы разработки и исследования моделей на компьютере	88
2.6. Исследование интерактивных компьютерных моделей	89
2.6.1. Исследование физических моделей	89
2.6.2. Исследование астрономических моделей	91
2.6.3. Исследование алгебраических моделей	92
2.6.4. Исследование геометрических моделей (планиметрия)	94
2.6.5. Исследование геометрических моделей (стереометрия)	95
2.6.6. Исследование химических моделей	97
2.6.7. Исследование биологических моделей	98

Глава 3. Базы данных. Системы управления базами данных (СУБД)	101
3.1. Табличные базы данных	101
3.2. Система управления базами данных	104
3.2.1. Основные объекты СУБД: таблицы, формы, запросы, отчеты	104
<i>Практическая работа 3.1. Создание табличной базы данных</i>	106
3.2.2. Использование формы для просмотра и редактирования записей в табличной базе данных	108
<i>Практическая работа 3.2. Создание формы в табличной базе данных</i>	109
3.2.3. Поиск записей в табличной базе данных с помощью фильтров и запросов	113
<i>Практическая работа 3.3. Поиск записей в табличной базе данных с помощью фильтров и запросов</i>	114
3.2.4. Сортировка записей в табличной базе данных	117
<i>Практическая работа 3.4. Сортировка записей в табличной базе данных</i>	118
3.2.5. Печать данных с помощью отчетов	119
<i>Практическая работа 3.5. Создание отчета в табличной базе данных</i>	119
3.3. Иерархические базы данных	120
3.4. Сетевые базы данных	124
<i>Практическая работа 3.6. Создание генеалогического древа семьи</i>	125
Глава 4. Информационное общество	127
4.1. Право в Интернете ..	127
4.2. Этика в Интернете	128
4.3. Перспективы развития информационных и коммуникационных технологий	131
Глава 5. Повторение. Подготовка к ЕГЭ. Тесты по темам курса «Информатика и ИКТ»	137
Тема 1. Информация. Кодирование информации ..	138
Тема 2. Устройство компьютера и программное обеспечение	142

Тема 3. Алгоритмизация и программирование . . .	146
Тема 4. Основы логики и логические основы компьютера	156
Тема 5. Моделирование и формализация	159
Тема 6. Информационные технологии	161
Тема 7. Коммуникационные технологии	168
Ответы на тесты	170
Тема 1. Информация. Кодирование информации	170
Тема 2. Устройство компьютера и программное обеспечение	171
Тема 3. Алгоритмизация и программирование . . .	172
Тема 4. Основы логики и логические основы компьютера	181
Тема 5. Моделирование и формализация	182
Тема 6. Информационные технологии	184
Тема 7. Коммуникационные технологии	188


Рекомендации по использованию учебника

1. Учебник «Информатика и ИКТ-11» обеспечивает изучение профильного курса «Информатика и ИКТ» в 11 классе на базовом уровне.
2. Учебник входит в состав учебно-программного комплекса по основному курсу, профильному курсу на базовом и профильном уровнях и элективному курсу, включающего:
 - учебники для основной школы: «Информатика и ИКТ-8» и «Информатика и ИКТ-9»;
 - учебники по профильному курсу для старшей школы на базовом уровне: «Информатика и ИКТ-10. Базовый уровень» и «Информатика и ИКТ-11. Базовый уровень»;
 - учебники по профильному курсу для старшей школы на профильном уровне: «Информатика и ИКТ-10. Профильный уровень» и «Информатика и ИКТ-11. Профильный уровень»;
 - учебное пособие и CD-ROM по элективному курсу для старшей школы «Исследование информационных моделей»;
 - методическое пособие для учителей «Преподавание курса «Информатика и ИКТ» в основной и старшей школе», к которому прилагаются:
 - Windows-CD, содержащий свободно распространяемую программную поддержку курса, готовые компьютерные проекты, рассмотренные в учебниках, тесты и методические материалы для учителей;
 - Visual Studio-CD (выпускается по лицензии Microsoft), содержащий дистрибутивы систем объектно-ориентированного программирования языков Visual Basic .NET, Visual C# и Visual J#;
 - Linux-DVD (выпускается по лицензии компании AltLinux), содержащий операционную систему Linux и программную поддержку курса;
 - TurboDelphi-CD (выпускается по лицензии компании Borland), содержащий систему объектно-ориентированного программирования Turbo Delphi.
3. В практических работах указано необходимое для их выполнения программное обеспечение, которое обозначается значком приложения. В случае выделения часов на предмет «Информатика и ИКТ» не больше, чем в Федеральном базисном учебном плане, рекомендуется выпол-

- нять практические задания компьютерного практикума в одной операционной системе (Windows или Linux).
4. Возможно выполнение практических занятий во внеурочное время в компьютерном школьном классе или дома.
 5. В учебнике используются ссылки на внешние источники информации (учебники, CD-диски и Интернет), а также на параграфы и пункты самого учебника:

Кодовые таблицы Windows-CD








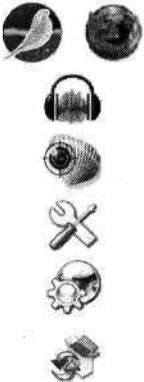

1.5. Кодирование и обработка звуковой информации

6. В тексте пособия приняты следующие шрифтовые выделения:
 - Шрифтом Arial выделены имена программ, файлов и Интернет-адреса.
 - Шрифтом Courier New выделены программы на языках программирования.
 - *Курсивом* выделены названия диалоговых окон, пунктов меню и элементов управления (текстовых полей, кнопок и т. д.) графического интерфейса операционных систем и приложений.
 - **Полужирным шрифтом** выделены важные понятия.
7. Важная информация выделена в тексте восклицательным знаком, а формулы — цифровым обозначением.
8. Абзацы, содержащие дополнительную интересную информацию, выделены значком .
9. Дополнительные материалы и интерактивные тесты для проверки усвоения материала находятся в Интернете по адресу: <http://iit.metodist.ru>

Глава 1

Компьютер как средство автоматизации информационных процессов

При изучении данной главы рекомендуется установить следующее программное обеспечение для операционных систем Windows и Linux:

	<ul style="list-style-type: none">• браузеры Internet Explorer, SeaMonkey, Mozilla, Opera;• программу тестирования компьютера SiSoftware Sandra;• звуковой редактор Audacity;• Антивирус Касперского.	<p>Windows-CD</p>  <p>Первая помощь ПО 1.0. CD-1</p> 	 
	<p>Служебные программы обслуживания компьютера и его защиты от вредоносных программ.</p>	<p>Windows OS</p>	
	<ul style="list-style-type: none">• браузеры SeaMonkey, Mozilla;• звуковой редактор Audacity;• Антивирус KlamAV;• Центр управления графическим интерфейсом KDE;• файловый менеджер и браузер Konqueror. <p>Использовать репозитории для установки программ.</p>	<p>Linux-DVD</p> 	

1.1. История развития вычислительной техники

Вычисления в доэлектронную эпоху. Потребность счета предметов у человека возникла еще в доисторические времена. Древнейший метод счета предметов заключался в сопоставлении предметов некоторой группы (например, животных) с предметами другой группы, играющей роль счетного эталона. У большинства народов первым таким эталоном были пальцы (счет на пальцах).

Расширяющиеся потребности в счете заставили людей употреблять другие счетные эталоны (зарубки на палочке, узлы на веревке и т. д.).

Каждый школьник хорошо знаком со счетными палочками, которые использовались в качестве счетного эталона в первом классе.

В древнем мире при счете больших количеств предметов для обозначения определенного их количества (у большинства народов — десяти) стали применять новый знак, например зарубку на другой палочке. Первым вычислительным устройством, в котором стал применяться этот метод, стал **абак**.

Древнегреческий абак представлял собой посыпанную морским песком дощечку. На песке проводились бороздки, на которых камешками обозначались числа. Одна бороздка соответствовала единицам, другая — десяткам и т. д. Если в какой-то бороздке при счете набиралось более 10 камешков, их снимали и добавляли один камешек в следующий разряд. Римляне усовершенствовали абак, перейдя от песка и камешков к мраморным доскам с выточенными желобками и мраморными шариками (рис. 1.1).



Рис. 1.1.
Древнеримский абак

По мере усложнения хозяйственной деятельности и социальных отношений (денежных расчетов, задач измерений расстояний, времени, площадей и т. д.) возникла потребность в арифметических вычислениях. Для выполнения простейших арифметических операций (сложения и вычитания) стали использовать абак, а по прошествии веков — **счеты** (рис. 1.2).

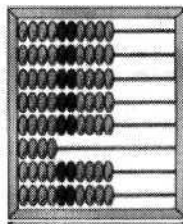


Рис. 1.2. Счеты

Развитие науки и техники требовало проведения все более сложных математических расчетов, и в XIX веке были изобретены механические счетные машины — **арифмометры** (рис. 1.3). Арифмометры могли не только складывать, вычитать, умножать и делить числа, но и запоминать промежуточные результаты, печатать результаты вычислений и т. д.

В середине XIX века английский математик Чарльз Бэббидж выдвинул идею создания программно управляемой счетной машины, имеющей арифметическое устройство, устройство управления, а также устройства ввода и печати.

Аналитическую машину Бэббиджа (прообраз современных компьютеров) по сохранившимся описаниям и чертежам построили энтузиасты из Лондонского музея науки (рис. 1.4). Аналитическая машина состоит из четырех тысяч стальных деталей и весит три тонны.

Вычисления производились Аналитической машиной в соответствии с инструкциями (программами), которые разработала леди Ада Лавлейс (дочь английского поэта Джорджа Байрона). Графиню Лавлейс считают первым программистом, и в ее честь назван язык программирования АДА.

Первыми носителями информации, которые использовались для хранения программ, были **перфокарты** (рис. 1.5). Программы записывались на перфокарты путем пробития в определенном порядке отверстий в плотных бумажных карточках. Затем перфокарты помещались в Аналитическую машину, которая считывала расположение отверстий и выполняла вычислительные операции в соответствии с заданной программой.

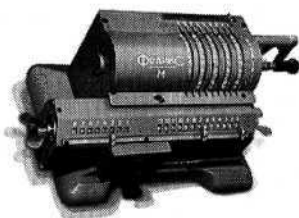


Рис. 1.3. Арифмометр середины XX века

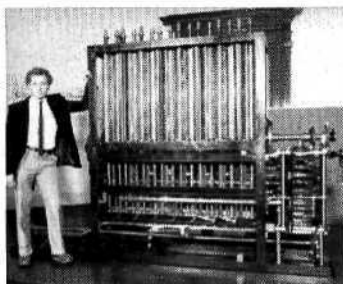


Рис. 1.4. Аналитическая машина Бэббиджа (реконструкция)

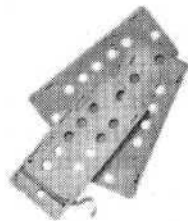


Рис. 1.5. Перфокарты к Аналитической машине

Развитие электронно-вычислительной техники

ЭВМ первого поколения. В 40-е годы XX века начались работы по созданию первых электронно-вычислительных машин, в которых на смену механическим деталям пришли **электронные лампы** (см. таблицу в конце параграфа). ЭВМ первого поколения требовали для своего размещения больших залов, так как в них использовались десятки тысяч электронных ламп. Такие ЭВМ создавались в единичных экземплярах, стоили очень дорого и устанавливались в крупнейших научно-исследовательских центрах.

В 1945 году в США был построен ENIAC (Electronic Numerical Integrator and Computer — электронный числовой интегратор и калькулятор), а в 1950 году в СССР была создана МЭСМ (Малая Электронная Счетная Машина) (рис. 1.6).



Рис. 1.6. МЭСМ

ЭВМ первого поколения могли выполнять вычисления со скоростью несколько тысяч операций в секунду, последовательность выполнения которых задавалась программами. Программы писались на машинном языке, алфавит которого состоял из двух знаков: 1 и 0.

Программы вводились в ЭВМ с помощью перфокарт или перфолент (рис. 1.7), причем наличие отверстия на перфокарте соответствовало знаку 1, а его отсутствие — знаку 0.

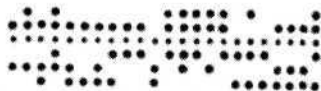


Рис. 1.7. Перфолента с записанной программой

Результаты вычислений выводились с помощью печатающих устройств в форме длинных последовательностей нулей и единиц. Писать программы на машинном языке и расшифровывать результаты вычислений могли только высококвал-

лифицированные программисты, понимавшие язык первых ЭВМ.

ЭВМ второго поколения. В 60-е годы XX века были созданы ЭВМ второго поколения, основанные на новой элементной базе — **транзисторах** (см. таблицу в конце параграфа), которые имеют в десятки и сотни раз меньшие размеры и массу, более высокую надежность и потребляет значительно меньшую электрическую мощность, чем электронные лампы. Такие ЭВМ производились малыми сериями и устанавливались в крупных научно-исследовательских центрах и ведущих высших учебных заведениях.

В СССР в 1967 году вступила в строй наиболее мощная в Европе ЭВМ второго поколения БЭСМ-6 (Большая Электронная Счетная Машина), которая могла выполнять 1 миллион операций в секунду (рис. 1.8).



Рис. 1.8. БЭСМ-6

В БЭСМ-6 использовалось 260 тысяч транзисторов, устройства внешней памяти на магнитных лентах для хранения программ и данных, а также алфавитно-цифровые печатающие устройства для вывода результатов вычислений.

Работа программистов по разработке программ существенно упростилась, так как стала проводиться с использованием языков программирования высокого уровня (Алгол, Бейсик и др.).

ЭВМ третьего поколения. Начиная с 70-х годов прошлого века, в качестве элементной базы ЭВМ третьего поколения стали использовать **интегральные схемы** (см. таблицу в конце параграфа). В интегральной схеме (маленькой полупроводниковой пластине) могут быть плотно упакованы тысячи транзисторов, каждый из которых имеет размеры, сравнимые с толщиной человеческого волоса.

ЭВМ на базе интегральных схем стали гораздо более компактными, быстродействующими и дешевыми. Такие мини-ЭВМ (рис. 1.9) производились большими сериями и были доступными для большинства научных институтов и высших учебных заведений.



Рис. 1.9. Мини-ЭВМ

Персональные компьютеры. Развитие высоких технологий привело к созданию больших интегральных схем — БИС (см. таблицу в конце параграфа), включающих десятки тысяч транзисторов. Это позволило приступить к выпуску компактных персональных компьютеров, доступных для массового пользователя.

Первым персональным компьютером был Apple II (рис. 1.10) («дедушка» современных компьютеров Macintosh), созданный в 1977 году. В 1982 году фирма IBM приступила к изготовлению персональных компьютеров IBM PC («дедушек» современных IBM-совместимых компьютеров).



Рис. 1.10. Первый персональный компьютер Apple II

Современные персональные компьютеры компактны и обладают в тысячи раз большим быстродействием по сравнению с первыми персональными компьютерами (могут выполнять несколько миллиардов операций в секунду). Ежегодно в мире производится почти 200 миллионов компьютеров, доступных по цене для массового потребителя.

Персональные компьютеры могут быть различного конструктивного исполнения: настольные, портативные (ноутбуки) и карманные (наладонники) (рис. 1.11).



Рис. 1.11. Современные персональные компьютеры

Контрольные вопросы

1. Используя текст параграфа и таблицу, ответьте на вопросы:

- Почему современные персональные компьютеры в сотни раз меньше, но при этом в сотни тысяч раз быстрее ЭВМ первого поколения?
- Почему современные персональные компьютеры доступны для массового потребителя?

Характеристика	Поколения			
	Первое	Второе	Третье	Персональные компьютеры
Годы использования	40–50 гг. XX в.	60-е гг. XX в.	70-е гг. XX в.	80-е гг. XX в. – настоящее время
Основной элемент	 Электронная лампа	 Транзистор	 Интегральная схема	 Сверхбольшая интегральная схема — СБИС (процессор)
Быстродействие (операций в секунду)	Десятки тысяч	Сотни тысяч	Миллионы	Миллиарды
Количество ЭВМ в мире, шт.	Сотни	Тысячи	Сотни тысяч	Около миллиарда

Практическая работа 1.1

Виртуальные компьютерные музеи

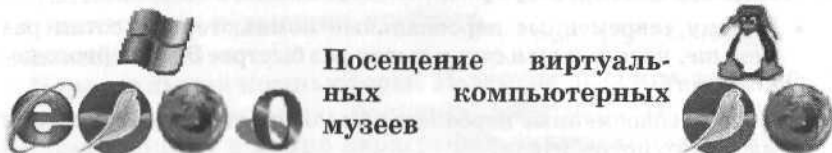
Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux, подключенный к Интернету.

Цель работы. Ознакомиться с историей развития отечественной и зарубежной вычислительной техники.

Задание. В браузере посетить в Интернете или на диске Windows-CD виртуальные компьютерные музеи и ознакомиться с историей развития отечественной и зарубежной вычислительной техники.

Варианты выполнения работы:

- в операционной системе Windows или в операционной системе Linux;
- просмотр Web-страниц виртуальных компьютерных музеев в различных браузерах (Internet Explorer, SeaMonkey, Mozilla, Opera или др.).



Посетим в Интернете Виртуальный музей информатики школы № 444 Москвы.

1. Запустить браузер в операционной системе Windows (например, Internet Explorer) или в операционной системе Linux и в поле *Адрес:* ввести <http://schools.keldysh.ru/sch444/MUSEUM/index.htm> (доменное имя и путь к начальной странице Виртуального музея информатики школы № 444 Москвы).
2. На появившейся начальной странице музея активизировать ССЫЛКИ.

Виртуальный музей ВТ(первая страница) - Windows Internet Explorer

http://schools.keldysh.ru/sch444/MUSEUM/index.htm

Виртуальный музей ВТ(первая страница)

Голосуйте за сайт!
Музей открыт в октябре 1998 года

Виртуальный музей Информатики

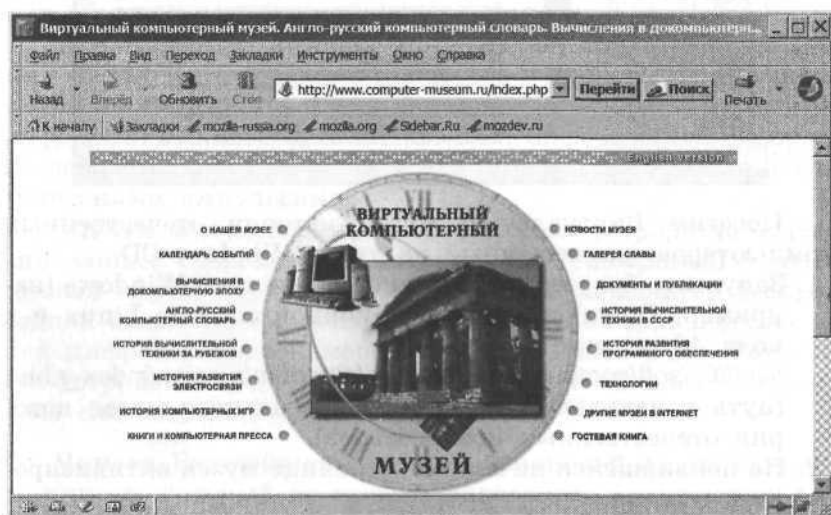
зарегистрирован на сервере "Музей России" в 1998 году в Отраслевом фонде алгоритмов и программ в 2003 под номером 2341

Дата последней коррективы 29 ноя 2007
более 200 страниц,
более 300 иллюстраций,
более 150 статей из журналов и газет

Галерея портретов	Музеи мира	Россия
Здесь найдете краткие очерки о жизни людей, посвятивших себя созданию вычислительной техники.	Приведен перечень виртуальных музеев мира, которые Вы сможете посетить.	История развития вычислительной техники у нас в России. Перечень наиболее известной вычислительной техники, разработанной российскими учеными.

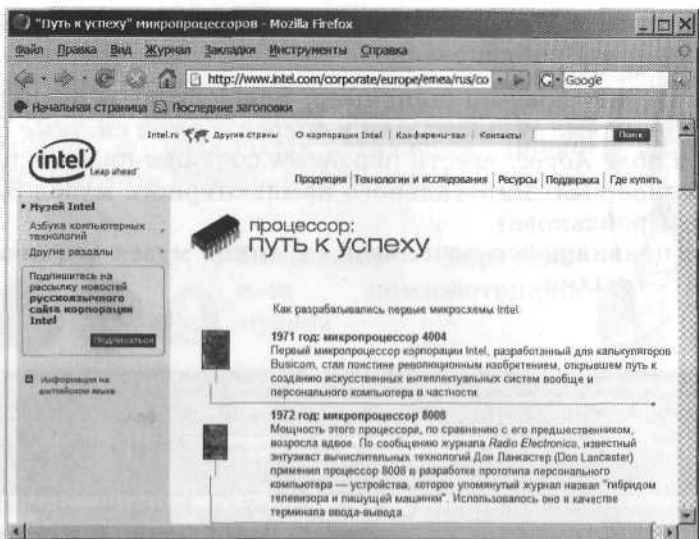
Посетим в Интернете Виртуальный компьютерный музей Эдуарда Пройдакова.

1. Запустить браузер (например, SeaMonkey) в операционной системе Windows или в операционной системе Linux и в поле *Адрес:* ввести <http://www.computer-museum.ru> (доменное имя Виртуального компьютерного музея Эдуарда Пройдакова).
2. На появившейся начальной странице музея активизировать ссылки.



Посетим в Интернете Виртуальный музей фирмы Intel.

1. Запустить браузер (например, Mozilla) в операционной системе Windows или в операционной системе Linux и в поле *Адрес:* ввести http://www.intel.com/corporate/europe/emea/rus/country/museum/history/hof/hof_main.htm (доменное имя и путь к начальной странице Виртуального музея фирмы Intel).
2. На появившейся начальной странице музея активизировать ссылки.



Посетим Виртуальный музей истории отечественных компьютеров, размещенный на диске Windows-CD.

1. Запустить браузер в операционной системе Windows (например, Opera) или в операционной системе Linux и в поле *Адрес:* ввести `file:///G:/soft/computer/virtmuseum/museum/russian/index_r.htm` (путь к начальной странице Виртуального музея истории отечественных компьютеров).
2. На появившейся начальной странице музея активизировать **ССЫЛКИ**.



1.2. Архитектура персонального компьютера

Магистрально-модульный принцип построения компьютера. В основу архитектуры современных персональных компьютеров положен магистрально-модульный принцип. Модульность позволяет потребителю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости ее модернизацию. Модульная организация компьютера опирается на магистральный (шинный) принцип обмена информацией между устройствами.

К магистральной, которая представляет собой три различные шины, подключаются процессор и оперативная память, а также периферийные устройства ввода, вывода и хранения информации, которые обмениваются информацией в форме последовательностей нулей и единиц, реализованных электрическими импульсами.

Многие необходимые дополнительные устройства интегрированы в современные материнские (системные) платы: сетевая карта, внутренний модем, сетевой адаптер беспроводной связи Wi-Fi, контроллер IEEE 1394 для подключения цифровой видекамеры, звуковая плата и др. Раньше эти устройства подключались к материнской плате с помощью слотов расширения и разъемов.

Чипсет. Важнейшей частью материнской платы является чипсет, который во многом определяет архитектуру современного персонального компьютера. Современные компьютеры содержат две основные большие микросхемы чипсета (рис. 1.12):

- контроллер-концентратор памяти, или Северный мост (англ. North Bridge), который обеспечивает работу процессора с оперативной памятью и с видеоподсистемой;
- контроллер-концентратор ввода/вывода, или Южный мост (англ. South Bridge), обеспечивающий работу с внешними устройствами.

Пропускная способность шины. Быстродействие процессора, оперативной памяти и периферийных устройств существенно различается. Быстродействие устройства зависит от тактовой частоты обработки данных (обычно измеряется в мегагерцах — МГц) и разрядности, т. е. количества битов данных, обрабатываемых за один такт. (Такт — это промежу-

ток времени между подачами электрических импульсов, синхронизирующих работу устройств компьютера.)

Соответственно, скорость передачи данных (пропускная способность) соединяющих эти устройства шин также должна различаться. Пропускная способность шины (измеряется в бит/с) равна произведению разрядности шины (измеряется в битах) и частоты шины (измеряется в герцах — Гц, 1 Гц = 1 такт в секунду):

$$\begin{aligned} \text{пропускная способность шины} &= \\ &= \text{разрядность шины} \times \text{частота шины}. \end{aligned}$$

Системная шина (см. рис. 1.12). Между Северным мостом и процессором данные передаются по системной шине (FSB от *англ.* FrontSide Bus). В наиболее быстрых компьютерах (2008 год) частота системной шины составляет 400 МГц. Однако между Северным мостом и процессором эффективная частота передачи данных в 4 раза выше. Таким образом, процессор может получать и передавать данные с частотой $400 \text{ МГц} \cdot 4 = 1600 \text{ МГц}$. Так как разрядность системной шины равна разрядности процессора и составляет 64 бита, то пропускная способность системной шины равна:

$$\begin{aligned} 64 \text{ бита} \cdot 1600 \text{ МГц} &= 102400 \text{ Мбит/с} = \\ &= 100 \text{ Гбит/с} = 12,5 \text{ Гбайт/с}. \end{aligned}$$

Частота процессора. В процессоре используется внутреннее умножение частоты, поэтому частота процессора в несколько раз больше, чем частота системной шины. Например, в современных процессорах используется коэффициент умножения частоты 8. Это означает, что процессор за один такт шины способен генерировать 8 своих внутренних тактов и, следовательно, частота процессора составляет $400 \text{ МГц} \cdot 8 = 3,2 \text{ ГГц}$.

Шина памяти (см. рис. 1.12). Обмен данными между северным мостом и оперативной памятью производится по шине памяти, частота которой может быть больше (например, в 4 раза), чем частота системной шины. У современных модулей памяти (DDR3 от *англ.* double-data-rate) частота шины памяти может составлять $400 \text{ МГц} \cdot 4 = 1600 \text{ МГц}$, т. е. оперативная память получает данные с такой же частотой, что и процессор. Так как разрядность шины памяти равна разрядности процессора и составляет 64 бита, то пропускная способность шины памяти также равна:

$$\begin{aligned} & 64 \text{ бита} \cdot 1600 \text{ МГц} = 102\,400 \text{ Мбит/с} = \\ & = 100 \text{ Гбит/с} = 12,5 \text{ Гбайт/с} = 12\,800 \text{ Мбайт/с}. \end{aligned}$$

i Модули памяти маркируются своей пропускной способностью, выраженной в Мбайт/с: PC4200, PC8500, PC12800 и др.

Шина PCI Express (см. рис. 1.12). По мере усложнения графики приложений требования к быстродействию шины, связывающей видеопамять с процессором и оперативной памятью, возрастают.

В настоящее время для подключения видеоплаты к северному мосту все большее распространение получает шина PCI Express (Peripheral Component Interconnect bus Express — ускоренная шина взаимодействия периферийных устройств). Пропускная способность этой шины может достигать 32 Гбайт/с.

К видеоплате с помощью аналогового разъема VGA (Video Graphics Array — графический видеоадаптер) или цифрового разъема DVI (Digital Visual Interface — цифровой видеоинтерфейс) подключается электронно-лучевой или жидкокристаллический монитор или проектор.

Шина SATA (см. рис. 1.12). Устройства внешней памяти (жесткие диски, CD- и DVD-дисководы) подключаются к южному мосту по шине SATA (англ. Serial Advanced Technology Attachment — последовательная шина подключения накопителей), скорость передачи данных по которой может достигать 300 Мбайт/с.

Шина USB (см. рис. 1.12). Для подключения принтеров, сканеров, цифровых камер и других периферийных устройств обычно используется шина USB (Universal Serial Bus — универсальная последовательная шина). Эта шина обладает пропускной способностью до 60 Мбайт/с и обеспечивает подключение к компьютеру одновременно до 127 периферийных устройств (принтер, сканер, цифровая камера, Web-камера, модем и др.).

Увеличение производительности процессора. Увеличение производительности процессоров за счет увеличения частоты имеет свой предел из-за тепловыделения. Выделение процессором теплоты Q пропорционально потребляемой мощности P , которая, в свою очередь, пропорциональна квадрату частоты v^2 :

$$Q \sim P \sim v^2.$$

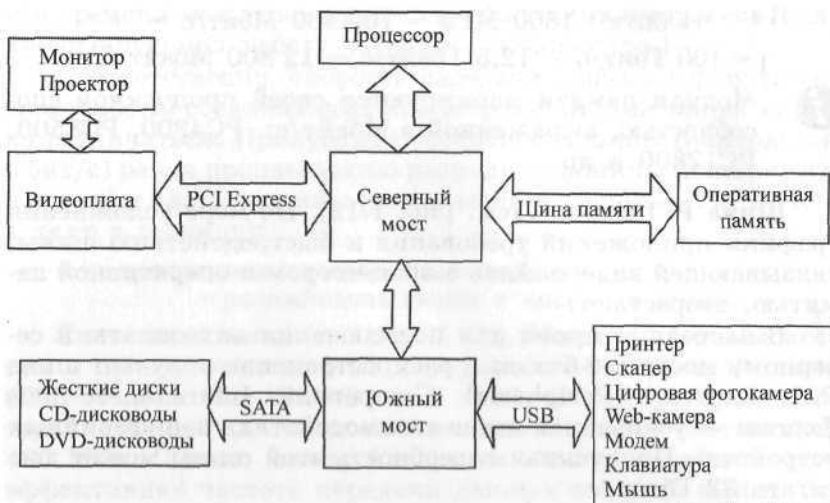


Рис. 1.12. Архитектура персонального компьютера

Уже в настоящее время для отвода тепла от процессора используются массивные воздушные кулеры, состоящие из вентилятора и металлических теплоотводящих ребер.

Увеличение производительности процессора, а значит и компьютера, достигается за счет увеличения количества ядер процессора (арифметических логических устройств). Вместо одного ядра процессора используются два или четыре ядра, что позволяет распараллелить вычисления и повысить производительность процессора.

Контрольные вопросы

1. В чем состоит магистрально-модульный принцип построения компьютера?
2. Какие устройства обмениваются информацией через Северный мост?
3. Какие устройства обмениваются информацией через Южный мост?
4. В каком направлении развивается архитектура процессоров?

Практическая работа 1.2

Сведения об архитектуре компьютера

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows.

Цель работы. Научиться получать сведения об архитектуре компьютера и отдельных его устройствах.

Задание. С помощью системы тестирования компьютера получить сведения об его архитектуре компьютера и процессора.

Варианты выполнения работы:

в операционной системе Windows с помощью программы тестирования компьютера SiSoftware Sandra получить сведения о разных параметрах архитектуры компьютера, используя различные тестирующие модули программы.



Получение сведений об архитектуре компьютера и процессора

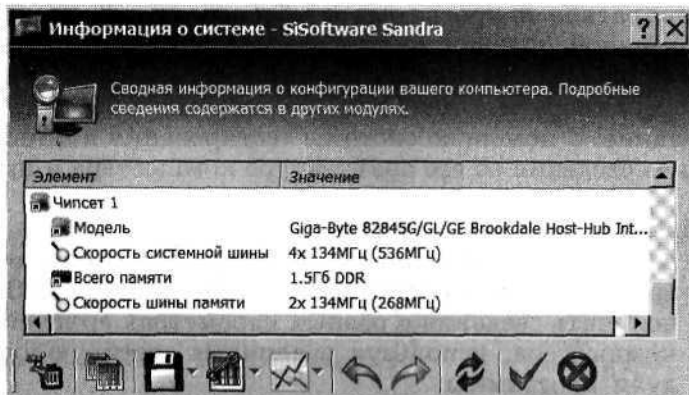
В операционной системе Windows получим сведения об архитектуре компьютера и процессора.

1. В операционной системе Windows запустить программу тестирования компьютера SiSoftware Sandra командой [Программы-SiSoftware-SiSoftware Sandra].

В появившемся диалоговом окне программы на вкладке *Устройства* выберем устройства, о которых хотелось бы получить сведения.



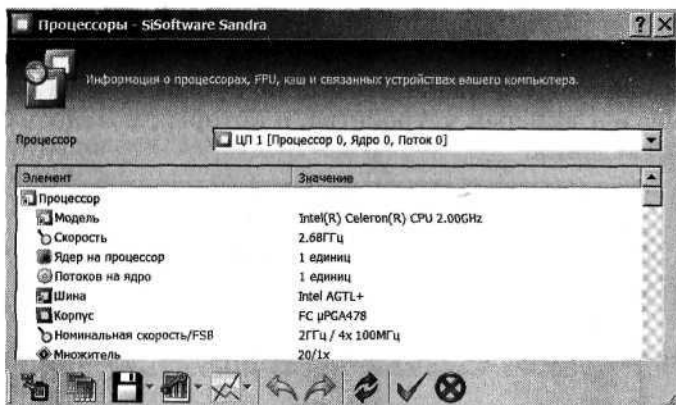
2. Выбрать пункт *Информация о системе*. В появившемся диалоговом окне с помощью полос прокрутки выбрать интересующие сведения.



Для данного компьютера получим:

- скорость системной шины — 134 МГц;
- эффективная скорость системной шины — 536 МГц;
- скорость шины памяти — 268 МГц.

3. Выбрать пункт *Процессоры*. В появившемся диалоговом окне с помощью полос прокрутки выбрать интересующие сведения.



Для данного компьютера получим:


- одно ядро;
- частота процессора — 2,68 ГГц;
- частота системной шины — 100 МГц;
- коэффициент умножения частоты процессора — 20.

1.3. Операционные системы

1.3.1. Основные характеристики операционных систем

Операционная система — это базовый комплекс компьютерных программ, обеспечивающий управление аппаратными средствами компьютера, работу с файловой системой, ввод и вывод данных с помощью периферийных устройств, а также выполнение прикладных программ. Важной частью современных операционных систем являются средства, обеспечивающие работу в локальных сетях и глобальной сети Интернет. Современные операционные системы предоставляют пользователю графический интерфейс.

При включении компьютера операционная система загружается в оперативную память раньше остальных программ и затем обеспечивает их выполнение.

