

# ИНФОРМАТИКА

Н. Угринович, Л. Босова, Н. Михайлова

## ПРАКТИКУМ по информатике и информационным технологиям



БИНОМ

---

# И Н Ф О Р М А Т И К А

---

Н. Угринович, Л. Босова, Н. Михайлова

# П Р А К Т И К У М

по информатике  
и информационным  
технологиям

Издание 2-е, исправленное

Д о п у щ е н о  
Министерством образования Российской Федерации  
в качестве учебного пособия  
для учащихся профильных классов  
общеобразовательных учреждений



Москва  
БИНОМ. Лаборатория знаний  
2004

УДК 004.9  
ББК 32.97  
У27

**У27 Угринович Н. Д.**  
Практикум по информатике и информационным технологиям. Учебное пособие для общеобразовательных учреждений. Изд. 2-е, испр./Н. Д. Угринович, Л. Л. Босова, Н. И. Михайлова — М.: БИНОМ. Лаборатория знаний, 2004. 394 с.: ил.

ISBN 5-94774-171-7

Практикум по информатике и информационным технологиям содержит 450 задач и заданий по всем разделам курса с решениями. Практикум может быть использован как при изучении базового курса в 8–9 классах, так и при изучении профильных курсов в 10–11 классах, а также при подготовке к экзаменам по информатике в вузы, так как содержит большое количество разобранных задач по системам счисления, основам логики и программированию. На CD-ROM, который является приложением к практикуму, размещены результаты выполнения практических заданий (проекты на языке Visual Basic, компьютерные презентации, Web-сайты и др.), а также полная программная и методическая поддержка курса.

УДК 004.9  
ББК 32.97

**По вопросам приобретения обращаться:**

**(095) 955-03-98, e-mail: lbz@aha.ru**

ISBN 5-94774-171-7

© Угринович Н. Д., Босова Л. Л.,  
Михайлова Н. И., 2002, 2004  
© БИНОМ. Лаборатория знаний, 2004

# Содержание

---

<b>Рекомендации по использованию практикума</b> . . . . .	<b>6</b>
<b>Глава 1. Компьютер и программное обеспечение</b> . . . . .	<b>7</b>
1.1. Тестирование компьютера . . . . .	7
1.1.1. Тестирование процессора и оперативной памяти. . . . .	8
1.1.2. Форматирование и тестирование жестких дисков . . . . .	12
1.2. BIOS и загрузка операционной системы . . . . .	15
1.3. Файловые менеджеры и архиваторы . . . . .	22
1.4. Вирусы и антивирусные программы . . . . .	27
<b>Глава 2. Кодирование информации. Системы счисления</b> . . . . .	<b>32</b>
2.1. Количество информации . . . . .	32
2.1.1. Количество информации как мера уменьшения неопределенности знаний . . . . .	32
2.1.2. Единицы измерения количества информации. . . . .	36
2.1.3. Определение количества информации, представленной с помощью знаковых систем . . . . .	37
2.2. Системы счисления . . . . .	38
2.2.1. Непозиционные системы счисления . . . . .	38
2.2.2. Позиционные системы счисления . . . . .	42
2.3. Перевод чисел из одной системы счисления в другую . . . . .	47
2.3.1. Перевод целых чисел из одной системы счисления в другую . . . . .	47
2.3.2. Перевод дробных чисел из одной системы счисления в другую . . . . .	48
2.3.3. Перевод произвольных чисел . . . . .	49
2.3.4. Перевод чисел из системы счисления с основанием 2 в систему счисления с основанием $2^n$ и обратно. . . . .	50
2.4. Арифметические операции в позиционных системах счисления . . . . .	53
2.5. Вычисления в позиционных системах счисления с использованием калькулятора . . . . .	56
2.6. Представление чисел в компьютере . . . . .	58
2.7. Кодирование текстовой информации. . . . .	65
2.8. Кодирование графической информации . . . . .	69
2.9. Кодирование звуковой информации . . . . .	73
2.10. Хранение информации . . . . .	77
<b>Глава 3. Основы логики и логические основы компьютера</b> . . . . .	<b>84</b>
3.1. Основы логики . . . . .	84
3.2. Алгебра высказываний . . . . .	88
3.3. Логические выражения и таблицы истинности . . . . .	96



3.4. Логические функции . . . . .	99
3.5. Логические законы и правила преобразования логических выражений . . . . .	100
3.6. Логические основы компьютера . . . . .	105
<b>Глава 4. Информационные технологии . . . . .</b>	<b>109</b>
4.1. Создание текстовых документов с использованием StarOffice Writer . . . . .	109
4.2. Создание и редактирование растровых изображений с использованием StarOffice Image . . . . .	116
4.3. Создание и редактирование векторных изображений с использованием StarOffice Draw . . . . .	120
4.4. Создание мультимедийных презентаций . . . . .	124
4.4.1. StarOffice Impress — приложение для создания презентаций . . . . .	124
4.4.2. Создание презентации . . . . .	127
4.5. Система автоматизированного проектирования КОМПАС-3D . . . . .	133
4.6. Работа в электронных таблицах StarOffice Calc . . . . .	137
<b>Глава 5. Объектно-ориентированное программирование на языке Visual Basic . . . . .</b>	<b>143</b>
5.1. Интегрированная среда разработки языка Visual Basic . . . . .	144
5.2. Форма и размещение на ней управляющих элементов . . . . .	146
5.3. Размещение на формах изображений и графических полей . . . . .	151
5.4. Массивы . . . . .	158
5.5. Ввод и вывод данных . . . . .	166
5.6. Арифметические операции и функции преобразования типов данных . . . . .	170
5.7. Строковые операции и функции . . . . .	174
5.8. Создание меню и панелей инструментов проектов . . . . .	179
5.9. Логические операторы в языке Visual Basic . . . . .	191
5.10. Графические возможности языка Visual Basic . . . . .	192
5.11. Анимация . . . . .	197
5.12. Отладка программных кодов . . . . .	201
<b>Глава 6. Моделирование и формализация . . . . .</b>	<b>204</b>
6.1. Исследование математических моделей . . . . .	205
6.1.1. Приближенное решение уравнений . . . . .	205
6.1.2. Вероятностные модели . . . . .	209
6.1.3. Геометрические модели . . . . .	212
6.2. Исследование физических моделей . . . . .	215
6.3. Биологические модели развития популяций . . . . .	220
6.4. Геоинформационные модели . . . . .	226
6.5. Химические модели . . . . .	228
6.6. Оптимизационное моделирование . . . . .	230

6.7. Логические модели . . . . .	234
6.8. Информационные модели управления объектами . . . . .	237
<b>Глава 7. Информационные ресурсы Интернета . . . . .</b>	<b>240</b>
7.1. Настройка параметров соединения с Интернетом . . . . .	241
7.1.1. Подключение к Интернету по коммутируемым телефонным каналам . . . . .	241
7.1.2. Тестирование соединения с Интернетом (маршрут и скорость передачи данных) . . . . .	248
7.2. Работа с электронной почтой . . . . .	255
7.2.1. Настройка почтовой программы Outlook Express . . . . .	255
7.2.2. Работа с Web-почтой . . . . .	258
7.3. Настройка браузеров . . . . .	266
7.4. Работа с серверами файловых архивов и Web-серверами . . . . .	274
7.4.1. Загрузка файлов из файловых архивов с помощью браузеров . . . . .	274
7.4.2. Загрузка файлов из файловых архивов с помощью специализированных менеджеров загрузки файлов (Download Manager) . . . . .	276
7.4.3. Обмен файлами с FTP-серверами. . . . .	277
7.4.4. Загрузка Web-страниц и Web-сайтов с помощью off-line браузеров . . . . .	279
7.5. Поиск информации в Интернете . . . . .	281
7.6. Интерактивное общение в Интернете. . . . .	291
7.7. Поток аудио- и видео в Интернете. . . . .	293
7.8. Покупки в Интернете . . . . .	294
<b>Глава 8. Разработка Web-сайтов . . . . .</b>	<b>296</b>
8.1. HTML — язык разметки гипертекста . . . . .	296
8.2. Цветовая схема, вставка изображений, гиперссылок и таблиц . . . . .	299
8.3. Интерактивные формы на Web-страницах. . . . .	310
<b>Ответы и указания по выполнению заданий . . . . .</b>	<b>317</b>
Ответы к главе 1. Компьютер и программное обеспечение . . . . .	317
Ответы к главе 2. Кодирование информации. Системы счисления . . . . .	320
Ответы к главе 3. Основы логики и логические основы компьютера . . . . .	327
Ответы к главе 4. Информационные технологии . . . . .	332
Ответы к главе 5. Объектно-ориентированное программирование на языке Visual Basic . . . . .	338
Ответы к главе 6. Моделирование и формализация . . . . .	379
Ответы к главе 7. Информационные ресурсы Интернета . . . . .	387
Ответы к главе 8. Разработка Web-сайтов. . . . .	391

# РЕКОМЕНДАЦИИ по использованию практикума

1. В состав программно-методического комплекса по информатике и информационным технологиям входят:
  - Информатика и информационные технологии. Учебник для 10–11 классов (входит в Федеральный перечень учебников);
  - Практикум по информатике и информационным технологиям. Учебное пособие для общеобразовательных учреждений;
  - Практикум по информатике и информационным технологиям, программная и методическая поддержка курса. Электронный учебник на CD-ROM;
  - Преподавание курса «Информатика и информационные технологии» в компьютерном классе. Методическое пособие для учителей.
2. Комплекс представляет собой единую образовательную среду, связанную гиперссылками:

Установить систему  
программирования VB5.0 CSE CD-ROM 

3. Практикум ориентирован на работу в среде операционной системы Windows. Примеры пошагового выполнения практических заданий и разработки проектов обозначаются значком :



4. В тексте пособия приняты следующие обозначения и шрифтовые выделения:
  - Шрифтом Arial выделены имена программ, файлов, папок, дисков и URL-адреса в Интернете.
  - Курсивом выделены важные понятия и термины, а также названия диалоговых панелей, пунктов меню и управляющих элементов (текстовых полей, кнопок и так далее) графического интерфейса операционной системы Windows и ее приложений.
  - Шрифтом Courier выделены тексты программ на языках программирования VBA, Visual Basic и представление Web-страниц на языке разметки гипертекста (HTML).
5. Дополнительные материалы и интерактивные тесты для проверки усвоения материала находятся в Интернете по адресу: <http://iit.metodist.ru>

# Глава 1

---

## Компьютер и программное обеспечение

---

В процессе изучения данной темы рекомендуется установить следующее свободно распространяемое и условно-бесплатное программное обеспечение:

CD-ROM 

- программы тестирования компьютера SiSoft Sandra и Super System Information;
- редактор реестра WinTune;
- менеджер загрузки Acronis OS Selector;
- файловые менеджеры Windows Commander, FAR, PowerDesk;
- архиваторы WinRAR, WinZip, PowerArchiver;
- антивирусные программы: ревизор ADInf, полифаги Kaspersky Anti-Virus Personal Pro и Dr.Web.

### 1.1. Тестирование компьютера

#### 1.2. Аппаратная реализация компьютера Учебник

Производительность компьютера зависит от технических характеристик составляющих его устройств. В случаях возникновения сбоев в работе компьютера или перед его модернизацией полезно осуществить тестирование различных устройств компьютера (материнской платы, процессора, оперативной памяти, жестких дисков и др.).

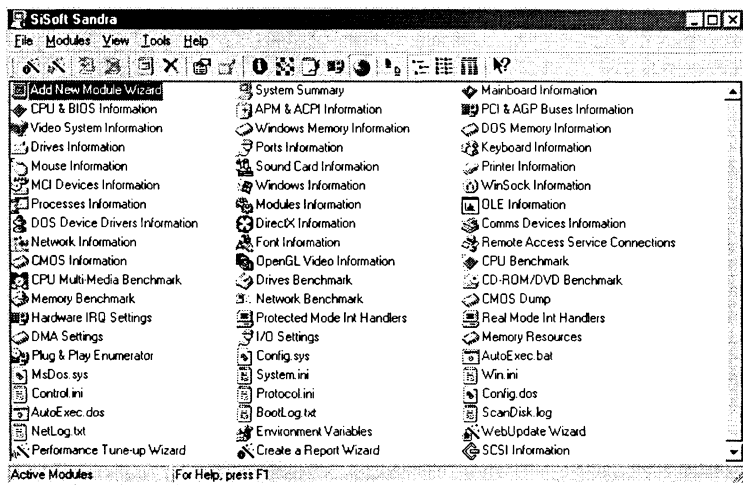
Для тестирования компьютеров используются специальные программы. Одной из наиболее удобных является программа SiSoft Sandra, которая позволяет осуществить всестороннее тестирование компьютера непосредственно из операционной системы Windows.

**Практическое задание 1.1. «Тестирование компьютера».** Произвести тестирование устройств, входящих в состав компьютера.



## Тестирование компьютера

- 1 Запустить тестирующую программу SiSoft Sandra.  
В окне программы появятся названия 62 модулей (информационные, эталонные тесты, справочные и др.).



### 1.1.1. Тестирование процессора и оперативной памяти

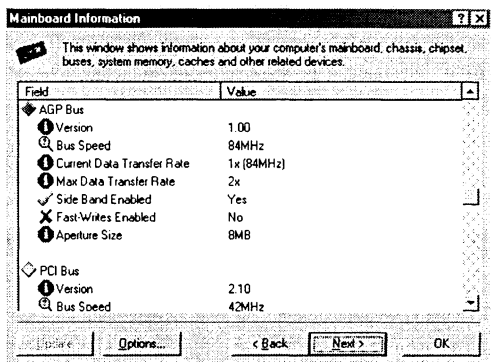
**Тестирование частоты процессора и шин.** Важной характеристикой является частота, с которой компьютер может выполнять операции по обработке данных. В процессоре используется внутреннее умножение частоты, поэтому частота процессора в несколько раз больше, чем частота системной шины. В современных компьютерах частота процессора может превышать частоту системной шины в 10 раз (например, частота процессора 1 ГГц, а частота шины — 100 МГц).

Частота шины PCI (Peripheral Component Interconnect bus — шина взаимодействия периферийных устройств), которая обеспечивает обмен информацией с контроллерами периферийных устройств, меньше частоты системной шины. Например, если частота системной шины составляет 100 МГц, то частота шины PCI обычно в три раза меньше — 33 МГц.

Для подключения видеоплаты обычно используется специальная шина AGP (Accelerated Graphic Port — ускоренный графический порт), использующая частоту в несколько раз больше, чем шина PCI.

## 2 Запустить информационный модуль *Mainboard Information* (информация о системной плате).

В окне появится информация о тестируемом компьютере, например: частота системной шины и частота шины AGP — 84 МГц, частота шины PCI — 42 МГц и др.



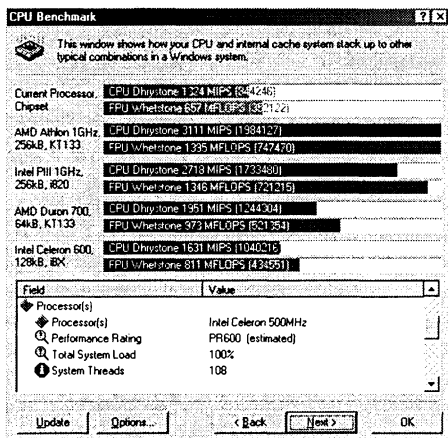
## 3 Запустить информационный модуль *CPU & BIOS Information* (информация о процессоре и BIOS).

В окне появится информация о процессоре тестируемого компьютера, например: частота 500 МГц, коэффициент умножения частоты 6 и др.

Можно не только протестировать частоту процессора, но и сравнить его производительность с эталонными современными процессорами.

## 4 Запустить тестирующий модуль *CPU benchmark* (эталонный тест процессора).

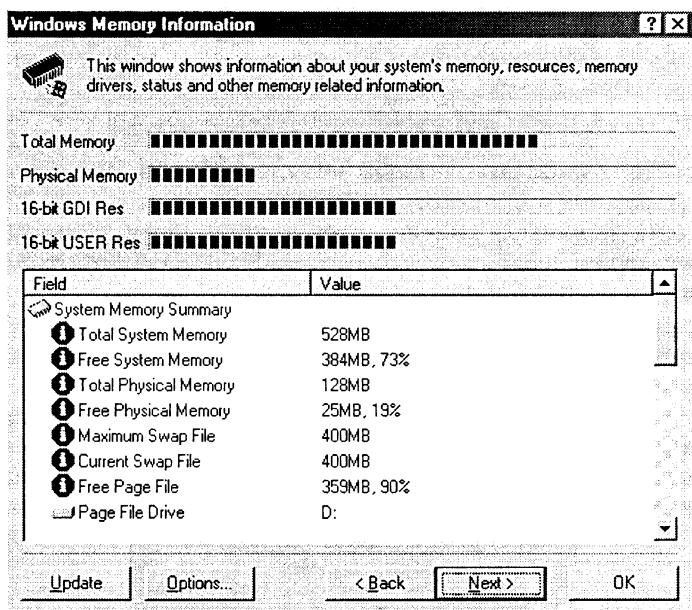
В окне появится информация о быстродействии тестируемого процессора по сравнению с другими процессорами.



**Тестирование системной оперативной памяти.** Важное значение для производительности компьютера имеет объем оперативной памяти, причем не только объем установленных модулей памяти (Physical Memory), но и общий объем системной оперативной памяти (System Memory) с учетом объема файла подкачки (Swap file).

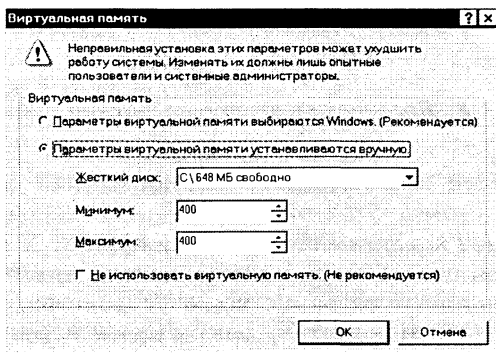
5 Запустить информационный модуль *Windows Memory Information* (информация о системной памяти).

В окне появится информация об объемах памяти тестируемого компьютера, например: системная оперативная память — 528 Мбайт, физическая память — 128 Мбайт, объем файла подкачки — 400 Мбайт и др.



Файл подкачки размещается на жестком диске, но по своей логической организации является частью оперативной памяти. Размер файла подкачки можно изменять.

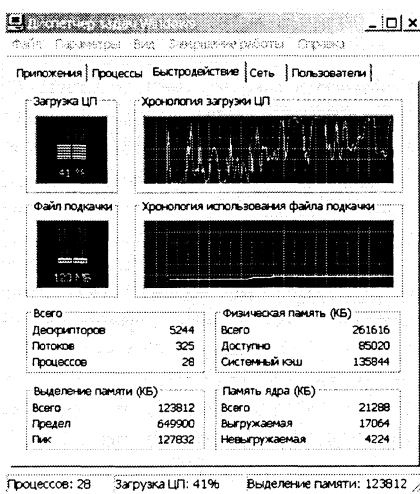
6 Щелкнуть правой кнопкой мыши по значку *Мой компьютер* и в контекстном меню активизировать пункт *Свойства*. На панели *Свойства системы* выбрать вкладку *Быстродействие* и щелкнуть по кнопке *Виртуальная память*. На появившейся одноименной панели выбрать место размещения файла подкачки и его размер.



Производительность компьютера зависит не только от частоты процессора и объема системной оперативной памяти, но и от загрузки процессора и процента использования виртуальной памяти. Производительность компьютера уменьшается, если загрузка процессора достигает 100%, а также начинает использоваться более медленная, чем физическая, виртуальная память на жестком диске. Загрузку процессора и использование виртуальной памяти в Windows 9x/Me можно наблюдать с помощью служебной программы Системный монитор, а в Windows NT/2000/XP для этого используется Диспетчер задач Windows.

7 Для вызова диспетчера задач одновременно нажать три клавиши {Ctrl}+{Alt}+{Del}.

На диалоговой панели *Диспетчер задач Windows* выбрать вкладку *Быстродействие* и наблюдать с помощью гистограмм и графиков изменение во времени загрузки процессора и использования виртуальной памяти.





## 1.1.2. Форматирование и тестирование жестких дисков

### 1.8. Логическая структура дисков Учебник

Жесткий диск состоит из следующих основных частей: магнитных пластин, оси, головок чтения/записи и встроенной электроники:

- *Магнитные пластины* собственно и являются теми жесткими дисками, сделанными из металла или пластика, от которых произошло и название всего устройства. Обе стороны каждой пластины покрыты тонким слоем намагничиваемого материала.
- Магнитные пластины закреплены на одной *оси* и вращаются на ней, как единое целое.
- У каждой магнитной стороны каждой пластины есть своя *магнитная головка чтения/записи*, головки также соединены вместе и движутся радиально по отношению к магнитным пластинам, обеспечивая, таким образом, доступ к любой точке любой пластины.
- *Встроенная электроника* предназначена для управления движением магнитных головок чтения/записи.

**Физическое форматирование.** Физическое форматирование жесткого диска выполняется производителями и делит поверхности магнитных пластин на дорожки и сектора:

- *Дорожки* — это концентрические окружности, описываемые головками чтения/записи на магнитных поверхностях. Дорожки нумеруются, начиная с нуля, по направлению к центру.
- Дорожки в свою очередь делятся на небольшие области, называемые *секторами*, содержащими фиксированное количество информации. Обычный размер сектора составляет 512 байт.

**Логическое форматирование.** Физически отформатированный жесткий диск должен быть отформатирован логически. Логическое форматирование заключается в размещении на диске файловой системы, при этом различные операционные системы могут использовать разные файловые системы:

- файловая система FAT16 широко используется операционными системами Windows 95/98/ME, а также может использоваться операционными системами Windows NT/2000/XP. Файловая система FAT16 может адресовать 65536 кластеров объемом не более 128 секторов и поэтому используется для дисков объемом не более 4 Гбайт;
- файловая система FAT32 используется для дисков большого объема, так как может адресовать более четырех миллиардов кластеров объемом по 4 Кбайта;

- файловая система NTFS является основной файловой системой Windows NT/2000/XP. В NTFS размер кластера не зависит от размера диска;
- файловая система LinuxExt2 используется в операционной системе Linux.

**Разбиение жесткого диска на разделы и их форматирование.** На жесткий диск может быть установлено одновременно несколько операционных систем. Для этого жесткий диск должен быть разбит на *разделы*, то есть независимые области на диске, в каждой из которых может быть создана своя файловая система. Таким образом, последовательность действий перед установкой операционных систем на жесткий диск должна быть следующей:

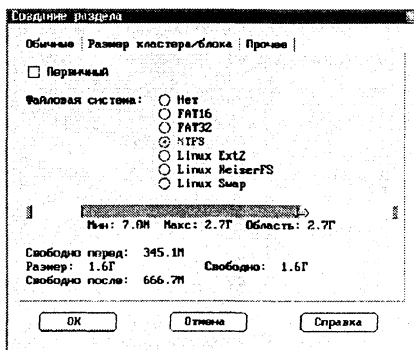
- разбиение диска на разделы;
- логическое форматирование каждого раздела под свою операционную систему.

Для разбиения диска на разделы можно использовать утилиту FDISK, а для логического форматирования утилиту FORMAT, которые входят в состав всех версий операционной системы Windows. Однако гораздо удобнее использовать специализированные дисковые утилиты (например, менеджер загрузки Acronis OS Selector). В состав Acronis OS Selector входит Администратор дисков, который позволяет разбивать жесткие диски на разделы и форматировать их.

**Практическое задание 1.2.** «Форматирование жесткого диска». Произвести разбиение диска на логические разделы и их форматирование.

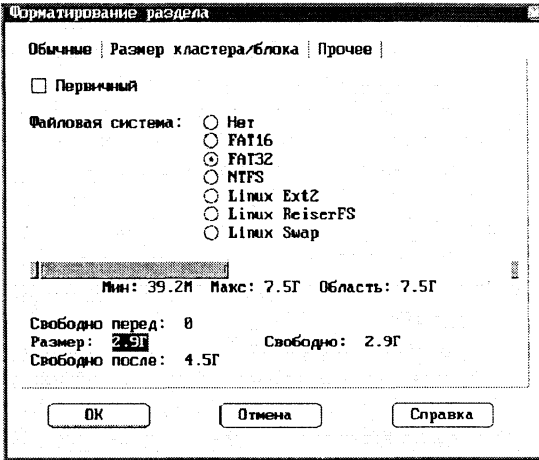
- 1 В процессе установки Acronis OS Selector создать установочную дискету. Запустить с установочной дискеты Администратор дисков.
- 2 В окне *Администратор дисков* ввести команду [Раздел-Создать...].

На появившейся панели *Создание раздела* установить с помощью ползунков границы раздела и выбрать с помощью переключателей тип файловой системы. Щелкнуть по кнопке *ОК*. Повторить процедуру для создания других разделов.



3 В окне *Администратор дисков* ввести команду [Раздел-Форматировать...].

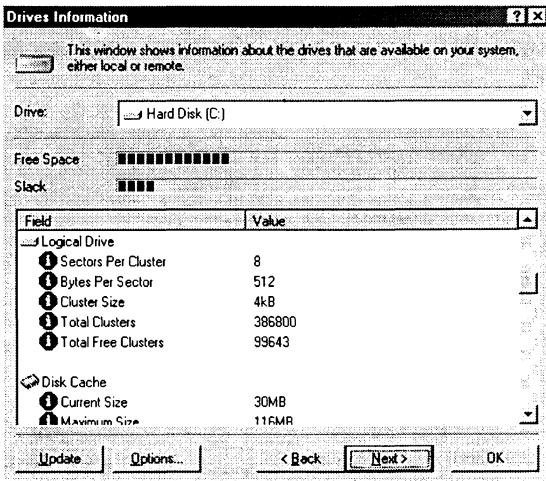
На появившейся панели *Форматирование раздела* уточнить границы раздела и тип файловой системы.



Для получения информации о дисках и логических разделах дисков можно воспользоваться программой тестирования.

4 Запустить информационный модуль *Drives Information* (информация о жестких дисках).

В окне появится информация об объемах жестких дисков, размерах сектора — 512 байт, кластера — 4 Кбайта и др.



## Задания для самостоятельного выполнения

- 1.1. Протестировать ваш компьютер с использованием программы SiSoft Sandra.
- 1.2. Получить сведения о компьютере с помощью служебной программы Windows Сведения о системе.
- 1.3. Получить сведения о компьютере с помощью программы Super System Information.
- 1.4. Если в компьютере установлен жесткий диск более 1 Гбайта, рекомендуется провести преобразование таблицы размещения файлов из FAT16 в FAT32 с помощью служебной программы Windows Преобразование диска в FAT32 или менеджера загрузки Acronis OS Selector.
- 1.5. Если на компьютере установлен жесткий диск небольшого объема (менее 500 Мбайт) можно увеличить его информационный объем путем сжатия данных на диске с помощью служебной программы Windows Сжатие диска.
- 1.6. Определить с помощью служебной программы Windows Системный монитор степень загрузки процессора и памяти в процессе работы компьютера.
- 1.7. Ознакомиться с историей развития вычислительной техники, посетив виртуальные компьютерные музеи, размещенные в Интернете (на CD-ROM имеются их описания и адреса).
- 1.8. Ознакомиться более подробно с аппаратными компонентами компьютера и периферийными устройствами на сайте [www.ixbt.ru](http://www.ixbt.ru).

## 1.2. BIOS и загрузка операционной системы

### 1.4. Загрузка операционной системы Учебник

**BIOS.** После включения компьютера процессор начинает считывать и выполнять микропрограммы, которые хранятся в микросхеме BIOS (Basic Input/Output System — базовая система ввода/вывода). Прежде всего начинает выполняться программа тестирования POST (Power On Self Test), которая проверяет работоспособность основных устройств компьютера: процессора, видеоадаптера, оперативной памяти, последовательных и параллельных портов, дисководов, контроллеров жестких дисков и клавиатуры.

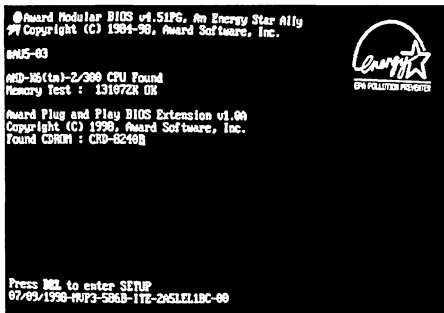
В случае обнаружения неисправностей выдаются последовательности коротких и длинных звуковых сигналов, а после инициализации видеоадаптера процесс тестирования отображается на экране монитора.

**Практическое задание 1.3. «BIOS и загрузка операционной системы».** Включить компьютер и наблюдать процесс загрузки операционной системы. С помощью программы BIOS Setup произвести установку новых параметров конфигурации компьютера.



## BIOS и загрузка операционной системы

- 1 Включить компьютер. Наблюдать процесс тестирования компьютера. В случае возникновения звуковых сигналов или сообщений об ошибке устранить неисправность.

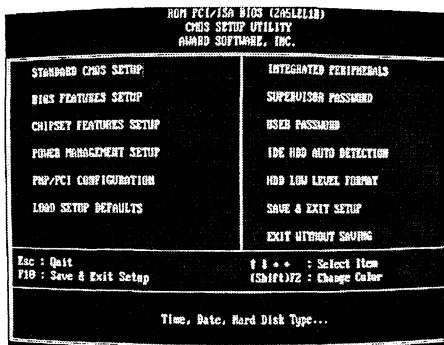


В некоторых случаях появление сообщения об ошибке связано с забывчивостью пользователя. Например, в случае появления сообщения *Invalid Boot Diskette* (невозможно загрузить операционную систему с дискеты) для продолжения загрузки необходимо просто извлечь несистемную дискету из дисковода.

В процессе тестирования BIOS сравнивает получаемые данные о конфигурации компьютера с информацией, хранящейся в CMOS — специальной микросхеме памяти, расположенной на системной плате. Если данные не совпадают, то появляется сообщение *CMOS System Option Not Set*. В этом случае необходимо с помощью утилиты BIOS Setup установить новые конфигурационные параметры.

- 2 Для входа в BIOS Setup в процессе тестирования нажать клавишу *{Del}*.

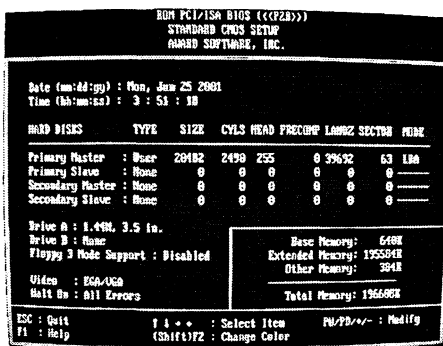
Утилита имеет интерфейс в виде системы иерархического меню, перемещение по которому производится с помощью клавиш управления курсором.



Для правильной установки даты, времени и параметров жестких и гибких дисков необходимо использовать панель *STANDARD CMOS SETUP* (стандартная установка).

- 3 Установить курсор на пункт меню *STANDARD CMOS SETUP* и нажать клавишу {Enter}.

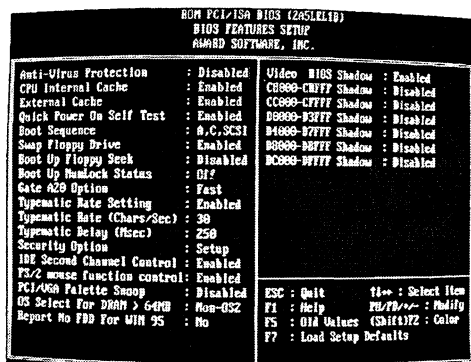
На появившейся панели установить курсор на элемент конфигурационных данных (выделяется цветом) и с помощью клавиш {PageUp} и {PageDown} установить требуемое значение параметра.



С помощью панели *BIOS FEATURES SETUP* (полная установка) можно установить защиту от заражения вирусом загрузочного сектора системного диска. В этом случае будет заблокировано любое изменение загрузочного сектора, и поэтому при переустановке или установке операционной системы защиту необходимо снять.

- 4 Установить курсор на пункт меню *BIOS FEATURES SETUP* и нажать клавишу {Enter}.

На появившейся панели установить курсор на элемент конфигурационных данных *Anti-Virus Protection* и установить значение *Enable*.



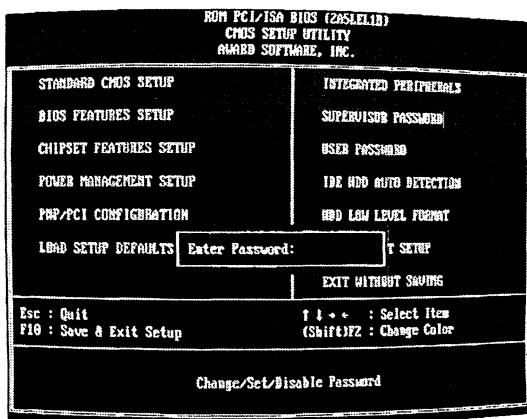
Для предохранения данных от несанкционированного доступа можно с помощью BIOS Setup установить пароль для входа в систему или в установку конфигурационных параметров. Обязательно хорошо запомните или запишите пароль, т. к. в случае его утери вход в систему становится невозможным (сброс пароля возможен только аппаратно, путем отключения микросхемы CMOS от источника питания — аккумулятора).

5 На панели *BIOS FEATURES SETUP* выбрать пункт *Security Option* и установить значение *System* или *Setup*.

6 Открыть панель *SUPERVISOR PASSWORD*.

На появившейся панели *Enter Password* ввести пароль и нажать клавишу {*Enter*}.

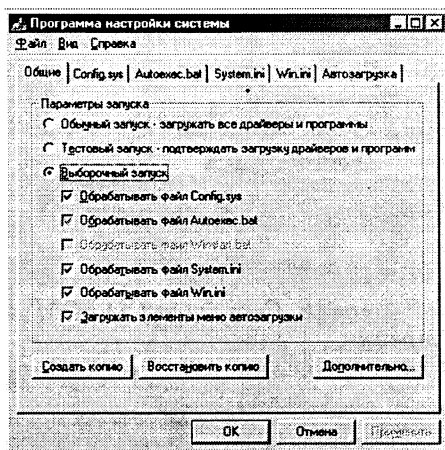
Повторно ввести пароль для подтверждения его правильности.



**Загрузка операционной системы.** После того как POST-тестирование успешно завершается, BIOS приступает к поиску программы-загрузчика Master Boot операционной системы. Современные версии BIOS позволяют загружать операционную систему не только с гибких и жестких дисков, но и с дисководов CD-ROM, ZIP и LS-120. Если программа-загрузчик найдена, она помещается в оперативную память и начинает процесс загрузки файлов операционной системы.

Системная конфигурация и загрузка драйверов устройств производится путем последовательной обработки конфигурационных файлов. Сначала обрабатываются файлы config.sys и autoexec.bat, оставшиеся в операционной системе Windows 95/98 от MS-DOS. Затем обрабатываются файлы system.ini и win.ini, которые остались от операционной системы Windows 3.x. Далее загружаются элементы, находящиеся в меню Автозагрузка.

- 7 Запустить служебную программу Windows Сведения о системе. В окне приложения ввести команду [Сервис-Программа настройки системы].  
На вкладке *Общие* можно отключить обработку конфигурационных файлов, а на других вкладках ознакомиться с их содержанием.



**Системный реестр Windows.** На самом деле, можно отказаться от обработки всех конфигурационных файлов в процессе загрузки операционной системы. В этом случае конфигурация системы будет осуществляться с помощью *системного реестра* Windows. Системный реестр 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. В Windows входит редактор реестра — программа regedit.exe, однако более удобно для работы с реестром использовать специальные утилиты, например, WinTune.

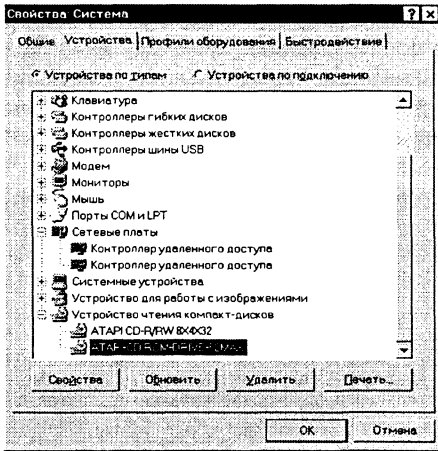
**Практическое задание 1.4. «Ознакомление с системным реестром Windows».** Ознакомьтесь с реестром на примере включения и выключения возможности автоматической загрузки CD-ROM.



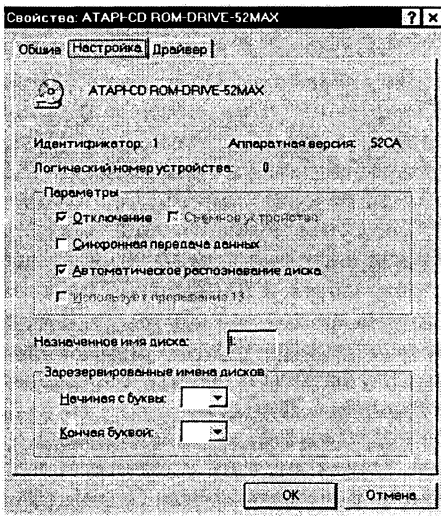
### Ознакомление с системным реестром Windows

- 1 Сначала воспользуемся для установки этой опции графическим интерфейсом. В контекстном меню объекта *Мой компьютер* выбрать пункт *Свойства*.  
На панели *Свойства: Система* выбрать вкладку *Устройства* и в списке устройств выбрать CD-ROM.  
Щелкнуть по кнопке *Свойства*.

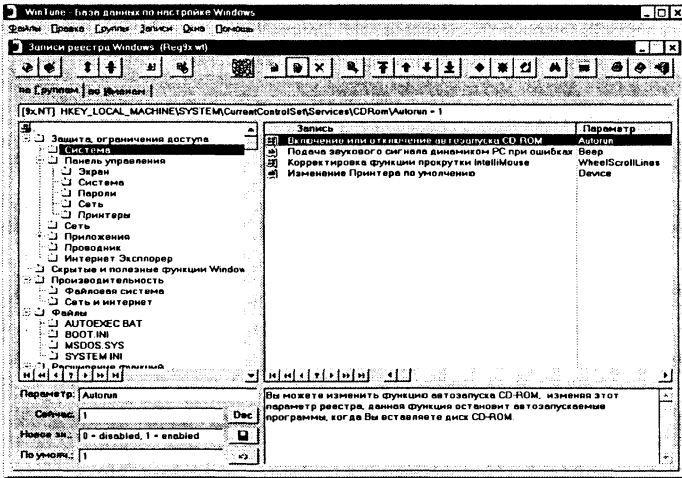




- 2 На диалоговой панели *Свойства:CD-ROM* активизировать вкладку *Настройка*.  
Установить флажок *Автоматическое распознавание диска*.



- 3 Для установки этого параметра через реестр запустить программу *WinTune*.  
В левом окне выбрать пункт *Система*, а в правом *Включение или отключение автозапуска CD-ROM*. В нижнем левом окне появится справка по установке параметра.  
В левом нижнем окне можно установить одно из возможных значений параметра: 0 — автозапуск выключен, 1 — автозапуск включен.



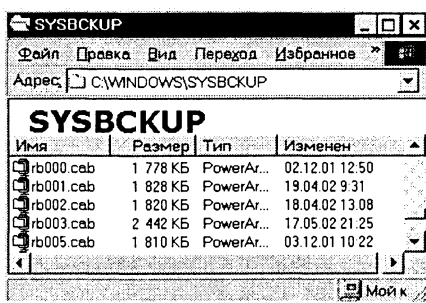
Категорически не рекомендуется изменять параметры реестра без четкого понимания производимых действий, так как ошибка при изменении реестра может серьезно повредить систему. Если система все же повреждена, реестр можно исправить или восстановить в том виде, который он имел в момент последнего удачного запуска компьютера.

4 Ввести команду [Программы-Стандартные-Служебные-Сведения о системе].

На панели *Сведения о системе* в меню *Сервис* выбрать команду *Проверка реестра*. Если программа обнаружит неполадки, реестр автоматически заменится на резервную копию.

Заменить поврежденный реестр на его резервную копию можно также вручную. Файлы пяти резервных копий реестра с именами `rb000.cab`, `rb001.cab` и так далее хранятся в каталоге `\Windows\Sysbckup\`.

5 Просмотреть каталог `\Windows\Sysbckup\` в режиме *Таблица*. Найти последнюю по времени копию реестра и скопировать ее в каталог `\Windows\`.



## Задания для самостоятельного выполнения

- 1.9. С помощью утилиты BIOS Setup ознакомиться с конфигурационными данными вашего компьютера.
- 1.10. С помощью служебной программы Сведения о системе ознакомиться с конфигурационными файлами вашего компьютера.
- 1.11. С помощью служебной программы Редактор реестра (regedit.exe, которая находится в папке Windows) ознакомиться с системным реестром вашего компьютера.

## 1.3. Файловые менеджеры и архиваторы

Для проведения различных операций над файлами и папками (копирование, перемещение, удаление, переименование и др.) используются специальные приложения — *файловые менеджеры*.

Для долговременного хранения или передачи по компьютерным сетям файлы архивируются (сжимаются) с помощью файловых менеджеров и специализированных приложений — *архиваторов*.

**Алгоритмы и методы архивации.** Существуют различные алгоритмы архивации данных без потери информации, при которых при разархивации данные будут восстановлены в исходном виде. Самый простой алгоритм сжатия данных (run-length) основан на замене повторяющихся битов (в тексте может иметься последовательность одинаковых символов, в графическом файле — закрашенная одним цветом область и так далее). Например, в тексте подряд идут 10 пробелов, которые кодируются 10-ю байтами. При архивации они заменяются 3-мя байтами (первый байт — кодирует заменяемый символ; второй байт — специальный байт «флажка» архивации, который указывает на необходимость развернуть первый байт в последовательность байтов; третий байт указывает количество повторяющихся байтов).

Алгоритм кодирования одинаковых последовательностей символов (LZW) ищет в текстовых файлах одинаковые слова, а в графических — одинаковые «узоры». Каждый такой фрагмент файла представляется определенным кодом (последовательностью бит) и в процессе архивации при повторных появлениях заменяется ссылкой на первичный код.

Существуют различные методы архивации файлов (ZIP, RAR, ARJ и др.), которые используют вышеописанные и другие алгоритмы архивации. Методы архивации различаются степенью сжатия файлов, скоростью выполнения и другими параметрами. Лучше всего сжимаются текстовые и графические файлы и практически не сжимаются файлы архивов.

**Файловые менеджеры.** В Windows стандартным файловым менеджером является Проводник, который реализует оконный интерфейс и технологию работы с помощью мыши. Широко известен файловый менеджер Norton Commander, использующий две панели для отображения файлов.

Некоторые современные файловые менеджеры, например, Windows Commander используют оконный интерфейс, панельное отображение файлов и поддерживают технологию работы с помощью мыши.

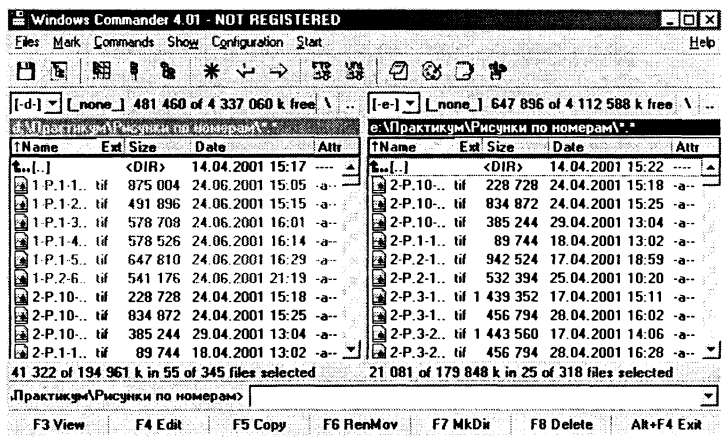
**Практическое задание 1.5. «Операции над файлами».** С помощью файлового менеджера Windows Commander осуществить сравнение содержимого папок, копирование, просмотр иерархической файловой системы и архивирование файлов.

Часто необходимо сравнить содержание двух папок и скопировать отсутствующие файлы из одной папки в другую.



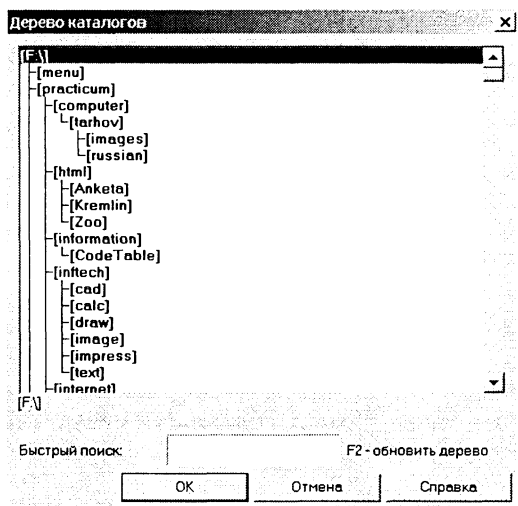
## Операции над файлами

- 1 Запустить файловый менеджер Windows Commander. Ввести команду [Выделение-Сравнить каталоги], отсутствующие в другой папке файлы будут выделены красным цветом. Щелкнуть по кнопке *F5 Копия*, начнется процесс копирования файлов.



С помощью файловых менеджеров можно ознакомиться с иерархической файловой системой компьютера.

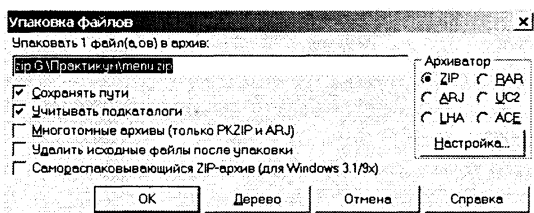
- 2 Ввести команду [Команды-Дерево каталогов]. В появившемся дополнительном окне *Дерево каталогов* будет отображена иерархическая файловая система одного из дисков.



Менеджеры файлов обычно включают функцию архивации и разархивации файлов. Для архивации файлов необходимо выделить файл или группу файлов, указать метод архивации и место хранения архивного файла.

### 3 Ввести команду [Файл-Упаковать...].

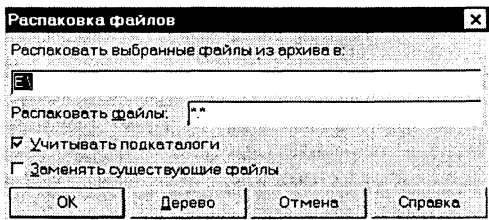
В появившемся окне *Упаковка файлов* указать параметры архивации.



Для извлечения файлов из архивов необходимо выделить файл архива и указать место сохранения извлеченных файлов.

### 4 Ввести команду [Файл-Распаковать...].

В появившемся окне *Распаковка файлов* указать параметры разархивации.



**Архиваторы.** Существует большое количество специализированных программ — архиваторов файлов (WinZip, WinRAR, PowerArchiver и др.).

Одним из наиболее популярных средств создания архивов и управления ими является полностью русифицированный архиватор WinRAR, версии которого существуют для различных операционных систем: MS-DOS, Windows, Linux и др.

WinRAR позволяет работать с архивами RAR и ZIP, использует высокоэффективный алгоритм сжатия данных, создает многотомные и самораспаковывающиеся архивы и так далее.

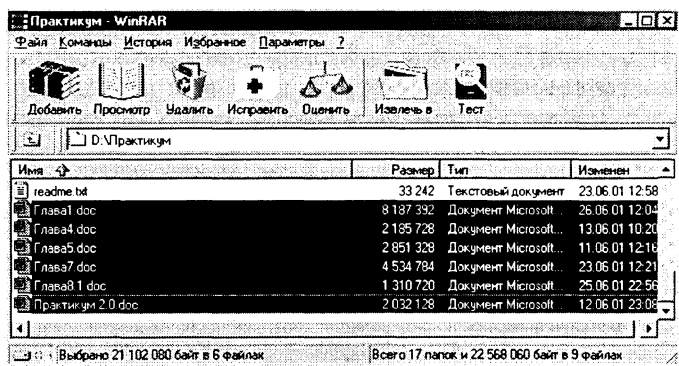
**Практическое задание 1.6. «Архивирование файлов».** С помощью архиватора WinRAR осуществить архивирование файлов и извлечение файлов из архивов.



## Архивирование файлов

1 Запустить архиватор WinRAR.

Выделить файлы и ввести команду [Команды-Добавить файлы в архив].



При создании нового архива нужно задать параметры архивирования. Прежде всего, необходимо задать имя архивного файла и место его сохранения на диске. Далее, нужно выбрать *формат архивации* RAR или ZIP (формат ZIP более широко распространен, а метод RAR обеспечивает больше возможностей и более сильное сжатие).

В обоих форматах поддерживаются шесть *методов архивации*: *Без сжатия*, *Скоростной*, *Быстрый*, *Обычный*, *Хороший* и *Максимальный*. *Максимальный* метод обеспечивает наиболее высокую степень сжатия, но с наименьшей скоростью. Напротив, *Скоростной* сжимает плохо, но очень быстро. Метод *Без сжатия* просто помещает файлы в архив без их упаковки. Если вы создаете архив для передачи по компьютерным сетям

или для долговременного хранения, имеет смысл выбрать метод *Максимальный* для получения наилучшего сжатия. Если же вы создаете ежедневную резервную копию данных, то, как правило, лучше использовать *Обычный* метод.

Следующий параметр архивации — *размер словаря*. Он может принимать значения 64, 128, 256, 512 и 1024 Кб. Чем больше размер словаря, тем лучше, но медленнее сжатие.

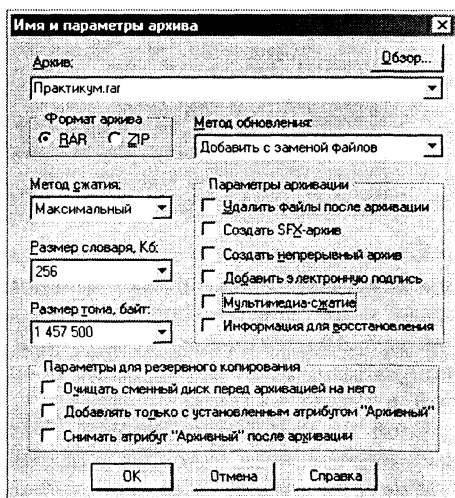
WinRAR позволяет создавать *многотомные архивы*, то есть архивы, состоящие из нескольких частей. Обычно тома используются для сохранения большого архива на нескольких дисках или других сменных носителях. Первый том архива имеет обычное расширение rar, а расширения последующих томов нумеруются как r00, r01, r02 и так далее.

Архив может быть *непрерывным* (позволяет добиться максимальной степени сжатия) и *самораспаковывающимся* (SFX, от англ. Self-extracting). Для разархивации такого архива не нужна специальная программа, достаточно запустить файл архива на выполнение, так как он является исполняемым файлом и имеет расширение .exe.

Для архивации звуковых и графических файлов может использоваться дополнительный специальный метод *мультимедиа-сжатие*, при котором может быть достигнута на 30% более высокая степень сжатия, чем при обычном сжатии.

**2** На панели *Имя и параметры* архива задать параметры архивирования:

- имя архивного файла;
- место хранения архива на диске;
- формат архива;
- метод сжатия;
- размер словаря;
- размер тома;
- метод обновления;
- параметры архивации.



**3** Для извлечения файлов из архива необходимо выбрать файл архива и ввести команду [Команды-Извлечь из выделенных архивов].

## Задания для самостоятельного выполнения

- 1.12. С помощью файлового менеджера FAR (бесплатного аналога Norton Commander) осуществить сравнение содержимого папок, копирование, просмотр иерархической файловой системы и архивирование файлов.
- 1.13. С помощью файлового менеджера PowerDesk осуществить сравнение содержимого папок, копирование, просмотр иерархической файловой системы и архивирование файлов.
- 1.14. С помощью архиватора WinZip осуществить архивирование файлов и извлечение файлов из архивов.
- 1.15. С помощью архиватора PowerArchiver осуществить архивирование файлов и извлечение файлов из архивов.

## 1.4. Вирусы и антивирусные программы

### 1.10. Компьютерные вирусы и антивирусные программы

Учебник



**Классификация вирусов.** Существуют различные типы компьютерных вирусов: *загрузочные, файловые, макро-вирусы и сетевые.*

*Загрузочные вирусы* заражают загрузочный сектор гибкого диска или винчестера. При заражении дисков загрузочный вирус «заставляет» систему при ее перезапуске считать в память и отдать управление не программному коду загрузчика операционной системы, а коду вируса.

*Файловые вирусы* при своем размножении тем или иным способом используют файловую систему операционной системы. Файловые вирусы могут поражать исполняемые файлы различных типов (EXE, COM, BAT, SYS и др.).

Практически все загрузочные и файловые вирусы *резидентны*, то есть 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 так далее). Лечение от резидентных вирусов затруднено, так как даже после удаления зараженных файлов с дисков вирус остается в оперативной памяти и возможно повторное заражение файлов.

*Макро-вирусы* являются программами на языках, встроенных в некоторые системы обработки данных (текстовые редакторы, электронные таблицы и так далее). Для своего размножения такие вирусы используют возможности макро-языков и при их помощи переносят себя из одного зараженного файла (документа или таблицы) в другие. Наибольшее распространение получили макро-вирусы для Microsoft Office, использующие возможности языка Visual Basic for Applications.



При работе с документом пользователь выполняет различные действия: открывает документ, сохраняет, печатает, закрывает и так далее. При этом приложение ищет и выполняет соответствующие *стандартные макросы*. Макро-вирусы содержат стандартные макросы, вызываются вместо них и заражают каждый открываемый или сохраняемый документ. Вредные действия макро-вирусов реализуются с помощью встроенных макросов (вставки текстов, запрета выполнения команд меню приложения и так далее).

Макро-вирусы являются *ограниченно резидентными*, то есть они находятся в оперативной памяти и заражают документы до тех пор, пока открыто приложение. Кроме этого, макро-вирусы заражают шаблоны документов, и поэтому активизируются уже при запуске зараженного приложения.

*Сетевые вирусы* для своего распространения используют протоколы и возможности локальных и глобальных компьютерных сетей. Основным принципом работы сетевых вирусов является возможность передать и запустить свой код на удаленном компьютере.

**Антивирусные программы.** Для защиты от вирусов и лечения зараженного компьютера используются антивирусные программы, которые по принципу действия можно разделить на *блокировщики, ревизоры и полифаги*.

*Антивирусные блокировщики* — это резидентные программы, перехватывающие «вирусоопасные» ситуации и сообщающие об этом пользователю. Например, «вирусоопасной» является запись в загрузочные сектора дисков, которую можно запретить с помощью программы BIOS Setup (смотри параграф 1.2).

*Ревизоры.* Принцип работы ревизоров основан на подсчете контрольных сумм для хранящихся на диске файлов. Эти суммы, а также некоторая другая информация (длины файлов, даты их последней модификации и др.) сохраняются в базе данных антивируса. При последующем запуске ревизоры сверяют данные, содержащиеся в базе данных, с реально подсчитанными значениями. Если информация о файле, записанная в базе данных, не совпадает с реальными значениями, то ревизоры сигнализируют о том, что файл был изменен или заражен вирусом.

*Полифаги.* Принцип работы полифагов основан на проверке файлов, секторов и системной памяти и поиске в них известных и новых (неизвестных полифагу) вирусов. Для поиска известных вирусов используются маски вирусов (некоторая постоянная последовательность программного кода, специфичная для каждого конкретного вируса).

Во многих полифагах используются также алгоритмы *эвристического сканирования*, то есть анализ последовательности

команд в проверяемом объекте, набор некоторой статистики и принятие решения (*возможно заражен или не заражен*) для каждого проверяемого объекта.

*Полифаги-мониторы* постоянно находятся в оперативной памяти компьютера и проверяют все файлы в реальном режиме времени. *Полифаги-сканеры* производят проверку системы по команде пользователя.

**Практическое задание 1.7. «Защита от вирусов: обнаружение и лечение».** С помощью антивирусных программ проверить ваш компьютер на наличие вирусов и при их обнаружении вылечить.

Надежную защиту от загрузочных вирусов можно установить с помощью антивирусного блокировщика, имеющегося в BIOS Setup.



### Защита от вирусов: обнаружение и лечение

- 1 Установить с помощью программы BIOS Setup защиту загрузочного сектора системного диска от вирусов.

С помощью ревизора ADInf32 можно получить информацию о подозрительных изменениях в компьютере. При первом запуске в таблицах запоминаются объем оперативной памяти, образы главного загрузочного сектора, загрузочных секторов, список сбойных кластеров, структура дерева каталогов, длины и контрольные суммы файлов.

- 2 Запустить ревизор ADInf32.

Выбрать в окне диски для проверки и щелкнуть по кнопке *Старт*.

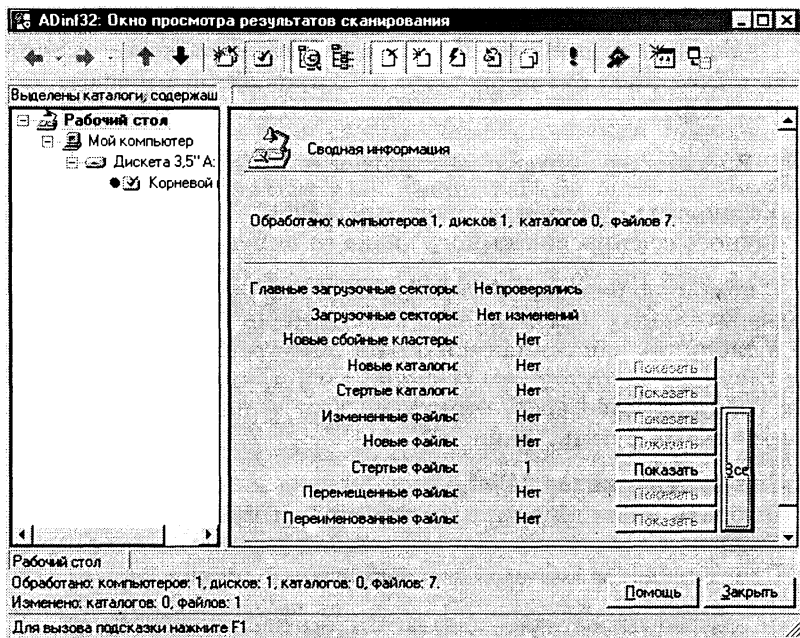
Будут созданы таблицы для выбранных дисков.



Когда вирус заражает компьютер, он изменяет объект, в который внедряется (файл или загрузочный сектор). Запуск ревидора ADInf32 позволяет обнаружить изменения, которые могут свидетельствовать о заражении вирусами.

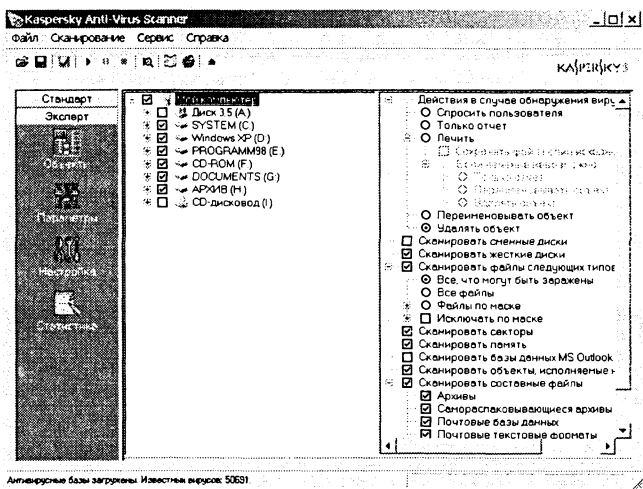
3 Выбрать в окне диски для проверки и щелкнуть по кнопке *Старт*.

Будет проведено сравнение данных, хранящихся в таблицах, с реальными данными. Щелкнуть по кнопке *Результат* и проанализировать полученные данные.

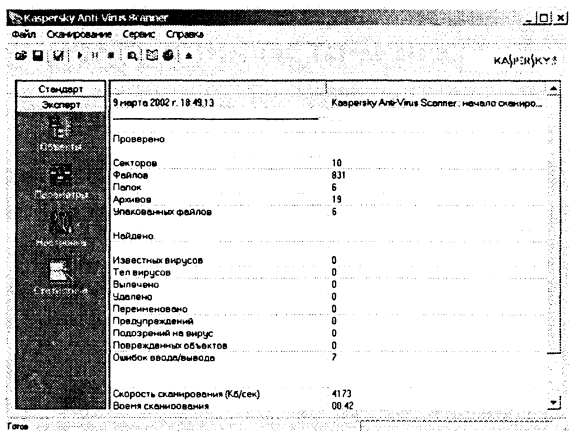


В случае обнаружения изменений в таблицах ревидора ADInf и/или сбоев в работе компьютера необходимо для лечения компьютера от вируса использовать антивирусный полифаг, например, Kaspersky Anti-Virus Scanner.

4 Запустить полифаг-сканер Kaspersky Anti-Virus Scanner. В появившемся окне приложения активизировать значок *Объекты*. В левой панели выбрать диски и папки для проверки. В правой панели выбрать тип действия в случае обнаружения вируса (*Спросить пользователя, Только отчет, Лечить, Удалить объект*) и тип проверяемых объектов (оперативная память, загрузочный сектор, файлы различных типов и так далее).



- 5 Щелчком по кнопке *Пуск* начать проверку. После окончания проверки щелкнуть по значку *Статистика*. В окне приложения появится информация о количестве проверенных секторов, папок и файлов, об обнаруженных вирусах и вылеченных файлах.