 В настоящее время для настольных и портативных персональных компьютеров чаще всего используются операционные системы Microsoft Windows и Linux. На персональные компьютеры фирмы Apple устанавливается операционная система Mac OS. Для карманных компьютеров разработаны компактные операционные системы Windows Mobile и Palm OS.

Файловая система. В процессе работы компьютера происходит обмен файлами между устройствами. Разные операционные системы используют различные файловые системы, т. е. различные способы организации, хранения и именования данных на носителях информации. В операционной системе имеются программные модули, управляющие файловой системой.

Минимальным адресуемым элементом носителя информации является кластер, который может включать в себя несколько секторов (объем сектора составляет 512 байтов). Размер кластера может составлять от 512 байтов до 64 Кбайт. Кластеры нумеруются в линейной последовательности (на магнитных дисках от первого кластера нулевой дорожки до последнего кластера последней дорожки).

Файловая система организует кластеры в файлы и каталоги (каталог реально является файлом, содержащим список файлов в этом каталоге). Файловая система отслеживает, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.

Командный процессор. В состав операционной системы входит специальная программа — командный процессор, которая запрашивает у пользователя команды и выполняет их.

Пользователь может дать команду запуска программы, выполнения какой-либо операции над файлами (копирование, удаление, переименование), вывода документа на печать и т. д. Операционная система должна эту команду выполнить.

Драйверы устройств. К системной плате компьютера подключаются различные устройства (жесткие магнитные диски, оптические дисководы, монитор, клавиатура, мышь, принтер и др.). Каждое устройство выполняет определенную функцию (ввод информации, хранение информации, вывод информации), при этом техническая реализация устройств существенно различается.

Драйверы устройств — специальные программы, которые обеспечивают управление работой устройств и согласование информационного обмена с другими устройствами, а также позволяют производить настройку некоторых их параметров. Каждому типу устройств соответствует свой драйвер.

В процессе установки операционной системы определяются тип и конкретные модели установленных устройств и подключает необходимые для их функционирования драйверы. При включении компьютера производится загрузка драйверов в оперативную память. Пользователь имеет возможность вручную установить или переустановить драйверы.

Графический интерфейс. Для упрощения работы пользователя в состав современных операционных систем входят программные модули, создающие графический пользовательский интерфейс. В операционных системах с графическим интерфейсом пользователь может вводить команды с помощью диалоговых окон и элементов управления: кнопок, текстовых полей, списков, переключателей, флажков, счетчиков, ползунков и др.

2.5. Графический интерфейс операционных систем и приложений

Информатика и ИКТ-8 

Служебные программы. В состав операционной системы входят также служебные программы, которые позволяют обслуживать диски (проверять, сжимать, дефрагментировать и т. д.), выполнять операции с файлами (архивировать и т. д.), работать в компьютерных сетях и т. д.

Справочная система. Для удобства пользователя в состав операционной системы обычно входит также справочная система. Справочная система позволяет оперативно получить необходимую информацию как о функционировании операционной системы в целом, так и о работе ее отдельных модулей.

Загрузка операционной системы. Файлы операционной системы хранятся во внешней, долговременной памяти (на магнитных, оптических или USB флэш-дисках). Однако программы могут выполняться, только если они находятся в оперативной памяти, поэтому файлы операционной системы необходимо загрузить в оперативную память. Разрешение этого противоречия состоит в последовательной, поэтапной загрузке операционной системы.



Диск, на котором находятся файлы операционной системы и с которого производится ее загрузка, называется **системным**.

После включения компьютера производится его самотестирование. Затем наступает первый этап загрузки операционной системы — специальная программа, содержащаяся в микросхеме BIOS, начинает поиск загрузчика операционной системы. Современные версии BIOS позволяют загружать операционную систему не только с магнитных и оптических дисков, но и с USB флэш-дисков.

Если диск системный и программа-загрузчик оказывается на месте, то она загружается в оперативную память и ей передается управление работой компьютера. Программа ищет файлы операционной системы на системном диске и загружает их в оперативную память в качестве программных модулей.

Если системные диски в компьютере отсутствуют, на экране монитора появляется сообщение «Non system disk», и компьютер «зависает», т. е. загрузка операционной системы прекращается и компьютер остается неработоспособным.

Установка на компьютер различных операционных систем. На компьютер могут быть установлены одновременно несколько разных операционных систем. Для этого жесткий диск должен быть разбит на **логические разделы** (независимые области на диске), в каждом из которых может быть создана своя файловая система.

Таким образом, на компьютер в разные логические разделы могут быть установлены операционные системы семейства Microsoft Windows, Linux и Mac OS. Для перехода от одной операционной системы к другой необходимо перезагрузить компьютер.



В настоящее время все большее распространение получают **виртуальные машины**, которые позволяют устанавливать операционную систему в логический раздел, принадлежащий другой файловой системе. Это позволяет переходить от одной операционной системы к другой, не перезагружая компьютера.

Контрольные вопросы

1. Каков состав и функции операционных систем?
2. Как происходит загрузка операционной системы?

Практическая работа 1.3

Сведения о логических разделах дисков

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux.

Цель работы. Научиться получать сведения о логических разделах дисков.

Задание. В операционных системах Windows и Linux получить сведения о логических разделах дисков.

Варианты выполнения работы:

в операционной системе Windows или в операционной системе Linux.



Получение сведений о логических разделах дисков



В операционной системе Windows получим сведения о логических разделах дисков.

1. В операционной системе Windows запустить служебную программу управления дисками командой [*Панель управления-Администрирование-Управление компьютером*].

2. В левой части появившегося диалогового окна *Управление компьютером* выбрать пункт *Управление дисками*.
В графической форме будут показаны логические разделы дисков с информацией об их объемах и файловых системах.

Диск 0 Основной 76,33 Гб Подключен	Windows Vista 34,10 Гб NTFS Исправен (Систе		AltLinux (D: 4,88 Гб NTFS Исправен (Лк		19,75 Гб Исправен (Осно	1Multimedia (I 10,31 Гб NTFS Исправен (Лог)	1Documents 7,28 Гб NTFS Исправен (Лог)	
	2Windows 1,05 Гб N Исправен		2Turbot 700 Мб N Исправен		2Original (R: 10,05 Гб NTFS Исправен (Ло		2Micros 706 Мб N Исправен	
	2Model- 800 Мб I Исправе		5,81 Гб Исправен (Ос					
Диск 1 Основной 19,08 Гб Подключен								

В данном компьютере Диск 0 поделен на пять логических разделов, а Диск 1 — на 6 логических разделов. Важно отметить, что два логических раздела без названия принадлежат другой операционной системе (в данном случае Linux).

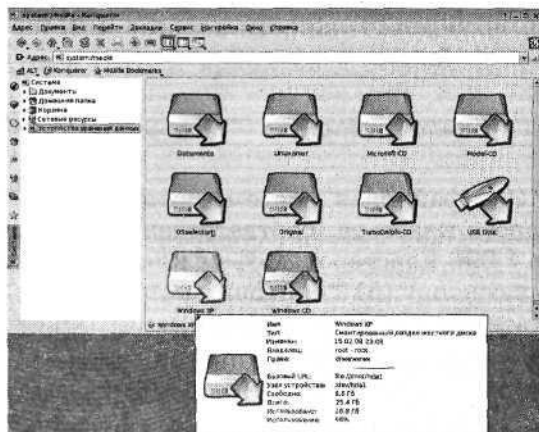
В операционной системе Linux получим сведения о логических разделах дисков.

3. В операционной системе Linux запустить файловый менеджер Konqueror.

4. В левой части появившегося диалогового окна выбрать команду [Система-Устройства хранения].

В правой части будут показаны значки логических разделов дисков.

При наведении курсора на значок логического раздела будет появляться окно с информацией о его объеме, имени и т. д.




В данном компьютере жесткий диск поделен на девять логических разделов.

Операционная система Windows XP установлена в логический раздел hda1, а операционная система Linux — в раздел hda12.

USB флэш-диск получил логическое имя sda1.

1.3.2. Операционная система Windows

 Операционная система Windows остается наиболее популярной операционной системой для персональных компьютеров. Эта операционная система установлена на большинстве персональных компьютеров в мире.

Операционные системы корпорации Microsoft, ориентированные на настольные и портативные компьютеры, включают семейство Windows 95/98/Me (для устаревших компьютеров) и семейство Windows 2000/XP/Vista (для современных компьютеров), а также семейство Windows Mobile для карманных компьютеров.

Файловые системы. Полная информация о кластерах, которые занимают файлы, содержится в таблице размещения файлов FAT. В целях более надежного сохранения информации о размещении файлов на носителе информации хранятся две идентичные копии таблицы FAT. Количество ячеек таблицы FAT соответствует количеству кластеров на диске, а значениями ячеек являются цепочки размещения файлов, т. е. последовательности адресов кластеров, в которых хранятся файлы.

2.3. Файлы и файловая система Информатика и ИКТ-8

FAT12. Эта файловая система выделяет 12 битов для хранения адреса кластера, соответственно, она может адресовать $2^{12} = 4\,096$ кластеров. Объем кластера по умолчанию равен одному сектору (512 байтов), поэтому FAT12 не может использоваться для носителей информации объемом более:

$$\begin{aligned} 512 \text{ байтов} \cdot 4\,096 &= 2\,097\,152 \text{ байта} = \\ &= 2\,048 \text{ Кбайт} = 2 \text{ Мбайт.} \end{aligned}$$

Поэтому FAT12 используется для дискет.

FAT16. Эта файловая система выделяет 16 битов для хранения адреса кластера, соответственно, она может адресовать $2^{16} = 65\,536$ кластеров. Объем кластера не может быть более 128 секторов (64 Кбайт), поэтому FAT16 не может использоваться для носителей информации объемом более:

$$\begin{aligned} 64 \text{ Кбайт} \cdot 65\,536 &= 4\,194\,304 \text{ Кбайт} = \\ &= 4\,096 \text{ Мбайт} = 4 \text{ Гбайт}. \end{aligned}$$

В настоящее время такой объем имеет флэш-память, поэтому FAT16 используется именно для флэш-памяти.

FAT32. Эта файловая система выделяет 32 бита для хранения адреса кластера, соответственно, она может адресовать $2^{32} = 4\,294\,967\,296$ кластеров. Объем кластера по умолчанию составляет 8 секторов (4 Кбайт), поэтому FAT32 может использоваться для носителей информации объемом:

$$\begin{aligned} 4 \text{ Кбайт} \cdot 4\,294\,967\,296 &= 17\,179\,869\,184 \text{ Кбайт} = \\ &= 16\,777\,216 \text{ Мбайт} = 16\,384 \text{ Гбайт} = 16 \text{ Тбайт}. \end{aligned}$$

Таким образом, FAT32 может использоваться для жестких дисков самого большого объема.

NTFS. Эта файловая система позволяет устанавливать различные объемы кластера (от 512 байтов до 64 Кбайт, по умолчанию 4 Кбайт). NTFS по сравнению с FAT32 увеличивает надежность и эффективность использования дискового пространства.

NTFS использует систему журналирования для повышения надежности файловой системы. Журналируемая файловая система хранит список изменений, которые она будет проводить с файловой системой перед фактической записью изменений. Эти записи хранятся в отдельной части файловой системы, называемой «журналом» или «логом». Как только изменения файловой системы будут внесены в журнал, журналируемая файловая система применит эти изменения к файлам.

Графический интерфейс. *Рабочий стол* — это основа и вершина графического интерфейса операционной системы. Он появляется после включения компьютера и входа в операционную систему Windows. Подобно поверхности обычного стола, он служит рабочей поверхностью, на которой появляются диалоговые окна запущенных приложений и открытых папок.

Панель задач располагается у нижнего края экрана. На ней отображаются значки запущенных приложений, позволяющие переключаться между ними. Она также содержит слева кнопку *Пуск*, которой можно воспользоваться для доступа к программам, папкам и параметрам компьютера, а справа располагаются индикаторы раскладки клавиатуры, сетевых подключений и часы.

На *Рабочем столе* располагаются значки — это маленькие рисунки, обозначающие программы, файлы, папки и другие объекты. При первом запуске Windows на *Рабочем столе* имеется, как минимум, один значок: *Корзина*. Пользователь компьютера может добавить на *Рабочий стол* и другие значки. Двойной щелчок по значку на *Рабочем столе* запускает или открывает объект, который значок представляет.

Для упрощения доступа к ресурсам компьютера с *Рабочего стола* можно создать *ярлыки* избранных файлов и приложений. **Ярлык** — это значок, представляющий ссылку на объект, а не сам объект (ярлык можно узнать по стрелке на значке). Двойной щелчок по ярлыку открывает объект. При удалении ярлыка удаляется только ярлык, но не исходный объект. Многие приложения при установке создают ярлыки на *Рабочем столе* автоматически.

Открываемые приложения, файлы или папки появляются на *Рабочем столе* в окнах (именно от них операционная система Windows получила свое название). Хотя содержимое каждого окна индивидуально, у всех окон есть одни и те же основные элементы:











- *Строка заголовка*. Отображает название документа, программы или папки.
- *Кнопки сворачивания, разворачивания и закрытия окна*. С помощью этих кнопок можно, соответственно, скрыть окно, развернуть на весь экран и закрыть его (более подробно об этом немного далее).
- *Строка меню*. Содержит команды, выбираемые щелчком мышью.
- *Полоса прокрутки*. Позволяет прокручивать содержимое окна для просмотра информации.
- *Границы и углы*. Можно перетаскивать их указателем мыши, изменяя размер окна.
- *Меню, кнопки, полосы прокрутки, флажки* и другие **элементы управления** позволяют выбирать команды, изменять параметры и работать с окнами.

2.5. Графический интерфейс операционных систем и приложений

Информатика и ИКТ-8 

Безопасность компьютера. Центр обеспечения безопасности Windows показывает текущее состояние защиты компьютера и рекомендует меры по усилению безопасности. Центр обеспечения безопасности проверяет следующие основные элементы защиты компьютера (рис. 1.13).

- *Брандмауэр (межсетевой экран).* Брандмауэр защищает компьютер, предотвращая доступ к нему хакеров и вредоносных программ.
- *Автоматическое обновление.* Windows может регулярно проверять наличие обновлений и автоматически их устанавливать.
- *Защита от вредоносных программ.* Антивирусное программное обеспечение (не входит в состав операционной системы) должно быть установлено, это поможет защитить компьютер от вирусов, червей и других угроз безопасности.
- *Другие параметры безопасности.* Центр обеспечения безопасности проверяет параметры безопасности браузера Internet Explorer.
- *Контроль учетных записей пользователей.* Он позволяет предотвратить несанкционированные изменения в компьютере. Контроль учетных записей обеспечивает защиту, запрашивая разрешение перед совершением потенциально опасных для компьютера действий.

Брандмауэр	Включено  
Автоматическое обновление	Включено  
Защита от вредоносных программ	Включено  
Другие параметры безопасности	Проверить параметры  
Параметры безопасности Интернета	ОК 
Все параметры безопасности Интернета настроены на рекомендуемый уровень. Как параметры безопасности Интернета помогают защитить компьютер?	
Контроль учетных записей пользователей	Выкл 
Контроль учетных записей отключен. Как контроль учётных записей помогает защитить компьютер?	
	Включить сейчас

1.13. Центр обеспечения безопасности Windows

Появившееся сообщение следует внимательно прочитать, проконтролировать, соответствует ли название выполняемого действия тому, которое действительно производится. Проверка этих действий перед запуском, контроль учетных записей помогает предотвратить установку вредоносного программного обеспечения и шпионских программ.

Системный реестр Windows. Конфигурация системы осуществляется с помощью системного реестра ОС Windows. Системный реестр операционной системы Windows является иерархической базой данных, в которой хранится информация о конфигурации Windows. В реестре содержатся сведения об оборудовании системы, установленных программах и параметрах настройки, к которым операционная система постоянно обращается во время работы.

Контрольные вопросы

1. Какие файловые системы используются в операционной системе Windows?
2. Какие элементы входят в графический интерфейс операционной системы Windows?
3. Как обеспечивается безопасность в операционной системе Windows?

Практическая работа 1.4

Значки и ярлыки на *Рабочем столе*

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux.

Цель работы. Научиться устанавливать нужные значки и ярлыки на *Рабочем столе*.

Задание 1. В операционной системе Windows установить нужные значки на *Рабочем столе*.

Задание 2. В операционной системе Windows поместить на *Рабочий стол* ярлык часто используемого приложения, папки или устройства (например, ярлык принтера).

Варианты выполнения работы:

- выбрать разные наборы значков на *Рабочем столе*;
- выбрать создание ярлыков различных приложений, папок и устройств.

1.3.3. Операционная система Linux

Операционная система Linux — ядро и основные компоненты системы, а также большинство пользовательских приложений для Linux — свободно распространяемые программы. Это означает, что их можно запускать на любом количестве компьютеров, без ограничений распространять, получать исходные тексты этих программ и вносить в них любые исправления. Свобода использования программ обеспечила их широкое применение и интерес к ним со стороны тысяч разработчиков и пользователей.

Операционная система Linux обладает высокой надежностью и имеет отличную систему защиты. Благодаря доступности сети Интернет и открытости исходных текстов программ, сообщить об ошибке и принять участие в ее исправлении независимому программисту или даже пользователю так же просто, как и специалисту фирмы-разработчика или автору проекта. Именно поэтому ошибки защиты выявляются особенно эффективно и быстро исправляются.

Файловая система. Операционные системы хранят данные на диске при помощи файловых систем. Классическая иерархическая файловая система представляет собой вложенные друг в друга папки, в которых могут содержаться и файлы. Одна из папок является вершиной иерархической файловой системы (а выражаясь технически — корнем), в ней содержатся все остальные папки и файлы. Если жесткий диск разбит на разделы, то в каждом разделе организуется отдельная файловая система с собственным корнем и структурой папок, т. е. корневых папок столько, сколько носителей информации и логических разделов.

В операционной системе Linux корневая папка всегда только одна, а все остальные папки в нее вложены, т. е. для пользователя файловая система представляет собой единое целое. В действительности, разные части файловой системы могут находиться на совершенно разных устройствах: разных разделах жесткого диска, на разнообразных съемных носителях (оптических дисках, дискетах, флэш-картах), даже на других компьютерах (с доступом через сеть). Для того, чтобы получить единое файловое дерево с одним корнем, используется процедура **монтирования**.

Монтирование — это подключение в одну из папок целой файловой системы, находящейся где-то на другом устройстве. Для монтирования необходима пустая папка —

она называется **точкой монтирования**. Подключенную (смонтированную) файловую систему можно в любой момент отключить — **размонтировать**, после чего та папка, куда она была смонтирована, снова окажется пустой.



Рис. 1.14. Смонтированные файловые системы логических разделов (показана смонтированная в папку hda1 файловая система логического раздела, куда установлена операционная система Windows Vista)

Пользователю обычно не требуется выполнять монтирование и размонтирование вручную: при загрузке системы будут смонтированы все устройства, на которых хранятся части файловой системы, а при останове (перед выключением) системы все они будут размонтированы. Файловые системы на съемных носителях (оптических дисках, дискетах и пр.) также монтируются и размонтируются автоматически — либо при подключении носителя, либо при обращении к соответствующему каталогу.

В операционной системе Linux чаще всего используется журналируемая файловая система ext3, в которой кластер может иметь размер от 1 до 8 Кбайт.

Дистрибутивы операционной системы Linux. Большинство пользователей для установки Linux используют дистрибутивы. Дистрибутив — это не просто набор программ, а ряд решений для разных задач пользователей, объединенных едиными системами установки, обновления и удаления программных пакетов.

Для установки, удаления и обновления программ и поддержания целостности системы в Linux используются **менеджеры пакетов**. С точки зрения менеджера пакетов программное обеспечение представляет собой набор компонентов — **программных пакетов**. Такие компоненты содержат в себе набор исполняемых программ и вспомогательных файлов, необходимых для корректной работы программного обеспечения.

Стабильные дистрибутивы компании ALT Linux, выпускаются на основе репозитория Sisyphus. Sisyphus — ежедневно обновляемый репозиторий пакетов, позволяет пользователям легко обновлять свои системы и быть в курсе всех новостей мира свободных программ.

**Дистрибутивы Linux
от компании AltLinux**

<http://altlinux.ru> 



Sisyphus (Сизиф) — персонаж греческой мифологии. Миф о Сизифе, который непрерывно катил в гору камень, символизирует постоянный труд команды по усовершенствованию решений, заложенных в репозиторий.

Школьный дистрибутив операционной системы Linux. ALT Linux Junior содержит дистрибутив операционной системы, а также необходимые приложения для преподавания курса «Информатика и ИКТ». В состав ALT Linux Junior входят:

- полнофункциональный офисный пакет OpenOffice;
- приложения для просмотра Web-страниц и общения в сети Интернет;
- редакторы векторной и растровой графики;
- приложения для обработки цифровых фотографий и нелинейного монтажа цифрового видео;
- звуковые редакторы для прослушивания и редактирования аудиофайлов;
- современные системы алгоритмического и объектно-ориентированного программирования.

Вход в систему. Операционная система Linux — это многопользовательская система. На практике это означает, что для работы в системе нужно в ней **зарегистрироваться**, т. е. ввести **системное имя** (login name) и **пароль** (password). Загрузка операционной системы заканчивается интерфейсом

входа в систему: выводится приглашение ввести системное имя пользователя и пароль.

В операционной системе Linux у каждого пользователя обязательно есть свой домашний каталог, предназначенный для хранения всех собственных данных пользователя. Именно с этого каталога пользователь начинает работу после регистрации в системе. Домашние каталоги пользователей обычно собраны в каталоге /home, их название чаще всего совпадает с учетным именем пользователя в системе, например, для пользователя test домашним каталогом будет /home/test.

Пользователь является полным хозяином внутри своего каталога, однако остальная часть файловой системы доступна ему только для чтения, но не для записи. Такая система разграничения доступа к файлам позволяет не бояться вирусов.

Графический интерфейс. Графический интерфейс операционной системы Linux можно настраивать. Один из вариантов — это использование графической оболочки KDE (от англ. K Desktop Environment), которая позволяет создать графический интерфейс, очень похожий на графический интерфейс операционной системы Windows. Графический интерфейс базируется на *Рабочем столе*, есть *Панель задач* и слева кнопка *Пуск*, открывающая доступ к меню запуска приложений. Справа на *Панели задач* располагаются индикаторы раскладки клавиатуры, сетевых подключений и часы.

Приложения и папки открываются на *Рабочем столе* в диалоговых окнах, содержащих элементы управления: *меню, кнопки, полосы прокрутки, флажки, переключатели* и т. д.

Выключение компьютера. Компьютер с загруженной операционной системой Linux нельзя выключать, просто лишив компьютер электропитания. Большое количество информации, которая должна располагаться на диске, система держит в оперативной памяти для повышения быстродействия. Неожиданное выключение питания приводит к потере этой информации. Если вы работаете в графической среде, то для завершения работы нужно выбрать соответствующий пункт в главном меню. Если перед вами графический интерфейс входа в систему, то там также есть кнопка меню, в котором имеется пункт «выключить компьютер».

Контрольные вопросы

1. Какая файловая система используется в операционной системе Linux?
2. В чем состоит смысл монтирования и размонтирования в операционной системе Linux?
3. Какие элементы входят в графический интерфейс операционной системы Linux?

Практическая работа 1.5

Настройка графического интерфейса для операционной системы Linux

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Linux.

Цель работы. Научиться настраивать графический интерфейс операционной системы Linux.

Задание. В операционной системе Linux осуществить настройку графического интерфейса в среде KDE.

Варианты выполнения работы:

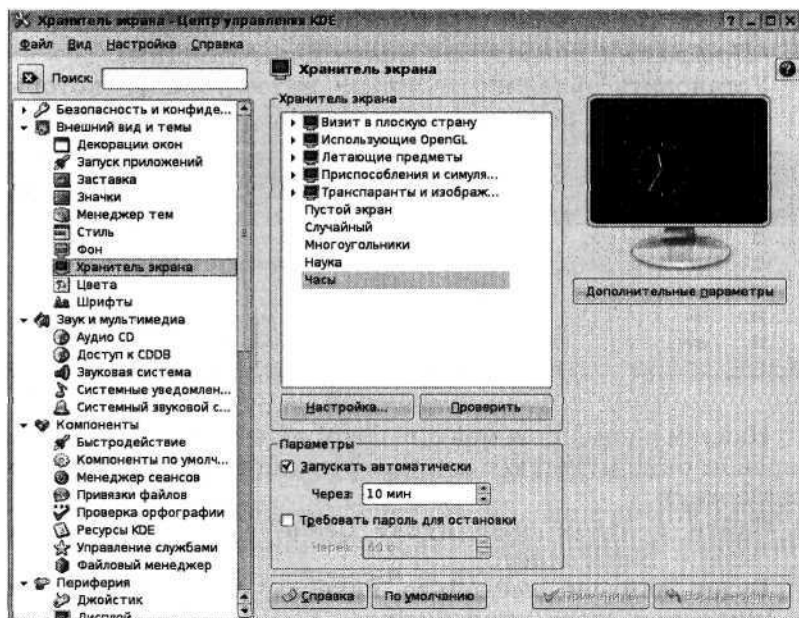
настроить различные параметры графического интерфейса.

Настройка графического интерфейса в среде KDE



В операционной системе Linux осуществим настройку графического интерфейса в среде KDE.

1. В операционной системе Linux в меню выбрать пункт *Центр управления KDE*.
2. В левой части появившегося диалогового окна *Центр управления KDE* активизировать ссылку (например, *Хранитель экрана*).
3. В появившемся справа диалоговом окне *Хранитель экрана* установить нужные параметры. Выбрать из списка заставку (например, *Часы*) и с помощью счетчика установить, через сколько минут бездействия она появляется (например, *10 мин.*).



Практическая работа 1.6

Установка пакетов в операционной системе Linux

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Linux.

Цель работы. Научиться устанавливать программные пакеты приложений в операционной системе Linux.

Задание. В операционной системе Linux установить браузер SeaMonkey, с использованием репозитория.

Варианты выполнения работы:

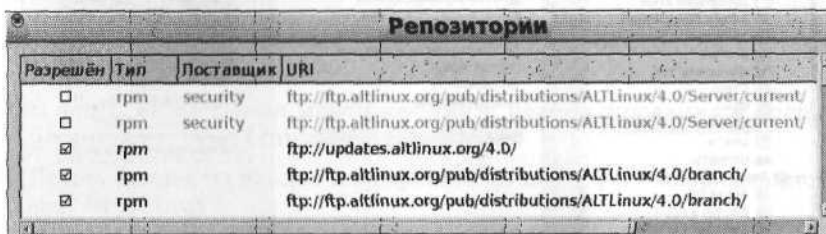
выбрать различные комбинации пакетов (для различных комбинаций пакетов изменяется внешний вид и функциональные возможности браузера).

Задание. Установка программных пакетов приложений в операционной системе Linux



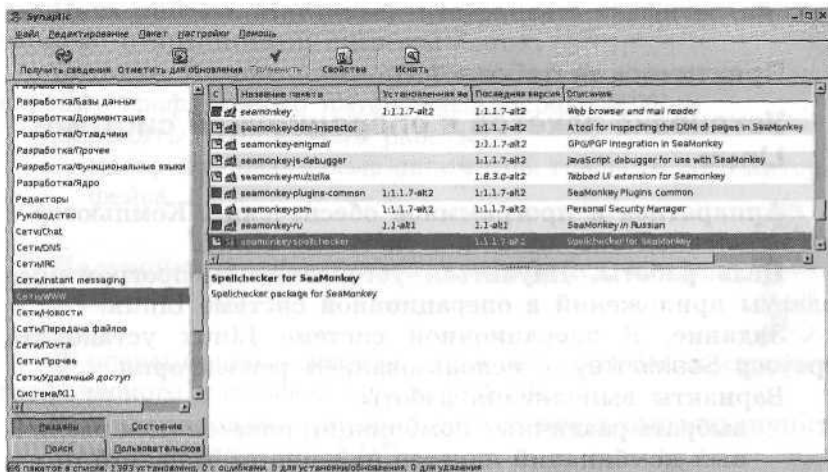
Определим репозитории, в которых могут быть пакеты требуемого приложения.

1. Ввести команду [*Настройка-Менеджер пакетов (Программа управления пакетами Synaptic)*].
Установить флажки нужных репозиториях, которые определить с помощью команды [*Получить сведения*].

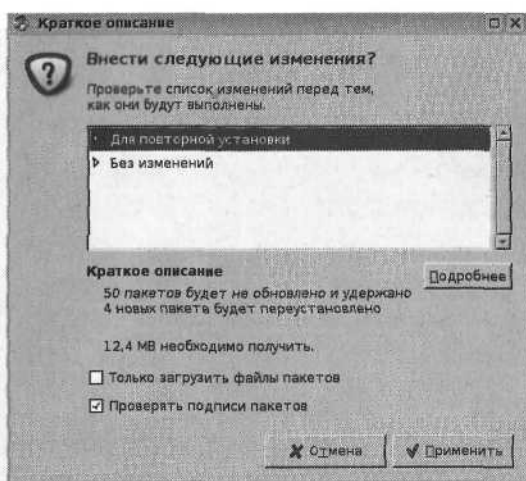


Найдем с помощью меню в левом окне и перечне пакетов в правом окне нужную программу (в данном случае браузер SeaMonkey).

2. Найти в левом окне пункт *Сети/WWW*, а в правом окне — *SeaMonkey*.
Будет найдено несколько пакетов, необходимых для установки этого приложения.



3. В контекстном меню выбранных пакетов активизировать пункт *Отметить для установки*.
В меню программы *Установка пакетов* выбрать команду [*Применить*].
В появившемся диалоговом окне выбрать пункт *Для установки* и щелкнуть по кнопке *Применить*.



Пакеты (приложение) будут установлены вместе с требуемыми для функционирования приложения пакетами. Программа автоматически получит пункт в меню.

1.4. Защита от несанкционированного доступа к информации

1.4.1. Защита с использованием паролей

Для защиты от несанкционированного доступа к программам и данным, хранящимся на компьютере, используются **пароли**. Компьютер разрешает доступ к своим ресурсам только тем пользователям, которые зарегистрированы и ввели правильный пароль. Каждому конкретному пользователю может быть разрешен доступ только к определенным информационным ресурсам. При этом может производиться регистрация всех попыток несанкционированного доступа.

Вход по паролю может быть установлен в программе BIOS Setup, компьютер не начнет загрузку операционной системы, если не введен правильный пароль (рис. 1.15). Преодолеть такую защиту нелегко, более того, возникнут серьезные проблемы доступа к данным, если пользователь забудет этот пароль.

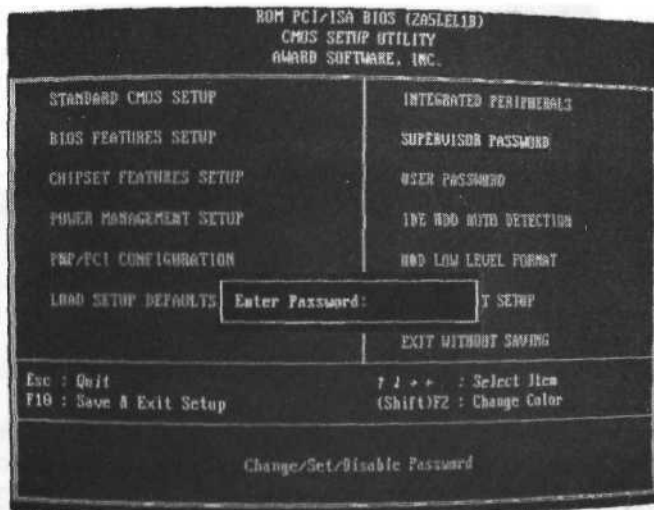


Рис. 1.15. Вход по паролю в BIOS Setup

Защита с использованием пароля используется при загрузке операционной системы (при загрузке системы каждый пользователь должен ввести свой пароль) (рис. 1.16).



Рис. 1.16-1. Ввод пароля в операционной системе Windows



Рис. 1.16-2. Ввод пароля в операционной системе Linux

От несанкционированного доступа может быть защищен каждый диск, папка или файл локального компьютера. Для них могут быть установлены определенные права доступа (полный доступ, изменение, чтение, запись и др.), причем права могут быть различными для различных пользователей (рис. 1.17).

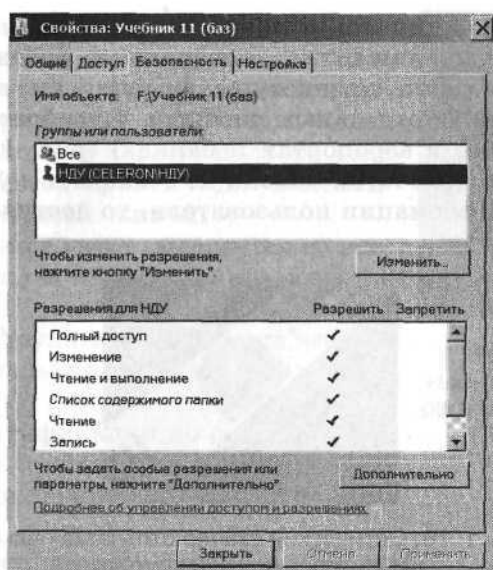


Рис. 1.17. Установка прав доступа к папке в операционной системе Windows

Контрольные вопросы

1. Как защищается информация в компьютере с использованием паролей?

1.4.2. Биометрические системы защиты

В настоящее время для защиты от несанкционированного доступа к информации все более часто используются **биометрические системы идентификации**. Используемые в этих системах характеристики являются неотъемлемыми качествами личности человека и поэтому не могут быть утраченными и подделанными. К биометрическим системам защиты информации относятся системы идентификации:

- по отпечаткам пальцев;
- по характеристикам речи;
- по радужной оболочке глаза;
- по изображению лица;
- по геометрии ладони руки.

Идентификация по отпечаткам пальцев. Оптические сканеры считывания отпечатков пальцев устанавливаются на ноутбуки, мыши, клавиатуры, флэш-диски, а также применяются в виде отдельных внешних устройств и терминалов (например, в аэропортах и банках) (рис. 1.18).