### Задания для самостоятельного выполнения

- 1.16. С помощью полифага-сканера Dr.Web проверить ваш компьютер на наличие вирусов и при их обнаружении вылечить.
- 1.17. С использованием Вирусной энциклопедии Касперского ознакомиться с классификацией вирусов и методами антивирусной защиты. Энциклопедия находится в Интернете по адресу <http://www.kaspersky.ru> и на CD-ROM в каталоге \soft\computer\antivir\VirEnc\.

# Глава 2

---

## Кодирование информации. Системы счисления

---

В процессе изучения данной темы рекомендуется установить следующее свободно распространяемое программное обеспечение:

CD-ROM 

- программы перевода единиц измерения Versaverter и Advanced Converter;
- калькуляторы Wise Calculator и NumLock Calculator для проведения вычислений в различных системах счисления;
- текстовый редактор-конвертер Hieroglyph для перевода текстов из одной кодировки кириллицы в другую;
- звуковой редактор Cool Edit 2000.

### 2.1. Количество информации

---

2.4. Формула Шеннона Учебник 

---

#### 2.1.1. Количество информации как мера уменьшения неопределенности знаний

**Количество информации как мера уменьшения неопределенности знаний.** Информацию, которую получает человек, можно считать мерой уменьшения неопределенности знаний. Если некоторое сообщение приводит к уменьшению неопределенности наших знаний, то можно говорить, что такое сообщение содержит информацию.

Сообщения обычно содержат информацию о каких-либо событиях. Количество информации для событий с различными вероятностями определяется по формуле:

$$I = -\sum_{i=1}^N p_i \log_2 p_i, \quad (2.1)$$

где  $I$  — количество информации;

$N$  — количество возможных событий;

$p_i$  — вероятности отдельных событий.

Если события равновероятны, то количество информации определяется по формуле:

$$I = \log_2 N \quad (2.2)$$

или из показательного уравнения:

$$N = 2^I. \quad (2.3)$$

**Пример 2.1.** После экзамена по информатике, который сдавали ваши друзья, объявляются оценки («2», «3», «4» или «5»). Какое количество информации будет нести сообщение об оценке учащегося *A*, который выучил лишь половину билетов, и сообщение об оценке учащегося *B*, который выучил все билеты.

Опыт показывает, что для учащегося *A* все четыре оценки (события) равновероятны и тогда количество информации, которое несет сообщение об оценке, можно вычислить по формуле (2.2):

$$I = \log_2 4 = 2 \text{ бита.}$$

На основании опыта можно также предположить, что для учащегося *B* наиболее вероятной оценкой является «5» ( $p_1 = 1/2$ ), вероятность оценки «4» в два раза меньше ( $p_2 = 1/4$ ), а вероятности оценок «2» и «3» еще в два раза меньше ( $p_3 = p_4 = 1/8$ ). Так как события неравновероятны, воспользуемся для подсчета количества информации в сообщении формулой (2.1):

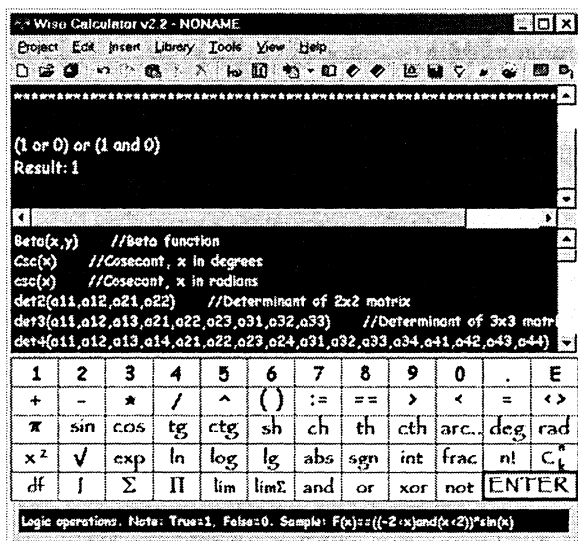
$$I = - (1/2 \cdot \log_2 1/2 + 1/4 \cdot \log_2 1/4 + 1/8 \cdot \log_2 1/8 + 1/8 \cdot \log_2 1/8) \text{ бит} = 1,75 \text{ бита.}$$

Вычисления показали, что при равновероятных событиях мы получаем большее количество информации, чем при неравновероятных событиях.

**Компьютерные калькуляторы.** Для вычисления количества информации, перевода чисел из одной системы счисления в другую и проведения вычислений в различных системах счисления можно использовать приложения, которые называются обычно научными или инженерными калькуляторами. В состав операционной системы Windows входит стандартное приложение Калькулятор, существуют также самостоятельные приложения такого назначения.

Одним из наиболее удачных и многофункциональных научных калькуляторов является Wise Calculator (умный калькулятор), который позволяет не только производить вычисления в десятичной, двоичной, восьмеричной и шестнадцатеричной системах счисления, но определять значения логических выражений, осуществлять перевод из одних единиц измерения в другие, строить графики функций и так далее.

Рис. 2.1  
Wise Calculator



**Пример 2.2.** В непрозрачном мешочке хранятся 10 белых, 20 красных, 30 синих и 40 зеленых шариков. Какое количество информации будет содержать зрительное сообщение о цвете вынутого шарика.


Так как количество шариков различных цветов неодинаково, то вероятности зрительных сообщений о цвете вынутого из мешочка шарика также различаются и равны количеству шариков данного цвета деленному на общее количество шариков:

$$p_b = 0,1; \quad p_k = 0,2; \quad p_c = 0,3; \quad p_z = 0,4.$$

События неравновероятны, поэтому для определения количества информации, содержащегося в сообщении о цвете шарика, воспользуемся формулой (2.1):

$$I = -(0,1 \cdot \log_2 0,1 + 0,2 \cdot \log_2 0,2 + 0,3 \cdot \log_2 0,3 + 0,4 \cdot \log_2 0,4) \text{ бит.}$$

Для вычисления этого выражения, содержащего логарифмы, воспользуемся компьютерным калькулятором.

 **Вычисление количества информации с использованием калькулятора**

- 1 Запустить программу Wise Calculator.
- 2 Ввести выражение и получить результат:

$$-(0,1 \cdot \log_2(2,0,1) + 0,2 \cdot \log_2(2,0,2) + 0,3 \cdot \log_2(2,0,3) + 0,4 \cdot \log_2(2,0,4))$$

Result: 1.846439344671015

Таким образом,  $I \approx 1,85$  бита.

**Пример 2.3.** Какое количество вопросов достаточно задать вашему собеседнику, чтобы наверняка определить месяц, в котором он родился?

Будем рассматривать 12 месяцев как 12 возможных событий. Если спрашивать о конкретном месяце рождения, то, возможно, придется задать 11 вопросов (если на 11 первых вопросов был получен отрицательный ответ, то 12-й задавать не обязательно, так как он и будет правильным).

Правильнее задавать «двоичные» вопросы, то есть вопросы, на которые можно ответить только «Да» или «Нет». Например, «Вы родились во второй половине года?». Каждый такой вопрос разбивает множество вариантов на два подмножества: одно соответствует ответу «Да», а другое — ответу «Нет».

Правильная стратегия состоит в том, что вопросы нужно задавать так, чтобы количество возможных вариантов каждый раз уменьшалось вдвое. Тогда количество возможных событий в каждом из полученных подмножеств будет одинаково и их отгадывание равновероятно. В этом случае на каждом шаге ответ («Да» или «Нет») будет нести максимальное количество информации (1 бит).

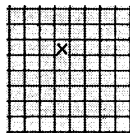
По формуле (2.2) и с помощью калькулятора получаем:

$$I = \log_2 12 \approx 3,6 \text{ бита.}$$

Количество полученных бит информации соответствует количеству заданных вопросов, однако количество вопросов не может быть нецелым числом. Округляем до большего целого числа и получаем ответ: при правильной стратегии необходимо задать не более 4 вопросов.

### Задания для самостоятельного выполнения

- 2.1. Какое количество информации несет в себе сообщение о том, что нужная вам программа находится на одной из восьми дискет?
- 2.2. Какое количество информации получит второй игрок при игре в крестики-нолики на поле  $8 \times 8$ , после первого хода первого игрока, играющего крестиками?
- 2.3. В рулетке общее количество лунок равно 128. Какое количество информации мы получаем в зрительном сообщении об остановке шарика в одной из лунок?
- 2.4. Происходит выбор одной карты из колоды в 32 карты. Какое количество информации мы получаем в зрительном сообщении о выборе определенной карты?
- 2.5. Какое количество информации будет содержать зрительное сообщение о цвете вынутого шарика, если в непрозрачном мешочке хранятся:
  - а) 25 белых, 25 красных, 25 синих и 25 зеленых шариков;
  - б) 30 белых, 30 красных, 30 синих и 10 зеленых шариков?
- 2.6. Какое количество вопросов достаточно задать вашему собеседнику, чтобы точно определить день и месяц его рождения?





## 2.1.2. Единицы измерения количества информации

**Единицы измерения количества информации.** За единицу количества информации принят 1 бит — количество информации, содержащейся в сообщении, уменьшающем неопределенность знаний в два раза.

Принята следующая система единиц измерения количества информации:

1 байт = 8 бит;

1 Кбайт =  $2^{10}$  байт;

1 Мбайт =  $2^{10}$  Кбайт =  $2^{20}$  байт;

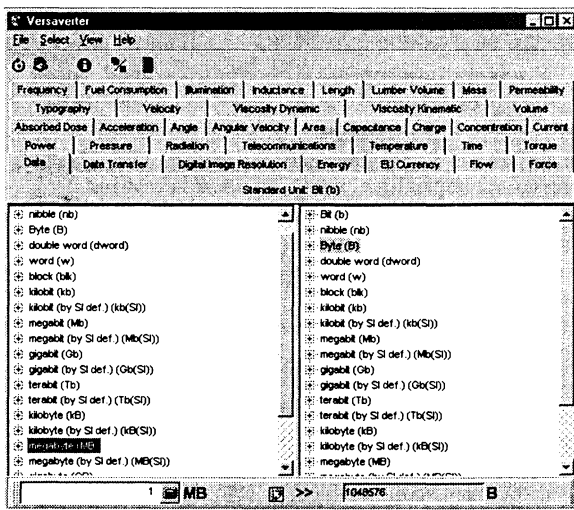
1 Гбайт =  $2^{10}$  Мбайт =  $2^{20}$  Кбайт =  $2^{30}$  байт.

**Пример 2.4.** Произведем перевод количества информации из одной единицы в другую с помощью специализированной программы Преобразование единиц измерения.



### Единицы измерения количества информации

- 1 Запустить программу Versaverter (преобразование единиц измерения).
- 2 На открывшейся диалоговой панели выбрать вкладку *Data*.
- 3 В левом окне диалоговой панели выбрать исходную единицу измерения (например, *megabyte*) и в текстовом окне ввести числовое значение (например, 1).  
После выбора в правом окне конечной единицы измерения (например, *Byte*), в текстовом окне появится числовое значение (в данном случае, 1 048 576).



## Задания для самостоятельного выполнения

2.7. Заполнить пропуски числами и проверить правильность вычислений с помощью программы перевода единиц измерения Advanced Converter:

- а) 5 Кбайт = \_\_ байт = \_\_ бит;
- б) \_\_ Кбайт = \_\_ байт = 12288 бит;
- в) \_\_ Кбайт = \_\_ байт =  $2^{13}$  бит;
- г) \_\_ Гбайт = 1536 Мбайт = \_\_ Кбайт;
- д) 512 Кбайт = 2\_\_ байт = 2\_\_ бит.

2.8 Найти  $x$  из следующих соотношений:

- а)  $16^x$  бит = 32 Мбайт;
- б)  $8^x$  Кбайт = 16 Гбайт.

### 2.1.3. Определение количества информации, представленной с помощью знаковых систем

Если рассматривать символы алфавита как множество возможных сообщений (событий)  $N$ , то количество информации, которое несет один знак, можно определить из формулы (2.1). Если считать появление каждого знака алфавита в тексте событиями равновероятными, то для определения количества информации можно воспользоваться формулой (2.2) или уравнением (2.3).

Количество информации, которое несет один знак алфавита, тем больше, чем больше знаков входят в этот алфавит, то есть чем больше мощность алфавита.

Количество информации, содержащейся в сообщении, закодированном с помощью знаковой системы, равно количеству информации, которое несет один знак, умноженному на число знаков в сообщении.

**Пример 2.5.** Какова мощность алфавита, с помощью которого записано сообщение, содержащее 2048 символов, если его объем составляет 1,25 Кбайта.

С помощью программы Versaverter или арифметически перевести информационный объем сообщения в биты:

$$I = 10\ 240 \text{ бит.}$$

Определить количество бит, приходящееся на один символ:

$$10\ 240 \text{ бит} : 2\ 048 = 5 \text{ бит.}$$

По формуле (2.3) определить количество символов в алфавите:

$$N = 2^I = 2^5 = 32.$$

## Задания для самостоятельного выполнения

- 2.9. Какова мощность алфавита, с помощью которого записано сообщение, содержащее 2048 символов, если его объем составляет  $1/512$  часть одного мегабайта?
- 2.10. Пользователь компьютера, хорошо владеющий навыками ввода информации с клавиатуры, может вводить в минуту 100 знаков. Мощность алфавита, используемого в компьютере, равна 256. Какое количество информации в байтах может ввести пользователь в компьютер за 1 минуту?
- 2.11. Система оптического распознавания символов позволяет преобразовывать отсканированные изображения страниц документа в текстовый формат со скоростью 4 страницы в минуту и использует алфавит мощностью 65536 символов. Какое количество информации будет нести текстовый документ после 5 минут работы приложения, страницы которого содержат 40 строк по 50 символов?

## 2.2. Системы счисления

### 2.6. Представление числовой информации с помощью систем счисления

Учебник 

«Все есть число», — говорили пифагорейцы, подчеркивая необычайно важную роль чисел в практической деятельности. Известно множество способов представления чисел. В любом случае число изображается символом или группой символов (словом) некоторого алфавита. Будем называть такие символы цифрами. Для представления чисел используются непозиционные и позиционные системы счисления.

### 2.2.1. Непозиционные системы счисления

Как только люди начали считать, у них появилась потребность в записи чисел. Находки археологов на стоянках первобытных людей свидетельствуют о том, что первоначально количество предметов отображали равным количеством каких-либо значков (бирок): зарубок, черточек, точек.

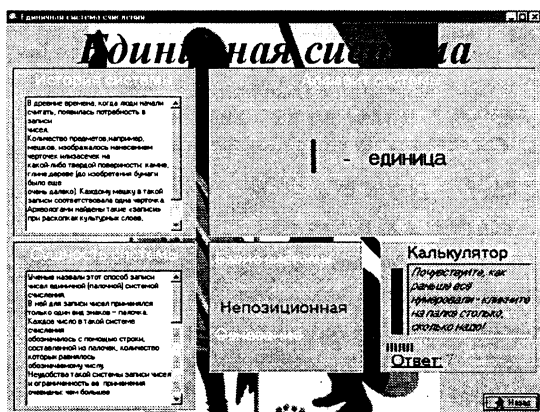
Позже, для облегчения счета, эти значки стали группировать по три или по пять. Такая система записи чисел называется единичной (унарной), так как любое число в ней образуется путем повторения одного знака, символизирующего единицу. Отголоски единичной системы счисления встречаются и сегодня. Так, чтобы узнать, на каком курсе учится курсант военного училища, нужно сосчитать, какое количество полосок нашито на его рукаве. Сами того не осознавая, единичной системой счисления пользуются малыши, показывая на пальцах свой возраст, а счетные палочки используется для обучения учеников 1-го класса счету.

**Пример 2.6.** Ознакомление с различными системами счисления.



## Системы счисления

- 1 Запустить программу Системы счисления (файл system.exe хранится на CD-ROM в каталоге \soft\information\Systemс\).
- 2 Ввести команду [Системы-Единичная].
- 3 В появившемся диалоговом окне *Единичная система* ознакомиться с содержанием текстовых окон *История системы* и *Сущность системы*, а в окне *Калькулятор* набрать какое-либо число.



Единичная система — не самый удобный способ записи чисел. Записывать таким образом большие количества утомительно, да и сами записи при этом получаются очень длинными. С течением времени возникли иные, более удобные, системы счисления.

**Древнеегипетская десятичная непозиционная система счисления.** Примерно в третьем тысячелетии до нашей эры древние египтяне придумали свою числовую систему, в которой для обозначения ключевых чисел 1, 10, 100 и так далее использовались специальные значки — иероглифы.

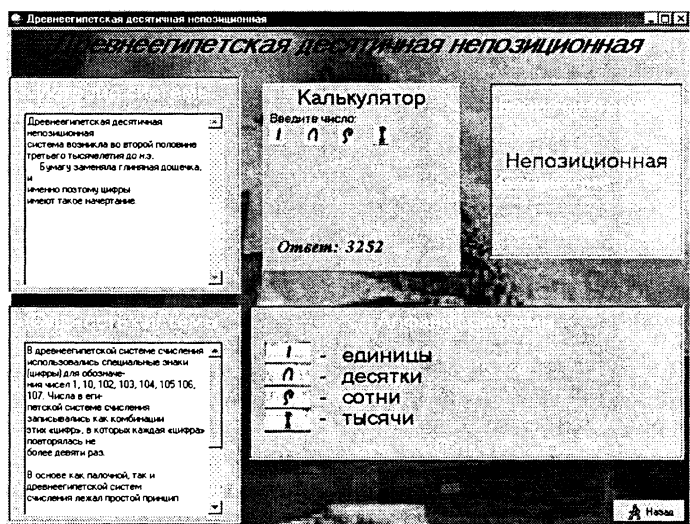
Все остальные числа составлялись из этих ключевых при помощи операции сложения. Система счисления Древнего Египта является десятичной, но непозиционной.



В непозиционных системах счисления количественный эквивалент каждой цифры не зависит от ее положения (места, позиции) в записи числа.

Например, чтобы изобразить 3252, рисовали три цветка лотоса (три тысячи), два свернутых пальмовых листа (две сотни), пять дуг (пять десятков) и два шеста (две единицы). Величина числа не зависела от того, в каком порядке располагались составляющие его знаки: их можно было записывать сверху вниз, справа налево или в произвольном порядке.

- 4 Ввести команду [Системы-Древнеегипетская].
- 5 В появившемся диалоговом окне *Древнеегипетская система* ознакомиться с содержанием текстовых окон *История системы* и *Сущность системы*, а в окне *Калькулятор* набрать число, например, 3252.



**Римская система счисления.** Примером непозиционной системы, которая сохранилась до наших дней, может служить система счисления, которая применялась более двух с половиной тысяч лет назад в Древнем Риме. В основе римской системы счисления лежали знаки I (один палец) для числа 1, V (раскрытая ладонь) для числа 5, X (две сложенные ладони) для 10, а для обозначения чисел 100, 500 и 1000 стали применять первые буквы соответствующих латинских слов (Centum — сто, Demimille — половина тысячи, Mille — тысяча).

Чтобы записать число, римляне разлагали его на сумму тысяч, полутысяч, сотен, полусотен, десятков, пятков, единиц. Например, десятичное число 28 представляется следующим образом:

$$\text{XXVIII} = 10 + 10 + 5 + 1 + 1 + 1$$

(два десятка, пяток, три единицы).

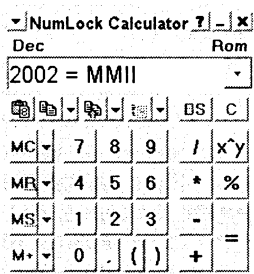
Для записи промежуточных чисел римляне использовали не только сложение, но и вычитание. При этом применялось следующее *правило*: каждый меньший знак, поставленный справа от большего, прибавляется к его значению, а каждый меньший знак, поставленный слева от большего, вычитается из него.

Например, IX — обозначает 9, XI — обозначает 11.

Десятичное число 99 имеет следующее представление:

$$XCIX = -10 + 100 - 1 + 10.$$

- 6 Запустить программу NumLock Calculator.
- 7 Ввести команду [Формат результата-Римский].
- 8 В окне ввода данных ввести число, например, 2002, и нажать клавишу со знаком «=». Появится результат, число, записанное в римской системе счисления.



Римскими цифрами пользовались очень долго. Еще 200 лет назад в деловых бумагах числа должны были обозначаться римскими цифрами (считалось, что обычные арабские цифры легко подделать). Римская система счисления сегодня используется, в основном, для наименования знаменательных дат, томов, разделов и глав в книгах.

**Алфавитные системы счисления.** Более совершенными непозиционными системами счисления были алфавитные системы. К числу таких систем счисления относились греческая, славянская, финикийская и другие. В них числа от 1 до 9, целые количества десятков (от 10 до 90) и целые количества сотен (от 100 до 900) обозначались буквами алфавита.

В алфавитной системе счисления Древней Греции числа 1, 2, ..., 9 обозначались первыми девятью буквами греческого алфавита, например  $\alpha = 1$ ,  $\beta = 2$ ,  $\gamma = 3$  и так далее. Для обозначения чисел 10, 20, ..., 90 применялись следующие 9 букв ( $\iota = 10$ ,  $\kappa = 20$ ,  $\lambda = 30$ ,  $\mu = 40$  и так далее), а для обозначения чисел 100, 200, ..., 900 — последние 9 букв ( $\rho = 100$ ,  $\sigma = 200$ ,  $\tau = 300$  и так далее). Например, число 141 обозначалось  $\rho\mu\alpha$ .

У славянских народов числовые значения букв установились в порядке славянского алфавита, который использовал сначала глаголицу, а затем кириллицу. Подробнее с происхождением и развитием русской письменности можно ознакомиться на сайте «История русской письменности», размещенном на нашем CD-ROM.

**Рис. 2.2**  
Древнерусская  
алфавитная  
система  
счисления

БУКВЫ ИВРИЯНИЦ	ЦИФРОВОЕ ЗНАЧЕНИЕ ИВРИЯНИЦ	БУКВЫ ГЛАГОЛИЦ	ЦИФРОВОЕ ЗНАЧЕНИЕ ГЛАГОЛИЦ	ИВРИЯНИЦНОЕ НАЗВАНИЕ
א	1	א	1	Азь
ב		ב	2	Буки
ב	2	ב	3	Вѣди
ג	3	ג	4	Глаголь
ד	4	ד	5	Добро
ה	5	ה	6	Есть
ז		ז	7	Живѣте
ז	6	ז	8	Зѣло
ח	7	ח	9	Земля
ט	8	ט	10	Иже
י	10	י	20	І

В России славянская нумерация сохранилась до конца XVII века. При Петре I возобладали так называемая арабская нумерация, которой мы пользуемся и сейчас. Славянская нумерация сохранилась только в богослужебных книгах.

Непозиционные системы счисления имеют ряд существенных недостатков:

1. Существует постоянная потребность введения новых знаков для записи больших чисел.
2. Невозможно представлять дробные и отрицательные числа.
3. Сложно выполнять арифметические операции, так как не существует алгоритмов их выполнения.

### 2.2.2. Позиционные системы счисления

Основные достоинства любой позиционной системы счисления — простота выполнения арифметических операций и ограниченное количество символов (цифр), необходимых для записи любых чисел.

Основанием позиционной системы счисления называется возводимое в степень целое число, которое равно количеству цифр, используемых для изображения чисел в данной системе счисления. Основание показывает также, во сколько раз изменяется количественное значение цифры при перемещении ее на соседнюю позицию.

Возможно множество позиционных систем, так как за основание системы счисления можно принять любое число, не меньшее 2. Наименование системы счисления соответствует ее основанию (десятичная, двоичная, восьмеричная, шестнадцатеричная и так далее).



В позиционных системах счисления количественный эквивалент (значение) цифры зависит от ее места (позиции) в записи числа.

Десятичная система характеризуется тем, что в ней 10 единиц какого-либо разряда образуют единицу следующего старшего разряда. Другими словами, единицы различных разрядов представляют собой различные степени числа 10.

В системе счисления с основанием  $q$  ( $q$ -ичная система счисления) единицами разрядов служат последовательные степени числа  $q$ , иначе говоря,  $q$  единиц какого-либо разряда образуют единицу следующего разряда. Для записи чисел в  $q$ -ичной системе счисления требуется  $q$  различных цифр (0, 1, ...,  $q-1$ ).

В позиционной системе счисления число в развернутой форме может быть представлено в следующем виде:

$$A_q = \pm(a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_0q^0 + a_{-1}q^{-1} + a_{-2}q^{-2} + \dots + a_{-m}q^{-m}) \quad (2.4)$$

или

$$A_q = \pm \sum_{i=1}^{n-1} a_i q^i. \quad (2.5)$$

Здесь  $A$  — само число,

$q$  — основание системы счисления;

$a_i$  — цифры, принадлежащие алфавиту данной системы счисления;

$n$  — число целых разрядов числа;

$m$  — число дробных разрядов числа.

Свернутой формой записи числа называется запись в виде

$$A = a_{n-1}a_{n-2}\dots a_1a_0, a_{-1}\dots a_{-m}. \quad (2.6)$$

Именно такой формой записи чисел мы и пользуемся в повседневной жизни. Иначе свернутую форму записи называют естественной или цифровой.

**Пример 2.7.** Десятичное число  $A_{10} = 4718,63$  в развернутой форме запишется так:

$$A_{10} = 4 \cdot 10^3 + 7 \cdot 10^2 + 1 \cdot 10^1 + 8 \cdot 10^0 + 6 \cdot 10^{-1} + 3 \cdot 10^{-2}.$$

**Пример 2.8.** Двоичная система счисления.

В двоичной системе счисления основание  $q = 2$ . В этом случае формула (2.4) принимает вид:

$$A_2 = \pm(a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_02^0 + a_{-1}2^{-1} + a_{-2}2^{-2} + \dots + a_{-m}2^{-m}).$$

Здесь  $a_i$  — возможные цифры (0, 1).



Итак, двоичное число представляет собой цепочку из нулей и единиц. При этом оно имеет достаточно большое число разрядов. Быстрый рост числа разрядов — самый существенный недостаток двоичной системы счисления.

Записав двоичное число  $A_2=1001,1$  в развернутом виде и произведя вычисления, получим это число, выраженное в десятичной системе счисления:

$$A_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 8 + 1 + 0,5 = 9,5_{10}.$$

**Пример 2.9.** Восьмеричная система счисления.

Основание:  $q = 8$ .

Алфавит: 0, 1, 2, 3, 4, 5, 6, 7.

Записав восьмеричное число  $A_8 = 7764,1$  в развернутом виде и произведя вычисления, получим это число, выраженное в десятичной системе счисления:

$$A_8 = 7 \cdot 8^3 + 7 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0 + 1 \cdot 8^{-1} = 3584 + 448 + 48 + 4 + 0,125 = 4084,125_{10}.$$

**Пример 2.10.** Шестнадцатеричная система счисления.

Основание:  $q = 16$ .

Алфавит: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Здесь только десять цифр из шестнадцати имеют общепринятое обозначение 0, 1, ..., 9. Для записи остальных цифр (10, 11, 12, 13, 14 и 15) обычно используются первые шесть букв латинского алфавита.

Таким образом, запись  $3AF_{16}$  означает:

$$3AF_{16} = 3 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0 = 768 + 160 + 15 = 943_{10}.$$

**Пример 2.11.** Запишем начало натурального ряда чисел в десятичной и двоичной системах счисления:

$A_{10}$	$A_2$	$A_{10}$	$A_2$
0	0	8	1000
1	1	9	1001
2	10	10	1010
3	11	11	1011
4	100	12	1100
5	101	13	1101
6	110	14	1110
7	111	15	1111

### Задания для самостоятельного выполнения

**2.12.** Какой числовой эквивалент имеет цифра 6 в десятичных числах:

6789      3650      16      69?

**2.13.** Сравните числа III и 111, записанные в римской и десятичной системах счисления.

2.14. Какие числа записаны римскими цифрами:

а) MCMXCIX; б) CMLXXXVIII; в) MCXLVII?

2.15. Запишите год, месяц и число своего рождения с помощью римских цифр.

2.16. Некоторые римские цифры легко изобразить, используя палочки или спички. Ниже написано несколько неверных равенств. Как можно получить из них верные равенства, если разрешается переложить с одного места на другое только одну спичку (палочку)?

$$VII - V = XI$$

$$IX - V = VI$$

$$VI - IX = III$$

$$VIII - III = X$$

2.17. Заполните следующую таблицу:

Система счисления	Основание	Цифры
шестнадцатеричная	16	
десятичная		0,1,2,3,4,5,6,7,8,9
	8	0,1,2,3,4,5,6,7
	2	

2.18. Заполните следующую таблицу:

Система счисления	Основание	Разряды (степени)				
десятичная	10	10000	1000	100	10	1
восьмеричная	8					
двоичная	2					

2.19. Запишите в развернутом виде числа:

а)  $A_8=143511$ ; г)  $A_{10}=143,511$ ;

б)  $A_2=100111$ ; д)  $A_8=0,143511$ ;

в)  $A_{16}=143511$ ; е)  $A_{16}=1A3,5C1$ .

2.20. Запишите в свернутой форме следующие числа:

а)  $A_{10}=9 \cdot 10^1 + 1 \cdot 10^0 + 5 \cdot 10^{-1} + 3 \cdot 10^{-2}$ ;

б)  $A_{16}=A \cdot 16^1 + 1 \cdot 16^0 + 7 \cdot 16^{-1} + 5 \cdot 16^{-2}$ .

2.21. Правильно ли записаны числа в соответствующих системах счисления:

а)  $A_{10}=A,234$ ; в)  $A_{16}=456,46$ ;

б)  $A_8=-5678$ ; г)  $A_2=22,2?$

2.22. Какое минимальное основание имеет система счисления, если в ней записаны числа 127, 222, 111? Определите десятичный эквивалент данных чисел в найденной системе счисления.

2.23. Чему равен десятичный эквивалент чисел  $10101_2$ ,  $10101_8$ ,  $10101_{16}$ ?

2.24. Трехзначное десятичное число оканчивается цифрой 3. Если эту цифру переместить на два разряда влево, то есть с нее будет начинаться запись нового числа, то это новое число будет на единицу больше утроенного исходного числа. Найдите исходное число.

- 2.25. Шестизначное десятичное число начинается слева цифрой 1. Если эту цифру перенести с первого места слева на последнее место справа, то значение образованного числа будет втрое больше исходного. Найдите исходное число.
- 2.26. Какое из чисел  $110011_2$ ,  $111_4$ ,  $35_8$  и  $1B_{16}$  является:
- а) наибольшим;
  - б) наименьшим?
- 2.27. Существует ли треугольник, длины сторон которого выражаются числами  $12_8$ ,  $11_{16}$  и  $11011_2$ ?
- 2.28. Какое наибольшее десятичное число можно записать тремя цифрами в двоичной, восьмеричной и шестнадцатеричной системах счисления?
- 2.29. «Несерьезные» вопросы.  
Когда  $2 \times 2 = 100$ ?  
Когда  $6 \times 6 = 44$ ?  
Когда  $4 \times 4 = 20$ ?
- 2.30. Выпишите целые десятичные числа, принадлежащие следующим числовым промежуткам:
- а)  $[101101_2; 110000_2]$ ;
  - б)  $[14_8; 20_8]$ ;
  - в)  $[28_{16}; 30_{16}]$ .
- 2.31. В классе  $1111_2$  девочек и  $1100_2$  мальчиков. Сколько учеников в классе?
- 2.32. В классе  $36_q$  учеников, из них  $21_q$  девочек и  $15_q$  мальчиков. В какой системе счисления велся счет учеников?
- 2.33. В саду  $100_q$  фруктовых деревьев, из них  $33_q$  яблони,  $22_q$  груши,  $16_q$  слив и  $5_q$  вишен. В какой системе счисления посчитаны деревья?
- 2.34. Было  $100_q$  яблока. После того как каждое из них разрезали пополам, стало  $1000_q$  половинок. В системе счисления с каким основанием вели счет?
- 2.35. У меня 100 братьев. Младшему 1000 лет, а старшему 1111 лет. Старший учится в 1001 классе. Может ли такое быть?
- 2.36. Некогда был пруд, в центре которого рос один лист водяной лилии. Каждый день число таких листьев удваивалось, и на десятый день вся поверхность пруда уже была заполнена листьями лилий. Сколько дней понадобилось, чтобы заполнить листьями половину пруда? Сколько листьев было после девятого дня?
- 2.37. Путем подбора степеней числа 2, в сумме дающих заданное число, переведите в двоичную систему счисления следующие числа:
- а) 5;
  - б) 7;
  - в) 12;
  - г) 25;
  - д) 32;
  - е) 33.

Проверить правильность перевода с помощью программы Advanced Converter.

## 2.3. Перевод чисел из одной системы счисления в другую

### 2.3.1. Перевод целых чисел из одной системы счисления в другую

Можно сформулировать алгоритм перевода целых чисел из системы с основанием  $p$  в систему с основанием  $q$ :

1. Основание новой системы счисления выразить цифрами исходной системы счисления и все последующие действия производить в исходной системе счисления.
2. Последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления до тех пор, пока не получим частное, меньшее делителя.
3. Полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления.
4. Составить число в новой системе счисления, записывая его, начиная с последнего остатка.

**Пример 2.12.** Перевести десятичное число  $173_{10}$  в восьмеричную систему счисления:

$$\begin{array}{r|l} 173 & 8 \\ \hline 5 & 21 & 8 \\ & 5 & 2 \end{array}$$

Получаем:  $173_{10} = 255_8$ .

**Пример 2.13.** Перевести десятичное число  $173_{10}$  в шестнадцатеричную систему счисления:

$$\begin{array}{r|l} 173 & 16 \\ \hline 13 & 10 \\ (D) & (A) \end{array}$$

Получаем:  $173_{10} = AD_{16}$ .

**Пример 2.14.** Перевести десятичное число  $11_{10}$  в двоичную систему счисления.

$$\begin{array}{r|lll} 11 & 2 & & \\ \hline 1 & 5 & 2 & \\ & 1 & 2 & 2 \\ & & 0 & 1 \end{array}$$

Получаем:  $11_{10} = 1011_2$ .

**Пример 2.15.** Иногда более удобно записать алгоритм перевода в форме таблицы. Переведем десятичное число  $363_{10}$  в двоичное число.

Делимое	363	181	90	45	22	11	5	2	1
Делитель	2	2	2	2	2	2	2	2	2
Остаток	1	1	0	1	0	1	1	0	1

←  
Получаем:  $363_{10} = 101101011_2$ .

### 2.3.2. Перевод дробных чисел из одной системы счисления в другую

Можно сформулировать алгоритм перевода правильной дроби с основанием  $p$  в дробь с основанием  $q$ :

1. Основание новой системы счисления выразить цифрами исходной системы счисления и все последующие действия производить в исходной системе счисления.
2. Последовательно умножать данное число и получаемые дробные части произведений на основание новой системы до тех пор, пока дробная часть произведения не станет равной нулю или будет достигнута требуемая точность представления числа.
3. Полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления.
4. Составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

**Пример 2.16.** Перевести число  $0,65625_{10}$  в восьмеричную систему счисления.

$$\begin{array}{r|l}
 0, & 65625 \\
 \hline
 & \times 8 \\
 \hline
 5 & 25000 \\
 & \times 8 \\
 \hline
 2 & 00000
 \end{array}$$

Получаем:  $0,65625_{10} = 0,52_8$ .

**Пример 2.17.** Перевести число  $0,65625_{10}$  в шестнадцатеричную систему счисления.

$$\begin{array}{r|l}
 0, & 65625 \\
 \hline
 & \times 16 \\
 \hline
 10 & 50000 \\
 (A) & \times 16 \\
 \hline
 8 & 00000
 \end{array}$$

Получаем:  $0,65625_{10} = 0,A8_{16}$ .

**Пример 2.18.** Перевести десятичную дробь  $0,5625_{10}$  в двоичную систему счисления.

0,	5625
	× 2
1	1250
	× 2
0	2500
	× 2
0	5000
	× 2
1	0000

Получаем:  $0,5625_{10} = 0,1001_2$ .

**Пример 2.19.** Перевести в двоичную систему счисления десятичную дробь  $0,7_{10}$ .

0,	7
	× 2
1	4
	× 2
0	8
	× 2
1	6
	× 2
1	2

.....

Очевидно, что этот процесс может продолжаться бесконечно, давая все новые и новые знаки в изображении двоичного эквивалента числа  $0,7_{10}$ . Так, за четыре шага мы получаем число  $0,1011_2$ , а за семь шагов число  $0,1011001_2$ , которое является более точным представлением числа  $0,7_{10}$  в двоичной системе счисления, и так далее. Такой бесконечный процесс обрывают на некотором шаге, когда считают, что получена требуемая точность представления числа.

### 2.3.3. Перевод произвольных чисел

Перевод произвольных чисел, то есть чисел, содержащих целую и дробную части, осуществляется в два этапа. Отдельно переводится целая часть, отдельно — дробная. В итоговой записи полученного числа целая часть отделяется от дробной запятой.

**Пример 2.20.** Перевести число  $17,25_{10}$  в двоичную систему счисления.

Переводим целую часть:

17	2			
1	8	2		
	0	4	2	
		0	2	2
			0	1

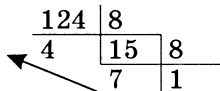
Переводим дробную часть:

0,	25
	× 2
0	50
	× 2
1	00

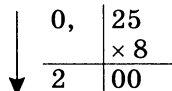
Получаем:  $17,25_{10} = 1001,01_2$ .

**Пример 2.21.** Перевести число  $124,25_{10}$  в восьмеричную систему.

Переводим целую часть:



Переводим дробную часть:



Получаем:  $124,25_{10} = 174,2_8$ .

### 2.3.4. Перевод чисел из системы счисления с основанием 2 в систему счисления с основанием $2^n$ и обратно

**Перевод целых чисел.** Если основание  $q$ -ичной системы счисления является степенью числа 2, то перевод чисел из  $q$ -ичной системы счисления в двоичную и обратно можно проводить по более простым правилам. Для того чтобы целое двоичное число записать в системе счисления с основанием  $q = 2^n$ , нужно:

1. Двоичное число разбить справа налево на группы по  $n$  цифр в каждой.
2. Если в последней левой группе окажется меньше  $n$  разрядов, то ее надо дополнить слева нулями до нужного числа разрядов.
3. Рассмотреть каждую группу как  $n$ -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием  $q = 2^n$ .

**Пример 2.22.** Число  $101100001000110010_2$  переведем в восьмеричную систему счисления.

Разбиваем число справа налево на триады и под каждой из них записываем соответствующую восьмеричную цифру:

101	100	001	000	110	010
5	4	1	0	6	2

Получаем восьмеричное представление исходного числа:  $541062_8$ .

**Пример 2.23.** Число  $100000000111110000111_2$  переведем в шестнадцатеричную систему счисления.

Разбиваем число справа налево на тетрады и под каждой из них записываем соответствующую шестнадцатеричную цифру:

0010	0000	0000	1111	1000	0111
2	0	0	F	8	7

Получаем шестнадцатеричное представление исходного числа:  $200F87_{16}$ .

**Перевод дробных чисел.** Для того, чтобы дробное двоичное число записать в системе счисления с основанием  $q = 2^n$ , нужно:

1. Двоичное число разбить слева направо на группы по  $n$  цифр в каждой.
2. Если в последней правой группе окажется меньше  $n$  разрядов, то ее надо дополнить справа нулями до нужного числа разрядов.
3. Рассмотреть каждую группу как  $n$ -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием  $q = 2^n$ .

**Пример 2.24.** Число  $0,10110001_2$  переведем в восьмеричную систему счисления.

Разбиваем число слева направо на триады и под каждой из них записываем соответствующую восьмеричную цифру:

0,	101	100	010
0,	5	4	2

Получаем восьмеричное представление исходного числа:  $0,542_8$ .

**Пример 2.25.** Число  $0,10000000011_2$  переведем в шестнадцатеричную систему счисления. Разбиваем число слева направо на тетрады и под каждой из них записываем соответствующую шестнадцатеричную цифру:

0,	1000	0000	0011
0,	8	0	3

Получаем шестнадцатеричное представление исходного числа:  $0,803_{16}$ .

**Перевод произвольных чисел.** Для того чтобы произвольное двоичное число записать в системе счисления с основанием  $q = 2^n$ , нужно:

1. Целую часть данного двоичного числа разбить справа налево, а дробную — слева направо на группы по  $n$  цифр в каждой.
2. Если в последних левой и/или правой группах окажется меньше  $n$  разрядов, то их надо дополнить слева и/или справа нулями до нужного числа разрядов.
3. Рассмотреть каждую группу как  $n$ -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием  $q = 2^n$ .



**Пример 2.26.** Число  $111100101,0111_2$  переведем в восьмеричную систему счисления.

Разбиваем целую и дробную части числа на триады и под каждой из них записываем соответствующую восьмеричную цифру:

111	100	101,	011	100
7	4	5,	3	4

Получаем восьмеричное представление исходного числа:  $745,34_8$ .

**Пример 2.27.** Число  $11101001000,11010010_2$  переведем в шестнадцатеричную систему счисления.

Разбиваем целую и дробную части числа на тетрады и под каждой из них записываем соответствующую шестнадцатеричную цифру:

0111	0100	1000,	1101	0010
7	4	8,	D	2

Получаем шестнадцатеричное представление исходного числа:  $748,D2_{16}$ .

**Перевод чисел из систем счисления с основанием  $q = 2^n$  в двоичную систему.** Для того, чтобы произвольное число, записанное в системе счисления с основанием  $q = 2^n$ , перевести в двоичную систему счисления, нужно каждую цифру этого числа заменить ее  $n$ -значным эквивалентом в двоичной системе счисления.

**Пример 2.28.** Переведем шестнадцатеричное число  $4AC35_{16}$  в двоичную систему счисления.

В соответствии с алгоритмом:

4	A	C	3	5
0100	1010	1100	0011	0101

Получаем:  $1001010110000110101_2$ .

### Задания для самостоятельного выполнения

2.38. Заполните таблицу, в каждой строке которой одно и то же целое число должно быть записано в различных системах счисления.

Двоичная	Восьмеричная	Десятичная	Шестнадцатеричная
101010			
	127		
		269	
			9B

2.39. Заполните таблицу, в каждой строке которой одно и то же дробное число должно быть записано в различных системах счисления.

Двоичная	Восьмеричная	Десятичная	Шестнадцатеричная
0,101			
	0,6		
		0,125	
			0,4

2.40. Заполните таблицу, в каждой строке которой одно и то же произвольное число (число может содержать как целую, так и дробную часть) должно быть записано в различных системах счисления.

Двоичная	Восьмеричная	Десятичная	Шестнадцатеричная
111101,1			
	233,5		
		46,5625	
			59,B

## 2.4. Арифметические операции в позиционных системах счисления

**Арифметические операции в двоичной системе счисления.** Рассмотрим более подробно арифметические операции в двоичной системе счисления. Арифметика двоичной системы счисления основывается на использовании таблиц сложения, вычитания и умножения цифр. Арифметические операнды располагаются в верхней строке и в первом столбце таблиц, а результаты на пересечении столбцов и строк:

+	0	1
0	0	1
1	1	10

-	0	1
0	0	$\bar{1}1$
1	1	0

×	0	1
0	0	0
1	0	1

Рассмотрим подробно каждую операцию.

**Сложение.** Таблица двоичного сложения предельно проста. Только в одном случае, когда производится сложение  $1 + 1$ , происходит перенос в старший разряд.

**Пример 2.29.** Рассмотрим несколько примеров сложения двоичных чисел:

$$\begin{array}{r} +1001 \\ +1010 \\ \hline 10011 \end{array}$$

$$\begin{array}{r} +1101 \\ +1011 \\ \hline 11000 \end{array}$$

$$\begin{array}{r} +11111 \\ +1 \\ \hline 100000 \end{array}$$

$$\begin{array}{r} +1010011,111 \\ +11001,110 \\ \hline 1101101,101 \end{array}$$

**Вычитание.** При выполнении операции вычитания всегда из большего по абсолютной величине числа вычитается меньшее и ставится соответствующий знак. В таблице вычитания 1 с чертой означает заем в старшем разряде.

**Пример 2.30.** Рассмотрим несколько примеров вычитания двоичных чисел:

$$10111001,1 - 10001101,1 = 101100,0$$

$$101011111 - 110101101 = -1001110$$

$$\begin{array}{r} \underline{10111001,1} \\ - 10001101,1 \\ \hline 00101100,0 \end{array}$$

$$\begin{array}{r} \underline{110110101} \\ - 101011111 \\ \hline 001010110 \end{array}$$

**Умножение.** Операция умножения выполняется с использованием таблицы умножения по обычной схеме, применяемой в десятичной системе счисления с последовательным умножением множимого на очередную цифру множителя.

**Пример 2.31.** Рассмотрим несколько примеров умножения двоичных чисел:

$$11001 \times 1101 = 101000101$$

$$11001,01 \times 11,01 = 1010010,0001$$

$$\begin{array}{r} 11001 \\ \times 1101 \\ \hline \end{array}$$

$$11001$$

$$11001$$

$$11001$$

$$\hline 101000101$$

$$\begin{array}{r} 11001,01 \\ \times 11,01 \\ \hline \end{array}$$

$$1100101$$

$$1100101$$

$$1100101$$

$$\hline 1010010,0001$$

Вы видите, что умножение сводится к сдвигам множимого и сложениям.

**Деление.** Операция деления выполняется по алгоритму, подобному алгоритму выполнения операции деления в десятичной системе счисления.

**Пример 2.32.** Рассмотрим пример деления двоичных чисел:

$$101000101:1101=11001$$

$$\begin{array}{r} 101000101 \quad | \quad 1101 \\ - \underline{1101} \quad | \quad 11001 \\ \hline \end{array}$$

$$1110$$

$$- \underline{1101}$$

$$1101$$

$$- \underline{1101}$$

$$0$$

**Сложение в других системах счисления.** Ниже приведена таблица сложения в восьмеричной системе счисления:

+	1	2	3	4	5	6	7
1	2	3	4	5	6	7	10
2	3	4	5	6	7	10	11
3	4	5	6	7	10	11	12
4	5	6	7	10	11	12	13
5	6	7	10	11	12	13	14
6	7	10	11	12	13	14	15
7	10	11	12	13	14	15	16

### Задания для самостоятельного выполнения

**2.41.** Выполните арифметические операции:

- а)  $1110_2 + 1001_2$ ; г)  $1110_2 - 1001_2$ ; ж)  $1110_2 \cdot 1001_2$ ; к)  $1010_2 : 10_2$ ;  
 б)  $67_8 + 23_8$ ; д)  $67_8 - 23_8$ ; з)  $67_8 \cdot 23_8$ ; л)  $74_8 : 24_8$ ;  
 в)  $AF_{16} + 97_{16}$ ; е)  $AF_{16} - 97_{16}$ ; и)  $AF_{16} \cdot 97_{16}$ ; м)  $5A_{16} : 1E_{16}$ .

**2.42.** Расставьте знаки арифметических операций так, чтобы были верны следующие равенства в двоичной системе:

- а)  $1100 ? 11 ? 100 = 100000$ ;  
 б)  $1100 ? 10 ? 10 = 100$ ;  
 в)  $1100 ? 10 ? 10 = 110000$ ;  
 г)  $1100 ? 10 ? 10 = 1011$ ;  
 д)  $1100 ? 11 ? 100 = 0$ .

**2.43.** Какое число следует за каждым из данных:

- а)  $10_{10}$ ; в)  $AF_{16}$ ;  
 б)  $677_8$ ; г)  $101_2$ .

Ответ для каждого числа запишите в указанной и десятичной системах счисления.

**2.44.** Какое число предшествует каждому из данных:

- а)  $10_{10}$ ; в)  $9A_{16}$ ;  
 б)  $56_8$ ; г)  $110_2$ ?

**2.45.** Выпишите целые числа, принадлежащие следующим числовым промежуткам:

- а)  $[101101_2; 110000_2]$  в двоичной системе;  
 б)  $[14_8; 20_8]$  в восьмеричной системе;  
 в)  $[28_{16}; 30_{16}]$  в шестнадцатеричной системе.

Ответ для каждого числа запишите в указанной и десятичной системах счисления.

2.46. Вычислите выражения:

а)  $(1111101_2 + AF_{16})/36_8$ ;

б)  $125_8 + 11101_2 \times A2_{16} - 1417_8$ .

2.47. Найдите среднее арифметическое следующих чисел:

а)  $10010110_2$ ,  $1100100_2$  и  $110010_2$ ;

б)  $226_8$ ,  $64_{16}$  и  $62_8$ .

2.48. Сумму восьмеричных чисел  $17_8 + 1700_8 + 170000_8 + 17000000_8 + 1700000000_8$  перевели в шестнадцатеричную систему счисления. Найдите в записи числа, равного этой сумме, пятую цифру слева.

2.49. Восстановите неизвестные цифры, обозначенные знаком вопроса, в следующих примерах на сложение и вычитание, определив вначале, в какой системе изображены числа.

$$\begin{array}{r} \text{а) } \quad 5?55 \\ + ?327 \\ \hline ?16?4 \end{array}$$

$$\begin{array}{r} \text{б) } \quad 1536 \\ - ?42 \\ \hline 67? \end{array}$$

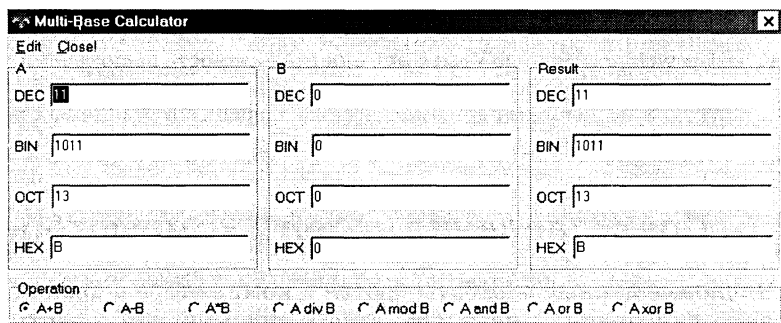
## 2.5. Вычисления в позиционных системах счисления с использованием калькулятора

Для перевода чисел между десятичной, двоичной, восьмеричной и шестнадцатеричной системами счисления воспользуемся *Wise Calculator*. В режиме *Multi-Base Calculator* появляется многооконная панель, позволяющая ввести число в любой системе счисления и автоматически получить значения этого числа в других системах счисления.

**Пример 2.33.** Перевод чисел из одной системы в другую с помощью программы *Wise Calculator*.

### Перевод чисел из одной системы счисления в другую

- 1 Запустить *Wise Calculator*.
- 2 Ввести команду [*Tools-Multi-Base Calculator...*].
- 3 На появившейся многооконной панели *Multi-Base Calculator* ввести число в выбранной системе счисления в соответствующее этой системе окно. Например, число 11 в окно *DEC*.
- 4 В окнах *BIN*, *OCT* и *HEX* появятся значения числа в двоичной ( $1011_2$ ), восьмеричной ( $13_8$ ) и шестнадцатеричной ( $B_{16}$ ) системах счисления.

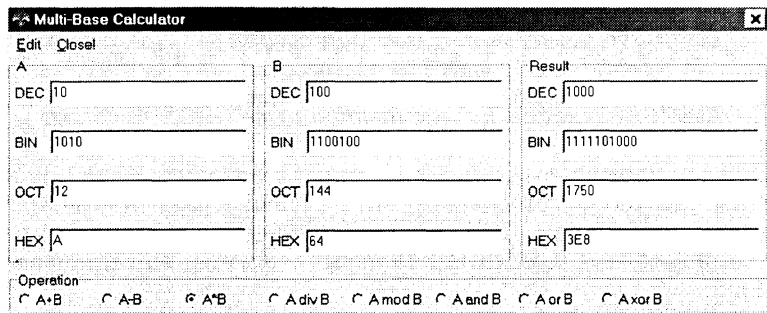


В режиме *Multi-Base Calculator* возможно проведение арифметических операций (сложение, вычитание и умножение) в различных системах счисления. Для этого в окна операндов *A* и *B* необходимо ввести числа (можно в различных системах счисления) и выбрать в группе переключателей *Operation* арифметическую операцию.

**Пример 2.34.** Выполнение арифметических операций с помощью программы *Wise Calculator*.

#### Арифметические операции в различных системах счисления

- 1 Запустить *Wise Calculator* и ввести команду [Tools-Multi-Base Calculator...].
- 2 На появившейся многооконной панели *Multi-Base Calculator* ввести пару чисел в окна операндов *A* и *B*. Например, восьмеричное число  $12_8$  в окно *OCT* (операнд *A*) и шестнадцатеричное число  $64_{16}$  в окно *HEX* (операнд *B*).
- 3 Выбрать в группе переключателей *Operation*, например, операцию умножения  $A*B$ .
- 4 В окнах *Result* появится результат выполнения арифметической операции одновременно в четырех системах счисления: *DEC* ( $1000_{10}$ ), *BIN* ( $1111101000_2$ ), *OCT* ( $1750_8$ ) и *HEX* ( $3E8_{16}$ ).



## Задания для самостоятельного выполнения

- 2.50. Проверить с помощью Wise Calculator правильность выполнения заданий этой главы по переводу чисел из одной системы счисления в другую и выполнения арифметических операций в различных системах счисления.

## 2.6. Представление чисел в компьютере

Числовые данные обрабатываются в компьютере в двоичной системе счисления. Числа хранятся в оперативной памяти в виде последовательностей нулей и единиц, то есть в двоичном коде.

**Представление чисел в формате с фиксированной запятой.** Целые числа в компьютере хранятся в памяти в формате с *фиксированной запятой*. В этом случае каждому разряду ячейки памяти соответствует всегда один и тот же разряд числа, а запятая находится справа после младшего разряда, то есть вне разрядной сетки.

Для хранения *целых неотрицательных чисел* отводится одна ячейка памяти (8 бит). Например, число  $A_2 = 10101010_2$  будет храниться в ячейке памяти следующим образом:

1	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---

Максимальное значение целого неотрицательного числа достигается в случае, когда во всех ячейках хранятся единицы. Для  $n$ -разрядного представления оно будет равно:

$$2^n - 1.$$

**Пример 2.35.** Определить диапазон чисел, которые могут храниться в оперативной памяти в формате *целое неотрицательное число*.

Минимальное число соответствует восьми нулям, хранящимся в восьми ячейках памяти, и равно нулю.

Максимальное число соответствует восьми единицам, хранящимся в ячейках памяти и равно:

$$\begin{aligned} A &= 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = \\ &= 1 \cdot 2^8 - 1 = 255_{10}. \end{aligned}$$

Диапазон изменения *целых неотрицательных чисел* от 0 до 255.

Для хранения *целых чисел со знаком* отводится две ячейки памяти (16 бит), причем старший (левый) разряд отводится под знак числа (если число положительное, то в знаковый разряд записывается 0, если число отрицательное, то записывается 1).

Представление в компьютере положительных чисел с использованием формата «знак–величина» называется *прямым кодом числа*. Например, число  $2002_{10} = 11111010010_2$  будет представлено в 16-разрядном представлении следующим образом:

0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

При представлении целых чисел в  $n$ -разрядном представлении со знаком максимальное положительное число (с учетом выделения одного разряда на знак) равно:

$$A = 2^{n-1} - 1.$$

**Пример 2.36.** Определить максимальное положительное число, которое может храниться в оперативной памяти в формате *целое число со знаком*.

$$A_{10} = 2^{15} - 1 = 32767_{10}.$$

Для представления отрицательных чисел используется *дополнительный код*. Дополнительный код позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно упрощает работу процессора и увеличивает его быстродействие.




---



---

Дополнительный код отрицательного числа  $A$ , хранящегося в  $n$  ячейках, равен  $2^n - |A|$ .

---



---

Дополнительный код представляет собой дополнение модуля отрицательного числа  $A$  до 0, поэтому в  $n$ -разрядной компьютерной арифметике:

$$2^n - |A| + |A| = 0.$$

Это равенство тождественно справедливо, так как в компьютерной  $n$ -разрядной арифметике  $2^n \equiv 0$ . Действительно, двоичная запись такого числа состоит из одной единицы и  $n$  нулей, а в  $n$ -разрядную ячейку может уместиться только  $n$  младших разрядов, то есть  $n$  нулей.

**Пример 2.37.** Записать дополнительный код отрицательного числа  $-2002$  для 16-разрядного компьютерного представления.

Проведем вычисления в соответствии с определением дополнительного кода:

$2^{16}$	=	$1000000000000000_2$	$65536_{10}$
$2002_{10}$	=	$0000011111010010_2$	$2002_{10}$
$2^{16} -  2002_{10} $	=	$1111100000101110_2$	$63534_{10}$



Проведем проверку с использованием десятичной системы счисления. Дополнительный код  $63534_{10}$  в сумме с модулем отрицательного числа  $2002_{10}$  равен  $65536_{10}$ , то есть дополнительный код дополняет модуль отрицательного числа до  $2^{16}$  (до нуля 16-разрядной компьютерной арифметики).

Для получения дополнительного кода отрицательного числа можно использовать довольно простой алгоритм:

1. Модуль числа записать *прямым кодом* в  $n$  двоичных разрядах;
2. Получить *обратный код* числа, для этого значения всех бит инвертировать (все единицы заменить на нули и все нули заменить на единицы);
3. К полученному *обратному коду* прибавить единицу.

**Пример 2.38.** Записать дополнительный код отрицательного числа  $-2002$  для 16-разрядного компьютерного представления с использованием алгоритма.

Прямой код	$ -2002 _{10}$	$0000011111010010_2$
Обратный код	Инвертирование	$1111100000101101_2$
	Прибавление единицы	$1111100000101101_2$ + $0000000000000001_2$
Дополнительный код		$1111100000101110_2$

При  $n$ -разрядном представлении отрицательного числа  $A$  дополнительным кодом старший разряд выделяется для хранения знака числа (единицы). В остальных разрядах записывается положительное число:

$$2^{n-1} - |A|.$$

Чтобы число было положительным, должно выполняться условие:

$$|A| = 2^{n-1}.$$

Следовательно, максимальное значение модуля числа  $A$  в  $n$ -разрядном представлении равно:

$$|A| \leq 2^{n-1}.$$

Тогда минимальное отрицательное число равно:

$$A = -2^{n-1}.$$

**Пример 2.39.** Определить диапазон чисел, которые могут храниться в оперативной памяти в формате *больших целых чисел со знаком* (для хранения таких чисел отводится четыре ячейки памяти — 32 бита).

Максимальное положительное целое число (с учетом выделения одного разряда на знак) равно:

$$A = 2^{31} - 1 = 2\,147\,483\,647_{10}.$$

Минимальное отрицательное целое число равно:

$$A = -2^{31} = -2\ 147\ 483\ 648_{10}.$$

Достоинствами представления чисел в формате с *фиксированной запятой* являются простота и наглядность представления чисел, а также простота алгоритмов реализации арифметических операций (вычитание благодаря использованию дополнительного кода для представления отрицательных чисел сводится к сложению).

**Пример 2.40.** Выполнить арифметическое действие  $3000_{10} - 5000_{10}$  в 16-разрядном компьютерном представлении.

Представим положительное число в прямом, а отрицательное число в дополнительном коде:

Десятичное число	Прямой код	Обратный код	Дополнительный код
3000	0000101110111000		
-5000	0001001110001000	1110110001110111	1110110001110111 + 0000000000000001 1110110001111000

Сложим прямой код положительного числа с дополнительным кодом отрицательного числа. Получим результат в дополнительном коде:

3000-5000			1111100000110000
-----------	--	--	------------------

Переведем полученный дополнительный код в десятичное число:

- 1) инвертируем дополнительный код: 0000011111001111;
- 2) прибавим к полученному коду 1 и получим модуль отрицательного числа:

$$\begin{array}{r} 0000011111001111 \\ + 0000000000000001 \\ \hline 0000011111010000 \end{array}$$

- 3) переведем в десятичное число и припишем знак отрицательного числа:  $-2000$ .

Недостатком представления чисел в формате с *фиксированной запятой* является конечный диапазон представления величин, недостаточный для решения математических, физических, экономических и других задач, в которых используются как очень малые, так и очень большие числа.

**Представление чисел в формате с плавающей запятой.** Вещественные числа (конечные и бесконечные десятичные дроби) хранятся и обрабатываются в компьютере в формате *с плавающей запятой*. В этом случае положение запятой в записи числа может изменяться.

Формат чисел *с плавающей запятой* базируется на экспоненциальной форме записи, в которой может быть представлено любое число. Так, число  $A$  может быть представлено в виде:

$$A = m \times q^n, \quad (2.7)$$

где  $m$  — мантисса числа;

$q$  — основание системы счисления;

$n$  — порядок числа.

Для однозначности представления чисел *с плавающей запятой* используется нормализованная форма, при которой мантисса отвечает условию:

$$1/n \leq |m| < 1.$$

Это означает, что мантисса должна быть правильной дробью и иметь после запятой цифру, отличную от нуля.

**Пример 2.41.** Преобразуйте десятичное число 888,888, записанное в естественной форме, в экспоненциальную форму с нормализованной мантиссой.

$$888,888 = 0,888888 \cdot 10^3$$

Нормализованная мантисса  $m = 0,888888$ , порядок  $n = 3$ .

Число в форме с плавающей запятой занимает в памяти компьютера четыре (*число обычной точности*) или восемь байтов (*число двойной точности*). При записи числа с плавающей запятой выделяются разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.

Диапазон изменения чисел определяется количеством разрядов, отведенных для хранения порядка числа, а точность (количество значащих цифр) определяется количеством разрядов, отведенных для хранения мантиссы.