Если узор отпечатка пальца не совпадает с узором допущенного к информации пользователя, то доступ к информации невозможен.



Рис. 1.18. Оптический сканер отпечатка пальца, вмонтированный в ноутбук

Идентификация по характеристикам речи. Идентификация человека по голосу — один из традиционных способов распознавания, интерес к этому методу связан и с прогнозами внедрения голосовых интерфейсов в операционные системы. Можно легко узнать собеседника по телефону, не видя его. Также можно определить психологическое состояние по эмоциональной окраске голоса. Голосовая идентификация бесконтактна и существуют системы ограничения доступа к информации на основании частотного анализа речи (рис. 1.19).

Каждому человеку присуща индивидуальная частотная характеристика каждого звука (фонемы).

В романе А. И. Солженицына «В круге первом» описана голосовая идентификация человека еще в 40-е годы прошлого века.

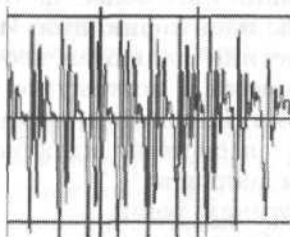


Рис. 1.19. Голосовая идентификация по частотной характеристике речи

Идентификация по радужной оболочке глаза. Радужная оболочка глаза является уникальной для каждого человека биометрической характеристикой. Она формируется в первые полтора года жизни и остается практически неизменной в течение всей жизни.

Изображение глаза выделяется из изображения лица и на него накладывается специальная маска штрих-кодов (рис. 1.20). Результатом является матрица, индивидуальная для каждого человека.

Для идентификации по радужной оболочке глаза применяются специальные сканеры, подключенные к компьютеру.

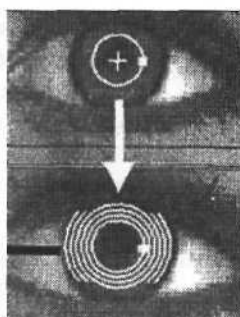


Рис. 1.20.
Идентификация по радужной оболочке глаза

Идентификация по изображению лица. Для идентификации личности часто используются технологии распознавания по лицу. Они ненавязчивы, так как распознавание человека происходит на расстоянии, без задержек и отвлечения внимания и не ограничивают пользователя в свободе перемещений.

По лицу человека можно узнать его историю, симпатии и антипатии, болезни, эмоциональное состояние, чувства и намерения по отношению к окружающим. Всё это представляет особый интерес для автоматического распознавания лиц (например, для выявления потенциальных преступников).

Идентификационные признаки учитывают форму лица, его цвет, а также цвет волос. К важным признакам можно отнести также координаты точек лица в местах, соответствующих смене контраста (брови, глаза, нос, уши, рот и овал).

В настоящее время начинается выдача новых загранпаспортов, в микросхеме которых хранится цифровая фотография владельца.

Идентификация по ладони руки. В биометрике в целях идентификации используется простая геометрия руки — размеры и форма, а также некоторые информационные знаки на тыльной стороне руки (образы на сгибах между фалангами пальцев, узоры расположения кровеносных сосудов).

Сканеры идентификации по ладони руки установлены в некоторых аэропортах, банках и на атомных электростанциях (рис. 1.21).



Рис. 1.21. Идентификация по ладони руки

Контрольные вопросы

1. Какие существуют биометрические методы защиты информации?

Практическая работа 1.7

Биометрическая защита: идентификация по характеристикам речи

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux с подключенными наушниками и микрофоном.

Цель работы. Научиться идентифицировать человека по частотной характеристике его речи.

Задание. В операционной системе Windows или Linux с помощью звукового редактора Audacity записать одно и то же слово (например, «информатика»), произнесенное несколькими людьми. По частотной характеристике речи попробовать идентифицировать этих людей.

Варианты выполнения работы:

- в операционной системе Windows или в операционной системе Linux;
- записать и проанализировать частотные характеристики речи разных людей.



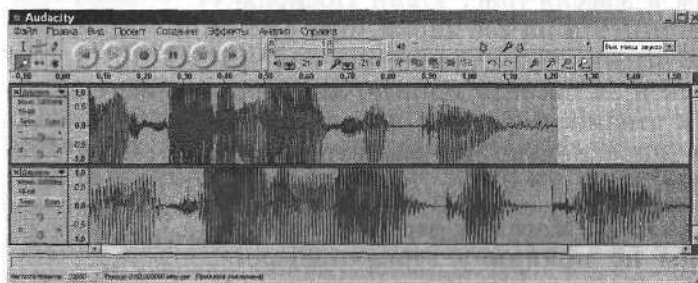
Запись с помощью звукового редактора Audacity одного и того же слова, произнесенного несколькими людьми.



Идентификация этих людей по частотной характеристике речи



1. В операционной системе Windows или Linux запустить звуковой редактор Audacity.
2. Записать одно и то же слово (например, «информатика»), произнесенное несколькими людьми.
3. Проанализировать частотные характеристики записей и попробовать идентифицировать этих людей по темпу речи и характерным особенностям произношения звуков (фоном).



1.5. Физическая защита данных на дисках

Для обеспечения большей скорости чтения/записи и надежности хранения данных на жестких дисках используются RAID-массивы (Redundant Arrays of Independent Disks — избыточный массив независимых дисков). Несколько жестких дисков подключаются к RAID-контроллеру, который рассматривает их как единый логический носитель информации.

Существует два способа реализации RAID-массива: аппаратный и программный. Аппаратный дисковый массив состоит из нескольких жестких дисков, управляемых при помощи специальной платы контроллера RAID-массива. Программный RAID-массив реализуется при помощи специального драйвера. В программный массив организуются дисковые разделы, которые могут занимать как весь диск, так

и его часть. Программные RAID-массивы, как правило, менее надежны, чем аппаратные, но обеспечивают более высокую скорость работы с данными.

Существует несколько разновидностей RAID-массивов, так называемых уровней. Операционные системы поддерживают несколько уровней RAID-массивов.

RAID 0. Для создания массива этого уровня понадобится как минимум два диска одинакового размера. Запись осуществляется по принципу чередования: данные делятся на порции одинакового размера (A1, A2, A3 и т. д.), и поочередно распределяются по всем дискам, входящим в массив (рис. 1.22). Поскольку запись ведется на все диски, при отказе одного из них будут утрачены все хранившиеся на массиве данные, однако запись и чтение на разных дисках происходит параллельно и, соответственно, быстрее.

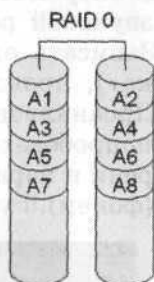


Рис. 1.22. Массив жестких дисков уровня RAID 0

RAID 1. Массивы этого уровня построены по принципу зеркалирования, при котором все порции данных (A1, A2, A3 и т. д.), записанные на одном диске, дублируются на другом (рис. 1.23). Для создания такого массива потребуются два или более дисков одинакового размера. Избыточность обеспечивает отказоустойчивость массива: в случае выхода из строя одного из дисков, данные на другом остаются неповрежденными. Расплата за надежность — фактическое сокращение дискового пространства вдвое. Скорость чтения и записи остается на уровне обычного жесткого диска.

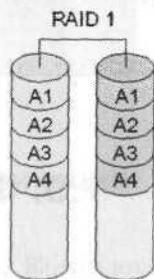


Рис. 1.23. Массив жестких дисков уровня RAID 1

Контрольные вопросы

1. Какие существуют массивы дисков RAID?

1.6. Защита от вредоносных программ

1.6.1. Вредоносные и антивирусные программы

Типы вредоносных программ. Вредоносными программами являются программы, наносящие вред данным и программам, хранящимся на компьютере. Основными типами вредоносных программ являются:

- **Вирусы, черви, троянские и хакерские программы.** Эта группа объединяет наиболее распространенные и опасные категории вредоносных программ. Защита от них обеспечивает минимально допустимый уровень безопасности.
- **Шпионское, рекламное программное обеспечение, программы скрытого дозвона.** Данная группа объединяет в себе потенциально опасное программное обеспечение, которое может причинить неудобство пользователю или даже нанести значительный ущерб.
- **Потенциально опасное программное обеспечение.** Эта группа включает программы, которые не являются вредоносными или опасными, однако при некотором стечении обстоятельств могут быть использованы для нанесения вреда вашему компьютеру.



За создание, использование и распространение вредоносных программ в России и большинстве стран предусмотрена уголовная ответственность.

Антивирусные программы. Современные антивирусные программы обеспечивают комплексную защиту программ и данных на компьютере от всех типов вредоносных программ и методов их проникновения на компьютер (Интернет, локальная сеть, электронная почта и съемные носители информации). Для защиты от вредоносных программ каждого типа в антивирусе предусмотрены отдельные компоненты.

Принцип работы антивирусных программ основан на проверке файлов, загрузочных секторов дисков и оперативной памяти и поиске в них известных и новых вредоносных программ.

Для поиска известных вредоносных программ используются сигнатуры. Сигнатура — это некоторая постоянная последовательность программного кода, специфичная для конкретной вредоносной программы. Если антивирусная программа обнаружит такую последовательность в каком-

либо файле, то файл считается зараженным вирусом и подлежит лечению или удалению.

Для поиска новых вирусов используются алгоритмы эвристического сканирования, т. е. анализа последовательности команд в проверяемом объекте. Если «подозрительная» последовательность команд обнаруживается, то антивирусная программа выдает сообщение о возможном заражении объекта.

Большинство антивирусных программ сочетает в себе функции постоянной защиты (антивирусный монитор) и функции защиты по требованию пользователя (антивирусный сканер).

Антивирусный монитор запускается автоматически при старте операционной системы и работает в качестве фонового системного процесса, проверяя на вредоносность совершаемые другими программами действия. Основная задача антивирусного монитора состоит в обеспечении максимальной защиты от вредоносных программ при минимальном замедлении работы компьютера.

Антивирусный сканер запускается по заранее выбранному расписанию или в произвольный момент пользователем. Антивирусный сканер производит поиск вредоносных программ в оперативной памяти, а также на жестких и сетевых дисках компьютера.

К недостаткам антивирусных программ можно отнести большие размеры используемых ими антивирусных баз данных, которые должны содержать информацию о максимально возможном количестве вирусов (в настоящее время десятках тысяч), что, в свою очередь, приводит к относительно небольшой скорости поиска вирусов.

Признаки заражения компьютера. Есть ряд признаков, свидетельствующих о проникновении на компьютер вредоносных программ:

- вывод на экран непредусмотренных сообщений или изображений;
- подача непредусмотренных звуковых сигналов;
- неожиданное открытие и закрытие лотка CD/DVD дисководов;
- произвольный запуск на компьютере каких-либо программ;
- частые зависания и сбои в работе компьютера;
- медленная работа компьютера при запуске программ;
- исчезновение или изменение файлов и папок;

- частое обращение к жесткому диску (часто мигает лампочка на системном блоке);
- зависание или неожиданное поведение браузера (например, окно программы невозможно закрыть).

Кроме того, есть некоторые характерные признаки поражения сетевым вирусом через электронную почту:

- друзья или знакомые говорят о полученных от вас сообщениях, которые вы не отправляли;
- в вашем почтовом ящике находится большое количество сообщений без обратного адреса и заголовка.

Действия при наличии признаков заражения компьютера.

Прежде чем предпринимать какие-либо действия, необходимо сохранить результаты работы на внешнем носителе (дискете, CD- или DVD-диске, флэш-карте и пр.). Далее необходимо:

- отключить компьютер от локальной сети и Интернета, если он к ним был подключен;
- если симптом заражения состоит в том, что невозможно загрузиться с жесткого диска компьютера (компьютер выдает ошибку, когда вы его включаете), попробовать загрузиться в режиме защиты от сбоев или с диска аварийной загрузки Windows;
- запустить антивирусную программу.

Контрольные вопросы

1. В чем различие между антивирусными сканерами и мониторами?
2. Какие существуют признаки заражения компьютера вирусом?
3. Что необходимо сделать в первую очередь в случае заражения компьютера вирусом?

1.6.2. Компьютерные вирусы и защита от них

Обязательным свойством компьютерного вируса является способность к «размножению» (самокопированию). Вирусы могут также незаметно для пользователя внедряться в файлы, загрузочные секторы дисков и документы. Название

«вирус» по отношению к компьютерным программам пришло из биологии именно по признаку способности к саморазмножению.

После заражения компьютера вирус может активизироваться и заставить компьютер выполнять какие-либо действия. Активизация вируса может быть связана с различными событиями (наступлением определенной даты или дня недели, запуском программы, открытием документа и т. д.).



Компьютерные вирусы являются вредоносными программами, которые могут «размножаться» и скрытно внедрять свои копии в файлы, загрузочные секторы дисков и документы. Активизация компьютерного вируса может вызывать уничтожение программ и данных.

В настоящее время известно несколько десятков тысяч вирусов, заражающих компьютеры различных операционных систем. По «среде обитания» вирусы можно разделить на загрузочные, файловые и макро-вирусы.

Загрузочные вирусы. Загрузочные вирусы заражают загрузочный сектор гибкого или жесткого диска. Принцип действия загрузочных вирусов основан на алгоритмах запуска операционной системы при включении или перезагрузке компьютера. После необходимых тестов установленного оборудования программа системной загрузки считывает первый физический сектор загрузочного диска (гибкого, жесткого, оптического или флэш-диска в зависимости от параметров, установленных в BIOS Setup) и передает на него управление.

При заражении дисков загрузочные вирусы «подставляют» свой код вместо программы, получающей управление при загрузке системы, и отдают управление не оригинальному коду загрузчика, а коду вируса. При инфицировании диска вирус в большинстве случаев переносит оригинальный загрузочный сектор в какой-либо другой сектор диска (например, в первый свободный).

Профилактическая защита от таких вирусов состоит в отказе от загрузки операционной системы с гибких дисков и установке в BIOS вашего компьютера защиты загрузочного сектора от изменений. С помощью программы BIOS Setup

можно провести настройку BIOS таким образом, что будет запрещена (заблокирована) любая запись в загрузочный сектор диска и компьютер будет защищен от заражения загрузочными вирусами.

Файловые вирусы. Файловые вирусы различными способами внедряются в исполнимые файлы и обычно активизируются при их запуске. После запуска зараженного файла вирус находится в оперативной памяти компьютера и является активным (т. е. может заражать другие файлы) вплоть до момента выключения компьютера или перезагрузки операционной системы.

Практически все загрузочные и файловые вирусы **резидентны**, т. е. они находятся в оперативной памяти компьютера и в процессе работы пользователя могут осуществлять опасные действия (стирать данные на дисках, изменять названия и другие атрибуты файлов и т. д.). Лечение от резидентных вирусов затруднено, так как даже после удаления зараженных файлов с дисков, вирус остается в оперативной памяти и возможно повторное заражение файлов.

Профилактическая защита от файловых вирусов состоит в том, что не рекомендуется запускать на исполнение файлы, полученные из сомнительных источников и предварительно не проверенные антивирусными программами.

Макровирусы. Существуют макровирусы для интегрированного офисного приложения Microsoft Office (Word, Excel, PowerPoint и Access). Макро-вирусы фактически являются макрокомандами (макросами), на встроенном языке программирования Visual Basic for Applications (VBA), которые помещаются в документ.

При работе с документом пользователь выполняет различные действия: открывает документ, сохраняет, печатает, закрывает и т. д. При этом приложение ищет и выполняет соответствующие стандартные макросы. Макровирусы содержат стандартные макросы, вызываются вместо них и заражают каждый открываемый или сохраняемый документ.

Макровирусы являются **ограниченно резидентными**, т. е. они находятся в оперативной памяти и заражают документы, пока открыто приложение. Кроме того, макровирусы заражают шаблоны документов, и поэтому активизируются уже при запуске зараженного приложения.

Профилактическая защита от макровирусов состоит в предотвращении запуска вируса. При открытии документа в приложениях Microsoft Office сообщается о присутствии в них макросов (потенциальных вирусов) и предлагается запретить их загрузку. Выбор запрета на загрузку макросов надежно защитит ваш компьютер от заражения макровирусами, однако отключит и полезные макросы, содержащиеся в документе.

Контрольные вопросы

1. Каковы характерные особенности компьютерных вирусов как типа вредоносных программ?
2. Какие существуют типы компьютерных вирусов?

Практическая работа 1.8

Защита от компьютерных вирусов

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux, подключенный к Интернету.

Цель работы. Научиться лечить или удалять файловые вирусы из зараженных объектов.

Задание 1. В операционной системе Windows лечить и удалять файловые вирусы в режиме реального времени.

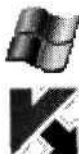
Задание 2. В операционной системе Linux лечить и удалять файловые вирусы в режиме реального времени.

Задание 3. В операционной системе Windows лечить и удалять файловые вирусы на выбранных носителях информации при признаках заражения.

Задание 4. В операционной системе Linux лечить и удалять файловые вирусы в выбранных папках при признаках заражения.

Варианты выполнения работы:

в операционной системе Windows или в операционной системе Linux.



Защита от файловых вирусов в режиме реального времени с использованием файлового монитора *Файловый Антивирус*, входящего в **Антивирус Касперского**

С помощью антивирусной программы Антивирус Касперского проверим компьютер на наличие вирусов, и при их обнаружении вылечим или удалим зараженные файлы.

Файловый Антивирус контролирует файловую систему компьютера в реальном времени (файловый монитор). Он проверяет все открываемые, запускаемые и сохраняемые файлы на вашем компьютере и всех присоединенных дисках. Каждое обращение к файлу перехватывается приложением, и файл проверяется на присутствие известных вирусов. Дальнейшая работа с файлом возможна только в том случае, если файл не заражен или был успешно вылечен Антивирусом Касперского. Если же файл по каким-либо причинам невозможно вылечить, он будет удален.

Прежде всего, необходимо через Интернет обновить саму антивирусную программу и вирусную базу данных.

1. В операционной системе Windows соединиться с Интернетом.

На *Панели задач* в контекстном меню значка антивирусной программы Антивирус Касперского выбрать пункт *Обновление*.

В появившемся диалоговом окне будет отображаться процесс обновления антивирусной программы и вирусной базы данных.

50% - Обновление : работает

Загружается: dailyc.avc
 Расположение: http://dnl-eu12.kaspersky-labs.com/

Размер обновлений: 80 КБ Запуск: 01.02.2008 20:12:24
 Скорость: 2,22 Кб/сек Длительность: 00:04:19
 Завершение: 01.02.2008 20:17:53

Событие	Имя объекта	Время	Трафик
Файл загружен	bases/av/avc/386/av-386-0607g.xml...	01.02.2008 20:16:08	1,2 КБ
Файл загружен	bases/av/avc/386_w/avw-386-0607g.x...	01.02.2008 20:16:24	1,1 КБ
Файл загружен	bases/info/info-0607g.xml.dif	01.02.2008 20:16:33	1 КБ
Файл загружен	diffs/bases/av/avc/386/ornup.avc.chk	01.02.2008 20:16:38	1,2 КБ
Файл загружен	diffs/bases/av/avc/386/dailyc.avc.3wb	01.02.2008 20:16:40	7,1 КБ
Файл загружен	diffs/bases/av/avc/386/dailyc.avc.div	01.02.2008 20:16:41	20 Б

Показывать все события

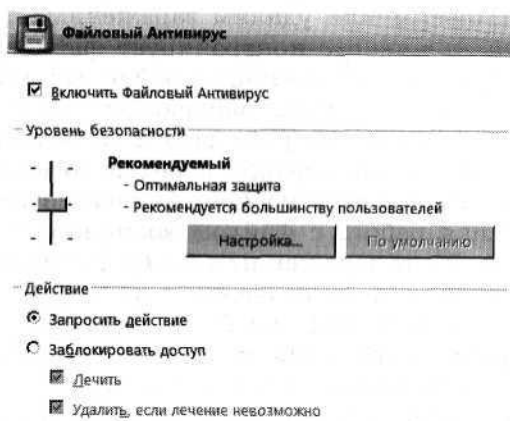
Действия

Настроим параметры антивирусного монитора (*Файловый Антивирус*).

- В контекстном меню значка антивирусной программы выбрать пункт *Настройка....*

В меню левой части появившегося диалогового окна выбрать *Файловый Антивирус*.

В правой части окна щелкнуть по кнопке *Настройка*.

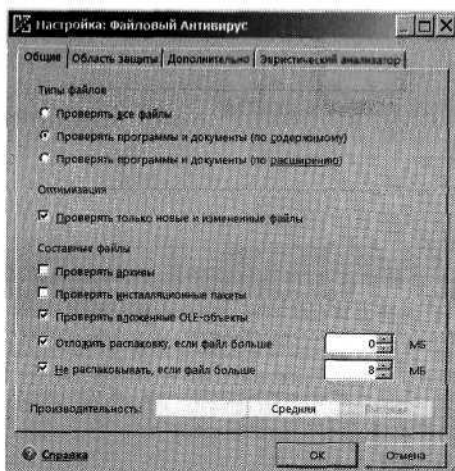


- В появившемся диалоговом окне на вкладке *Общие* выбрать типы проверяемых файлов.

На вкладке *Область защиты* выбрать проверяемые диски.

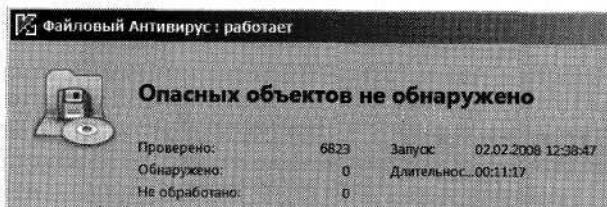
На вкладке *Дополнительно* выбрать режим проверки.

На вкладке *Эвристический анализатор* выбрать режим его использования и уровень эвристической проверки.



4. Антивирусный монитор (*Файловый Антивирус*) запускается автоматически при старте операционной системы и работает в качестве фонового системного процесса, проверяя на вредоносность совершаемые другими программами действия.

Активизировать ссылку *Открыть отчет*. Появится информационное окно с результатами проверки.



Защита от файловых вирусов в режиме реального времени с использованием файлового монитора, входящего в антивирус KlamAV

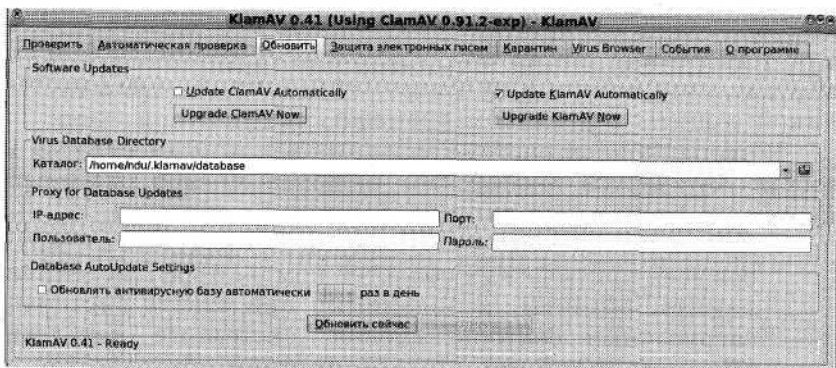


Прежде всего, необходимо через Интернет обновить саму антивирусную программу и вирусную базу данных.

1. В операционной системе Linux соединиться с Интернетом.

Запустить антивирус KlamAV и в диалоговом окне приложения выбрать пункт *Обновить*.

В появившемся диалоговом окне будет отображаться процесс обновления антивирусной программы и вирусной базы данных.



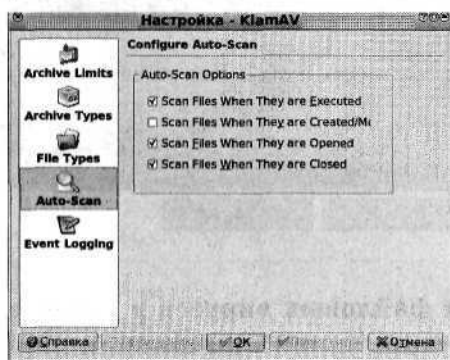
Настроим параметры антивирусного монитора антивируса KlamAV.

- В диалоговом окне антивирусной программы выбрать пункт *События*.

Щелкнуть по кнопке *Options*.

Появится диалоговое окно *Настройка*.

Выбрать пункт *Auto-Scan* и установить режим проверки файлов в реальном времени.



- Антивирусный монитор антивируса KlamAV запускается автоматически при старте операционной системы и работает в качестве фонового системного процесса, проверяя на вредоносность совершаемые другими программами действия.



Поиск, лечение или удаление файловых вирусов в Windows

Крайне важно периодически проводить проверку вашего компьютера на присутствие вирусов. Это необходимо делать для того, чтобы исключить возможность распространения вирусов, которые не были обнаружены компонентами постоянной защиты в реальном времени.

В операционной системе Windows для поиска вирусов в состав Антивируса Касперского включен сканер, который проверяет на наличие вирусов следующие области:

- Критические области.** Проверка на присутствие вирусов всех критических областей компьютера (системная память, объекты, исполняемые при старте системы, загрузочные секторы дисков, системные каталоги Microsoft Windows). Цель — быстрое обнаружение в

системе активных вирусов без запуска полной проверки компьютера.

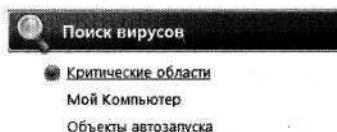
- *Мой Компьютер*. Поиск вирусов на вашем компьютере с тщательной проверкой всех подключенных дисков, памяти, файлов.
- *Объекты автозапуска*. Проверка на присутствие вирусов объектов, загрузка которых осуществляется при старте операционной системы, а также оперативной памяти и загрузочных секторов дисков.

Проведем проверку на вирусы выбранных областей хранения программ и данных.

1. В контекстном меню значка антивирусной программы выбрать пункт *Антивирус Касперского*.

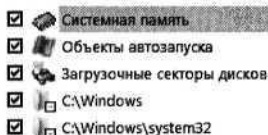
В меню левой части появившегося диалогового окна выбрать *Поиск вирусов*.

Раскроется список областей поиска вирусов.



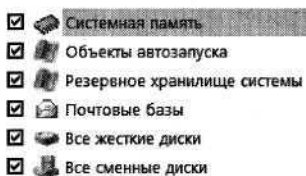
2. Выбрать пункт *Критические области*.

В правой части диалогового окна появится список критических областей.



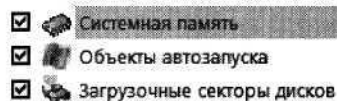
3. Выбрать пункт *Мой Компьютер*.

В правой части диалогового окна появится список некоторых критических областей и всех дисков компьютера.

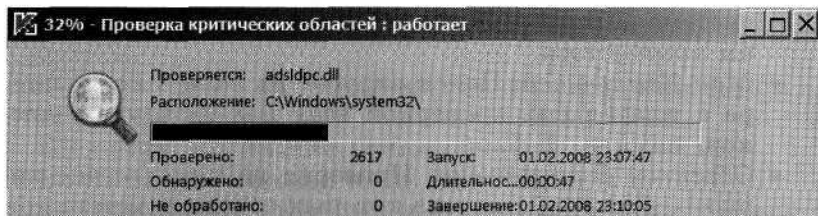


4. Выбрать пункт *Объекты автозапуска*.

В правой части диалогового окна появится список некоторых критических областей, объекты автозапуска и загрузочные секторы дисков.



5. Запустить проверку выбранной области хранения программ и данных (например, *Критические области*). Появится информационное окно, показывающее, сколько файлов проверено, сколько вирусов обнаружено (если они есть) и другая информация.



Поиск, лечение или удаление файловых вирусов в Linux



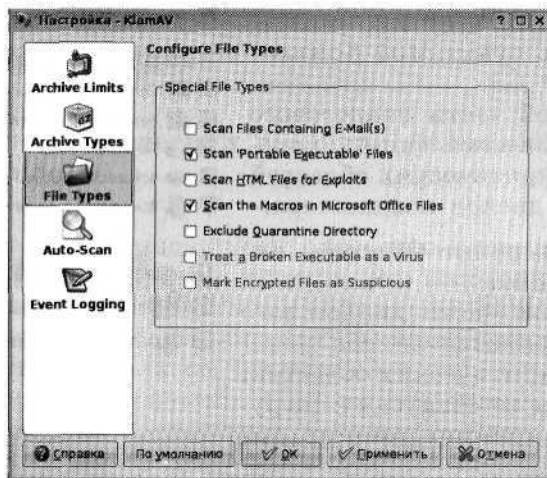
В операционной системе Linux для поиска вирусов в состав антивируса KlamAV включен сканер, который проверяет на наличие вирусов выбранные пользователем папки.

1. В диалоговом окне антивирусной программы выбрать пункт *События*.

Щелкнуть по кнопке *Options*.

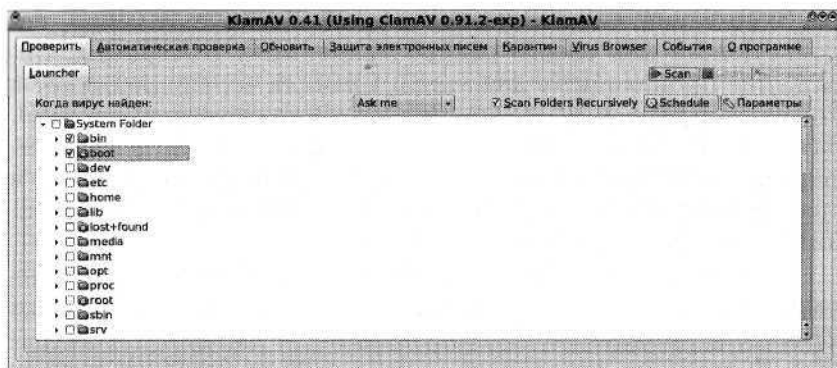
Появится диалоговое окно *Настройка*.

Выбрать для проверки исполнимые файлы (пункт *Scan 'Portable Executable' Files*) и файлы документов, которые могут содержать макровирусы (пункт *Scan the Macros in Microsoft Office Files*).



2. На вкладке *Проверить* диалогового окна антивирусной программы выбрать папки для проверки на наличие файловых вирусов.

Начать проверку, щелкнув по кнопке *Scan*.



3. После сканирования на наличие вирусов будет показан отчет о результатах проверки.

1.6.3. Сетевые черви и защита от них

К сетевым червям (англ. worm) относятся вредоносные программы, распространяющие свои копии по локальным и/или глобальным сетям. Для своего распространения сетевые черви используют разнообразные сервисы глобальных и локальных компьютерных сетей: Всемирную паутину, электронную почту и т. д.

Сетевые черви кроме вредоносных действий, которыми обладают и классические компьютерные вирусы, могут выполнять шпионскую функцию троянских программ.

1.6.4. Троянские программы и защита от них



Сетевые черви являются вредоносными программами, которые проникают на компьютер, используя сервисы компьютерных сетей. Активизация сетевого червя может вызывать уничтожение программ и данных, а также похищение персональных данных пользователя.

Основным признаком, по которому типы червей различаются между собой, является способ распространения червя — как он передает свою копию на удаленные компьютеры. Однако многие сетевые черви используют более одного способа распространения своих копий по компьютерам локальных и глобальных сетей.

Web-черви. Отдельную категорию составляют черви, использующие для своего распространения Web-серверы. Заражение происходит в два этапа. Сначала червь проникает в компьютер-сервер и модифицирует Web-страницы сервера. Затем червь «ждет» посетителей, которые запрашивают информацию с зараженного сервера (например, открывают в браузере зараженную Web-страницу), и таким образом проникает на другие компьютеры сети.

Разновидностью Web-червей являются скрипты — активные элементы (программы) на языках JavaScript или VBScript, которые могут содержаться в файлах Web-страниц. Заражение локального компьютера происходит при их передаче по Всемирной паутине с серверов Интернета в браузер локального компьютера.

Профилактическая защита от таких червей состоит в том, что в браузере можно запретить получение активных элементов на локальный компьютер.

Еще более эффективны Web-антивирусные программы, которые включают **межсетевой экран** и **модуль проверки скриптов** на языках JavaScript и VBScript.

Межсетевой экран. Межсетевой экран (брандмауэр) — это программное или аппаратное обеспечение, которое проверяет информацию, входящую в компьютер из локальной сети или Интернета, а затем либо отклоняет ее, либо пропускает в компьютер, в зависимости от параметров брандмауэра.

Межсетевой экран обеспечивает проверку всех Web-страниц, поступающих на компьютер пользователя. Каждая Web-страница перехватывается и анализируется межсетевым экраном на присутствие вредоносного кода. Распознавание вредоносных программ происходит на основании баз, используемых в работе меж сетевого экрана, и с помощью эвристического алгоритма. Базы содержат описание всех известных на настоящий момент вредоносных программ и способов их обезвреживания. Эвристический алгоритм позволяет обнаруживать новые вирусы, еще не описанные в базах.

Если Web-страница, к которой обращается пользователь, содержит вредоносный код, доступ к нему блокируется. При этом на экран выводится уведомление о том, что запрашиваемая страница заражена. Если Web-страница не содержит вредоносного кода, она сразу же становится доступной для пользователя.

Если на компьютере используются такие программы, как программа передачи мгновенных сообщений или сетевые игры, которым требуется принимать информацию из Интернета или локальной сети, межсетевой экран запрашивает пользователя о блокировании или разрешении подключения. Если пользователь разрешает подключение, брандмауэр Windows создает исключение, чтобы в будущем не тревожить пользователя запросами по поводу поступления информации для этой программы.

Проверка скриптов в браузере. Проверка всех скриптов, обрабатываемых в браузере, производится следующим образом:

- каждый запускаемый на Web-странице скрипт перехватывается модулем проверки скриптов и анализируется на присутствие вредоносного кода;
- если скрипт содержит вредоносный код, модуль проверки скриптов блокирует его, уведомляя пользователя специальным всплывающим сообщением;
- если в скрипте не обнаружено вредоносного кода, он выполняется.

Почтовые черви. Почтовые черви для своего распространения используют электронную почту. Червь либо отправляет свою копию в виде вложения в электронное письмо, либо отправляет ссылку на свой файл, расположенный на каком-либо сетевом ресурсе. В первом случае код червя активизируется при открытии (запуске) зараженного вложения, во втором — при открытии ссылки на зараженный файл. В обоих случаях эффект одинаков — активизируется код червя.

Лавинообразная цепная реакция распространения почтового червя базируется на том, что червь после заражения компьютера начинает рассылать себя по всем адресам электронной почты, которые имеются в адресной книге пользователя.

Профилактическая защита от почтовых червей состоит в том, что не рекомендуется открывать вложенные в почтовые сообщения файлы, полученные из сомнительных источников.

Профилактическая защита от таких червей состоит в том, что рекомендуется своевременно скачивать из Интернета и устанавливать обновления системы безопасности операционной системы и приложений.

Контрольные вопросы

1. Как сетевые черви проникают на компьютер?
2. Какие типы сетевых червей существуют?

Практическая работа 1.9

Защита от сетевых червей

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux, подключенный к Интернету.

Цель работы. Научиться предотвращать проникновение сетевых червей из локальной или глобальной сети Интернет на локальный компьютер.

Задание 1. В операционной системе Windows предотвратить проникновение Web-червей из локальной или глобальной сети Интернет на локальный компьютер.

Задание 2. В операционной системе Linux предотвратить проникновение Web-червей из локальной или глобальной сети Интернет на локальный компьютер.

Задание 3. В операционной системе Windows предотвратить проникновение почтовых червей из локальной или глобальной сети Интернет на локальный компьютер.

Задание 4. В операционной системе Linux предотвратить проникновение почтовых червей из локальной или глобальной сети Интернет на локальный компьютер.

Варианты выполнения работы:

в операционной системе Windows или в операционной системе Linux.



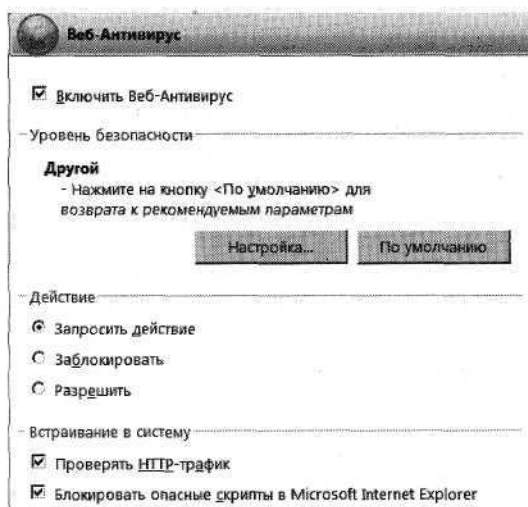
Защита от Web-червей с использованием межсетевого экрана *Web-Антивирус*, входящего в Антивирус Касперского

Настроим параметры межсетевого экрана *Web-Антивирус*.

1. В контекстном меню значка антивирусной программы выбрать пункт *Настройка...*

В меню левой части появившегося диалогового окна выбрать *Web-Антивирус*.

В правой части окна щелкнуть по кнопке *Настройка*.

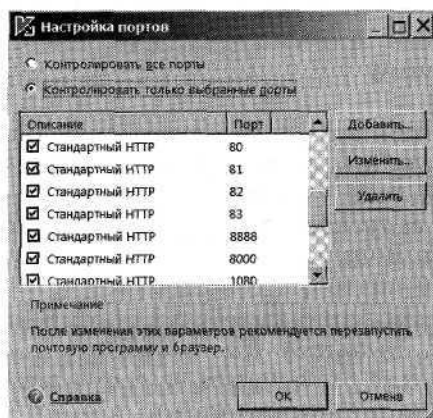


Локальный компьютер получает Web-страницы через логические порты, обозначающиеся числовым кодом.

2. Щелкнуть по ссылке *Проверить HTTP-трафик*.

В диалоговом окне *Настройка портов* перечислен список портов, которые чаще всего используются для передачи Web-страниц.

Межсетевой экран предусматривает контроль прохождения Web-страниц через эти порты.



Защита от Web-червей с использованием меж- сетевого экрана, входящего в KlamAV



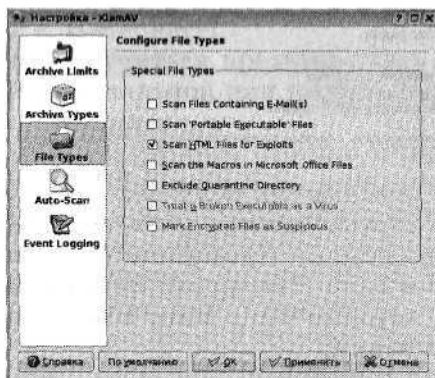
В операционной системе Linux для предотвращения проникновения Web-червей из локальной и глобальной сети Интернет на локальный компьютер в состав антивируса KlamAV включен межсетевой экран.

1. В диалоговом окне антивирусной программы выбрать пункт *События*.

Щелкнуть по кнопке *Options*.

Появится диалоговое окно *Настройка*.

Выбрать для проверки файлы Web-страницы (пункт *Scan HTML Files for Exploits*).



Защита от почтовых червей с использованием Почтового Антивируса, входящего в Антивирус Касперского

В состав Антивируса Касперского включен специальный компонент, обеспечивающий защиту входящей и исходящей почты на наличие почтовых червей, — *Почтовый Антивирус*. Он запускается при старте операционной системы, постоянно находится в оперативной памяти компьютера и проверяет все почтовые сообщения.

Каждое письмо, принимаемое или отправляемое пользователем, перехватывается *Почтовым Антивирусом*, затем почтовое сообщение разбирается на составляющие его части: заголовок письма, тело, вложения.

Тело и вложения почтового сообщения проверяются на присутствие в них вредоносных программ. Распознавание вредоносных объектов происходит на основании баз, используемых в работе приложения, и с помощью эвристического алгоритма. Базы содержат описание всех известных на настоящий момент вредоносных программ и способов их обезвреживания. Эвристический алгоритм позволяет обнаруживать новые вирусы, еще не описанные в базах.

Если тело или вложение письма содержит вредоносный код, *Почтовый Антивирус* блокирует письмо, помещает копию зараженного объекта в **резервное хранилище** и пытается обезвредить объект. В результате успешного лечения письмо становится доступным для пользователя, если же лечение произвести не удастся, зараженный объект из письма удаляется.

Если тело или вложение письма содержит код, похожий на вредоносный, но стопроцентной гарантии этого нет, подозрительная часть письма помещается в специальное хранилище — **карантин**.

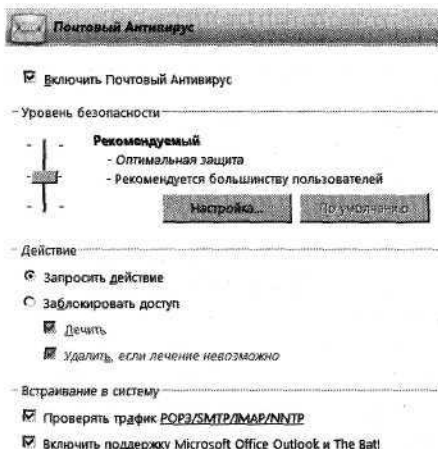
Если в письме не обнаружено вредоносного кода, оно сразу же становится доступным для пользователя.

Настроим параметры почтового антивируса *Почтовый Антивирус*.

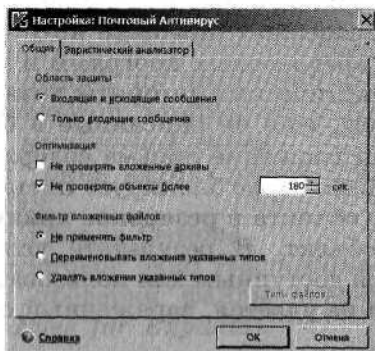
1. В контекстном меню значка антивирусной программы выбрать пункт *Настройка....*

В меню левой части появившегося диалогового окна выбрать *Почтовый Антивирус*.

В правой части окна щелкнуть по кнопке *Настройка*.



2. В появившемся диалоговом окне *Настройка: Почтовый Антивирус* на вкладке *Общие* выбрать параметры защиты почтовых сообщений от почтовых вирусов.



Защита от почтовых червей с использованием антивируса KlamAV



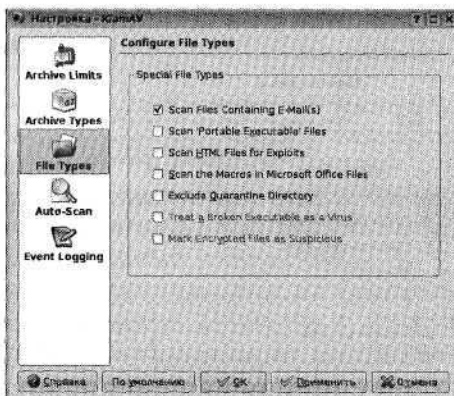
В операционной системе Linux для предотвращения прикновения почтовых червей из локальной и глобальной сети Интернет на локальный компьютер в состав антивируса KlamAV включен почтовый антивирус.

1. В диалоговом окне антивирусной программы выбрать пункт *События*.

Щелкнуть по кнопке *Options*.

Появится диалоговое окно *Настройка*.

Выбрать для проверки файлы электронной почты (пункт *Scan Files Containing E-Mail(s)*).



1.6.4. Троянские программы и защита от них

Троянские программы осуществляют несанкционированные пользователем действия по сбору и передаче информации злоумышленнику, а также ее разрушение или злонамеренную модификацию. Кроме того, троянские программы могут вызывать нарушение работоспособности компьютера или незаметно для пользователя использовать ресурсы компьютера в целях злоумышленника.

Троянские программы обычно проникают на компьютер как сетевые черви. Они различаются между собой по тем действиям, которые они производят на зараженном компьютере.



Троянская программа, троянец (от англ. trojan) — вредоносная программа, которая выполняет несанкционированную пользователем передачу управления компьютером удаленному пользователю, а также действия по удалению, модификации, сбору и пересылке информации третьим лицам.

Троянские утилиты удаленного администрирования. Троянские программы этого класса являются утилитами удаленного администрирования компьютеров в сети. Утилиты скрытого управления позволяют принимать или отсылать файлы, запускать и уничтожать их, выводить сообщения, стирать информацию, перезагружать компьютер и т. д.

При запуске троянец устанавливает себя в системе и затем следит за ней, при этом пользователю не выдается никаких сообщений о действиях троянской программы в системе. В результате «пользователь» этой троянской программы может и не знать о ее присутствии в системе, в то время как его компьютер открыт для удаленного управления.

Троянские программы данного типа являются одним из самых опасных видов вредоносного программного обеспечения, поскольку в них заложена возможность самых разнообразных злоумышленных действий, в том числе они могут быть использованы для обнаружения и передачи конфиденциальной информации.

Троянские программы — шпионы. Данные троянцы осуществляют электронный шпионаж за пользователем зараженного компьютера: вводимая с клавиатуры информация, снимки экрана, список активных приложений и дей-

ствия пользователя с ними сохраняются в каком-либо файле на диске и периодически отправляются злоумышленнику.

Троянские программы этого типа часто используются для кражи информации пользователей различных систем онлайн-платежей и банковских систем.

Рекламные программы. Рекламные программы (англ. *Adware: Advertisement* — реклама и *Software* — программное обеспечение) встраивают рекламу в основную полезную программу и могут выполнять функцию троянских программ. Рекламные программы могут скрытно собирать различную информацию о пользователе компьютера и затем отправлять ее злоумышленнику.

Защита от троянских программ. Троянские программы часто изменяют записи системного реестра операционной системы, который содержит все сведения о компьютере и установленном программном обеспечении. Для их удаления необходимо восстановление системного реестра, поэтому компонент, восстанавливающий системный реестр, входит в современные операционные системы.

Контрольные вопросы

1. Какие вредоносные действия выполняют троянские программы?

Практическая работа 1.10

Защита от троянских программ

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows, подключенный к Интернету.

Цель работы. Научиться обнаруживать и обезвреживать троянские программы.

Задание 1. В операционной системе Windows предотвратить проникновение и действия троянских программ.

Задание 2. В операционной системе Windows восстановить систему (системный реестр) после проникновения троянских программ.

Варианты выполнения работы:

включить разные параметры проактивной защиты.



Защита от троянских или других вредоносных программ с использованием *Проактивной защиты*, входящей в **Антивирус Касперского**

В состав Антивируса Касперского включен специальный компонент *Проактивная защита*, который анализирует и сообщает пользователю:

- действия каждого запускаемого на компьютере приложения, которые являются характерными при опасной (шпионской) активности;
- попытки изменения системного реестра (удаление, добавление ключей системного реестра, ввод значений для ключей в недопустимом формате), характерные для шпионской деятельности.

Настроим параметры компонента *Проактивная защита*.

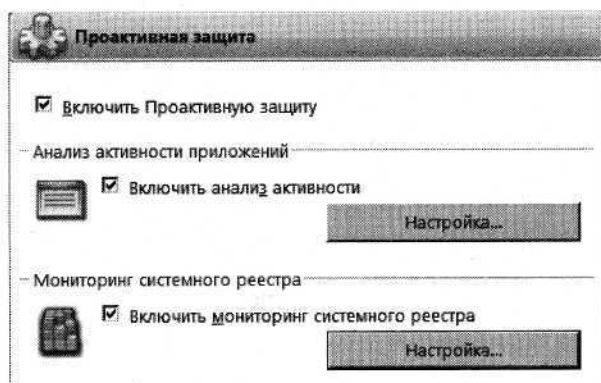
1. В контекстном меню значка антивирусной программы выбрать пункт *Настройка...*

В меню левой части появившегося диалогового окна выбрать *Проактивная защита*.

2. В правой части окна установить флажки:

- *Включить Проактивную защиту*;
- *Включить анализ активности*;
- *Включить мониторинг системного реестра*.

Установить необходимые параметры анализа активности и мониторинга системного реестра, щелкнув по кнопкам *Настройка*.



3. Пользователь оценивает потенциальную опасность активности приложения или изменения системного реестра и может разрешить или запретить этот процесс.



Восстановление операционной системы (системного реестра) после проникновения троянских или других вредоносных программ

В операционной системе Windows Vista восстановление системы можно провести следующим образом.

1. В операционной системе Windows Vista ввести команду [Панель управления-Центр архивации и восстановления].

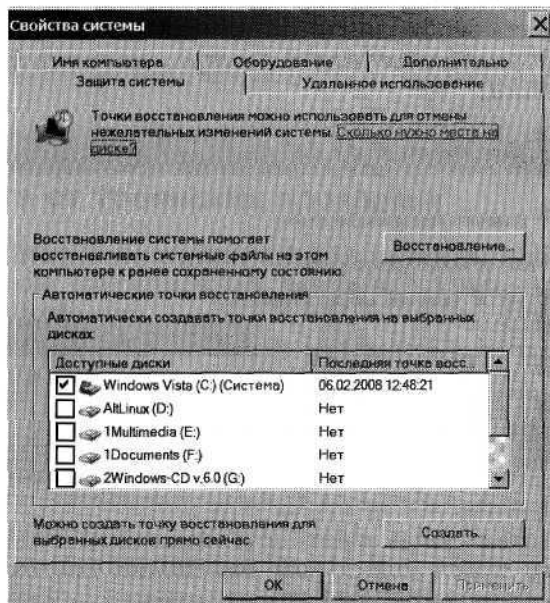
В появившемся диалоговом окне *Центр архивации и восстановления* активировать ссылку *Устранение неполадок Windows с помощью восстановления системы*.

Появится диалоговое окно *Восстановление системных файлов и параметров*, в котором можно выбрать точку восстановления системы (системного реестра).

2. Активировать ссылку *Создание точки восстановления и настройка параметров*.

Появится диалоговое окно *Свойства системы*, в котором на вкладке *Защита системы* можно выбрать или создать точку восстановления системы (системного реестра).

Кроме того, можно создать точки восстановления для любых логических дисков.



1.6.5. Хакерские утилиты и защита от них

Сетевые атаки. Сетевые атаки на удаленные серверы реализуются с помощью специальных программ, которые посылают на них многочисленные запросы. Это приводит к отказу в обслуживании (зависанию сервера), если ресурсы атакуемого сервера недостаточны для обработки всех поступающих запросов.

DoS-программы (от англ. Denial of Service — отказ в обслуживании) реализуют атаку с одного компьютера с ведома пользователя. DoS-программы обычно наносят ущерб удаленным компьютерам и сетям, не нарушая работоспособности зараженного компьютера.

DDoS-программы (от англ. Distributed DoS — распределенный DoS) реализуют распределенные атаки с разных компьютеров, причем без ведома пользователей зараженных компьютеров. Для этого DDoS-программа засылается на компьютеры «жертв-посредников» и после запуска в зависимости от текущей даты или по команде от хакера начинает сетевую атаку на указанный сервер в сети.

Некоторые хакерские утилиты реализуют фатальные сетевые атаки. Такие утилиты используют уязвимости в операционных системах и приложениях и отправляют специально оформленные запросы на атакуемые компьютеры в сети. В результате сетевой запрос специального вида вызывает критическую ошибку в атакуемом приложении, и система прекращает работу.

Утилиты взлома удаленных компьютеров. Утилиты взлома удаленных компьютеров предназначены для проникновения в удаленные компьютеры с целью дальнейшего управления ими (используя методы троянских программ типа утилит удаленного администрирования) или для внедрения во взломанную систему других вредоносных программ.

Утилиты взлома удаленных компьютеров обычно используют уязвимости в операционных системах или приложениях, установленных на атакуемом компьютере. Профилактическая защита от таких хакерских утилит состоит в своевременной загрузке из Интернета обновлений системы безопасности операционной системы и приложений.

Руткиты. Руткит (от англ. root kit — «набор для получения прав root») — программа или набор программ для скрытого взятия под контроль взломанной системы. Это утилиты, используемые для сокрытия вредоносной актив-

ности. Они маскируют вредоносные программы, чтобы избежать их обнаружения антивирусными программами. Руткиты модифицируют операционную систему на компьютере и заменяют основные ее функции, чтобы скрыть свое собственное присутствие и действия, которые предпринимает злоумышленник на зараженном компьютере.

Защита от хакерских атак, сетевых червей и троянских программ. Защита компьютерных сетей или отдельных компьютеров от несанкционированного доступа может осуществляться с помощью межсетевой экран.

Межсетевой экран позволяет:

- 1) блокировать хакерские DoS-атаки, не пропуская на защищаемый компьютер сетевые пакеты с определенных серверов (определенных IP-адресов или доменных имен);
- 2) не допускать проникновение на защищаемый компьютер сетевых червей (почтовых, Web и др.);
- 3) препятствовать троянским программам отправлять конфиденциальную информацию о пользователе и компьютере.

Контрольные вопросы

1. Какие типы хакерских атак и методы защиты от них существуют?
2. К какому типу вредоносных программ относятся руткиты?

Практическая работа 1.11

Защита от хакерских атак

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows, подключенный к Интернету.

Цель работы. Научиться обнаруживать и обезвреживать руткиты и защищать компьютер от хакерских атак.

Задание 1. В операционной системе Windows обнаружить и обезвредить руткиты.

Задание 2. В операционной системе Windows включить брандмауэр Windows (межсетевой экран), который предот-

вращает попытки доступа хакеров или вредоносных программ к локальному компьютеру через локальную сеть или Интернет.

Варианты выполнения работы:

по разному настроить сетевой экран (брандмауэр).



Защита от руткитов с помощью Поиска руткитов, входящего в Антивирус Касперского

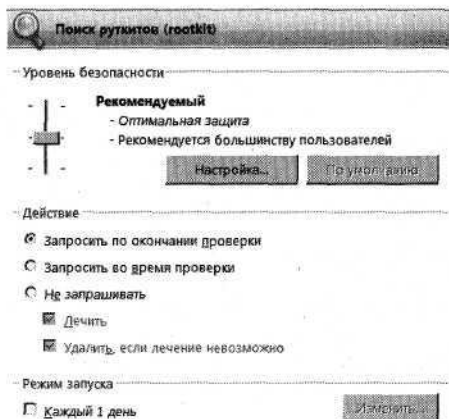
В состав Антивируса Касперского включен специальный компонент *Поиск руткитов*. Поиск руткитов ведется как на основе поиска известного программного кода руткита, так и с использованием эвристического метода, который нацелен на обнаружение типичных последовательностей операций, позволяющих сделать вывод о наличии руткита с достаточной долей вероятности.

Настроим параметры модуля *Поиск руткитов (rootkit)*.

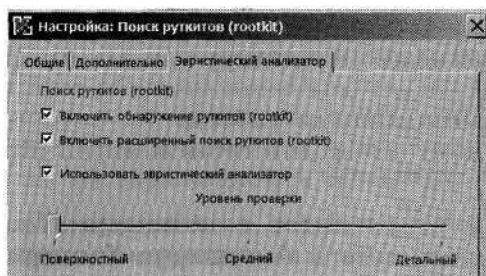
1. В контекстном меню значка антивирусной программы выбрать пункт *Настройка....*

В меню левой части появившегося диалогового окна выбрать *Поиск руткитов (rootkit)*.

В правой части окна щелкнуть по кнопке *Настройка*.



2. В появившемся диалоговом окне *Настройка: Поиск руткитов (rootkit)* на вкладке *Эвристический анализатор* установить флажки различных методов поиска руткитов и установить *Уровень проверки эвристического анализатора*.



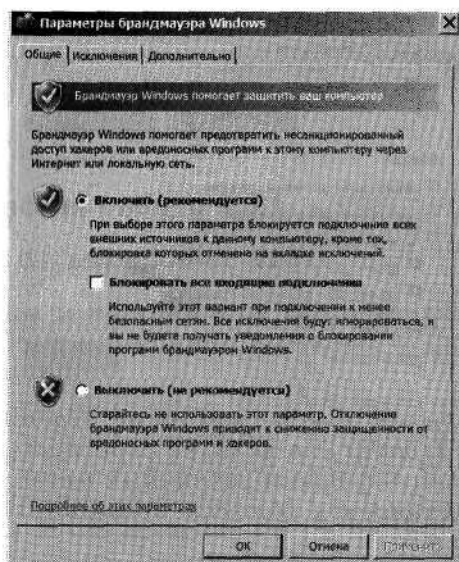
Настройка межсетевой экран (брандмауэра) в операционной системе Windows

В операционной системе Windows Vista настроим межсетевой экран (брандмауэр).

1. В операционной системе Windows Vista ввести команду [Панель управления-Брандмауэр Windows].
В появившемся диалоговом окне *Брандмауэр Windows* активировать ссылку *Изменить параметры*.
2. Появится диалоговое окно *Параметры Брандмауэра Windows*.

На вкладке *Общие* установить переключатель в положение *Включить*.

Брандмауэр Windows предотвращает попытки доступа хакеров или вредоносных программ к локальному компьютеру через локальную сеть или Интернет.



Глава 2

Моделирование и формализация

При изучении данной главы рекомендуется установить следующее программное обеспечение для операционной системы Windows.

	<p>Интерактивные компьютерные модели:</p> <ul style="list-style-type: none">• Открытая физика. Части 1 и 2;• Открытая астрономия;• Открытая математика. Функции и графики;• Открытая математика. Планиметрия;• Открытая математика. Стереометрия;• Открытая химия;• Открытая биология.	<p>ЦОР на CD</p>       
--	--	---

2.1. Моделирование как метод познания

Моделирование. Человечество в своей деятельности (научной, образовательной, технологической, художественной) постоянно создает и использует модели окружающего мира. Строгие правила построения моделей сформулировать невозможно, однако человечество накопило богатый опыт моделирования различных объектов и процессов.

Модели позволяют представить в наглядной форме объекты и процессы, недоступные для непосредственного восприятия (очень большие или очень маленькие объекты, очень быстрые или очень медленные процессы и др.). Наглядные модели часто используются при обучении. В курсе географии первые представления о нашей планете Земля мы получаем, изучая ее модель — глобус, в курсе физики изучаем работу двигателя внутреннего сгорания по его модели, в химии при изучении строения вещества используем модели молекул и кристаллических решеток, в биологии изучаем строение человека по анатомическим муляжам и др.

Модели имеют чрезвычайно важную роль в проектировании и создании различных технических устройств, машин и механизмов, зданий, электрических цепей и т. д. Без предварительного создания чертежа (рис. 2.1) невозможно изготовить даже простую деталь, не говоря уже о самолете.

В процессе проектирования зданий и сооружений кроме чертежей часто изготавливают их макеты. В процессе разработки летательных аппаратов поведение их моделей в воздушных потоках исследуют в аэродинамической трубе. Разработка электрической схемы обязательно предшествует созданию электрических цепей и т. д.

Развитие науки невозможно без создания **теоретических моделей** (теорий, законов, гипотез и т. д.), отражающих строение, свойства и поведение реальных объектов. Создание новых теоретических моделей иногда коренным образом меняет представление человечества об окружающем мире (гелиоцентрическая система мира Коперника, модель атома Резерфорда–Бора, модель расширяющейся Вселенной, модель генома человека и др.). Истинность теоретических моделей, т. е. их соответствие законам реального мира, проверяется с помощью опытов и экспериментов.



Рис. 2.1. Чертеж самолета

Все художественное творчество фактически является процессом создания моделей. Например, такой литературный жанр, как басня, переносит реальные отношения между людьми на отношения между животными и фактически создает модели человеческих отношений. Более того, практически любое литературное произведение может рассматриваться как модель реальной человеческой жизни. Моделями, в художественной форме отражающими реальную действительность, являются также живописные полотна, скульптуры, театральные постановки и т. д.



Моделирование — это метод познания, состоящий в создании и исследовании моделей.

Модель. Каждый объект имеет большое количество различных свойств. В процессе построения модели выделяются главные, наиболее существенные для проводимого исследования (с точки зрения цели моделирования) свойства. В процессе исследования аэродинамических качеств модели самолета в аэродинамической трубе важно чтобы модель имела геометрическое подобие оригиналу, но не важен, например, ее цвет. При построении электрических схем — моделей электрических цепей — необходимо учитывать порядок подключения элементов цепи друг к другу, но неважно их геометрическое расположение друг относительно друга и т. д.

Разные науки исследуют объекты и процессы под разными углами зрения и строят различные типы моделей. В физике изучаются процессы взаимодействия и изменения объектов, в химии — их химический состав, в биологии — строение и поведение живых организмов и т. д.

Возьмем в качестве примера человека, в разных науках он исследуется в рамках различных моделей. В рамках механики его можно рассматривать как материальную точку, в химии — как объект, состоящий из различных химических веществ, в биологии — как систему, стремящуюся к самосохранению и т. д.



Модель — это такой новый объект, который отражает **существенные** с точки зрения цели проводимого исследования (цели моделирования) **свойства** изучаемого объекта, явления или процесса.

Никакая модель не может заменить сам объект. Но при решении конкретной задачи, когда нас интересуют определенные свойства изучаемого объекта, модель оказывается полезным, а подчас и единственным инструментом исследования.

Контрольные вопросы

1. Может ли объект иметь несколько моделей? Приведите пример.
2. Могут ли разные объекты описываться одной и той же моделью? Если да, приведите пример.

2.2. Системный подход в моделировании

Понятие о системе. Окружающий нас мир состоит из множества различных объектов, каждый из которых имеет разнообразные свойства, и при этом объекты взаимодействуют между собой. Например, такие объекты, как планеты нашей Солнечной системы, имеют различные свойства (массу, геометрические размеры и т. д.) и по закону всемирного тяготения взаимодействуют с Солнцем и друг с другом.

Планеты входят в состав более крупного объекта — Солнечной системы, а Солнечная система — в состав нашей галактики Млечный Путь. Вместе с тем, планеты состоят из атомов различных химических элементов, а атомы — из элементарных частиц. Можно сделать вывод, что практически каждый объект состоит из других объектов, т. е. представляет собой **систему**.



Система состоит из объектов, которые называются **элементами** системы.

Важным признаком системы является ее **целостное функционирование**. Система является не набором отдельных элементов, а совокупностью взаимосвязанных элементов. Например, компьютер является системой, состоящей из различных устройств, при этом устройства связаны между собой и аппаратно (физически подключены друг к другу), и

функционально (между устройствами происходит обмен информацией).

Состояние системы характеризуется ее **структурой**, т. е. составом и свойствами элементов, их отношениями и связями между собой. Система сохраняет свою целостность под воздействием различных внешних воздействий и внутренних изменений до тех пор, пока она сохраняет неизменной свою структуру. Если структура системы меняется (например, удаляется один из элементов), то система может перестать функционировать как целое. Так, если удалить одно из устройств компьютера (например, процессор), компьютер выйдет из строя, т. е. прекратит свое существование как система.

Статические информационные модели. Любая система существует в пространстве и времени. В каждый момент времени система находится в определенном состоянии, которое характеризуется составом элементов, значениями их свойств, величиной и характером взаимодействия между элементами и т. д.

Так, состояние Солнечной системы в любой момент времени характеризуется составом входящих в нее объектов (Солнце, планеты и др.), их свойствами (размерами, положением в пространстве и т. д.), величиной и характером взаимодействия между собой (силами тяготения, с помощью электромагнитных волн и т. д.).

Модели, описывающие состояние системы в определенный момент времени, называются **статическими информационными моделями**.

В физике примерами статических информационных моделей являются модели, описывающие простые механизмы, в биологии — модели строения растений и животных, в химии — модели строения молекул и кристаллических решеток и т. д.

Динамические информационные модели. Состояние систем изменяется во времени, т. е. происходят процессы изменения и развития систем. Так, планеты движутся, изменяется их положение относительно Солнца и друг друга, Солнце, как и любая другая звезда, развивается, меняются ее химический состав, излучение и т. д.

Модели, описывающие процессы изменения и развития систем, называются **динамическими информационными моделями**.

В физике динамические информационные модели описывают движение тел, в биологии — развитие организмов или популяций животных, в химии — процессы прохождения химических реакций и т. д.

Контрольные вопросы

1. Образуют ли систему комплектующие компьютера: до сборки? После сборки? После включения компьютера?
2. В чем разница между статическими и динамическими информационными моделями? Приведите примеры статических и динамических информационных моделей.

2.3. Формы представления моделей

Модели материальные и модели информационные. Модели можно разбить на два больших класса: модели **предметные (материальные)** и модели **информационные**. Предметные модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме (глобус, анатомические муляжи, модели кристаллических решеток, макеты зданий и сооружений и др.).

Информационные модели представляют объекты и процессы в **образной** или **знаковой** форме.

Образные модели (рисунки, фотографии и др.) представляют собой зрительные образы объектов, зафиксированные на каком-либо носителе информации (бумаге, фото- и киноплёнке и др.). Широко используются образные информационные модели в образовании (вспомните учебные плакаты по различным предметам) и науках, где требуется классификация объектов по их внешним признакам (в ботанике, биологии, палеонтологии и др.).

Знаковые информационные модели строятся с использованием различных языков (знаковых систем). Знаковая информационная модель может быть представлена в форме текста (например, программы на языке программирования), формулы (например, второго закона Ньютона $\vec{F} = m \cdot \vec{a}$), таблицы (например, таблицы Периодической системы элементов Д. И. Менделеева) и т. д.

Иногда при построении знаковых информационных моделей используются одновременно несколько различных языков. Примерами таких моделей могут служить географические карты, графики, диаграммы и др. Во всех этих моделях используются одновременно как язык графических элементов, так и язык символов — букв, цифр и пр.

На протяжении своей истории человечество использовало различные способы и инструменты для создания информационных моделей. Эти способы постоянно совершенствовались. Так, первые информационные модели создавались в форме наскальных рисунков. В настоящее время информационные модели обычно строятся и исследуются с использованием современных компьютерных технологий.

Алгоритм как информационная модель. Алгоритмы лежат в основе современных информационных технологий. Алгоритм является информационной моделью процесса решения задачи. Исполнитель алгоритма выполняет алгоритм формально, не вникая в содержание поставленной задачи.

Человек при разработке и исполнении алгоритмов использует язык блок-схем. Блок-схема позволяет сделать алгоритм более наглядным и выделить в нем основные алгоритмические структуры (линейная, ветвление, цикл и др.). Человек может по блок-схеме легко проследить выполнение алгоритма, так как элементы блок-схем соединены стрелками, указывающими последовательность действий.

Элементы алгоритма изображаются на блок-схеме с помощью различных геометрических фигур (табл. 2.1), внутри которых записывается программный код.

Таблица 2.1. Блок-схемы основных алгоритмических структур



Контрольные вопросы



1. Какие существуют типы информационных моделей?
2. Каковы основные типы алгоритмических структур?

2.4. Формализация

Естественные (разговорные) языки используются для создания **описательных информационных моделей**. В истории науки известны многочисленные описательные информационные модели, например, гелиоцентрическая модель мира, которую предложил Коперник, формулировалась следующим образом:

- Земля вращается вокруг своей оси и вокруг Солнца;
- орбиты всех планет проходят вокруг Солнца.

С помощью формальных языков строятся **формальные информационные модели** (математические, логические и др.). Одним из наиболее широко используемых формальных языков является язык математики. Модели, построенные с использованием математических обозначений и формул, называются **математическими моделями**. Язык математики является совокупностью формальных языков, с некоторыми из них (алгебра, геометрия, тригонометрия) вы знакомитесь в школе, с другими (теория множеств, теория вероятностей и др.), сможете ознакомиться в процессе дальнейшего обучения.

Язык алгебры позволяет формализовать функциональные зависимости между величинами. Так, Ньютон формализовал гелиоцентрическую систему мира, открыв законы механики и закон всемирного тяготения и записав их в виде алгебраических функциональных зависимостей. В школьном курсе физики рассматривается много разнообразных функциональных зависимостей, выраженных на языке алгебры, которые представляют собой математические модели изучаемых явлений или процессов.

Язык алгебра логики (алгебры высказываний) позволяет строить **формальные логические модели**. С помощью алгеб-

ры высказываний можно формализовать (записать в виде логических выражений) простые и сложные высказывания, выраженные на естественном языке. Построение логических моделей позволяет решать логические задачи, строить логические модели устройств компьютера (сумматора, триггера) и т. д.