**Пример 2.42.** Определить максимальное число и его точность для формата чисел *обычной точности*, если для хранения порядка и его знака отводится 8 разрядов, а для хранения мантиссы и ее знака 24 разряда.

0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
знак и порядок									знак и мантисса																							

$1111111_2 = 127_{10}$ , и, следовательно, максимальное значение числа составит:

$$2^{127} = 1,7014118346046923173168730371588 \cdot 10^{38}$$

Точность вычислений определяется количеством разрядов, отведенных для хранения мантиисы чисел. Максимальное значение положительной мантиисы равно:

$$2^{23} - 1 \approx 2^{23} = 2^{(10 \cdot 2,3)} \approx 1000^{2,3} = 10^{(3 \cdot 2,3)} \approx 10^7.$$

Таким образом, максимальное значение *чисел обычной точности* с учетом возможной точности вычислений составит  $1,701411 \cdot 10^{38}$  (количество значащих цифр десятичного числа в данном случае ограничено 7 разрядами).

При сложении и вычитании чисел в формате *с плавающей запятой* сначала производится подготовительная операция *выравнивания порядков*. Порядок меньшего (по модулю) числа увеличивается до величины порядка большего (по модулю) числа. Для того чтобы величина числа не изменилась, мантииса уменьшается в такое же количество раз (сдвигается в ячейке памяти вправо на количество разрядов, равное разности порядков чисел).

После выполнения операции выравнивания одинаковые разряды чисел оказываются расположенными в одних и тех же разрядах ячеек памяти. Теперь операции сложения и вычитания чисел сводятся к сложению или вычитанию мантиис.

После выполнения арифметической операции для приведения полученного числа к стандартному формату с плавающей запятой производится нормализация, то есть мантииса сдвигается влево или вправо так, чтобы ее первая значащая цифра попала в первый разряд после запятой.

**Пример 2.43.** Произвести сложение чисел  $0,1 \cdot 2^3$  и  $0,1 \cdot 2^5$  в формате с плавающей запятой.

Произведем выравнивание порядков и сложение мантиис:

$$\begin{array}{r} 0,001 \cdot 2^5 \\ + 0,100 \cdot 2^5 \\ \hline 0,101 \cdot 2^5 \end{array}$$

При умножении чисел в формате *с плавающей запятой* порядки складываются, а мантиисы перемножаются. При делении из порядка делимого вычитается порядок делителя, а мантииса делимого делится на мантиису делителя.

**Пример 2.44.** Произвести умножение чисел  $0,1 \cdot 2^3$  и  $0,1 \cdot 2^5$  в формате с плавающей запятой.

После умножения будет получено число  $0,01 \cdot 2^8$ , которое после нормализации примет вид  $0,1 \cdot 2^7$ .

## Задания для самостоятельного выполнения

**2.51.** Заполнить таблицу, записав отрицательные десятичные числа в прямом, обратном и дополнительном кодах в 16-разрядном представлении:

Десятичные числа	Прямой код	Обратный код	Дополнительный код
-10			
-100			
-1000			
-10000			

**2.52.** Заполнить таблицу, записав десятичные числа в заданном компьютерном представлении:

Десятичные числа	Компьютерное представление	
	целые неотрицательные числа	целые числа со знаком
255		
-255		
32768		
-32768		

**2.53.** Заполнить таблицу, записав максимальные и минимальные значения чисел в заданном компьютерном представлении:

Компьютерное представление	Максимальное значение	Минимальное значение
целые неотрицательные числа		
целые числа со знаком		
большое целое число со знаком		

**2.54.** Выполнить арифметическое действие  $20_{10} - 60_{10}$  в 16-разрядном компьютерном представлении.

**2.55.** Записать следующие числа в форме с плавающей запятой и нормализованной мантиссой:

а)  $217,934_{10}$ ; б)  $75321_{10}$ ; в)  $10,0101_{10}$ ; г)  $200450_{10}$ .

**2.56.** Определить максимальное число и его точность для формата чисел двойной точности, если для хранения порядка и его знака отводится 11 разрядов, а для хранения мантиссы и ее знака 53 разряда.

**2.57.** Произвести сложение, вычитание, умножение и деление чисел  $0,1 \cdot 2^2$  и  $0,1 \cdot 2^{-2}$  в формате с плавающей запятой.

## 2.7. Кодирование текстовой информации

### 2.10. Двоичное кодирование текстовой информации

Учебник 

В традиционных кодировках для кодирования одного символа используется 8 бит. Легко подсчитать по формуле (2.3), что такой 8-разрядный код позволяет закодировать 256 различных символов.

Присвоение символу определенного числового кода — это вопрос соглашения. В качестве международного стандарта принята кодовая таблица ASCII (American Standard Code for Information Interchange), кодирующая первую половину символов с числовыми кодами от 0 до 127 (коды от 0 до 32 отведены не символам, а функциональным клавишам).

sp 32	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0 48	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@ 64	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P 80	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
` 96	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p 112	q	r	s	t	u	v	w	x	y	z	{		}	~	

Рис. 2.3. Международная кодировка ASCII

Национальные стандарты кодировочных таблиц включают международную часть кодовой таблицы без изменений, а во второй половине содержат коды национальных алфавитов, символы псевдографики и некоторые математические знаки. К сожалению, в настоящее время существуют пять различных кодировок кириллицы (КОИ8-Р, Windows, MSDOS, Macintosh и ISO), что вызывает дополнительные трудности при работе с русскоязычными документами.

Хронологически одним из первых стандартов кодирования русских букв на компьютерах был КОИ8 («Код обмена информацией, 8-битный»). Эта кодировка применялась еще в 70-ые годы прошлого века на компьютерах серии ЕС ЭВМ, а с середины 80-х стала использоваться в первых русифицированных версиях операционной системы UNIX.

—		Г	Г	Л	Л	Т	Т	Т	Т	Т	■	■	■	■	■
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
▨	▨	▨	Г	■	●	√	≈	≤	≥	nbsp	Ј	•	²	•	÷
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
=		F	ё	П	Г	Г	П	П	Л	Л	Л	Л	Л	Л	Л
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
		Г	ё			Т	П	Т	Л	Л	Л	Л	Л	Л	©
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
п	я	р	с	т	у	ж	в	ь	ы	э	ш	э	щ	ч	ъ
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
П	Я	Р	С	Т	У	Ж	В	Ь	Ы	Э	Ш	Э	Щ	Ч	Ъ
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Рис. 2.4. Кодировка КОИ8-Р

Наиболее распространенной в настоящее время является кодировка Microsoft Windows, обозначаемая сокращением CP1251 («CP» означает «Code Page», «кодовая страница»).

Á	à	,	è	„	...	†	‡	€	%	É	<	й	Й	ó	ú
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
á	‘	’	“	”	•	—	—	€	™	é	>	ò	й	ó	ú
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
nbsp	ÿ	Ы	Э	х	ы		§	€	©	Ю	«	¬	shy	®	Я
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
•	±	Ы	Э	’	µ	¶	•	ё	№	»	э	ю	я	я	я
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Рис. 2.5. Кодировка CP1251

От начала 90-х годов, времени господства операционной системы MS DOS, остается кодировка CP866. Компьютеры фирмы Apple, работающие под управлением операционной системы Mac OS, используют свою собственную кодировку Mac. Кроме того, Международная организация по стандартизации (International Standards Organization, ISO) утвердила в качестве стандарта для русского языка еще одну кодировку под названием ISO 8859-5.

В конце 90-х годов появился новый международный стандарт Unicode, который отводит под один символ не один байт, а два, и поэтому с его помощью можно закодировать не 256, а 65536 различных символов. Полная спецификация стандарта Unicode включает в себя все существующие, вымершие и искусственно созданные алфавиты мира, а также множество математических, музыкальных, химических и прочих символов.

**Пример 2.45.** Представьте в форме шестнадцатеричного кода слово «ЭВМ» во всех пяти кодировках. Воспользуйтесь CD-ROM для получения кодировочных таблиц CP866, Mac и ISO и компьютерным калькулятором для перевода чисел из десятичной в шестнадцатеричную систему счисления.

Последовательности десятичных кодов слова «ЭВМ» в различных кодировках составляем на основе кодировочных таблиц:

КОИ8-Р:	252	247	237
CP1251:	221	194	204
CP866:	157	130	140
Mac:	157	130	140
ISO:	205	178	188

Переводим с помощью калькулятора последовательности кодов из десятичной системы в шестнадцатеричную:

КОИ8-Р:	FC	F7	ED
CP1251:	DD	C2	CC
CP866:	9D	82	8C
Mac:	9D	82	8C
ISO:	CD	B2	BC

Для преобразования русскоязычных текстовых документов из одной кодировки в другую используются специальные программы-конверторы. Одной из таких программ является текстовый редактор Hieroglyph, который позволяет осуществлять перевод набранного текста из одной кодировки в другую и даже использовать различные кодировки в одном тексте.

**Пример 2.46.** Представить в пяти различных кодировках слово «Кодировка».

Выполним это задание с использованием текстового редактора Hieroglyph.

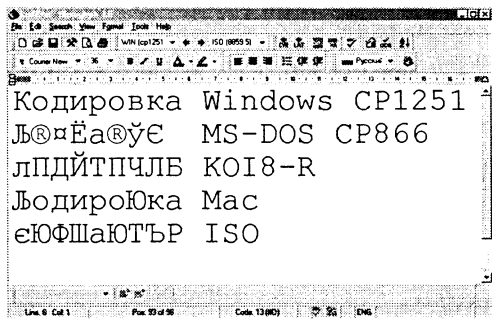


### Перекодирование текста

- 1 Запустить текстовый редактор Hieroglyph.
- 2 В раскрывающемся списке исходных кодировок выбрать кодировку WIN(cp1251) и ввести текст: «Кодировка Windows CP1251».



- 3 Скопировать текст четыре раза и, выделяя строки, последовательно выбрать в раскрывающемся списке конечные кодировки (DOS, KOI8-R, Mac и ISO), каждый раз нажимая кнопку перекодирования. Для каждой кодировки отредактировать ее название.
- 4 В результате текст будет состоять из пяти строк, записанных в различных кодировках.



### Задания для самостоятельного выполнения

- 2.58. Закодируйте с помощью кодировочной таблицы ASCII и представьте в шестнадцатеричной системе счисления следующие тексты:  
 а) Password; б) Windows; в) Norton Commander.
- 2.59. Декодируйте с помощью кодировочной таблицы ASCII следующие тексты, заданные шестнадцатеричным кодом:  
 а) 54 6F 72 6E 61 64 6F;  
 б) 49 20 6C 6F 76 65 20 79 6F 75;  
 в) 32 2A 78 2B 79 3D 30.
- 2.60. Перейдите от двоичного кода к десятичному и декодируйте следующие тексты:  
 а) 01010101 01110000 00100000 00100110 00100000 01000100  
 01101111 01110111 01101110;  
 б) 01001001 01000010 01001101;  
 в) 01000101 01101110 01110100 01100101 01110010.
- 2.61. Декодируйте следующие тексты, заданные десятичным кодом:  
 а) 087 111 114 100;  
 б) 068 079 083;  
 в) 080 097 105 110 116 098 114 117 115 104.
- 2.62. Представьте в форме шестнадцатеричного кода слово «БИС» во всех пяти кодировках. Воспользуйтесь CD-ROM для получения кодировочных таблиц.
- 2.63. Как будет выглядеть слово «диск», записанное в кодировке CP1251, в других кодировках.
- 2.64. В текстовом режиме экран обычно разбивается на 25 строк по 80 символов в строке. Определите объем текстовой информации, занимающей весь экран монитора.

- 2.65. Во сколько раз уменьшится информационный объем страницы текста при его преобразовании из кодировки Unicode (таблица кодировки содержит 65536 символов) в кодировку Windows CP1251 (таблица кодировки содержит 256 символов)?
- 2.66. Каков информационный объем текста, содержащего слово ИНФОРМАТИКА, в 8-битной кодировке? в 16-битной кодировке?

## 2.8. Кодирование графической информации

### 2.12. Двоичное кодирование графической информации

Графические изображения, хранящиеся в аналоговой (непрерывной) форме на бумаге, фото- и киноплёнке, могут быть преобразованы в цифровой компьютерный формат путем пространственной дискретизации. Это реализуется путем сканирования, результатом которого является растровое изображение. Растровое изображение состоит из отдельных точек (пикселей — англ. pixel образовано от словосочетания picture element, что означает элемент изображения), каждая из которых может иметь свой цвет.

Качество растрового изображения определяется его разрешением (количеством точек по вертикали и по горизонтали) и используемой палитрой цветов (16, 256, 65536 цветов и более). Из формулы (2.2) можно определить, какое количество бит информации необходимо выделить для хранения цвета точки (глубину цвета) для каждой палитры цветов.

**Пример 2.47.** Определить глубину цвета в графическом режиме *True Color*, в котором палитра состоит из более чем 4 миллиардов (4 294 967 296) цветов.

$$I = \log_2 4\,294\,967\,296 = 32 \text{ бита.}$$

В современных компьютерах используются различные графические режимы экрана монитора, каждый из которых характеризуется разрешающей способностью и глубиной цвета. Для реализации каждого графического режима требуется определенный объем видеопамати компьютера.

**Пример 2.48.** Определить объем видеопамати компьютера, который необходим для реализации графического режима монитора *High Color* с разрешающей способностью 1024×768 точек и палитрой из 65536 цветов.

Глубина цвета составляет:

$$I = \log_2 65\,536 = 16 \text{ бит.}$$

Количество точек изображения равно:

$$1024 \times 768 = 786\,432.$$

Требуемый объем видеопамати равен:

$$16 \text{ бит} \cdot 786 \, 432 = 12 \, 582 \, 912 \text{ бит} \approx 1,2 \text{ Мбайта.}$$

Важнейшими характеристиками монитора являются размеры его экрана, которые задаются величиной его диагонали в дюймах (15", 17", 21" и так далее) и размером точки экрана (0,25 мм или 0,28 мм), а разрешающая способность экрана монитора задается количеством точек по вертикали и горизонтали (640×480, 800×600 и так далее). Следовательно, для каждого монитора существует физически максимально возможная разрешающая способность экрана.

**Пример 2.49.** Определить максимально возможную разрешающую способность экрана для монитора с диагональю 15" и размером точки экрана 0,28 мм.

Выразим размер диагонали в сантиметрах:

$$2,54 \text{ см} \cdot 15 = 38,1 \text{ см.}$$

Определим соотношение между высотой и шириной экрана для режима 1024×768 точек:

$$768 : 1024 = 0,75.$$

Определим ширину экрана. Пусть ширина экрана равна  $L$ , тогда высота равна  $0,75L$ . По теореме Пифагора имеем:

$$\begin{aligned} L^2 + (0,75L)^2 &= 38,1^2, \\ 1,5625L^2 &= 1451,61, \\ L^2 &\approx 929, \\ L &\approx 30,5 \text{ см.} \end{aligned}$$

Количество точек по ширине экрана равно:

$$305 \text{ мм} : 0,28 \text{ мм} = 1089.$$

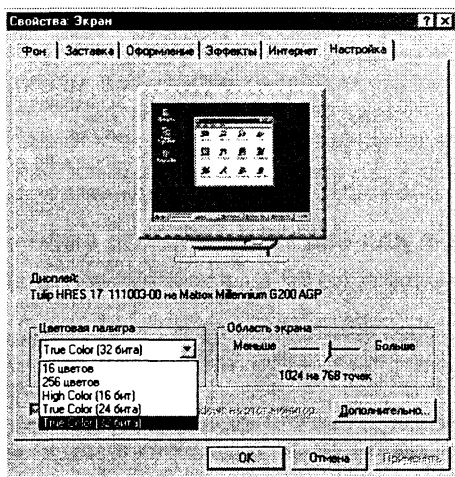
Максимально возможным разрешением экрана монитора является 1024×768.

**Пример 2.50.** Установить графический режим экрана монитора, исходя из объема установленной видеопамати и параметров монитора.



### Установка графического режима экрана монитора

- 1 Ввести команду [Настройка-Панель управления-Экран] или щелкнуть по индикатору монитора на панели задач.
- 2 На появившейся диалоговой панели *Свойства: экран* выбрать вкладку *Настройка*.
- 3 С помощью раскрывающегося списка *Цветовая палитра* выбрать глубину цвета.  
С помощью ползунка *Область экрана* выбрать разрешение экрана.



Цветное растровое изображение формируется в соответствии с цветовой моделью RGB, в которой тремя базовыми цветами являются Red (красный), Green (зеленый) и Blue (синий). В режиме *True Color (24 бита)* интенсивность каждого цвета задается 8-битным двоичным кодом, который часто для удобства выражают в шестнадцатеричной системе счисления. В этом случае используется следующий формат записи RRGGBB.

**Пример 2.51.** Запишите код красного цвета в двоичном, шестнадцатеричном и десятичном представлении.

Красный цвет соответствует максимальному значению интенсивности красного и минимальным значениям интенсивностей зеленого и синего базовых цветов. Таким образом, числовой код красного цвета следующий:

Коды/Цвета	Красный	Зеленый	Синий
двоичный	11111111	00000000	00000000
шестнадцатеричный	FF	00	00
десятичный	256	0	0

**Пример 2.52.** Сканируется цветное изображение размером  $10 \times 10$  см. Разрешающая способность сканера 600 dpi и глубина цвета 32 бита. Какой информационный объем будет иметь полученный графический файл.

Разрешающая способность сканера 600 dpi (dot per inch — точек на дюйм) означает, что на отрезке длиной 1 дюйм сканер способен различить 600 точек.

Переведем разрешающую способность сканера из точек на дюйм (1 дюйм = 2,54 см) в точки на сантиметр:

$$600 \text{ dpi} : 2,54 \approx 236 \text{ точек/см.}$$

Следовательно, размер изображения в точках составит  $2360 \times 2360$  точек.

Общее количество точек изображения равно:

$$2360 \cdot 2360 = 5\,569\,600.$$

Информационный объем файла равен:

$$32 \text{ бита} \cdot 5\,569\,600 = 178\,227\,200 \text{ бит} \approx 21 \text{ Мбайт}.$$

### Задания для самостоятельного выполнения

- 2.67. Определите количество цветов в палитре при глубине цвета 4, 8, 16, 24, 32 бита.
- 2.68. Черно-белое (без градаций серого) растровое графическое изображение имеет размер  $10 \times 10$  точек. Какой объем памяти займет это изображение?
- 2.69. Цветное (с палитрой из 256 цветов) растровое графическое изображение имеет размер  $10 \times 10$  точек. Какой объем памяти займет это изображение?
- 2.70. В процессе преобразования растрового графического изображения количество цветов уменьшилось с 65536 до 16. Во сколько раз уменьшится объем занимаемой им памяти?
- 2.71. В процессе преобразования растрового графического изображения количество цветов увеличилось с 16 до 42 949 67 296. Во сколько раз увеличился объем, занимаемый им в памяти?
- 2.72. 256-цветный рисунок содержит 120 байт информации. Из скольких точек он состоит?
- 2.73. Для хранения изображения размером  $64 \times 32$  точек выделено 64 Кбайт памяти. Определите, какое максимальное число цветов допустимо использовать в этом случае.
- 2.74. Определить соотношение между высотой и шириной экрана монитора для различных графических режимов. Различается ли это соотношение для различных режимов?  
а)  $640 \times 480$ ; б)  $800 \times 600$ ; в)  $1024 \times 768$ ; а)  $1152 \times 864$ ; а)  $1280 \times 1024$ .
- 2.75. Определить максимально возможную разрешающую способность экрана для монитора с диагональю 17" и размером точки экрана 0,25 мм.
- 2.76. Определите требуемый объем видеопамати для различных графических режимов экрана монитора. Заполните таблицу.

Разрешающая способность экрана	Глубина цвета (битов на точку)				
	4	8	16	24	32
<b>640 × 480</b>					
<b>800 × 600</b>					
<b>1024 × 768</b>					
<b>1280 × 1024</b>					

- 2.77. Достаточно ли видеопамати объемом 256 Кбайт для работы монитора в режиме  $640 \times 480$  и палитрой из 16 цветов?

- 2.78. Какие графические режимы работы монитора может обеспечить видеопамять объемом в 1 Мбайт?
- 2.79. Установить различные графические режимы экрана монитора вашего компьютера:
- режим с максимально возможной глубиной цвета;
  - режим с максимально возможной разрешающей способностью;
  - оптимальный режим.
- 2.80. Заполните таблицу цветов при 24-битной глубине цвета в шестнадцатеричном представлении.

Название цвета	Интенсивность		
	Красный	Зеленый	Синий
Черный			
Красный			
Зеленый			
Синий			
Белый			

- 2.81. Сканируется цветное изображение стандартного размера А4 (21×29,7 см). Разрешающая способность сканера 1200 dpi и глубина цвета 24 бита. Какой информационный объем будет иметь полученный графический файл.

## 2.9. Кодирование звуковой информации

### 2.13. Двоичное кодирование звуковой информации

Учебник 

В аналоговой форме звук представляет собой волну с непрерывно меняющейся амплитудой и частотой. При преобразовании звука в цифровую форму производится временная дискретизация, при которой в определенные моменты времени амплитуда звуковой волны измеряется и квантуется, то есть ей присваивается определенное значение из некоторого фиксированного набора. Данный метод называется еще импульсно-кодовой модуляцией РСМ (Pulse Code Modulation).

Преобразование непрерывной звуковой волны в последовательность звуковых импульсов различной амплитуды производится с помощью аналого-цифрового преобразователя, размещенного на звуковой плате. Современные 16-битные звуковые карты обеспечивают возможность кодирования 65536 различных уровней громкости или 16-битную глубину кодирования звука. Качество кодирования звука зависит и от частоты дискретизации — количества измерений уровня сигнала в единицу времени. Эта величина может принимать значения от 8 до 48 кГц.

**Пример 2.53.** Оцените информационный объем высококачественного стереоаудиофайла длительностью звучания 1 минута, если «глубина» кодирования 16 бит, а частота дискретизации 48 кГц.

Информационный объем звукового файла длительностью в 1 секунду равен:

$$16 \text{ бит} \cdot 48\,000 \cdot 2 = 1\,536\,000 \text{ бит} = 187,5 \text{ Кбайт.}$$

Информационный объем звукового файла длительностью 1 минута равен:

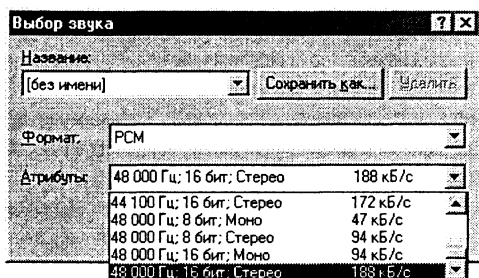
$$187,5 \text{ Кбайт/с} \cdot 60 \text{ с} \approx 11 \text{ Мбайт.}$$

**Пример 2.54.** Запишите звуковой файл длительностью 1 минута с «глубиной» кодирования 16 бит и частотой дискретизации 48 кГц. Сравните его объем с вычисленным значением в предыдущем примере.



#### Запись звукового файла с помощью стандартного приложения Звукозапись

- 1 Ввести команду [Программы-Стандартные-Мультимедиа-Звукозапись].
- 2 В появившемся окне приложения *Звук-Звукозапись* ввести команду [Файл-Свойства].
- 3 На появившейся диалоговой панели *Свойства: Звук* щелкнуть по кнопке *Преобразовать*.
- 4 На появившемся диалоговом окне *Выбор звука* с помощью раскрывающегося списка *Формат*: выбрать стандартный формат *PCM*.  
В раскрывающемся списке *Атрибуты*: выбрать качество дискретизации, например, *48 000 Гц; 16 бит; Стерео*.



- 5 Получаемое значение объема 1 секунды оцифрованного звука (188 Кбайт) приблизительно равно вычисленному (187,5 Кбайт).
- 6 Запишите звуковой файл с выбранным качеством, сохраните его на диске и сравните его объем с вычисленным.

Записанные звуковые файлы можно редактировать, то есть вырезать, копировать и вставлять фрагменты файла. Кроме того, можно увеличивать или уменьшать громкость, применять различные звуковые эффекты (эхо, уменьшение или увеличение скорости воспроизведения, воспроизведение в обратном направлении и др.), а также накладывать файлы друг на друга (микшировать). Можно также изменять качество звука путем уменьшения или увеличения глубины кодирования и частоты дискретизации. Для редактирования звуковых файлов применяются специальные программы — звуковые редакторы.

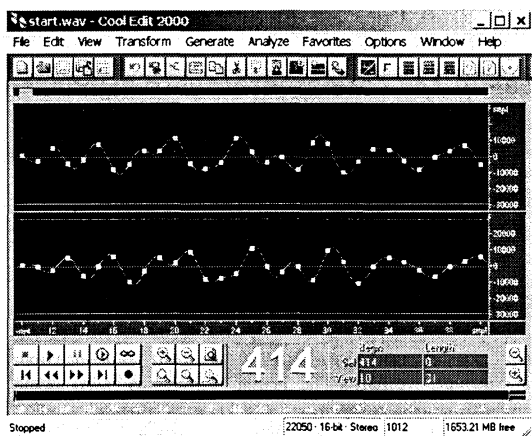
**Пример 2.55.** В звуковом редакторе Cool Edit 2000 улучшить качество звукового файла, например `ding.wav`, который хранится в каталоге `\\WINDOWS\\Media\\`. Сравнить объемы исходного и полученного файлов.

### Редактирование звукового файла в звуковом редакторе Cool Edit 2000

- 1 Запустить звуковой редактор Cool Edit 2000.

Открыть звуковой файл `start.wav` командой [File-Open].

В окне наблюдать зависимость амплитуды сигнала от времени с точками дискретизации для двух каналов.

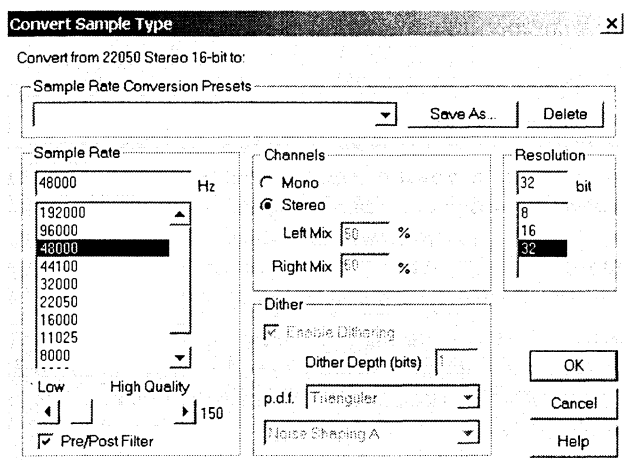


- 2 Ввести команду [Edit-Convert Sample Type ...].

Загруженный файл записан с глубиной кодирования 16 бит и частотой дискретизации 22050 Гц.

В списке *Sample Rate* выбрать частоту дискретизации, например 48 000 Гц, в списке *Resolution* глубину кодирования, например 32 бита.





- 3 Полученный файл имеет объем 344 Кбайта, а исходный файл имеет объем 79 Кбайт. Определим во сколько раз полученный файл больше:

$$344 \text{ Кбайта} / 79 \text{ Кбайт} = 4,35.$$

- 4 Вычислим, во сколько раз полученный файл должен иметь больший объем:

$$48000 \text{ Гц} / 22050 \text{ Гц} \cdot 32 \text{ бита} / 16 \text{ бит} = 4,35.$$

### Задания для самостоятельного выполнения

- 2.82. Оцените информационный объем моноаудиофайла длительностью звучания 1 мин, если «глубина» кодирования и частота дискретизации звукового сигнала равны соответственно:

- 16 бит и 8 кГц;
- 16 бит и 24 кГц.

Запишите звуковые файлы с такими параметрами и сравните полученные объемы с вычисленными.

- 2.83. Определите качество звука (качество радиотрансляции, среднее качество, качество аудио-CD), если известно, что объем моноаудиофайла длительностью звучания в 10 сек. равен:

- 940 Кбайт;
- 157 Кбайт.

- 2.84. Рассчитайте время звучания моноаудиофайла, если при 16-битном кодировании и частоте дискретизации 32 кГц его объем равен:

- 700 Кбайт;
- 6300 Кбайт.

- 2.85. Определите длительность звукового файла, который уместится на гибкой дискете 3,5". Учтите, что для хранения данных на такой дискете выделяется 2847 секторов объемом 512 байт.
- а) при низком качестве звука: моно, 8 бит, 8 кГц;
  - б) при высоком качестве звука: стерео, 16 бит, 48 кГц.
- 2.86. Аналоговый звуковой сигнал был дискретизирован сначала с использованием 256 уровней интенсивности сигнала (качество звучания радиотрансляции), а затем с использованием 65536 уровней интенсивности сигнала (качество звучания аудио-CD). Во сколько раз различаются информационные объемы оцифрованного звука?

## 2.10. Хранение информации

### 1.8. Логическая структура дисков Учебник

Для долговременного хранения информации используются материальные носители информации самой разнообразной природы: аналоговые (бумага, магнитная лента, фото- и киноплёнка и др.) и цифровые, дискретные (молекулы ДНК, микросхемы памяти, магнитные и оптические диски).

Цифровые носители информации очень чувствительны к повреждениям, так как даже потеря одного бита информации может привести к негативным последствиям. Например, изменение в структуре ДНК может вызвать опасные генетические мутации в организме, а повреждение магнитных и оптических носителей — потерю программ и данных.

**Пример 2.56.** Какие физические воздействия (магнитные поля, нагревание, удары, загрязнения) могут привести к потере информации:

- а) на гибких магнитных дисках;
- б) на жестких магнитных дисках;
- в) на оптических дисках?

Магнитные поля и нагревание могут повредить информацию на гибких и жестких магнитных дисках. Удары могут повредить жесткие магнитные диски. Загрязнение поверхности оптических дисков может привести к невозможности считывания информации.

**Пример 2.57.** Определить объем генетической информации молекулы ДНК человека, которая состоит из около 6 миллиардов нуклеотидов четырех типов (А, Г, Т, С), которые являются знаками генетического алфавита.

Мощность генетического алфавита равна четырем, следовательно, каждый знак (нуклеотид) несет количество информации, которое можно определить по формуле (2.2):

$$I = \log_2 4 = 2 \text{ бита.}$$

Объем генетической информации в ДНК человека составляет:

$$2 \text{ бита} \cdot 6\,000\,000\,000 \approx 1,4 \text{ Гбайта.}$$

**Пример 2.58.** Какое количество ячеек оперативной памяти будет занято словом «информатика», записанным в формате Unicode.

Оперативная память компьютера состоит из ячеек, объем которых равен 1 байту.

В формате Unicode каждый символ кодируется двумя байтами, следовательно, количество занятых ячеек памяти будет 22.

**Пример 2.59.** Какое максимальное количество страниц текста, содержащего 60 символов в строке и 40 строк на странице, может содержать текстовый файл без символов форматирования (формат TXT), сохраненный в кодировке Windows на гибком магнитном диске?

На гибком магнитном диске формата 3,5" минимальным адресуемым элементом является сектор емкостью 512 байт. Всего таких секторов 2880, из них для хранения данных отводится 2847 секторов, один сектор (1-й) отводится для размещения загрузчика операционной системы и 32 сектора отводятся для хранения каталога диска и таблицы размещения файлов FAT.

Информационный объем гибкого диска, предназначенный для хранения данных, составляет:

$$512 \text{ байт} \cdot 2847 = 1423,5 \text{ Кбайта.}$$

Информационный объем страницы составляет:

$$1 \text{ байт} \cdot 60 \cdot 40 = 2400 \text{ байт} \approx 2,34 \text{ Кбайта.}$$

Максимальное количество страниц в текстовом файле, размещенном на гибком магнитном диске, составляет:

$$1423,5 \text{ Кбайта} : 2,34 \text{ Кбайта} \approx 608.$$

**Пример 2.60.** Какой информационный объем будет занимать текстовый файл, содержащий слово «информатика», сохраненный в кодировке Windows на гибком магнитном диске формата 3,5", на жестком диске 50 Гбайт с FAT16 и с FAT32?

Минимальным адресуемым элементом жесткого диска является кластер, размер которого зависит от типа используемой таблицы размещения файлов FAT и емкости жесткого диска. Таблица FAT16 позволяет адресовать  $2^{16} = 65536$  кластеров, что приводит к большим размерам кластеров на жестких дисках большой емкости и нерациональному использованию дискового пространства.

Таблица FAT32 логически разбивает жесткий диск на кластеры, содержащие по восемь секторов. Таким образом, независимо от информационной емкости жесткого диска размер кластера составляет 4 Кбайта.

Информационный объем текста равен:

$$1 \text{ байт} \cdot 11 = 11 \text{ байт.}$$

На гибком диске этот файл будет занимать один сектор, т. е. 512 байт.

На жестком диске с FAT16 файл будет занимать один кластер, объем которого равен:

$$50 \text{ Гбайт} : 65536 = 800 \text{ Кбайт.}$$

На жестком диске с FAT32 файл будет занимать один кластер, объем которого равен 4 Кбайта.



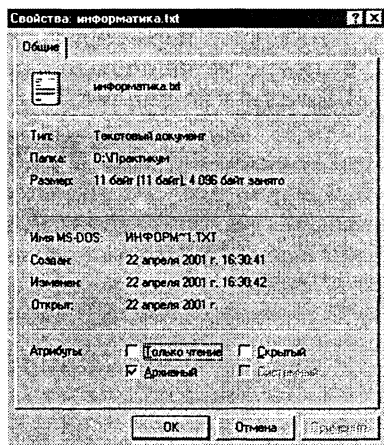
### Объем файла на гибком и жестком магнитных дисках

- 1 Создать в стандартном приложении Windows Блокнот файл, содержащий слово «информатика».
- 2 Сохранить файл на гибком и жестком дисках под именем информатика.txt.
- 3 Найти в каталогах гибкого и жесткого дисков файл информатика.txt и в контекстном меню выбрать пункт *Свойства*.

На появившейся диалоговой панели *Свойства: информатика.txt* ознакомиться со строкой *Размер*:

Для гибкого диска: *11 байт (11 байт), 512 байт занято.*

Для жесткого диска: *11 байт (11 байт), 4096 байт занято.*



Каждый файл на диске занимает определенное количество секторов (кластеров). Кластеры нумеруются и каталог диска содержит указание на начало файла (содержит номер первого кластера файла). Информация о последовательности кластеров (номера кластеров), в которых хранится файл, содержится в таблице FAT.

В процессе работы компьютера могут происходить сбои (зависание программ, внезапное отключение питания и др.), в результате происходит неправильное завершение работы прило-

жений и операционной системы, что может приводить к повреждению отдельных кластеров и файлов. Могут появиться сбойные (нечитаемые) кластеры, в каталогах могут быть изменены имена файлов, а в таблицах FAT могут появиться нарушения в цепочках размещения файлов (некоторые цепочки могут быть оборваны, один и тот же кластер может принадлежать различным файлам и др.).

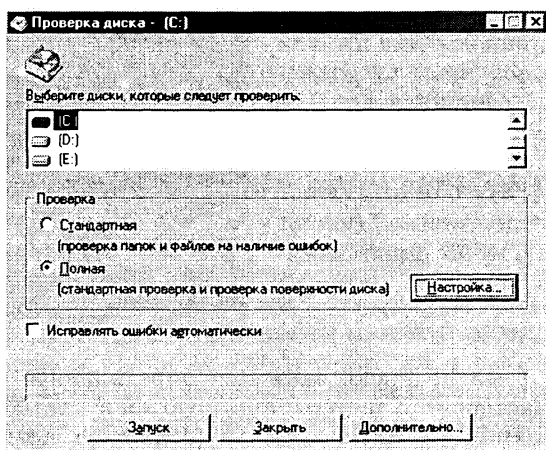
Для восстановления файловой системы используются специальные программы. В операционной системе Windows такой программой является служебная программа Проверка диска, которая автоматически запускается при загрузке Windows после неправильного завершения работы или может быть при необходимости запущена пользователем в произвольный момент.

**Пример 2.61.** Осуществить проверку файловой системы диска.

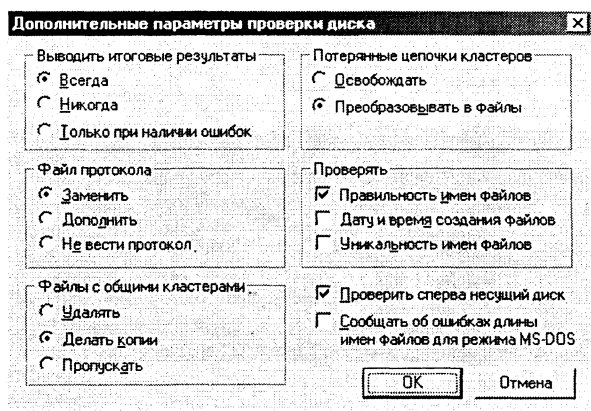


### Проверка файловой системы диска

- 1 Запустить служебную программу Проверка диска командой [Программы-Стандартные-Служебные-Проверка диска].
- 2 На появившемся диалоговом окне программы выбрать проверяемый диск (например, C:).  
Установить переключатель *Проверка* в положение *Полная*, если требуется проверка поверхности диска.  
Щелкнуть по кнопке *Дополнительно*.



- 3 На появившейся диалоговой панели *Дополнительные параметры проверки диска* с помощью переключателей и флажков установить требуемые параметры и щелкнуть по кнопке *ОК*.



4 В окне программы щелкнуть по кнопке *Запуск*. После окончания проверки и восстановления сбойных кластеров и файлов появится окно с информацией о состоянии диска после проверки.

С течением времени в процессе записи и удаления файлов происходит их дефрагментация, то есть нарушается первоначальное размещение файлов в последовательно идущих друг за другом кластерах. В результате файлы могут быть размещены в кластерах, хаотически разбросанных по всему диску, что замедляет доступ к ним и может привести к преждевременному износу жесткого диска.

Рекомендуется периодически проводить дефрагментацию дисков, то есть восстановление первоначального упорядоченного размещения файлов в последовательных секторах. Дефрагментация дисков осуществляется с помощью специальных программ, в состав Windows входит служебная программа Дефрагментация диска.

**Пример 2.62.** Осуществить дефрагментацию диска.

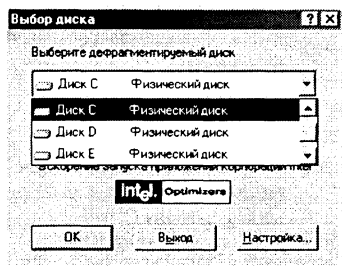


## Дефрагментация диска

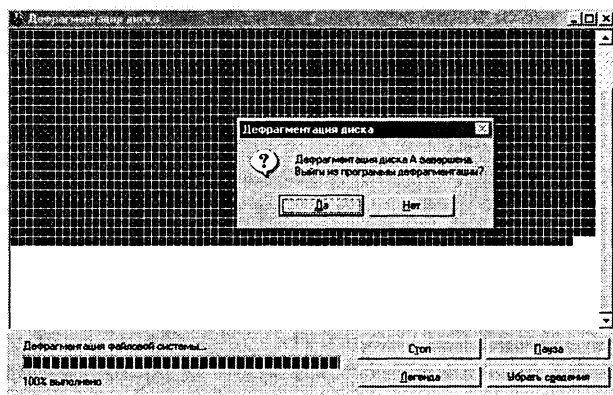
1 Запустить служебную программу Дефрагментация диска командой [Программы-Стандартные-Служебные-Дефрагментация диска].

2 На появившейся диалоговой панели *Выбор диска* выбрать дефрагментируемый диск (например, С:).

Щелкнуть по кнопке *OK*.



- 3 В появившемся окне *Дефрагментация диска* можно визуаль-но наблюдать процесс дефрагментации и его результат, если щелкнуть по кнопке *Сведения*.



### Задания для самостоятельного выполнения

- 2.87. Информация на каких носителях (гибких магнитных дисках, жестких магнитных дисках, оптических дисках) может быть утеряна (перестать считываться), если:
- хранить носители несколько часов под прямыми лучами солнца;
  - уронить носитель со стола;
  - случайно прикоснуться загрязненной рукой к поверхности носителя?
- 2.88. Какой объем оперативной памяти требуется для хранения текста статьи объемом 4 страницы, на каждой из которых размещены 32 строки по 64 символа?
- 2.89. Часть страниц многотомной энциклопедии является цветными изображениями в шестнадцатичетной палитре и в формате 320 × 640 точек; страницы, содержащие текст, имеют формат — 32 строки по 64 символа в строке. Сколько страниц книги можно сохранить на жестком магнитном диске объемом 20 Мб, если каждая девятая страница энциклопедии — цветное изображение?
- 2.90. Сколько текстовых файлов можно записать на гибкий диск формата 3,5", если информационный объем текста:
- 10 байт;
  - 500 байт;
  - 1030 байт.
- 2.91. Информация о каждом из 88 сотрудников фирмы объемом 18390 знаков находится в отдельном файле. Можно ли, не прибегая к архивированию, переписать все эти файлы на 1 гибкий магнитный диск формата 3,5"?

- 2.92.** Информация о каждом из 88 сотрудников фирмы объемом 18390 знаков находится в одном общем файле `sotr.txt`. Можно ли, не прибегая к архивированию, переписать этот файл на один гибкий магнитный диск формата 3,5"?
- 2.93.** Сколько текстовых файлов объемом 400 байт можно записать на жесткий диск, если используется таблица размещения файлов FAT16 и емкость жесткого диска равна:
- а) 200 Мбайт;
  - б) 2 Гбайта;
  - в) 20 Гбайт?
- 2.94.** Сколько текстовых файлов объемом 400 байт можно записать на жесткий диск, если используется таблица размещения файлов FAT32 и емкость жесткого диска равна:
- а) 200 Мбайт;
  - б) 2 Гбайта;
  - в) 20 Гбайт?
- 2.95.** Определите размер кластера жесткого диска, установленного на вашем компьютере.
- 2.96.** Если на вашем компьютере установлен жесткий диск объемом более 1 Гбайта, преобразовать таблицу размещения файлов из FAT16 в FAT32 с использованием служебной программы Windows Преобразование диска в FAT32.
- 2.97.** Осуществите проверку дисков вашего компьютера с помощью служебной программы Windows Проверка диска.
- 2.98.** Осуществите дефрагментацию дисков вашего компьютера с помощью служебной программы Windows Дефрагментация.



# Глава 3

---

## Основы логики и логические основы компьютера

---

В процессе изучения данной темы рекомендуется установить следующее свободно распространяемое программное обеспечение:

CD-ROM 

- калькуляторы Wise Calculator и NumLock Calculator для выполнения логических операций;
- электронные таблицы StarOffice Calc, входящие в состав интегрированного офисного приложения StarOffice, для построения таблиц истинности логических выражений.

### 3.1. Основы логики

Первые учения о формах и способах рассуждений возникли в странах Древнего Востока (Китай, Индия), но в основе современной логики лежат учения, созданные в 4 веке до нашей эры древне-греческими мыслителями. Основы формальной логики заложил Аристотель, который впервые отделил логические формы речи от ее содержания. Он исследовал терминологию логики, подробно разобрал теорию умозаключений и доказательств, описал ряд логических операций, сформулировал основные законы мышления.

Логика изучает внутреннюю структуру процесса мышления, который реализуется в таких естественно сложившихся формах как понятие, суждение, умозаключение и доказательство.

**Понятие.** Понятие — это форма мышления, отражающая наиболее существенные свойства предмета, отличающие его от других предметов. В структуре каждого понятия нужно различать две стороны: содержание и объем. Содержание понятия составляет совокупность существенных признаков предмета. Чтобы раскрыть содержание понятия, следует выделить признаки, необходимые и достаточные для выделения данного предмета по отношению к другим предметам.

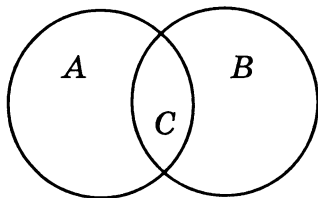
Объем понятия определяется совокупностью предметов, на которую оно распространяется, и может быть представлено в

форме множества объектов, состоящего из элементов множества. *Алгебра множеств*, одна из основополагающих современных математических теорий, позволяет исследовать отношения между множествами и, соответственно, объемами понятий.

Между множествами (объемами понятий) могут быть различные виды отношений:

- равнозначность, когда объемы понятий полностью совпадают;
- пересечение, когда объемы понятий частично совпадают;
- подчинение, когда объем одного понятия полностью входит в объем другого, и так далее.

Для наглядной геометрической иллюстрации объемов понятий и соотношений между ними используются диаграммы Эйлера–Венна. Если имеются какие-либо понятия  $A$ ,  $B$ ,  $C$  и так далее, то объем каждого понятия (множество) можно представить в виде круга, а отношение между этими объемами (множествами) в виде пересекающихся кругов.



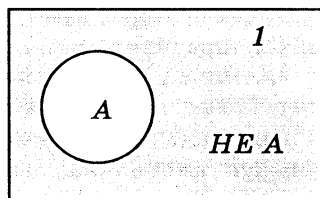
**Пример 3.1.** Отобразить с помощью диаграммы Эйлера–Венна соотношение между объемами понятий *натуральные числа* и *четные числа*.

Объем понятия *натуральные числа* включает в себя множество целых положительных чисел  $A$ , а объем понятия *четные числа* включает в себя множество отрицательных и положительных четных чисел  $B$ . Эти множества пересекаются, т. к. включают в себя множество положительных четных чисел  $C$ .

Совокупность всех существующих множеств образует *всеобщее универсальное множество*  $1$ , которое позволяет отобразить множество, логически противоположное к заданному. Так, если задано множество  $A$ , то существует множество  $НЕ A$ , которое объединяет все объекты, не входящие во множество  $A$ . Множество  $НЕ A$  дополняет множество  $A$  до универсального множества  $1$ .

**Пример 3.2.** Отобразить с помощью диаграммы Эйлера–Венна множество натуральных чисел  $A$  и множество  $НЕ A$ .

На диаграмме Эйлера–Венна универсальное множество  $1$  изображается в виде прямоугольника, множество  $A$  в форме круга, а множество  $НЕ A$  — в форме «прямоугольник минус круг».



**Высказывание.** Высказывание (суждение) — это форма мышления, выраженная с помощью понятий, посредством которой что-либо утверждают или отрицают о предметах, их свойствах и отношениях между ними.

О предметах можно судить верно или неверно, то есть высказывание может быть *истинным* или *ложным*. Истинным будет суждение, в котором связь понятий правильно отражает свойства и отношения реальных вещей. Ложным суждение будет в том случае, когда связь понятий искажает объективные отношения, не соответствует реальной действительности.

Обоснование истинности или ложности простых высказываний решается вне алгебры логики. Например, истинность или ложность высказывания: «Сумма углов треугольника равна 180 градусов» устанавливается геометрией, причем — в геометрии Евклида это высказывание является истинным, а в геометрии Лобачевского — ложным.

В естественном языке высказывания выражаются повествовательными предложениями. Высказывание не может быть выражено повелительным или вопросительным предложением, оценка истинности или ложности которых невозможна. Высказывания могут выражаться с помощью математических, физических, химических и прочих знаков. Из двух числовых выражений можно составить высказывания, соединив их знаками равенства или неравенства.

Высказывание называется *простым*, если никакая его часть сама не является высказыванием. Высказывание, состоящее из простых высказываний, называется *составным* (сложным).

Высказывания имеют определенную логическую форму. Понятие о предмете мысли называется *субъектом* и обозначается буквой  $S$ , а понятие о свойствах и отношениях предмета мысли называется *предикатом* и обозначается буквой  $P$ . Оба эти понятия — субъект и предикат называются *терминами* суждения. Отношения между субъектом и предикатом выражаются *связкой* «есть», «не есть», «является», «состоит» и так далее.

Таким образом, каждое высказывание состоит из трех элементов — субъекта, предиката и связки (двух терминов и связки). Состав суждения можно выразить общей формулой « $S$  есть  $P$ » или « $S$  не есть  $P$ ».

**Пример 3.3.** Определить, что в суждении «Компьютер состоит из процессора, памяти и внешних устройств» является субъектом, предикатом и связкой.

«Компьютер» — субъект, «процессора, памяти и внешних устройств» — предикат, «состоит» — связка.

**Предикат.** В современной логике предикат рассматривается как функциональная зависимость. В общем случае предикат от

$n$  переменных (от  $n$  неопределенных понятий) выражается формулой:

$$P(x_1, x_2, \dots, x_n), \text{ где } n > 0.$$

При  $n = 1$ , когда один из терминов является неопределенным понятием, мы имеем предикат первого порядка, например, « $x$  — человек».

При  $n = 2$ , когда два термина не определены, мы имеем предикат второго порядка, например, « $x$  любит  $y$ ».

При  $n = 3$ , когда неопределенны три термина, мы имеем предикат третьего порядка, например, « $z$  — сын  $x$  и  $y$ ».

**Пример 3.4.** В вышеописанных предикатах заменить неопределенные термины на конкретные понятия.

Преобразуем предикаты в высказывания путем подстановки вместо переменных соответствующих понятий:  $x =$  «Сократ»,  $y =$  «Ксантиппа»,  $z =$  «Софрониск»:

«Сократ — человек»,

«Ксантиппа любит Сократа»,

«Софрониск — сын Сократа и Ксантиппы».

**Умозаключение.** Умозаключение — это форма мышления, посредством которой из одного или нескольких суждений, называемых посылками, по определенным правилам логического вывода получается новое знание о предметах реального мира (вывод).

Умозаключения бывают дедуктивные, индуктивные и по аналогии. В дедуктивных умозаключениях рассуждения ведутся от общего к частному. Например, из двух суждений: «Все металлы электропроводны» и «Ртуть является металлом» путем умозаключения можно сделать вывод, что «Ртуть электропроводна».

В индуктивных умозаключениях рассуждения ведутся от частного к общему. Например, установив, что отдельные металлы — железо, медь, цинк, алюминий и так далее — обладают свойством электропроводности, можно сделать вывод, что все металлы электропроводны.

Умозаключение по аналогии представляет собой движение мысли от общности одних свойств и отношений у сравниваемых предметов или процессов к общности других свойств и отношений. Например, химический состав Солнца и Земли сходен по многим показателям, поэтому, когда на Солнце обнаружили неизвестный еще на Земле химический элемент гелий, то по аналогии заключили: такой элемент есть и на Земле.

**Доказательство.** Доказательство есть мыслительный процесс, направленный на подтверждение или опровержение какого-либо положения посредством других несомненных, ранее

обоснованных доводов. Доказательство по своей логической форме не отличается от умозаключения. Однако, если в умозаключении заранее исходят из истинности посылок и следят только за правильностью логического вывода, в доказательстве подвергается логической проверке истинность самих посылок.

### Задания для самостоятельного выполнения

- 3.1. Отобразить с помощью диаграммы Эйлера–Венна соотношения между следующими объемами понятий:
- целые и натуральные числа;
  - четные и нечетные числа.
- 3.2. Определить, что является субъектом, предикатом и связкой в следующих суждениях:
- Сканер — это устройство ввода информации.
  - Луна является спутником Земли.
  - Атом состоит из ядра и электронов.
- 3.3. Приведите примеры понятий, суждений, умозаключений и доказательств из различных наук: математики; информатики; физики и химии.
- 3.4. При каких значениях числа  $X$  предикат первого порядка **не**  $((X > 8)$  **или**  $(X < -3))$  примет значение:
- ложь;
  - истина?
- 3.5. Какие предикаты первого порядка описывают условие: «Точка  $X$  не принадлежит отрезку  $[A; B]$ »?
- не**  $(X \geq A)$  **и**  $X < B$ ;
  - $X < A$  **или**  $X > B$ ;
  - не**  $(X \leq B$  **или**  $X \geq A)$ ;
  - $X \leq A$  **и**  $X \geq B$ .
- 3.6. Изобразите в декартовой прямоугольной системе координат область, в которой и только в которой истинны следующие предикаты второго порядка:
- $(Y \geq X)$  **и**  $(Y + X \geq 0)$  **и**  $(Y \leq 1)$ ;
  - $(|X| \leq 1)$  **и**  $(|Y| \leq 1)$ .

## 3.2. Алгебра высказываний

*Алгебра* в широком смысле этого слова — наука об общих операциях, аналогичных сложению и умножению, которые могут выполняться над различными математическими объектами (алгебра переменных и функций, алгебра векторов, алгебра множеств и так далее). Объектами алгебры логики являются высказывания.

Алгебра логики отвлекается от смысловой содержательности высказываний. Ее интересует только один факт — истинно или ложно данное высказывание, что дает возможность определять истинность или ложность составных высказываний алгебраическими методами.

Простые высказывания в алгебре логики обозначаются заглавными латинскими буквами:

$A = \{\text{Аристотель — основоположник логики}\};$

$B = \{\text{На яблонях растут бананы}\}.$

Истинному высказыванию ставится в соответствие 1, ложному — 0. Таким образом,  $A = 1, B = 0$ .

Составные высказывания на естественном языке образуются с помощью союзов, которые в алгебре высказываний заменяются на логические операции. Логические операции задаются таблицами истинности и могут быть графически проиллюстрированы с помощью диаграмм Эйлера–Венна.

**Логическая операция КОНЪЮНКЦИЯ (логическое умножение):**

- в естественном языке соответствует союзу и;
- в алгебре высказываний обозначение &;
- в языках программирования обозначение **And**.

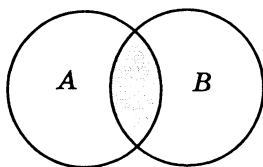
Конъюнкция — это логическая операция, ставящая в соответствие каждому двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания истинны.

В алгебре множеств конъюнкции соответствует операция *пересечения множеств*, то есть множеству, получившемуся в результате умножения множеств  $A$  и  $B$ , соответствует множество, состоящее из элементов, принадлежащих одновременно двум множествам.

Таблица истинности

$A$	$B$	$A \& B$
0	0	0
0	1	0
1	0	0
1	1	1

Диаграмма Эйлера–Венна



**Логическая операция ДИЗЪЮНКЦИЯ (логическое сложение):**

- в естественном языке соответствует союзу или;
- обозначение  $\vee$ ;
- в языках программирования обозначение **Or**.

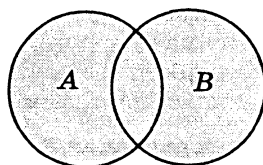
Дизъюнкция — это логическая операция, которая каждому двум простым высказываниям ставит в соответствие составное высказывание, являющееся ложным тогда и только тогда, когда оба исходных высказывания ложны и истинным, когда хотя бы одно из двух образующих его высказываний истинно.

В алгебре множеств дизъюнкции соответствует операция *объединения множеств*, то есть множеству, получившемуся в результате сложения множеств  $A$  и  $B$ , соответствует множество, состоящее из элементов, принадлежащих либо множеству  $A$ , либо множеству  $B$ .

Таблица истинности

$A$	$B$	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Диаграмма Эйлера–Венна



### Логическая операция ИНВЕРСИЯ (отрицание):

- в естественном языке соответствует словам *неверно, что...* и частице *не*;  $\bar{\quad}$
- обозначение  $\bar{A}$ ;
- в языках программирования обозначение **Not**.

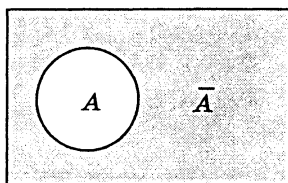
Отрицание — это логическая операция, которая каждому простому высказыванию ставит в соответствие составное высказывание, заключающееся в том, что исходное высказывание отрицается.

В алгебре множеств логическому отрицанию соответствует операция *дополнения до универсального множества*, то есть множеству, получившемуся в результате отрицания множества  $A$ , соответствует множество  $\bar{A}$ , дополняющее его до универсального множества.

Таблица истинности

$A$	$\bar{A}$
0	1
1	0

Диаграмма Эйлера–Венна




**Пример 3.5.** Получить таблицы истинности операций логического умножения, логического сложения и отрицания с использованием электронных таблиц StarOffice Calc.

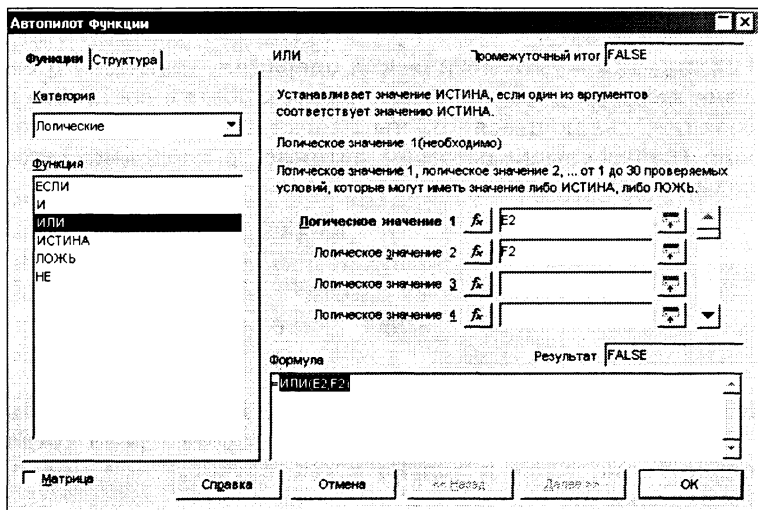
Электронные таблицы StarCalc обладают встроенными логическими функциями. Функция логического умножения И(ЛогЗнач1;ЛогЗнач2;...) дает значение TRUE (1), только тогда, когда все логические аргументы имеют значение TRUE (1).

Функция логического сложения ИЛИ(ЛогЗнач1;ЛогЗнач2;...) дает значение TRUE (1), тогда, когда хотя бы один логический аргумент имеет значение TRUE (1).

Функция логического отрицания НЕ(ЛогЗнач) дает значение TRUE (1), когда логический аргумент имеет значение FALSE (0) и, наоборот, значение FALSE (0), когда логический аргумент имеет значение TRUE (1).

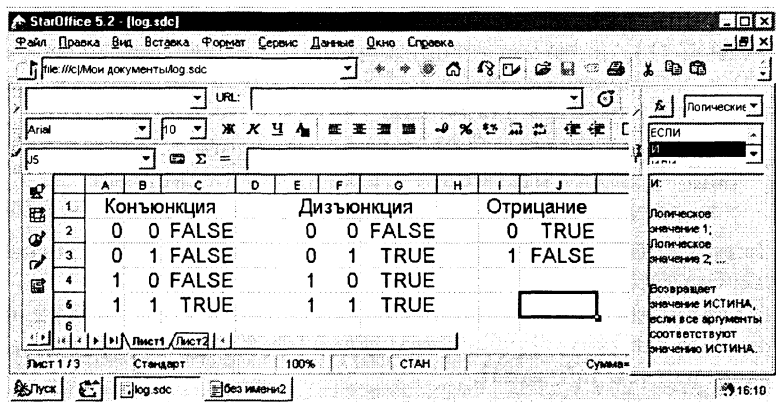
 **Получение таблиц истинности операций логического умножения, логического сложения и отрицания с использованием электронных таблиц StarOffice Calc**

- 1 Запустить интегрированное приложение командой [Программы-StarOffice].
- 2 В появившемся окне приложения StarOffice запустить электронные таблицы StarCalc командой [Файл-Создать-Документ электронной таблицы].
- 3 На листе *Лист1* создать таблицы истинности. Создать заголовки и ввести в столбцы *A* и *B*, *E* и *F*, *I* значения логических аргументов, а в столбцы *C*, *G*, *J* соответствующие логические функции.
- 4 Для ввода логических функций воспользоваться командой [Вставка-Функция]. На появившейся диалоговой панели *Автопилот функции* в раскрывающемся списке *Категории* выбрать *Логические*, а в окне *Функция* — функцию.





- 5 В окнах *Логическое значение 1* и так далее можно видеть и выбирать местонахождение аргументов функции. В окне *Формула* можно видеть текущее состояние заданной функции, а в окне *Результат* — текущее состояние ее значения.
- 6 После ввода аргументов функций и их формул на листе появятся таблицы истинности трех базовых логических функций.



- 7 Переименовать лист *Лист1* в *Логические операции* и сохранить в файле *log.sdc*.

Файл *log.sdc* хранится  
в каталоге *\practicum\logical*

CD-ROM

**Логическая операция ИМПЛИКАЦИЯ (логическое следование):**

- в естественном языке соответствует обороту *если ..., то ...*;
- обозначение  $\Rightarrow$ .

Импликация — это логическая операция, ставящая в соответствие каждому двум простым высказываниям составное высказывание, являющееся ложным тогда и только тогда, когда условие (первое высказывание) истинно, а следствие (второе высказывание) ложно.

$A$	$B$	$A \Rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

**Логическая операция ЭКВИВАЛЕНЦИЯ (равнозначность):**

- в естественном языке соответствует оборотам речи *тогда и только тогда*; *в том и только в том случае*;
- обозначения  $\Leftrightarrow$ ,  $\sim$ .

Эквиваленция — это логическая операция, ставящая в соответствие каждому двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания одновременно истинны или одновременно ложны.

Таблица истинности эквиваленции:

A	B	$A \Leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Логические операции имеют следующий приоритет: действия в скобках, инверсия,  $\&$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ .

**Пример 3.6.** Определите истинность составного высказывания:  $(\bar{A} \& \bar{B}) \& (C \vee D)$ , состоящего из простых высказываний:

- A = {Принтер — устройство вывода информации},
- B = {Процессор — устройство хранения информации},
- C = {Монитор — устройство вывода информации},
- D = {Клавиатура — устройство обработки информации}.


Сначала на основании знания устройства компьютера устанавливаем истинность простых высказываний: A = 1, B = 0, C = 1, D = 0.