Процесс построения информационных моделей с помощью формальных языков называется **формализацией**.

В процессе познания окружающего мира человечество постоянно использует моделирование и формализацию. При изучении нового объекта сначала обычно строится его описательная информационная модель на естественном языке, затем она формализуется, т. е. выражается с использованием формальных языков (математики, логики и др.).

Визуализация формальных моделей. В процессе исследования формальных моделей часто производится их визуализация. Для визуализации алгоритмов используются блок-схемы, пространственных соотношений между объектами — чертежи, моделей электрических цепей — электрические схемы, логических моделей устройств — логические схемы и т. д.

Так при визуализации формальных физических моделей с помощью анимации может отображаться динамика процесса, производится построение графиков изменения физических величин и т. д. Визуальные модели обычно являются интерактивными, т. е. исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели.

Контрольные вопросы

1. Какие бывают модели? Приведите примеры материальных и информационных моделей.
2. Что такое формализация? Приведите примеры формальных моделей.

2.5. Основные этапы разработки и исследования моделей на компьютере

Использование компьютера для исследования информационных моделей различных объектов и процессов позволяет изучить их изменения в зависимости от значения тех или иных параметров. Процесс разработки моделей и их исследования на компьютере можно разделить на несколько основных этапов.

На первом этапе исследования объекта или процесса обычно строится **описательная информационная модель**. Такая модель выделяет существенные, с точки зрения целей проводимого исследования (целей моделирования), свойства объекта, а несущественными свойствами пренебрегает.

На втором этапе создается **формализованная модель**, т. е. описательная информационная модель записывается с помощью какого-либо формального языка. В такой модели с помощью формул, уравнений, неравенств и т. д. фиксируются формальные соотношения между начальными и конечными значениями свойств объектов, а также накладываются ограничения на допустимые значения этих свойств.

Однако далеко не всегда удается найти формулы, явно выражающие искомые величины через исходные данные. В таких случаях используются приближенные математические методы, позволяющие получать результаты с заданной точностью.

На третьем этапе необходимо формализованную информационную модель преобразовать в **компьютерную модель**, т. е. выразить ее на понятном для компьютера языке. Компьютерные модели разрабатывают преимущественно программисты, а пользователи могут проводить компьютерные эксперименты.

В настоящее время широкое распространение получили **компьютерные интерактивные визуальные модели**. В таких моделях исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели.

Контрольные вопросы

1. В каких случаях могут быть опущены отдельные этапы построения и исследования модели? Приведите примеры создания моделей в процессе обучения.

2.6. Исследование интерактивных компьютерных моделей



Далее мы рассмотрим ряд учебных интерактивных моделей, разработанных компанией ФИЗИКОН для образовательных курсов. Учебные модели компании ФИЗИКОН представлены на CD-дисках и в виде Интернет-проектов. Каталог интерактивных моделей содержит 342 модели по пяти предметам: физике (106 моделей), астрономии (57 моделей), математике (67 моделей), химии (61 модель) и биологии (51 модель). Часть моделей в Интернете на сайте <http://www.college.ru> интерактивны, а другие представлены только картинкой и описанием. Все модели вы найдете в соответствующих учебных курсах на CD-дисках.

2.6.1. Исследование физических моделей

Рассмотрим процесс построения и исследования модели на примере модели математического маятника, которая является идеализацией физического маятника.

Качественная описательная модель. Можно сформулировать следующие основные предположения:

- подвешенное тело значительно меньше по размеру длины нити, на которой оно подвешено;
- нить тонкая и нерастяжимая, масса которой пренебрежимо мала по сравнению с массой тела;
- угол отклонения тела мал (значительно меньше 90°);
- вязкое трение отсутствует (маятник колеблется в вакууме).

Формальная модель. Для формализации модели используем известные из курса физики формулы. Период T колебаний математического маятника равен:

$$T = 2\pi\sqrt{\frac{l}{g}}, \quad (2.1)$$

где l — длина нити, g — ускорение свободного падения.

Интерактивная компьютерная модель. Модель демонстрирует свободные колебания математического маятника. В полях можно изменять длину нити l , угол φ_0 начального отклонения маятника, коэффициент вязкого трения b .

2.3. Свободные колебания.

Модель 2.3. Математический маятник

Открытая физика.

Часть 1 (ЦОР на CD)



Запуск интерактивной модели математического маятника производится щелчком по кнопке *Старт*.

С помощью анимации показывается движение тела и действующие силы, строятся графики зависимости от времени угловой координаты или скорости, диаграммы потенциальной и кинетической энергий (рис. 2.2).

Это можно увидеть при свободных колебаниях, а также при затухающих колебаниях при наличии вязкого трения.

Обратите внимание, что колебания математического маятника являются гармоническими только при достаточно малых амплитудах.

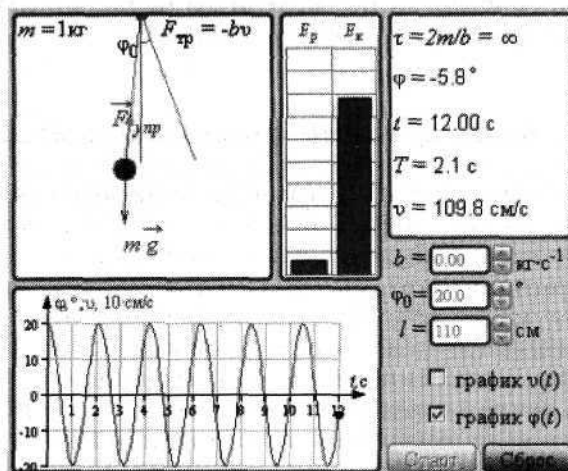


Рис. 2.2. Интерактивная модель математического маятника



Задание

для самостоятельного выполнения

<http://www.physics.ru>



2.1. *Практическое задание.* Провести компьютерный эксперимент с интерактивной физической моделью, размещенной в Интернете.

2.6.2. Исследование астрономических моделей

Рассмотрим гелиоцентрическую модель Солнечной системы.

Качественная описательная модель. Гелиоцентрическая модель мира Коперника на естественном языке формулировалась следующим образом:

- Земля вращается вокруг своей оси и Солнца;
- все планеты вращаются вокруг Солнца.

Формальная модель. Ньютон формализовал гелиоцентрическую систему мира, открыв закон всемирного тяготения и законы механики и записав их в виде формул:

$$F = \gamma \cdot \frac{m_1 \cdot m_2}{r^2} \quad \vec{F} = m \cdot \vec{a} . \quad (2.2)$$

Интерактивная компьютерная модель (рис. 2.3). Трехмерная динамическая модель показывает вращение планет Солнечной системы. В центре модели изображено Солнце, вокруг него — планеты Солнечной системы.

4.1.2. Вращение планет Солнечной системы. Модель 4.1.
Солнечная система

Открытая астрономия
(ЦОР на CD)



В модели выдержаны реальные отношения орбит планет и их эксцентриситеты. Солнце находится в фокусе орбиты каждой планеты. Обратите внимание на то, что орбиты Нептуна и Плутона пересекаются. Изобразить в небольшом окне все планеты сразу достаточно сложно, поэтому предусмотрены режимы *Меркурий...Марс* и *Юпитер...Плутон*, а также режим *Все планеты*. Выбор нужного режима производится при помощи соответствующего переключателя.

Во время движения можно менять значение угла зрения в окне ввода. Получить представление о реальных эксцентриситетах орбит можно, выставив значение угла зрения 90°.

Можно изменить внешний вид модели, отключив отображение названий планет, их орбит или системы координат, показываемой в левом верхнем углу. Кнопка *Старт* запускает модель, *Стоп* — приостанавливает, а *Сброс* — возвращает в исходное состояние.



Рис. 2.3. Интерактивная модель гелиоцентрической системы



Задание для самостоятельного выполнения

<http://www.college.ru> 

2.2. *Практическое задание.* Провести компьютерный эксперимент с интерактивной астрономической моделью, размещенной в Интернете.

2.6.3. Исследование алгебраических моделей

Формальная модель. В алгебре формальные модели записываются с помощью уравнений, точное решение которых основывается на поиске равносильных преобразований алгебраических выражений, позволяющих выразить переменную величину с помощью формулы.

Точные решения существуют только для некоторых уравнений определенного вида (линейные, квадратные, тригонометрические и др.), поэтому для большинства уравнений приходится использовать методы приближенного решения с заданной точностью (графические или численные).

Например, нельзя найти корень уравнения $\sin(x) = 3 \cdot x - 2$ путем равносильных алгебраических преобразований. Однако такие уравнения можно решать приближенно графическими и численными методами.

Построение графиков функций может использоваться для грубо приближенного решения уравнений. Для уравнений вида $f_1(x) = f_2(x)$, где $f_1(x)$ и $f_2(x)$ — некоторые непрерывные функции, корень (или корни) этого уравнения являются точкой (или точками) пересечения графиков функций.


Графическое решение таких уравнений можно осуществить путем построения интерактивных компьютерных моделей.

2.5.1. Функции и графики.

Модель 2.17.

Решение уравнений

Открытая математика.

Функции и графики 

(ЦОР на CD)

Интерактивная компьютерная модель. Введите в верхнее поле ввода уравнение в виде $f_1(x) = f_2(x)$, например, $\sin(x) = 3 \cdot x - 2$.

Нажмите кнопку *Решить*. Подождите некоторое время. Будет построен график правой и левой частей уравнения, зелеными точками будут отмечены корни.

Чтобы ввести новое уравнение, нажмите кнопку *Сброс*. Если вы сделаете ошибку при вводе, в нижнем окне появится соответствующее сообщение.

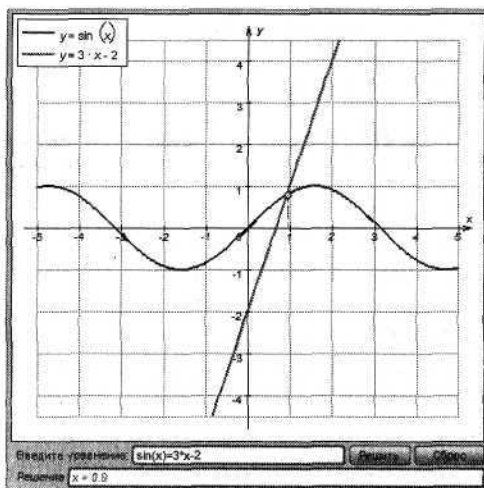


Рис. 2.4. Интерактивная компьютерная модель графического решения уравнений



Задание для самостоятельного выполнения

<http://www.mathematics.ru> 

2.3. Практическое задание. Провести компьютерный эксперимент с интерактивной математической моделью, размещенной в Интернете.

2.6.4. Исследование геометрических моделей (планиметрия)

Формальная модель. Треугольник ABC называется прямоугольным, если один из его углов (например, угол B) прямой (т. е. равен 90°). Сторона треугольника, противолежащая прямому углу, называется гипотенузой; две другие стороны — катетами.

Теорема Пифагора гласит, что в прямоугольном треугольнике сумма квадратов катетов равна квадрату гипотенузы: $AB^2 + BC^2 = AC^2$.

Интерактивная компьютерная модель (рис. 2.5). Интерактивная модель демонстрирует основные соотношения в прямоугольном треугольнике.

4.5. Прямоугольный треугольник. Открытая математика.

Модель 5.1.

Теорема Пифагора

Планиметрия 

(ЦОР на CD)

При помощи мыши можно перемещать точку A (в вертикальном направлении) и точку C (в горизонтальном направлении). Показываются длины сторон прямоугольного треугольника, градусные меры углов.

Переключившись в демонстрационный режим при помощи кнопки со значком кинопроектора, можно просмотреть анимацию. Кнопка *Start* запускает ее, кнопка *Stop* — приостанавливает, а кнопка *Сброс* возвращает анимацию в исходное состояние.

Кнопка со значком руки переводит модель обратно в интерактивный режим.

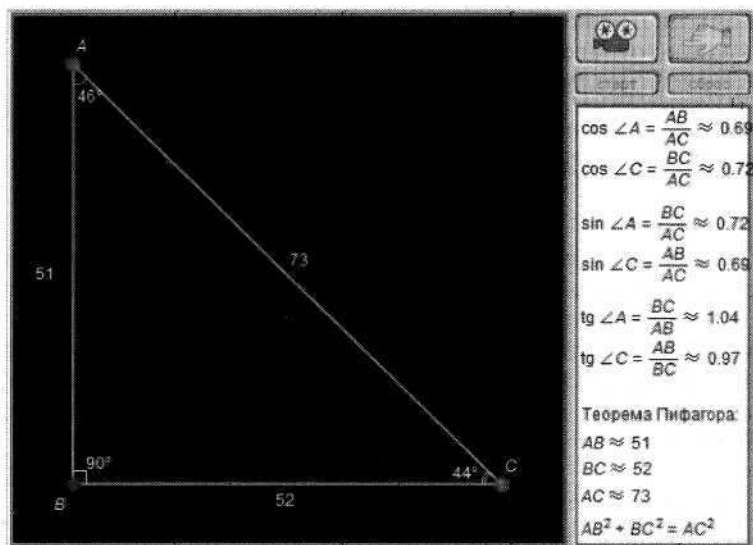


Рис. 2.5. Интерактивная математическая модель теоремы Пифагора



Задание для самостоятельного выполнения

<http://www.mathematics.ru>

2.4. *Практическое задание.* Провести компьютерный эксперимент с интерактивной планиметрической моделью, размещенной в Интернете.

2.6.5. Исследование геометрических моделей (стереометрия)

Формальная модель. Призма, основанием которой является параллелограмм, называется параллелепипедом. Противоположные грани любого параллелепипеда равны и параллельны. Прямоугольным называется параллелепипед, все грани которого прямоугольники. Прямоугольный параллелепипед с равными ребрами называется кубом.

Три ребра, выходящие из одной вершины прямоугольного параллелепипеда, называются его измерениями. Квадрат

диагонали прямоугольного параллелепипеда равняется сумме квадратов его измерений:

$$d^2 = a^2 + b^2 + c^2.$$

Объем прямоугольного параллелепипеда равен произведению его измерений:

$$V = a \cdot b \cdot c.$$

Интерактивная компьютерная модель. Перетаскивая мышью точки, можно изменять измерения параллелепипеда. Понаблюдайте, как изменяется длина диагонали, площадь поверхности и объем параллелепипеда при изменении длин его сторон. Флажок *Прямой* превращает произвольный параллелепипед в прямоугольный, а флажок *Куб* превращает его в куб.

4.6. Параллелепипед.
Модель 6.2.
Параллелепипед

Открытая математика.
Стереометрия
(ЦОР на CD)



Переключившись в демонстрационный режим при помощи кнопки со значком кинопроектора, можно просмотреть анимацию. Кнопка *Старт* запускает ее, кнопка *Стоп* — приостанавливает, а кнопка *Сброс* возвращает анимацию в исходное состояние. Кнопка со значком руки переводит модель обратно в интерактивный режим.

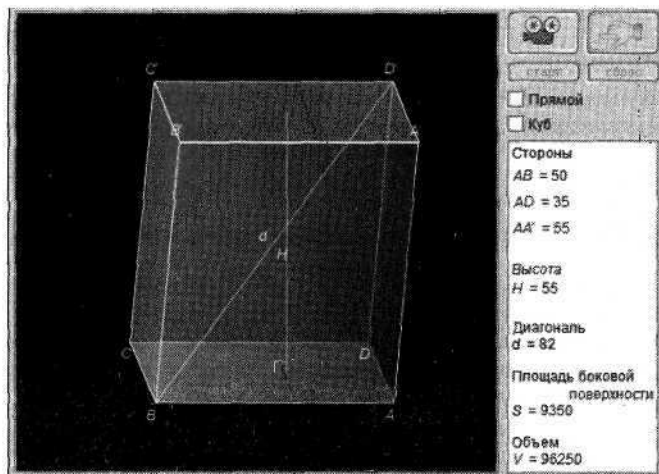


Рис. 2.6. Интерактивная стереометрическая модель параллелепипеда



Задание для самостоятельного выполнения

<http://www.mathematics.ru>

2.5. *Практическое задание.* Провести компьютерный эксперимент с интерактивной стереометрической моделью, размещенной в Интернете.

2.6.6. Исследование химических моделей

Рассмотрим процесс взаимодействия кислот и оснований.

Качественная описательная модель. Взаимодействие кислоты и основания в растворе приводит к образованию соли.

Формальная модель. Взаимодействие химических соединений записывается с помощью химических уравнений, отражающих соотношения всех реагирующих веществ. Это достигается с помощью коэффициентов перед формулами соединений. Запишем химическую формулу реакции, которая происходит между кислотой и основанием, в данном случае между серной кислотой и оксидом калия.



Интерактивная компьютерная модель (рис. 2.7). Эта модель схематически демонстрирует детали реакции нейтрализации. С помощью полей ввода с кнопками прокрутки можно выбрать вид кислоты и основания, которые будут участвовать в реакции.

1.5. Химические реакции. Модель 6.6. Открытая химия
Реакции кислот и оснований (ЦОР на CD)

С помощью кнопки *Старт* реакция запускается, можно пронаблюдать, как кислые и основные ионы образуют соль. От кислоты отщепляется водород, образуя вместе с молекулой воды положительный ион гидроксония. Затем этот ион взаимодействует с гидроксильной группой, отщепленной от основания, в результате чего образуются две молекулы воды.

Кнопка *Стоп* приостанавливает процесс, а кнопка *Сброс* возвращает модель к начальному состоянию.

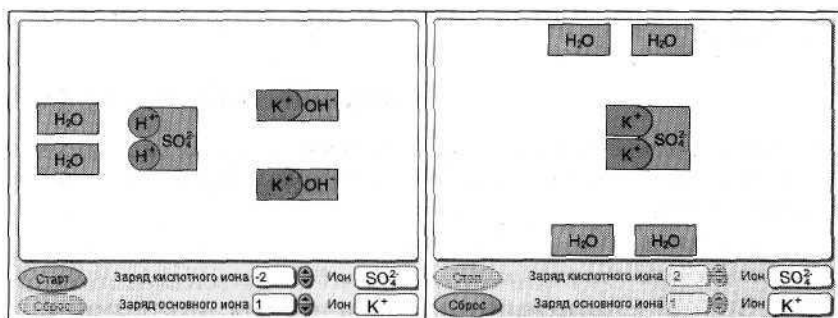


Рис. 2.7. Интерактивная модель химической реакции кислоты и основания



Задание для самостоятельного выполнения

<http://www.chemistry.ru> 

2.6. *Практическое задание.* Провести компьютерный эксперимент с интерактивной химической моделью, размещенной в Интернете.

2.6.7. Исследование биологических моделей

Рассмотрим процесс репликации (копирования) ДНК.

Качественная описательная модель. Нуклеотиды аденин, тимин, гуанин и цитозин являются «кирпичиками» (генетическим алфавитом), из которых строится ДНК, содержащая генетическую информацию организма. Клетки организма данного вида (даже принадлежащие разным тканям) содержат ДНК с одинаковой нуклеотидной последовательностью, и эта последовательность не зависит ни от питания, ни от окружающей среды, ни от возраста организма. При этом нуклеотидный состав ДНК, а значит и генетическая информация, разных видов различен.

Формальная модель. Репликация (копирование) ДНК в ходе деления клеток начинается с разделения двух цепей,

каждая из которых становится матрицей, синтезирующей нуклеотидную последовательность новых цепей. Специальные белки-ферменты расплетают ДНК, удерживают матрицу в разведенном состоянии и вращают молекулу ДНК.

Правильность репликации обеспечивается точным соответствием комплементарных пар оснований. Азотистые основания нуклеотидов могут соединяться друг с другом только в определенном порядке (аденин с тимином, гуанин — с цитозином).

Репликация (копирование) катализируется несколькими ДНК-полимеразами. После репликации дочерние спирали закручиваются обратно уже без затрат энергии и каких-либо ферментов.

Интерактивная компьютерная модель (рис. 2.8). В этой модели показан синтез комплементарной цепи ДНК. Азотистые основания (аденин, тимин, гуанин и цитозин) обозначены условными значками разного цвета. Цепь синтезируется слева направо. Реакция обеспечивается ДНК-полимеразой, перемещающейся вдоль цепи.

8.2.2. Репликация и транскрипция ДНК	Открытая биология
Модель 8.5. Синтез комплементарной цепи ДНК	(ЦОР на CD)

В процессе репликации требуется, чтобы обеспечивалась комплементарность нуклеотидов (аденин соединяется только с тимином, а гуанин — с цитозином). Это придуманный природой способ избежать ошибок при синтезе двойной спирали ДНК.


Кнопка *Старт* позволяет запустить интерактивную модель, кнопка *Стоп* — приостановить анимацию, кнопка *Сброс* — вернуть модель в исходное состояние.



Рис. 2.8. Интерактивная модель репликации ДНК



Задание для самостоятельного выполнения








<http://www.biology.ru> 

- 2.7. *Практическое задание.* Провести компьютерный эксперимент с интерактивной биологической моделью, размещенной в Интернете.

Глава 3

Базы данных. Системы управления базами данных (СУБД)

При изучении данной главы рекомендуется установить следующее программное обеспечение для операционных систем Windows и Linux:

	<ul style="list-style-type: none">• Система управления базами данных (СУБД) OpenOffice Base;• программа составления генеалогических деревьев GenoPro (Живая Родословная).	Windows-CD  	
	<ul style="list-style-type: none">• Система управления базами данных (СУБД) OpenOffice Base.	Linux-DVD 	

3.1. Табличные базы данных

Базы данных. Любой из нас, начиная с раннего детства, многократно сталкивался с «базами данных». Это — всевозможные справочники (например, телефонный), энциклопедии и т. п. Записная книжка — это тоже «база данных», которая есть у каждого из нас.

Базы данных представляют собой информационные модели, содержащие данные об объектах и их свойствах. Базы данных хранят информацию о группах объектов с одинаковым набором свойств.

Например, база данных «Записная книжка» хранит информацию о людях, каждый из которых имеет фамилию,

имя, телефон и т. д. Библиотечный каталог хранит информацию о книгах, каждая из которых имеет название, автора, год издания и т. д.

Информация в базах данных хранится в упорядоченном виде. Так, в записной книжке все записи упорядочены по алфавиту, а в библиотечном каталоге либо по алфавиту (алфавитный каталог) или по области знания (предметный каталог).



База данных позволяет упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

Табличные базы данных. Базу данных, хранящую данные о группе объектов с одинаковыми свойствами, удобно представлять в виде двумерной таблицы: в каждой ее строке последовательно размещаются значения свойств одного из объектов; каждое значение свойства — в своем столбце, озаглавленном именем свойства.

Столбцы такой таблицы называют **полями**; каждое поле характеризуется своим именем (именем соответствующего свойства) и типом данных, представляющих значения данного свойства.



Поле базы данных — это столбец таблицы, содержащий значения определенного свойства.

Строки таблицы являются **записями** об объекте; эти записи разбиты на поля столбцами таблицы, поэтому каждая запись представляет набор значений, содержащихся в полях.



Запись базы данных — это строка таблицы, содержащая набор значений свойств, размещенный в полях базы данных.

Каждая таблица должна содержать, по крайней мере, одно **ключевое поле**, содержимое которого уникально для

каждой записи в этой таблице. Ключевое поле позволяет однозначно идентифицировать запись в таблице.



Ключевое поле — это поле, значения которого однозначно определяют запись в таблице.

В качестве ключевого поля чаще всего используют поле, содержащее тип данных *счетчик*. Однако иногда удобнее в качестве ключевого поля таблицы использовать другие поля, код объекта, например инвентарный номер, и т. п.

Тип поля. Тип поля определяется типом данных, которые оно содержит. Поля могут содержать данные следующих основных типов:

- *Счетчик.* Содержит последовательность целых чисел, которые задаются автоматически при вводе записей. Эти числа не могут быть изменены пользователем.
- *Текстовый.* Содержит символы различных типов.
- *Числовой.* Содержит числа различных типов.
- *Дата/Время.* Содержит даты или время.
- *Картинка.* Содержит изображения.
- *Логический.* Содержит значения *Истина (Да)* или *Ложь (Нет)*.

Каждый тип поля имеет свой набор свойств. Наиболее важными свойствами полей являются:

- *Размер поля.* Определяет максимальную длину текстового или числового поля.
- *Формат поля.* Устанавливает формат данных.
- *Обязательное поле.* Указывает на то, что данное поле обязательно надо заполнить.

База данных «Процессоры». Рассмотрим, например, базу данных «Процессоры» (табл. 3.1), которая содержит перечень объектов (процессоров), каждый из которых имеет имя (название). В качестве характеристик (свойств) можно рассмотреть количество элементов в процессоре и частоту. Поле *Процессор* являются текстовым, а поля *Кол-во элементов*, *Частота* и, естественно, *Счетчик* — числовыми полями.

При этом каждое поле обладает определенным набором свойств. Например, для поля *Счетчик* задан формат данных *целое число*.

Таблица 3.1. Табличная база данных «Процессоры»

Счетчик	Процессор	Кол-во элементов	Частота, МГц
1	80286	120000	25
2	Pentium	3100000	266
3	Pentium 4	42000000	3200
4	Pentium Extreme Edition	376000000	3700

Контрольные вопросы

1. В чем заключается разница между записью и полем в табличной базе данных?
2. Поля каких типов могут присутствовать в базе данных?
3. Чем отличается ключевое поле от остальных полей?

3.2. Система управления базами данных

3.2.1. Основные объекты СУБД: таблицы, формы, запросы, отчеты

Системы управления базами данных (СУБД). Развитие информационных технологий привело к созданию компьютерных баз данных. Создание баз данных, а также операции поиска и сортировки данных выполняются специальными программами — **системами управления базами данных (СУБД)**. Таким образом, необходимо различать собственно базы данных (БД), которые являются упорядоченными наборами данных, и системы управления базами данных (СУБД) — программы, управляющие хранением и обработкой данных.



Система управления базами данных (СУБД) — это программа, позволяющая создавать базы данных, а также обеспечивающая обработку (сортировку) и поиск данных.

В СУБД Microsoft Access и OpenOffice Base используется стандартный для операционных систем многооконный интерфейс, но в отличие от других приложений, не многодокументный. Единновременно может быть открыта только одна база данных, содержащая обязательное *окно базы данных* и *окна для работы с объектами базы данных*. В каждый момент времени одно из окон является активным и в нем курсором отмечается активный объект.

Окно базы данных — один из главных элементов интерфейса СУБД. Здесь систематизированы все объекты базы данных: *таблицы, запросы, формы, отчеты*.

Таблицы. В базах данных вся информация хранится в двумерных таблицах. Это *базовый* объект базы данных, все остальные объекты создаются на основе существующих таблиц (производные объекты). Каждая строка в таблице — *запись* базы данных, а столбец — *поле*. Запись содержит набор данных об одном объекте, а поле — однородные данные обо всех объектах.

Запросы. В СУБД запросы являются важнейшим инструментом. Главное предназначение запросов — это отбор данных на основании заданных условий.

Формы. Формы позволяют отображать данные, содержащиеся только в одной записи. При помощи форм можно добавлять в таблицы новые данные, а также редактировать или удалять существующие. Форма может содержать рисунки, графики и другие внедренные объекты.

Отчеты. Они предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде.

Существует достаточно много различных СУБД, но для первого знакомства мы рекомендуем СУБД OpenOffice Base.

Контрольные вопросы

1. Какую функцию выполняют СУБД?
2. Можно ли открыть в СУБД несколько разных баз данных?
3. Какие основные объекты входят в СУБД, и какие функции они выполняют?

Практическая работа 3.1

Создание табличной базы данных

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux.

Цель работы. Научиться создавать табличные базы данных в системе управления базами данных (СУБД).

Задание. В СУБД OpenOffice Base создать табличную базу данных «Процессоры» (см. табл. 3.1).

Варианты выполнения работы:

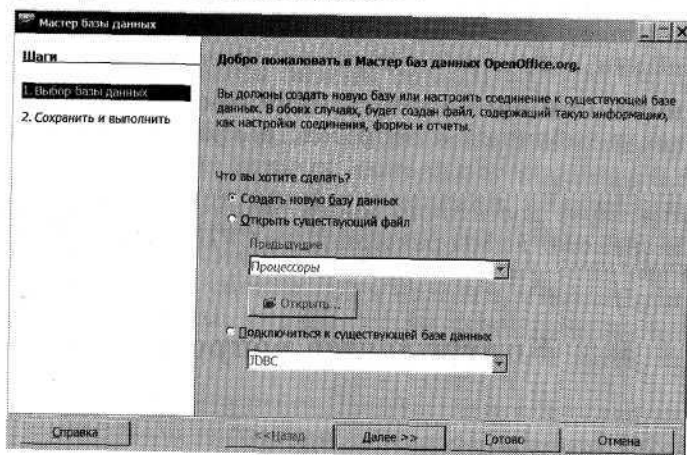
- включить в табличную базу данных различные поля;
- заполнить табличную базу данных различными записями.



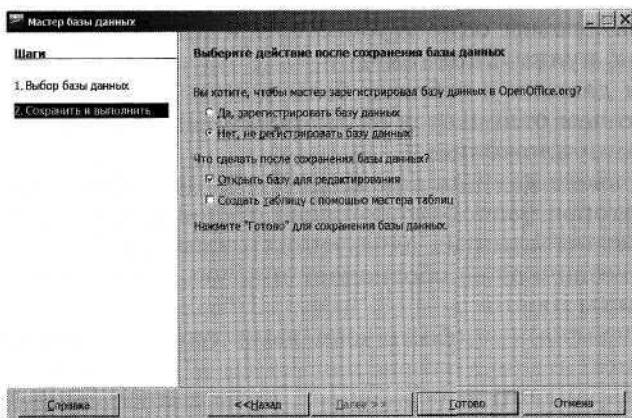
Создание табличной базы данных «Процессоры»



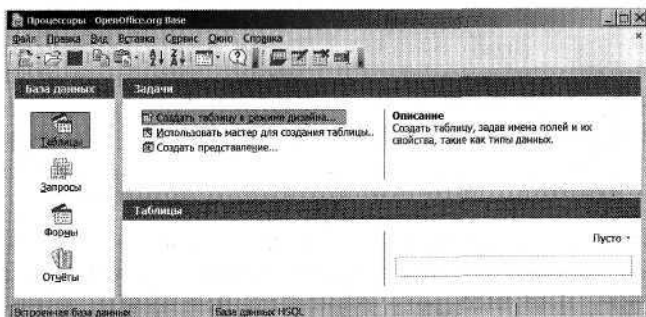
1. В операционной системе Windows или Linux запустить СУБД OpenOffice Base командой [Пуск-Программы-OpenOffice-OpenOffice Base]
2. В появившемся диалоговом окне *Мастер базы данных* (шаг 1) выбрать с помощью переключателя пункт *Создать новую базу данных*. Щелкнуть по кнопке *Далее >*.



3. В появившемся диалоговом окне *Мастер базы данных* (шаг 2) установить флажок *Открыть базу для редактирования*. Щелкнуть по кнопке *Готово*.

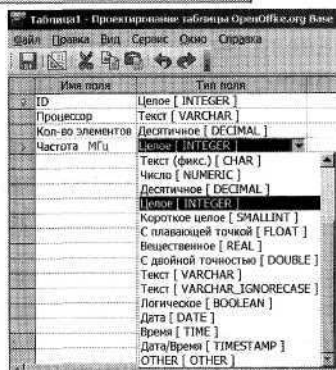


- Откроется диалоговое окно *Сохранить как*, в котором выбрать место сохранения и имя базы данных («Процессоры»).
- В левой части появившегося диалогового окна *Процессоры* в разделе *Базы данных* выбрать пункт *Таблицы*, в средней части окна в разделе *Задачи* выбрать пункт *Создать таблицу в режиме дизайна...*



- В появившемся окне *Таблица 1* в столбце *Имя поля* ввести имена полей базы данных «Процессоры».

В столбце *Тип поля* с помощью раскрывающегося списка выбрать для каждого поля его тип.



7. В диалоговом окне *Процессоры* в разделе *Таблицы* появится пункт *Таблица 1*.

Ввод данных в таблицу базы данных и их редактирование мало чем отличается от аналогичных действий в других офисных приложениях.

При вводе данных в режиме *Таблица* в поле маркера записи, которое расположено слева от полей таблицы, может отображаться один из следующих символов:

- * (звездочка) — обозначает пустую запись в конце таблицы;
- ▶ (стрелка) — обозначает выделенную (активную) запись;
- ✍ (карандаш) — обозначает, что в записи были сделаны изменения.

8. Осуществить двойной щелчок по пункту *Таблица 1* и в появившемся диалоговом окне *Процессоры: Таблица 1* ввести записи базы данных «Процессоры».

ID	Процессор	Кол-во эл...	Частота МГц
1	80286	120000	25
2	Pentium	3100000	266
3	Pentium 4	42000000	3200
4	Pentium Extreme Edition	376000000	3700

Запись 1 из 4

Перемещение между записями можно осуществлять с помощью мыши, клавиш управления курсором или полосы прокрутки. Для быстрого перемещения между записями в базе данных можно использовать кнопки перемещения на панели *Запись*, которая находится в нижней части окна таблицы.

3.2.2. Использование формы для просмотра и редактирования записей в табличной базе данных

Записи базы данных можно просматривать и редактировать в виде **таблицы** или в виде **формы**. Выше мы работали с базой данных, представленной в виде таблицы, когда запись образует строку в этой таблице. Такое представление базы

данных позволяет наблюдать несколько записей одновременно, и в этом состоит достоинство табличного представления.

Однако часто вид *Таблица* не позволяет видеть полностью всю информацию на экране. Если база данных содержит достаточно много полей, а значения полей содержат много символов, то не все поля таблицы могут уместиться на экране, а значения полей могут быть видны не полностью.

Форма отображает одну запись в удобном для пользователя виде. В процессе создания формы можно указать, какие поля базы данных включить в форму, как расположить поля в окне формы, а также как можно сделать форму визуально привлекательной.

Фактически с помощью формы создается графический интерфейс доступа к базе данных, который может содержать различные элементы управления (*текстовые поля, кнопки, переключатели* и т. д.), а также *надписи*. Обычно на форме размещаются *надписи*, являющиеся именами полей базы данных, и *текстовые поля*, содержащие данные из базы данных.

Пользователь может изменять дизайн формы (размер, цвет и т. д.), элементов управления и надписей.

Примерами форм могут являться *Визитка* в базе данных «Записная книжка» или *Карточка* в базе данных «Библиотечный каталог», которые содержат лишь одну запись базы данных, зато представленную в удобном для пользователя виде.

Контрольные вопросы

1. В чем заключается разница между представлением табличной базы данных с помощью таблицы и формы?

Практическая работа 3.2

Создание формы в табличной базе данных

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux.

Цель работы. Научиться создавать формы для табличных баз данных.

Задание. В СУБД OpenOffice Base создать форму для базы данных «Процессоры».

Варианты выполнения работы:

задать различные дизайны формы и расположение полей и надписей на форме.



Создание формы для базы данных «Процессоры»

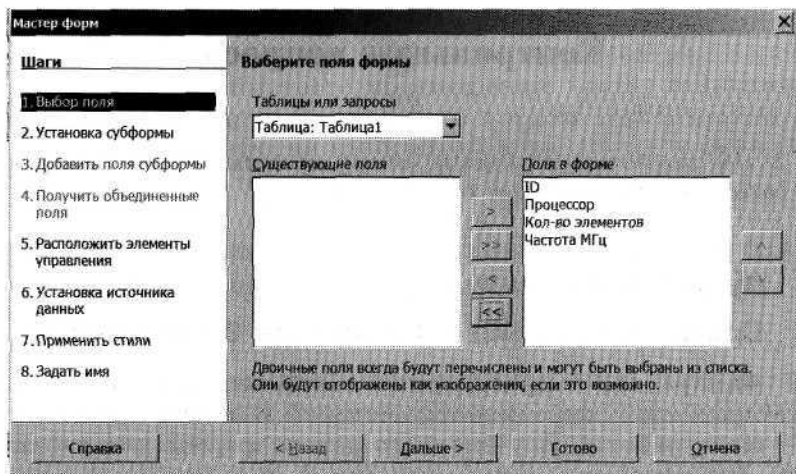


Создание формы можно проводить различными способами с использованием:

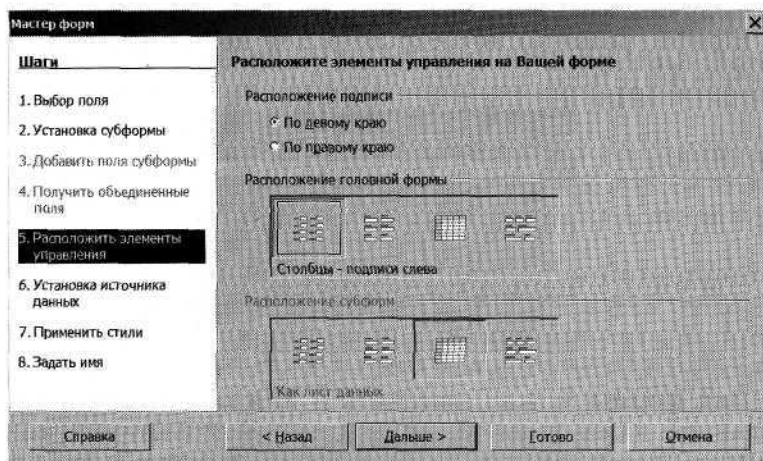
- самостоятельно (в *Режиме разработки*) — этот способ позволяет начать создание формы «с нуля»;
- с использованием *Мастера форм* (более простой путь), который с помощью серии диалоговых окон помогает пользователю в создании формы.

Создадим теперь форму для базы данных «Процессоры» с помощью *Мастера форм*.

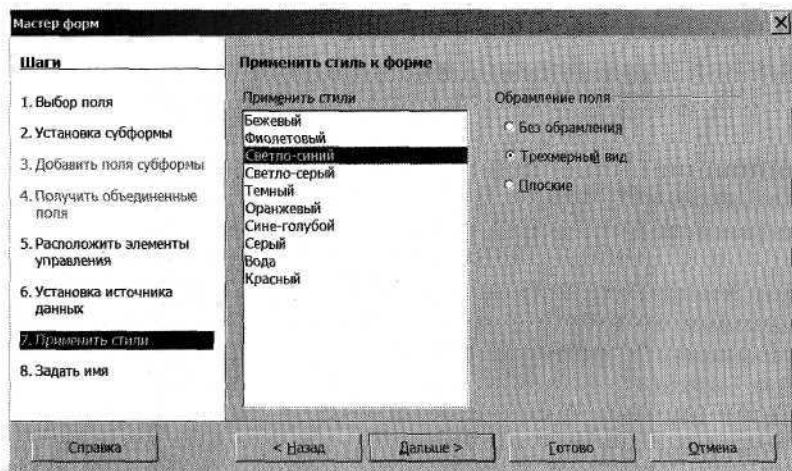
1. В левой части появившегося диалогового окна *Процессоры* в разделе *Базы данных* выбрать пункт *Формы*, в средней части окна в разделе *Задачи* выбрать пункт *Использовать мастер для создания формы ...*
2. В появившемся диалоговом окне *Мастер форм* (шаг 1: *Выбор поля*) выбрать поля базы данных из *Таблицы 1*, которые будут отображаться на форме. Щелкнуть по кнопке *Дальше >*.



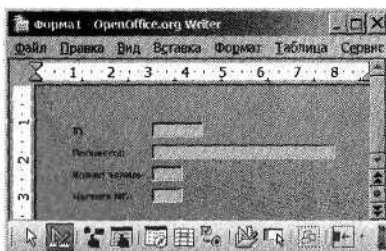
3. В диалоговом окне *Мастер форм* (шаг 5: *Расположить элементы управления*) выбрать расположение полей, которые будут отображаться на форме.
Щелкнуть по кнопке *Дальше >*.



4. В диалоговом окне *Мастер форм* (шаг 6: *Установка источника данных*) установить переключатель в положение *Форма для отображения всех данных*.
Щелкнуть по кнопке *Дальше >*.
5. В диалоговом окне *Мастер форм* (шаг 7: *Применить стили*) выбрать стиль и обрамление поля.
Щелкнуть по кнопке *Дальше >*.

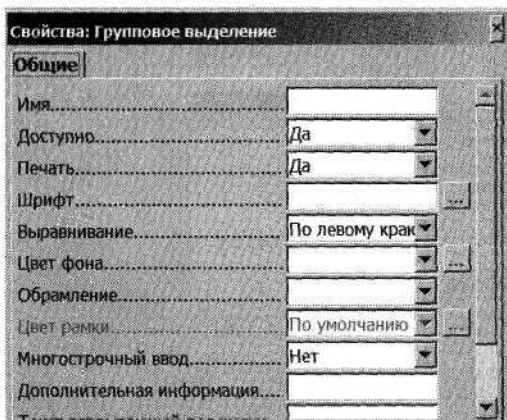


6. В диалоговом окне *Мастер форм* (шаг 8: *Задать имя*) ввести название формы (например, *Форма 1*) и установить переключатель в положение *Модифицировать форму*. Щелкнуть по кнопке *Готово*.
7. Получим заготовку формы, в которой слишком мелкий шрифт, размеры некоторых полей не вмещают данные и т. д.



Вид формы можно изменять в *Режиме разработки*. В созданной нами форме базы данных «Компьютеры» не все надписи хорошо видны, поэтому размер шрифта нужно увеличить. Некоторые поля вывода числовых значений также нужно увеличить, чтобы числовые данные в них помещались.

8. В режиме разработки (необходимо щелкнуть по кнопке *Режим разработки*, в нижнем ряду второй слева). Выделить поле с надписью и осуществить двойной щелчок. В появившемся диалоговом окне *Свойства* установить необходимые параметры (в том числе размер шрифта). Прodelать это для всех полей.



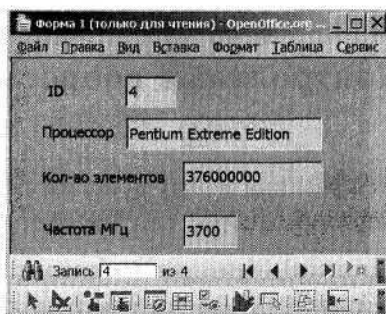
Для изменения параметров надписи и поля по отдельности их нужно разгруппировать.

9. Выделить надпись и поле, в контекстном меню выбрать пункт [*Группировка-Разгруппировать*]. Прodelать это для всех полей.

Теперь форма для базы данных «Процессоры» готова. С ее помощью можно просматривать записи, редактировать их или вводить новые записи.

После открытия форма содержит запись № 1. При работе с формой для перехода от одной записи к другой необходимо воспользоваться панелью *Запись*, которая находится в нижней части окна формы.

10. В окне *Форма 1* щелкнуть по кнопке *Режим разработки*. С использованием панели *Запись*, которая содержит кнопки со стрелками, можно перемещаться по записям. Также можно в поле номера записи ввести номер иско-мой записи.



3.2.3. Поиск записей в табличной базе данных с помощью фильтров и запросов

Поиск записей с помощью фильтров. Фильтры позволяют отбирать записи, которые удовлетворяют заданным условиям. Условия отбора записей создаются с использованием операторов сравнения (=, >, < и т. д.).

Простой фильтр содержит условие отбора записей только для одного поля. **Сложный фильтр** содержит несколько условий для различных полей. В результате применения сложного фильтра будут отобраны только те записи, которые удовлетворяют всем условиям одновременно. Можно сказать, что условия в сложных фильтрах связаны между собой операцией логического умножения.

Поиск записей с помощью запросов. Запросы осуществляют поиск записей в табличной базе данных так же, как и фильтры. Различие между ними состоит в том, что запрос является самостоятельным объектом базы данных, а фильтр привязан к конкретной таблице.

Запрос является производным объектом от таблицы. Однако результатом выполнения запроса является также таблица, т. е. запросы могут использоваться вместо таблиц. Например, форма может быть создана как для таблицы, так и для запроса.

Запросы позволяют отобрать те записи, которые удовлетворяют заданным условиям. Запросы, как и фильтры, бывают простые и сложные. **Простой запрос** содержит одно условие, а **сложный запрос** содержит несколько условий для различных полей.

В процессе создания запроса можно отбирать не только записи, но и поля, которые будут присутствовать в запросе.

Контрольные вопросы

1. В чем заключается разница между поиском записей в табличной базе данных с помощью фильтров и запросов?

Практическая работа 3.3

Поиск записей в табличной базе данных с помощью фильтров и запросов

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux.

Цель работы. Научиться осуществлять поиск записей в табличной базе данных с использованием фильтров и запросов.

Задание 1. В табличной базе данных «Процессоры» осуществить поиск записей с использованием простого фильтра (например, будем искать записи, удовлетворяющие условию: частота процессора больше 300 МГц).

Задание 2. В табличной базе данных «Процессоры» осуществить поиск записей с использованием сложного запроса.

Варианты выполнения работы:

задать различные условия поиска записей в фильтрах и запросах.



Поиск записей с использованием простого фильтра в базе данных «Процессоры»




1. Открыть *Таблицу 1* базы данных «Процессоры», дважды щелкнув по соответствующей ссылке в окне базы данных.

Появится таблица базы данных с кнопками на *Панели инструментов*, позволяющими создавать и отменять фильтры.



ID	Процессор	Кол-во элементов	Частота МГц
1	80286	120000	25
2	Pentium	3100000	266
3	Pentium 4	42000000	3200
4	Pentium Extreme Edition	376000000	3700


Запись 5 из 5

2. Щелкнуть по кнопке , позволяющей выбрать параметры фильтра. В появившемся диалоговом окне *Фильтр по умолчанию* ввести условия поиска записей (*Имя поля*, *Условие* и *Значение*). Фильтр будет создан. Щелкнуть по кнопке *ОК*.

В появившемся окне *Процессоры: Таблица 1* будут выведены записи, удовлетворяющие условиям поиска. В данном случае — это записи 3 и 4.

Оператор	Имя поля	Условие	Значение
	Частота МГц	>	300
AND	- нет -		
AND	- нет -		

Кнопки: ОК, Отмена, Справка

3. Для того чтобы отменить фильтр и вернуть все записи базы данных, щелкнуть по кнопке .



Поиск записей с использованием сложного запроса в базе данных «Процессоры»

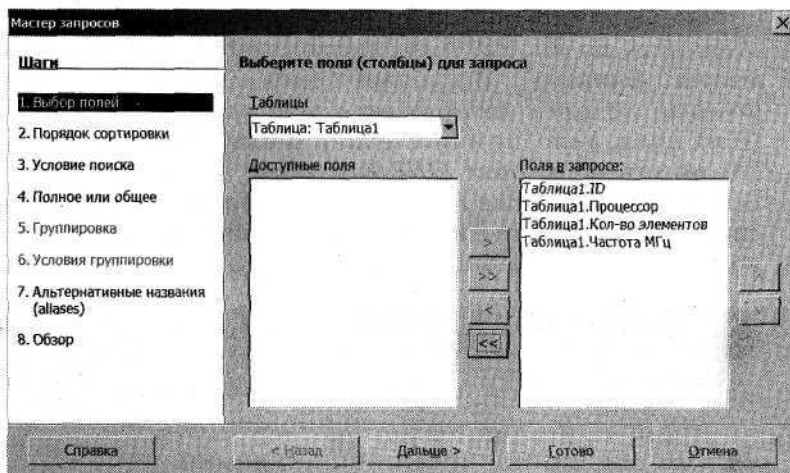


Создание запроса можно проводить различными способами:

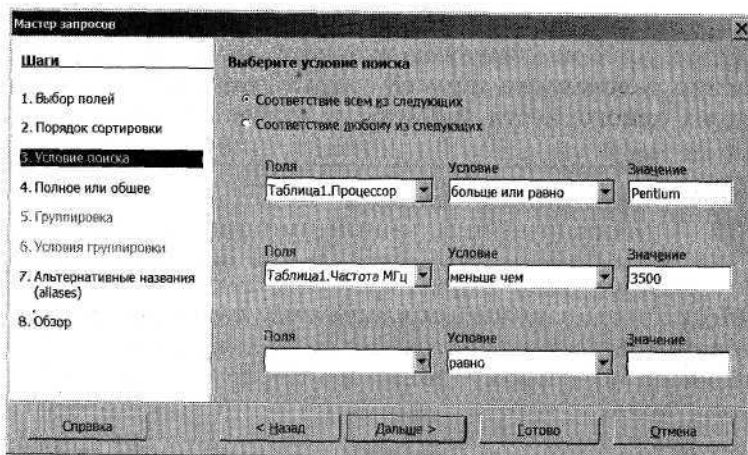
- самостоятельно (в *Режиме разработки*) — этот способ позволяет начать создание запроса с нуля.
- с использованием *Мастера запросов* (более простой путь), который с помощью серии диалоговых окон помогает пользователю в создании запроса.

Создадим сложный запрос для табличной базы данных «Процессоры» с помощью *Мастера запросов*.

1. В левой части диалогового окна *Процессоры* в разделе *Базы данных* выбрать пункт *Запросы*, в средней части окна в разделе *Задачи* выбрать пункт *Использовать мастер для создания запроса ...*
2. В появившемся диалоговом окне *Мастер запросов* (шаг 1: *Выбор полей*) выбрать поля базы данных из *Таблицы 1*, которые будут отображаться в запросе. Щелкнуть по кнопке *Дальше >*.



3. В диалоговом окне *Мастер запросов* (шаг 3: *Условие поиска*) выбрать поля базы данных из *Таблицы 1*, *Условия* и *Значения*, которые будут в запросе. Щелкнуть по кнопке *Готово*.



В появившемся окне *Процессоры: Запрос_Таблица 1* будут выведены записи, удовлетворяющие условиям поиска. В данном случае это записи 2 и 3.

ID	Процессор	Кол-во элементов	Частота МГц
2	Pentium	3100000	266
3	Pentium 4	42000000	3200

Данная таблица, полученная в результате запроса, сохраняется в базе данных «Процессоры» как самостоятельный объект.

3.2.4. Сортировка записей в табличной базе данных

Базы данных могут содержать сотни и тысячи записей. Часто бывает необходимо их упорядочить, т. е. расположить в определенной последовательности. Упорядочение записей называется сортировкой.

Сортировка записей производится по какому-либо полю. Значения, содержащиеся в этом поле, располагаются в определенном порядке, который определяется типом поля:

- по алфавиту, если поле текстовое;
- по величине числа, если поле числовое;
- по дате, если тип поля Дата/Время и т. д.

Сортировка записей может производиться либо по возрастанию, либо по убыванию значений полей. В процессе сортировки целостность записей сохраняется, т. е. они переносятся из одного места таблицы в другое целиком.



Сортировка записей базы данных — это их упорядочение по значениям одного из полей.

Могут реализовываться **вложенные сортировки**. Они последовательно производятся по нескольким полям. После сортировки по первому полю производится сортировка по второму полю и т. д.

Контрольные вопросы



1. Меняет ли сортировка в табличной базе данных содержание записей?

Практическая работа 3.4

Сортировка записей в табличной базе данных

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux.

Цель работы. Научиться осуществлять сортировку записей в табличной базе данных.

Задание. В табличной базе данных «Процессоры» осуществить сортировку записей по разным полям (например, по полю *Процессоры*).

Варианты выполнения работы:

выбрать различные поля для сортировки записей в табличной базе данных.




Задание. Сортировка записей в табличной базе данных «Процессоры»



1. Открыть *Таблицу 1* базы данных «Процессоры», дважды щелкнув по соответствующей ссылке в окне базы данных.

Появится таблица базы данных с кнопками на *Панели инструментов*, позволяющими сортировать записи по возрастанию и убыванию.



2. Выделить поле базы данных, по которому будет вестись сортировка (например, *Процессор*). Осуществить сортировку записей по убыванию, щелкнув по кнопке  на *Панели инструментов*.
3. Результатом будет отсортированная база данных.

ID	Процессор	Кол-во элементов	Частота МГц
4	Pentium Extreme Edition	376000000	3700
3	Pentium 4	42000000	3200
2	Pentium	3100000	266
1	80286	120000	25

3.2.5. Печать данных с помощью отчетов

Можно осуществлять печать непосредственно таблиц, форм и запросов с помощью команды [*Файл-Печать*]. Однако для красивой печати документов целесообразно использовать отчеты. Отчеты являются производными объектами базы данных и создаются на основе таблиц, форм и запросов.

Контрольные вопросы

1. Для чего нужны отчеты в табличной базе данных?

Практическая работа 3.5

Создание отчета в табличной базе данных

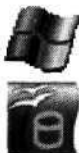
Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows или Linux.

Цель работы. Научиться создавать отчеты в табличной базе данных.

Задание. В табличной базе данных «Процессоры» создать отчет.

Варианты выполнения работы:

задать различные дизайны отчетов в табличных базах данных.



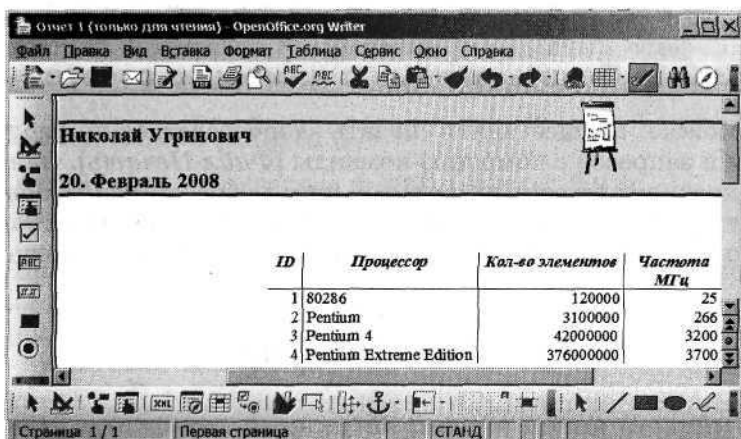
Создание отчета в табличной базе данных
«Процессоры»



Создадим отчет для базы данных «Процессоры» с помощью *Мастера отчетов*.

1. В левой части появившегося диалогового окна *Процессоры* в разделе *Базы данных* выбрать пункт *Отчеты*, в средней части окна в разделе *Задачи* выбрать пункт *Использовать мастер для создания отчета ...*
2. В диалоговых окнах *Мастер отчетов* выполнить шаги 1–6 самостоятельно.

Будет получен один из вариантов отчета.



База данных «Процессоры»
хранится в папке ..\ИКТ11bazi

Windows-CD

3.3. Иерархические базы данных

Иерархическая база данных графически может быть представлена как перевернутое дерево, состоящее из объектов различных уровней. Верхний уровень (*корень*) занимает один объект, второй — объекты второго уровня и т. д.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении **предка** (объект более близкий к корню) к **потомку** (объект, более низкого уровня), при этом объект-предок может не иметь потомков или иметь их несколько, тогда как объект-потомок обязательно имеет только одного предка. Объекты, имеющие общего предка, называются **близнецами**.

Иерархическая база данных Папки Windows. Иерархической базой данных является каталог папок Windows (например, гибкого диска). Ее можно увидеть, например, запустив файловый менеджер Total Commander, и введя команду [*Вид-Дерево каталогов*]. Верхний уровень занимает корневая папка диска A:. На втором уровне находятся папки Документы, Изображения и Школа, которые являются потомками корневой папки диска A:, а между собой являются близнецами. В свою очередь, папка Изображения является предком по отношению к папке третьего уровня Фото. Наконец, файл Класс.bmp является потомком папки Фото (рис. 3.1).

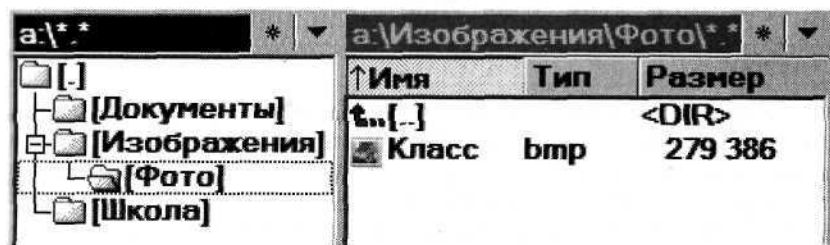


Рис. 3.1. Иерархическая база данных каталога папок диска A:

Иерархическая база данных Системный реестр Windows. Иерархической базой данных является также системный реестр Windows (рис. 3.2), в котором хранится вся информация, необходимая для нормального функционирования компьютерной системы (данные о конфигурации компьютера и установленных драйверах, сведения об установленных программах, настройки графического интерфейса и др.).

Содержание реестра автоматически обновляется при установке нового оборудования, инсталляции программ и т. д.

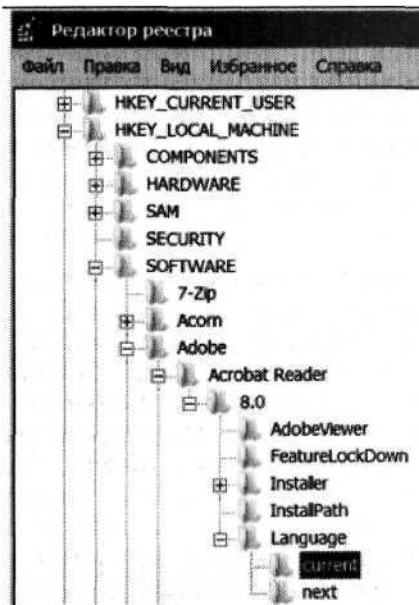


Рис. 3.2. Иерархическая база данных «Системный реестр Windows»

Иерархическая базы данных «Доменная система имен».

Еще одним примером иерархической базы данных является доменная система имен подключенных к Интернету компьютеров. На верхнем уровне находится табличная база данных, содержащая перечень доменов верхнего уровня (всего 269 домена), из которых 12 — административные, а остальные 257 — географические. Наиболее многочисленным доменом (данные на январь 2008 года) является административный домен net (около 190 миллионов серверов), а некоторых доменах (например, в географическом домене Zr) до сих пор не зарегистрировано ни одного сервера.

На втором уровне находятся табличные базы данных, содержащие перечень доменов второго уровня для каждого домена первого уровня.

На третьем уровне могут находиться табличные базы данных, содержащие перечень доменов третьего уровня для каждого домена второго уровня и таблицы, содержащие IP-адреса компьютеров, находящихся в домене второго уровня.



Рис. 3.3. Иерархическая база данных *Доменная система имен*

База данных «Доменная система имен» должна содержать записи обо всех компьютерах, подключенных к Интернету, т. е. более 500 миллионов записей. Размещение такой огромной базы данных на одном компьютере сделало бы поиск информации очень медленным и неэффективным. Решение этой проблемы было найдено путем размещения отдельных составных частей базы данных на различных DNS-серверах. Таким образом, иерархическая база данных «Доменная система имен» является **распределенной базой данных**.

Поиск информации в такой иерархической распределенной базе данных ведется следующим образом. Например, мы хотим ознакомиться с содержанием WWW-сервера фирмы Microsoft.

Сначала наш запрос, содержащий доменное имя сервера `www.microsoft.com`, будет отправлен на DNS-сервер нашего

провайдера, который переадресует его на DNS-сервер самого верхнего уровня базы данных. В таблице первого уровня будет найден интересующий нас домен com и запрос будет адресован на DNS-сервер второго уровня, который содержит перечень доменов второго уровня, зарегистрированных в домене com.

В таблице второго уровня будет найден домен microsoft и запрос будет переадресован на DNS-сервер третьего уровня. В таблице третьего уровня будет найдена запись, соответствующая доменному имени, содержащемуся в запросе. Поиск информации в базе данных «Доменная система имен» будет завершен и начнется поиск компьютера в сети по его IP-адресу.

Контрольные вопросы

1. В чем заключаются характерные особенности иерархических баз данных?
2. Чем различаются между собой иерархические и распределенные базы данных?

3.4. Сетевые базы данных

Сетевая база данных является обобщением иерархической за счет допущения объектов, имеющих более одного предка, т. е. каждый элемент вышестоящего уровня может быть связан одновременно с любыми элементами следующего уровня. Вообще, на связи между объектами в сетевых моделях не накладывается никаких ограничений.

Сетевой базой данных фактически является *Всемирная паутина* глобальной компьютерной сети Интернет. Гиперссылки связывают между собой миллиарды документов в единую распределенную сетевую базу данных.

Сетевой базой данных является также генеалогическое древо семьи, так как потомки (объекты нижележащего уровня) имеют всегда более одного предка (объекта вышестоящего уровня).

Программа GenePro (Живая Родословная) предназначена для создания, редактирования, распечатывания и представ-

ления в Интернете больших генеалогических деревьев, которые являются сетевыми моделями. Особенностью программы является легкость, с которой можно создавать очень сложные родословные, включающие иллюстративные материалы и различные комментарии. Живая Родословная позволяет организовать проектную деятельность учащихся на разных ступенях школы и в рамках различных предметных областей.

Контрольные вопросы

1. В чем заключаются характерные особенности сетевых баз данных?

Практическая работа 3.6

Создание генеалогического древа семьи

Аппаратное и программное обеспечение. Компьютер с установленной операционной системой Windows.

Цель работы. Научиться создавать генеалогическое древо семьи.

Задание. Создать сетевую базу данных «Генеалогическое древо семьи».

Варианты выполнения работы:



построить генеалогическое древо своей семьи.





Создание сетевой базы данных «Генеалогическое древо семьи»



Создадим сетевую базу данных «Генеалогическое древо семьи» в программе GenePro (Живая Родословная).

1. Запустить программу GenePro (Живая Родословная) командой [*Программы-Живая Родословная*].
2. В появившемся диалоговом окне программы на *Панели инструментов* выбрать себя, т. е. воспользоваться кнопками  (*Новый мужчина*) или  (*Новая женщина*).

3. С помощью кнопки  на *Панели инструментов* вставить в генеалогическое древо своих родителей.
4. Поочередно выделить родителей и с помощью той же кнопки  вставить в генеалогическое древо родителей своих родителей, т. е. дедушек и бабушек.
5. Поочередно выделить всех родственников в генеалогическом древе семьи и в контекстном меню объектов выбрать пункт *Сведения*. В появившемся диалоговом окне ввести сведения о родственнике.

Сын - сведения

Иллюстрации | Вид

Сведения о члене семьи

Имя
Сын

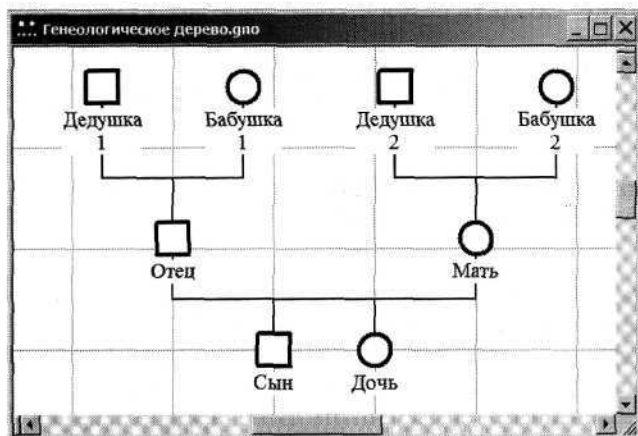
Отчество

Фамилия

Род занятий

Пол
 Мужчина Животное
 Женщина Неизвестен

В результате мы получим генеалогическое древо семьи, которое является примером сетевой базы данных.



Генеалогическое древо семьи хранится в папке ..\ИКТ11baz\

Windows-CD 

Глава 4

Информационное общество

4.1. Право в Интернете

Глобальный характер Интернета создает значительные проблемы в определении того, какие органы должны рассматривать споры по правовым вопросам. Интернет, по крайней мере, в настоящий момент, представляет собой пример того, насколько удачно и эффективно может развиваться сложная техническая система практически в отсутствие единого управляющего центра и правового регулирования.

Имеющиеся нормативные правовые акты по отношению к Интернету можно охарактеризовать следующим образом.

1. Ни в одной стране мира нет всеобъемлющего законодательства по Интернету. Существующие нормативные (подзаконные) акты регулируют частные аспекты функционирования сети.
2. Нормы, которые можно было бы применить к правовым отношениям в Интернете, «разбросаны» по законодательным актам иных отраслей права. В первую очередь они содержатся в нормах, относящихся к интеллектуальной и промышленной собственности.
3. Практически отсутствует регулирование правовых отношений в Интернете на международном (межгосударственном) уровне. Основная проблема заключается в том, что Интернет не имеет территориальных границ своего распространения.

Подходы к решению одинаковых правовых проблем могут сильно различаться в разных странах. Признание определенных действий в Интернете уголовным преступлением в одной стране может считаться вполне легальной деятельностью в другой стране. Во многих случаях интересы госуда-

рства легко могут быть затронуты и действиями лиц, находящимися за его пределами.

Оценить наличие связи лица с конкретным государством весьма трудно, поскольку любой материал, размещенный в Интернете, становится одинаково доступным для лиц, находящихся в любой точке земного шара. Если лицо занимается продажей товаров или оказанием услуг через Интернет, его клиентами могут оказаться представители разных стран. Поэтому для такого лица важно иметь представление о законодательстве этих стран — всех, где находятся его клиенты, ведь возможные споры будут рассматриваться в местных судах.

Государства устанавливают свою юрисдикцию над лицом, если существует определенная связь между этим лицом и территорией такого государства. Наиболее очевидна связь с территорией при размещении информации на определенном сервере, делающем доступным эту информацию для пользователей Интернета.

Однако страна размещения сервера может не совпадать со страной регистрации доменного имени, причем для владельца доменного имени не составит особого труда заменить один компьютер, использующий в Интернете это доменное имя, другим, находящимся за тысячи километров от первого.

Оптимальным решением указанных проблем стала бы как можно более полная унификация национальных и региональных законодательств в области Интернета.

Контрольные вопросы

1. В чем заключаются основные правовые проблемы в Интернете?

4.2. Этика в Интернете

Достоинство Интернета в том, что он расширяет среду общения до размеров земного шара. Написал несколько слов, нажал на кнопку — письмо ушло, хоть соседу по парте, хоть президенту США. В этой простоте кроются опасности: возникает иллюзия доступности и вседозволенности, которая еще усугубляется тем, что при электронном общении

мы не видим лица собеседника и не всегда можем правильно понять его реакцию.

Сетевой этикет — понятие, возникшее с появлением электронной почты. Интернет развивается, форумы и телеконференции позволяют общаться, не только обмениваясь почтовыми сообщениями, рассылаемыми подписчикам, но и просто оставляя сообщения на специальных сайтах в Интернете, которые можно просматривать стандартными браузерами.

В дополнение к электронной почте появились и стали популярными и другие средства общения: чаты, обеспечивающие возможность беседовать в реальном времени или, как говорят, в режиме он-лайн (от англ. on-line) людям, удаленным друг от друга на тысячи километров. Некоторые из чатов позволяют слышать и видеть друг друга.

Для того чтобы, общаясь в Интернете с использованием различных его сервисов, не доставлять неприятностей собеседникам и не иметь их самому, полезно следовать некоторым несложным правилам, называемым сетевым этикетом. Правила сетевого этикета просты и похожи на правила поведения в реальной жизни.

Правила этикета для электронной почты. Всегда заполняйте поле «Тема» своего письма. Ориентируясь по темам, проще выделить нужные письма в большом списке поступающей корреспонденции, а также отфильтровать спам (от англ. spam) — навязчивую рекламу. Отвечая на чье-либо письмо, в поле темы принято вписывать Re: Исходная_тема. Большинство программ для работы с почтой эту фразу вписывают автоматически.

Отвечая на пришедшее письмо, принято цитировать некоторые его отрывки. Цитируемые фразы нужно выделять каким-нибудь символом, обычно это «>», и отделять их от ваших фраз пустой строкой. Большинство программ для работы с электронной почтой при написании ответа сами выделяют текст исходного письма подобными знаками, и пользователю достаточно лишь удалить ненужные фразы.