Определим теперь истинность составного высказывания, используя таблицы истинности логических операций:

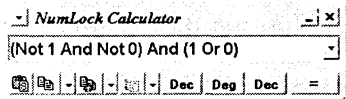
$$(\bar{1} \& \bar{0}) \& (1 \vee 0) = (0 \& 1) \& (1 \vee 0) = 0.$$

Составное высказывание ложно.

Проверим правильность полученного результата с использованием NumLock Calculator.

 **Определение истинности составного логического высказывания с использованием NumLock Calculator**

- 1 Запустить программу NumLock Calculator.
- 2 Установить удобный вид калькулятора командой [Вид калькулятора-Универсальный].
- 3 Ввести составное логическое выражение, подставив в него значения простых логических выражений.

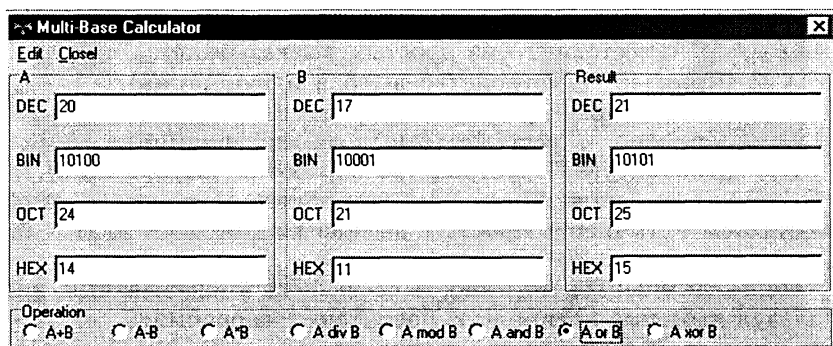


- 4 Нажав кнопку «=», получим значение (0) составного логического выражения, которое совпадает со значением, вычисленным алгебраически.

**Пример 3.7.** Даны три числа в различных системах счисления:  $A = 20_{10}$ ,  $B = 11_{16}$ ,  $C = 30_8$ . Переведите  $A$ ,  $B$  и  $C$  в двоичную систему счисления и выполните поразрядно логические операции  $(A \vee B) \& C$ . Ответ дайте в десятичной системе счисления.

### ■ Арифметические и логические операции с использованием Wise Calculator

- 1 Запустить программу Wise Calculator.
- 2 Ввести команду [Tools-Multi-Base Calculator].
- 3 Ввести значение логического аргумента  $A$  в текстовое поле *Dec* и аргумента  $B$  в текстовое поле *Hex* и с помощью группы переключателей *Operation* выбрать оператор логического сложения  $A$  or  $B$ .



- 4 В окне *Bin* столбца *Result* появится результат (10101) поразрядного логического сложения чисел  $A = 20_{10}$ ,  $B = 11_{16}$ .
- 5 Ввести значение промежуточного результата в текстовое поле *Bin* и аргумента  $C$  в текстовое поле *Oct* и с помощью группы переключателей *Operation* выбрать оператор логического умножения  $A$  and  $B$ .
- 6 В окне *Bin* столбца *Result* появится результат (10000) поразрядного логического умножения чисел  $A \vee B = 10101_2$ ,  $C = 30_8$ , а в окне *Dec* (16) результат в десятичной системе счисления.

**Пример 3.8.** Какие из высказываний  $A$ ,  $B$ ,  $C$  должны быть истинны и какие ложны, чтобы было ложно логическое выражение  $((A \vee B) \& B) \Rightarrow C$ .

Импликация ложна на единственном наборе логических значений (1, 0). Значит,  $((A \vee B) \& B) = 1$ .

Конъюнкция истинна на единственном наборе логических значений (1, 1). Значит,  $(A \vee B) = 1$  и  $B = 1$ .

Дизъюнкция истинна при наборах логических значений (0, 1) и (1, 1).

Следовательно, существуют два набора логических значений, удовлетворяющих условию задачи:  $(A = 0, B = 1, C = 0)$  и  $(A = 1, B = 1, C = 0)$ .

### Задания для самостоятельного выполнения

3.7. Выделите в составных высказываниях простые. Обозначьте каждое из них буквой; запишите с помощью логических операций каждое составное высказывание.

- 1) Число 376 четное и трехзначное.
- 2) Неверно, что Солнце движется вокруг Земли.
- 3) Если сумма цифр числа делится на 3, то число делится на 3.
- 4) Число 15 делится на 3 тогда и только тогда, когда сумма цифр числа 15 делится на 3.

3.8. Ниже приведена таблица, левая колонка которой содержит основные логические союзы (связки), с помощью которых в естественном языке строятся сложные высказывания. Заполните правую колонку таблицы соответствующими названиями логических операций.

В естественном языке	В логике
... и ...	
... или ...	
Неверно, что ...	
... в том и только в том случае ...	
... если ..., то ...	
... тогда и только тогда, когда ...	
... не ...	

3.9. Найдите значения логических выражений:

- а)  $(1 \vee 1) \vee (1 \vee 0)$ ;
- б)  $((1 \vee 0) \vee 1) \vee 1$ ;
- в)  $(0 \vee 1) \vee (1 \vee 0)$ ;
- г)  $(0 \& 1) \& 1$ ;
- д)  $1 \& (1 \& 1) \& 1$ ;
- е)  $((1 \vee 0) \& (1 \& 1)) \& (0 \vee 1)$ ;
- ж)  $((1 \& 0) \vee (1 \& 0)) \vee 1$ ;
- з)  $((1 \& 1) \vee 0) \& (0 \vee 1)$ ;
- и)  $((0 \& 0) \vee 0) \& (1 \vee 1)$ .

3.10. Даны два простых высказывания:

$$A = \{2 \cdot 2 = 4\}, B = \{2 \cdot 2 = 5\}.$$

Какие из составных высказываний истинны:

- а)  $\bar{A}$ ;      б)  $\bar{B}$ ;
- в)  $A \& B$ ;    г)  $A \vee B$ ;

д)  $A \Rightarrow B$ ; е)  $A \Leftrightarrow B$ ?

3.11. Даны простые высказывания:

$A = \{5 > 3\}$ ,  $B = \{2 = 3\}$  и  $C = \{4 < 2\}$ .

Определите истинность составных высказываний:

а)  $(A \vee B) \& C \Rightarrow (A \& C) \vee (B \& C)$ ;

б)  $(A \& B) \vee C \Leftrightarrow (A \vee C) \& (A \& B)$ .

3.12. Даны простые высказывания:

$A = \{\text{Принтер} — \text{устройство ввода информации}\}$ ,

$B = \{\text{Процессор} — \text{устройство обработки информации}\}$ ,

$C = \{\text{Монитор} — \text{устройство хранения информации}\}$ ,

$D = \{\text{Клавиатура} — \text{устройство ввода информации}\}$ .

Определите истинность составных высказываний:

а)  $(A \& B) \& (C \vee D)$ ; б)  $(A \& B) \Rightarrow (B \& C)$ ;

в)  $(A \vee B) \Leftrightarrow (C \& D)$ ; г)  $\overline{A} \Leftrightarrow \overline{B}$ .

3.13. Дано составное высказывание **не (не  $A$  и  $B$ )**, где  $A$  и  $B$  — простые высказывания. В каком случае данное высказывание будет ложным?

3.14. Выполните поразрядное логическое сложение двоичных чисел:

а) 100 и 110; б) 1010 и 1000; в) 101010 и 111111.

3.15. Даны два числа  $56_8$  и  $A_{F_{16}}$ . Переведите числа в двоичную систему и выполните операции арифметического сложения и умножения, а также операции поразрядного логического сложения и умножения. Проверьте правильность полученных результатов с помощью калькулятора Wise Calculator.

3.16. Проверьте правильность выполнения заданий 3.6–3.9 с помощью калькулятора NumLock Calculator.

### 3.3. Логические выражения и таблицы истинности

Таблицу, показывающую, какие значения принимает составное высказывание при всех сочетаниях (наборах) значений входящих в него простых высказываний, называют *таблицей истинности* составного высказывания.

Составные высказывания в алгебре логики записываются с помощью логических выражений. Для любого логического выражения достаточно просто построить таблицу истинности.

*Алгоритм построения таблицы истинности:*

1) подсчитать количество переменных  $n$  в логическом выражении;

2) определить число строк в таблице, которое равно  $m = 2^n$ ;

3) подсчитать количество логических операций в логическом выражении и определить количество столбцов в таблице,

которое равно количеству переменных плюс количество операций;

4) ввести названия столбцов таблицы в соответствии с последовательностью выполнения логических операций с учетом скобок и приоритетов;

5) заполнить столбцы входных переменных наборами значений;

6) провести заполнение таблицы истинности по столбцам, выполняя логические операции в соответствии с установленной в п.4 последовательностью.

Наборы входных переменных, во избежание ошибок, рекомендуется перечислять следующим образом:

а) разделить колонку значений первой переменной пополам и заполнить верхнюю часть колонки нулями, а нижнюю единицами;

б) разделить колонку значений второй переменной на четыре части и заполнить каждую четверть чередующимися группами нулей и единиц, начиная с группы нулей;

в) продолжать деление колонок значений последующих переменных на 8, 16 и так далее частей и заполнение их группами нулей или единиц до тех пор, пока группы нулей и единиц не будут состоять из одного символа.

**Пример 3.9.** Для формулы  $A \& (B \vee \bar{B} \& \bar{C})$  построить таблицу истинности алгебраически и с использованием электронных таблиц.

Количество логических переменных 3, следовательно, количество строк в таблице истинности должно быть  $2^3 = 8$ .

Количество логических операций в формуле 5, следовательно количество столбцов в таблице истинности должно быть  $3 + 5 = 8$ .

A	B	C	$\bar{B}$	$\bar{C}$	$\bar{B} \& \bar{C}$	$B \vee (\bar{B} \& \bar{C})$	$A \& (B \vee \bar{B} \& \bar{C})$
0	0	0	1	1	1	1	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	1	0
0	1	1	0	0	0	1	0
1	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0
1	1	0	0	1	0	1	1
1	1	1	0	0	0	1	1

## Построение таблицы истинности логического выражения с использованием электронных таблиц StarOffice Calc

- 1 Запустить интегрированное приложение командой [Программы-StarOffice].
- 2 В появившемся окне приложения StarOffice запустить электронные таблицы Calc командой [Файл-Создать-Документ электронной таблицы].
- 3 Создать заголовки и ввести в столбцы A, B и C значения логических аргументов, а в столбцы D, E, F, G, H соответствующие логические функции. Для ввода логических функций воспользоваться командой [Вставка-Функция].
- 4 После ввода аргументов и формул на листе появится таблица истинности логического выражения.

	A	B	C	D	E	F	G	H
1	A	B	C	Не В	Не С	Не В и Не С	В или (Не В и Не С)	А и (В или (Не В и Не С))
2	0	0	0	TRUE	TRUE	TRUE	TRUE	FALSE
3	0	0	1	TRUE	FALSE	FALSE	FALSE	FALSE
4	0	1	0	FALSE	TRUE	FALSE	TRUE	FALSE
5	0	1	1	FALSE	FALSE	FALSE	TRUE	FALSE
6	1	0	0	TRUE	TRUE	TRUE	TRUE	TRUE
7	1	0	1	TRUE	FALSE	FALSE	FALSE	FALSE
8	1	1	0	FALSE	TRUE	FALSE	TRUE	TRUE
9	1	1	1	FALSE	FALSE	FALSE	TRUE	TRUE
10								

- 5 Переименовать лист *Лист1* в *Таблицу истинности* и сохранить в файле log.sdc.

Файл log.sdc хранится  
в каталоге \practicum\logical\

CD-ROM

### Задания для самостоятельного выполнения

- 3.17. Построить таблицы истинности для следующих формул:
  - a)  $A \vee (B \vee \bar{B} \Rightarrow \bar{C})$ ;
  - б)  $A \& (B \& \bar{B} \Rightarrow \bar{C})$ ;
  - в)  $A \vee (B \vee \bar{B}) \& A \vee (B \Rightarrow C)$ .
- 3.18. Выбрать составное высказывание, имеющее ту же таблицу истинности, что и не (не A и не(B и C)).
  - 1) A и B или C и A;
  - 2) (A или B) и (A или C);
  - 3) A и (B или C);
  - 4) A или (не B или не C).

3.19. Докажите с помощью таблиц истинности равносильность следующих логических выражений:

- а)  $(A \Rightarrow B) \& (A \vee \bar{B})$ ;
- б)  $(A \Leftrightarrow B) \& (A \& B) \vee (\bar{A} \& \bar{B})$ .

### 3.4. Логические функции

*Логической функцией* называют функцию  $F(X_1, X_2, \dots, X_n)$ , аргументы которой  $X_1, X_2, \dots, X_n$  (логические переменные) и сама функция (логическая переменная) принимают значения 0 или 1.

Таблицу, показывающую, какие значения принимает логическая функция при всех сочетаниях значений ее аргументов, называют таблицей истинности логической функции. Таблица истинности логической функции  $n$  аргументов содержит  $2^n$  строк,  $n$  столбцов значений аргументов и 1 столбец значений функции.

Логические функции могут быть заданы табличным способом или аналитически — в виде соответствующих формул.

Существуют 16 различных логических функций от двух переменных.

Таблица 3.1

Логические функции двух переменных

Аргументы		Логические функции															
		$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$	$F_{13}$	$F_{14}$	$F_{15}$	$F_{16}$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Если логическая функция представлена с помощью базовых логических функций (дизъюнкции, конъюнкции и инверсии), то такая форма представления называется *нормальной*.

**Пример 3.10.** По имеющимся таблицам истинности выразите через базовые логические функции (конъюнкцию, дизъюнкцию и отрицание) следующие функции:

- а)  $F_9(X, Y)$ ;
- б)  $F_{15}(X, Y)$ .

Из таблицы истинности видно, что  $F_9(X, Y) = \overline{F_8(X, Y)}$  (отрицание дизъюнкции).

Из таблицы истинности видно, что  $F_{15}(X, Y) = \overline{F_2(X, Y)}$  (отрицание конъюнкции).



### Задания для самостоятельного выполнения

- 3.20. По имеющимся таблицам истинности выразите через базовые логические функции (конъюнкцию, дизъюнкцию и отрицание) следующие функции:
- а)  $F_3(X, Y)$ ; б)  $F_5(X, Y)$ ; в)  $F_7(X, Y)$ ; г)  $F_{10}(X, Y)$ ;  
 д)  $F_{11}(X, Y)$ ; е)  $F_{12}(X, Y)$ ; ж)  $F_{13}(X, Y)$ ; з)  $F_{14}(X, Y)$ .
- 3.21. С помощью электронных таблиц построить таблицу истинности для всех возможных логических функций двух переменных.

## 3.5. Логические законы и правила преобразования логических выражений

Логические выражения называются *равносильными*, если их истинностные значения совпадают при любых значениях входящих в них логических переменных.

В алгебре логики имеется ряд законов, позволяющих производить равносильные преобразования логических выражений. Приведем соотношения, отражающие эти законы.

### 1. Закон двойного отрицания:

$$A = \overline{\overline{A}}.$$

Двойное отрицание исключает отрицание.

### 2. Переместительный (коммутативный) закон:

— для логического сложения:

$$A \vee B = B \vee A;$$

— для логического умножения:

$$A \& B = B \& A.$$

Результат операции над высказываниями не зависит от того, в каком порядке берутся эти высказывания.

В обычной алгебре  $a + b = b + a$ ,  $a \times b = b \times a$ .

### 3. Сочетательный (ассоциативный) закон:

— для логического сложения:

$$(A \vee B) \vee C = A \vee (B \vee C);$$

— для логического умножения:

$$(A \& B) \& C = A \& (B \& C).$$

При одинаковых знаках скобки можно ставить произвольно или вообще опускать.

В обычной алгебре  $(a + b) + c = a + (b + c) = a + b + c$ ,  
 $(a \times b) \times c = a \times (b \times c) = a \times b \times c$ .

### 4. Распределительный (дистрибутивный) закон:

— для логического сложения:

$$(A \vee B) \vee C = (A \& C) \vee (B \& C);$$

— для логического умножения:

$$(A \& B) \vee C = (A \vee C) \& (B \vee C).$$

Определяет правило выноса общего высказывания за скобку.

В обычной алгебре справедлив распределительный закон только для сложения:  $(a + b) \times c = a \times c + b \times c$ .

**5. Закон общей инверсии (законы де Моргана):**

— для логического сложения

$$\overline{A \vee B} = \overline{A} \& \overline{B};$$

— для логического умножения:

$$\overline{A \& B} = \overline{A} \vee \overline{B}.$$

**6. Закон идемпотентности** (от латинских слов *idem* — тот же самый и *potens* — сильный; дословно — равносильный):

— для логического сложения:

$$A \vee A = A;$$

— для логического умножения:

$$A \& A = A.$$

Закон означает отсутствие показателей степени.

**7. Законы исключения констант:**

— для логического сложения:

$$A \vee 1 = 1, \quad A \vee 0 = A;$$

— для логического умножения:

$$A \& 1 = A, \quad A \& 0 = 0.$$

**8. Закон противоречия:**

$$A \& \overline{A} = 0.$$

Невозможно, чтобы противоречащие высказывания были одновременно истинными.

**9. Закон исключения третьего:**

$$A \vee \overline{A} = 1.$$

Из двух противоречащих высказываний об одном и том же предмете одно всегда истинно, а второе — ложно, третьего не дано.

**10. Закон поглощения:**

— для логического сложения:

$$A \vee (A \& B) = A;$$

— для логического умножения:

$$A \& (A \vee B) = A.$$

**11. Закон исключения (склеивания):**

— для логического сложения:

$$(A \& B) \vee (\bar{A} \& B) = B;$$

— для логического умножения:

$$(A \vee B) \& (\bar{A} \vee B) = B.$$

**12. Закон контрапозиции (правило перевертывания):**

$$(A \Leftrightarrow B) = (B \Leftrightarrow A).$$

Справедливость приведенных законов можно доказать табличным способом: выписать все наборы значений  $A$  и  $B$ , вычислить на них значения левой и правой частей доказываемого выражения и убедиться, что результирующие столбцы совпадут.

**Пример 3.11.** Найдите  $X$ , если  $\bar{X} \vee A \vee X \vee \bar{A} = B$ .

Для преобразования левой части равенства последовательно воспользуемся законом де Моргана для логического сложения и законом двойного отрицания:

$$(\bar{X} \& \bar{A}) \vee (X \& A).$$

Согласно распределительному закону для логического сложения:

$$\bar{X} \& (\bar{A} \vee A).$$

Согласно закону исключения третьего и закона исключения констант:

$$\bar{X} \& 1 = \bar{X}.$$

Полученную левую часть приравняем правой:

$$\bar{X} = B.$$

Окончательно получим:  $X = \bar{B}$ .**Пример 3.12.** Упростите логическое выражение

$$(A \vee B \vee C) \& A \vee \bar{B} \vee C.$$

Правильность упрощения проверьте с помощью таблиц истинности для исходного и полученного логического выражения.

Согласно закону общей инверсии для логического сложения (первому закону де Моргана) и закону двойного отрицания:

$$(A \vee B \vee C) \& (A \vee \bar{B} \vee C) = (A \vee B \vee C) \& (\bar{A} \& B \& \bar{C}).$$

Согласно распределительному (дистрибутивному) закону для логического сложения:

$$(A \vee B \vee C) \& (\bar{A} \& B \& \bar{C}) = (A \& \bar{A}) \vee (B \& \bar{A}) \vee (C \& \bar{A}) \vee (A \& B) \vee (B \& B) \vee (C \& B) \vee (A \& \bar{C}) \vee (B \& \bar{C}) \vee (C \& \bar{C}).$$

Согласно закону противоречия:

$$(A \& \bar{A}) = 0; (C \& \bar{C}) = 0.$$

Согласно закону идемпотентности

$$(B \& B) = B.$$

Подставляем значения и, используя переместительный (коммутативный) закон и группируя слагаемые, получаем:

$$0 \vee (A \& B) \vee (\bar{A} \& B) \vee B \vee (C \& B) \vee (\bar{C} \& B) \vee (C \& \bar{A}) \vee (A \& \bar{C}) \vee 0.$$

Согласно закону исключения (склеивания)

$$(A \& B) \vee (\bar{A} \& B) = B, \\ (C \& B) \vee (\bar{C} \& B) = B.$$

Подставляем значения и получаем:

$$0 \vee B \vee B \vee B \vee (C \& \bar{A}) \vee (A \& \bar{C}) \vee 0.$$

Согласно закону исключения констант для логического сложения и закону идемпотентности:

$$0 \vee B \vee 0 \vee B \vee B = B.$$

Подставляем значения и получаем:

$$B \vee (C \& \bar{A}) \vee (A \& \bar{C}).$$

Согласно распределительному (дистрибутивному) закону для логического умножения:

$$(C \& \bar{A}) \vee (A \& \bar{C}) = (C \vee A) \& (C \vee \bar{C}) \& (\bar{A} \vee A) \& (\bar{A} \vee \bar{C}).$$

Согласно закону исключения третьего:

$$(C \vee \bar{C}) = 1, \\ (A \vee \bar{A}) = 1.$$

Подставляем значения и окончательно получаем:

$$B \& \bar{A} \& \bar{C}.$$



### Проверка правильности преобразования логического выражения с использованием электронных таблиц StarOffice Calc

- 1 Запустить интегрированное приложение командой [Программы-StarOffice].
- 2 В появившемся окне приложения StarOffice запустить электронные таблицы Calc командой [Файл-Создать-Документ электронной таблицы].
- 3 Построить таблицы истинности для исходного и конечного логических выражений.
- 4 После ввода аргументов и формул на листе появится таблица истинности логического выражения. Сравнить последние столбцы в таблицах между собой.

StarOffice 5.2 (log1.sdc)

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Имя: //X:\Мои документы\log1.sdc

URL:

Адрес: 10

	A	B	C	D	E	F	G	H
1	A	B	C	Не В	А или В или С	А или Не В или С	Не (А или Не В или С)	(А или В или С) и (Не А или Не В или С)
2	0	0	0	TRUE	FALSE	TRUE	FALSE	FALSE
3	0	0	1	TRUE	TRUE	TRUE	FALSE	FALSE
4	0	1	0	FALSE	TRUE	TRUE	TRUE	TRUE
5	0	1	1	FALSE	TRUE	TRUE	FALSE	FALSE
6	1	0	0	TRUE	TRUE	TRUE	FALSE	FALSE
7	1	0	1	TRUE	TRUE	TRUE	FALSE	FALSE
8	1	1	0	FALSE	TRUE	TRUE	FALSE	FALSE
9	1	1	1	FALSE	TRUE	TRUE	FALSE	FALSE
10								
11	A	B	C	Не А	Не С	В и Не А и Не С		
12	0	0	0	TRUE	TRUE	FALSE		
13	0	0	1	TRUE	FALSE	FALSE		
14	0	1	0	TRUE	TRUE	TRUE		
15	0	1	1	TRUE	FALSE	FALSE		
16	1	0	0	FALSE	TRUE	FALSE		
17	1	0	1	FALSE	FALSE	FALSE		
18	1	1	0	FALSE	TRUE	FALSE		
19	1	1	1	FALSE	FALSE	FALSE		
20								

Лист 4 / 4      Стандарт      100%      Стан      Сумма=0

Лист log1.sdc      15:40

- 5 Переименовать лист *Лист1* в *Преобразование* и сохранить в файле *log.sdc*.

Файл *log.sdc* хранится в каталоге *\practicum\logical*

CD-ROM 

### Задания для самостоятельного выполнения

- 3.22. Какое тождество записано неверно:
- $X \vee \bar{X} = 1$ ;
  - $X \vee X \vee X \vee X \vee X \vee X = 1$ ;
  - $X \& X \& X \& X \& X = X$ ?
- 3.23. Определите, каким законам алгебры чисел (сочетательному; переместительному; распределительному; аналога нет) соответствуют следующие логические тождества:
- $A \vee B = B \vee A$ ;
  - $(A \& B) \& C = A \& (B \& C)$ ;
  - $A \vee (B \& C) = (A \vee B) \& (A \vee C)$ ;
  - $(A \vee B) \& C = (A \& C) \vee (B \& C)$ .
- 3.24. Логическое выражение называется *тождественно-ложным*, если оно принимает значения 0 на всех наборах входящих в него простых высказываний. Упростите следующее выражение и покажите, что оно тождественно-ложное.
- $$(A \& B \& \bar{B}) \vee (A \& \bar{A}) \vee (B \& C \& \bar{C}).$$

3.25. Логическое выражение называется *тождественно-истинным*, если оно принимает значения 1 на всех наборах входящих в него простых высказываний. Упростите следующее выражение и покажите, что оно тождественно-истинное.

$$(A \& B \& \bar{C}) \vee (A \& B \& C) \vee \overline{A \vee B}.$$

3.26. Упростите логические выражения. Правильность упрощения проверьте с помощью таблиц истинности для исходных и полученных логических формул.

а)  $A \vee (\bar{A} \& B)$ ;

б)  $A \& (\bar{A} \vee B)$ ;

в)  $(A \vee B) \& (\bar{B} \vee A) \& (\bar{C} \vee B)$ ;

### 3.6. Логические основы компьютера

Дискретный преобразователь, который после обработки входных двоичных сигналов выдаёт на выходе сигнал, являющийся значением одной из логических операций, называется *логическим элементом*.

Ниже приведены условные обозначения (схемы) базовых логических элементов, реализующих логическое умножение (конъюнктор), логическое сложение (дизъюнктор) и отрицание (инвертор).

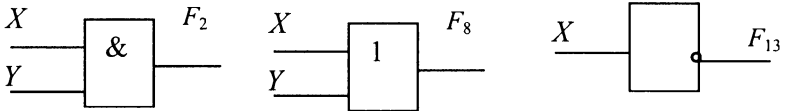
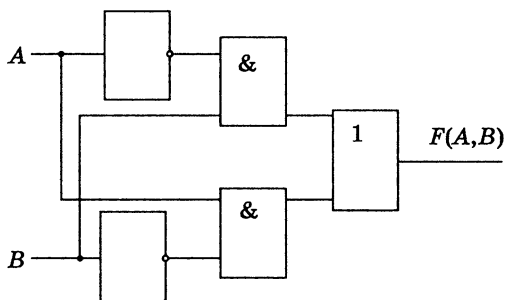


Рис. 3.1. Конъюнктор, дизъюнктор и инвертор

Устройства компьютера (сумматоры в процессоре, ячейки памяти в оперативной памяти и др.) строятся на основе базовых логических элементов.

**Пример 3.13.** По заданной логической функции  $F(A, B) = B \& A \vee \bar{B} \& A$  построить логическую схему.

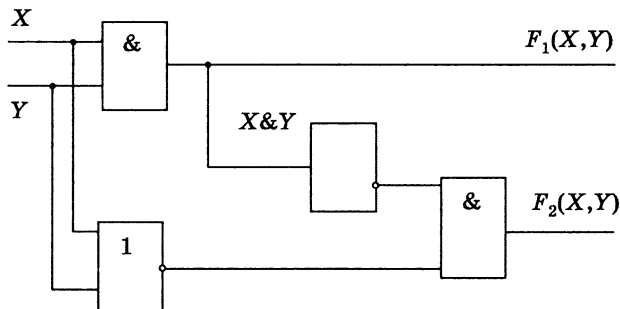
Построение необходимо начинать с логической операции, которая должна выполняться последней. В данном случае такой операцией является логическое сложение, следовательно, на выходе логической схемы должен быть дизъюнктор. На него сигналы подаются с двух конъюнкторов, на которые в свою очередь подаются один входной сигнал нормальный и один инвертированный (с инверторов).



**Пример 3.14.** Логическая схема имеет два входа  $X$  и  $Y$ . Определить логические функции  $F_1(X,Y)$  и  $F_2(X,Y)$ , которые реализуются на ее двух выходах.

Функция  $F_1(X,Y)$  реализуется на выходе первого конъюнктора, то есть  $F_1(X,Y) = X \& Y$ .

Одновременно сигнал с конъюнктора подается на вход инвертора, на выходе которого реализуется сигнал  $X \vee Y$ , который, в свою очередь, подается на один из входов второго конъюнктора.



На другой вход второго конъюнктора подается сигнал  $X \vee Y$  с дизъюнктора, следовательно, функция  $F_2(X,Y) = X \& Y \& (X \vee Y)$ .

Рассмотрим схему сложения двух  $n$ -разрядных двоичных чисел.

$$\begin{array}{r}
 a_n \dots \boxed{a_i} \dots a_1 a_0 \\
 + \quad b_n \dots \boxed{b_i} \dots b_1 b_0 \\
 \hline
 s_{n+1} s_n \dots \boxed{s_i} \dots s_1 s_0
 \end{array}$$

При сложении цифр  $i$ -го разряда складываются  $a_i$  и  $b_i$ , а также  $p_{i-1}$  — перенос из  $i-1$  разряда. Результатом будет  $s_i$  — сумма и  $p_i$  — перенос в старший разряд.

Таким образом, одноразрядный двоичный сумматор — это устройство с тремя входами и двумя выходами.

**Пример 3.15.** Построить таблицу истинности одноразрядного двоичного сумматора, воспользовавшись таблицей сложения двоичных чисел.

Входы			Выходы	
$A_i$	$B_i$	$P_{i-1}$	$S_i$	$P_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

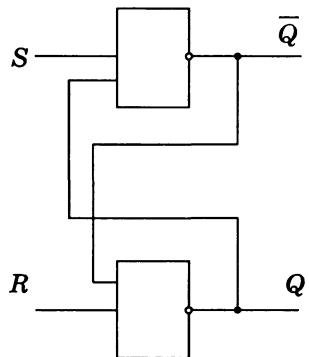
**Триггер.** Для хранения информации в оперативной памяти компьютера, а также во внутренних регистрах процессора используются триггеры. Триггер может находиться в одном из двух устойчивых состояний, что позволяет запоминать, хранить и считывать 1 бит информации.

Самый простой триггер —  $RS$ -триггер. Он состоит из двух логических элементов ИЛИ-НЕ, которые реализуют логическую функцию  $F_9$  (смотри таблицу 3.1).

Входы и выходы элементов соединены кольцом: выход первого соединен со входом второго и выход второго — со входом первого.

Триггер имеет два входа  $S$  (от англ. set — установка) и  $R$  (от англ. reset — сброс) и два выхода  $Q$  (прямой) и  $\bar{Q}$  (инверсный).

**Рис. 3.2**  
Логическая схема  $RS$ -триггера





**Пример 3.16.** Построить таблицу, описывающую состояние входов и выходов  $RS$ -триггера.

Если на входы поступают сигналы  $R = 0$  и  $S = 0$ , то триггер находится в режиме хранения, на выходах  $Q$  и  $\bar{Q}$  сохраняются установленные ранее значения.

Если на установочный вход  $S$  поступает на короткое время сигнал 1, то триггер переходит в состояние 1 и после того, как сигнал на входе  $S$  станет равен 0, триггер будет сохранять это состояние, то есть будет хранить 1.

При подаче 1 на вход  $R$  триггер перейдет в состояние 0.

Подача на оба входа  $S$  и  $R$  логической единицы может привести к неоднозначному результату, поэтому такая комбинация входных сигналов запрещена.

Входы		Состояние $Q$
$S$	$R$	
0	0	$Q$
1	0	1
0	1	0
1	1	Недопустимо

### Задания для самостоятельного выполнения

- 3.27. Существуют 16 логических функций от двух переменных (смотри таблицу 3.1). Постройте их логические схемы с помощью базовых логических элементов: конъюнктора, дизъюнктора и инвертора.
- 3.28. Доказать, что рассмотренная в примере 3.10 логическая схема является одноразрядным двоичным полусумматором (не учитывается перенос из младшего разряда).
- 3.29. Доказать, построив таблицу истинности, что логическая функция  $P = (A \& B) \vee (A \& P_0) \vee (B \& P_0)$  определяет перенос в старший разряд при сложении двоичных чисел ( $A$  и  $B$  — слагаемые,  $P_0$  — перенос из младшего разряда).
- 3.30. Доказать, построив таблицу истинности, что логическая функция  $S = (A \vee B \vee P_0) \& P \vee (A \& B \& P_0)$  определяет сумму при сложении двоичных чисел ( $A$  и  $B$  — слагаемые,  $P_0$  — перенос из младшего разряда).
- 3.31. Построить логическую схему одноразрядного двоичного сумматора.
- 3.32. Какое количество базовых логических элементов необходимо для реализации 64-разрядного сумматора двоичных чисел?
- 3.33. Какое количество базовых логических элементов образуют оперативную память современного компьютера объемом 64 Мбайта?

В процессе изучения данной темы рекомендуется установить свободно распространяемое корпорацией Sun полностью русифицированное интегрированное офисное приложение StarOffice 5.2, которое включает:

CD-ROM 

- текстовый редактор StarOffice Writer;
- растровый графический редактор StarOffice Image;
- векторный графический редактор StarOffice Draw;
- инструментальное средство для разработки мультимедийных презентаций StarOffice Impress и проигрыватель презентаций StarOffice Player;
- электронные таблицы StarOffice Calc;
- систему управления базами данных StarOffice Base.

Для создания точных чертежей и геометрических построений рекомендуется использовать систему автоматизированного проектирования КОМПАС-3D, разработанную российской компанией АСКОН, которая предоставляет возможность бесплатного использования этой системы в учебных заведениях.

### 4.1. Создание текстовых документов с использованием StarOffice Writer

StarOffice Writer — приложение для создания текстовых документов. StarOffice Writer является средством создания, редактирования и форматирования текстовых документов, входящим в состав интегрированного офисного пакета StarOffice и аналогом широко известного текстового редактора Microsoft Word. Writer полностью совместим с Word по формату файлов (поддерживает формат DOC), по функциональным возможностям (поддерживает вставку рисунков, диаграмм, гиперссылок, сохранение документа в формате HTML и так далее), а также по графическому интерфейсу (набор команд меню этих приложений практически одинаков).

**Создание шаблонов документов.** Writer имеет обширную библиотеку шаблонов для создания текстовых документов различного назначения (факсов, рефератов, визитных карточек и так далее) и позволяет создавать собственные шаблоны документов с помощью мастера (*Автопилота*).

В современном документообороте широко используются факсы, то есть документы, отправляемые и получаемые по коммутируемым телефонным линиям с помощью специальных факсимильных аппаратов. Современные модемы (иногда их называют факс-модемами) также позволяют отправлять и принимать факсы непосредственно с компьютера.

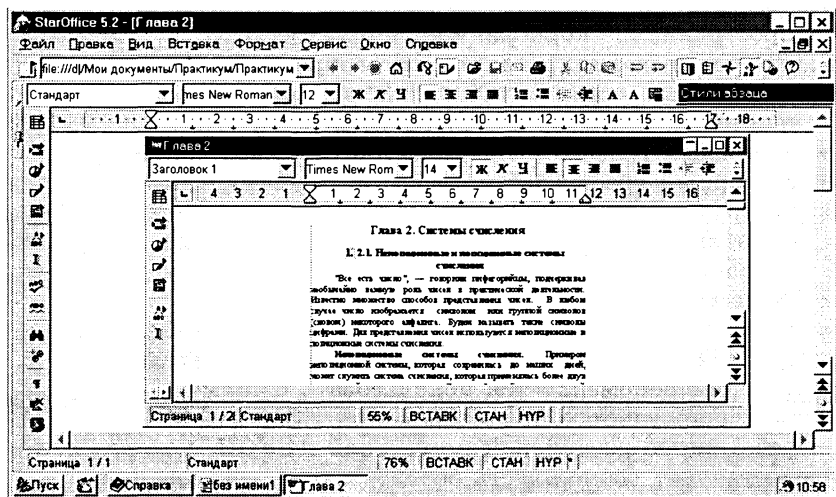


Рис. 4.1. Окно приложения StarOffice Writer

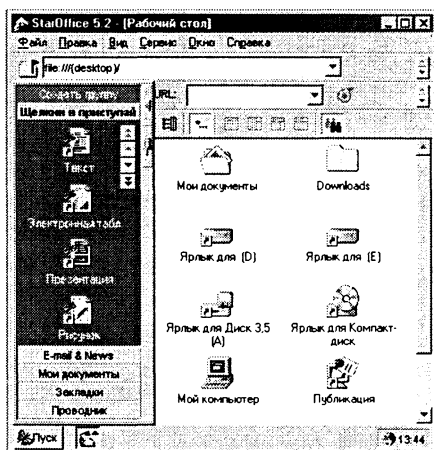
**Практическое задание 4.1.** «Создание шаблона факса с использованием *Автопилота*». На основе существующего шаблона факса создать собственный шаблон, в который внести данные отправителя (организация, фамилия, номер факса и др.), что упростит и ускорит отправку факсов.



### Создание шаблона факса с использованием *Автопилота*

1 Запустить интегрированное офисное приложение StarOffice 5.2 командой [Пуск-StarOffice 5.2].

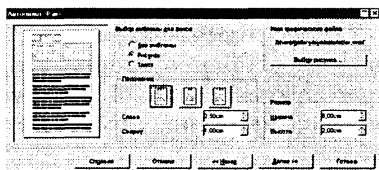
Появится стартовое окно интегрированного приложения *Рабочий стол*. Для запуска текстового редактора StarOffice Writer щелкнуть по ярлыку *Текст*.



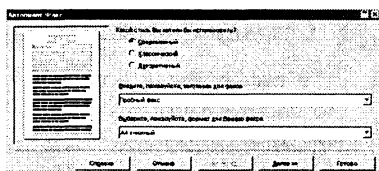
Создание шаблона для документов определенного типа производится с помощью *Автопилота* путем внесения необходимых данных в последовательно появляющиеся панели.

2 Для создания документа определенного типа (например, факса) с помощью мастера (*Автопилота*) необходимо ввести команду [Файл-Автопилот-Факс...].

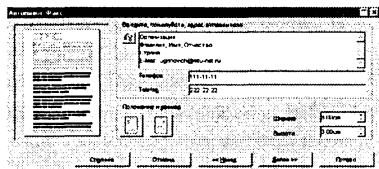
На первой панели *Автопилот Факс* необходимо выбрать стиль факса, его заголовок и формат. Щелкнуть по кнопке *Далее*.



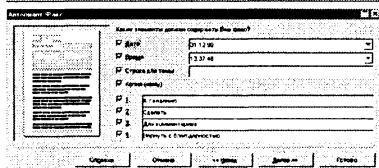
На второй панели *Автопилот Факс* необходимо выбрать файл рисунка и положение рисунка на факсе. Щелкнуть по кнопке *Далее*.



На третьей панели *Автопилот Факс* необходимо внести данные об отправителе факса (организация, фамилия, e-mail, номер факса и др.). Щелкнуть по кнопке *Далее*.

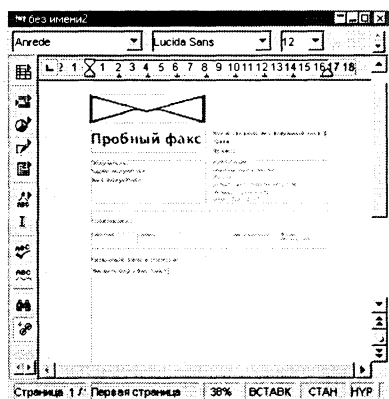


На пятой панели *Автопилот Факс* необходимо определить (установив флажки) перечень полей, которые будут на факсе. Щелкнуть по кнопке *Далее*.



Пройдя еще несколько панелей, щелкнуть по кнопке *Готово*. Будет создан новый шаблон факсов.

Теперь для подготовки факса достаточно открыть созданный шаблон и внести данные получателя, тему и текст сообщения.



### 3 Сохранить шаблон факса в файле fax.vog.

**Шаблон факса хранится в каталоге**  
 \practicum\inftech\text\

CD-ROM 

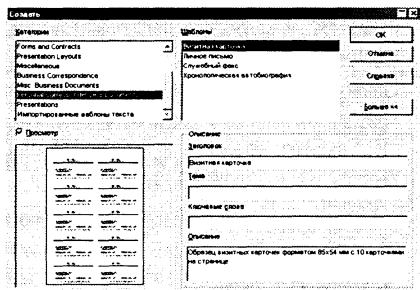
**Создание документов на основе шаблонов.** На основе существующего или созданного шаблона можно создавать документы.

**Практическое задание 4.2. «Создание визитных карточек на основе шаблона *Визитная карточка*».** Создать на основе существующего шаблона *Визитная карточка* документ, содержащий на стандартном листе А4 десять ваших визитных карточек.



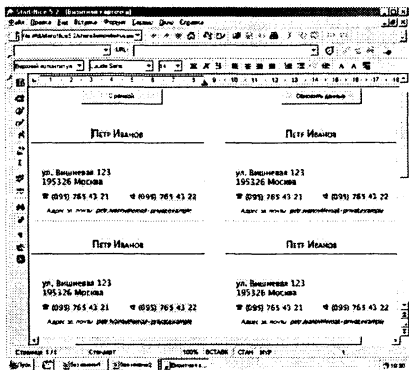
### Создание визитных карточек на основе шаблона

- 1 Для создания нового документа на основе шаблона необходимо ввести команду [Файл-Создать-Из шаблонов...].
- 2 На появившейся панели *Создать* выбрать в окне *Категории*: набор шаблонов (*Personal Correspondence&Documents*), а в окне *Шаблоны* выбрать конкретный шаблон (*Визитная карточка*).

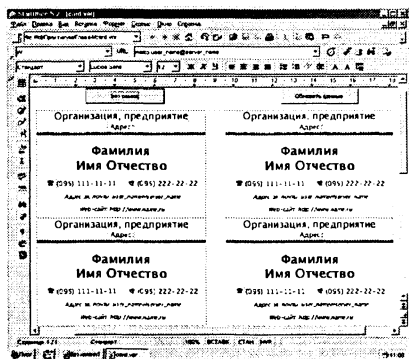


3 В окне приложения откроется шаблон, содержащий 10 экземпляров визитных карточек.

Внести в левую верхнюю карточку собственные данные и щелкнуть по кнопке *Обновить данные*. Новые данные будут внесены во все карточки.



4 Созданный документ можно сохранить в форме шаблона (расширение VOR), либо в форме документа в формате Writer (расширение SDW) или в формате Word 97/2000 (расширение DOC).



5 Сохранить шаблон ваших визитных карточек в файле card.vor.

**Шаблон визитных карточек хранится  
в каталоге \practicum\inftech\textl.**

**CD-ROM**



**Открытие, редактирование и форматирование документов.** Редактор Writer позволяет открывать файлы различных форматов: собственного формата StarWriter (SDW), формата Microsoft Word 95 и 97/2000 (DOC), текста без форматирования из операционных систем Windows, MS-DOS, Unix и Mac Os (TXT), универсального формата Rich Text Format (RTF), формата Web-страниц (HTML) и др.

**Практическое задание 4.3. «Открытие, редактирование и форматирование текстового документа».** Открыть текстовый файл text.doc и отформатировать его в соответствии с указаниями, содержащимися в тексте файла.

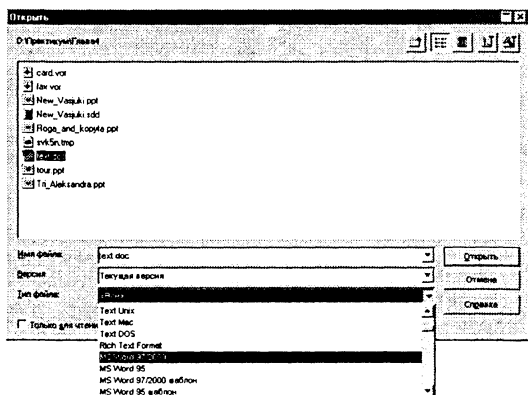


## Открытие, редактирование и форматирование текстового документа

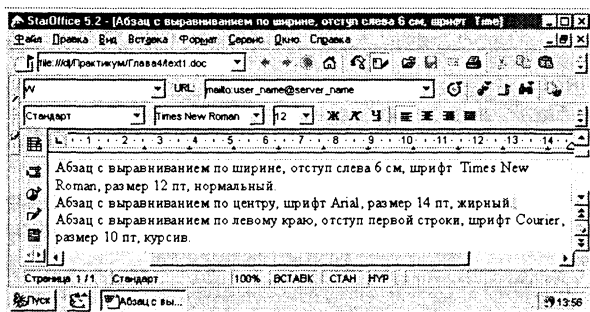
- 1 Ввести команду [Файл-Открыть...].
- 2 На появившейся панели перейти в каталог CD-ROM \practicum\inftech\text\.

В раскрывающемся списке *Тип файла*: выбрать формат *MS Word 97/2000*, а в окне выбрать файл.

Щелкнуть по кнопке *Открыть*.

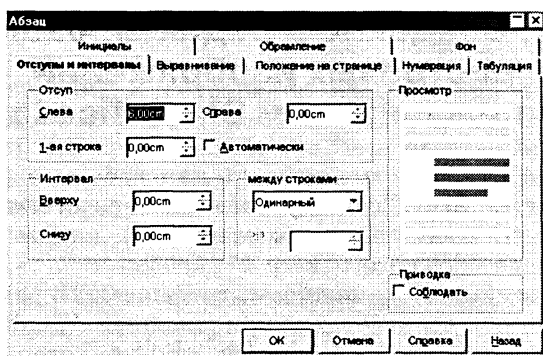


- 3 В окне приложения откроется неотформатированный документ с текстом задания.



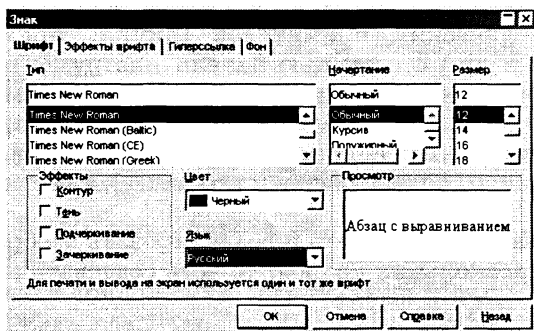
Форматирование текста документа производится с помощью диалоговых панелей *Абзац* и *Знак*.

- 4 Ввести команду [Формат-Абзац...].  
На появившейся диалоговой панели *Абзац* на вкладках *Отступы и интервалы* и *Выравнивание* установить требуемые параметры форматирования.



## 5 Ввести команду [Формат-Знак...].

На появившейся диалоговой панели *Знак* на вкладке *Шрифт* установить требуемые параметры форматирования.



## 6 Сохранить отформатированный документ в файле text.sdw.

**Отформатированный документ хранится  
в каталоге \practicum\inftech\text\**

**CD-ROM** 

### Задания для самостоятельного выполнения

- 4.1. Создать с помощью *Автопилота* собственный шаблон факса и отправить факс.
- 4.2. Создать на основе шаблона *Расписание занятий* ваше расписание уроков.
- 4.3. Создать на основе шаблона *Кассета* вкладыш для кассеты, на основе шаблона *Компакт-диски* внутренний и внешний вкладыши для дисков и на основе шаблона *Наклейка для дискет* наклейку для гибких дисков.
- 4.4. Открыть файл text1.doc, который находится на CD-ROM в каталоге \practicum\inftech\text\, и отформатировать его в соответствии с указаниями, содержащимися в тексте документа.



## 4.2. Создание и редактирование растровых изображений с использованием StarOffice Image

StarOffice Image — приложение для работы с растровыми изображениями. StarOffice Image является средством создания и редактирования растровых изображений, входящим в состав интегрированного офисного пакета StarOffice.

Image объединяет возможности графического редактора Paint, являющегося стандартным приложением Windows, и программы обработки изображений PhotoEditor, которая входит в состав Microsoft Office. Image позволяет экспортировать в растровом формате рисунки созданные в векторном редакторе StarOffice Draw.

Image позволяет создавать рисунки с использованием различных графических примитивов (линия, прямоугольник, окружность и др.) с помощью панели инструментов.

Растровые рисунки можно изменять с помощью большого количества эффектов (набросок углем, мозаика, рельеф и так далее), выполнять над ними различные действия (обрезать, изменять размер, поворачивать и так далее), изменять масштаб и др.

Image позволяет создавать и редактировать изображения, состоящие из нескольких кадров, то есть создавать анимированные изображения в формате GIF.

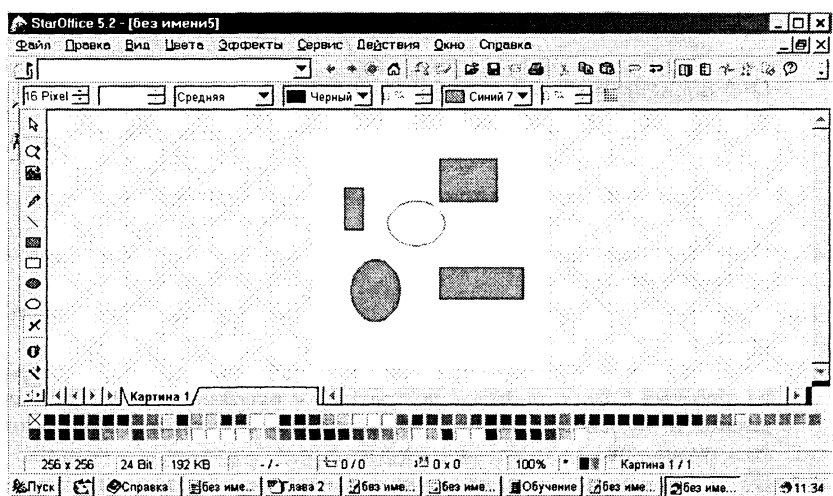


Рис. 4.2. Окно приложения StarOffice Image

Image может открывать и сохранять изображения в различных графических форматах: GIF, JPG, PNG, BMP, TIF и др.

**Редактирование и сохранение изображений в различных форматах.** Довольно часто приходится (например, при подготовке описаний для приложений) получать изображения с экрана монитора. Полученные изображения обычно подвергаются редактированию и сохраняются в различных графических форматах: BMP — растровое изображение без сжатия, TIF — растровое изображение, используемое в различных издательских системах, GIF, JPG, PNG — сжатые растровые изображения, используемые для размещения графики на Web-страницах.

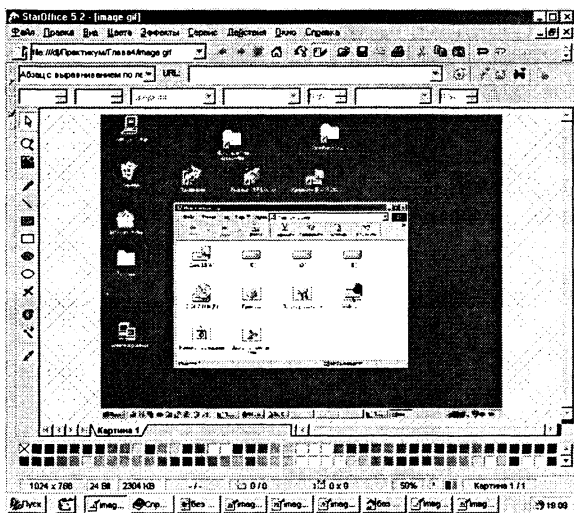
**Практическое задание 4.4. «Получение изображения значка Windows и его сохранение в различных графических форматах».** Рассмотрим процесс получения изображения значка CD-ROM и его сохранения в различных графических форматах.

#### **Получение, редактирование и сохранение растрового изображения**

- 1 Запустить интегрированный пакет StarOffice 5.2 и ввести команду [Файл-Создать-Картинка]. Появится окно растрового графического редактора StarOffice Image.

Получим изображение экрана монитора и поместим его в приложение Image.

- 2 Свернуть все приложения, щелкнуть по значку *Мой компьютер* и нажать клавишу {Print Screen}. Изображение экрана монитора будет помещено в *Буфер обмена Windows*.

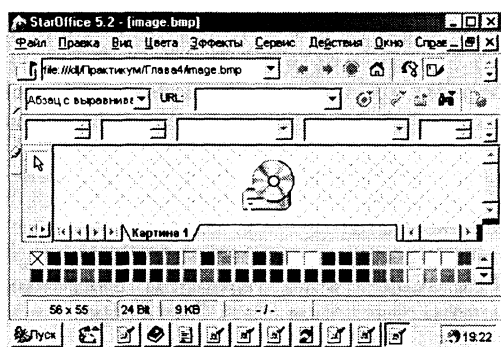


- 3 Активизировать приложение Image и ввести команду [Правка-Вставить]. В окне приложения будет помещено изображение экрана компьютера.

Вырежем из изображения экрана монитора изображение значка.

- 4 На панели инструментов выбрать инструмент выделения (кнопка со стрелкой) и нарисовать прямоугольник вокруг значка.

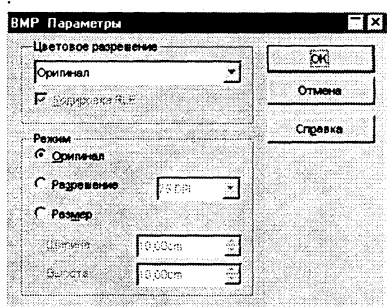
Ввести команду [Действия-Обрезать]. На поле рисования останется изображение значка.



Сохраним это изображение в различных графических форматах: BMP, GIF, JPEG, PNG и TIFF, установив для каждого формата запрашиваемые параметры сохранения.

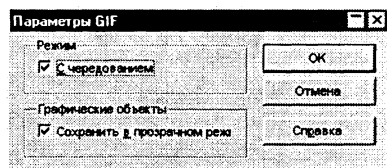
- 5 Ввести команду [Файл-Сохранить как...]. На появившейся панели *Сохранить* выбрать формат *BMP – MS Windows* и ввести имя файла *image.bmp*.

На появившейся панели *BMP Параметры* установить глубину цвета, разрешение и размер.



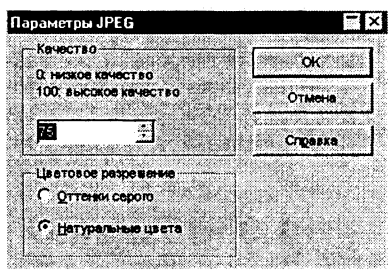
- 6 Ввести команду [Файл-Сохранить как...], выбрать формат *GIF – Graphics Interchange* и имя *image.gif*.

На панели *Параметры GIF* установить режимы вывода изображения (чередование строк и прозрачность).



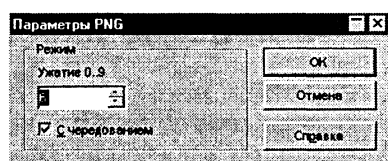
- 7 Открыть файл изображения image.bmp. Ввести команду [Файл-Сохранить как...], выбрать формат *JPG – JPEG* и имя image.jpg.

На панели *Параметры JPEG* установить степень сжатия и цветовую палитру.



- 8 Открыть файл изображения image.bmp. Ввести команду [Файл-Сохранить как...], выбрать формат *PNG – Portable Network Graphics* и имя image.png.

На панели *Параметры PNG* установить степень сжатия и режим чередования строк.



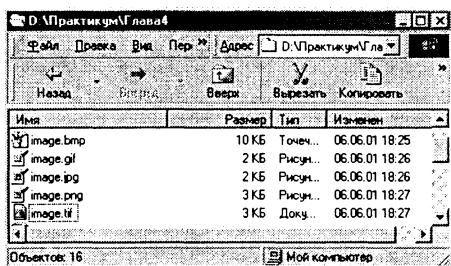
- 9 Открыть файл изображения image.bmp. Ввести команду [Файл-Сохранить как...], выбрать формат *TIF – Tag Image Files* и имя image.tif.

Можно сравнить качество изображений в полученных графических файлах различных форматов и информационный объем файлов.

- 10 Последовательно просмотреть файлы различных графических форматов с использованием пунктов меню *Окно*. Использовать инструмент *Луна* для рассмотрения деталей изображений.

- 11 Открыть папку, где хранятся файлы изображений в различных форматах и сравнить их объемы.

Самый большой объем имеет файл в формате BMP, а самый маленький — файлы в форматах GIF и JPG.



### Задания для самостоятельного выполнения

- 4.5. Получить изображение значка *Мой компьютер* и сохранить его в различных графических форматах.
- 4.6. Создать произвольный рисунок с использованием панелей *Инструменты* и *Палитра цветов*.

## 4.3. Создание и редактирование векторных изображений с использованием StarOffice Draw

StarOffice Draw — приложение для работы с векторными изображениями. StarOffice Draw является средством создания и редактирования векторных изображений, входящим в состав интегрированного офисного пакета StarOffice.

StarOffice Draw позволяет рисовать прямые линии, различные типы кривых, прямоугольники, окружности и другие фигуры. Инструменты 3D-графики предназначены для рисования трехмерных геометрических тел, таких как параллелепипеды, конусы, цилиндры, торы и др.

StarOffice Draw позволяет создать различные управляющие элементы (кнопки, текстовые поля, переключатели, флажки и др.) для Web-страниц.

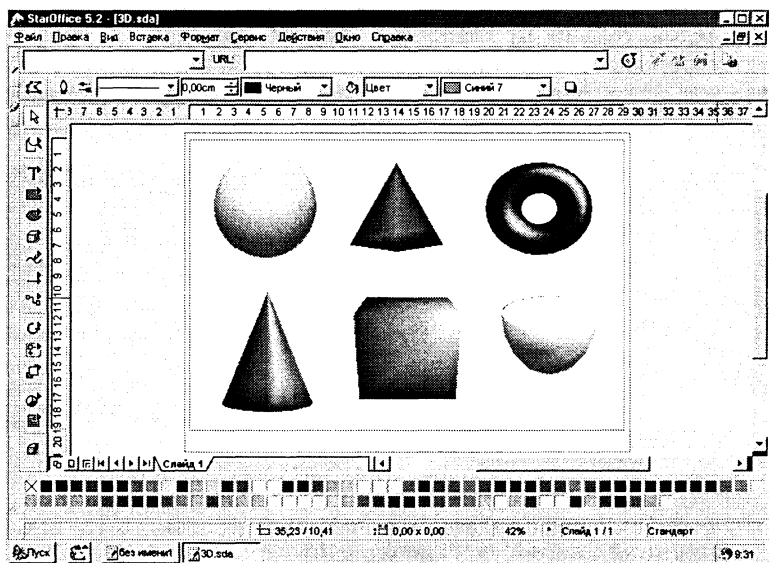


Рис. 4.3. Окно приложения StarOffice Draw

Созданные рисунки можно сохранять в собственном формате SDA, а также экспортировать в растровые форматы BMP, GIF, JPG, PNG и др.

**Практическое задание 4.5. «Создание векторного рисунка и его сохранение в различных графических форматах».** Нарисовать блок-схему линейного алгоритма.



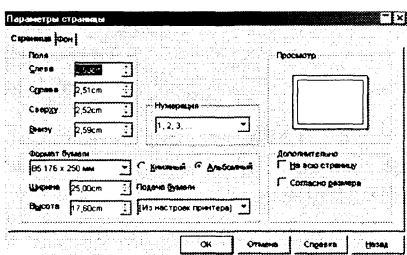
### Создание и сохранение векторного рисунка

1 Запустить интегрированный пакет StarOffice 5.2 и ввести команду [Файл-Создать-Рисунок]. Появится окно приложения *Draw*.

Установим размеры и ориентацию области рисования.

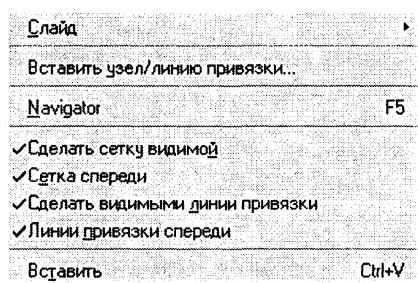
2 Щелкнуть по области рисования правой кнопкой мыши и выбрать в контекстном меню пункт *Слайд-Параметры страницы*.

На появившейся панели выбрать формат бумаги, ее ориентацию, поля и др.



Для большей точности рисования привяжем рисуемые фигуры к сетке и сделаем сетку видимой.

3 Вызвать контекстное меню области рисования и активизировать его пункты, задающие параметры сетки.

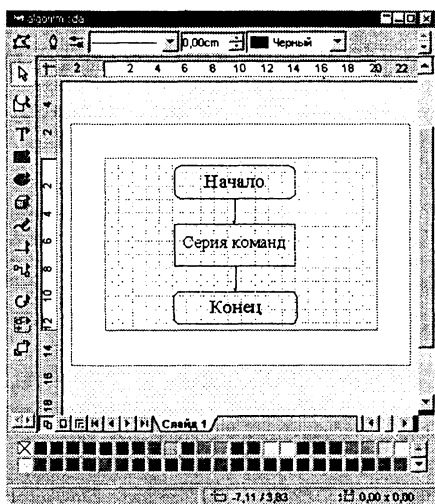


Нарисуем блок-схему линейного алгоритма.

4 Выбрать на отрывной панели *Прямоугольник* прямоугольник с закругленными вершинами и нарисовать его.

Выбрать на отрывной панели *Текст* текстовую область без рамки и нарисовать ее внутри прямоугольника. Ввести текст. Повторить действия при рисовании других элементов блок-схемы.

Выбрать на отрывной панели *Линии и стрелки* стрелку и нарисовать две стрелки.



Для удобства перемещения или копирования блок-схемы, состоящей из отдельных объектов, эти объекты необходимо сгруппировать.

- 5 Последовательно выделить мышью все объекты при нажатой клавише *Shift* и ввести команду [Действия-Группировать].

Сохраним созданный векторный рисунок в собственном формате редактора Draw, а затем экспортируем его в растровый формат JPG.

- 6 Ввести команду [Файл-Сохранить как...] и на панели *Сохранить как* выбрать формат *StarDraw 5.0* и имя файла *algorithm.sda*.
- 7 Ввести команду [Файл-Экспорт...] и на панели *Экспорт* выбрать формат *JPG-JPEG* и имя файла *algorithm.jpg*.