В Интернете считается, что фраза, написанная **БОЛЬШИМИ БУКВАМИ**, означает, что автор громко кричит. К тому же предложения, написанные буквами верхнего регистра, плохо читаются, что создает неудобства вашему корреспонденту. В связи с этим используйте прописные буквы, только если это является действительно необходимым.

В официальной переписке принято использовать подпись, содержащую некоторую информацию об авторе письма: полное имя, должность или другие регалии, контактную информацию. Поскольку во многих случаях программы для работы с электронной почтой автоматически добавляют к письму однажды подготовленную подпись, будьте внимательны и не помещайте в подпись информацию, которую вы не хотели бы показывать всем своим корреспондентам.

Старайтесь не допускать грамматических ошибок. Многие современные текстовые редакторы и программы для работы с электронной почтой имеют встроенные системы проверки правописания. В любом случае полезно перед отправкой еще раз перечитать письмо.

Не вставляйте в электронное письмо файлы большого объема, не узнав у вашего корреспондента, принимает ли такие вложения его почтовый сервер. Если есть проблема с объемом и количеством приложенных файлов, большие файлы разбейте на несколько меньших и разошлите их отдельными письмами.

В обычной (не деловой) переписке часто используются смайлики — комбинации текстовых символов, например :-), которые напоминают лицо, если смотреть на них, повернув голову набок. Использование смайликов способно придать письму живой характер и даже заменить жестикуляцию. Но не следует этим злоупотреблять — это будет уже плохим тоном. Смайликов придумано очень много, наиболее часто используются такие:

- :-) или :) — улыбка; обычно используется для выражения радости;
- :-(или :(— несчастное лицо; выражает сожаление или разочарование;
- ;-) или ;) — подмигивающее лицо; обычно выражает иронию и означает, что слова не следует понимать слишком буквально.

Правила этикета для общения в чате, форуме, телеконференции. В чате, форуме, гостевой книге общаются большое количество разных людей, с разными мнениями и интересами. Следует быть тактичным и корректным в своих высказываниях. Не нападайте на человека только из-за того, что его мнение не совпало с вашим.

Выбирайте себе псевдоним, или ник (от англ. nickname — прозвище, кличка), не оскорбляющий других участников чата. Избегайте нейтральных имен, не дающих возможности представить кто вы: женщина или мужчина.

Обращаясь к кому-либо, пишите его ник в начале вашей фразы.

Не повторяйте многократно одну и ту же фразу, не забывайте «эфир».

Как и в электронной почте, текст, написанный прописными буквами или с большим количеством восклицательных знаков (например, ПОМОГИТЕ!!!!), интерпретируется как громкий крик, поэтому не нужно злоупотреблять этим средством выделения своих сообщений.

Используйте смайлики, но не злоупотребляйте ими. Не факт, что экзотический смайлик знаком участникам общения ;-).

Уходя из чата, не забудьте попрощаться с вашими собеседниками и, возможно, договориться о времени следующей совместной беседы.

Контрольные вопросы

1. Каковы основные этические правила при общении по электронной почте?
2. Каковы основные этические правила при общении в чатах и форумах?

4.3. Перспективы развития информационных и коммуникационных технологий

Развитие новых информационных и коммуникационных технологий имеет общие законы. Большинство новых технологий проходит в процессе своего развития пять этапов, однако некоторые технологии развиваются очень быстро и «пропускают» некоторые этапы, другие же, наоборот, периодически возвращаются на начальный этап развития.

Лучше всего этапы развития ИКТ можно представить в графической форме (рис. 4.8). По оси абсцисс отложены этапы. Различные технологии проходят этапы своего развития за различное время (от 2 до 10 лет), т. е. шкала оси времени для разных технологий неодинакова. По оси ординат отло-

жен уровень оценки технологии обществом, что носит довольно субъективный характер.

Первый этап. «Восход надежд», время теоретических разработок и первых экспериментальных реализаций новой информационной или коммуникационной технологии. Разработчикам и экспертам кажется, что данная новая технология разрешит многие проблемы развития информационных технологий.

Примерами таких технологий являются разработки транзисторов молекулярных и атомных размеров. Действительно, современные транзисторы уже имеют размеры в несколько десятков атомов, и дальнейшая миниатюризация должна строиться на новой основе.

Основой квантового компьютера может стать любая квантовая частица, обладающая двумя состояниями (логические 0 и 1). Например, это может быть спин электрона, имеющий два состояния (вниз и вверх), основное и возбужденное состояние атома и другие объекты, подчиняющиеся законам квантовой механики (рис. 4.1). Первые теоретические разработки квантовых компьютеров начались около 20 лет назад, проведены успешные лабораторные опыты по созданию элементов таких компьютеров. Однако предсказать сроки появления квантовых компьютеров сейчас невозможно.

ДНК-вычисления предполагают создание новых алгоритмов вычислений на основе знаний о строении и функциях молекулы ДНК. Так же, как и любой другой процессор, ДНК-процессор характеризуется структурой и набором команд. В нашем случае структура процессора — это структура молекулы ДНК (рис. 4.2). А набор команд — это перечень биохимических операций с молекулами.

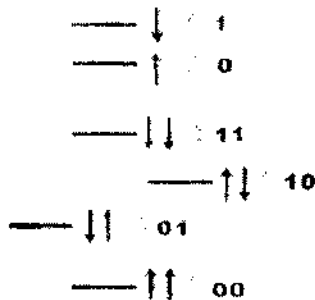


Рис. 4.1. Квантовый компьютер (квантовый бит — это спин электрона или ядра атома)



Рис. 4.2. Модель ДНК-процессора

На базе ДНК-вычислений ведется разработка нанокomпьютера, который можно будет вживлять в клетку организма и производительность которого будет исчисляться миллиардами операций в секунду при энергопотреблении не более одной миллиардной ватта. В настоящее время ДНК-вычисления находятся на стадии лабораторных исследований, поэтому создание биологического компьютера прогнозируется только через несколько десятков лет.

Молекулярный транзистор (рис. 4.3) — это молекула, которая может существовать в двух устойчивых состояниях, обладающих разными свойствами (логические 0 и 1). Транзистор на одной молекуле в десятки раз меньше современных транзисторов. Переводить молекулу из одного состояния в другое можно с помощью света, тепла, магнитного поля и других физических воздействий. Уже в настоящее время существуют логические схемы на молекулярных транзисторах и, планируется, что уже в ближайшее десятилетие начнется их промышленное производство.

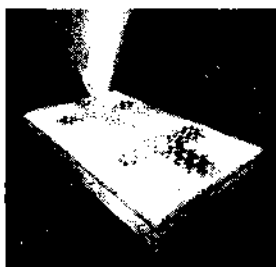


Рис. 4.3.
Молекулярный транзистор

Второй этап. «Пик завышенных ожиданий», когда работчики и средства массовой информации внушают обществу высокую ценность новой технологии и эффективность первых промышленных образцов.

Примером такой технологии являются «электронные чернила» (рис. 4.4). В ходе многолетних исследований удалось создать тип устройств визуализации информации, которые обладают механическими свойствами обычной бумаги (например, их можно свертывать в рулон).

Базовыми элементами таких устройств являются микрокапсулы (пиксели), заполненные микрочастицами двух цветов: белого и черного. Слой микрокапсул расположен между двумя прозрачными и гибкими электродами. При подаче напряжения определенной полярности, микрочастицы белого цвета собираются в верхней части капсулы, а микрочастицы черного цвета — в нижней части. При перемене полярности напряжения все происходит наоборот. Так формируется черно-белое изображение. Однако существенным недостатком таких устройств является большое время переключения

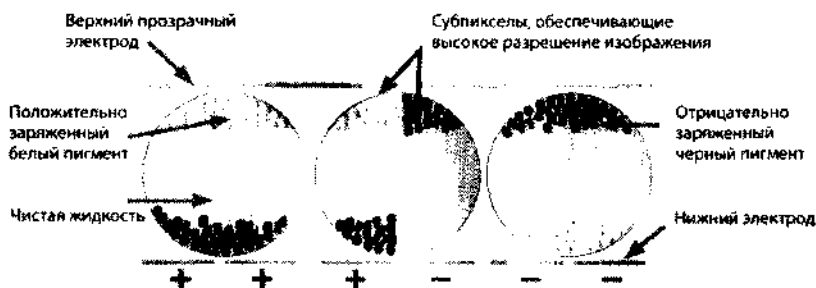


Рис. 4.4. Электронные чернила

пикселей (около 1 с), что препятствует их широкому промышленному производству.

Третий этап. «Котловина разочарований», когда широко разрекламированная новая технология теряет свою привлекательность в глазах конечных потребителей. В процессе использования первых массовых экземпляров новой технологии выявляются конструктивные недостатки.

Компактные топливные элементы (рис. 4.5) предназначены для прямого преобразования энергии, высвобождающейся в ходе реакции окисления топлива, в электрическую энергию. В отличие от аккумуляторов, заряд которых возобновляется при подключении к внешнему источнику тока, восстановление работоспособности топливных элементов осуществляется путем пополнения запаса топлива.

Однако у топливных элементов обнаружился серьезный недостаток: проблема зарядки топливом, высокая температура топливного элемента при работе. Все это откладывает массовое промышленное производство топливных элементов.

Четвертый этап. «Подъем жизнестойкости», когда на основе новых исследований оптимизируется технологический процесс и начинается массовое серийное производство.

Примером такой технологии является машинный перевод. Системы машинного перевода (рис. 4.6) получили ши-



Рис. 4.5. Компактный топливный элемент

рское распространение и дают приемлемое качество перевода. С помощью систем машинного перевода можно переводить тексты как off-line, так и on-line (Web-страницы и письма электронной почты). Кроме того, расширился набор языков и направлений перевода.

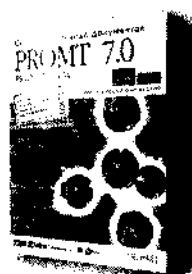


Рис. 4.6. Система машинного перевода

Пятый этап. «Плато продуктивности», когда массовое серийное производство изделий по новой технологии находит массовый устойчивый спрос потребителей и приносит стабильную прибыль производителям.

Определение местоположения на поверхности земли стало широко применяться в спутниковых системах (GPS — США или ГЛОНАСС — Россия). Для этого запущено требуемое количество спутников и развернуто массовое промышленное производство приемников спутникового сигнала (с нескольких спутников). На экране такого приемника (рис. 4.7) отображаются карты местности с указанием местоположения. Точность такого определения местоположения в открытом гражданском секторе составляет несколько десятков метров, а в закрытом военном — несколько метров.

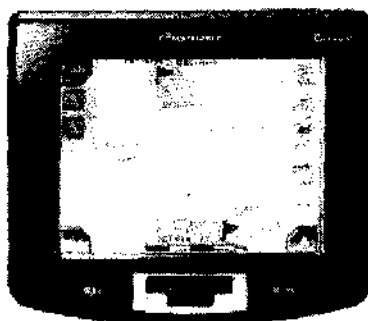


Рис. 4.7. Приемник определения местоположения

Определение местоположения предоставляют и операторы мобильной связи, правда точность определения местоположения составляет обычно несколько сот метров и зависит от количества и расположения базовых станций.

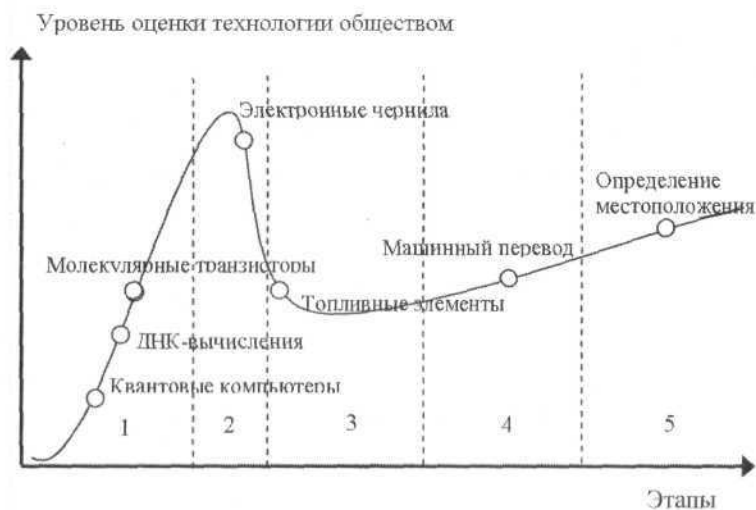


Рис. 4.8. Развитие информационных и коммуникационных технологий

Контрольные вопросы

1. Назовите информационные и коммуникационные технологии, соответствующие различным этапам развития технологии.

Глава 5

Повторение. Подготовка к ЕГЭ. Тесты по темам курса «Информатика и ИКТ»

1. На выполнение экзаменационной работы в форме тестов по курсу «Информатика и ИКТ» отводится 4 часа (240 минут), включая работу на компьютере.
2. Тесты включают 160 заданий, которые делятся на 4 части (А, В, С и D), аналогично единому государственному экзамену (ЕГЭ):
 - Часть А (ВО — выборочный ответ). Задание считается выполненным, если учащийся записал номер верного варианта ответа.
 - Часть В (КО — краткий ответ). Задание с кратким ответом считается выполненным, если учащийся дал ответ (число, значение переменной, путь к файлу или логическое значение выражения), соответствующий верному варианту ответа.
 - Часть С (РО — развернутый ответ). Задание с развернутым ответом считается выполненным, если учащийся правильно записал последовательность преобразований логического выражения или программу.
 - Часть D (ПЗ — практическое задание). Практическое задание считается выполненным, если файл задания, сохраненный учащимся, соответствует заданному эталону.
3. Тесты распределены по темам курса «Информатика и ИКТ».
4. Установлены следующие уровни сложности заданий:
 - Б — базовый,
 - П — повышенный,
 - В — высокий.
5. Практические задания должны выполняться на базе классов современных компьютеров с установленными операционной системой Windows 95/98/Me/2000/XP/Vista, интегрированными офисными приложениями Microsoft Office или OpenOffice, системами программиро-

вания (например, VisualStudio 2005 Express Edition и Delphi), графическим редактором (например, GIMP) и системой компьютерного черчения КОМПАС.

6. Файлы-задания, необходимые для выполнения практических заданий, размещены на диске Windows-CD в папке ..\Test\.

Тема 1. Информация. Кодирование информации

1.1. Единицы измерения количества информации

1.1.1. За минимальную единицу измерения количества информации принят:

- 1) 1 бод; 2) 1 пиксель; 3) 1 байт; 4) 1 бит.

1.1.2. Чему равен 1 байт?

- 1) 2^3 битов; 2) 10^3 битов; 3) 2^{10} битов; 4) 10^{10} битов.

1.1.3. Сколько бит в 1 килобайте?

- 1) 1000 битов; 3) 1024 бита;
2) $8 \cdot 2^{10}$ битов; 4) $8 \cdot 10^3$ битов.

1.1.4. Чему равен 1 мегабайт?

- 1) 10^6 битов; 3) 2^{10} Кбайт;
2) 10^6 байтов; 4) 2^{10} байтов.

1.2. Определение количества информации (вероятностный подход)

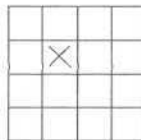
1.2.1. В рулетке общее количество лунок равно 32. Какое количество информации (с точки зрения вероятностного подхода) мы получаем в зрительном сообщении об остановке шарика в одной из лунок?

- 1) 8 битов; 2) 5 битов; 3) 2 бита; 4) 1 бит.

1.2.2. Производится бросание симметричной четырехгранной пирамидки. Какое количество информации (с точки зрения вероятностного подхода) мы получаем в зрительном сообщении о ее падении на одну из граней?

- 1) 1 бит; 2) 2 бита; 3) 4 бита; 4) 8 битов.

1.2.3. Какое количество информации (с точки зрения вероятностного подхода) получит второй игрок при игре в крестики-нолики на поле 4×4 , после первого хода первого игрока, играющего крестиками?



- 1) 1 бит; 2) 2 бита; 3) 4 бита; 4) 8 битов.

- 1.2.1. Какое количество информации (с точки зрения вероятностного подхода) получит при игре в шахматы играющий черными после первого хода белых (при условии, что ходить конями запрещено)?



- 1) 1 бит; 2) 2 бита; 3) 4 бита; 4) 1 байт.

1.3. Определение количества информации (алфавитный подход)

- 1.3.1. Какое количество информации (с точки зрения алфавитного подхода) содержит двоичное число 101_2 ?
 1) 3 байта; 2) 2 байта; 3) 3 бита; 4) 2 бита.
- 1.3.2. Какое количество информации (с точки зрения алфавитного подхода) содержит восьмеричное число 55_8 ?
 1) 10 битов; 2) 8 битов; 3) 6 битов; 4) 5 битов.
- 1.3.3. Какое количество информации (с точки зрения алфавитного подхода) содержит шестнадцатеричное число AB_{16} ?
 1) 16 битов; 2) 8 битов; 3) 4 бита; 4) 2 бита.
- 1.3.4. Какое количество информации (с точки зрения алфавитного подхода) содержит слово «информатика», если считать, что алфавит состоит из 32 букв?
 1) 55 битов; 2) 55 байтов; 3) 11 битов; 4) 11 байтов.

1.4. Кодирование текстовой информации

- 1.4.1. Во сколько раз увеличится информационный объем страницы текста (текст не содержит управляющих символов форматирования) при его преобразовании из кодировки Windows (таблица кодировки содержит 256 символов) в кодировку Unicode (таблица кодировки содержит 65 536 символов)?
 1) в 2 раза; 2) в 8 раз; 3) в 16 раз; 4) в 256 раз.
- 1.4.2. Во сколько раз уменьшится информационный объем страницы текста (текст не содержит управляющих символов форматирования) при его преобразовании из

кодировки Unicode (таблица кодировки содержит 65 536 символов) в кодировку Windows (таблица кодировки содержит 256 символов)?

- | | |
|---------------|--------------|
| 1) в 256 раз; | 3) в 4 раза; |
| 2) в 8 раз; | 4) в 2 раза. |
- 1.4.3. Какое количество информации необходимо для кодирования каждого из 256 символов алфавита?
- | | |
|---------------|-------------|
| 1) 256 битов; | 3) 8 битов; |
| 2) 16 битов; | 4) 4 бита. |
- 1.4.4. Какое количество информации необходимо для кодирования каждого из 65 536 символов алфавита?
- | | | | |
|------------|-------------|-------------|-------------|
| 1) 1 байт; | 2) 2 байта; | 3) 8 битов; | 4) 32 бита. |
|------------|-------------|-------------|-------------|

1.5. Кодирование графической информации

- 1.5.1. Черно-белое (без градаций серого цвета) растровое графическое изображение имеет размер 10×10 точек. Какой объем памяти займет это изображение?
- | | |
|----------------|-----------------|
| 1) 100 битов; | 3) 1000 битов; |
| 2) 100 байтов; | 4) 1000 байтов. |
- 1.5.2. Цветное (с палитрой из 256 цветов) растровое графическое изображение имеет размер 10×10 точек. Какой объем памяти займет это изображение?
- | | |
|---------------|----------------|
| 1) 100 битов; | 3) 100 байтов; |
| 2) 800 битов; | 4) 800 байтов. |
- 1.5.3. В процессе преобразования растрового графического изображения количество цветов уменьшилось с 65 536 до 16. Во сколько раз уменьшился информационный объем графического файла?
- | | | | |
|--------------|--------------|-------------|--------------|
| 1) в 2 раза; | 2) в 4 раза; | 3) в 8 раз; | 4) в 16 раз. |
|--------------|--------------|-------------|--------------|
- 1.5.4. В процессе преобразования растрового графического изображения количество цветов увеличилось с 256 до 65 536. Во сколько раз увеличился информационный объем графического файла?
- | | | | |
|--------------|--------------|-------------|--------------|
| 1) в 2 раза; | 2) в 4 раза; | 3) в 8 раз; | 4) в 16 раз. |
|--------------|--------------|-------------|--------------|

1.6. Кодирование звуковой информации

- 1.6.1. Аналоговый звуковой сигнал был дискретизирован сначала с использованием 65 536 уровней интенсивности сигнала (качество звучания аудио-CD), а затем — с использованием 256 уровней интенсивности сигнала (качество звучания радиотрансляции). Во

сколько раз различаются информационные объемы оцифрованных звуковых сигналов?

1) в 256 раз; 2) в 16 раз; 3) в 8 раз; 4) в 2 раза.

1.6.2. Звуковая плата реализует 16-битовое двоичное кодирование аналогового звукового сигнала. Это позволяет воспроизводить звук с:

- 1) 8 уровнями интенсивности;
- 2) 16 уровнями интенсивности;
- 3) 256 уровнями интенсивности;
- 4) 65 536 уровнями интенсивности.

1.6.3. Звуковая плата производит двоичное кодирование аналогового звукового сигнала. Какое количество информации необходимо для кодирования каждого из 65 536 возможных уровней интенсивности сигнала?

1) 256 битов; 2) 16 битов; 3) 8 битов; 4) 1 бит.

1.6.4. Звуковая плата реализует 8-битовое двоичное кодирование аналогового звукового сигнала. Это позволяет воспроизводить звук с:

- 1) 8 уровнями интенсивности;
- 2) 16 уровнями интенсивности;
- 3) 256 уровнями интенсивности;
- 4) 65 536 уровнями интенсивности.

1.7. Представление числовой информации. Сложение чисел в двоичной и десятичной системах счисления

1.7.1. Вычислить сумму двоичного и десятичного чисел $10_2 + 10_{10}$. Представить результат в десятичной системе счисления.

1) 11_{10} ; 2) 12_{10} ; 3) 13_{10} ; 4) 14_{10} .

1.7.2. Вычислить сумму двоичного и десятичного чисел $10_2 + 10_{10}$. Представить результат в двоичной системе счисления.

1) 1000_2 ; 2) 1100_2 ; 3) 1110_2 ; 4) 1111_2 .

1.7.3. Вычислить сумму двоичного и десятичного чисел $11_2 + 11_{10}$. Представить результат в двоичной системе счисления.

1) 1000_2 ; 2) 1100_2 ; 3) 1110_2 ; 4) 1111_2 .

1.7.4. Вычислить сумму двоичного и десятичного чисел $11_2 + 11_{10}$. Представить результат в десятичной системе счисления.

1) 12_{10} ; 2) 13_{10} ; 3) 14_{10} ; 4) 15_{10} .

1.8. Представление числовой информации. Сложение чисел в двоичной, восьмеричной, десятичной и шестнадцатеричной системах счисления.

- 1.8.1. Вычислить сумму чисел $11_2 + 11_8 + 11_{10} + 11_{16} = \text{---}_2$.
Представить результат в двоичной системе счисления.
- 1.8.2. Вычислить сумму чисел $11_2 + 11_8 + 11_{10} + 11_{16} = \text{---}_{10}$.
Представить результат в десятичной системе счисления.
- 1.8.3. Вычислить сумму чисел $11_2 + 11_8 + 11_{10} + 11_{16} = \text{---}_8$.
Представить результат в восьмеричной системе счисления.
- 1.8.4. Вычислить сумму чисел $11_2 + 11_8 + 11_{10} + 11_{16} = \text{---}_{16}$.
Представить результат в шестнадцатеричной системе счисления.

Тема 2. Устройство компьютера и программное обеспечение

2.1. Устройство компьютера

- 2.1.1. Драйвер — это:
- 1) устройство компьютера;
 - 2) компьютерный вирус;
 - 3) программа, обеспечивающая работу устройства компьютера;
 - 4) антивирусная программа.
- 2.1.2. При выключении компьютера вся информация теряется:
- 1) на гибком диске;
 - 2) на жестком диске;
 - 3) на CD-ROM диске;
 - 4) в оперативной памяти.
- 2.1.3. Программа может управлять работой компьютера, если она находится:
- 1) на гибком диске;
 - 2) на жестком диске;
 - 3) на CD-ROM диске;
 - 4) в оперативной памяти.
- 2.1.4. Процессор обрабатывает информацию, представленную:
- 1) в десятичной системе счисления;
 - 2) на языке программирования высокого уровня;
 - 3) на алгоритмическом языке;
 - 4) на машинном языке (в двоичном коде).

2.2. Безопасность и технические условия эксплуатации

- 2.2.1. В целях сохранения информации жесткие магнитные диски необходимо оберегать от:
- 1) пониженной температуры;
 - 2) царапин;
 - 3) света;
 - 4) ударов при установке.
- 2.2.2. В целях сохранения информации гибкие магнитные диски необходимо оберегать от:
- 1) пониженной температуры;
 - 2) магнитных полей;
 - 3) света;
 - 4) перепадов атмосферного давления.
- 2.2.3. В целях сохранения информации оптические CD- и DVD-диски необходимо оберегать от:
- 1) пониженной температуры;
 - 2) магнитных полей;
 - 3) света;
 - 4) загрязнений.
- 2.2.4. В целях сохранения нормальной работоспособности модули оперативной памяти необходимо оберегать от:
- 1) электростатических зарядов при установке;
 - 2) магнитных полей;
 - 3) света;
 - 4) загрязнений.

2.3. Операционная система: назначение и функциональные возможности

- 2.3.1. Операционная система — это:
- 1) программа, обеспечивающая управление базами данных;
 - 2) антивирусная программа;
 - 3) программа, управляющая работой компьютера;
 - 4) система программирования.
- 2.3.2. Процесс загрузки операционной системы представляет собой:
- 1) копирование файлов операционной системы с гибкого диска на жесткий диск;
 - 2) копирование файлов операционной системы с CD-диска на жесткий диск;
 - 3) последовательную загрузку файлов операционной системы в оперативную память;
 - 4) копирование содержимого оперативной памяти на жесткий диск.

- 2.3.3. Системный диск необходим для:
- 1) загрузки операционной системы;
 - 2) хранения важных файлов;
 - 3) систематизации файлов;
 - 4) лечения компьютера от вирусов.
- 2.3.4. В логический раздел диска одновременно может быть установлено:
- 1) несколько различных операционных систем;
 - 2) несколько копий одной операционной системы;
 - 3) только одна операционная система;
 - 4) фрагменты различных операционных систем.

2.4. Архитектура компьютера

- 2.4.1. Какова пропускная способность системной шины (с точностью до целых), если ее разрядность составляет 64 бита, а частота — 1066 МГц?
- 2.4.2. Какова пропускная способность шины памяти (с точностью до целых), если ее разрядность составляет 64 бита, а частота — 533 МГц?
- 2.4.3. Какова пропускная способность шины AGP (с точностью до целых), если ее разрядность составляет 32 бита, а частота — 528 МГц?
- 2.4.4. Какова пропускная способность шины PCI (с точностью до целых), если ее разрядность составляет 64 бита, а частота — 66 МГц?

2.5. Файлы и файловые системы

- 2.5.1. Файл — это:
- 1) единица измерения количества информации;
 - 2) программа или данные на диске, имеющие имя;
 - 3) программа в оперативной памяти;
 - 4) текст, распечатанный на принтере.
- 2.5.2. При полном форматировании гибкого диска:
- 1) стираются все данные;
 - 2) производится только очистка каталога диска;
 - 3) диск становится системным;
 - 4) производится дефрагментация размещения файлов на диске.
- 2.5.3. Разные файлы могут иметь одинаковые имена, если они:
- 1) имеют разные объемы;
 - 2) созданы в различные дни;

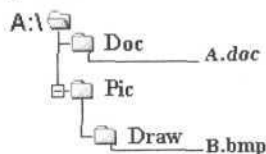
- 3) созданы в различное время суток;
- 4) хранятся в разных папках.

2.5.4. Информационный объем файла на гибком диске не может быть меньше, чем:

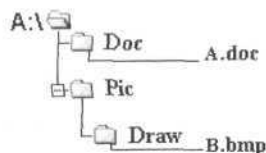
- 1) размер сектора диска; 3) 1 байт;
- 2) 1 бит; 4) 1 Кбайт.

2.6. Путь к файлу

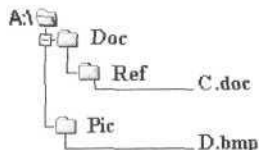
2.6.1. Записать полное имя файла *B.bmp* (включая путь к файлу) в иерархической файловой системе, изображенной на рисунке.



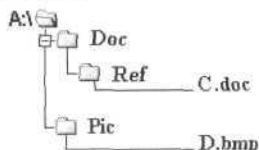
2.6.2. Записать полное имя файла *A.doc* (включая путь к файлу) в иерархической файловой системе, изображенной на рисунке.



2.6.3. Записать полное имя файла *C.doc* (включая путь к файлу) в иерархической файловой системе, изображенной на рисунке.



2.6.4. Записать полное имя файла *D.bmp* (включая путь к файлу) в иерархической файловой системе, изображенной на рисунке.



2.7. Защита информации

- 2.7.1. Отличительной особенностью компьютерных вирусов от других вредоносных программ является :
- 1) проникновение на компьютер по компьютерным сетям;
 - 2) способность к размножению (самокопированию);
 - 3) воровство информации;
 - 4) сетевые атаки.
- 2.7.2. Отличительной особенностью сетевых червей от других вредоносных программ является:
- 1) проникновение на компьютер по компьютерным сетям;
 - 2) способность к размножению (самокопированию);
 - 3) воровство информации;
 - 4) сетевые атаки.
- 2.7.3. Отличительной особенностью троянских программ от других вредоносных программ является:
- 1) проникновение на компьютер по компьютерным сетям;
 - 2) способность к размножению (самокопированию);
 - 3) воровство информации;
 - 4) сетевые атаки.
- 2.7.4. Отличительной особенностью хакерских утилит от других вредоносных программ является:
- 1) проникновение на компьютер по компьютерным сетям;
 - 2) способность к размножению (самокопированию);
 - 3) воровство информации;
 - 4) сетевые атаки.

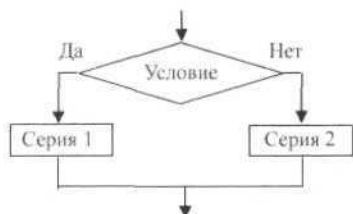
Тема 3. Алгоритмизация и программирование

3.1. Основные алгоритмические структуры

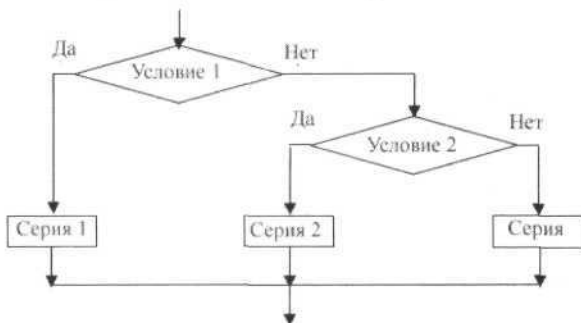
- 3.1.1. Нарисовать блок-схему алгоритмической структуры «ветвление».
- 3.1.2. Нарисовать блок-схему алгоритмической структуры «выбор».
- 3.1.3. Нарисовать блок-схему алгоритмической структуры «цикл со счетчиком».
- 3.1.4. Нарисовать блок-схему алгоритмической структуры «цикл с предусловием».

3.2. Кодирование алгоритмических структур на языках программирования

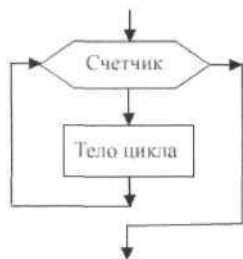
3.2.1. Записать алгоритмическую структуру «ветвление» на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi.



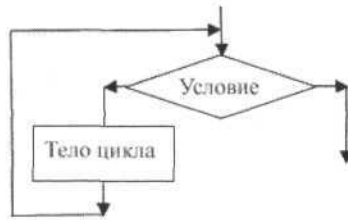
3.2.2. Записать алгоритмическую структуру «выбор» на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi.



3.2.3. Записать алгоритмическую структуру «цикл со счетчиком» на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi.

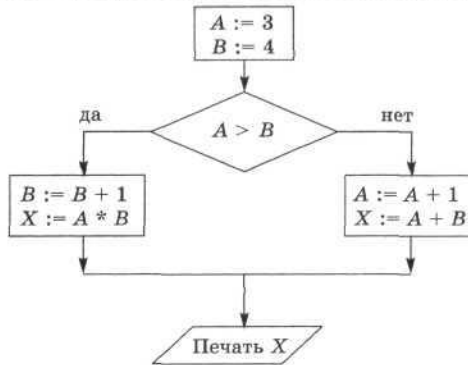


3.2.4. Записать алгоритмическую структуру «цикл с пред-условием» на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi.



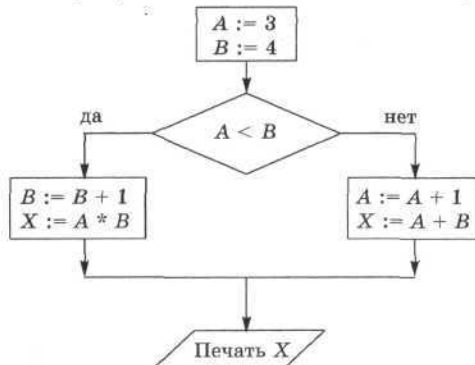
3.3. Формальное исполнение простого алгоритма

3.3.1. Фрагмент алгоритма изображен в виде блок-схемы. Определить, какое значение переменной X будет напечатано в результате выполнения алгоритма.



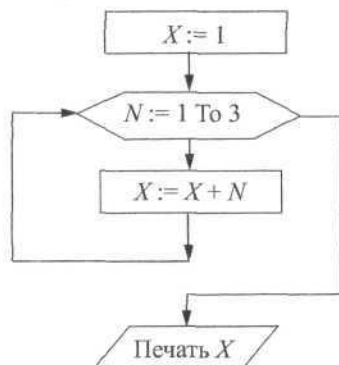
1) 8; 2) 10; 3) 15; 4) 18.

3.3.2. Фрагмент алгоритма изображен в виде блок-схемы. Определить, какое значение переменной X будет напечатано в результате выполнения алгоритма.