**Практическое задание 4.6. «Создание 3М-графики».** Нарисовать различные трехмерные тела (шар, конус и так далее).

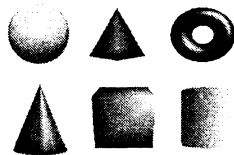


### Создание 3М-графики

- 1 Ввести команду [Файл-Создать-Рисунок].
- 2 Активизировать отрывную панель *Объекты 3-М*.



- 3 Последовательно выбрать и нарисовать *Шар*, *Пирамиду*, *Тор*, *Конус*, *Параллелепипед* и *Цилиндр*.



При рисовании трехмерных тел можно устанавливать различные параметры (режим освещенности, цвет и текстуру поверхности и др.).

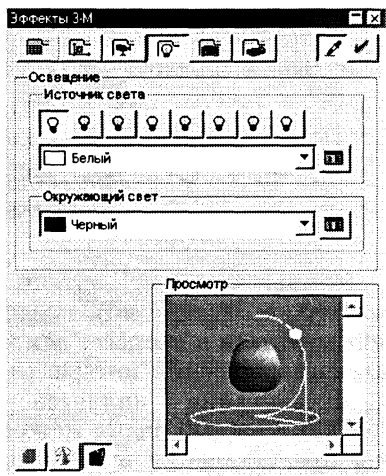
- 4 Активизировать отрывную панель *Эффекты 3-М*.

Щелкнуть по кнопке *Освещенность* и выбрать источник света.

Установить цвет источника и цвет окружающего света.

Перемещая положение источника с помощью мыши или полос прокрутки, в окне *Просмотр* наблюдать изменение освещенности трехмерного тела.

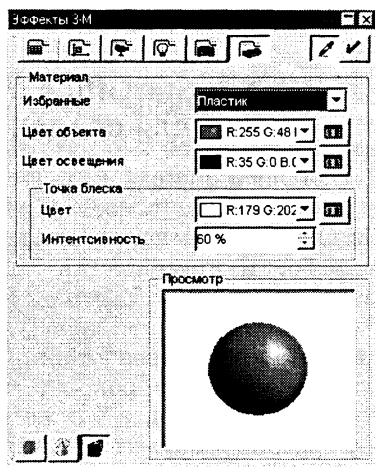
Для присвоения установленных свойств выбрать объект и щелкнуть по кнопке *Присвоить* (кнопка с зеленой «птичкой»).



- 5 Щелкнуть по кнопке *Материал* и выбрать тип материала, цвет объекта и цвет освещения.

Выбрать цвет и интенсивность для точки блеска.

В окне *Просмотр* наблюдать результат применения выбранных установок к объекту. Щелкнуть по кнопке *Присвоить*.



Сохраним созданный векторный рисунок в собственном формате редактора Draw.

- 6 Ввести команду [Файл-Сохранить как...] и на панели *Сохранить как* выбрать формат *StarDraw 5.0* и имя файла *3D.sda*.



### Задания для самостоятельного выполнения

- 4.7. Нарисовать блок-схемы алгоритмических конструкций «полное ветвление» и «неполное ветвление».
- 4.8. Нарисовать блок-схемы алгоритмических конструкций «цикл со счетчиком», «цикл с предусловием» и «цикл с послеусловием».
- 4.9. Нарисовать структурную схему устройства компьютера.

## 4.4. Создание мультимедийных презентаций

Компьютерная презентация представляет собой набор слайдов (электронных страниц), последовательность показа которых может меняться в процессе демонстрации презентации, то есть презентация является интерактивным документом.

Презентация является мультимедийным документом, так как каждый слайд может включать в себя различные формы представления информации (текст, таблицы, диаграммы, изображения, звук, анимацию и др.).

Презентации обычно используют в процессе выступлений на конференциях, для рекламы товаров на выставках, при объяснении нового материала на уроке и так далее.

### 4.4.1. StarOffice Impress — приложение для создания презентаций

StarOffice Impress является средством разработки презентаций, входящим в состав интегрированного офисного пакета StarOffice и является аналогом широко известного средства разработки презентаций Microsoft PowerPoint. StarOffice Impress полностью совместим с PowerPoint по формату файлов презентации (поддерживает формат PPT), по функциональным возможностям (предоставляет выбор шаблонов слайдов, показ слайдов может сопровождаться анимационными и звуковыми эффектами и др.), а также по графическому интерфейсу (набор команд меню этих приложений практически одинаков).

StarOffice Impress позволяет работать как с растровой, так и векторной графикой и позволяет помещать на слайды кубы, шары, цилиндры и другие трехмерные объекты (эти объекты можно различным образом ориентировать, освещать источником света и так далее). В процессе создания презентации можно воспользоваться библиотекой изображений *ClipArts*.

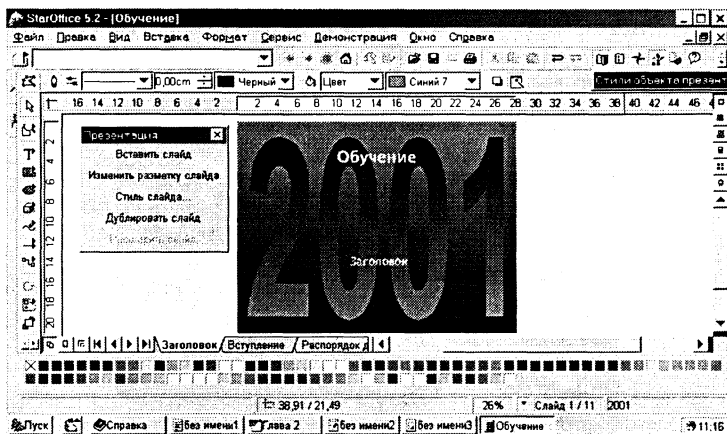


Рис. 4.4. Окно приложения StarOffice Impress

**Практическое задание 4.7. «Знакомство со StarOffice Impress».**

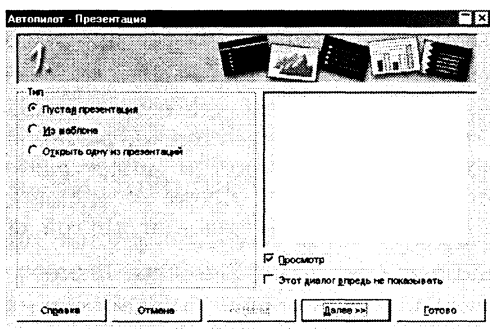
### Знакомство со StarOffice Impress

1 Запустить интегрированный пакет StarOffice 5.2 и ввести команду [Файл-Создать-Презентация]. Появится окно приложения Impress.

Можно создать презентацию (вручную или с помощью мастера) или открыть существующую презентацию.

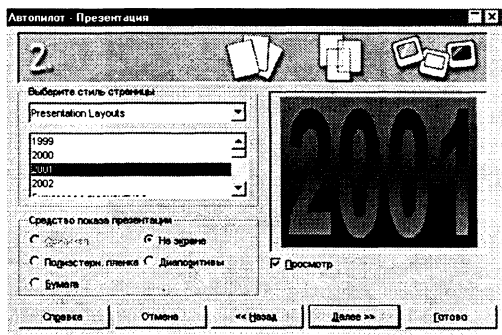
2 Запустится мастер создания презентаций и появится его первая панель *Автопилот – Презентация*.

Выбрать вариант создания новой презентации (*Пустая презентация*, *Из шаблона*) или *Открыть одну из презентаций*. Щелкнуть по кнопке *Далее*.



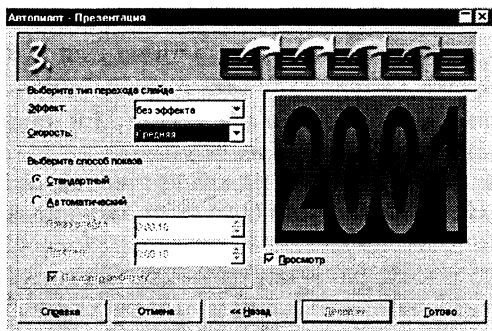
Дизайн презентации можно подобрать из существующих шаблонов.

- 3 На появившейся второй панели из раскрывающегося списка выбрать вариант дизайна создаваемой презентации. Щелкнуть по кнопке *Далее*.



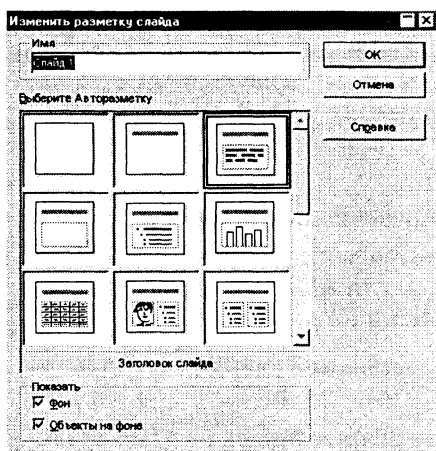
В процессе демонстрации презентации могут выполняться различные анимационные эффекты при смене одного слайда другим (наплыв, вытеснение, растворение и др.).

- 4 На появившейся третьей панели из раскрывающихся списков можно выбрать тип эффекта, реализуемого при переходе слайдов и скорость смены слайдов. Щелкнуть по кнопке *Готово*.



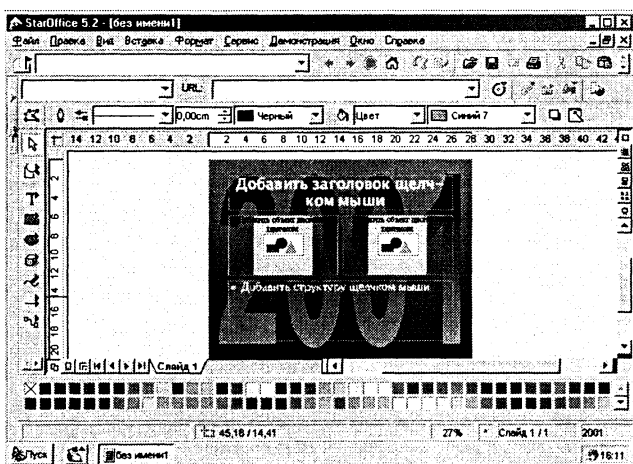
В большинстве случаев у слайда есть заголовок. Текст на слайде может выводиться в виде списка, располагаться в одну или в две колонки. Рядом с текстом может размещаться картинка или диаграмма (график). Кроме того, слайд может занимать целиком таблица, диаграмма или другой объект.

- 5 Появится окно приложения StarOffice Impress (смотри рис. 4.4), а в нем диалоговая панель выбора типа разметки слайдов *Изменить разметку слайда*. Выбрать для каждого слайда будущей презентации один из 20 возможных вариантов разметки.



В режиме разработки презентации в каждый слайд необходимо внести тексты, вставить изображения, диаграммы, таблицы и другие объекты.

- 6 На слайде выбрать структурный элемент щелчком мыши и произвести ввод или редактирование текста или вставку объекта.



#### 4.4.2. Создание презентации

Презентации часто используются в бизнесе для того, чтобы ярче донести до потенциальных клиентов, компаньонов или инвесторов свои идеи.

**Практическое задание 4.8. «Создание презентации».** Создадим литературно-фантастическую презентацию «Нью-Васюки» (по мотивам романа И. Ильфа и Е. Петрова «Двенадцать стуль-

ев»), которая должна помочь Остапу Бендеру убедить васюкинцев профинансировать его идею международного шахматного турнира.

Начиная подготовку презентации, необходимо разработать ее структуру, т. е. определить количество слайдов, их содержание и разметку.



## Создание презентации «Нью-Васюки»

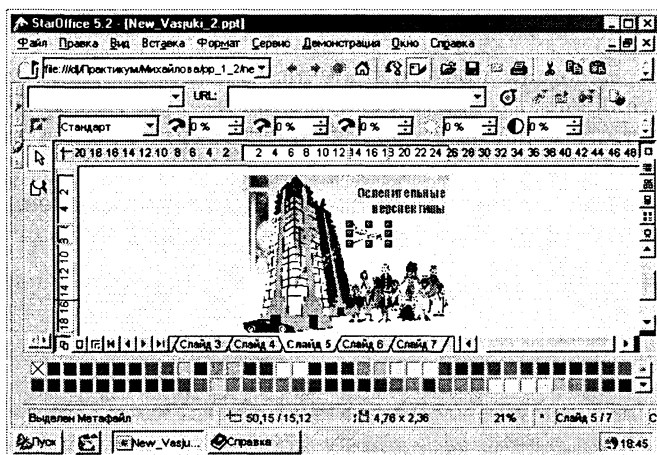
- 1 Создать новую презентацию из семи слайдов, выбрать дизайн презентации, а для каждого слайда выбрать требуемый тип разметки. Внести на слайды тексты.

Номер слайда	Содержание слайда	Разметка слайда
#1	<b><i>Превратим Васюки в Нью-Москву</i></b> О. И. Бендер	
#2	Судьбоносная идея — шахматы обогащают страну.  Организовать «Международный васюкинский турнир 1927 года». Приезд Хозе-Рауля Капабланки, Эммануила Ласкера, Алехина, Нимцовича, Рети, Рубинштейна, Мароцци, Тарраша, Видмара и доктора Григорьева обеспечен.	
#3	<b><i>Необходимое и достаточное</i></b> Личные связи Остапа Бендера. Самодетельность васюкинцев.	
#4	<b><i>Любители шахмат всего мира устремятся в Васюки</i></b> Речной транспорт не справляется. НКПС построит железнодорожную магистраль. Гостиницы и небоскребы для размещения гостей. Поднятие сельского хозяйства в радиусе на тысячу километров: гостей нужно снабжать. Дворец, в котором будет происходить турнир. Постройка гаражей для гостевого автотранспорта. Строительство сверхмощной радиостанции. Аэропорт «Большие Васюки» — регулярное отправление почтовых самолетов и дирижаблей во все концы света.	 

- #5 **Ослепительные перспективы**  
[На этот слайд позднее будут добавлены картинки].
- #6 **После окончания турнира**  
Неслыханный расцвет производительных сил.  
Жители Москвы, стесненные жилищным кризисом, бросятся в ваш великолепный город.  
Столица автоматически переходит в Васюки.
- #7 **Через 8 лет**  
В Васюках состоится первый в истории мироздания межпланетный шахматный турнир.

На слайдах можно размещать как растровые, так и векторные изображения, либо из файлов, либо создаваемые во встроенном *Редакторе изображений*.

- 2 Поместить на слайды № 2, 5, 6 и 7 изображения с помощью команды [Вставка-Рисунок-Из файла...].



Любой объект, размещенный на слайде (изображение, таблица, область текста), можно перемещать или изменять его размеры. Для этого необходимо выделить объект, щелкнув по нему мышью. В результате появляются маркеры выделения объекта в виде маленьких квадратиков по его периметру.

Если поместить указатель мыши на такой маркер, то курсор принимает вид стрелки, направленной в две противоположные стороны. Перетаскивая маркер, можно изменять размер объекта.

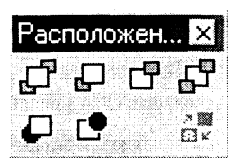
Для перемещения объекта необходимо установить курсор внутри выделенной области (курсор мыши принимает вид стрелок, указывающих «на все четыре стороны») и перетащить объект.

- 3 Поместить на слайд № 5 изображение людей и изменить их размеры и местоположение.



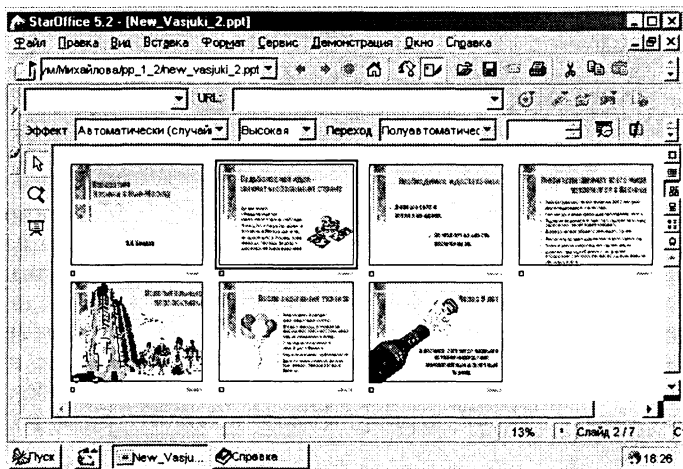
Если одна из картинок перекрывает другие, или наоборот, закрыта другими, то можно, выделив ее, переместить на задний или передний план слайда.

- 4 С помощью панели инструментов *Расположение* последовательно установить для каждого объекта его положение (по «глубине» рисунка) относительно других рисунков.



После создания презентации полезно просмотреть всю последовательность созданных слайдов и, при необходимости, произвести изменение в порядке следования слайдов, перетаскивая их мышью.

- 5 Для просмотра одновременно всех слайдов созданной презентации ввести команду [Вид-Рабочий режим-Режим слайдов].



**Анимационные эффекты появления объектов на слайде.** Просмотрев полученную презентацию, можно увидеть, что слайды 4 и 6 перегружены текстом. Шрифт довольно мелкий — читать трудно и скучно. А если оживить действие, добавить движение? Тогда усилится и восприятие информации, размещенной на слайдах. Любой объект можно заставить возникнуть на экране необычно: вылететь сбоку, развернуться до заданного размера, уменьшиться, вспыхнуть, проявиться, вращаться и т. д. А текст еще может появляться целиком или, по словам или, даже по отдельным буквам.

- 6 Выделить на слайде объект с текстом и ввести команду [Демонстрация-Эффект].

На появившейся панели *Эффект* выбрать вкладку *Эффекты текста*. В раскрывающемся списке выбрать тип эффекта, например, *Наплыв*, а в окне выбрать значок с направлением наплыва, например, *Наплыв из верхнего левого угла*.



**Анимационные эффекты при переходах слайдов.** Можно добавить эффекты анимации при переходах слайдов, то есть при смене одного слайда следующим. Настройку анимационных эффектов при переходах слайдов удобно производить в рабочем режиме *Режим слайдов*.

Настройки можно применить как к одному выбранному слайду, так и сразу ко всем слайдам презентации. Для этого надо выделить все слайды командой [Правка-Выделить все].

- 7 Выделить слайд и ввести команду [Демонстрация-Переход слайда]. На появившейся панели *Переход слайда* выбрать вкладку *Эффекты*. В раскрывающемся списке выбрать тип эффекта, например, *Избранные*, а в окне выбрать значок со способом перехода, например, *Растворение*.

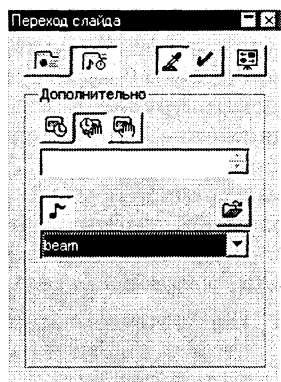


Переходы слайдов можно сопровождать также звуковыми эффектами.




- 8 Выделить слайд и ввести команду [Демонстрация-Переход слайда]. На появившейся панели *Переход слайда* выбрать вкладку *Дополнительно*.

Щелкнуть по кнопке *Дополнительно* и в раскрывающемся списке выбрать звуковой файл.

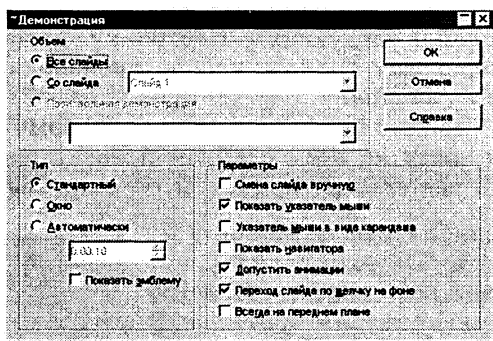


- 9 Сохранить демонстрацию в файле New\_Vasjuki.ppt.

Презентация хранится в каталоге \practicum\inftech\impress\ CD-ROM 

**Демонстрация презентации.** Процесс демонстрации презентации также можно настраивать. Можно определить с какого слайда должна начинаться презентация, какое событие вызывает переход на следующий слайд и т. д.

- 10 Ввести команду [Демонстрация-Настройки демонстрации...]. На появившейся панели *Демонстрация* произвести настройку.



- 11 Запустить демонстрацию презентации командой [Демонстрация-Демонстрация] или щелчком по кнопке *Демонстрация*. Переход к следующему кадру осуществлять щелчком мыши или нажатием клавиши *Enter*.

Если сохранить презентацию в собственном формате StarOffice Impress (SDD), то можно будет проводить демонстрацию презентации с помощью проигрывателя презентаций StarOffice Player (без установки на компьютер интегрированного приложения StarOffice).

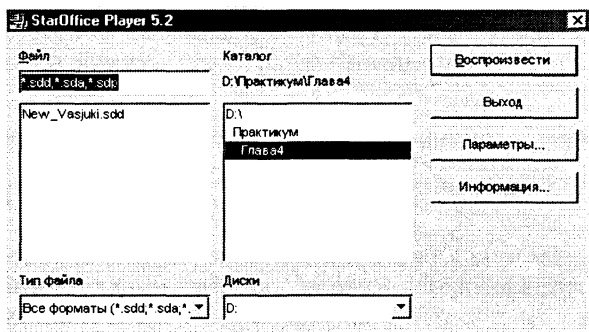
12 Сохранить презентацию в файле New\_Vasjuki.sdd.

*Презентация хранится в каталоге*  
\\practicum\inftech\impress\

CD-ROM 

13 Запустить проигрыватель презентаций StarOffice Player 5.2 командой [Пуск-StarOffice Player 5.2].

На появившейся панели выбрать файл презентации для демонстрации.



### Задания для самостоятельного выполнения

- 4.10. Создать презентацию «Путешествие во времени», которая в автоматическом режиме рекламирует хронотур (путешествие во времени) на ежегодной выставке туристических фирм. Предусмотреть различные эффекты при переходе слайдов и появлении на слайдах объектов.
- 4.11. Создать презентацию «Рога и копыта», которая могла бы сопровождать выступление Остапа Бендера при представлении фирмы «Рога и копыта» (по мотивам романа И. Ильфа и Е. Петрова «Золотой теленок»). Включить в презентацию слайды, показывающие организационную структуру фирмы (структурные схемы) и итоги финансовой деятельности (диаграммы).
- 4.12. Создать презентацию «Три Александра», которая могла бы сопровождать доклад на уроке или школьной конференции по истории России. Предусмотреть возможность управления демонстрацией презентации в процессе выступления.

## 4.5. Система автоматизированного проектирования КОМПАС-3D

Окно приложения КОМПАС-ГРАФИК. Система автоматизированного проектирования (САПР) КОМПАС-3D позволяет создавать чертежи любого уровня сложности с полной поддержкой российских стандартов.

В центре рабочего окна КОМПАС-3D размещается система координат. Положение курсора отсчитывается от начала системы координат, а текущие значения его координат  $X$  и  $Y$  отображаются в правой части *Строки текущего состояния*, расположенной в нижней части окна приложения.

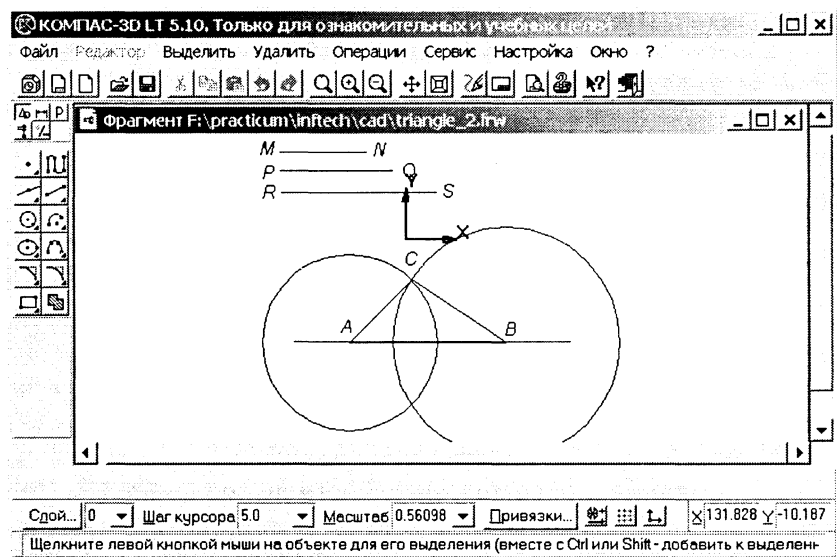


Рис. 4.5. Окно САПР КОМПАС-3D

Создание и редактирование чертежа реализуется с помощью *Инструментальной панели*, которая по умолчанию размещается в левом верхнем углу окна приложения. *Инструментальная панель* включает в себя пять различных рабочих панелей, каждая из которых содержит набор кнопок определенного функционального назначения и *Панель переключения*, которая обеспечивает переход от одной рабочей панели к другой.

Основная рабочая панель *Геометрические построения* содержит кнопки, позволяющие рисовать на чертеже определенные объекты: точка, отрезок, окружность, прямоугольник и др.

Панель *Редактирование* содержит кнопки, которые позволяют вносить изменения в чертеж, производя над объектами различные операции: перемещение, копирование, масштабирование и так далее.

Панель *Выделение* позволяет осуществить различные варианты выделения объектов: отдельные объекты, группы объектов и так далее.

Панель *Измерения* позволяет измерять расстояния (вычисляются и отображаются в миллиметрах), углы (в градусах), периметры и площади различных объектов.

Панель *Размеры и технологические обозначения* позволяет грамотно оформить чертеж: обозначить на чертеже размеры деталей, сделать надписи и так далее.

**Геометрические построения.** С помощью САПР удобно проводить геометрические построения, которые занимают существенное место в курсе геометрии.

**Практическое задание 4.9.** «Построение угла, равного заданному». Отложить угол, равный заданному углу  $A$ , от луча  $OM$ .



### Геометрическое построение угла, равного заданному

- 1 Составим сначала алгоритм выполнения заданного построения:
  1. Построить окружность произвольного радиуса с центром в вершине заданного угла  $A$ , которая пересечет стороны угла в точках  $B$  и  $C$ .
  2. Построить окружность того же радиуса с центром в начале заданного луча  $OM$ , которая пересечет луч в точке  $D$ .
  3. Построить окружность с центром в точке  $D$  и радиусом  $BC$ .
  4. Обозначить точку пересечения окружностей с центрами  $O$  и  $D$ , не лежащую на луче  $OM$ , буквой  $E$ .
  5. Полученный угол  $MOE$  равен заданному  $A$ .

В процессе геометрического построения нам будет необходимо построить четыре отрезка и три окружности. Для выбора объекта рисования необходимо на панели *Геометрические построения* щелкнуть мышью по соответствующей кнопке. Появится *Строка параметров* объекта, причем каждый объект обладает определенным набором параметров, которые характеризуют его размеры и положение на чертеже.

Например, после выбора на панели *Геометрические построения* кнопки *Ввод отрезка* появится строка с параметрами отрезка: координатами его начальной ( $p1$ ) и конечной ( $p2$ ) точек, длиной ( $ln$ ), углом наклона ( $an$ ) и стилем линии.

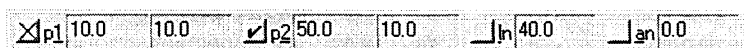
*Строка параметров* включает в себя кнопки *состояния полей* и сами *поля*. По внешнему виду кнопки можно судить о состоянии поля, которое может находиться в одном из трех состояний: *фиксированном* (обозначается «крестиком»), *в режиме ожидания ввода* (обозначается «галочкой») и просто *доступном для ввода*.

При создании и редактировании объектов работа со *Строкой параметров* сводится к активизации нужных полей и вводу в них заданных параметров. После ввода минимального набора параметров, достаточных для построения объекта (для отрезка — координат начальной и конечной точек), система автоматически создает объект.

Можно осуществлять *Автоматический ввод параметров* (считываются координаты точек, на которых устанавливается курсор), *Ручной ввод параметров* (обеспечивается прямым вводом чисел в поля строки параметров) и *Ввод параметров с использованием Геометрического калькулятора* (геометрические размеры вычисляются с использованием координат заданных на чертеже точек).

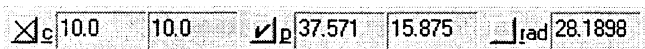
Обозначение точек на чертеже осуществить с помощью панели *Размеры и технологические обозначения*.

2 Построить произвольный угол  $A$  (ввести отрезки с использованием автоматического ввода параметров).



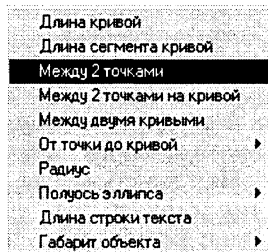
3 Построить произвольный луч  $OM$  (ввести отрезок с использованием автоматического ввода параметров).

4 Построить окружность с центром в точке  $A$  (с использованием ручного ввода) и произвольного радиуса (с использованием автоматического ввода).



Обозначить точки пересечения окружности со сторонами угла буквами  $B$  и  $C$ .

5 Построить окружность с центром в точке  $O$  (с использованием ручного ввода) и заданного радиуса  $AB$  (с использованием геометрического калькулятора). Для этого щелкнуть по полю *rad* правой кнопкой мыши и в контекстном меню выбрать пункт *Между 2 точками*.



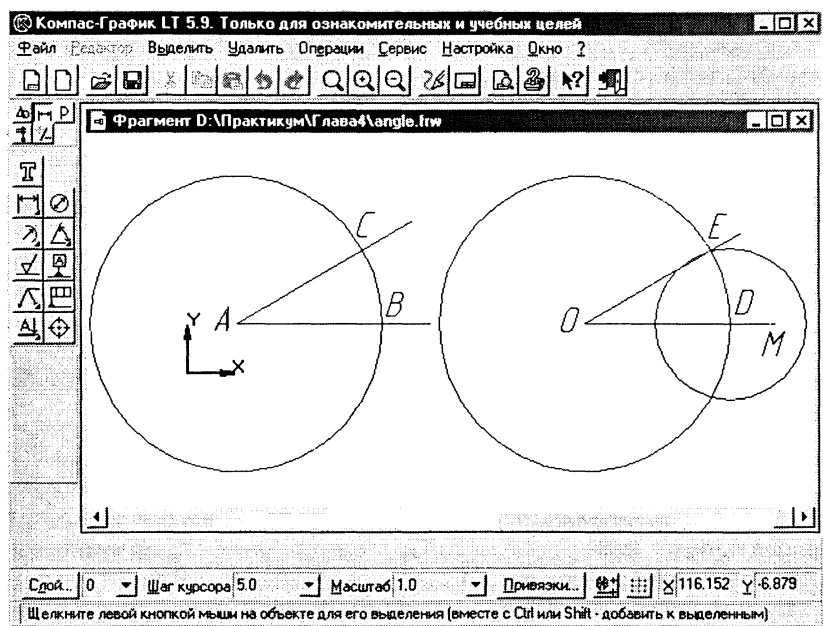
Отметить на чертеже точки  $A$  и  $B$ , окружность заданного радиуса будет построена.

Обозначить точку пересечения окружности с лучом  $OM$  буквой  $D$ .

6 Построить окружность с центром в точке  $D$  заданного радиуса  $BC$ .

Обозначить точку пересечения окружностей буквой  $E$ .

- 7 Соединить отрезком точки  $O$  и  $E$ , угол  $EOM$ , равный углу  $A$ , построен.



- 8 Сохранить чертеж в файле angle.frw.

Чертеж хранится в каталоге  
 \practicum\inftech\cad\

CD-ROM 

### Задания для самостоятельного выполнения

- 4.13. Ознакомьтесь с возможностями КОМПАС-3D по созданию чертежей трехмерных объектов (деталей).

## 4.6. Работа в электронных таблицах StarOffice Calc

**StarOffice Calc** — электронные таблицы. StarOffice Calc входит в состав интегрированного офисного пакета StarOffice и является аналогом широко распространенных электронных таблиц Microsoft Excel. Calc полностью совместим с Excel по формату файлов (поддерживает формат XLS), по функциональным возможностям (позволяет обрабатывать числовые данные

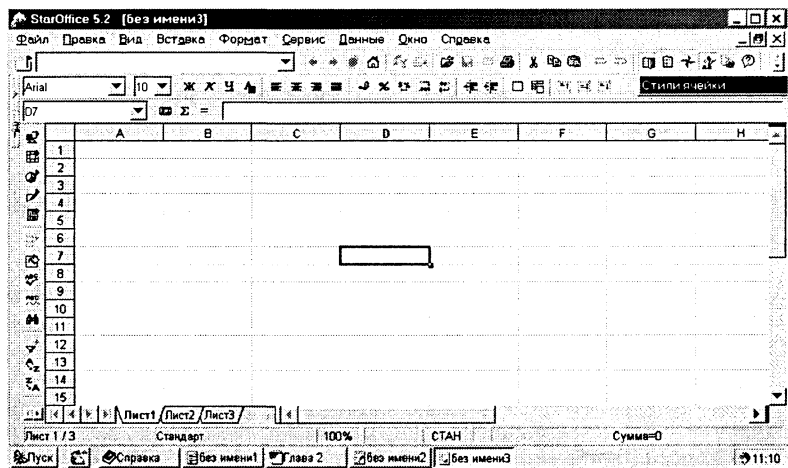


Рис. 4.6. Окно приложения StarOffice Calc

в различных форматах, обеспечивает построение диаграмм и графиков, сортировку и поиск данных так далее), а также по графическому интерфейсу (набор команд меню этих приложений практически одинаков).

#### Практическое задание 4.10. «Цена устройств компьютера».

В целях приобретения компьютера оптимальной конфигурации оценить доли, которые составляют цены входящих в него устройств, от общей цены компьютера. Представить результат в форме диаграммы.

#### Цена устройств компьютера

- 1 Запустить интегрированный пакет StarOffice 5.2 и ввести команду [Файл-Создать-Документ электронной таблицы]. Появится окно приложения Calc.
- 2 Ввести наименования устройств, входящих в компьютер, и их цены в условных единицах.

Для удобства расчетов переведем цены в рубли по курсу, значение которого будет храниться в ячейке B1.

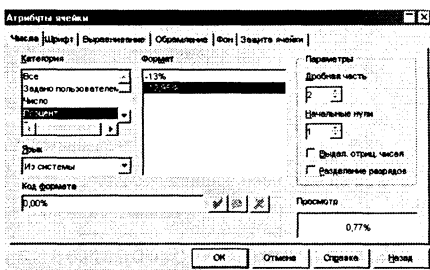
- 3 Введем в ячейку C4 формулу перевода цены в рубли  $=\$B\$1*B4$ . Воспользуемся абсолютной ссылкой на ячейку B1 для правильного копирования формулы в ячейки C5-C15 с помощью команды [Правка-Заполнить-Вниз].

Подсчитаем в ячейке C16 общую цену компьютера с помощью функции СУММ и определим долю цены каждого устройства в процентах.

4 Введем в ячейку D4 формулу определения доли цены устройства в общей цене компьютера =B4/\$B\$16. Воспользуемся абсолютной ссылкой на ячейку B16 для правильного копирования формулы в ячейки D5-D15.

5 Для представления чисел в столбце D в форме процентов необходимо ввести команду [Формат-Ячейка...].

На появившейся панели *Атрибуты ячейки* в окне *Категория* выбрать *Процент*.



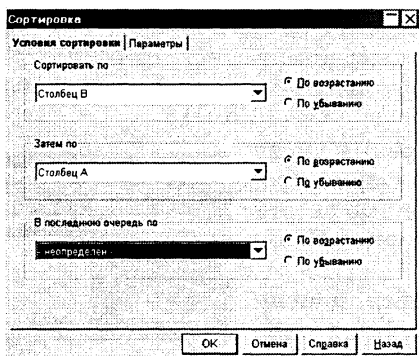
	A	B	C	D
1	1 у.е. =	29р.		
2				
3	<b>Наименование устройства</b>	<b>Цена (у.е.)</b>	<b>Цена (рубли)</b>	<b>Процент</b>
4	Видеоплата 16 Мб	10	290р.	1,55%
5	Дисковод 3,5"	12	348р.	1,85%
6	Дисковод CD-ROM x50	25	725р.	3,86%
7	Жесткий диск 50 Гб	30	870р.	4,64%
8	Звуковая карта	30	870р.	4,64%
9	Клавиатура	35	1 015р.	5,41%
10	Корпус	40	1 160р.	6,18%
11	Монитор 15"	70	2 030р.	10,82%
12	Мышь	5	145р.	0,77%
13	Память 64 Мб	80	2 320р.	12,36%
14	Процессор Celeron (600 МГц)	130	3 770р.	20,09%
15	Системная плата	180	5 220р.	27,82%
16	<b>ВСЕГО:</b>	<b>647</b>	<b>18 763р.</b>	

Данные в таблице можно сортировать по возрастанию или убыванию их значений.

6 Для сортировки устройств по возрастанию их цены выделить диапазон ячеек A4:D15 и ввести команду [Данные-Сортировка].

На появившейся панели *Сортировка* выбрать из списков последовательность столбцов для реализации вложенной сортировки и порядок сортировки (*По возрастанию* или *По убыванию*).

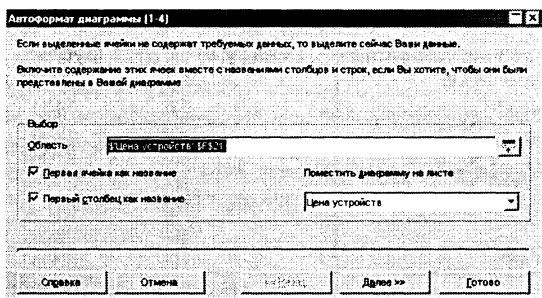




Для наглядного отображения полученных числовых данных построим диаграмму.

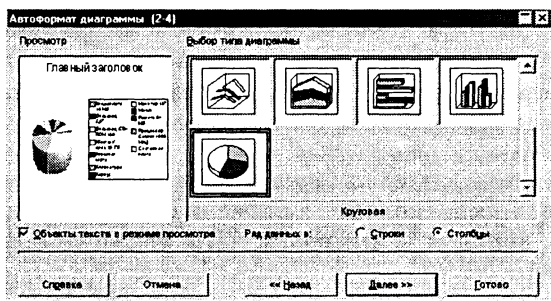
7 Выделить область данных и ввести команду [Вставка-Диаграмма].

На появившейся первой панели *Автоформат диаграммы* (1-4) можно изменить область данных.



8 На второй панели необходимо выбрать тип диаграммы (в данном случае *Круговая объемная*) и расположение ряда данных (в данном случае *Столбцы*).

В окне *Просмотр* демонстрируется вид выбранной диаграммы.



9 На третьей и четвертой панелях можно уточнить детали внешнего вида диаграммы и легенды.

В результате мы получили круговую диаграмму, которая дает наглядное представление о «вкладе» каждого устройства в цену компьютера.



10 Сохранить таблицу с данными и диаграмму на листе *Цена устройств* в файле *calc.xls*.

Файл хранится в каталоге  
`\practicum\inftech\calc\`

CD-ROM



**Практическое задание 4.11. «Построение графиков функций».** Построить графики функций  $y = x^2 - 0,5$  и  $y = x^3$ . Приблизленно (графически) решить уравнение  $x^3 - x^2 + 0,5 = 0$ .

### Построение графиков функций

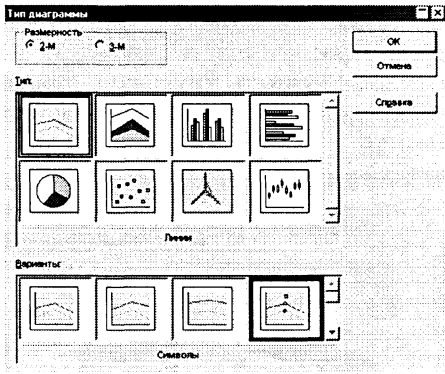
1 Перейти на другой лист электронных таблиц файла *calc.xls* и переименовать его в *Графики функций*.

Ввести формулы функций и заполнить таблицу значений функций на интервале от  $-1$  до  $1$  с шагом  $0,2$ .

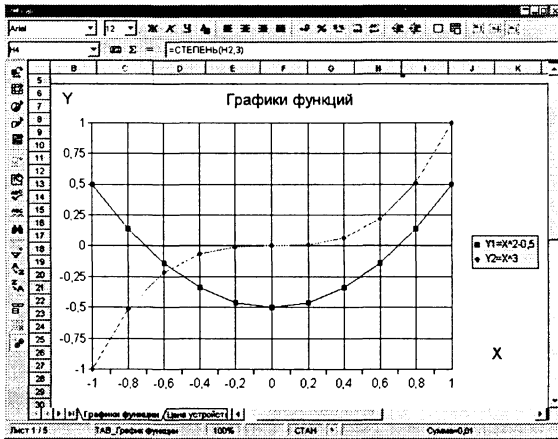
Таблица значений функций												
X	-1	-0,8	-0,6	-0,4	-0,2	0	0,2	0,4	0,6	0,8	1	
Y1=X <sup>2</sup> -0,5	0,5	0,14	-0,14	-0,34	-0,46	-0,5	-0,46	-0,34	-0,14	0,14	0,5	
Y2=X <sup>3</sup>	-1	-0,51	-0,22	-0,06	-0,01	0	0,01	0,06	0,22	0,51	1	

2 Ввести команду [Вставка-Диаграмма...].

На панелях *Автоформат диаграммы* выбрать *Ряд данных в: Строки* и тип диаграммы *Линии с символами*.



- 3 Будут построены два графика функций, координата  $X$  точки пересечения которых и будет решением уравнения ( $X \approx -0,55$ ).



- 4 Сохранить таблицу с данными и диаграмму на листе *Графики функций* в файле calc.xls.

Файл хранится в каталоге  
 \practicum\inftech\calc\


CD-ROM 

### Задания для самостоятельного выполнения

- 4.14. Построить диаграмму, показывающую рост количества серверов Интернета с 1991 по 2002 год (статистические данные имеются на CD-ROM в каталоге \soft\Internet\Internet-statistic\).
- 4.15. Построить диаграмму, показывающую распределение серверов Интернета по странам.
- 4.16. Построить графики изменения курсов доллара и евро по отношению к рублю за несколько месяцев (данные найти в Интернете).

# Глава 5

## Объектно-ориентированное программирование на языке Visual Basic

Для изучения объектно-ориентированного программирования рекомендуется установить свободно распространяемую версию системы программирования Visual Basic (VB5.0 CCE – Visual Basic 5.0 Control Creation Edition). CD-ROM 

Для автоматического вызова системы программирования VB5.0 CCE при щелчке по имени файла проекта или при активизации ссылки на него необходимо зарегистрировать тип файлов проектов VBP.

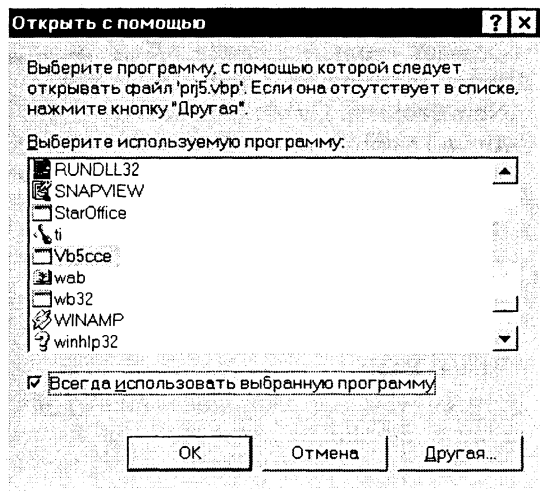
Для этого необходимо при нажатой клавише *{Shift}* щелкнуть правой кнопкой мыши по имени файла.

Появится диалоговая панель *Открыть с помощью*, содержащая список установленных на данном компьютере программ.

Необходимо выбрать в этом списке программу (в данном случае *Vb5cce*), которая будет запускаться при активизации файлов выбранного типа.

Щелчок по кнопке *OK* завершит регистрацию.

Рис. 5.1  
Регистрация файлов проектов типа VBP



В ряде проектов (проект 5.2 «Компьютер в картинках», проект 5.5 «Сортировка строкового массива», в заданиях 5.9, 5.10, 5.11, 5.17, 5.18, 5.19, 5.20) в процессе выполнения проектов происходит обращение к файлам, хранящимся на CD-ROM.

Если проекты загружаются из среды Visual Basic или по гиперссылке, то в программном коде нужно указывать полное имя внешнего файла (включая путь к этому файлу).

В хранящихся на CD-ROM готовых проектах считается, что CD-ROM имеет логическое имя E:. Если на вашем компьютере CD-ROM имеет другое логическое имя, то необходимо внести соответствующие изменения в программные коды проектов.

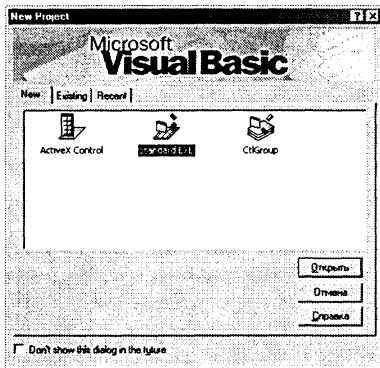
Для подключения справочной системы языка программирования Visual Basic необходимо запустить среду программирования VB5.0 CSE и ввести команду [Help-Microsoft Visual Basic Help Topic]. На появившейся информационной панели *Справка Windows* будет сообщено, что файл справки не найден и рекомендуется найти его самостоятельно. Для подключения файла щелкнуть по кнопке *Да* и в открывшемся диалоговом окне найти файл справки VB5.hlp, который находится в каталоге \soft\programm\VisualBasic.

## 5.1. Интегрированная среда разработки языка Visual Basic



### Знакомство с интерфейсом системы программирования Visual Basic

- 1 Запустить систему программирования VB5.0 CSE.
- 2 Создать новый проект. Для этого ввести команду [File-New Project]. На открывшейся диалоговой панели *New Project* выбрать тип создаваемого проекта *Standart.exe*.
- 3 После щелчка по кнопке *Открыть* появится окно интегрированной среды разработки Visual Basic.



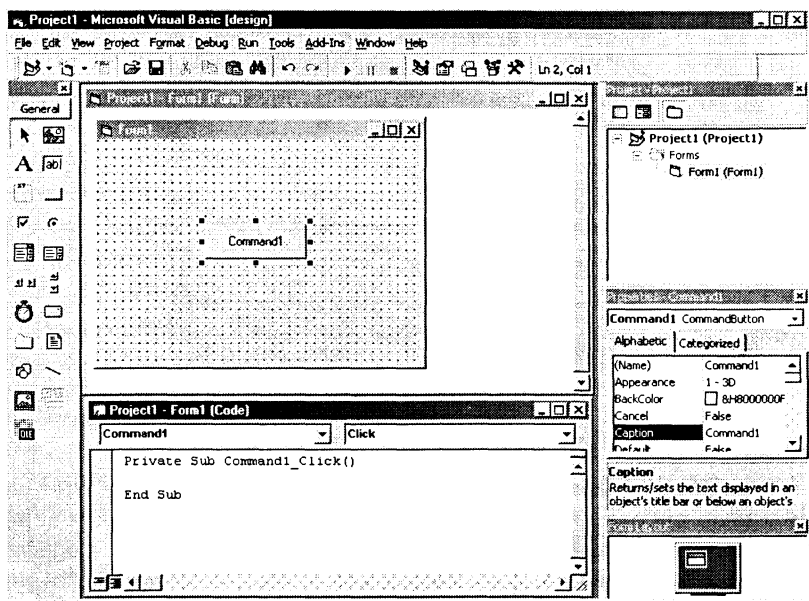


Рис. 5.2. Окно интегрированной среды разработки Visual Basic

В центре окна *Visual Basic* расположено окно *Конструктор форм*, в котором размещается главный объект проекта *Форма* (Form1). *Форма* является основой для создания графического интерфейса создаваемого проекта и на ней размещаются различные *Управляющие элементы* (Controls).

Выбор размещаемых на форме *Управляющих элементов* производится с помощью *Панели инструментов* (ToolBox), которая обычно размещается в левой части окна приложения. *Стандартная* (General) панель инструментов содержит 21 класс управляющих элементов: *Метка* (Label), *Текстовое поле* (TextBox), *Командная кнопка* (CommandButton) и др.

Окно *Свойства объекта* (Properties) предоставляет возможность просмотра и редактирования значений свойств выбранного объекта. В верхней части окна имеется раскрывающийся список всех объектов проекта. В нижней части окна содержится список всех свойств выбранного объекта, а для каждого свойства — список его возможных значений. Окно *Свойств объекта* располагается в середине правой части окна приложения.

Окно *Программный код* (Code) позволяет просматривать и редактировать программный код проекта. Вызывается окно *Программный код* командой [View-Code] и располагается под окном *Формы*.

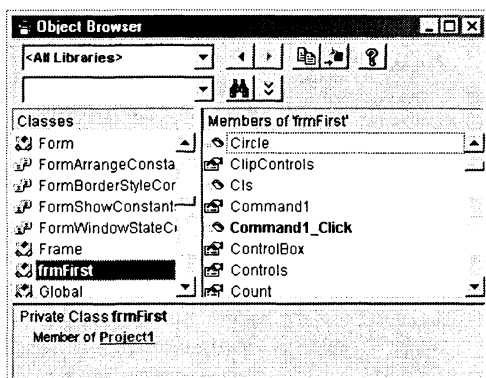


Рис. 5.3. Окно Свойства объекта

Окно *Проводник проекта (Project Explorer)* представляет содержимое проекта, т. е. входящие в него файлы, в форме дерева файлов. В состав проекта входит собственно файл проекта (имеет расширение *vbp*), файлы форм, которых может быть несколько (имеют расширение *frm*), и файлы программных модулей (имеют расширение *bas*). Это окно располагается в верхней правой части окна приложения.

Окно *Расположение формы (Form Layout)* позволяет установить положение формы (фактически окна созданного приложения) на экране монитора в процессе выполнения программы. Окно *Расположение формы* размещается в нижнем правом углу окна приложения.

Наконец, командой [View-Object Browser] можно вызвать окно *Просмотр объектов (Object Browser)*. Окно содержит в левом списке все доступные классы объектов и объекты, текущего проекта (выделены жирным шрифтом), а в левом списке для выбранного объекта показывает перечень его свойств, методов и событий.

## 5.2. Форма и размещение на ней управляющих элементов

Форма является основой для создания графического интерфейса разрабатываемого проекта. На форме можно разместить различные управляющие элементы (кнопки, текстовые поля и др.).

**Проект 5.1. «Расположение формы и управляющих элементов».** Создать проект, в котором после запуска форма располагается в центре экрана монитора, а четыре кнопки располага-

ются в центре формы. После щелчка по кнопкам они должны перемещаться в углы формы.

### Проект «Расположение формы на экране монитора и управляющих элементов на форме»

- 1 Создать новый проект. Активизировать форму Form1 щелчком мыши. В окне *Свойства объекта* выбрать свойство Caption и присвоить ему значение Первый проект.
- 2 Разместить в центре формы четыре кнопки Command1, Command2, Command3 и Command4.

Целесообразно объектам проекта присваивать имена, которые дают возможность распознать их тип и назначение. Принято, что имя начинается с префикса, который определяет тип объекта. Для форм принят префикс frm, для надписей — lbl, для командных кнопок — cmd и так далее. После префикса идет информативная часть имени, которая пишется с заглавной буквы (например: frmFirst, lblText, cmdExit) или содержит число (например, frm1, frm2, frm3).

- 3 Последовательно активизировать форму и кнопки. В окне *Свойства объекта* последовательно для всех объектов изменить свойство Name на frm1, cmd1, cmd2, cmd3 и cmd4.

**Расположение формы на экране монитора.** Местоположение формы, которое она будет занимать на экране монитора после запуска приложения на выполнение, можно легко изменять с использованием окна *Расположение формы*. Для этого достаточно перетащить форму с помощью мыши в любое место экрана.

Точное местоположение и размеры формы отображаются двумя парами чисел в правой части линейки инструментов окна приложения. Первая пара чисел показывает расстояние от левого верхнего угла монитора до левого верхнего угла формы, а вторая пара — это размеры формы (ширина и высота). Размеры отображаются в особых единицах — твипах (один твип равен примерно 0,018 мм).

Если необходимо установить точные значения местоположения и размеров формы, то это можно сделать, установив значения этих свойств формы с помощью окна *Свойства объекта*:

- Left (расстояние по горизонтали от левого верхнего угла монитора до верхнего левого угла формы);
  - Top (расстояние по вертикали от левого верхнего угла монитора до верхнего левого угла формы);
  - Width (ширина формы);
  - Height (высота формы).
- 4 Активизировать форму frm1. В окне *Свойства объекта* последовательно присвоить свойствам Left, Top, Width и Height значения 1000, 3000, 4000, 3000.



**Выравнивание формы с использованием событийной процедуры и метода Move.** Любой объект можно связать с набором процедур, исполняемых при наступлении определенных событий. Процедура, связанная с определенным *событием*, называется *событийной процедурой*. Событиями являются, например, Click (щелчок по объекту мышью), Resize (изменение размера объекта), Load (загрузка объекта) и так далее.

Выравнивание формы по центру экрана монитора можно производить автоматически при запуске проекта. В этот момент производится загрузка формы и активируется событие Form Load(). Напишем код этой событийной процедуры, который будет производить выравнивание.

Объект Screen (экран монитора) обладает свойствами Screen.Width (ширина экрана) и Screen.Height (высота экрана). Форма frm1 также обладает свойствами frm1.Width и frm1.Height. Форма будет размещаться в центре экрана, если:

left (левая координата верхнего левого угла формы) будет равна:

$$(\text{Screen.Width} - \text{frm1.Width})/2$$

top (верхняя координата верхнего левого угла формы) будет равна:

$$(\text{Screen.Height} - \text{frm1.Height})/2$$

Для перемещения объектов в определенное место используется метод Move. Формат метода следующий:

```
object.Move left, top
```

Воспользуемся методом Move для перемещения формы в заданное координатами left и top место экрана монитора:

```
frm1.Move(Screen.Width - frm1.Width)/2,  
(Screen.Height - frm1.Height)/2
```

Таким образом, код событийной процедуры выравнивания формы по центру экрана монитора будет следующий:

```
Private Sub Form_Load()  
frm1.Move(Screen.Width - frm1.Width)/2,  
(Screen.Height - frm1.Height)/2
```

```
End Sub
```

5 Двойным щелчком по форме вызвать окно *Программный код* с пустой процедурой Form\_Load(). Ввести код процедуры выравнивания.

**Расположение управляющих элементов на форме.** Расположение на форме и размеры управляющих элементов можно легко изменять с помощью мыши. Для этого необходимо активировать объект щелчком мыши (он будет выделен восьмью

квадратиками-маркерами) и перетащить его на новое место или «ухватить» маркер и перетащить границу элемента.

Точное местоположение и размеры выделенных элементов управления отображаются двумя парами чисел в правой части линейки инструментов окна приложения. Первая пара чисел показывает расстояние от левого верхнего угла формы до левого верхнего угла элемента, а вторая пара — это размеры элемента (ширина и высота).

- 6 Разместить четыре кнопки в центре формы с помощью мыши. При необходимости задать точное положение каждой кнопки на форме и их размеры с помощью окна *Свойства объекта*.

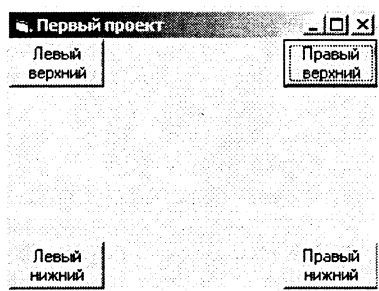


Создать для каждой кнопки событийные процедуры, реализующие их перемещение в углы формы.

- 7 Для кнопки cmd3, которая должна перемещаться в правый нижний угол код событийной процедуры следующий:

```
Private Sub cmd3_Click()  
cmd3.Move frm1.Width - cmd3.Width, frm1.Height -  
cmd3.Height  
End Sub
```

- 8 Запустить проект. Форма появится в центре экрана монитора. Последовательно щелкнуть по всем кнопкам, они переместятся в углы формы.



**Сохранение проекта.** Проект готов и его необходимо сохранить на диске. Найдите в меню *File* пункт *Save as...* Сначала Visual Basic предложит вам сохранить форму, причем по умолчанию даст то имя, которое вы задали ей в свойстве *Name*. Файл с формой и кодами будет храниться под именем вашей формы с расширением *frm*.


После того как форма сохранена, Visual Basic предложит вам записать на диск файл проекта и по умолчанию предложит вам имя Project1.vbp. В соответствии с соглашением об именах объектов целесообразно сохранить проект под именем с префиксом prj, например, prj1.

На диске будут сохранены три файла: файл формы с расширением frm, файл проекта с расширением vbp и еще один вспомогательный файл с расширением vbw. Так как проекты могут содержать большее количество файлов (например, несколько файлов форм), рекомендуется для каждого проекта создавать свою папку на диске.

9 Сохранить файл формы командой [Save Form1.frm As ...] как frm1.frm и файл проекта командой [Save Project As ...] как prj1.vbp.

Сохраненный проект может выполняться только в системе программирования Visual Basic. Для того чтобы проект мог запускаться в среде операционной системы, необходимо сохранить его в исполняемом файле (типа EXE). Для трансляции проекта в исполняемый файл используется команда [File-Make ...] (в свободно распространяемой версии VB5.0 CSE такая возможность, к сожалению, отсутствует).

---

*Проект хранится в каталоге* CD-ROM   
practicum\VB\Projects\project1\

---

### Задания для самостоятельного выполнения

- 5.1. Модернизировать проект «Местоположение формы и управляющих элементов» так, чтобы местоположение кнопок задавалось с помощью свойств Top и Left.
- 5.2. Создать проект, который позволяет перемещать по форме надпись с помощью управляющих элементов *Кнопка* (CommandButton).
- 5.3. Создать проект, который позволяет перемещать по форме надпись с помощью управляющих элементов *Переключатель* (OptionButton).
- 5.4. Создать проект, который позволяет перемещать по форме надпись с помощью управляющего элемента *Список* (ListBox).
- 5.5. Создать проект «Перемещение формы на экране», в котором будет изменяться местоположение формы на экране монитора с помощью четырех командных кнопок. Кнопки расположить в углах формы. После щелчка по кнопке форма должна переместиться в соответствующую часть экрана. Перемещение формы должно сопровождаться комментариями, выводимыми с помощью надписи в центре формы.
- 5.6. Модернизировать предыдущий проект 5.5 так, чтобы осуществлять передвижение формы в углы экрана монитора с использованием метода Move.

## 5.3. Размещение на формах изображений и графических полей

Автоматический вывод изображений на форму после запуска проекта. Для того чтобы изображение автоматически размещалось на форме после запуска проекта, необходимо в процессе разработки проекта воспользоваться свойством формы `Picture` (изображение). Для этого в окне *Свойства объекта* из списка свойств формы необходимо выбрать и активизировать пункт `Picture`. В появившемся диалоговом окне выбрать файл, содержащий нужное изображение.

**Проект 5.2. «Компьютер в картинках».** Создадим проект, который будет выводить на форму изображения компьютера и его устройств.

### Проект «Компьютер в картинках»

- 1 Создать новый проект.
- 2 В окне *Свойства объекта* установить для свойства формы `Name` значение `frm1` и для свойства `Caption` значение Первая форма.
- 3 Активизировать свойство `Picture`, щелкнуть по значку «...» в окошке этого свойства и с помощью появившегося диалогового окна *Load Picture* выбрать нужный графический файл (например, `computer.gif`, который на CD-ROM находится в каталоге `\practicum\VB\Projects\project2\`).
- 4 После запуска проекта на выполнение появится форма с размещенным на ней изображением, которое размещается в верхнем левом углу формы. Если изображение больше формы, то будет видна только та его часть, которая помещается на форме.



**Вывод изображений на форму с помощью событийных процедур.** Для вывода изображений на форму в процессе выполнения программы необходимо воспользоваться событийной процедурой, содержащей функцию `LoadPicture`. Эта функция имеет синтаксис:

```
LoadPicture [(имя файла)]
```

Имя файла должно указывать на графический файл одного из форматов (ICO, BMP, WMF, JPG и др.). При отсутствии имени файла функция возвратит пустую рамку.

Если проект загружается из папки с проектом и если рисунок лежит в той же папке, что и файл проекта, можно указывать только имя файла изображения.

Если проект загружается из среды Visual Basic, то нужно указывать полный путь к файлу с изображением, т. е. аргументом функции LoadPicture должно являться полное имя файла (включая путь к этому файлу).

Будем использовать файл изображения, который хранится на CD-ROM, причем CD-ROM имеет логическое имя E:. Тогда полное имя файла с изображением монитора принимает следующий вид:

```
E:\practicum\VB\Projects\project2\monitor.gif.
```

Присвоим свойству Picture формы frm1 значение функции LoadPicture в событийной процедуре Click, связанной с кнопкой cmd1:

```
Private Sub cmd1_Click()  
    frm1.Picture=LoadPicture  
    ("E:\practicum\VB\Projects\project2\monitor.gif")  
End Sub
```

- 5 Разместить на форме кнопку и присвоить свойству Name значение cmdMon, а свойству Caption — значение Монитор.
- 6 Осуществить двойной щелчок по кнопке и в окне *Программный код* ввести текст событийной процедуры.
- 7 После запуска проекта щелкнуть по кнопке *Монитор*, на форме появится изображение монитора, которое размещается в верхнем левом углу формы.



- 8 Сохранить проект на диске в файле prj2.vbp.

После сохранения проекта на диске в папке проекта появится файл формы с расширением frx. Этот файл создается автоматически при размещении на форме изображения. Имя этого файла совпадает с именем файла формы.

**Вывод изображений на форму с использованием элемента управления Image.**

Элементы управления Image позволяют выводить на форму изображения из графических файлов различных типов (BMP, JPG, GIF и др.) в заданную область. Префикс для имен элементов Image — img.

Для того чтобы изображение автоматически выводилось после запуска проекта, необходимо в процессе разработки проекта воспользоваться свойством Picture элемента управления Image.

Для вывода изображений в процессе выполнения программы необходимо воспользоваться событийной процедурой, содержащей функцию LoadPicture.

Добавим в проект вторую форму и разместим на ней элемент управления Image. Установим автоматический вывод изображения в этот элемент.

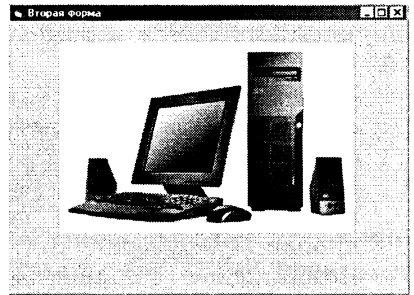
9 Добавить в проект форму командой [Project-Add Form].

10 Разместить в центре формы элемент управления Image, присвоить ему имя img1 и активизировать его.



11 В окне *Свойства объекта* для свойства Picture выбрать нужный графический файл (например, computer.gif).

12 Запустить проект на выполнение. Изображение выводится в элементе img1 в центре формы.



Запрограммируем теперь возможность вывода изображения процессора (файл processor.jpg) в элемент img1 в событийной процедуре:

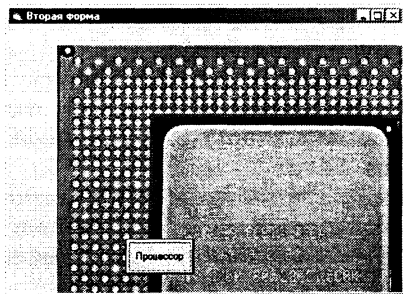
```
Private Sub cmdProc_Click()  
img1.Picture = LoadPicture  
("E:\practicum\VB\Projects\project2\  
processor.jpg")  
End Sub
```

13 Разместить на форме кнопку и присвоить свойству Name значение cmdProc, а свойству Caption — значение Процессор.

14 Осуществить двойной щелчок по кнопке и в окне *Программный код* ввести текст событийной процедуры.

- 15 Запустить проект на выполнение и щелкнуть по кнопке *Процессор*.

Изображение имеет гораздо большие размеры, чем размеры элемента Image и размеры самой формы, и поэтому видно лишь частично.



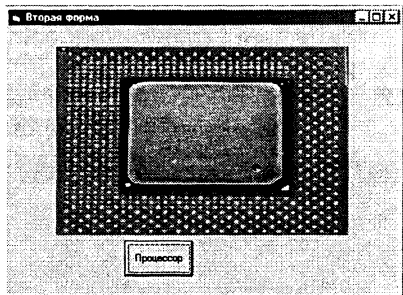
Для того чтобы сделать изображение видимым полностью, можно увеличить значения свойств Width и Height для элемента Image и формы.

Можно также сделать изображение полностью видимым, не меняя размеров элемента Image, воспользовавшись свойством Stretch (*подгонка*). По умолчанию свойству Stretch задано значению **False**. Когда значение свойства Stretch равно **True**, размер изображения подгоняется под размер элемента управления путем масштабирования. Если отношение высоты к ширине изображения отличается от отношения высоты к ширине элемента Image, то коэффициент масштабирования по высоте и ширине различается и изображение искажается.

- 16 Выбрать элемент img1 и в окне *Свойства объекта* свойству Stretch задать значение **True**.

- 17 Запустить проект на выполнение и щелкнуть по кнопке *Процессор*.

Изображение полностью размещено в элементе Image, но искажено.



Вывод изображений на форму с использованием элемента управления PictureBox. Элемент PictureBox (*графическое окно*) позволяет размещать графическую информацию из файлов различных типов (BMP, JPG, GIF и др.) в определенных участках формы. Префикс для имен элементов PictureBox — pic.

Элементы управления PictureBox позволяют не только выводить изображения, но и являются фактически «формами внутри форм», в которые можно помещать элементы управления, выводить текст и рисовать при помощи графических методов. Графические окна могут применяться также для перемещения изображений по форме. Они имеют больше возможностей, чем элементы Image, но требуют больше памяти и времени на обработку.

Для того чтобы изображение автоматически выводилось после запуска проекта, необходимо в процессе разработки проекта воспользоваться свойством `Picture` элемента управления `PictureBox`.

Для вывода изображений в процессе выполнения программы необходимо воспользоваться событийной процедурой, содержащей функцию `LoadPicture`.

Добавим в проект третью форму и разместим на ней элемент управления `PictureBox`, а внутри графического окна поместим метку с надписью *Компьютер*. Установим автоматический вывод изображения и метки в графическое окно.

18 Добавить в проект форму командой [Project-Add Form].

19 Разместить в центре формы элемент управления `Picture Box` и активизировать его. В окне *Свойства объекта* для свойства `Picture` выбрать нужный графический файл (например, `computer.gif`).



20 Разместить метку в графическом окне и для свойства `Caption` установить значение *Компьютер*.

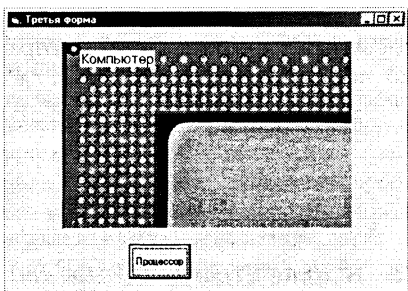
21 Запустить проект на выполнение. Изображение и метка выводятся в элементе `PictureBox`.



Запрограммируем теперь возможность вывода изображения процессора в элемент `PictureBox` в событийной процедуре, аналогичной рассмотренной выше.

22 Разместить на форме кнопку, осуществить двойной щелчок по кнопке и в окне *Программный код* ввести текст событийной процедуры (смотри пункты 13,14).

23 Запустить проект на выполнение и щелкнуть по кнопке *Процессор*. Изображение имеет гораздо большие размеры, чем размеры элемента `PictureBox` и поэтому будет видно лишь частично.





Можно сделать изображение полностью видимым, увеличив размеры элемента PictureBox, воспользовавшись свойством `AutoSize` (*авторазмер*). По умолчанию свойству `AutoSize` задано значению `False`. Если свойству `AutoSize` присвоить значение `True`, размеры элемента управления подгоняются под размеры изображения.

Теперь надпись на метке не соответствует выведенному в графическое окно рисунку. Практически все объекты, в том числе и метка, имеют свойство `Visible` (*видимый*), которое может принимать два значения: `True` (по умолчанию) и `False`.

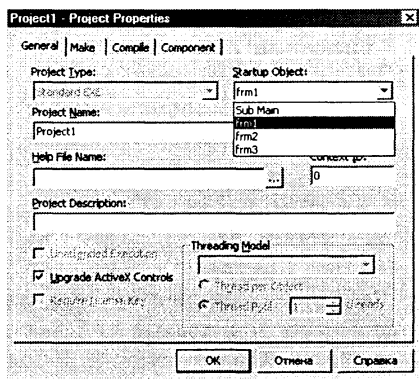
Если присвоить свойству `Visible` метки значение `False`, то метка станет невидимой.

**Организация переходов между формами, входящими в проект.** В наш проект входят три формы, необходимо создать возможность просмотра каждой из них.

Во-первых, можно установить в свойствах проекта, какая из входящих в него форм загружается первой при запуске проекта и соответственно становится видимой.

**24** Ввести команду [Project-Project1 Properties...].

На появившейся диалоговой панели *Project1-Project Properties* в раскрывающемся списке *Startup Object* выбрать форму *frm1*, которая будет загружаться первой.



Во-вторых, можно создать связанные с кнопками событийные процедуры, которые делают видимой ту или иную форму, используя свойство `Visible`.

Разместим на форме *frm1* кнопку *cmd2* и напишем для нее событийную процедуру, которая делает видимыми одновременно все три формы *frm1*, *frm2* и *frm3*:

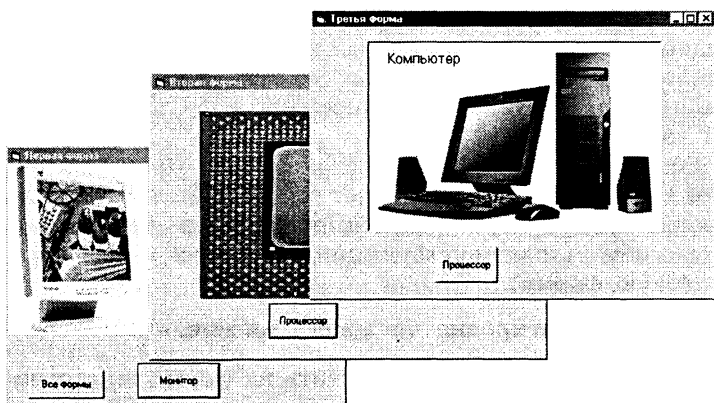
```
Private Sub cmd2_Click()
    frm1.Visible = True
    frm2.Visible = True
    frm3.Visible = True
End Sub
```

**25** В окне *Программный код* ввести событийную процедуру `cmd2_Click()` для кнопки *Все формы*.

26 Запустить проект на выполнение и щелкнуть по кнопке *Все формы*. На экране монитора появятся все три формы.

Для перехода с одной формы на другую событийная процедура должна делать видимой только одну выбранную форму. Например, для перехода со второй или третьей формы на первую событийная процедура будет следующей:

```
Private Sub cmd2_Click()  
frm1.Visible = True  
frm2.Visible = False  
frm3.Visible = False  
End Sub
```



27 Сохранить полученный проект в файле `prj2.vbp`.

Проект хранится в каталоге  
`\practicum\VB\Projects\project2\` CD-ROM



### Задания для самостоятельного выполнения

- 5.7. Создать проект «Видимость метки», в котором по щелчку по одной кнопке метка выводится на форму, а по щелчку по второй кнопке — исчезает.
- 5.8. Создать проект «Видимость формы», в котором по щелчку по кнопке на первой форме выводится вторая форма, и наоборот.
- 5.9. Усовершенствовать проект «Компьютер в картинках» (файл проекта `prj2.vbp`) так, чтобы обеспечить возможность перехода с каждой из трех форм на любую другую в процессе выполнения проекта.
- 5.10. Усовершенствовать проект 5.9 (файл проекта `prjZ5_9.vbp`) так, чтобы обеспечить вывод изображения сканера в элемент `Image` на вторую форму и вывод изображения цифровой камеры в элемент `PictureBox` на третью форму. Предусмотреть соответствие надписи на метке выводимому изображению.

- 5.11. Создать проект «Собери картинку». Изображение разбить на фрагменты и неупорядоченно вывести их на форму. Предусмотреть возможность вывода фрагментов на форму таким образом, чтобы получилось первоначальное изображение.

## 5.4. Массивы

**Проект 5.3. «Сортировка числового массива».** Разработаем проект, который позволит заполнять числовой массив целыми случайными числами, а затем предоставит возможность сортировать массив различными способами.

**Заполнение массива случайными числами.** Для генерации последовательности целых случайных чисел используем функцию генерации псевдослучайных (то есть каждый раз повторяющихся) последовательностей чисел `Rnd` и функцию выделения целой части числа `Int`.

Заполнение массива `bytA`, состоящего из 10 элементов, реализуем с помощью цикла со счетчиком. В целях визуального контроля предусмотрим вывод массива в текстовое поле `txtDim` с использованием функции `Str` (преобразования числа в строковую форму).

### Проект «Сортировка числового массива»

- 1 Создать новый проект. Разместить на форме текстовое поле `txtDim` и кнопку `cmdDim`. Присвоить свойству `Caption` значение `Заполнить`.
- 2 Установить для свойства `MultiLine` текстового поля значение `True` и подобрать размер шрифта так, чтобы массив распечатывался в одной строке текстового поля.
- 3 Ввести для кнопки событийную процедуру заполнения массива и его вывода в текстовое поле:

```
Dim bytI, bytA(1 To 10) As Byte
Private Sub cmdDim_Click()
Randomize
For bytI = 1 To 10
bytA(bytI) = Int(Rnd * 10)
txtDim.Text = txtDim.Text + Str(bytA(bytI))
Next bytI
End Sub
```

- 4 Запустить проект и несколько раз щелкнуть по кнопке *Заполнить*.

В текстовом поле появятся последовательности случайных чисел, различающиеся между собой.



**Поиск минимального элемента.** Будем считать сначала, что минимальный элемент равен первому элементу массива `bytA(1)` и присвоим переменной `bytMin` его значение.

Затем в цикле сравним последовательно элементы массива со значением переменной `bytMin`, если какой-либо элемент окажется меньше, присвоим его значение переменной `bytMin`, а его индекс присвоим переменной `bytN`. Распечатаем результаты.

- 5 Разместить на форме кнопку `cmdMin` и присвоить свойству `Caption` значение `Мин.элемент`. Создать для нее событийную процедуру поиска минимального элемента:

```
Private Sub cmdMin_Click()
    bytMin = bytA(1)
    bytN = 1
    For bytI = 2 To 10
        If bytA(bytI) < bytMin Then bytMin =
            bytA(bytI): bytN = bytI
    Next bytI
    Print "Минимальный элемент "; bytMin;
        "его индекс"; bytN
End Sub
```

- 6 Запустить проект. Последовательно щелкать на кнопках *Заполнить* и *Мин. элемент*.

В текстовом окне будет появляться заполненный массив, а на форме будет печататься минимальный элемент и его индекс.



**Сортировка массива по возрастанию методом перестановок.** Идея алгоритма сортировки по возрастанию массива, состоящего из  $N$  элементов, методом перестановок элементов состоит в

последовательном поиске минимальных элементов и их перемещении к началу массива с помощью перестановок с элементами, имеющими меньший индекс.

На первом шаге находим минимальный элемент массива и меняем его местами с первым элементом массива. Остаются неупорядоченными  $N-1$  элемент. Проводим поиск минимального элемента среди элементов с индексами от 2 до  $N$  и делаем перестановку.

Повторяем процедуру поиска минимального элемента среди оставшихся неупорядоченных элементов многократно. Повторение реализуем с помощью цикла со счетчиком, максимальное значение которого составляет  $N-1$ . В результате массив сортируется по возрастанию.

Так как поиск минимального элемента массива проводится многократно, реализуем рассмотренную ранее событийную процедуру `cmdMin_Click()` как общую процедуру `МинЭлемент` (`bytI, bytN As Byte`), где `bytI` — является входным параметром, а `bytN` — выходным параметром.

7 Ввести программный код общей процедуры поиска минимального элемента и поместить ее в начало программного кода проекта. Объявить переменные, используемые во всех процедурах:

```
Dim bytA(1 To 10), bytMin, bytI, bytJ, bytK,
    bytR, bytN As Byte,
'общая процедура поиска минимального элемента
Sub МинЭлемент(bytI, bytN As Byte)
    bytMin = bytA(bytI)
    bytN = bytI
    For bytJ = bytI + 1 To 10
        If bytA(bytJ) < bytMin
            Then bytMin = bytA(bytJ) : bytN = bytJ
        Next bytJ
    End Sub
```

Событийная процедура, реализующая сортировку массива, будет включать в себя вызов общей процедуры поиска минимального элемента и код перестановки элементов.

Процесс сортировки массива можно сделать наглядным, если в текстовое поле выводить состояние массива на каждом шаге сортировки.

8 Поместить на форму текстовое поле `txtSort` и кнопку `cmdSort`. Присвоить свойству кнопки `Caption` значение `Сортировать`.

- 9 Ввести программный код событийной процедуры сортировки массива:

*'событийная процедура сортировки*

```
Private Sub cmdSort_Click()
```

```
txtSort.Text = ""
```

```
For bytI = 1 To 9
```

*'вызов общей процедуры поиска минимального элемента*

```
Call МинЭлемент(bytI, bytN)
```

*'перестановка*

```
bytR = bytA(bytI)
```

```
bytA(bytI) = bytA(bytN)
```

```
bytA(bytN) = bytR
```

*'печать массива для каждого цикла сортировки*

```
For bytK = 1 To 10
```

```
txtSort.Text = txtSort.Text + Str(bytA(bytK))
```

```
Next bytK
```

```
Next bytI
```

```
End Sub
```

- 10 Для большей наглядности разместить на форме две метки с номерами элементов массива (горизонтальная) и значениями счетчика цикла сортировки (вертикальная).
- 11 Запустить проект на выполнение. Щелкнуть по кнопкам *Заполнить массив*, *Мин. элемент* и *Сортировка*.
- 12 Процедуру можно производить многократно и наблюдать процесс сортировки для различных вариантов заполнения массива.



- 13 Сохранить проект в файле `prj3.vbp`.

Проект хранится в каталоге  
`\practicum\VB\Projects\project3\`

CD-ROM 

Проект 5.4. «Сортировка строкового массива». Разработаем проект, который позволяет сортировать строковый массив (например, содержащий компьютерные термины) по алфавиту.

**Заполнение строкового массива.** Как строковые, так и числовые массивы можно заполнять разными методами. Рассмотренный ранее числовой массив был заполнен непосредственно в самой программе. Можно заполнять массивы, вводя данные с клавиатуры с использованием функции *InputBox*, однако строковые массивы чаще всего заполняются путем считывания данных из текстовых файлов.

Такие текстовые файлы должны содержать только коды самих символов (не должны содержать управляющие коды форматирования текста, тэги языка HTML и так далее) и, следовательно, должны создаваться в простейших текстовых редакторах (типа Блокнот).

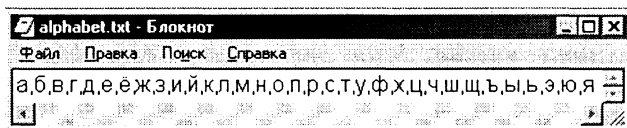
В таких файлах элементы данных (последовательности символов) отделяются друг от друга с помощью разделителей (чаще всего запятой).

Создадим два текстовых файла данных: первый, содержащий буквы алфавита, а второй — содержащий компьютерные термины.

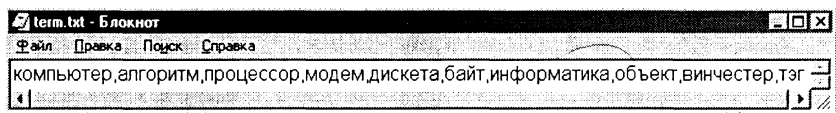


### Проект «Сортировка строкового массива»

- 1 Запустить Блокнот, ввести буквы русского алфавита и сохранить файл под именем `alphabet.txt`.



- 2 Создать файл, ввести компьютерные термины и сохранить под именем `term.txt`.



Прежде чем приступить к считыванию данных из текстового файла в строковый массив (или последовательность строковых переменных), необходимо открыть файл с помощью специального оператора **Open**:

```
Open Имя_файла For Режим_работы
As # Номер_файла
```

Имя\_файла должно содержать путь к файлу и собственно имя файла, Режим\_работы определяет, что файл открыт для считывания данных (**Input**) или, наоборот, для записи данных в файл (**Output**), а Номер\_файла содержит целое число в интервале от 1 до 511.

Например, чтобы открыть файл `alphabet.txt`, находящийся на CD-ROM, для считывания данных, необходимо ввести следующий код:

```
Open "E:\practicum\VB\Projects\project4\  
alphabet.txt" For Input As #1
```

Для чтения данных из файла используется функция `Input`:

```
Input # Номер_файла, Список_переменных
```

После завершения считывания данных файл необходимо закрыть с помощью оператора `Close`:

```
Close # Номер_файла
```

Произведем чтение данных из файла `alphabet.txt` в строковый массив с помощью цикла со счетчиком.

3 Создать проект, на форму поместить кнопку `cmdAlph` (здать свойству `Caption` значение `Считать алфавит`) и текстовое поле `txtAlph`, в котором можно наблюдать результат заполнения массива.

4 Объявить переменные, используемые во всех процедурах:

```
Dim bytI, bytK As Byte, strA(1 To 33),  
strB(1 To 10), strC(1 To 10) As String
```

Создать событийную процедуру считывания данных из файла `alphabet.txt` в строковый массив `strA`:

```
Private Sub cmdAlph_Click()  
Open "E:\practicum\VB\Projects\project4\  
alphabet.txt" For Input As #1  
For bytI = 1 To 33  
Input #1, strA(bytI)  
txtAlph.Text = txtAlph.Text + strA(bytI)  
Next bytI  
Close #1  
End Sub
```


Произведем теперь чтение данных из файла `term.txt` в строковый массив с помощью цикла с предусловием. Это вариант используется, если заранее неизвестно количество элементов данных в файле, и в качестве условия используется функция `EOF` (`End Off File`):

```
EOF (Номер_файла)
```

Для визуализации заполнения строкового массива терминами удобно использовать управляющий элемент *Окно списка* (`ListBox`), который представляет собой упорядоченный список значений.

Для размещения с окне списка нового элемента использует метод `AddItem`, который имеет в качестве аргумента строковую переменную, значение которой и добавляется в список.



- 5 Поместить на форму кнопку cmdTerm (задать свойству Caption значение **Считать термины**) и окно списка lstTerm, в котором можно наблюдать результат заполнения массива. 
- 6 Создать событийную процедуру считывания данных из файла term.txt в строковый массив strB и его отображения в окне списка lstTerm:

```
Private Sub cmdTerm_Click()
Open "E:\practicum\VB\Projects\project4\term.txt"
  For Input As #2
bytI = 1
Do Until EOF(2)
Input #2, strB(bytI)
lstTerm.AddItem strB(bytI)
bytI = bytI + 1
Loop
Close #2
End Sub
```

**Сортировка массива.** В исходном строковом массиве strB термины хранятся в неупорядоченном виде. Сравним последовательно во внутреннем цикле со счетчиком bytI букву «а» алфавита с первыми символами терминов, в случае их совпадения присвоим первому элементу результирующего символьного массива strC это значение. Повторим эту процедуру последовательно для всех букв алфавита с использованием внешнего цикла со счетчиком bytK. Получим упорядоченный строковый массив strC.

- 7 Поместить на форму кнопку cmdSort (задать свойству Caption значение **Сортировать**) и создать событийную процедуру сортировки массива:

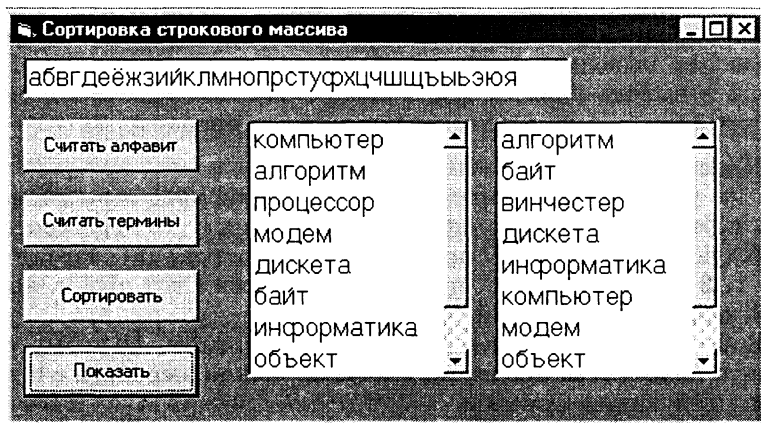
```
Private Sub cmdSort_Click()
bytJ = 1
For bytK = 1 To 33
For bytI = 1 To 10
If strA(bytK) = Left$(strB(bytI), 1) Then
  strC(bytJ) = strB(bytI): bytJ = bytJ + 1
Next bytI
Next bytK
End Sub
```

**Визуализация отсортированного массива.** Для визуализации отсортированного массива используем второе окно списка.

- 8 Поместить на форму кнопку cmdRez (задан свойству Caption значение Показать), окно списка lstSort и создать событийную процедуру визуализации отсортированного массива:

```
Private Sub cmdRez_Click()  
For bytJ = 1 To 10  
lstSort.AddItem strC(bytJ)  
Next bytJ  
End Sub
```

- 9 Запустить проект на выполнение. Последовательно щелкнуть по кнопкам *Считать алфавит*, *Считать термины*, *Сортировать*, *Показать*.



- 10 Сохранить проект в файле prj4.vbp.

Проект хранится в каталоге  
\\practicum\VB\Projects\project4\ CD-ROM 

### Задания для самостоятельного выполнения

- 5.12. Создать проект, реализующий поиск максимального элемента числового массива.
- 5.13. Создать проект, реализующий сортировку числового массива по убыванию методом перестановок.
- 5.14. Создать проект, реализующий сортировку массива по возрастанию «пузырьковым» методом.
- 5.15. Усовершенствовать проект 5.4 «Сортировка строкового массива». Предусмотреть возможность сохранения отсортированного массива в файле.
- 5.16. Создать проект «Словарь компьютерных терминов», который выводит определение термина, заданного пользователем. Термины и их определения хранить в файлах данных.

- 5.17. Создать проект «Русско-английский и англо-русский словарь», который обеспечивает перевод слов, хранящихся в файлах данных.
- 5.18. Создать проект «Проверка знаний» так, чтобы вопросы и правильные ответы хранились в файлах данных, вопросы задавались и ответы вводились с использованием цикла, а ответы учащегося сохранялись в файле.

## 5.5. Ввод и вывод данных

Ввод и вывод данных в процессе выполнения проекта может производиться различными способами. Функция `InputBox` (*Окно Ввода*) позволяет вводить данные с помощью диалоговой панели ввода, а функция `MsgBox` (*панель сообщений*), наоборот, позволяет выводить сообщения на специальной панели. Для ввода и вывода данных могут использоваться также `TextBox` (*текстовые поля*), а для печати данных оператор `Print`.

**Проект 5.5. «Игра Баше».** Разработаем проект, который позволит реализовывать игру Баше, в которой ввод и вывод данных осуществляется различными способами. Суть игры состоит в следующем: имеется  $N$  предметов, а два игрока по очереди берут 1, 2 или 3 предмета, причем проигрывает тот игрок, который забирает последний предмет.

В начале игры и после каждого хода существуют четыре варианта количества предметов, которое можно выразить с помощью следующих формул:

$$N = 4 \cdot K; \quad N = 4 \cdot K + 1; \quad N = 4 \cdot K + 2; \quad N = 4 \cdot K + 3,$$

где  $K = 0, 1, 2, \dots$

Выигрышная стратегия состоит в том, чтобы после своего хода осталось 1, 5, 9, 13 и так далее предметов, то есть, в общем случае, должно оставаться  $4 \cdot K + 1$  предмет, где  $K = 0, 1, 2, \dots$  Таким образом, в каждом из вышеприведенных случаев необходимо взять на своем ходе определенное количество предметов:

- при  $N = 4 \cdot K$  — взять 3 предмета;
- при  $N = 4 \cdot K + 1$  не существует 100% выигрышной стратегии, поэтому можно брать произвольное количество предметов, оставляя возможность осуществить выигрышную стратегию другому игроку;
- при  $N = 4 \cdot K + 2$  — взять 1 предмет;
- при  $N = 4 \cdot K + 3$  — взять 2 предмета.

Создадим проект, реализующий игру Баше, в котором одним игроком является компьютер, а другим — пользователь. Первоначальное случайное целочисленное количество предме-

тов (например, в интервале от 10 до 20) получим с помощью оператора `Randomize`, функции генератора случайных чисел `Rnd` и функции выделения целой части числа `Int`. Присвоим полученное значение переменной `bytN`. Выводить информацию о количестве предметов будем с помощью оператора `Print`.



### Проект «Игра Баше»

- 1 Поместить на форму кнопку и начать создание событийной процедуры `cmd1_Click()`.

```
Private Sub cmd1_Click()  
Randomize  
bytN = Int(10 + 10 * Rnd)  
Print "Первоначальное количество предметов";  
bytN  
End Sub
```

Для информации пользователя о количестве предметов и получении его согласия на начало игры воспользуемся функцией вывода `MsgBox`.

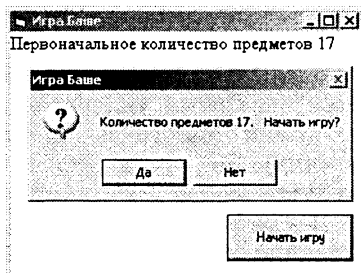
- 2 Ввести в код событийной процедуры строки:

```
bytB = MsgBox("Количество предметов" +  
Str(bytN) + "." + " Начать игру?",  
36, "Игра Баше")  
If bytB = 7 Then End
```

- 3 Запустить проект щелчком по кнопке *Начать игру*.

На форме и на панели *Игра Баше* будет напечатано первоначальное количество предметов.

Щелчок по кнопке *Да* позволит начать игру.



Процесс поочередного взятия предметов компьютером и пользователем должен продолжаться, пока не останется хотя бы один предмет. Удобнее всего использовать для этого цикл с предусловием:

```
Do While bytN > 1 - Loop.
```

Для подсчета сделанных ходов целесообразно использовать специальную переменную `bytI`:

```
bytI = bytI + 1
```

Пусть первый ход делает пользователь и вводит количество взятых предметов с использованием функции ввода `InputBox`, которое запоминается в строковой переменной `strX`:

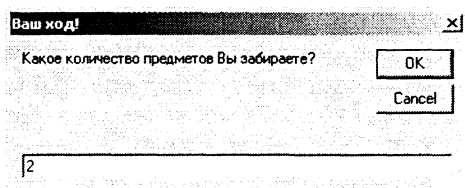
```
strX = InputBox("Какое количество предметов
Вы забираете?", "Ваш ход!")
```

Для определения количества оставшихся после хода пользователя предметов необходимо перевести строковую переменную в числовую форму с помощью функции `Val` и вычесть ее из переменной `bytN` (первоначального количества предметов):

```
bytN = bytN - Val(strX)
```

#### 4 Ввести в программный код строки:

```
Do While bytN > 1
bytI = bytI + 1
strX = InputBox
("Какое количество
предметов Вы забираете?", "Ваш ход!")
bytN = bytN - Val(strX)
Loop
```



При выборе выигрышной стратегии компьютер при ответном ходе должен учитывать оставшееся количество предметов. Проще всего это реализовать с использованием управляющей инструкции выбора. Инструкция начинается с ключевых слов **Select Case**, после которых записывается переменная. После ключевых слов **Case Is** заданная переменная сравнивается с определенными значениями (записываются условия, при выполнении одного из которых начинает выполняться серия команд). Заканчивается инструкция ключевыми словами **End Select**.

В каждой ветви инструкции выбора необходимо задать уменьшение количества предметов на определенную величину, а также печать оставшегося количества на форме с помощью оператора **Print**.

5 Для перебора возможных значений количества предметов инструкцию **Select Case** необходимо поместить внутрь цикла со счетчиком по переменной `bytK`, который, в свою очередь, поместить внутрь цикла с послеусловием:

```
For bytK = 0 To 6
Select Case bytN
Case Is = 4 * bytK
bytN = bytN - 3
Print "После"; bytI; "хода осталось предметов";
```

```
bytN
Case Is = 4 * bytK + 1
bytN = bytN - Int(1 + 2 * Rnd)
Print "После"; bytI; "хода осталось предметов";
bytN
Case Is = 4 * bytK + 2
bytN = bytN - 1
Print "После"; bytI; "хода осталось предметов";
bytN
Case Is = 4 * bytK + 3
bytN = bytN - 2
Print "После"; bytI; "хода осталось предметов";
bytN
End Select
Next bytK
```

С помощью операторов условного перехода необходимо отслеживать ход, после которого остается 1 предмет. При выполнении этого условия с помощью панели сообщений в операторной (безскобочной) форме можно предусмотреть вывод результатов игры, а с помощью оператора **End** — окончание игры.

#### 6 До цикла со счетчиком ввести строку:

```
If bytN = 1 Then MsgBox "После Вашего  
хода остался 1 предмет. Компьютер проиграл!", ,  
"Игра Баше": End
```

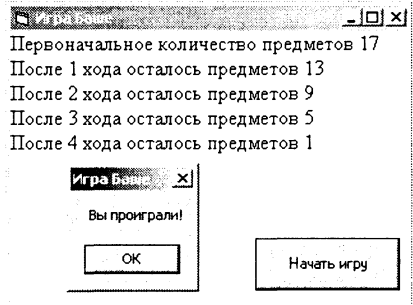
После цикла со счетчиком ввести строку:

```
If bytN = 1 Then MsgBox "После хода компьютера  
остался 1 предмет. Вы проиграли!", , "Игра  
Баше": End
```


#### 7 Щелкнуть по кнопке *Начать игру*. Последовательно с помощью панелей ввода ввести значения количеств забираемых предметов на каждом ходе.

Остающееся количество предметов после каждого хода будет печататься на форме.

Результат игры будет выведен с помощью панели сообщений.



## 8 Сохранить проект в файле prj5.vbp.

*Проект хранится в каталоге*  
 \practicum\VB\Projects\project5\ CD-ROM 

### Задания для самостоятельного выполнения

- 5.19. Модернизировать проект «Игра Баше» так, чтобы вывод количества предметов осуществлялся с помощью текстового поля.
- 5.20. Модернизировать проект «Игра Баше» так, чтобы первый ход делал компьютер.

## 5.6. Арифметические операции и функции преобразования типов данных

**Проект 5.6. «Мультисистемный калькулятор».** Для создания мультисистемного калькулятора воспользуемся возможностью проведения арифметических операций в десятичной системе счисления и функциями преобразования типов данных, позволяющими осуществлять перевод чисел из десятичной системы счисления в восьмеричную и шестнадцатеричную системы и, наоборот, из восьмеричной и шестнадцатеричной систем в десятичную.

Сначала создадим калькулятор для работы в десятичной системе. На форме разместим три текстовых поля: два поля txt1Dec и txt2Dec для ввода числовых данных и одно txt3Dec для вывода результата. Пять кнопок для реализации событийных процедур: сложения cmdPlus, вычитания cmdMinus, умножения cmdUmn, деления cmdDelen и завершения работы cmdExit.

**Событийные процедуры арифметических операций.** Событийная процедура сложения cmdPlus\_Click должна изменить свойство Text текстового поля txt3Plus так, чтобы оно являлось суммой числовых значений свойства Text объектов txt1Dec и txt2Dec. Для преобразования строковых значений, вводимых в текстовые поля, в десятичное число, воспользуемся функцией Val. Код событийной процедуры будет следующим:

```
Private Sub cmdPlus_Click()
    txt3Dec.Text = Val(txt1Dec.Text) +
    Val(txt2Dec.Text)
End Sub
```

Событийные процедуры вычитания, умножения и деления создаются аналогично.



## Проект «Мультисистемный калькулятор»

- 1 Создать новый проект. Разместить на форме три текстовых поля и пять кнопок. Присвоить им имена: txt1Dec, txt2Dec, txt3Dec, cmdPlus, cmdMinus, cmdUmn, cmdDelen, cmdExit.
- 2 Для каждой из кнопок ввести программные коды событийных процедур.
- 3 Установить для свойства Alignment текстовых полей значение Right Justify.
- 4 Запустить проект на выполнение. Ввести числа в два первых текстовых поля. Щелкнуть по любой кнопке арифметических операций и соответствующая операция будет выполнена.



Перевод целых положительных чисел из десятичной системы счисления в восьмеричную с помощью функции Oct. Для перевода чисел из десятичной системы счисления в восьмеричную воспользуемся функцией Oct, аргументом которой являются десятичные числа, а значением восьмеричные числа в строковой форме.

Введенные в десятичные поля числа переведем сначала из строковой формы в числовую с помощью функции Val, а затем из десятичной числовой переведем в строковую восьмеричную с помощью функции Oct. Создадим кнопку cmdDecOct и запишем для нее событийную процедуру перевода чисел:

```
Private Sub cmdDecOct_Click()  
txt1Oct.Text = Oct(Val(txt1Dec.Text))  
txt2Oct.Text = Oct(Val(txt2Dec.Text))  
txt3Oct.Text = Oct(Val(txt3Dec.Text))  
End Sub
```

- 5 Добавить на форму три текстовых поля txt1Oct, txt2Oct и txt3Oct.
- 6 Поместить на форму кнопку cmdDecOct и ввести программный код событийной процедуры cmdDecOct\_Click().



- 7 Поместить на форму две метки с надписями *Десятичная система* и *Восьмеричная система*.
- 8 Запустить проект на выполнение. Ввести числа в два первых текстовых поля десятичной системы. Щелкнуть по кнопке «+», а затем *Dec-Oct*. Будет выполнено сложение десятичных чисел и их перевод в восьмеричную систему.



Перевод целых положительных чисел из восьмеричной системы счисления в десятичную с помощью функции `Val`. Для перевода чисел из восьмеричной системы счисления в десятичную воспользуемся функцией `Val`, аргументом которой являются восьмеричные числа в строковой форме, а значением десятичные числа.

Введенные в восьмеричные поля в строковой форме числа переведем сначала в восьмеричное представление добавлением приставки "`&O`" с помощью операции конкатенации строк. Полученное восьмеричное число в строковой форме переведем в десятичную числовую с помощью функции `Val`. Создадим кнопку `cmdOctDec` и запишем для нее событийную процедуру перевода чисел:

```
Private Sub cmdOctDec_Click()
    txt1Dec.Text = Val("&O" + txt1Oct.Text)
    txt2Dec.Text = Val("&O" + txt2Oct.Text)
    txt3Dec.Text = Val("&O" + txt3Oct.Text)
End Sub
```

- 9 Поместить на форму кнопку `cmdOctDec` и ввести для нее программный код событийной процедуры перевода чисел.
- 10 Запустить проект на выполнение. Ввести числа в два первых текстовых поля восьмеричной системы. Щелкнуть по кнопке *Oct-Dec*. Будет выполнен их перевод в десятичную систему.



11 Выполнить арифметическую операцию в десятичной системе, щелкнув, например, по кнопке «-».

Перевести результат в восьмеричную систему, щелкнув по кнопке *Dec-Oct*.

12 Сохранить проект в файле *prj6.vbp*.

*Проект хранится в каталоге  
\\practicum\VB\Projects\project6\*

CD-ROM



### Задания для самостоятельного выполнения

- 5.21. Усовершенствовать проект «Мультисистемный калькулятор» (файл проекта *prj6.vbp*), расширив его функциональные возможности. Обеспечить возможность перевода чисел из десятичной системы счисления в шестнадцатеричную и обратно.
- 5.22. Преобразовать проект «Мультисистемный калькулятор» в проект «Инженерный калькулятор», который позволяет вычислять значение тригонометрических и других функций.
- 5.23. Создать проект «Подсчет цифр в числе», который позволяет определить количество цифр во введенном натуральном десятичном числе.
- 5.24. Модернизировать проект 5.23 (файл проекта *prjZ5\_23*), так чтобы можно было определить количество цифр во введенном целом десятичном числе.
- 5.25. Создать проект «Печать цифр числа», который обеспечивает вывод на печать (по зонам) цифр, составляющих введенное целое десятичное число.
- 5.26. Создать проект «Сумма цифр», который позволяет вычислить сумму десятичных цифр заданного целого десятичного числа.
- 5.27. Создать проект «Числа без одинаковых цифр», который печатает в возрастающей последовательности натуральные двухзначные числа, в которых нет одинаковых цифр.
- 5.28. Создать проект «Перевертыш чисел», который заменяет заданное натуральное число на число, которое получается из исходного путем записи его цифр в обратном порядке.
- 5.29. Создать проект «Перевод натуральных десятичных чисел в двоичную систему счисления».

## 5.7. Строковые операции и функции

Для обработки строк, то есть последовательностей символов, и строковых переменных (тип `String`) используются операция конкатенации (сложения строк) и строковые функции.

**Проект 5.7. «Строковый калькулятор».** Создадим строковый калькулятор, который позволит производить различные преобразования строк.

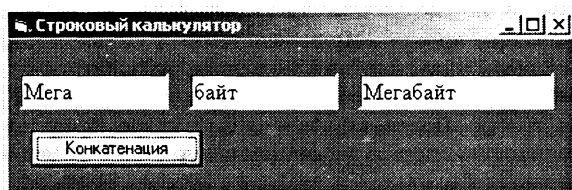


### Проект «Строковый калькулятор»

- 1 Создать новый проект. Разместить на форме три текстовых поля и кнопку. Присвоить им имена: `txt1`, `txt2`, `txt3`, `cmdCon`.
- 2 Для кнопки ввести программный код событийной процедуры `cmdCon_Click()`, реализующий операцию конкатенации:

```
Private Sub cmdCon_Click()
txt3.Text = txt1.Text + txt2.Text
End Sub
```

- 3 Запустить проект, в два первых поля ввести строки и щелкнуть по кнопке *Конкатенация*. В третьем поле появится результат сложения двух строк.

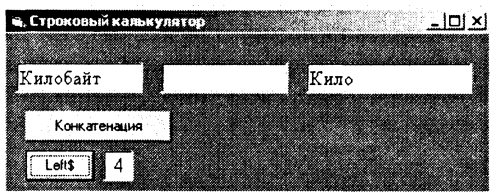


Воспользуемся теперь для преобразования строк строковой функцией `Left$(строка$, N)`. Функция вырезает из строки \$ подстроку, содержащую N символов, начиная с первого символа. В качестве строки будем рассматривать значение свойства первого текстового поля `txt1.Text`, а количество символов будет задаваться в текстовом поле `txtLeft.Text` и преобразовываться в числа с помощью функции `Val`.

- 4 Разместить на форме кнопку и текстовое поле. Присвоить им имена `cmdLeft` и `txtLeft`.
- 5 Для кнопки ввести программный код событийной процедуры `cmdLeft_Click()`, реализующий операцию вырезания подстроки:

```
Private Sub cmdLeft_Click()
txt3.Text = Left$(txt1.Text, Val(txtLeft.Text))
End Sub
```

- 6 Запустить проект, в первое поле ввести строку, в поля функции вырезки подстроки ввести числа и щелкнуть по кнопке *Left\$*. В третьем поле появится вырезанная подстрока.



Аналогично создается событийная процедура, позволяющая реализовать вырезку подстроки, начиная с последнего правого символа строки, с использованием функции *Right\$(строка\$, N)*.

- 7 Разместить на форме кнопку и текстовое поле *cmdRight* и *txtRight* и ввести программный код событийной процедуры *cmdLeft\_Click()*:

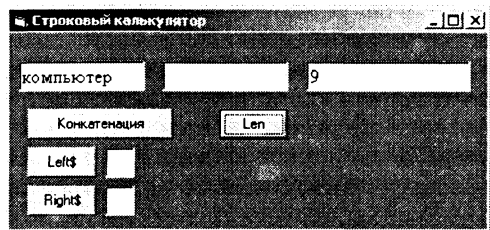
```
Private Sub cmdRight_Click()
    txt3.Text = Right$(txt1.Text,
    Val(txtRight.Text))
End Sub
```

Для определения количества символов в строке использует функция определения длины строки *Len(строка\$)*, аргументом которой является строка, а возвращает функция число, равное количеству символов в строке.

- 8 Разместить на форме кнопку и присвоить ей имя *cmdLen*.  
 9 Для кнопки ввести программный код событийной процедуры *cmdLen\_Click()*, реализующий операцию определения количества символов в строке:

```
Private Sub cmdLen_Click()
    txt3.Text = Len(txt1.Text)
End Sub
```

- 10 Запустить проект, в первое поле ввести строку и щелкнуть по кнопке *Len*. В третьем поле появится число символов в строке.

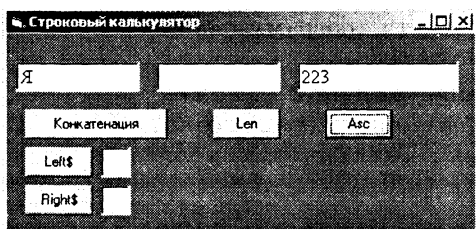


Для преобразования символов в соответствующий им числовой код используется функция `Asc(строка$)`, при этом необходимо иметь в виду, что функция возвращает числовой код первого символа строки.

- 11 Разместить на форме кнопку и присвоить ей имя `cmdAsc`.
- 12 Для кнопки ввести программный код событийной процедуры `cmdAsc_Click()`, реализующий операцию определения числового кода символа:

```
Private Sub cmdAsc_Click()
txt3.Text = Asc(txt1.Text)
End Sub
```

- 13 Запустить проект, в первое поле ввести символ и щелкнуть по кнопке `Asc`. В третьем поле появится числовой код символа.



- 14 Сохранить проект в файле `prj7.vbp`.

*Проект хранится в каталоге  
\practicum\VB\Projects\project7\*

CD-ROM 

**Проект 5.8. «Шифровка и дешифровка текстовых сообщений».** Данный проект позволит зашифровывать и расшифровывать текстовые сообщения. Используем алгоритм шифрования, который базируется на использовании ключевой фразы (ключа шифрования).

Пусть имеется некоторый секретный текст, который нужно зашифровать. В процессе шифровки необходимо заменить каждый символ секретного текста на порядковый номер этого символа в ключевой фразе.

При дешифровке последовательность чисел должна преобразовываться обратно в секретный текст с помощью той же ключевой фразы. Ключевая фраза должна содержать все символы, которые будут использоваться в сообщениях, и должна быть известна только шифровальщику и получателю сообщений.

Исходный секретный текст, зашифрованное и дешифрованное сообщения будем вводить и выводить в текстовых полях, а для запуска процедур шифровки и дешифровки используем кнопки.

 Проект «Шифровка и дешифровка текстовых сообщений»

- 1 Разместить на форме три текстовых поля и две кнопки. Присвоить им имена: `txtSecrtext`, `txtCod`, `txtDecod`, `cmdCod`, `cmdDecod`.

Сначала составим процедуру шифровки текстовых сообщений. Пусть ключевая фраза будет значением строковой переменной `strKF`, а секретный текст значением свойства `Text` текстового поля `txtSecrtext`.

Будем последовательно вырезать в цикле со счетчиком `bytI` символы из секретного текста, а затем во вложенном цикле со счетчиком `bytJ` будем искать такой же символ в ключевой фразе и запоминать его порядковый номер в целочисленном массиве `bytZ`. Таким образом, секретный текст будет зашифрован последовательностью чисел, хранящейся в целочисленном массиве.

- 2 Для кнопки `cmdCod` ввести программный код событийной процедуры `cmdCod_Click()`, реализующий шифрование текстового сообщения, введенного в текстовое поле `txtSecrtext`. Секретная фраза (значение переменной `strKF`) должна быть записана в программе в одну строку (без переносов).

```
Dim bytI, bytJ, bytZ(1 To 200) As Byte, strKF, strA, strB As String
Private Sub cmdCod_Click()
    strKF = "электронно-вычислительные машины, компьютеры (железо) позволяют находить значения функций, а ещё свойства готовых объектов"
    For bytI = 1 To Len(txtSecrtext.Text)
        strA = Mid$(txtSecrtext.Text, bytI, 1)
        For bytJ = 1 To Len(strKF)
            If Mid$(strKF, bytJ, 1) = strA Then
                bytZ(bytI) = bytJ: bytJ = Len(strKF) + 1
            Next bytJ
        txtCod.Text = txtCod.Text + Str(bytZ(bytI))
    Next bytI
End Sub
```

Рассмотрим теперь алгоритм дешифровки. Итак, в нашем распоряжении шифровка в форме целочисленного массива `bytZ`, которая выведена в текстовое окно `txtCod`. Необходимо с помощью ключевой фразы превратить эту числовую последовательность в секретный текст.

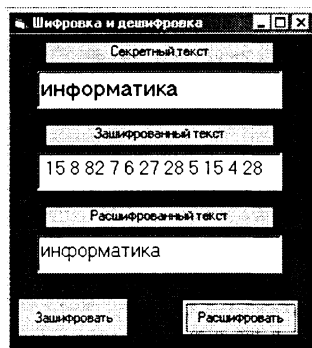
Считывая по очередности числа шифровки (элементы целочисленного массива `bytZ`), необходимо вырезать из ключевой фразы символ, номер которого по порядку равен значению числа шифровки. Полученный символ последовательно присо-


единяется к значению `txtDecod.Text`, которое и является исходным секретным текстом.

- 3 Для кнопки `cmdDecod` ввести программный код событийной процедуры `cmdDecod_Click()`, реализующий расшифровку текстового сообщения, и его вывод в текстовое поле `txtDecod`:

```
Private Sub cmdDecod_Click()
For bytI = 1 To Len(txtSecrtext.Text)
strB = Mid$(strKF, bytZ(bytI), 1)
txtDecod.Text = txtDecod.Text + strB
Next bytI
End Sub
```

- 4 Разместить на форме три метки с надписями *Секретный код*, *Зашифрованный текст* и *Расшифрованный текст*, которые сделают интерфейс проекта более наглядным.
- 5 Запустить проект на выполнение, ввести в верхнее текстовое поле секретный текст.
- 6 Щелкнуть по кнопке *Зашифровать*, в среднем текстовом поле появится шифровка в виде числовой последовательности.
- 7 Щелкнуть по кнопке *Расшифровать*, в нижнем текстовом поле должен появиться исходный секретный текст.
- 8 Сохранить проект в файле `prj8.vbp`.



Проект хранится в каталоге `\practicum\VB\Projects\project8\` CD-ROM 

### Задания для самостоятельного выполнения

- 5.30. Модернизировать проект «Строковый калькулятор» (файл `prj7.vbp`). Добавить возможности вырезки из строки произвольной подстроки, определения позиции подстроки в строке и преобразования числового кода символа в символ.
- 5.31. Создать проект «Подсчет количества экземпляров символа в тексте», который позволяет подсчитывать количество экземпляров какого-либо символа в заданном тексте.
- 5.32. Усовершенствовать предыдущий проект 5.31 так, чтобы ввод и вывод текста и символа производился с помощью текстовых окон.
- 5.33. Создать проект «Замена символа в тексте», который позволяет заменять в тексте один заданный символ на другой.

- 5.34. Усовершенствовать предыдущий проект 5.33 так, чтобы ввод и вывод текста и символов производился с помощью текстовых окон.
- 5.35. Создать проект «Слово-перевертыш», который изменяет прямую последовательность символов (слева направо) во введенном слове на обратную (справа налево).
- 5.36. Усовершенствовать предыдущий проект 5.35 так, чтобы ввод и вывод слова производился с помощью текстового окна.
- 5.37. Изменить проект 5.8 «Шифровка и дешифровка» таким образом, чтобы в программном коде событийных процедур использовался цикл с послеусловием.

## 5.8. Создание меню и панелей инструментов проектов

Большинство Windows-приложений обладают стандартным графическим интерфейсом. Ранее мы научились устанавливать требуемый размер окна проекта, его положение на экране монитора, цвет фона и шрифта и др. Однако работа со многими проектами невозможна без меню и панелей инструментов, которые обеспечивают доступ к тем или иным событийным процедурам.

**Проект 5.9. «Простой текстовый редактор».** Разработаем простейший текстовый редактор, который позволит выполнять основные операции редактирования текста. В качестве области ввода и редактирования текста используем многострочное текстовое поле.

### Проект «Простой текстовый редактор»

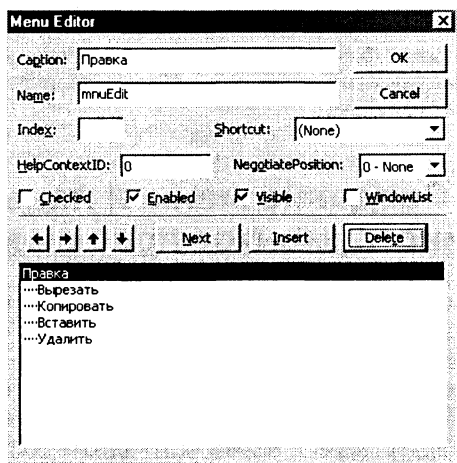
- 1 Разместить на форме текстовое поле и присвоить ему имя `txtEdit`.
- 2 Для того чтобы сделать поле многострочным, свойству поля `MultiLine` присвоить значение **True**.

Создадим меню для нашего текстового редактора. Меню состоит из одного или нескольких заголовков верхнего уровня, например, *Файл*, *Правка* и так далее, в каждый из которых входит хотя бы одна команда. Например, *Файл-Выход*, *Правка-Вырезать-Копировать-Вставить-Удалить*. Каждый заголовок меню и каждая команда являются управляющими элементами, которые обладают многими свойствами, но только одним событием `Click`.

Для создания меню используется специальный редактор меню *Menu Editor*.



- 3 Для запуска редактора меню перейти в окно формы командой [View-Object] и ввести команду [Tools-Menu Editor...].
- 4 На появившейся диалоговой панели *Menu Editor* создать заголовок первого уровня. В поле *Caption* ввести надпись *Правка* и в поле *Name* имя объекта *mnuEdit*.



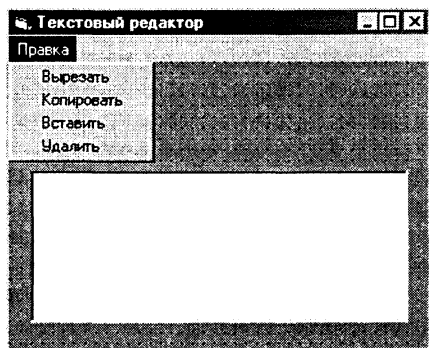
- 5 Для создания следующего пункта меню щелкнуть по кнопке *Next*.

Для создания пункта меню уровня команд (например, команды меню *Правка*) щелкнуть по кнопке со стрелкой вправо. Ввести надпись *Вырезать* и имя *mnuEditCut*.

- 6 Повторить процедуру и создать команды, входящие в меню *Правка*: *Копировать* (*mnuEditCopy*), *Вставить* (*mnuEditPaste*) и *Удалить* (*mnuEditDel*).

Теперь необходимо для каждого пункта меню создать событийную процедуру, которая будет выполняться по щелчку мыши.

- 7 В режиме конструирования проекта раскрыть меню и последовательно двойным щелчком мыши создать пустые заготовки событийных процедур для каждого пункта меню.



- 8 В процессе редактирования текста в окне txtEdit необходимо использовать для промежуточного хранения фрагментов текста объект Windows Clipboard (*Буфер обмена*).

Сначала необходимо очистить буфер с использованием метода Clear.

Затем поместить выделенный в текстовом поле фрагмент текста (свойство текстового поля SelText) в буфер с использованием метода буфера SetText.

Наконец, удалить выделенный текст, присвоив свойству SelText пустое значение "".

Таким образом, событийная процедура для команды меню *Вырезать* запишется следующим образом:

```
Private Sub mnuEditCut_Click()  
    ' Очистить Буфер обмена  
    Clipboard.Clear  
    ' Поместить выделенный текст в Буфер обмена  
    Clipboard.SetText frm1.txtEdit.SelText  
    ' Удалить выделенный текст  
    frm1.txtEdit.SelText = ""  
End Sub
```

- 9 Событийная процедура для команды меню *Копировать* запишется следующим образом:

```
Private Sub mnuEditCopy_Click()  
    ' Очистить Буфер обмена  
    Clipboard.Clear  
    ' Поместить выделенный текст в Буфер обмена  
    Clipboard.SetText frm1.txtEdit.SelText  
End Sub
```

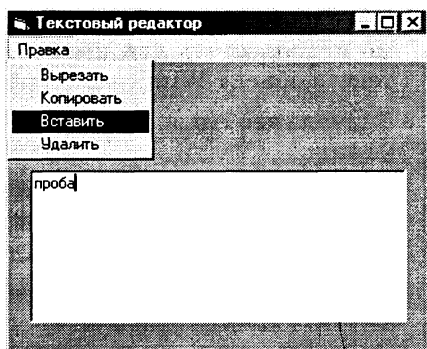
- 10 Событийная процедура для команды меню *Вставить* запишется следующим образом:

```
Private Sub mnuEditPaste_Click()  
    ' Вставить текст из Буфера обмена  
    frm1.txtEdit.SelText = Clipboard.GetText()  
End Sub
```

- 11 Событийная процедура для команды меню *Удалить* запишется следующим образом:

```
Private Sub mnuEditDel_Click()  
    ' Удалить выделенный текст  
    frm1.txtEdit.SelText = ""  
End Sub
```

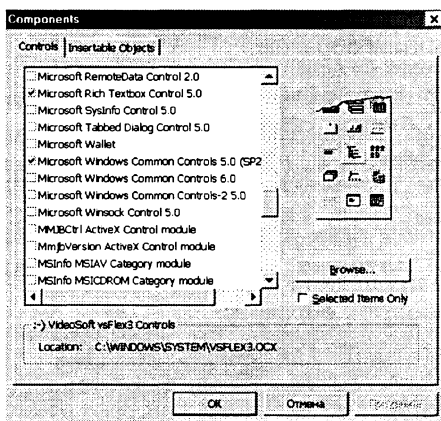
- 12 Запустить проект, ввести в текстовом окне текст и произвести его редактирование с использованием пунктов меню *Правка*.



Для быстрого ввода команд используются панели инструментов. Элемент управления *Панель инструментов* (ToolBar) не входит в стандартный набор управляющих элементов языка Visual Basic. Для работы с этим элементом необходимо воспользоваться одним из дополнительных наборов управляющих элементов *Microsoft Windows Common Controls 5.0 (SP2)*.

- 13 Ввести команду [Project-Components...].

На появившейся диалоговой панели *Components* из списка дополнительных наборов управляющих элементов выбрать набор *Microsoft Windows Common Controls 5.0 (SP2)*.



*Панель инструментов* окна Visual Basic дополнится новыми управляющими элементами и, в том числе, ToolBar и ImageList.

Создадим панель инструментов для меню *Правка*.

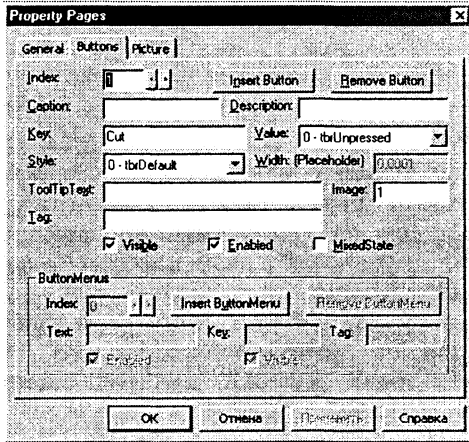
- 14 В режиме конструирования проекта поместить на форму управляющий элемент ToolBar и присвоить ему имя tbrToolbar.

### 15 Активизировать свойство *Custom*.

На появившейся диалоговой панели *Property Pages* выбрать вкладку *Buttons*.

Щелкнуть по кнопке *Insert Button*.

В поле *Key*: ввести имя первой кнопки панели инструментов *Cut*.



### 16 Повторить процедуру и поместить на панель инструментов проекта кнопки *Copy*, *Paste* и *Del*.

Для панели инструментов *Правка* необходимо создать событийную процедуру, которая должна обеспечивать необходимые действия по редактированию текста по щелчку по каждой из кнопок.

### 17 В режиме конструирования проекта дважды щелкнуть по панели инструментов проекта, будет создана пустая заготовка событийной процедуры.

Использовать инструкцию **Select Case** для вызова требуемых действий после щелчка по каждой из кнопок панели инструментов.

Для идентификации кнопок использовать значения свойства *Button.Key*:

```
Private Sub tbrToolbar_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Key
Case Is = "Cut"
Clipboard.Clear
Clipboard.SetText frm1.txtEdit.SelText
frm1.txtEdit.SelText = ""
Case Is = "Copy"
```

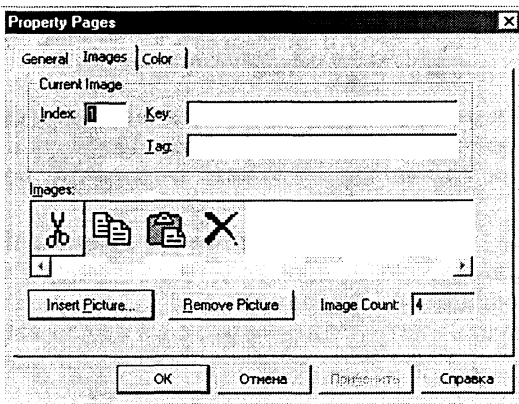
```
Clipboard.Clear
Clipboard.SetText frm1.txtEdit.SelText
Case Is = "Paste"
frm1.txtEdit.SelText = Clipboard.GetText()
Case Is = "Del"
frm1.txtEdit.SelText = ""
End Select
End Sub
```

Для удобства работы на кнопки панели инструментов проекта целесообразно поместить значки, как это сделано в Windows и ее приложениях.

18 Поместить на форму управляющий элемент *Список изображений* (ImageList) и присвоить ему имя *imlToolbar*.

19 В окне *Свойства объекта* активизировать свойство *Custom* и на появившейся диалоговой панели *Property Pages* выбрать вкладку *Images*.

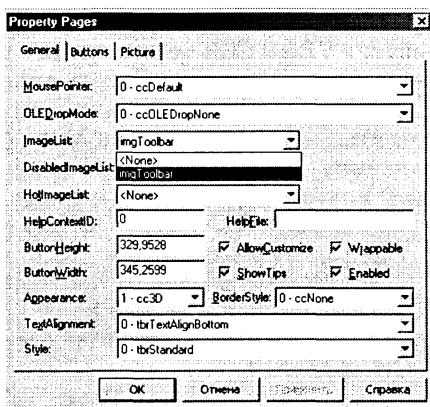
Щелкнуть по кнопке *Insert Picture...* и выбрать графический файл *cut.bmp*, который находится на CD-ROM в папке *\practicum\VB\Projects\project9\*, для размещения на первой кнопке *Cut*.