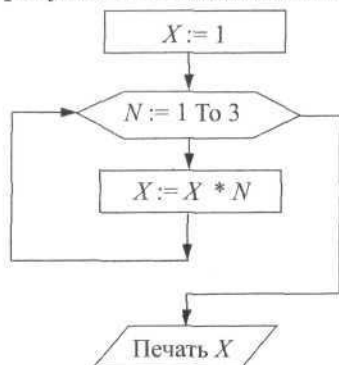
1) 8; 2) 10; 3) 15; 4) 18.

3.3.3. Фрагмент алгоритма изображен в виде блок-схемы. Определить, какое значение переменной X будет напечатано в результате выполнения алгоритма.



- 1) 5; 2) 6; 3) 7; 4) 8.

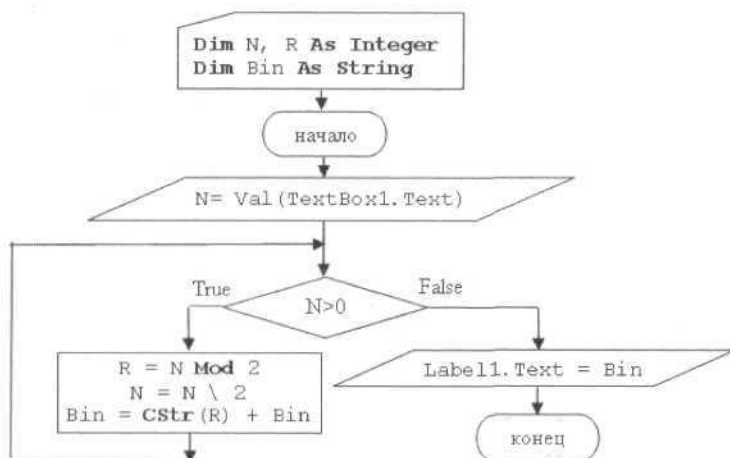
3.3.4. Фрагмент алгоритма изображен в виде блок-схемы. Определить, какое значение переменной X будет напечатано в результате выполнения алгоритма.



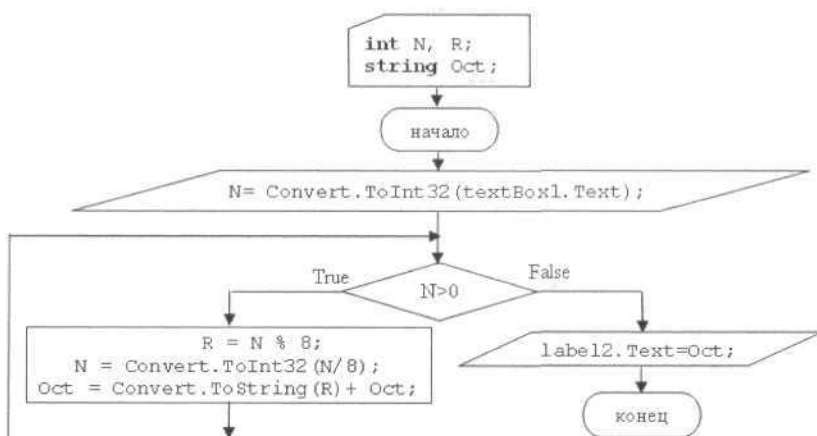
- 1) 3; 2) 4; 3) 5; 4) 6.

3.4. Формальное исполнение сложного алгоритма

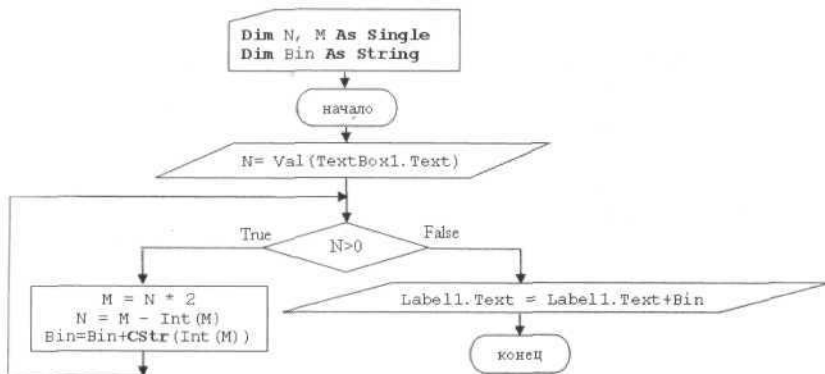
3.4.1. Алгоритм изображен в виде блок-схемы, в графических элементах которого приведен программный код на языке Visual Basic .NET. В текстовое поле введено целое десятичное число 10. Определить, какое целое двоичное число будет выведено на надпись в результате выполнения алгоритма.



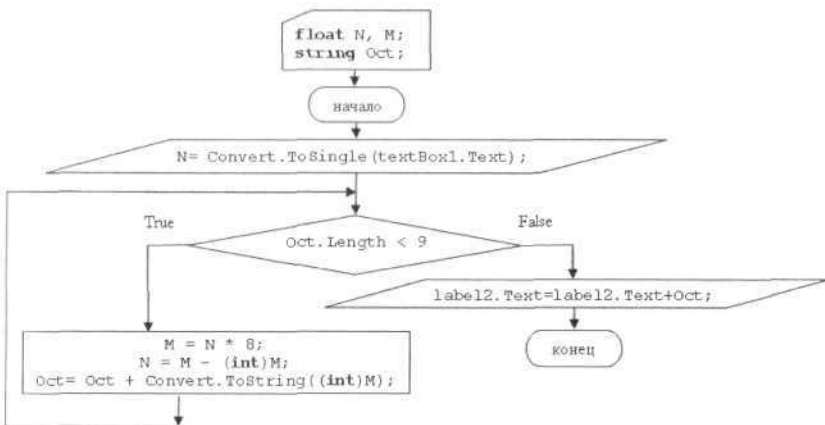
3.4.2. Алгоритм изображен в виде блок-схемы, в графических элементах которого приведен программный код на языке Visual C#. В текстовое поле введено целое десятичное число 20. Определить, какое целое восьмеричное число будет выведено на надпись в результате выполнения алгоритма.



3.4.3. Алгоритм изображен в виде блок-схемы, в графических элементах которого приведен программный код на языке Visual Basic .NET. В текстовое поле введено дробное десятичное число 0.25. Определить, какое дробное двоичное число будет выведено на надпись в результате выполнения алгоритма.



3.4.4. Алгоритм изображен в виде блок-схемы, в графических элементах которого приведен программный код на языке Visual C#. В текстовое поле введено дробное десятичное число 0.25. Определить, какое дробное восьмеричное число будет выведено на надпись в результате выполнения алгоритма.



3.5. Переменные

3.5.1. Имя переменной определяет:

- 1) данные, хранящиеся в выделенной области оперативной памяти;
- 2) выделенную область оперативной памяти;
- 3) количество выделяемых ячеек оперативной памяти;
- 4) диапазон значений переменной.

3.5.2. Значение переменной определяет:

- 1) данные, хранящиеся в выделенной области оперативной памяти;
- 2) выделенную область оперативной памяти;
- 3) количество выделяемых ячеек оперативной памяти;
- 4) диапазон значений переменной.

3.5.3. Тип переменной определяет:

- 1) данные, хранящиеся в выделенной области оперативной памяти;
- 2) выделенную область оперативной памяти;
- 3) количество выделяемых ячеек оперативной памяти;
- 4) присваивание переменной значения.

3.5.4. Присваивание переменной значения приводит:

- 1) к изменению данных, хранящихся в выделенной области оперативной памяти;
- 2) к изменению выделенной области оперативной памяти;
- 3) к изменению количества выделяемых ячеек оперативной памяти;
- 4) к изменению диапазона значений переменной.

3.6. Результаты выполнения программы на языках программирования

3.6.1. Приведены программы на языках объектно-ориентированного программирования Visual Basic .NET и Delphi. В текстовые поля введены десятичные числа 5 и 3. Определить число, которое будет выведено на надпись в результате выполнения алгоритма.

Visual Basic .NET	Delphi
<pre> Dim A, B, Max As Single Private Sub Button1_Click(...) A=Val(TextBox1.Text) B=Val(TextBox2.Text) If A>=B Then Max=A Else Max=B End If Label1.Text = Max End Sub </pre>	<pre> var A: string; B: string; Max: string; procedure TForm1.Button1Click (Sender: TObject); begin A:=EditA.Text; B:=EditB.Text; If A>=B Then Max:=A Else Max:=B; Label1.Caption:=Max; end; </pre>

- 3.6.2. Приведены программы на языках объектно-ориентированного программирования Visual Basic .NET и Delphi. В текстовые поля введены слово "информатика" и символ "а". Определить число, которое будет выведено на надпись в результате выполнения алгоритма.

Visual Basic .NET	Delphi
<pre> Dim N, K As Byte Private Sub Button1_Click(...) N=1 K=0 Do While N<=Len(TextBox1.Text) S=Mid(TextBox1.Text, N, 1) If S=TextBox2.Text Then K=K+1 N=N+1 Loop Label1.Text=K End Sub </pre>	<pre> var N: integer; K: integer; A: string; B: string; M: string; procedure TForm1.Button1Click (Sender: TObject); begin N:=0; K:=0; A:=Edit1.Text; B:=Edit2.Text; While N<Length(Edit1.Text) Do begin N:=N+1; M:=Copy(Edit1.Text,N,1); If M=B Then K:=K+1; end; Label1.Caption:=IntToStr(K); end; </pre>

- 3.6.3. Приведены программы на языках объектно-ориентированного программирования Visual Basic .NET и Delphi. В текстовое поле введено число 5. Определить, число будет выведено на надпись в результате выполнения алгоритма.

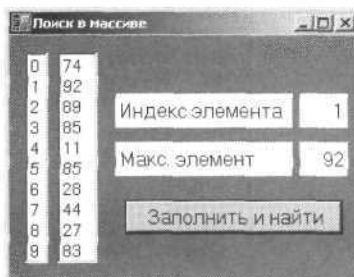
Visual Basic .NET	Delphi
<pre> Dim N As Byte, I As Integer, F As Decimal Private Sub TextBox1_TextChanged(...) N=Val(TextBox1.Text) F=1 I=1 Do F=F*I I=I+1 Loop While I<=N Label1.Text= F End Sub </pre>	<pre> var N: integer; I: integer; F: int64; procedure TForm1.Edit1Change (Sender: TObject); begin N:=StrToInt(Edit1.Text); F:=1; I:=1; Repeat F:=F*I; I:=I+1; Until I>N; Label1.Caption:=IntToStr(F); end; </pre>

3.6.4. Приведены программы на языках программирования Visual Basic .NET и Delphi. Определить, что будет выведено на надпись в результате выполнения алгоритма.

Visual Basic .NET	Delphi
<pre>Dim L1, L2, L3 As Boolean Private Sub Button1_Click(...) L1=5>3 L2=2*2=5 L3=L1 And L2 Label1.Text=L3 End Sub</pre>	<pre>var A: boolean; B: boolean; C: boolean; procedure TForm1.Button1Click (Sender: TObject); begin A:=5>3; B:=2*2=5; C:=A and B; Label1.Caption:=BoolToStr(C, True); end;</pre>

3.7. Составление программы на языках программирования

3.7.1. Составить программу заполнения массива случайными числами и поиска в массиве максимального элемента на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi. Индексы элементов массива и сами массивы выводятся в элементы управления списки, а индекс максимального элемента и сам максимальный элемент выводятся на метки.

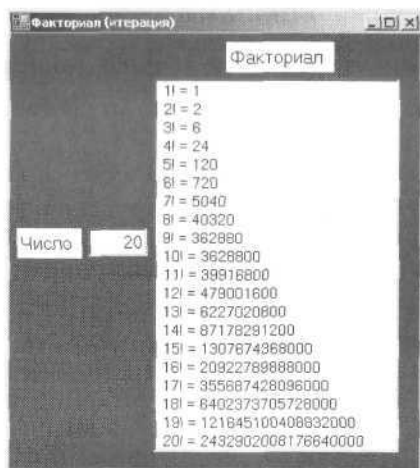


3.7.2. Составить программу заполнения массива случайными числами и поиска в массиве минимального элемента на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi. Индексы эле-

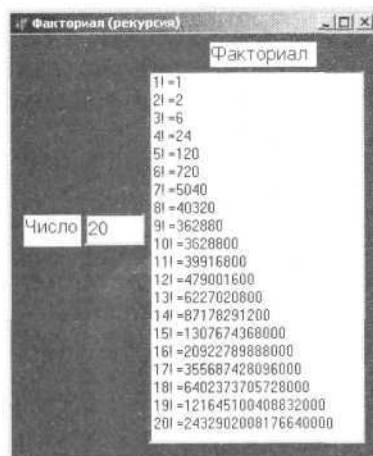
ментов массива и сами массивы выводятся в элементы управления списки, а индекс минимального элемента и сам минимальный элемент выводятся на метки.



- 3.7.1. Составить программу вычисления факториала числа итерационным методом с использованием цикла со счетчиком на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi. Число вводить в элемент управления текстовое поле, а каждый шаг вычисления факториала выводить в элемент управления список.



- 3.7.2. Составить программу вычисления факториала числа методом рекурсии на одном из языков программирования Visual Basic .NET, Visual C#, Visual J# или Delphi. Число вводить в элемент управления текстовое поле, а каждый шаг вычисления факториала выводить в элемент управления список.



Тема 4. Основы логики и логические основы компьютера

4.1. Определение истинности высказывания

4.1.1. Определить, истинно или ложно составное высказывание:

$$A = \{(2 \times 2 = 4 \text{ или } 3 \times 3 = 10) \text{ и } (2 \times 2 = 5 \text{ или } 3 \times 3 = 9)\}$$

4.1.2. Определить, истинно или ложно составное высказывание:

$$A = \{(2 \times 2 = 4 \text{ и } 3 \times 3 = 10) \text{ или } (2 \times 2 = 5 \text{ и } 3 \times 3 = 9)\}$$

4.1.3. Определить, истинно или ложно составное высказывание:

$$A = \{(2 \times 2 = 4 \text{ или } 3 \times 3 = 10) \text{ или } (2 \times 2 = 5 \text{ и } 3 \times 3 = 9)\}$$

4.1.4. Определить, истинно или ложно составное высказывание:

$$A = \{(2 \times 2 = 4 \text{ и } 3 \times 3 = 10) \text{ или } (2 \times 2 = 5 \text{ или } 3 \times 3 = 9)\}$$

4.2. Построение таблиц истинности логических выражений

4.2.1. Укажите таблицу истинности, которая соответствует логической функции $F = A \& \bar{B}$.

1) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1	2) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	3) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0	4) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	1	1	1	0
A	B	F																																																													
0	0	0																																																													
0	1	0																																																													
1	0	0																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	0																																																													
A	B	F																																																													
0	0	0																																																													
0	1	0																																																													
1	0	1																																																													
1	1	0																																																													

4.2.2. Укажите таблицу истинности, которая соответствует логической функции $F = \bar{A} \& B$.

1) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1	2) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	3) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	0	1	1	0	4) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	1	1	1	0
A	B	F																																																													
0	0	0																																																													
0	1	0																																																													
1	0	0																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	0																																																													
1	1	0																																																													
A	B	F																																																													
0	0	0																																																													
0	1	0																																																													
1	0	1																																																													
1	1	0																																																													

4.2.3. Укажите таблицу истинности, которая соответствует логической функции $F = A \vee \bar{B}$.

1) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1	2) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	3) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0	4) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	1	1	1	1
A	B	F																																																													
0	0	0																																																													
0	1	0																																																													
1	0	0																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	0																																																													
A	B	F																																																													
0	0	1																																																													
0	1	0																																																													
1	0	1																																																													
1	1	1																																																													

4.2.4. Укажите таблицу истинности, которая соответствует логической функции $F = \bar{A} \vee B$.

1) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	0	1	1	1	2) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	3) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	0	1	1	0	4) <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																																																													
0	0	1																																																													
0	1	1																																																													
1	0	0																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	1																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	0																																																													
1	1	0																																																													
A	B	F																																																													
0	0	0																																																													
0	1	1																																																													
1	0	1																																																													
1	1	0																																																													

4.3. Логические выражения и их преобразование

4.3.1. Упростить логическое выражение $(A \& B) \vee (A \& \bar{B})$.

4.3.2. Упростить логическое выражение $(\bar{A} \& B) \vee (A \& B)$.

4.3.3. Упростить логическое выражение $(A \vee B) \& (\bar{A} \vee B)$.

4.3.4. Упростить логическое выражение $(A \vee B) \& (A \vee \bar{B})$.

4.4. Построение таблиц истинности логических функций в электронных таблицах

4.4.1. Построить в электронных таблицах таблицу истинности функции логического сложения (дизъюнкции).

A	B	A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1

4.4.1. Построить в электронных таблицах таблицу истинности функции логического умножения (конъюнкции).

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

4.4.2. Построить в электронных таблицах таблицу истинности функции логического следования (импликации), которая равносильна логическому выражению $\bar{A} \vee B$.

A	B	F = A → B
0	0	1
0	1	1
1	0	0
1	1	1

- 4.4.3. Построить в электронных таблицах таблицу истинности функции логического равенства (эквивалентности), которая равносильна логическому выражению $(A \& B) \vee (\bar{A} \& \bar{B})$.

A	B	$F = A \sim B$
0	0	1
0	1	0
1	0	0
1	1	1

4.5. Построение логической схемы по логической функции

- 4.5.1. Построить логическую схему для логической функции $F(A, B) = A \vee B$.
- 4.5.2. Построить логическую схему для логической функции $F(A, B) = A \& B$.
- 4.5.3. Построить логическую схему для логической функции $F(A, B) = A \& \bar{B}$.
- 4.5.4. Построить логическую схему для логической функции $F(A, B) = \bar{A} \vee B$;

Тема 5. Моделирование и формализация

5.1. Информационные модели

- 5.1.1. Информационной (знаковой) моделью является:
- 1) анатомический муляж;
 - 2) макет здания;
 - 3) модель корабля;
 - 4) химическая формула.
- 5.1.2. Материальной моделью является:
- 1) анатомический муляж;
 - 2) техническое описание компьютера;
 - 3) рисунок функциональной схемы компьютера;
 - 4) программа на языке программирования.
- 5.1.3. Какие пары объектов находятся в отношении «объект – модель»?
- 1) компьютер – данные;
 - 2) компьютер – его функциональная схема;

- 3) компьютер – программа;
- 4) компьютер – алгоритм.

5.1.4. Какая модель является статической (описывающей состояние объекта)?

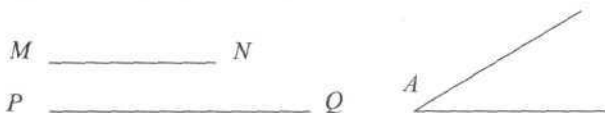
- 1) формула равноускоренного движения;
- 2) формула химической реакции;
- 3) формула химического соединения;
- 4) второй закон Ньютона.

5.2. Создание геометрических моделей

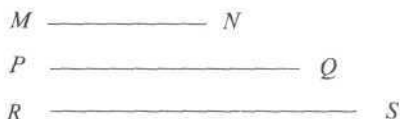
5.2.1. Выполнить в системе компьютерного черчения КОМПАС геометрическое построение «с помощью циркуля и линейки». Отложить от луча OM угол, равный заданному углу A .



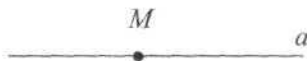
5.2.2. Выполнить в системе компьютерного черчения КОМПАС геометрическое построение «с помощью циркуля и линейки». Построить треугольник по двум сторонам и углу между ними.



5.2.3. Выполнить в системе компьютерного черчения КОМПАС геометрическое построение «с помощью циркуля и линейки». Построить треугольник по трем сторонам.



5.2.4. Выполнить в системе компьютерного черчения КОМПАС геометрическое построение «с помощью циркуля и линейки». Даны прямая и точка на ней. Построить прямую через данную точку и перпендикулярную к данной прямой.



Тема 6. Информационные технологии

6.1. Технология обработки текстовой информации. Объекты в текстовом документе

6.1.1. Абзацем в текстовом редакторе является:

- 1) фрагмент документа между двумя маркерами абзаца;
- 2) выделенный фрагмент документа;
- 3) строка символов;
- 4) фрагмент документа, начинающийся с отступа (красной строки).

6.1.2. В маркированном списке для обозначения элемента списка используются:

- 1) латинские буквы;
- 2) русские буквы;
- 3) римские цифры;
- 4) графические значки.

6.1.3. В каком случае изменится иерархическая структура оглавления документа?

- 1) изменятся стили форматирования заголовков;
- 2) изменятся тексты заголовков;
- 3) изменятся параметры форматирования абзацев;
- 4) изменятся уровни заголовков.

6.1.4. Какой текст является гипертекстом?

- 1) текст с большим размером шрифта;
- 2) текст, содержащий гиперссылки;
- 3) текст, содержащий много страниц;
- 4) текст, напечатанный на большом принтере.

6.2. Технология обработки текстовой информации. Форматирование абзацев и символов по заданному образцу

6.2.1. В текстовом редакторе открыть файл-задание Text1.txt, хранящийся на диске Windows-CD в папке ..\Test\.. Отформатировать текст, хранящийся в файле-задании, по указанному образцу. Сохранить документ в формате, сохраняющем форматирование.

Абзац с выравниванием по ширине, отступ слева 6 см, шрифт Times New Roman, размер 12 пт, обычный

Абзац с выравниванием по центру, шрифт Arial, размер 14 пт, полужирный.

Абзац с выравниванием по левому краю, отступ первой строки 1,25 см, шрифт Courier New, размер 10 пт, курсив, подчеркнутый.

6.2.2. В текстовом редакторе открыть файл-задание Text2.txt, хранящийся на диске Windows-CD в папке ..\Test\. Преобразовать текст, хранящийся в файле-задании, в многоуровневый список по указанному образцу. Сохранить документ в формате, сохраняющем форматирование.

1. Первый нумерованный элемент списка (первый уровень)
 - Первый маркированный элемент списка (второй уровень)
 - Второй маркированный элемент списка (второй уровень)
2. Второй нумерованный элемент списка (первый уровень)
 - ✓ Первый маркированный элемент списка (второй уровень)
 - ✓ Второй маркированный элемент списка (второй уровень)
3. Третий нумерованный элемент списка (первый уровень)

6.2.3. В текстовом редакторе открыть файл-задание Text3.txt, хранящийся на диске Windows-CD в папке ..\Test\. Преобразовать текст, хранящийся в файле-задании, в таблицу по указанному образцу. Вставить в таблицу столбец с номерами строк. Вычислить суммарную цену всех устройств. Оформить внешний вид таблицы в соответствии с образцом. Сохранить документ в формате, сохраняющем форматирование.

№	Наименование устройства	Цена (в у.е.)
1.	Системная плата	80
2.	Процессор	70
3.	Оперативная память	15
4.	Жесткий диск	100
5.	Монитор	200
6.	Дисковод 3,5"	12
7.	Дисковод CD-ROM	30
8.	Корпус	25
9.	Клавиатура	10
10.	Мышь	5
	Итого:	547

6.2.4. В текстовом редакторе открыть файл-задание Text4.txt, хранящийся на диске Windows-CD в папке ..\Test\. Отформатировать символы строк текста по указанному образцу. Сохранить документ в формате, сохраняющем форматирование.

Times New Roman, 14, полужирный, зеленый: N = 2!

Arial, 12, курсив, синий: 10₂ + 10₁₀ = 12₁₀;

Courier New, 16, подчеркнутый, красный: A & 1 = A.

6.3. Технология обработки графической информации. Формирование цветов в различных системах цветопередачи

- 6.3.1. Определить цвета и внести их в таблицу, если заданы интенсивности базовых цветов в системе цветопередачи RGB.

Цвет	Интенсивность базовых цветов		
	Красный	Зеленый	Синий
	00000000	00000000	00000000
	00000000	11111111	00000000
	00000000	11111111	11111111
	11111111	11111111	00000000

- 6.3.1. Определить цвета и внести их в таблицу, если заданы интенсивности базовых цветов в системе цветопередачи RGB.

Цвет	Интенсивность базовых цветов		
	Красный	Зеленый	Синий
	11111111	00000000	00000000
	00000000	00000000	11111111
	11111111	00000000	11111111
	11111111	11111111	11111111

- 6.3.1. Определить цвета, если на бумагу нанесены краски в системе цветопередачи CMYK (С — голубой, М — пурпурный, Y — желтый).

Y + M

Y + C

M + C


- 6.3.2. Определить цвета, если на бумагу нанесены краски в системе цветопередачи CMYK (С — голубой, М — пурпурный, Y — желтый).

C + Y

M + Y

C + M

6.4. Технология обработки графической информации. Создание и редактирование рисунка по заданному образцу

- 6.4.1. В векторном графическом редакторе нарисовать модель часов (круг и две стрелки). Получить уменьшенную и увеличенную копии без потери качества изображения (как показано в ответе). Сохранить графический файл.
- 6.4.2. Скопировать *Рабочий стол* компьютера в растровый графический редактор, вырезать значок *Мой компьютер* и создать изображение, состоящее из пяти значков  .
- Сохранить графический в формате, обеспечивающем минимальный информационный объем.
- 6.4.3. В графическом редакторе создать надпись «Информатика», растянуть ее в два раза, наклонить по вертикали на 45 градусов и повернуть на 180 градусов. Сохранить графический файл.
- 6.4.4. В графическом редакторе нарисовать функциональную схему компьютера. Сохранить графический файл.

6.5. Технология обработки числовой информации. Абсолютные, относительные и смешанные ссылки

- 6.5.1. Какой вид приобретут формулы, хранящиеся в диапазоне ячеек C1:C3 при их копировании в диапазон ячеек D2:D4?

	A	B	C	D	E	F	G
1			=A1+B1				
2			=\$A\$1*\$B\$1				
3			=\$A1*\$B\$1				
4							

- 6.5.2. Какой вид приобретут формулы, хранящиеся в диапазоне ячеек C1:C3 при их копировании в диапазон ячеек E2:E4?

	A	B	C	D	E	F	G
1			=A1+B1				
2			=\$A\$1*\$B\$1				
3			=\$A1*\$B\$1				
4							

- 6.5.3. Какой вид приобретут формулы, хранящиеся в диапазоне ячеек C1:C3 при их копировании в диапазон ячеек F2:F4?

	A	B	C	D	E	F	G
1			=A1+B1				
2			=\$A\$1*\$B\$1				
3			=\$A1*\$B\$1				
4							

- 6.5.4. Какой вид приобретут формулы, хранящиеся в диапазоне ячеек C1:C3 при их копировании в диапазон ячеек G2:G4?

	A	B	C	D	E	F	G
1			=A1+B1				
2			=\$A\$1*\$B\$1				
3			=\$A1*\$B\$1				
4							

6.6. Технология обработки числовой информации. Визуализация данных с помощью диаграмм и графиков

- 6.6.1. С помощью круговой диаграммы визуализировать данные о стоимости комплектующих компьютера, хранящиеся на диске Windows-CD в папке `..\Test\` в файле `Test.xls` на листе *Круговая диаграмма*. Сохранить файл электронных таблиц с круговой диаграммой в файле `Calc.xls`.

Наименование устройства	Цена (в у.е.)
Системная плата	80
Процессор	70
Оперативная память	15
Жесткий диск	100
Дисковод 3,5"	14
Монитор	200
Дисковод CD-ROM	30
Корпус	25
Клавиатура	10
Мышь	5

6.6.2. С помощью линейчатой диаграммы визуализировать данные о численности населения в некоторых странах мира, хранящиеся на диске Windows-CD в папке `..\Test\` в файле `Test.xls` на листе *Гистограмма*. Сохранить файл электронных таблиц с круговой диаграммой в файле `Calc.xls`.

Страна	Население
Китай	1273
Индия	1030
США	279
Индонезия	228
Бразилия	175
Россия	146
Бангладеш	131

6.6.3. Построить график функции $y = 5 \cdot x^2$ по ее числовому представлению, хранящийся на диске Windows-CD в папке `..\Test\` в файле `Test.xls` на листе *График*. Сохранить файл электронных таблиц с круговой диаграммой в файле `Calc.xls`.

x	-5	-4	-3	-2	-1	0	1	2	3	4	5
$y = 5 \cdot x^2$	125	80	45	20	5	0	5	20	45	80	125

- 6.6.4. С помощью линейчатой диаграммы наглядно представить сравнительную длину некоторых единиц измерения длины, хранящиеся на диске Windows-CD в папке ..\Test\ в файле Test.xls на листе *Линейчатая диаграмма*. Сохранить файл электронных таблиц с круговой диаграммой в файле Calc.xls.

Единица длины	Значение в миллиметрах
1 сантиметр	10
1 аршин	710
1 вершок	44,4
1 фут	304,8
1 дюйм	25,4

6.7. Базы данных

- 6.7.1. Записи в базе данных размещаются в:
1) ячейках; 2) строках; 3) столбцах; 4) таблицах.
- 6.7.2. Просмотр всех записей базы данных удобнее производить в:
1) отчете; 2) запросе; 3) форме; 4) таблице.
- 6.7.3. Просмотр отдельной записи базы данных удобнее производить в:
1) отчете; 2) таблице; 3) форме; 4) запросе.
- 6.7.4. Выбор записей базы данных, удовлетворяющих заданным условиям, удобнее производить в:
1) отчете; 2) таблице; 3) форме; 4) запросе.

6.8. Системы управления базами данных. Создание простой табличной базы данных

- 6.8.1. С помощью системы управления базами данных (Microsoft Access или OpenOffice Base) создать базу данных «Записная книжка», включающую столбцы (Фамилия — текстовый тип, Год рождения — дата и Возраст — числовой тип) и состоящую не менее чем из пяти записей. Произвести сортировку записей по первому столбцу. Сохранить в папке ..\Result\ файл базы данных.
- 6.8.2. С помощью системы управления базами данных (Microsoft Access или OpenOffice Base) создать базу данных «Процессоры», включающую столбцы (Название —

текстовый тип, Год выпуска — дата и Разрядность — числовой тип) и состоящую не менее чем из пяти записей. Произвести сортировку записей по второму столбцу. Сохранить в папке ..\Result\ файл базы данных.

- 6.8.3. С помощью системы управления базами данных (Microsoft Access или OpenOffice Base) создать базу данных «Переменные», включающую столбцы (Название — текстовый тип, Тип переменной — текстовый тип, и Количество занимаемых ячеек в оперативной памяти — числовой тип) и состоящую не менее чем из пяти записей. Произвести сортировку записей по третьему столбцу. Сохранить в папке ..\Result\ файл базы данных.
- 6.8.4. С помощью системы управления базами данных (Microsoft Access или OpenOffice Base) создать базу данных «Поисковые системы в Интернете», включающую столбцы (Название — текстовый тип, Адрес в Интернете — текстовый тип, и Наличие системы каталогов — логический тип) и состоящую не менее чем из пяти записей. Произвести сортировку записей по первому столбцу. Сохранить в папке ..\Result\ файл базы данных.

Тема 7. Коммуникационные технологии

7.1. Способы подключения к Интернету

- 7.1.1. Для подключения к Интернету домашнего настольного компьютера целесообразно использовать:
- 1) спутниковый канал;
 - 2) ADSL;
 - 3) GPRS;
 - 4) оптоволокно.
- 7.1.2. Для подключения к Интернету ноутбука в поездке целесообразно использовать:
- 1) спутниковый канал;
 - 2) ADSL;
 - 3) GPRS;
 - 4) оптоволокно.
- 7.1.3. Для подключения к Интернету компьютерного класса целесообразно использовать:
- 1) спутниковый канал;
 - 2) ADSL;
 - 3) GPRS;
 - 4) оптоволокно.
- 7.1.4. Для подключения к Интернету географически удаленного сервера целесообразно использовать:
- 1) спутниковый канал;
 - 2) ADSL;
 - 3) GPRS;
 - 4) оптоволокно.

7.2. Адресация в Интернете

- 7.2.1. Как правильно записывается доменное имя сервера в Интернете?
- | | |
|---------------------|---------------------|
| 1) ru.iit.metodist; | 3) iit.metodist.ru; |
| 2) ru.metodist.iit; | 4) iit.ru.metodist. |
- 7.2.2. Как правильно записывается IP-адрес компьютера в Интернете?
- | | |
|-------------------|----------------|
| 1) 83.237.199.60; | 3) 83.237.199; |
| 2) 8323719960; | 4) 237.199.60. |
- 7.2.3. При подключении к Интернету любой компьютер обязательно получает:
- 1) доменное имя;
 - 2) IP-адрес;
 - 3) доменное имя и IP-адрес;
 - 4) IP-адрес и доменное имя.
- 7.2.4. База данных доменных имен хранится:
- 1) на центральном компьютере Интернета;
 - 2) на каждом сервере Интернета;
 - 3) на серверах Интернет-провайдеров;
 - 4) иерархически распределена по серверам доменов.

7.3. Создание Web-страницы

- 7.3.1. С помощью Web-редактора (например, Компоновщика, входящего SeaMonkey — интегрированное приложение для работы в Интернете) создать Web-страницу. На странице должны быть: заголовок, разделительная линия, текст, рисунок и гиперссылка. Сохранить Web-страницу в папке ..\Result\, файле index1.htm.
- 7.3.2. С помощью Web-редактора (например, Компоновщика, входящего SeaMonkey — интегрированное приложение для работы в Интернете) создать Web-страницу. На странице должны быть: заголовок, разделительная линия, таблица и гиперссылка. Сохранить Web-страницу в папке ..\Result\, файле index2.htm.
- 7.3.3. С помощью Web-редактора (например, Компоновщика, входящего SeaMonkey — интегрированное приложение для работы в Интернете) создать Web-страницу. На странице должны быть: заголовок, разделительная линия, нумерованный список и гиперссылка. Сохранить Web-страницу в папке ..\Result\, файле index3.htm.

- 7.3.4. С помощью Web-редактора (например, Компоновщика, входящего SeaMonkey — интегрированное приложение для работы в Интернете) создать Web-страницу. На странице должны быть: заголовок, разделительная линия, маркированный список и гиперссылка. Сохранить Web-страницу в папке `..\Result\`, файле `index4.htm`.