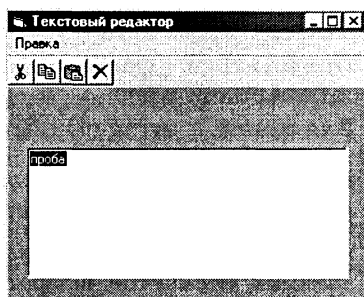
20 Повторить процедуру и выбрать графические файлы *copy.bmp*, *paste.bmp* и *delete.bmp* для кнопок *Copy*, *Paste* и *Del*.

Синхронизировать кнопки на панели инструментов *tbrToolbar* и значки из списка рисунков *imlToolbar*.

21 Для элемента *tbrToolbar* активизировать свойство *Custom*. На вкладке *General* в окне *ImageList* выбрать элемент *imlToolbar*. На вкладке *Buttons* синхронизировать значения свойства *Index* (упорядочивает кнопки) и свойства *Image* (упорядочивают значки).



- 22 Запустить проект и с помощью меню *Правка* и панели инструментов произвести редактирование введенного текста.



- 23 Сохранить проект в файле prj9.vbp.

*Проект хранится в каталоге*  
 \practicum\VB\Projects\project9\

CD-ROM 

**Проект 5.10. «Усовершенствованный текстовый редактор».**  
 На основе предыдущего проекта 5.9 разработаем текстовый редактор, который будет обладать дополнительными возможностями:

- открывать и сохранять текстовые файлы;
- форматировать текст (изменять параметры шрифта и выбирать цвет);
- поддерживать технологию «drag and drop».

Для реализации этого необходимо воспользоваться в качестве рабочего поля текстового редактора новым элементом управления RichTextBox (*Усовершенствованное текстовое поле*), которое является дополнительным компонентом и требует установки.

- 1 Ввести команду [Project-Components...]. На появившейся диалоговой панели *Components* из списка дополнительных управляющих элементов выбрать *Microsoft Rich Textbox Control 5.0*.

Требуемый управляющий элемент *Microsoft Rich Textbox Control 5.0* в дистрибутив VB5.0 CSE не входит. Его наличие в списке компонентов на диалоговой панели *Components* зависит от установленного на компьютере программного обеспечения: версии операционной системы Windows, версии Microsoft Office и др.

Элемент управления *RichTextBox* (*Усовершенствованное текстовое поле*) можно установить вручную. Для этого необходимо скопировать файл *RICHTX32.OCX* из каталога на диске `\practicum\VB\Projects\project10\` в каталог локального компьютера `\WINDOWS\system32\`. Для регистрации компонента перейти в этот каталог и ввести команду [Пуск-Выполнить...]. В появившемся текстовом поле ввести команду `regsvr32.exe RICHTX32.OCX`.

Панель инструментов окна Visual Basic дополнится управляющим элементом *RichTextBox*.



- 2 Открыть проект 5.9 и удалить текстовое поле `txtEdit`.

Поместить на форму усовершенствованное текстовое поле *RichTextBox* и присвоить ему имя `txtEdit`.

Для реализации операций с файлами и форматирования текста понадобится еще один дополнительный элемент управления *CommonDialog* (*Общий диалог*).

- 3 Ввести команду [Project-Components...]. На появившейся диалоговой панели *Components* из списка дополнительных управляющих элементов выбрать набор *Microsoft Common Dialog Controls 5.0 (SP2)*.

Панель инструментов окна Visual Basic дополнится управляющим элементом *CommonDialog*. Поместить этот элемент в любое место формы (он будет виден только в режиме конструирования проекта) и присвоить ему имя `dlg1`.



Для реализации дополнительных возможностей текстового редактора необходимо расширить систему меню и ввести дополнительные пункты *Файл* и *Форматирование*, содержащие команды второго и третьего уровня:

- 4 С помощью редактора меню *Menu Editor* расширить систему меню и ввести пункты:

*Файл*

...Открыть ...

...Сохранить

...Сохранить как ...

*Форматирование*

...Шрифт

...Цвет

.....Цвет шрифта

.....Цвет фона

Теперь для каждого пункта меню необходимо создать событийные процедуры. В процессе выполнения событийных процедур будут использоваться свойства управляющего элемента *Общий диалог* `dlg1`, для которых должны быть предварительно установлены определенные значения.

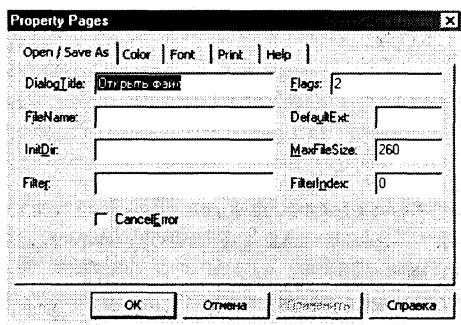
Создадим событийные процедуры открытия и сохранения файла.

5 Активизировать объект *Общий диалог* `dlg1`.

Выбрать свойство *Custom*, появится диалоговая панель *Property Pages* для установки значений свойств этого объекта.

Выбрать вкладку *Open/Save As* и в поле

*DialogTitle* ввести имя для окна, которое будет возникать при открытии файла (например, *Открыть файл*).



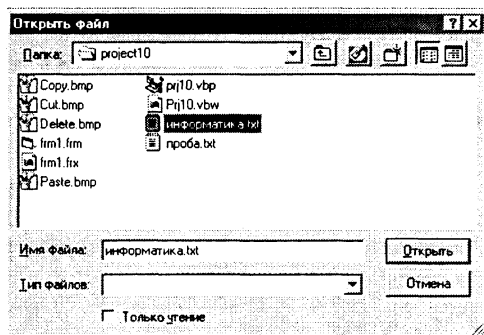
6 Ввести программный код событийной процедуры, которая будет производить открытие файла. В процедуре будет использован метод `ShowOpen`, который откроет соответствующую панель общего диалога, и метод `LoadFile`, который произведет открытие файла в текстовом поле:

```
Private Sub mnuOpen_Click()  
    dlg1.ShowOpen  
    txtEdit.LoadFile dlg1.filename  
End Sub
```

7 Запустить проект, ввести команду [Файл-Открыть...].

С помощью появившейся стандартной панели Windows *Открыть файл* выбрать файл и щелкнуть по кнопке *Открыть*.

Текст файла появится в окне нашего текстового редактора.





- 8 Ввести программный код событийных процедур *Сохранить* и *Сохранить как...*. В процедурах будет использован метод `ShowSave`, который откроет соответствующую панель общего диалога, и метод `SaveFile`, который произведет сохранение файла.

Процедура *Сохранить*:

```
Private Sub mnuSave_Click()
    dlg1.ShowSave
    If dlg1.filename = "" Then
        mnuSaveAs_Click
    Else
        txtEdit.SaveFile dlg1.filename
    End If
End Sub
```

Процедура *Сохранить как..*:

```
Private Sub mnuSaveAs_Click()
    dlg1.ShowSave
    txtEdit.SaveFile dlg1.filename
End Sub
```

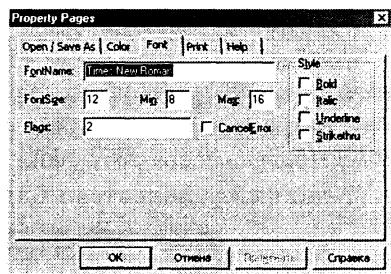
- 9 Запустить проект, ввести команду [Файл-Сохранить].

На появившейся панели *Открыть файл* в поле *Имя файла*: ввести имя и щелкнуть по кнопке *Сохранить*.

Создадим теперь событийную процедуру форматирования шрифта.

- 10 Для объекта `dlg1` активировать свойство `Custom` и на диалоговой панели *Property Pages* выбрать вкладку *Font*.

В полях *FontName*, *FontSize*, *Min* и *Max* ввести параметры шрифта, которые будут использоваться по умолчанию.



Обязательно установить в поле *Flags* значение 2 (определяет используемый тип шрифтов).

- 11 Ввести программный код событийной процедуры форматирования шрифта. В процедуре будут использованы метод `ShowFont`, а также пять свойств, определяющих начертание шрифта:

```

Private Sub mnuFont_Click()
    dlg1.ShowFont
    txtEdit.Font.Size = dlg1.FontSize
    txtEdit.Font.Name = dlg1.FontName
    txtEdit.Font.Bold = dlg1.FontBold
    txtEdit.Font.Italic = dlg1.FontItalic
    txtEdit.Font.Underline = dlg1.FontUnderline
End Sub

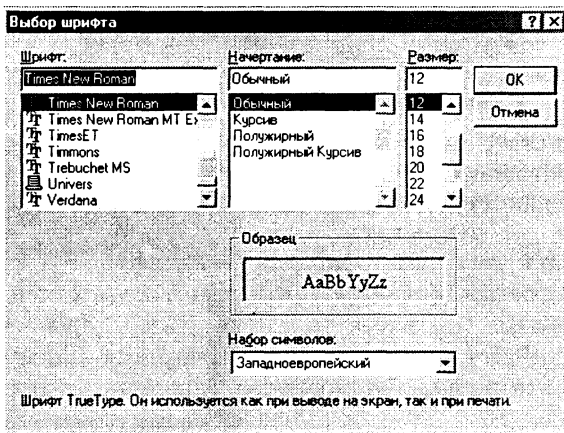
```

- 12 Запустить проект, ввести команду [Форматирование-Шрифт].

С помощью появившейся стандартной панели Windows *Выбор шрифта* установить требуемые параметры шрифты.

Создадим событийные процедуры выбора цвета шрифта (свойство `ForeColor`) и цвета фона (свойство `BackColor`).

- 13 Для объекта `dlg1` активизировать свойство `Custom` и на диалоговой панели *Property Pages* выбрать вкладку *Color*. В поле *Color* установить 0 (по умолчанию черный цвет), в поле *Flags* значение 2 (определяет открытие полного диало-



гового окна выбора цвета).

- 14 Ввести программный код событийных процедур выбора цвета шрифта и фона. В процедурах будут использованы метод `ShowColor`, а также свойства `SelColor` и `BackColor`. Процедура выбора цвета шрифта:

```

Private Sub mnuFontColor_Click()
    dlg1.ShowColor
    txtEdit.SelColor = dlg1.Color
End Sub

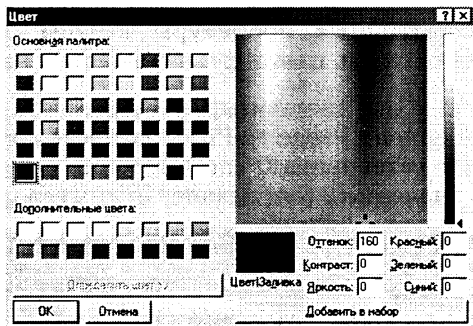
```

Процедура выбора цвета фона:

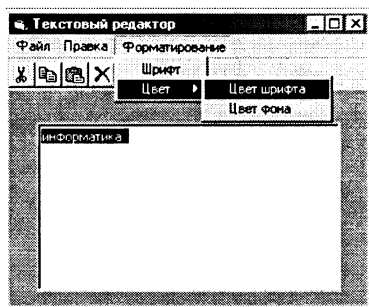
```
Private Sub mnuBackColor_Click()
    dlg1.ShowColor
    txtEdit.BackColor = dlg1.Color
End Sub
```

- 15 Запустить проект, ввести команду [Форматирование-Цвет-Цвет шрифта].

На появившейся панели *Цвет* задать числами интенсивность трех базовых составляющих цвета (*Красный*., *Зеленый*., и *Синий*.) или выбрать с помощью мыши нужный цвет.



- 16 В результате получаем вполне работоспособный текстовый редактор, который позволяет открывать и сохранять текстовые файлы, редактировать и форматировать текст.



- 17 Сохранить проект в файле prj10.vbp.

*Проект хранится в каталоге*  
 \practicum\VB\Projects\project10\

CD-ROM 

### Задания для самостоятельного выполнения

- 5.38. Усовершенствовать проект 5.9 «Простой текстовый редактор» введением меню *Файл* с командой *Выход*, которая обеспечивает выход из проекта.
- 5.39. Усовершенствовать проект 5.38, дополнив его панелью инструментов *Файл* с кнопкой *Выход*.

- 5.40. Усовершенствовать проект 5.39, предусмотрев возможность поиска и замены фрагментов текста, а также редактирования больших текстов.

## 5.9. Логические операторы в языке Visual Basic

Проект 5.11. «Логический калькулятор». В языке Visual Basic основные логические операции могут быть реализованы с помощью логических операторов **And** (логическое умножение), **Or** (логическое сложение), **Not** (логическое отрицание), **Xor** (исключающее **Or**, которое принимает логическое значение **True** тогда и только тогда, когда лишь один из аргументов имеет значение **True**) и **Eqv** (операция эквалентности, которая принимает логическое значение **True**, когда оба аргумента имеют значения **True** или **False**).

Логические операторы могут оперировать с логическими аргументами **True** (логическая единица) и **False** (логический нуль), а также с логическими переменными типа **Boolean**.

Разработаем проект «Логический калькулятор», который позволит определять истинность логических выражений.

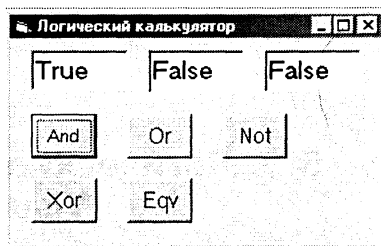
### Проект «Логический калькулятор»

- Разместить на форме три текстовых поля, два `txtA` и `txtB` (для ввода логических аргументов) и `txtC` (для вывода значения логических функций).
- 2 Разместить на форме пять кнопок `cmdAnd`, `cmdOr`, `cmdNot`, `cmdXor` и `cmdEqv` для реализации соответствующих событийных процедур.
- 3 Программный код событийной процедуры `cmdAnd_Click()`, реализующий операцию логического сложения, будет следующий:

```
Dim blnA, blnB, blnC As Boolean
Private Sub cmdAnd_Click()
    blnA = txtA.Text
    blnB = txtB.Text
    blnC = blnA And blnB
    txtC.Text = blnC
End Sub
```

- 4 Программные коды событийных процедур, реализующих другие логические функции, записываются аналогично.

- 5 Запустить проект. Ввести в текстовые поля значения логических аргументов и щелкнуть по любой кнопке логической операции. В третьем окне появится результат.



- 6 Сохранить проект в файле prj11.vbp.

*Проект хранится в каталоге  
 \practicum\VB\Projects\project11\*

CD-ROM 

### Задания для самостоятельного выполнения

- 5.41. Создать проект, позволяющий получать таблицы истинности для логических функций **And**, **Or**, **Not**, **Xor** и **Eqv**.
- 5.42. Дополнить проект 5.41 возможностями получения таблиц истинности других логических функций двух аргументов (смотри задание 3.27).
- 5.43. Создать проект, позволяющий получать таблицы истинности суммы и переноса для одноразрядного полусумматора двоичных чисел.
- 5.44. Создать проект, позволяющий получать таблицы истинности суммы и переноса для одноразрядного сумматора двоичных чисел.
- 5.45. Создать проект, моделирующий работу *RS*-триггера (смотри пример 3.12).

## 5.10. Графические возможности языка Visual Basic

На форме или в графическом поле можно рисовать различные графические примитивы с использованием графических методов:

**Scale** — позволяет задать систему координат и масштаб для формы или графического окна:

```
object.Scale (X1,Y1) - (X2,Y2)
```

**Pset** — установка точки с заданными координатами и цветом:

```
object.Pset (X,Y) [,color]
```

**Line** — рисование линии, прямоугольника или закрашенного прямоугольника заданного цвета:

```
object.Line (X1,Y1) - (X2,Y2) [,color][,B][F]
```

**Circle** – рисование окружности, овала или дуги с заданными координатами центра, радиусом, цветом, начальным и конечным углом дуги и коэффициентом сжатия:

```
object.Circle (X,Y),radius [,color, start, end, aspect]
```

**Проект 5.12. «Графический редактор».** Разработаем проект, который будет выполнять функции простого графического редактора, в котором в качестве поля для рисования будет использоваться графическое поле.

Прежде всего, необходимо задать удобную систему координат с помощью оператора **Scale**, который фиксирует координаты левого верхнего и нижнего правого углов поля.

### Проект «Графический редактор»

- 1 Разместить на форме графическое поле `picPaint`.
- 2 Разместить на форме кнопку `cmdScale` и создать событийную процедуру установки системы координат (0 в левом нижнем углу поля, максимальные значения координат X и Y равны 100):

```
Private Sub cmdScale_Click()  
picPaint.Scale (0, 100)-(100, 0)  
End Sub
```

**Процедура установки точки.** Координаты точки будем запрашивать с помощью функции ввода `InputBox`, а цвет зададим с помощью одной из восьми констант, определяющих цвет (`vbBlack` — черный, `vbBlue` — синий, `vbGreen` — зеленый, `vbCyan` — голубой, `vbRed` — красный, `vbMagenta` — сиреневый, `vbYellow` — желтый, `vbWhite` — белый).

- 3 Разместить на форме кнопку `cmdPoint` и создать событийную процедуру установки точки:

```
Private Sub cmdPoint_Click()  
bytX1 = Val(InputBox("Введите координату X",  
"Координата X"))  
bytY1 = Val(InputBox("Введите координату Y",  
"Координата Y"))  
picPaint.PSet (bytX1, bytY1), vbRed  
End Sub
```

**Процедура рисования линии.** Координаты концов линии и числовой код цвета будем запрашивать с помощью функции ввода `InputBox`. Цвет будем задавать с помощью функции `QBColor(number)`, аргументом которой являются числа от 0 до 15, а результатом — один из основных 16 цветов.

- 4 Разместить на форме кнопку cmdLine и создать событийную процедуру рисования линии:

```

Private Sub cmdLine_Click()
bytX1 = Val(InputBox("Введите координату X1",
"Координата X1"))
bytY1 = Val(InputBox("Введите координату Y1",
"Координата Y1"))
bytX2 = Val(InputBox("Введите координату X2",
"Координата X2"))
bytY2 = Val(InputBox("Введите координату Y2",
"Координата Y2"))
bytC = Val(InputBox("Введите цвет", "Цвет"))
picPaint.Line (bytX1, bytY1)-(bytX2, bytY2),
    QBColor(bytC)
End Sub

```

**Процедура рисования прямоугольника.** Координаты противоположных вершин прямоугольника и числовые коды интенсивностей красного, зеленого и синего цветов будем запрашивать с помощью функции ввода InputBox.

Цвет будем задавать с помощью функции RGB(bytRed, bytGreen, bytBlue), аргументами которой являются три числа в диапазонах 0 до 255 (интенсивности базовых цветов), а результатом — число типа Long в диапазоне от 0 до  $256^3$  (16 777 215). Таким образом определяется цветовая палитра с более чем 16 миллионами цветов, а каждый цвет задается числом, которое вычисляется по формуле  $\text{bytRed} + 256 * \text{bytGreen} + 256^2 * \text{bytBlue}$ .

- 5 Разместить на форме кнопку cmdLineB и создать событийную процедуру рисования прямоугольника:

```

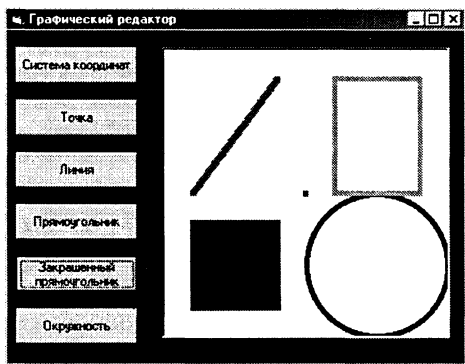
Private Sub cmdLineB_Click()
bytX1 = Val(InputBox("Введите координату X1",
"Координата X1"))
bytY1 = Val(InputBox("Введите координату Y1",
"Координата Y1"))
bytX2 = Val(InputBox("Введите координату X2",
"Координата X2"))
bytY2 = InputBox("Введите координату Y2",
"Координата Y2")
bytR = Val(InputBox("Введите интенсивность
красного цвета", "Красный цвет"))
bytG = Val(InputBox("Введите интенсивность
зеленого цвета", "Зеленый цвет"))
bytB = Val(InputBox("Введите интенсивность
синего цвета", "Синий цвет"))

```

```
picPaint.Line (bytX1, bytY1)-(bytX2, bytY2),
RGB(bytR, bytG, bytB), B
```

```
End Sub
```

- 6 Разместить на форме кнопки cmdLineBF и cmdCircle и самостоятельно создать событийные процедуры рисования закрашенных прямоугольников и окружностей.
- 7 Запустить проект. Последовательно щелкнуть по кнопкам рисования объектов и ввести необходимые параметры на появляющихся панелях ввода.



- 8 Сохранить проект в файле prj12.vbp.

Проект хранится в каталоге  
 \practicum\VB\Projects\project12\

CD-ROM 

**Проект 5.13. «Построение графика функции».** Разработаем проект построения графиков функций. В качестве примера рассмотрим построение графика функции  $y = \operatorname{tg}x$ .



### Проект «Построение графика функции»

- 1 Разместить на форме графическое поле picGraph, в котором будет производиться построение графика.
- 2 Разместить на форме кнопку cmd1 и создать событийную процедуру построения графика, в которой будет:
  - установлен масштаб;
  - в цикле осуществлено построение графика функции;
  - нарисованы оси координат и в циклах напечатаны на них числовые шкалы;
  - в цикле выведены вертикальные линии координатной сетки.

```
Private Sub cmd1_Click()
  'Установка масштаба
  picGraph.Scale (-10, 2)-(10, -2)
```

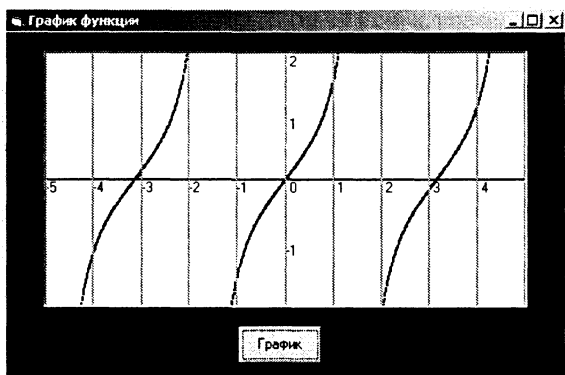


```

'Построение графика
For sngX = -10 To 10 Step 0.01
picGraph.PSet (sngX, Tan(sngX))
Next sngX
'Ось X
picGraph.Line (-10, 0)-(10, 0)
For intI = -10 To 10
picGraph.PSet (intI, 0)
picGraph.Print intI
Next intI
'Ось Y
picGraph.Line (0, 2)-(0, -2)
For intI = -2 To 2
picGraph.PSet (0, intI)
picGraph.Print intI
Next intI
'Координатная сетка
For intI = -5 To 5
picGraph.Line (intI, 2)-(intI, -2), vbGreen
Next intI
End Sub

```

3 Запустить проект и щелкнуть по кнопке *График*.



4 Сохранить проект в файле prj13.vbp.

Для построения графика другой функции необходимо в программном коде в цикле построения графика задать новую функцию.

Проект хранится в каталоге  
 \practicum\VB\Projects\project13\

CD-ROM 

## Задания для самостоятельного выполнения

- 5.46. Усовершенствовать проект 5.12 «Графический редактор» так, чтобы параметры для рисования графических примитивов задавались в текстовых полях, а также существовала возможность очистки поля рисования.
- 5.47. Усовершенствовать проект 5.46 так, чтобы координаты для рисования графических примитивов задавались щелчками мыши в определенных точках графического поля.
- 5.48. Усовершенствовать проект 5.47 так, чтобы управление графическим редактором реализовывалось не кнопками, а с помощью меню.
- 5.49. Придумать проект, реализующий демонстрацию палитры цветов.
- 5.50. Создать проект, позволяющий задавать цвета различными способами (с помощью цветовых констант, функции QBColor и функции RGB) и демонстрирующий заданный цвет.
- 5.51. Создать проект, который чертит треугольник по заданным щелчком мыши точкам и вычисляет его периметр и площадь.
- 5.52. Усовершенствовать проект 5.13 «Построение графика функции» так, чтобы можно было строить график функции для различных наборов значений ее аргумента  $X$ . Предусмотреть автоматическую корректировку масштаба и шкал осей координат. Предусмотреть возможность очистки графического поля.

## 5.11. Анимация

Для создания анимации (иллюзии движения на экране какого-либо объекта) применяется принцип смены кадров (изображений), как это делается в мультипликации. Программа, имитирующая движение, должна реализовывать следующие этапы:

- создание изображения на экране;
- реализация временной паузы для того, чтобы глаз зафиксировал изображение;
- проведение коррекции изображения.

Анимация часто используется для изображения движения объектов. Для регулирования скорости движения объекта используют пустой цикл, чем большее количество раз он будет выполняться, тем медленнее будет двигаться объект.

**Проект 5.14. «Движение Земли вокруг Солнца».** Разработаем проект, который отображает движение Земли по орбите вокруг Солнца.



## Проект «Движение Земли вокруг Солнца»

- 1 Разместить на форме графическое поле `picAnim`, в котором будет производиться изображение движения. Свойству `BackColor` графического поля присвоить значение `Blue`. Свойству `FillStyle` присвоить значение `Solid`.
- 2 Поместить на форму кнопку `cmdStart` и начать создание событийной процедуры, реализующей анимацию. Задать количество оборотов вокруг Солнца и удобный масштаб:

```
Private Sub cmdStart_Click()  
    'Количество оборотов  
    intRev = 100  
    'Масштаб  
    picAnim.Scale (-10, 10)-(10, -10)  
End Sub
```

Солнце и орбиту Земли вокруг Солнца будем рисовать с помощью метода `Circle` желтым цветом.

Движение Земли по орбите реализуем в цикле, в котором по известным формулам (по радиусу орбиты и углу поворота) вычисляется положение центра окружности, изображающей Землю.

Эффект анимации получим рисованием сначала круга голубого цвета, а затем его стиранием кругом цвета фона (синим). Для получения закрашенного круга требуется установить требуемое значение свойства `FillColor` (при рисовании `vbCyan`, а при стирании `vbBlue`), которое определяет цвет заполнения рисуемой фигуры.

Скорость движения Земли по орбите (скорость смены одного изображения другим) можно регулировать с помощью изменения количества повторений пустого цикла между рисованием и стиранием.

Для этого можно использовать управляющий элемент `Slider` (*Ползунок*), который входит в дополнительный набор управляющих элементов *Microsoft Windows Common Controls 5.0 (SP2)*. Установив указатель ползунка в определенное положение, мы присваиваем свойству `Value` значение из интервала, заданного свойством `Min` (по умолчанию 0) и свойством `Max` (по умолчанию 10).

- 3 Разместить на форме ползунок `sld1` и кнопку `cmdSld`. Создать событийную процедуру регулировки скорости движения:



```

Private Sub cmdSld_Click()
  bytS = sld1.Value
End Sub

```

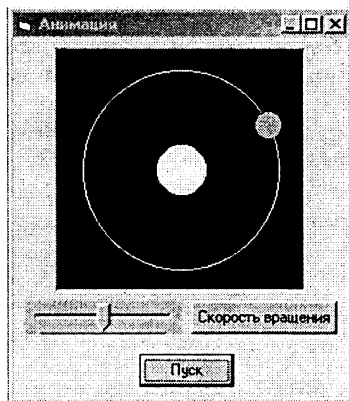
- 4 Продолжить создание событийной процедуры, реализующей анимацию, включить в нее рисование орбиты Земли, Солнца и движения Земли по орбите:

```

'Анимация
For intN = 1 To 360 * intRev
'Орбита Земли
picAnim.Circle (0, 0), 8, vbYellow
'Солнце
picAnim.FillColor = vbYellow
picAnim.Circle (0, 0), 2, vbYellow
'Рисование Земли
picAnim.FillColor = vbCyan
picAnim.Circle (8 * Sin(6.28 / 360 * intN), 8
* Cos(6.28 / 360 * intN)), 1, vbCyan
'Задержка стирания
For intI = 1 To 100000 * bytS
Next intI
'Стирание Земли
picAnim.FillColor = vbBlue
picAnim.Circle (8 *
Sin(6.28 / 360 * intN),
8 * Cos(6.28 / 360 *
intN)), 1, vbBlue
Next intN

```

- 5 Запустить проект.  
Установить с помощью ползунка скорость движения Земли по орбите и зафиксировать ее щелчком по кнопке *Скорость вращения*.  
Щелкнуть по кнопке *Пуск*.  
Для выхода из программы одновременно нажать клавиши  $\{Ctrl\}+\{Break\}$ .



- 6 Сохранить проект в файле prj14.vbp.

Проект хранится в каталоге  
 \practicum\VB\Projects\project14\ CD-ROM

Для воспроизведения готовых анимационных файлов (типа AVI) можно воспользоваться дополнительным управляющим элементом Animation (*Анимация*), который входит в набор дополнительных компонентов Visual Basic *Microsoft Windows Common Controls-2 5.0*.

**Проект 5.15. «Мультимедиа».** Создадим проект, который воспроизводит анимационные файлы типа AVI.



### Проект «Мультимедиа»

- 1 Установить набор дополнительных компонентов Visual Basic *Microsoft Windows Common Controls-2 5.0* и поместить на форму управляющий элемент Animation. Присвоить ему имя anim1.
- 2 Разместить на форме управляющий элемент dlg1 (*Общий диалог*).
- 3 Поместить на форму кнопку cmdStart и создать для нее процедуру открытия анимационного файла (с использованием элемента *Общий диалог*) и запуска его на воспроизведение:

```
Private Sub cmdStart_Click()
  dlg1.ShowOpen
  anim1.Open dlg1.filename
  anim1.Play
End Sub
```

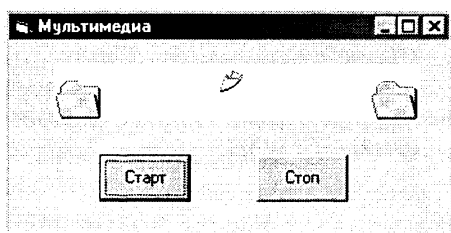
- 4 Поместить на форму кнопку cmdClose и создать для нее процедуру закрытия анимационного файла:

```
Private Sub cmdClose_Click()
  anim1.Close
End Sub
```

- 5 Запустить проект и щелкнуть по кнопке *Старт*.

На появившейся панели

*Открытие файла* выбрать файл с расширением AVI. Например, файл filesору.avi, который на CD-ROM находится в каталоге



\\practicum\VB\Projects\project15\.

Для окончания воспроизведения щелкнуть по кнопке *Стоп*.



6 Сохранить проект в файле prj15.vbp.

*Проект хранится в каталоге  
\\practicum\VB\Projects\project15\.*

CD-ROM

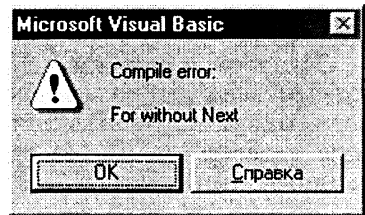
## Задания для самостоятельного выполнения

- 5.53. Создать проект, реализующий работу стрелочных часов. Стрелки (часовая, минутная и секундная) должны вращаться с использованием эффекта анимации.
- 5.54. Создать проект, реализующий работу стрелочных и электронных часов. Стрелки (часовая, минутная и секундная) должны вращаться с использованием эффекта анимации синхронно с системным временем компьютера.
- 5.55. Усовершенствовать проект 5.15 «Мультимедиа» и заменить кнопки на меню.

## 5.12. Отладка программных кодов

В процессе разработки проектов программный код может содержать ошибки. Ошибки бывают двух видов: *синтаксические*

Рис. 5.4  
Сообщение об ошибке



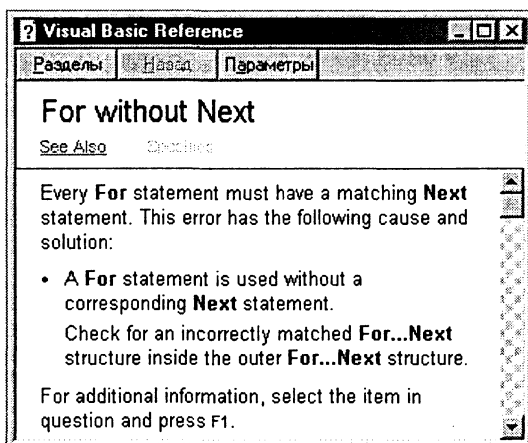
и *логические*.

**Синтаксические ошибки.** Синтаксические ошибки в процессе ввода программы в окне *Программный код* система Visual Basic обнаруживает автоматически и сообщает о них разработчику.

Например, если в операторе цикла со счетчиком отсутствует служебное слово *Next*, то строка с ошибкой будет выделена и появится информационная панель с сообщением об ошибке.

Если причина ошибки ясна, то можно щелкнуть по кнопке *OK* и исправить ошибку.

Рис. 5.5  
Справка




В противном случае следует обратиться за помощью к справочной системе Visual Basic, для чего щелкнуть по кнопке *Справка* (в версии VB 5.0 CSE отсутствует).

Если при переходе к следующей строке программы система Visual Basic не обнаружила синтаксических ошибок, то встроенный редактор автоматически выделит ключевые слова (операторы, методы и др.) синим цветом, а комментарии — зеленым.

Справочную систему можно вызвать в любой момент. Для этого необходимо установить курсор на ключевое слово и нажать клавишу  $\{F1\}$ .

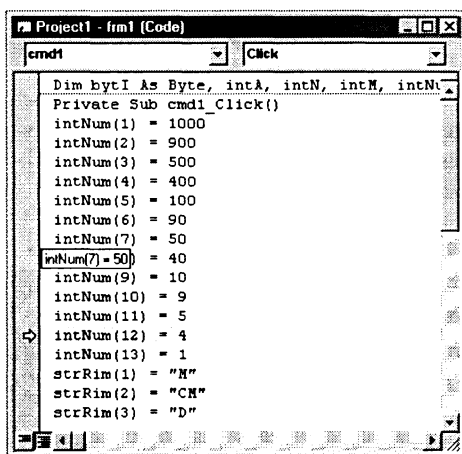
**Логические ошибки.** После запуска проекта могут обнаружиться логические ошибки, содержащиеся в программном коде. Например, если в цикле по условию условие сформулировано неверно, то может произойти заикливание (программа начинает выполняться бесконечное множество раз). Для выхода из цикла необходимо одновременно нажать клавиши  $\{Ctrl\} + \{Break\}$ , после чего в окне программного кода система автоматически выделит желтым цветом последнюю выполненную строку программы.

Если причина логической ошибки неясна, то можно использовать специальную возможность и запустить *Отладчик программы (Debug)*. Одной из важнейших возможностей отладчика является выполнение программы по шагам, когда разработчик может проследить последовательность выполнения программы и проследить процесс присваивания переменным их значений.

 **Проект 5.16. «Использование отладчика программы».**  
Запустить проект «Перевод чисел из римской системы счисления в десятичную» в режиме отладки и проследить ход выполнения программы.

### Проект «Использование отладчика программы»

- 1 Открыть проект `prj16.vbp`, хранящийся в папке `\practicum\VB\Projects\project16\`, и запустить отладчик командой [Debug-Step Into].
- 2 Выполнять программу по шагам, нажимая клавишу {F8}. Текущая (выполняемая) строка программы отмечается жел-



```
Project1 - frm1 (Code)
cmd1 Click
Dim bytI As Byte, intA, intN, intM, intN
Private Sub cmd1_Click()
intNum(1) = 1000
intNum(2) = 900
intNum(3) = 500
intNum(4) = 400
intNum(5) = 100
intNum(6) = 90
intNum(7) = 50
intNum(7) = 50
intNum(9) = 10
intNum(10) = 9
intNum(11) = 5
intNum(12) = 4
intNum(13) = 1
strRim(1) = "M"
strRim(2) = "CM"
strRim(3) = "D"
```

тым цветом.

Подвести курсор к имени любой переменной выше исполняемой строки.

Появится надпись с информацией о значении, присвоенном переменной.



# Глава 6

---

## Моделирование и формализация

---

В процессе изучения темы  
рекомендуется установить:

CD-ROM 

- систему программирования Visual Basic (VB5.0 CCE – Visual Basic 5.0 Control Creation Edition);
- электронные таблицы StarCalc, входящие в состав интегрированного офисного приложения StarOffice;
- систему компьютерного черчения КОМПАС-3D;
- геоинформационную модель Карта Москвы (MosMap);
- компьютерную модель периодической системы элементов Д. И. Менделеева Table.

Процесс разработки моделей и их исследование 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, параметры объекта, а несущественными параметрами пренебрегает.

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

На третьем этапе необходимо формализованную информационную модель преобразовать в *компьютерную модель*, то есть выразить ее на понятном для компьютера языке. Существуют два принципиально различных пути построения компьютерной модели:


- построение алгоритма решения задачи и его кодирование на одном из языков программирования;
- построение компьютерной модели с использованием одного из приложений (электронных таблиц, СУБД и так далее).

Четвертый этап исследования информационной модели состоит в проведении *компьютерного эксперимента*. Если компьютерная модель существует в виде программы на одном из языков программирования, ее нужно запустить на выполнение и получить результаты.

Если компьютерная модель исследуется в приложении, например, в электронных таблицах, можно провести сортировку или поиск данных, построить диаграмму или график и так далее

Пятый этап состоит в *анализе полученных результатов и корректировке исследуемой модели*. В случае различия результатов, полученных при исследовании информационной модели, с измеряемыми параметрами реальных объектов, можно сделать вывод, что на предыдущих этапах построения модели были допущены ошибки или неточности.

## 6.1. Исследование математических моделей

Алгебра-9, Геометрия-7 

Исследование математических моделей начинается с записи формальной модели на языке определенной области математики: алгебры, геометрии и так далее.

### 6.1.1. Приближенное решение уравнений

На языке алгебры 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, числовые и др.).

**Графический метод.** Построение графиков функций может использоваться для грубо приближенного решения уравнений. Для не имеющего точного алгебраического решения уравнения вида  $f_1(x) = f_2(x)$ , где  $f_1(x) = f_2(x)$  — некоторые непрерывные функции, корень (или корни) этого уравнения являются точкой (или точками) пересечения графиков этих функций.

**Задача.** Найти графическим методом корень уравнения  $x^3 = \sin x$ , которое не имеет точного алгебраического решения.

Формальная модель задана уравнением, для нахождения корня уравнения разработаем компьютерную модель на языке Visual Basic.



### Проект «Графическое решение уравнения»

- 1 Открыть проект «Построение графика функции» (файл prj13.vbp). Поместить на форму кнопку cmd2 и создать для нее событийную процедуру построения графика функции  $y = x^3$ :

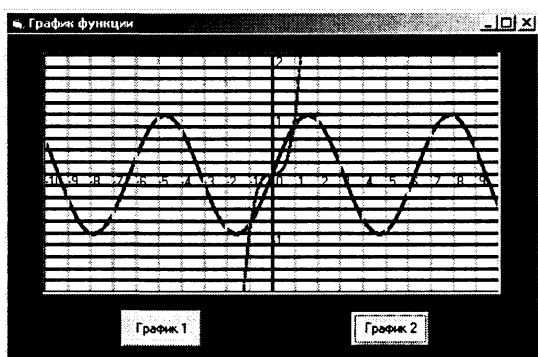
```
Private Sub cmd2_Click()
For sngX = -10 To 10 Step 0.01
picGraph.PSet (sngX, sngX ^ 3), vbRed
Next sngX
End Sub
```

- 2 Для более точного решения уравнения добавить в существующую процедуру построения графика первой функции  $y = \sin x$  координатную сетку:

```
'Координатная сетка
For bytI = -10 To 10
picGraph.Line (bytI, 2)-(bytI, -2), vbCyan
Next bytI
For bytI = -2 To 2 Step 0.2
picGraph.Line (-10, bytI)-(10, bytI)
Next bytI
```

- 3 Запустить проект и щелкнуть по кнопкам *График 1* и *График 2*.

Координаты  $x$  точек пересечения графиков и будут корнями данного уравнения:  $x_1 = -1$ ,  $x_2 = 0$ ,  $x_3 = 1$ .



**Числовой метод половинного деления.** Для решения уравнений с заданной точностью можно применить разработанные в вычислительной математике числовые итерационные методы решения уравнений. Если мы знаем отрезок, на котором существует корень, и функция на краях этого отрезка принимает значения разных знаков, то можно использовать метод половинного деления.

Идея метода состоит в выборе точности решения и сведении первоначального отрезка  $[A;B]$ , на котором существует корень уравнения, к отрезку заданной точности. Процесс сводится к последовательному делению отрезков пополам точкой  $C = (A+B)/2$  и отбрасыванию той половины отрезка ( $[A;C]$  или  $[C;B]$ ), на котором корня нет.

Выбор нужной половины отрезка основывается на проверке знаков значения функции на его краях. Выбирается та половина, на которой произведение значений функции на краях отрицательно, то есть когда функция имеет разные знаки и пересекает ось абсцисс.

Процесс продолжается до тех пор, пока длина отрезка не станет меньше удвоенной точности. Деление этого отрезка пополам дает значение корня с заданной точностью  $x = (A+B)/2$ .



### Проект «Числовое решение уравнения с заданной точностью»

- 1 Открыть проект «Графическое решение уравнения» (файл rj6-1.vbp). Вместо двух событийных процедур построения графиков функций  $x^3$  и  $\sin x$  создать событийную процедуру построения графика  $x^3 - \sin x$ .
- 2 Поместить на форму текстовые поля txtA и txtB для ввода числовых значений концов отрезка A и B (переменные dblA и dblB), поле txtE для ввода точности вычислений (переменная dblE) и поле txtX для вывода значений корня.
- 3 Поместить на форму кнопку и создать событийную процедуру, вычисляющую корень уравнения методом половинного деления с использованием цикла с послеусловием:

```
Dim dblA, dblB, dblE As Double
Private Sub cmdNum_Click()
dblA = Val(txtA.Text)
dblB = Val(txtB.Text)
dblE = Val(txtE.Text)
Do
dblC = (dblA + dblB) / 2
If (dblA^3 - Sin(dblA)) * (dblC^3 - Sin(dblC)) < 0
Then dblB = dblC
```

```

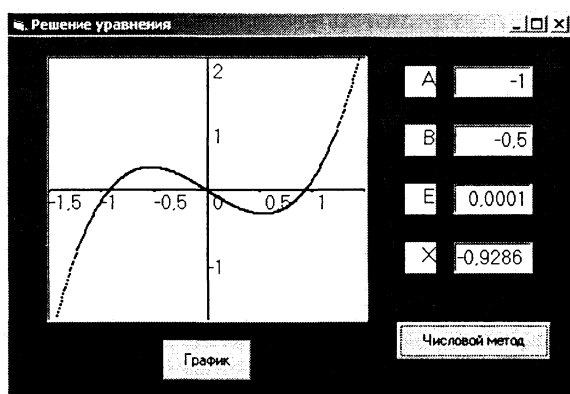
Else
dblA = dblC
End If
Loop While (dblB - dblA) / 2 > dblE
txtX.Text = (dblA + dblB) / 2
End Sub

```

- 4 Из графика функции видно, что первый корень находится на отрезке  $[-1; -0,5]$ , второй — на отрезке  $[-0,5; 0,5]$ , а третий — на отрезке  $[0,5; 1]$ .

Сначала введем в текстовые поля значения концов первого отрезка, а также точность вычислений (например, 0,0001).

В текстовое поле будет выведено значения первого корня, вычисленное с заданной точностью:  $x_1 = -0,9286$ .



- 5 После ввода в текстовые поля значений концов второго отрезка в текстовое поле будет выведено значение второго корня  $x_2 = 0$ .
- 6 После ввода в текстовые поля значений концов третьего отрезка в текстовое поле будет выведено значение третьего корня  $x_3 = 0,9286$ .

Проект хранится в каталоге  
`\practicum\VB\Projects\prj6-2\`

CD-ROM 

### Практические задания

- 6.1. С использованием компьютерной модели в электронных таблицах найти приближенное (графическое) решение уравнения  $x^3/10 = \sin x$ .
- 6.2. С использованием компьютерной модели в электронных таблицах с заданной точностью методом подбора параметра решить уравнение  $x^3/10 - \sin x = 0$ .

## 6.1.2. Вероятностные модели

Вероятностные модели базируются на использовании больших серий испытаний со случайными параметрами, причем точность полученных результатов зависит от количества проведенных опытов. Воспользуемся методом Монте-Карло для исследования вероятностной модели бросания монеты.

### 2.2. Количество информации как мера уменьшения неопределенности знаний Учебник

**Вероятностная модель бросания монеты.** При введении понятия «количество информации» рассматривался опыт по бросанию монеты. Если бросить симметричную монету на ровную поверхность, то можно предположить, что с равной вероятностью произойдет одно из двух возможных событий — монета окажется в одном из двух положений: «орел» или «решка». Доказательство этого утверждения можно получить при проведении большой серии опытов, когда количества выпадений «орла» и «решки» постепенно сближаются.

Сначала построим качественную вероятностную модель бросания монеты:

- поместим квадрат со стороной равной 1 в центр координат и разделим его на две равные части по оси  $Y$ , назовем эти части «орел» и «решка»;
- заменим бросание монеты на «бросание» точек в этот квадрат с помощью генератора случайных чисел, который будет задавать точкам случайные координаты внутри квадрата;
- будем считать, что количество точек, попавших в левую часть квадрата, соответствует выпадению «орла», а попадание в правую половину квадрата — выпадению «решки».

**Формальная модель.** Пусть  $N$  — количество точек, которые случайным образом генерируются внутри квадрата. Случайный выбор координат точек, которые попадают внутрь квадрата ( $N$  точек), должен производиться так, чтобы координаты точек  $x$  и  $y$  удовлетворяли условиям:

$$-1 \leq x \leq 1 \text{ и } -1 \leq y \leq 1 .$$

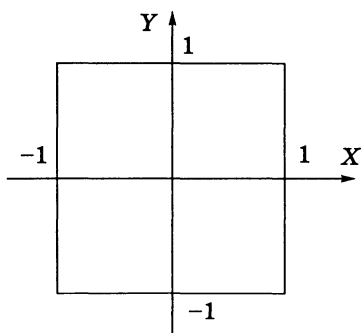


Рис. 6.1. Квадрат, разделенный на две части

Пусть  $O$  — количество точек («орел»), попавших в левую часть квадрата, координаты которых удовлетворяют условию:

$$-1 \leq X \text{ And } X < 0 \text{ And } -1 \leq Y \text{ And } Y \leq 1.$$

Тогда  $R$  — количество точек («решка»), попавших в правую часть квадрата, координаты которых удовлетворяют условию:

$$0 < X \text{ And } X \leq 1 \text{ And } -1 \leq Y \text{ And } Y \leq 1.$$

**Компьютерная модель.** Разработаем на языке Visual Basic компьютерную модель, позволяющую доказать, что выпадение монеты «орлом» или «решкой» равновероятно.



### Проект «Бросание монеты»

- 1 Поместить на форму графическое поле, в котором будет отображаться процесс случайной генерации точек, в нем нарисовать квадрат со стороной, равной 1, и оси координат.
- 2 Поместить на форму текстовое поле `txtN` для ввода числа генерируемых точек, поле `txtO` для вывода числа точек, попавших в левую половину квадрата («орел»), и поле `txtR` для вывода числа точек, попавших в правую половину квадрата («решка»).
- 3 Поместить на форму кнопку и создать для нее событийную процедуру, которая обеспечивает ввод количества генерируемых точек в переменную `lngN`, генерацию случайных точек, подсчет в переменной `lngO` количества точек попавших в левую половину квадрата и подсчет в переменной `lngR` количества точек попавших в правую половину квадрата:

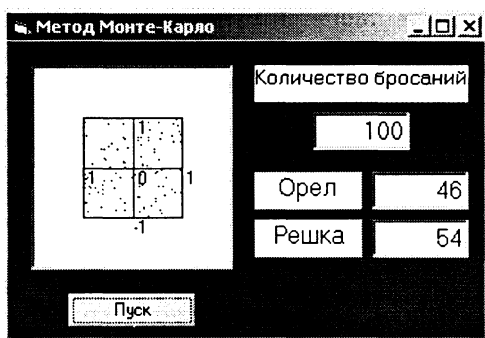
```
Dim dblX, dblY As Double, I, lngN, lngO, lngR
As Long
Private Sub cmd1_Click()
lngO = 0
lngR = 0
lngN = txtN.Text
pic1.Cls
pic1.Scale (-1, 1)-(1, -1)
pic1.Line (-1, 1)-(1, -1), , B
'Генерация точек
For I = 1 To lngN
dblX = 2 * Rnd - 1
dblY = 2 * Rnd - 1
pic1.PSet (dblX, dblY)
If -1 <= dblX And dblX < 0 And -1 <= dblY
And dblY <= 1 Then
lngO = lngO + 1
```

```

Else
lngR = lngR + 1
End If
Next I
txtO.Text = lngO
txtR.Text = lngR
'Ось X
pic1.Line (-1, 0)-(1, 0)
For I = -1 To 1
pic1.PSet (I, 0)
pic1.Print I
Next I
'Ось Y
pic1.Line (0, -1)-(0, 1)
For I = -1 To 1
pic1.PSet (0, I)
pic1.Print I
Next I
End Sub

```

- 4 Ввести количество генерируемых точек. После щелчка по кнопке *Пуск* в графическом поле будет отображен процесс генерации случайных точек, а в текстовые поля выведены количества выпадений «орла» и «решки».



**Исследование модели.** При увеличении количества генерируемых точек можно наблюдать все меньшее различие в количествах выпавших «орлов» и «решек».

Проект хранится в каталоге  
 \practicum\VB\Projects\prj6-3\

CD-ROM 

### Практические задания

- 6.3. Доказать методом Монте-Карло, что при бросании симметричного кубика его падения на все грани равновероятны.



### 6.1.3. Геометрические модели

Пространственные соотношения между реальными объектами (положение и ориентация объектов в пространстве и их размеры) изучаются с помощью геометрических моделей. Для визуализации геометрических моделей используются идеализированные геометрические объекты (точка, линия, плоскость и др.), которые в отличие от реальных объектов обладают набором только наиболее существенных свойств. Так, геометрическая точка отличается от реальной точки на чертеже тем, что имеет только координаты, но не имеет размеров, геометрическая линия не имеет ширины, геометрическая плоскость — толщины и так далее.

В школьном курсе геометрии не только изучаются различные геометрические модели (теоремы), но и рассматривается процесс их построения. Важное место занимают геометрические построения с использованием линейки и циркуля. Для создания геометрических моделей на компьютере удобно использовать системы автоматизированного проектирования (САПР).

В качестве примера выполнения геометрического построения рассмотрим задачу о построении перпендикуляра к прямой.

**Задача.** Даны прямая и точка на ней. Построить прямую через данную точку и перпендикулярную к данной прямой.

**Формальная модель.** Построим формальную модель процесса геометрического построения, зафиксировав его в форме алгоритма:

1. Построить прямую  $a$  и точку  $M$  на ней.
2. На равных расстояниях от точки  $M$  построить на прямой точки  $A$  и  $B$ .
3. Построить две окружности с центрами в точках  $A$  и  $B$  с радиусом  $AB$ .
4. Через точки пересечения окружностей  $P$  и  $Q$  провести прямую. Данная прямая пройдет через точку  $M$  и будет являться перпендикуляром к прямой  $a$ .

**Компьютерная модель.** Реализуем геометрическое построение в соответствии с разработанным алгоритмом с использованием системы КОМПАС-3D.



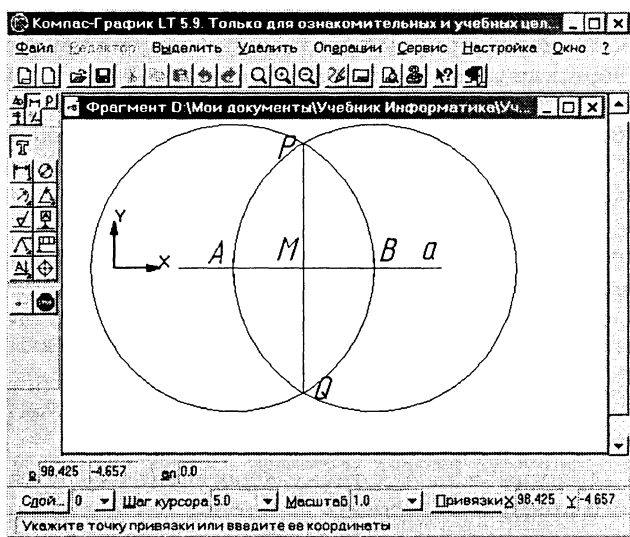
#### Построение перпендикуляра к заданной прямой

1. Построить прямую  $a$ . На панели *Геометрические построения* щелкнуть по кнопке *Ввод отрезка* и с использованием ручного ввода параметров задать координаты начальной точки  $p1$  (10,0) и конечной точки  $p2$  (70,0).

2. Построить точки  $M$ ,  $A$  и  $B$  на прямой  $a$ . На панели *Геометрические построения* щелкнуть по кнопке *Ввод точки* и с использованием ручного ввода параметров задать координаты точки  $M(40,0)$ , точки  $A(25,0)$  и точки  $B(55,0)$ .
3. Построить окружность с центром в точке  $A$  и с радиусом  $AB$ . На панели *Геометрические построения* щелкнуть по кнопке *Ввод окружности* и с использованием ручного ввода параметров задать координаты центра  $(25,0)$ .

Задать радиус окружности с использованием *Геометрического калькулятора*, для этого щелкнуть правой клавишей мыши в поле *Радиус окружности* и в появившемся меню выбрать пункт *Между двумя точками*. После того как курсор примет форму мишени, щелкнуть по точкам  $A$  и  $B$ . Окружность с заданным радиусом будет построена.

4. Аналогично построить окружность с центром в точке  $B$  и с радиусом  $AB$ .
5. Соединить точки пересечения окружностей отрезком. Задать начальную и конечную точки отрезка с использованием *Геометрического калькулятора*, выбрав пункт меню *Пересечение*.
6. Ввести на чертеже обозначения. Выбрать на *Панели управления* кнопку *Размеры и технологические обозначения*, и на появившейся панели щелкнуть по кнопке *Ввод текста*. Ввести обозначения.



7. Алгоритм построения перпендикуляра к заданной точке прямой выполнен.

**Исследование модели.** С помощью геометрических теорем необходимо доказать, что построенный отрезок  $PQ$  действительно является перпендикуляром к прямой  $a$ .

Чертеж хранится в файле `perp.frw`  
в каталоге `\practicum\inftech\cad\`

CD-ROM 

**Задача.** Дан неразвернутый угол  $A$ . Построить его биссектрису.

**Формальная модель.** Построим формальную модель процесса геометрического построения, зафиксировав его в форме алгоритма:

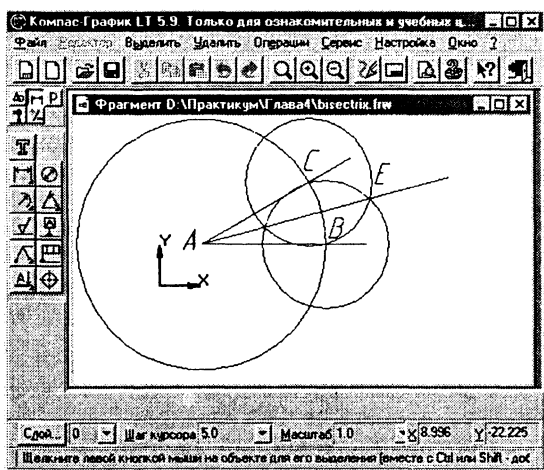
1. Построить окружность произвольного радиуса с центром в вершине заданного угла  $A$ , которая пересечет стороны угла в точках  $B$  и  $C$ .
2. Построить две окружности радиуса  $BC$  с центрами в точках  $B$  и  $C$ . Точку пересечения окружностей внутри угла обозначить буквой  $E$ .
3. Через вершину угла  $A$  и точку пересечения окружностей  $E$  провести прямую. Луч  $AE$  — биссектриса заданного угла.

**Компьютерная модель.** Реализуем геометрическое построение в соответствии с разработанным алгоритмом с использованием системы КОМПАС-3D.



### Построение биссектрисы неразвернутого угла

1. Построить неразвернутый угол и окружность с центром в точке  $A$  (вершине угла). На панели *Геометрические построения* щелкнуть по кнопке *Ввод отрезка* и построить два отрезка, выходящих из точки  $A$ . Щелкнуть по кнопке *Ввод окружности* и в автоматическом режиме построить окружность произвольного радиуса с центром в точке  $A$ .
2. Ввести обозначения точек пересечения окружности. Активизировать панель *Размеры и технологические обозначения*, щелкнуть по кнопке *Ввод текста* и ввести обозначения вершины угла  $A$  и точек пересечения окружности со сторонами угла  $B$  и  $C$ .
3. Построить две окружности одинакового радиуса с центрами в точках  $B$  и  $C$ . Задать радиусы окружностей в ручном режиме. Точку пересечения окружностей обозначить  $E$ .
4. Через вершину угла  $A$  и точку пересечения окружностей  $E$  провести прямую. Щелкнуть по кнопке *Ввод отрезка* и в автоматическом режиме последовательно указать точки  $A$  и  $E$ .
5. Алгоритм построения биссектрисы неразвернутого угла выполнен.



**Исследование модели.** С помощью геометрических теорем необходимо доказать, что построенный луч  $AE$  действительно является биссектрисой угла  $A$ .

*Чертеж хранится в файле bisectrix.frw  
в каталоге \practicum\inftech\cad\*


CD-ROM 

### Практические задания

6.4. Построить треугольник:

- а) по двум сторонам и углу между ними;
- б) по трем заданным сторонам.

## 6.2. Исследование физических моделей

Физика-9 

Рассмотрим процесс построения и исследования модели на конкретном примере движения тела, брошенного под углом к горизонту.

**Содержательная постановка задачи.** В процессе тренировок теннисистов используются автоматы по бросанию мячика в определенное место площадки. Необходимо задать автомату необходимую скорость и угол бросания мячика для попадания в площадку определенного размера, находящуюся на известном расстоянии.

**Качественная описательная модель.** Сначала построим качественную описательную модель процесса движения тела с использованием физических объектов, понятий и законов, то есть в данном случае идеализированную модель движения объекта.

Из условия задачи можно сформулировать следующие основные предположения:

- мячик мал по сравнению с Землей, поэтому его можно считать материальной точкой;
- изменение высоты мячика мало, поэтому ускорение свободного падения можно считать постоянной величиной  $g = 9,8 \text{ м/с}^2$  и движение по оси  $Y$  можно считать равноускоренным;
- скорость бросания тела мала, поэтому сопротивлением воздуха можно пренебречь и движение по оси  $X$  можно считать равномерным.

**Формальная модель.** Движение мячика по оси  $X$  равномерное, а по оси  $Y$  равноускоренное, поэтому для формализации модели используем известные из курса физики формулы равномерного и равноускоренного движения. При заданных начальной скорости  $v_0$  и угле бросания  $\alpha$  значения координат дальности полета  $x$  и высоты  $y$  от времени можно описать следующими формулами:

$$\begin{aligned} x &= v_0 \cdot \cos\alpha \cdot t, \\ y &= v_0 \cdot \sin\alpha \cdot t - g \cdot t^2/2. \end{aligned} \quad (6.1)$$

Площадка расположена на поверхности земли, поэтому из второй формулы можно выразить время, которое понадобится мячику, чтобы достичь площадки:

$$\begin{aligned} v_0 \cdot \sin\alpha \cdot t - g \cdot t^2/2 &= 0, \\ t \cdot (v_0 \cdot \sin\alpha - g \cdot t/2) &= 0. \end{aligned}$$

Значение времени  $t = 0$  не имеет физического смысла, поэтому:

$$\begin{aligned} v_0 \cdot \sin\alpha - g \cdot t/2 &= 0, \\ t &= (2 \cdot v_0 \cdot \sin\alpha)/g. \end{aligned}$$

Подставим полученное выражение для времени в формулу для вычисления координаты  $x$ :

$$x = (v_0 \cdot \cos\alpha \cdot 2 \cdot v_0 \cdot \sin\alpha)/g = (v_0^2 \cdot \sin 2\alpha)/g.$$

Формализуем теперь условие попадание мячика в площадку. Пусть площадка расположена на расстоянии  $s$  и имеет длину  $l$ . Тогда попадание произойдет, если значение координаты  $x$  мячика будет удовлетворять условию в форме неравенства:

$$s \leq x \leq s+l.$$

Если  $x < s$ , то это означает «недолет», а если  $x > s+l$ , то это означает «перелет».

**Компьютерная модель на языке Visual Basic.** Преобразуем формальную модель в компьютерную с использованием системы программирования Visual Basic. Создадим сначала графический интерфейс проекта.



## Проект «Движение тела, брошенного под углом к горизонту»

- 1 Разместить на форме шесть текстовых полей:
  - txtV0 для ввода значений начальной скорости;
  - txtA для ввода угла бросания;
  - txtS для ввода расстояния до площадки;
  - txtL для ввода длины площадки;
  - txtX для вывода координаты  $x$  падения мячика;
  - txtM для вывода текстового сообщения о результатах броска.
- 2 Поместить на форму метки для обозначения полей и единиц измерения.
- 3 Поместить на форму кнопку и создать для нее событийную процедуру, которая обеспечивает присвоение переменным значений, введенных в текстовые поля, вычисление координаты  $x$  падения мячика и вывод результатов на форму с использованием конструкции выбора **Select Case**:

```

Const G As Single = 9.81
Const Pi As Single = 3.14
Dim V0, A, S, L As Double
Private Sub CmdCalc_Click()
  'Ввод начальных значений
  V0 = Val(txtV0.Text)
  A = Val(txtA.Text)
  S = Val(txtS.Text)
  L = Val(txtL.Text)
  'Попадание в площадку
  X = V0 ^ 2 * Sin(2 * A * Pi / 180) / G
  txtX.Text = X
  Select Case X
  Case Is < S
    txtM.Text = "Недолет"
  Case Is > S+L
    txtM.Text = "Перелет"
  Case Else
    txtM.Text = "Попадание"
  End Select
End Sub

```

Для визуализации формальной модели построим траекторию движения тела (график зависимости высоты мячика над поверхностью земли от дальности полета). Снабдим график осями координат и выведем положение площадки.

- 4 Поместить на форму графическое поле, в котором будет осуществляться построение графика, и дополнить программный код событийной процедурой:

*'Построение графика*

```

For T = 0 To 10 Step 0.1
Y = V0 * Sin(A * Pi / 180) * T - G * T * T
/ 2
X = V0 * Cos(A * Pi / 180) * T
pic1.Scale (0, 15)-(S + 5, -5)
pic1.PSet (X, Y)
Next T
'Ось X
pic1.Line (0, 0)-(50, 0)
For I = 0 To 50 Step 5
pic1.PSet (I, 0)
pic1.Print I
Next I
'Ось Y
pic1.Line (0, -5)-(0, 15)
For I = -5 To 20 Step 5
pic1.PSet (0, I)
pic1.Print I
Next I
'Площадка
pic1.Line (S, 0.2)-(S + L, 0.2)

```

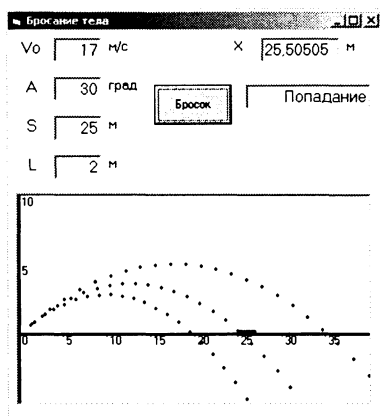
**Проект хранится в каталоге**  
**practicum\VB\Projects\prj6-4\**

**CD-ROM** 

**Компьютерный эксперимент.** Введем произвольные значения начальной скорости и угла бросания мячика; скорее всего, его попадания в площадку не будет. Затем, меняя один из параметров, например, угол, произведем пристрелку. Для этого лучше всего использовать известный артиллерийский прием «взятие в вилку», который использует наиболее эффективный метод «деление пополам». Для этого находят угол, при котором мячик перелетит площадку, затем угол, при котором мячик не долетит до площадки. Вычисляют среднее значение углов, составляющих «вилку», и смотрят, куда попадет мячик. Если он попадет в площадку, то задача выполнена, если не попадет, то рассматривается новая «вилка» и т. д.

**5** Запустить проект и ввести значения начальной скорости, угла, расстояния до площадки и ее длины.

Щелкнуть по кнопке *Бросок*. В текстовых полях будут выведены результаты, а в графическом поле появится траектория движения тела. Подобрать значения начальной скорости и угла бросания, обеспечивающие попадание в мишень.



**Анализ результатов и корректировка модели.** Модернизируем проект так, чтобы можно было получить с заданной точностью для каждого значения скорости значения диапазона углов, обеспечивающие попадание мячика в площадку.

6 Удалить с формы текстовые поля txtA для ввода значения угла и txtM для вывода результатов и графическое поле pic1.

Поместить на форму текстовые поля txtP для ввода точности определения диапазона углов и txtA1 для вывода значений этого диапазона.

7 Внести изменения в программный код событийной процедуры:

```

Private Sub CmdCalc_Click()
    'Ввод начальных значений
    V0 = Val(txtV0.Text)
    S = Val(txtS.Text)
    L = Val(txtL.Text)
    P = Val(txtP.Text)
    txtA1.Text = ""
    For A = 0 To 90 Step P
        'Попадание в площадку
        X = V0 ^ 2 * Sin(2 * A * Pi / 180) / G
        If S <= X And X <= S + L Then
            txtA1.Text = txtA1.Text + Str(A)
        End If
    Next A
End Sub
    
```

8 Запустить проект и ввести скорость бросания мячика, расстояние до площадки и ее длину, а также точность определения диапазона углов.



Бросание тела			
$V_0$	17 м/с	$L$	2 м
$S$	25 м	Точность	1 град
30 31 32 33 57 58 59 60 61			град

Диапазон углов

Анализ показывает, что получен не очевидный результат: существуют два диапазона углов от 30 до 33 и от 57 до 61 градуса, которые обеспечивают попадание мячика в площадку при заданных начальных условиях.

Проект хранится в каталоге  
 \practicum\VB\Projects\prj6-5\

CD-ROM 

### Практические задания

- 6.5. Рассмотренную выше модель бросания мячика под углом к горизонту исследовать в электронных таблицах. При заданных начальных условиях (расстояние до площадки 25 м, ее длина 2 м и начальная скорость 17 м/с) определить диапазоны углов, обеспечивающих попадание в площадку, с точностью до 0,1 градуса.

## 6.3. Биологические модели развития популяций

Общая биология 10-11 

В биологии при исследовании развития биосистем строятся динамические модели изменения численности популяций различных живых существ (бактерий, рыб, животных и так далее) с учетом различных факторов. Взаимовлияние популяций рассматривается в моделях типа «хищник–жертва».

**Формальная модель.** Изучение динамики популяций естественно начать с простейшей модели *неограниченного роста*, в которой численность популяции ежегодно увеличивается на определенный процент. Математическую модель можно записать с помощью рекуррентной формулы, связывающей численность популяции следующего года с численностью популяции текущего года, с использованием коэффициента роста  $a$ :

$$x_{n+1} = a \cdot x_n.$$

Например, если ежегодный прирост численности популяции составляет 5%, то  $a = 1,05$ .

В модели *ограниченного роста* учитывается эффект перенаселенности, связанный с нехваткой питания, болезнями и так да-

лее, который замедляет рост популяции с увеличением ее численности. Введем коэффициент перенаселенности  $b$ , значение которого обычно существенно меньше  $a$  ( $b < a$ ). Тогда коэффициент ежегодного увеличения численности равен  $(a - b \cdot x_n)$  и формула принимает вид:

$$x_{n+1} = (a - b \cdot x_n) \cdot x_n.$$

В модели *ограниченного роста с отловом* учитывается, что на численность популяций промысловых животных и рыб оказывает влияние величина ежегодного отлова. Если величина ежегодного отлова равна  $c$ , то формула принимает вид:

$$x_{n+1} = (a - b \cdot x_n) \cdot x_n - c.$$

Популяции обычно существуют не изолированно, а во взаимодействии с другими популяциями. Наиболее важным типом является взаимодействие между жертвами и хищниками (например, караси-щуки, зайцы-волки и так далее). В модели жертва-хищник количество жертв  $x_n$  и количество хищников  $y_n$  связаны между собой. Количество встреч жертв с хищниками можно считать пропорциональной произведению собственно количеств жертв и хищников, а коэффициент  $f$  характеризует возможность гибели жертвы при встрече с хищниками. В этом случае численность популяции жертв уменьшается на величину  $f \cdot x_n \cdot y_n$  и формула для расчета численности жертв принимает вид:

$$x_{n+1} = (a - b \cdot x_n) \cdot x_n - c - f \cdot x_n \cdot y_n.$$

Численность популяции хищников в отсутствие жертв (в связи с отсутствием пищи) уменьшается, что можно описать рекуррентной формулой

$$y_{n+1} = d \cdot y_n,$$

где значение коэффициента  $d < 1$  характеризует скорость уменьшения численности популяции хищников.

Увеличение популяции хищников можно считать пропорциональным произведению собственно количеств жертв и хищников, а коэффициент  $e$  характеризует величину роста численности хищников за счет жертв. Тогда для численности хищников можно использовать формулу:

$$y_{n+1} = d \cdot y_n + e \cdot x_n \cdot y_n.$$

**Компьютерная модель.** Построим на языке Visual Basic компьютерную модель, позволяющую исследовать изменение со временем численности популяций с использованием различных моделей: неограниченного роста, ограниченного роста, ограниченного роста с отловом и жертва-хищник.



## Рост численности популяций

- 1 На форму поместить текстовые поля для ввода:
  - значений коэффициентов  $a$ ,  $b$ ,  $c$  и  $f$ , влияющих на изменение численности жертв: txtA, txtB, txtC и txtF;
  - значений коэффициентов  $d$  и  $e$ , влияющих на изменение численности хищников: txtD и txtE.
  - начальной численности популяций жертв и хищников: txtX1 и txtY1.
  - количества рассматриваемых жизненных циклов (лет) txtN.
- 2 На форму поместить текстовые поля для вывода численности популяции через заданное количество лет:
  - при неограниченном росте txtXnn;
  - при ограниченном росте txtXno;
  - при ограниченном росте с отловом txtXnoo;
  - в модели жертва-хищник txtX\_Y и txtY\_X.
- 3 Поместить на форму кнопку cmd1 и начать создание событийной процедуры. Прежде всего, обеспечить ввод начальных данных в соответствующие переменные:

```
Dim bytN As Byte, intA, intB, intC, intD,
intE, intF, intX, intY As Integer
Private Sub cmd1_Click()
  'Ввод данных
  bytN = txtN.Text
  intA = txtA.Text
  intB = txtB.Text
  intC = txtC.Text
  intD = txtD.Text
  intE = txtE.Text
  intF = txtF.Text
  intX = txtX1.Text
  intY = txtY1.Text
End Sub
```

- 4 Поместить на форму графическое поле, в котором будут строиться графики зависимости численности популяций от количества прошедших лет. В событийную процедуру ввести:

```
'Масштаб и координаты
pic1.Scale (-1, 10)-(bytN + 1, -1)
pic1.Line (0, 0)-(bytN + 1, 0)
For I = 0 To bytN
  pic1.PSet (I, 0)
  pic1.Print I
Next I
```

```
pic1.Line (0, -1)-(0, 10)
For I = 0 To 10
pic1.PSet (0, I)
pic1.Print I
Next I
```

- 5 Ввести код модели *неограниченного роста*, где в цикле вычисляется численность популяции и строится график:

```
For I = 1 To bytN
pic1.PSet (I, intX)
intX = intA * intX
txtXnn = intX
Next I
```

- 6 Ввести код модели *ограниченного роста*:

```
intX = txtX1.Text
For I = 1 To bytN
pic1.PSet (I, intX), vbMagenta
intX = (intA - intB * intX) * intX
txtXno = intX
Next I
```

- 7 Ввести код модели *ограниченного роста с отловом*:

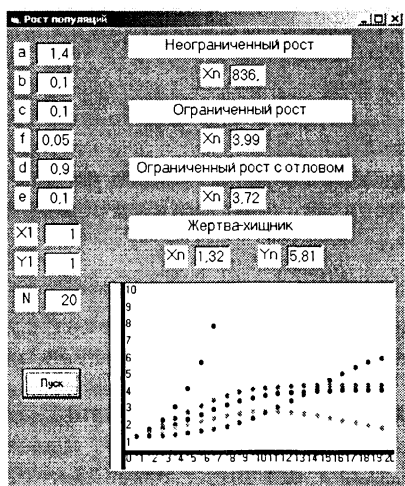
```
intX = txtX1.Text
For I = 1 To bytN
pic1.PSet (I, intX), vbBlue
intX = (intA - intB * intX) * intX - intC
txtXnoo = intX
Next I
```

- 8 Ввести код модели *жертва-хищник*:

```
intX = txtX1.Text
For I = 1 To bytN
pic1.PSet (I, intX), vbGreen
pic1.PSet (I, intY), vbRed
intX = (intA - intB * intX) * intX - intC -
intF * intX * intY
intY = intD * intY + intE * intX * intY
txtX_Y = intX
txtY_X = intY
Next I
```

- 9 Запустить проект и ввести значения коэффициентов, начального количества жертв и хищников, количества жизненных циклов.

Графики показывают динамику развития популяций.



Проект хранится в каталоге  
[\practicum\VB\Projects\prj6-6\](#)

CD-ROM 

### Практические задания

- 6.6. Провести исследование моделей роста популяций различного типа, задавая различные значения коэффициентов и начальные численности популяций. Подобрать значения коэффициентов, чтобы:
- численность популяции в моделях *ограниченного роста* и *ограниченного роста с отловом* стабилизировалась примерно на одном уровне (так определяют квоты на ловлю рыбы);
  - в модели *жертва-хищник* численность жертв и хищников стабилизировалась со временем (так определяют охотничьи квоты).

## 6.4. Геоинформационные модели

Геоинформационное моделирование базируется на создании многослойных электронных карт, в которых опорный слой описывает географию определенной территории, а каждый из остальных — один из аспектов состояния этой территории. На географическую карту могут быть выведены различные слои объектов: города, дороги, аэропорты и др.

Широкое распространение получили интерактивные географические карты (мира, различных частей света, России, Москвы и других городов). Такие карты обычно реализуются с использованием векторной графики и поэтому позволяют пользователю выбирать нужный ему масштаб.

Карты связаны с базами данных, которые хранят всю необходимую информацию об объектах, изображенных на картах. Пользователь может осуществлять поиск необходимого ему объекта на карте с помощью поисковой системы. Например, для того чтобы найти дом на интерактивной карте Москвы требуется ввести название улицы и номер дома.



## Интерактивная карта г. Москвы

- 1 Запустить геоинформационную систему Карта Москвы. Ввести команду [Поиск-Адресный поиск]. В диалоговом окне *Поиск по адресу* выбрать из списка название улицы и в текстовые поля ввести номер дома, корпус и так далее. Щелкнуть по кнопке *Найти дом*, результат поиска в базе данных будет выведен в нижнее текстовое поле. Если дом найден, щелкнуть по кнопке *Показать дом*.

**Поиск по адресу**

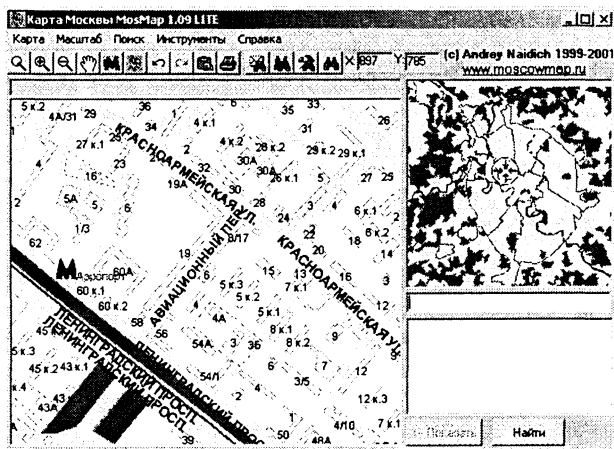
Улица

АВИАМОТОРНАЯ УЛ. (МОСКВА)	▲
АВИАТОРОВ УЛ. (МОСКВА)	▲
АВИАЦИОННАЯ УЛ. (МОСКВА)	▲
АВИАЦИОННАЯ УЛ. (ЩЕРБИНКА)	▲
<b>АВИАЦИОННЫЙ ПЕР. (МОСКВА)</b>	▲
АВТОЗАВОДСКАЯ ПЛ. (МОСКВА)	▼

По списку  В старых названиях  Только по первым буквам

Номер  Буква  Дробь  Корпус

- 2 На появившемся фрагменте карты найденный дом будет выделен более темным цветом. Щелчком по кнопкам *Ближе* и *Дальше* можно изменять масштаб карты. Щелчок по кнопке *Сдвинуть* позволяет производить перемещение видимого фрагмента карты.



Геоинформационные модели позволяют с помощью географических карт представлять статистическую информацию о различных регионах. Хранящаяся в базах данных информация о количестве населения, развитии промышленности, загрязнении окружающей среды и др. может быть связана с географическими картами и отображена на них. Отображение информации может производиться различными способами: закрасиванием регионов различными цветами, построением диаграмм и так далее.

Построим в электронных таблицах Excel с использованием специальной надстройки Microsoft Data Map геоинформационную модель, отображающую информацию о количестве населения в различных странах Европы.



### Геоинформационная модель «Численность населения в странах Европы»

- 1 Запустить установку Microsoft Office и установить надстройку Microsoft Data Map.

Открыть в электронных таблицах файл mapstats.xls, который записывается на диск в процессе установки Microsoft Office.

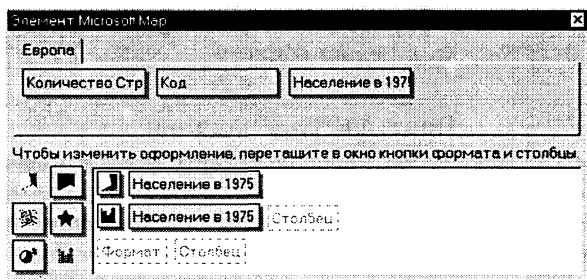
В листах электронных таблиц содержатся базы данных по численности населения стран мира и некоторых стран Европы.

	А	В	С
1	Код	Страна	Население в 1975
2	AT	АВСТРИЯ	7578903
3	AL	АЛБАНИЯ	2401602
4	AD	АНДОРРА	26568
5	BE	БЕЛЬГИЯ	9788200
6	BG	БОЛГАРИЯ	8720742
7	GB	ВЕЛИКОБРИТАНИЯ	56214800
8	HU	ВЕНГРИЯ	10531820
9	DE	ГЕРМАНИЯ	61991500
10	GR	ГРЕЦИЯ	9046431
11	DK	ДАНИЯ	5054410
12	IE	ИРЛАНДИЯ	3118100
13	IS	ИСЛАНДИЯ	218031
14	ES	ИСПАНИЯ	36208615
15	IT	ИТАЛИЯ	55293036
16	LI	ЛИХТЕНШТЕЙН	23413
17	LU	ЛЮКСЕМБУРГ	357400
18	MC	МОНАКО	25143
19	NL	НИДЕРЛАНДЫ	13599092
20	NO	НОРВЕГИЯ	4007313

- 2 Для вызова карты щелкнуть по кнопке *Карта*. В появившейся диалоговой панели *Обнаружено несколько карт* выбрать требуемую карту, например, *Европа*.
- 3 Для вывода на карту статистических данных ввести команду [Сервис-Данные]. Выделить на листе электронных таблиц столбцы, содержащие названия стран и численность населения. Созданная карта отображается на листе и открывается диалоговая панель *Оформление карты*.

Статистические данные на карте могут отображаться различными способами. Форматы *Тоновая заливка*, *Цветовая заливка* и *Плотность точек* позволяют отобразить один ряд данных, а форматы *Круговая диаграмма* и *Гистограмма* могут отображать несколько рядов данных.

- 4 На диалоговой панели *Оформление карты* выбрать форматы *Тоновая заливка* и *Гистограмма*.



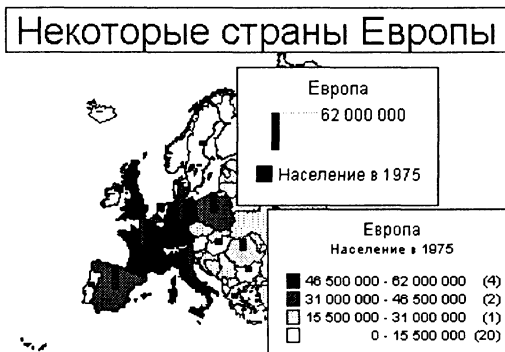
В выбранном формате *Тоновая заливка* значения численности населения разбиваются на интервалы, и страны, входящие в каждый из интервалов, закрашиваются своим оттенком цвета. Количество интервалов и способ разбиения на интервалы можно изменить.

- 5 Щелкнуть по кнопке *Тоновая заливка*. На появившейся диалоговой модели *Формат* установить количество интервалов (например, 4), способ разбиения по интервалам и базовый цвет.

В выбранном формате *Гистограмма* значение численности населения в каждой стране представляется столбцом гистограммы, высота которого пропорциональна численности. Масштаб столбцов гистограммы, их ширину и цвет можно изменить.

- 6 Щелкнуть по кнопке *Гистограмма*. На появившейся диалоговой модели *Формат* установить подходящие параметры.
- 7 На карте каждая из стран будет закрашена одним из четырех оттенков выбранного цвета и представлена столбцом гистограммы.





*Легенда карты содержит информацию о параметрах тоновой заливки и гистограммы.*

*Геоинформационная модель хранится в файле mapstats.xls в каталоге \textbook\Excel*

CD-ROM

### Практические задания

- 6.7. Найти в Интернете интерактивную карту России и вывести карту своего региона.
- 6.8. Создать геоинформационную модель, отображающую статистические данные о численности населения стран мира.

## 6.5. Химические модели

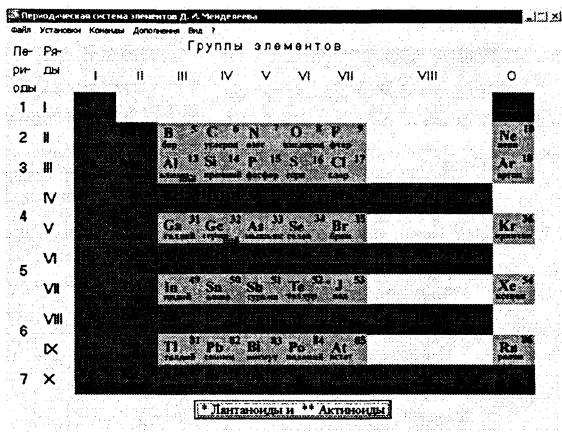
Химия 9

Представление объектов и их свойств в форме таблицы часто используется в научных исследованиях. Так, периодическая система элементов Д. И. Менделеева представляет собой табличную информационную модель, в которой химические элементы располагаются в ячейках таблицы по возрастанию атомных весов, а в столбцах — по количеству валентных электронов, причем по положению в таблице можно определить некоторые физические и химические свойства элементов.

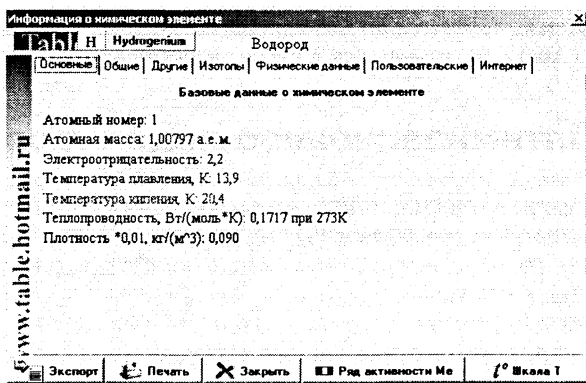
Компьютерная модель периодической системы позволяет в интерактивном режиме знакомиться с различными физическими и химическими свойствами химических элементов (атомная масса, электропроводность, плотность и так далее), уравнивать формулы химических реакций, решать стандартные химические задачи на нахождение массы веществ, участвующих в реакции и др.

## Периодическая таблица элементов Д. И. Менделеева

1 Запустить программу Table. С помощью меню *Отображать* выбрать физическое или химическое свойство элементов, значение которого будет выводиться в ячейки таблицы.

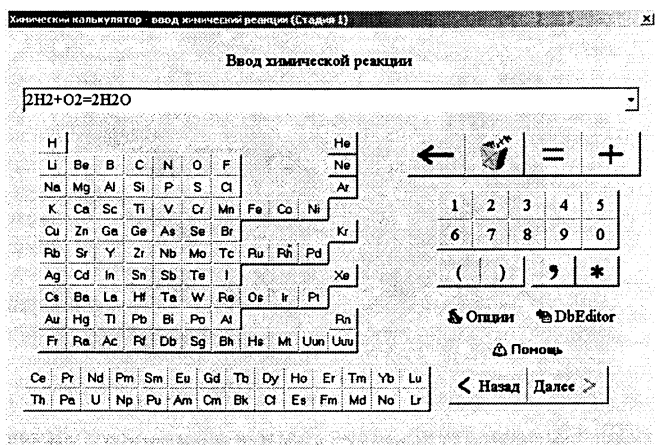


2 Для ознакомления с физическими и химическими свойствами элемента осуществить двойной щелчок на ячейке с названием элемента (например, водорода). На появившейся информационной панели элемента выбрать одну из вкладок и ознакомиться с физическими и химическими свойствами элемента.



Компьютерная модель периодической системы элементов имеет в своем составе химический калькулятор, который позволяет строить модели химических реакций (записывать их уравнения).

- 3 Ввести команду [Команды-Химический калькулятор...].  
 На появившейся панели *Химический калькулятор* в окно *Ввод химической реакции* ввести вещества (например,  $\text{H}_2 + \text{O}_2$  и нажать клавишу «=».  
 В окне появится уравнение химической реакции:  
 $2\text{H}_2 + \text{O}_2 = 2\text{H}_2\text{O}$ .



## Практические задания

- 6.9. С помощью компьютерной модели периодической системы Д. И. Менделеева ознакомиться с физическими и химическими свойствами элементов.
- 6.10. С помощью *Химического калькулятора* записать уравнения химических реакций, изученных в курсе неорганической химии.

## 6.6. Оптимизационное моделирование

В сфере управления сложными системами (например, в экономике) применяется оптимизационное моделирование, в процессе которого осуществляется поиск наиболее оптимального пути развития системы.

Критерием оптимальности могут быть различные параметры, например, в экономике можно стремиться к максимальному количеству выпускаемой продукции, а можно к ее низкой себестоимости. Оптимальное развитие соответствует экстремальному (максимальному или минимальному) значению выбранного целевого параметра.

Развитие сложных систем зависит от множества факторов (параметров), следовательно, значение целевого параметра за-

висит от множества параметров. Выражением такой зависимости является целевая функция

$$K = F(X_1, X_2, \dots, X_n),$$

где  $K$  — значение целевого параметра;

$X_1, X_2, \dots, X_n$  — параметры, влияющие на развитие системы.

Цель исследования состоит в нахождении экстремума этой функции и определении значений параметров, при которых этот экстремум достигается. Если целевая функция нелинейна, то она имеет экстремумы, которые находятся определенными методами.

Однако часто целевая функция линейна и, соответственно, экстремумов не имеет. Задача поиска оптимального режима при линейной зависимости приобретает смысл только при наличии определенных ограничений на параметры.

Рассмотрим в качестве примера моделирования поиск вариантов оптимальной погрузки при перевозке компьютерного класса.

**Содержательная постановка проблемы.** При получении школы нового компьютерного класса необходимо оптимально спланировать использование единственного легкового автомобиля для перевозки 15 компьютеров. Каждый компьютер упакован в две коробки (монитор и системный блок) и существуют три варианта погрузки коробок в автомобиль.

Таблица 6.1

**Способы погрузки**

Тип коробки	Варианты погрузки		
	1	2	3
Мониторы	3	2	1
Системный блок	1	2	4

Необходимо выбрать оптимальное сочетание вариантов погрузки для того, чтобы перевести 15 коробок с мониторами и 15 коробок с системными блоками за минимальное количество рейсов автомобиля.

**Формальная модель.** Параметрами, значения которых требуется определить, являются количества рейсов автомобиля, загруженного различными способами:

$X_1$  — количество рейсов автомобиля, загруженного по варианту 1;

$X_2$  — количество рейсов автомобиля, загруженного по варианту 2;

$X_3$  — количество рейсов автомобиля, загруженного по варианту 3.

Тогда целевая функция, равная количеству рейсов автомобиля, примет вид:

$$F = X_1 + X_2 + X_3.$$

Ограничения накладываются количествами коробок с мониторами и системными блоками, которые необходимо перевезти. Должны выполняться два равенства:

$$\begin{aligned} 3 \cdot X_1 + 2 \cdot X_2 + 1 \cdot X_3 &= 15, \\ 1 \cdot X_1 + 2 \cdot X_2 + 4 \cdot X_3 &= 15. \end{aligned}$$

Кроме того, количества рейсов не могут быть отрицательными, поэтому должны выполняться неравенства:

$$X_1 \geq 0; \quad X_2 \geq 0; \quad X_3 \geq 0.$$

Таким образом, необходимо найти удовлетворяющие ограничениям значения параметров, при которых целевая функция принимает минимальное значение.

**Компьютерная модель.** Будем искать решение задачи путем создания и исследования компьютерной модели в электронных таблицах Excel.



### Оптимизационное моделирование

1 Ячейки B2, C2 и D2 выделить для хранения значений параметров  $X_1$ ,  $X_2$  и  $X_3$ .

В ячейку B4 ввести формулу вычисления целевой функции:  $=B2+C2+D2$ .

В ячейку B7 ввести формулу вычисления количества коробок с мониторами:

$$=3*B2 + 2*C2 + 1*D2.$$

В ячейку B8 ввести формулу вычисления количества коробок с системными блоками:

$$=1*B2 + 2*C2 + 4*D2.$$

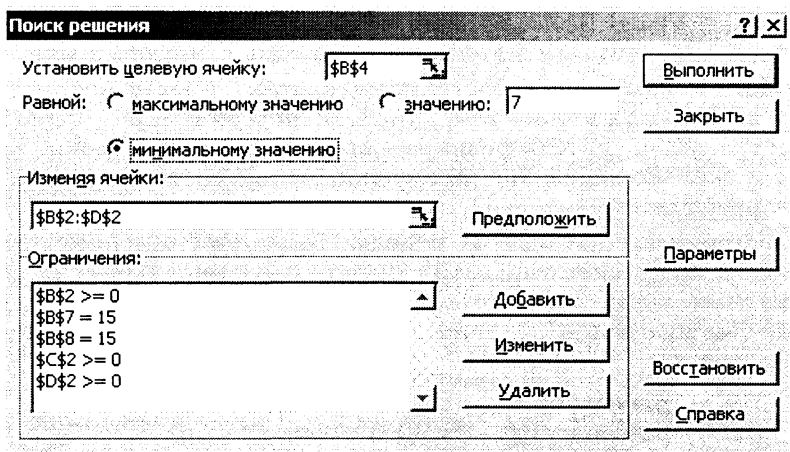
**Исследование модели.** Для поиска оптимального набора значений параметров, который соответствует минимальному значению целевой функции, воспользоваться надстройкой электронных таблиц *Поиск решений*.

2 Для активизации надстройки ввести команду [Сервис-Надстройки...]. На диалоговой панели поставить флажок перед элементом списка *Поиск решения*.

3 Ввести команду [Сервис-Поиск решений...]. На появившейся диалоговой панели *Поиск решения* установить:

- адрес целевой ячейки;
- вариант оптимизации значения целевой ячейки (максимизация, минимизация или подбор значения);

- адреса ячеек, значения которых изменяются в процессе поиска решения (в которых хранятся значения параметров);
- ограничения (типа *равно* для ячеек, хранящих количество деталей, и типа *больше или равно* для параметров).



4 Щелкнуть по кнопке *Выполнить*.

В ячейке целевой функции появится значение 7, а в ячейках параметров значения 3, 2, 2.

	A	B	C	D
1		X1	X2	X3
2	Параметры:	3	2	2
3				
4	Целевая функция:	7		
5				
6	<b>Ограничения</b>			
7	Кол-во коробок с мониторами:	15		
8	Кол-во коробок с системными блоками:	15		

Таким образом, для перевозки 15 коробок с мониторами и 15 коробок с системными блоками потребуется 7 рейсов автомобиля, при этом 3 рейса должны быть загружены по первому, 2 рейса по второму и 2 рейса по третьему варианту.

Модель хранится в файле model.xls  
в каталоге \practicum\inftech\calc\

CD-ROM

**Практические задания**

6.11. Построить компьютерную модель оптимизации перевозки компьютерного класса на языке Visual Basic.

## 6.7. Логические модели

При решении логических задач можно использовать компьютерные логические модели. Создадим компьютерную модель на языке Visual Basic, которая позволит составить расписание уроков для учителей математики, физики и информатики.

**Логическая задача.** Учитель математики просит поставить ему первый или второй урок, учитель информатики — первый или третий, а учитель физики — второй или третий уроки. Какие и сколько вариантов расписания можно составить, учитывая пожелания учителей?

**Формальная логическая модель.** Прежде всего, необходимо каждое пожелание учителей записать в форме логических переменных, введя соответствующие обозначения для используемых высказываний. В нашем случае используем следующие обозначения:

- $M1$  — {математика — первый урок},
- $M2$  — {математика — второй урок},
- $I1$  — {информатика — первый урок},
- $I3$  — {информатика — третий урок},
- $F2$  — {физика — второй урок},
- $F3$  — {физика — третий урок}.

Из условия задачи следует, что каждый учитель должен провести хотя бы один урок, следовательно, можно записать следующие логические условия:

$$M1 \vee M2 = 1;$$

$$I1 \vee I3 = 1;$$

$$F2 \vee F3 = 1.$$

Необходимо учесть также утверждения, содержащиеся в условии задачи неявно, а именно: «ни один из предметов не может быть дважды» и «на одном уроке не могут быть одновременно два предмета». Таким образом, не могут быть истинны шесть составных высказываний, образованных из пар простых высказываний ( $M1$  и  $M2$ ,  $I1$  и  $I3$ ,  $F2$  и  $F3$ ,  $M1$  и  $I1$ ,  $M2$  и  $F2$ ,  $I3$  и  $F3$ ) с помощью операции логического сложения.

Запишем составные высказывания в форме логических выражений и осуществим тождественное преобразование логических выражений, так чтобы они были равны единице. Если одно из двух взаимоисключающих высказываний ложно, то хотя бы одно из их отрицаний истинно, то есть логическое умножение простых высказываний можно заменить логическим сложением их отрицаний:

$$\begin{array}{ll}
 M1 \ \& \ M2 = 0 & \overline{M1} \vee \overline{M2} = 1; \\
 I1 \ \& \ I3 = 0 & \overline{I1} \vee \overline{I3} = 1; \\
 F2 \ \& \ F3 = 0 & \overline{F2} \vee \overline{F3} = 1; \\
 M1 \ \& \ I1 = 0 & \overline{M1} \vee \overline{I1} = 1; \\
 M2 \ \& \ F2 = 0 & \overline{M2} \vee \overline{F2} = 1; \\
 I3 \ \& \ F3 = 0 & \overline{I3} \vee \overline{F3} = 1.
 \end{array}$$

Полученные девять логических условий должны быть одновременно истинны, т. е. должно быть истинно составное высказывание, в котором логически перемножаются все исходные логические условия:

$$(M1 \vee M2) \ \& \ (I1 \vee I3) \ \& \ (F2 \vee F3) \ \& \ (\overline{M1} \vee \overline{M2}) \ \& \ (\overline{I1} \vee \overline{I3}) \ \& \ (\overline{F2} \vee \overline{F3}) \ \& \ (\overline{M1} \vee \overline{I1}) \ \& \ (\overline{M2} \vee \overline{F2}) \ \& \ (\overline{I3} \vee \overline{F3}) = 1.$$

Задача, таким образом, свелась к определению всех возможных комбинаций исходных данных, то есть  $M1, M2, I1, I3, F2, F3$ , при которых логическое выражение истинно.

**Компьютерная модель.** Таким образом, задача сводится к перебору всех возможных комбинаций логических аргументов и определению значений логического выражения для каждой такой комбинации. Это можно реализовать с помощью вложенных циклов со счетчиком, в каждом из которых рассматриваются два значения аргументов: **True** и **False**.

Однако в цикле со счетчиком переменная Счетчик должна быть обязательно числового типа, логические значения она принимать не может. Поэтому необходимо использовать числовую форму представления логических значений, логическому значению **False** соответствует число 0, а логическому значению **True** соответствует -1.

Перебор вариантов будем осуществлять с помощью вложенных циклов, а для того чтобы таблицы истинности выводились программой в привычном виде, необходимо при выводе их на печать перед аргументами и функцией ставить знак «-».



### Решение логической задачи

1 Поместить на форму кнопку и создать событийную процедуру, определяющую истинность логического выражения и печатающую на форме значения логических аргументов, при которых истинность достигается:

```

Private Sub cmd1_Click()
For M1 = -1 To 0
For M2 = -1 To 0
For I1 = -1 To 0
For I3 = -1 To 0

```



```

For F2 = -1 To 0
For F3 = -1 To 0
If (M1 Or M2) And (I1 Or I3) And (F2 Or F3)
And (Not M1 Or Not M2) And (Not I1 Or Not I3)
And (Not F2 Or Not F3) And (Not M1 Or Not I1)
And (Not M2 Or Not F2) And (Not I3 Or Not F3)
= -1
Then frm1.Print -M1; -M2; -I1; -I3; -F2; -F3
Next F3
Next F2
Next I3
Next I1
Next M2
Next M1
End Sub

```

- 2 Запустить проект и щелкнуть по кнопке *Составить расписание*. На форме будут напечатаны два набора значений логических переменных, при которых выполняются все логические условия.



В первом варианте истинны  $M1$ ,  $I3$ ,  $F2$ , то есть первый урок — математика, второй — физика, третий — информатика.

Во втором варианте истинны логические переменные  $M2$ ,  $I1$ ,  $F3$ , то есть первый урок — информатика, второй — математика, третий — физика.

---

Проект хранится в каталоге  
\practicum\VB\Projects\prjLog\

---

CD-ROM 

## Практические задания

- 6.12. Построить в электронных таблицах компьютерную модель одноразрядного полусумматора двоичных чисел.

## 6.8. Информационные модели управления объектами

В процессе функционирования сложных систем (биологических, технических и так далее) входящие в них объекты постоянно обмениваются информацией. Изменение сложных систем во времени имеет свои особенности. Так, для поддержания своей жизнедеятельности любой живой организм постоянно получает информацию из внешнего мира с помощью органов чувств, обрабатывает ее и управляет своим поведением (например, перемещаясь в пространстве, избегает опасности).

В любом процессе управления всегда происходит взаимодействие двух объектов — *управляющего* и *управляемого*, которые соединены каналами *прямой* и *обратной* связи. По каналу прямой связи передаются управляющие сигналы, а по каналу обратной связи — информация о состоянии управляемого объекта.

**Разомкнутые системы управления.** Если в процессе управления не учитывается состояние управляемого объекта и обеспечивается управление только по прямому каналу (от управляющего объекта к управляемому), то такие системы управления называются *разомкнутыми*. Информационную модель разомкнутой системы управления можно наглядно представить с помощью следующей схемы:

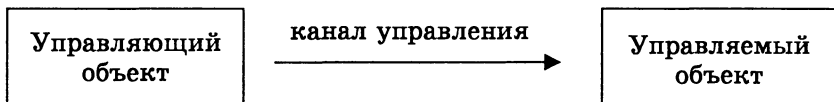


Рис. 6.2. Разомкнутая система управления

Для демонстрации принципа работы разомкнутых систем управления разработаем компьютерную модель на языке программирования Visual Basic. Пусть управляемым объектом будет точка, которую управляющий объект (пользователь) должен переместить в центр мишени (круга). Прямое управление положением точки будем производить путем нажатия на кнопки, которые перемещают объект вверх, вниз, влево и вправо. Обратная связь будет отсутствовать.

### Модель разомкнутой системы управления

- 1 Поместить на форму графическое поле, по которому будет перемещаться точка, кнопку для вывода первоначального положения точки, четыре кнопки для управления движением точки и кнопку для вывода положения мишени.

- 2 Событийная процедура первоначального вывода точки должна включать задание масштаба и случайную генерацию координат точки:

```
Dim bytX1, bytY1, bytX2, bytY2 As Byte
Private Sub cmdP_Click()
pic1.Scale (0, 20)-(20, 0)
bytX1 = Int(Rnd * 20)
bytY1 = Int(Rnd * 20)
pic1.PSet (bytX1, bytY1), vbRed
End Sub
```

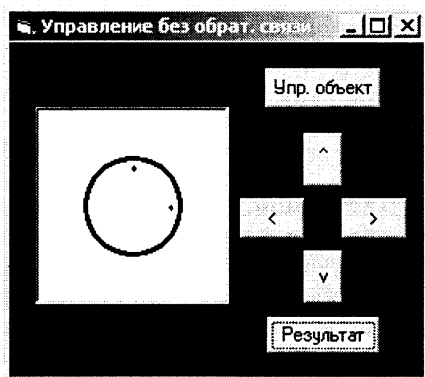
- 3 Четыре событийные процедуры перемещения точки должны обеспечивать изменение координат точки. Для перемещения влево событийная процедура:

```
Private Sub cmdL_Click()
pic1.Scale (0, 20)-(20, 0)
bytX1 = bytX1 - 1
End Sub
```

- 4 Событийная процедура вывода мишени и положения точки:

```
Private Sub cmd2_Click()
pic1.Scale (0, 20)-(20, 0)
pic1.Circle (10, 10), 5
pic1.PSet (bytX1, bytY1), vbBlack
End Sub
```

- 5 Щелкнуть по кнопке *Упр. объект* и перемещать его кнопками со стрелками. Щелкнуть по кнопке *Результат*. Отклонение точки от центра мишени, скорее всего, будет достаточно велико.



Проект хранится в каталоге  
 \practicum\WB\Projects\prjUpr1\ CD-ROM



**Замкнутые системы управления.** В замкнутых системах управления управляющий объект по прямому каналу управления производит необходимые действия над объектом управле-

ния, а по каналу обратной связи получает информацию о его реальных параметрах. Это позволяет осуществлять управление с гораздо большей точностью.

Информационную модель замкнутой системы управления можно наглядно представить с помощью следующей схемы:

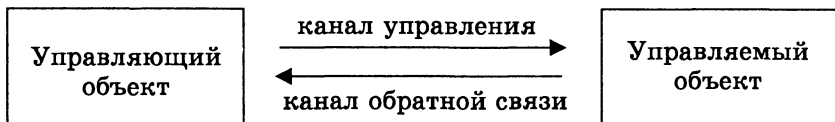


Рис. 6.3. Замкнутая система управления

Для демонстрации принципа работы замкнутых систем управления разработаем компьютерную модель. Для осуществления обратной связи будем при каждом шаге рисовать новое положение точки, а также выводить значения координат точки в текстовые поля.

 **Модель замкнутой системы управления**

- 1 Усовершенствовать предыдущий проект и поместить на форму два текстовых поля.

В коды процедур добавить строки:

```
pic1.PSet (bytX1,
bytY1), vbRed
txtX.Text = bytX1
txtY.Text = bytY1
```



Использование обратной связи обеспечивает гарантированное попадание точки в мишень.

*Проект хранится в каталоге  
|practicum\VB\Projects\prjUpr2\*

CD-ROM

**Практические задания**

- 6.13. Создать компьютерную модель замкнутой системы управления с автоматической обратной связью.

# Глава 7

---

## Информационные ресурсы Интернета

---

В процессе изучения темы  
рекомендуется установить:

CD-ROM 

- программы для тестирования параметров соединения с Интернетом AnalogX HyperTrace, VitalAgent и Modemgph;
- интегрированные приложения для работы в Интернете Microsoft Internet Explorer, Сибкон Коммуникатор, NeoPlanet, Opera;
- Интернет-утилиту NetSonic, ускоряющую загрузку Web-страниц;
- менеджеры загрузки файлов Go!Zilla и Reget;
- FTP-клиенты AceFTP и CuteFTP;
- off-line браузеры WebDownloader и WebZip;
- программу русификации приложений LingoWare;
- систему интерактивного общения ICQ;
- мультимедиа-проигрыватели RealPlayer, Windows Media Player, WinAmp, MusicMatch Jukebox.

Microsoft Internet Explorer интегрирует браузер (Обозреватель) и почтовый клиент (Outlook Express). Приложение локализовано и русифицировано, а его интерфейс является традиционным интерфейсом приложения Windows. Пользователь может произвести настройку большого количества параметров приложения. Приложение включено в состав операционной системы Windows и является одним из наиболее популярных у пользователей сети.

Сибкон Коммуникатор является полностью локализованной и русифицированной версией Netscape Communicator 4.7 и интегрирует браузер (Коммуникатор), почтовый клиент (Почта) и редактор Web-страниц (Web-мастер).

NeoPlanet интегрирует браузер, почтовый клиент (E-Mail Messages) и файловый менеджер для работы с файловыми архивами (Download Manager). Важным достоинством Download Manager является возможность продолжения скачивания файла после разрыва соединения. Пользователь может произвести настройку бо-

льшого количества параметров приложения, причем часть настроек импортируется из Microsoft Internet Explorer. Приложение обладает приятным дизайном, причем пользователь может скачать из Интернета (с сервера [www.neoplanet.com](http://www.neoplanet.com)) несколько сотен вариантов графического интерфейса приложения (skins).

Орега получила широкую известность за высокую скорость работы и удобный русскоязычный интерфейс со множеством настроек. Включает почтовый клиент, клиент конференций новостей и программу общения, совместимую с ICQ. Встроенная функция OperaShow позволяет использовать HTML-документы для презентаций. Орега хорошо подходит для компьютеров с небольшими винчестером и оперативной памятью.

## 7.1. Настройка параметров соединения с Интернетом

### 7.1.1. Подключение к Интернету по коммутируемым телефонным каналам

Для большинства индивидуальных пользователей доступ по цене является доступ в Интернет по коммутируемым телефонным каналам со скоростью до 56 Кбит/с. Подключение по коммутируемым телефонным линиям обеспечивается провайдерами услуг Интернета.

При выборе провайдера целесообразно, во-первых, получить от него информацию о количестве и пропускной способности каналов, соединяющих его с Интернетом, чем больше таких каналов и их пропускная способность, тем надежнее и быстрее ваша работа в Интернете. Во-вторых, полезно знать, каким количеством входных телефонных линий располагает провайдер, чем оно больше, тем быстрее можно до него дозвониться. Наконец, в-третьих, в целях экономии необходимо выяснить схему оплаты и стоимость подключения.

**Скорость модуляции и скорость передачи данных.** Для подключения компьютера к телефонной линии используется модем. На выходе компьютера данные представлены в форме последовательностей электрических импульсов, которые модем преобразует в модулированный аналоговый сигнал.

В простейшем случае производится амплитудная модуляция, когда амплитуда несущей (аналогового синусоидального сигнала) скачкообразно меняется и принимает одно из двух возможных значений. В процессе частотной модуляции скачкообразно меняется частота аналогового сигнала, а при фазовой модуляции — фаза сигнала. В каждом из этих случаев имеется

два возможных состояния несущего аналогового сигнала и, соответственно, каждое его изменение несет информацию 1 бит.

*Скорость модуляции* характеризуется количеством переключений с одного аналогового сигнала на другой за 1 секунду и измеряется в *бодах*. Например, если за одну секунду изменение аналогового сигнала происходит 9600 раз, то скорость модуляции составляет 9600 бод. Однако практически важной характеристикой модема является *скорость передачи данных*, измеряемая в *битах в секунду* (сокращенно бит/с).

Если применяется только один вид модуляции (например, только амплитудная модуляция), то 1 бод соответствует 1 бит/с и скорость передачи данных в вышеуказанном случае составляет 9600 бит/с. Если одновременно реализуются два вида модуляции (например, амплитудная и частотная), то при каждом изменении аналогового сигнала происходит выбор одного из четырех состояний и, следовательно, каждое такое изменение несет 2 бита информации.

**Задание 7.1. «Скорость передачи данных и скорость модуляции».** Определить скорость передачи данных, если реализуются одновременно три вида модуляции (амплитудная, частотная и фазовая) со скоростью 19200 бод.



### Определение скорости передачи данных

1 Если одновременно реализуются три вида модуляции, то при каждом изменении аналогового сигнала происходит выбор одного из восьми состояний и, следовательно, каждое такое изменение несет:

$$I = \log_2 8 = 3 \text{ бита.}$$

2 Скорость передачи данных составит:

$$3 \text{ бита} \cdot 19200 \text{ с}^{-1} = 57\,600 \text{ бит/с.}$$

**Стандарты скорости передачи данных (протоколы модемной связи).** Скорость передачи данных определяется используемой скоростью и методом модуляции, которые стандартизированы *протоколами модемной связи* для того, чтобы модемы могли «понимать» друг друга. Первый «древний» модемный протокол давал возможность модемам обмениваться информацией со скоростью всего лишь 300 бит/с. Затем, по мере совершенствования модемной связи, различными фирмами разрабатывались различные более высокоскоростные «фирменные» протоколы, которые обычно затем стандартизировались.

В 1994 году был принят стандартный протокол V.34, который обеспечивает скорость передачи данных до 28 800 бит/с, в 1996 году он был усовершенствован, получил название V.34+ и стал обеспечивать скорость передачи до 33 600 бит/с.

Современные модемы могут обеспечить скорость передачи до 57 600 бит/с, которая достигается при использовании «фирменных» протоколов x2 или K56Flex и стандартного протокола V.90. Обычно современные модемы поддерживают также несколько принятых ранее протоколов, что позволяет сохранить возможность обмена информацией с ранее выпущенными модемами.

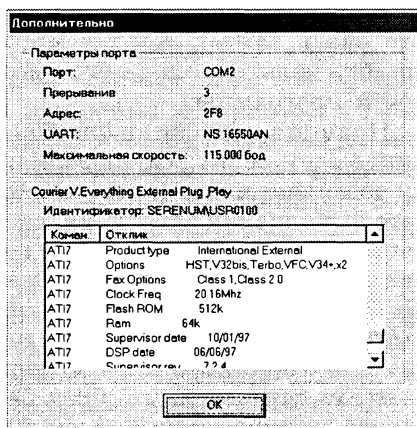
Для того чтобы определить протокол, по которому будут обмениваться информацией два модема, сначала между ними производится обмен специальными сигналами, с помощью которых модемы сообщают друг другу перечень поддерживаемых протоколов. Процесс установления соединения (по-английски *handshaking*, то есть рукопожатие) сопровождается характерным свистом различной частоты, который можно слышать из динамика модема. В результате модемы выбирают для обмена информацией наиболее высокоскоростной протокол из имеющихся у обоих модемов.

**Практическое задание 7.2. «Определение протоколов модемной связи».** Протестировать модем и определить протоколы модемной связи, которые он поддерживает.



### Определение протоколов модемной связи

- 1 Ввести команду [Настройка-Панель управления-Модемы].
- 2 На появившейся диалоговой панели *Свойства: Модемы* активизировать вкладку *Диагностика*.
- 3 Выбрать последовательный порт, к которому подключен модем, и щелкнуть по кнопке *Дополнительно*.
- 4 На появившейся панели *Дополнительно...* в окне с помощью полосы прокрутки найти строку *ATI7 Options*, в которой указаны протоколы связи, которые поддерживает данный модем.



Установленный модем поддерживает следующие протоколы: «фирменные» HST, Terbo, VFC (21,6 Кбит/с), x2 (56 Кбит/с) и стандартные V32bis (14,4 Кбит/с), V34+ (33,6 Кбит/с).



**Задание 7.3. «Время передачи звукового файла».** Модемное соединение с Интернетом установлено по протоколу V.90. Оценить величину минимального времени, которое потребуется для передачи звукового файла в формате WAV длительностью 10 секунд, в процессе записи которого были установлены следующие параметры двоичного кодирования: глубина кодирования — 16 бит, частота дискретизации — 48 кГц, стерео.



### Время передачи звукового файла

- 1 Для определения информационного объема звукового файла необходимо количество бит на одну выборку умножить на количество выборок в секунду, умножить на длительность файла и умножить на два (режим стерео):

$$16 \text{ бит} \cdot 48000 \text{ Гц} \cdot 10 \text{ с} \cdot 2 = 15\,360\,000 \text{ бит.}$$

- 2 Минимальное время передачи будет достигаться на максимально возможной для этого соединения скорости. Для определения времени передачи необходимо информационный объем звукового файла разделить на максимальную скорость передачи информации по каналу:

$$15\,360\,000 \text{ бит} : 57\,600 \text{ бит/с} \sim 267 \text{ с.}$$


**Сжатие данных и коррекция ошибок.** В целях сокращения времени передачи данных по линиям связи необходимо осуществлять *сжатие данных*. Модем сжимает данные, получаемые из последовательного порта, к которому подключен компьютер, и поэтому скорость передачи данных из порта должна быть в 2–3 раза выше, чем максимальная скорость модема.

В процессе передачи данных реализуется *коррекция ошибок*. Это очень важно, так как при передаче программного файла ошибка только в одном бите может привести к неработоспособности программы.

Принцип коррекции ошибок состоит в следующем: передающий модем делит всю информацию на блоки и к каждому блоку добавляет число (контрольную сумму), получаемое в результате выполнения алгоритма, состоящего из последовательности арифметических операций над всеми битами блока. Контрольная сумма меняется при изменении хотя бы одного бита в блоке. Принимающий модем знает алгоритм вычисления контрольной суммы и производит ее вычисление для блока. Если значения принятой и вычисленной контрольной суммы совпадают, то считается, что блок передан правильно. В противном случае, принимающий модем выдает запрос на повторную пересылку блока.

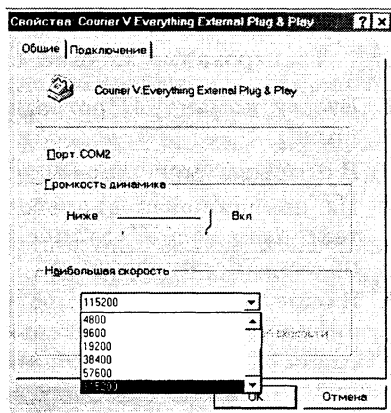
В современных модемах сжатие данных и коррекция ошибок производится в соответствии с протоколом V.42bis.

**Практическое задание 7.4. «Установка скорости последовательного порта».** Установить скорость передачи информации из последовательного порта компьютера, к которому подключен модем, в 2 раза выше, чем скорость модема.

 **Установка оптимальной скорости последовательного порта компьютера, к которому подключен модем**

- 1 Ввести команду [Настройка-Панель управления-Модемы].
- 2 На появившейся диалоговой панели *Свойства: Модемы* щелкнуть по кнопке *Свойства*.
- 3 На появившейся диалоговой панели *Свойства: Courier ...* активизировать вкладку *Общие*.

С помощью раскрывающегося списка *Наибольшая скорость* установить скорость порта примерно в два раза превышающую максимальную скорость передачи вашего модема.



Установленный модем поддерживает максимальную скорость 56 Кбит/с и поэтому скорость порта установлена 115200 бод.

**AT-команды модема.** Модем может находиться в одном из двух режимов работы: передачи данных или AT-команд. AT-команды используются для настройки и управления работой модема. AT-команды представляют собой последовательности символов, начинающиеся с латинских букв AT. Перечень AT-команд у различных модемов практически одинаков, а их синтаксис и назначение можно посмотреть в руководстве пользователя модема.

Для управления модемом с помощью AT-команд используются терминальные программы (в Windows — программа *Hyper Terminal*). Команды, введенные с клавиатуры, и отклики на них модема отображаются в окне терминальной программы. (Если отображения не происходит, необходимо ввести команду AT=1, которая включает режим отображения.)

Если будет введена «пустая» команда AT и нажата клавиша {Enter}, правильно подключенный модем должен дать отклик ОК.

Например, если мы хотим протестировать модем, то необходимо ввести последовательность команд

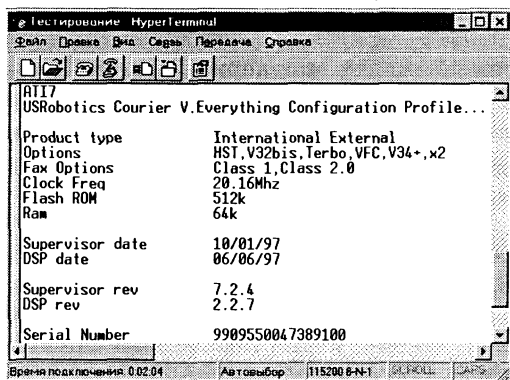
ATI1ATI2ATI3ATI4ATI5ATI6ATI7

(последовательность команд можно записать в сокращенной форме ATI1I2I3I4I5I6I7). Результат будет аналогичен тестированию модема с использованием графического интерфейса Windows, которое описано в практическом задании 7.2.

**Практическое задание 7.6. Тестирование модема с использованием AT-команд.** Произвести тестирование модема с использованием стандартной программы Windows HyperTerminal с использованием AT-команд.

### Тестирование модема с использованием AT-команд

- 1 Ввести команду [Программы-Стандартные-Связь -HyperTerminal].
- 2 В открывшемся окне запустить программу Hypertrm.exe.
- 3 На появившейся диалоговой панели *Новое подключение* в текстовом поле *Название* ввести название подключения (например, *Тестирование*).
- 4 В окне терминальной программы *Тестирование-HyperTerminal* набрать последовательность AT-команд: ATI1I2I3I4I5I6I7 и нажать клавишу *Enter*.



С помощью полос прокрутки просмотреть результат выполнения последовательности AT-команд.

**Телефонный звонок с помощью модема.** С помощью терминальной программы можно заставить модем «снять трубку» и позвонить по заданному телефонному номеру (для этого используется команда ATD). После буквы «D» в команде должен стоять символ, определяющий метод набора номера: «Т» — тоновый набор или «Р» — импульсный (в российских телефонных

сетях используется импульсный набор). Например, звонок в Москве для получения информации о точном времени может быть выполнен с помощью команды ATDP100.

**Практическое задание 7.7. «АТ-команды».** Произвести звонок по заданному телефонному номеру с помощью модема.



### Телефонный звонок с помощью АТ-команд

- 1 Ввести команду [Программы-Стандартные-Связь-HyperTerminal].
- 2 Если внутренняя АТС требует набора цифры 9 и паузы перед набором городского номера, то ввести команду ATDP9,100 (запятая означает паузу в 2 секунды).
- 3 Если вы звоните из другого города, до после выхода на межгород (набора цифры 8), необходимо дать команду ожидания гудка в линии (W) и набрать код города и номер телефона. В этом случае команда примет следующий вид: ATDP9,8W095100

Если номер свободен, то произойдет соединение, и вы услышите точное время (помните, что междугородний звонок платный).

**Регистры и энергонезависимая память модема.** Параметры настройки модема хранятся в его регистрах, которые можно рассматривать как ячейки его оперативной памяти. Модем имеет как минимум 28 регистров (в современных модемах более 60), а объем каждого из регистров составляет 1 байт (в каждом регистре может храниться число от 0 до 256).

Регистры обозначаются буквами S и нумеруются (S0, S1, S2 и т. д.). Информацию о функциональном назначении регистров можно найти в руководстве пользователя модема. Так, в регистре S0 хранится число, определяющее количество гудков, через которое модем снимет трубку; в регистре S6 - число, определяющее время ожидания в секундах, прежде чем модем начинает набор номера; регистры S9 и S10 определяют временные параметры (в десятых долях секунды) захвата модемом несущей в линии и т. д.

Просмотреть содержимое регистров можно с помощью команды ATI4, а изменить их значение с помощью команд типа ATS0=2 (при изменении значений регистров нужно соблюдать осторожность, т. к. после ввода ошибочных значений модем может потерять работоспособность).

Энергонезависимая память постоянно хранит установки настройки модема, которые не стираются при выключении питания. При включении модема они переписываются в оперативную память (регистры). В процессе управления настройками

модема можно осуществить считывание значений регистров из энергонезависимой памяти с помощью команды ATZ.

**Практическое задание 7.8. «Изменение настроек модема».** Произвести изменение настроек модема с целью повышения надежности соединения. Увеличить значения регистров S9 и S10, которые по умолчанию равны S9=6 и S10=7.

### Изменение настроек модема

- 1 Ввести команду [Программы-Стандартные-Связь-НурерTerminal].
- 2 Ввести команду AT S9=60 S10=120
- 3 Если качество соединения ухудшилось, вернуться к значениям по умолчанию и считать их из энергонезависимой памяти. Ввести команду ATZ.

## 7.1.2. Тестирование соединения с Интернетом (маршрут и скорость передачи данных)

**Топология Интернета.** Основой глобальной компьютерной сети Интернет являются магистральные высокоскоростные линии связи, по которым передается информация между региональными сетями.

В настоящее время наиболее мощные региональные сети функционируют в Северной Америке, Европе, Японии и Австралии, которые соединены между собой многочисленными оптоволоконными линиями связи с пропускной способностью 20 Гбит/с и выше.

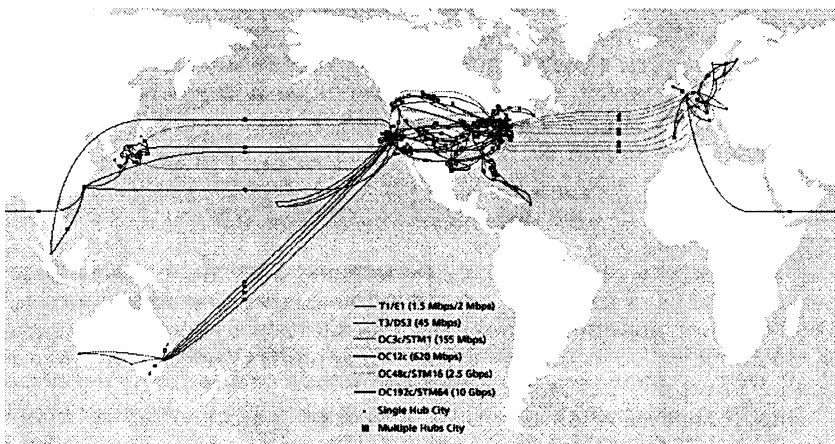
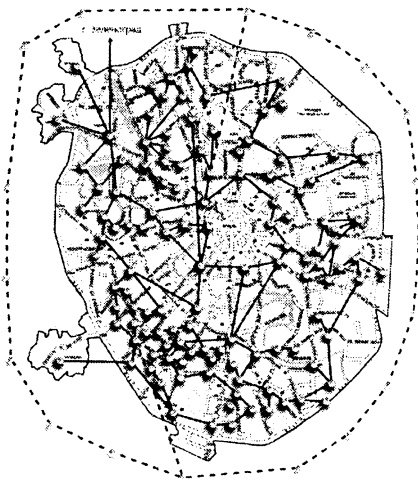
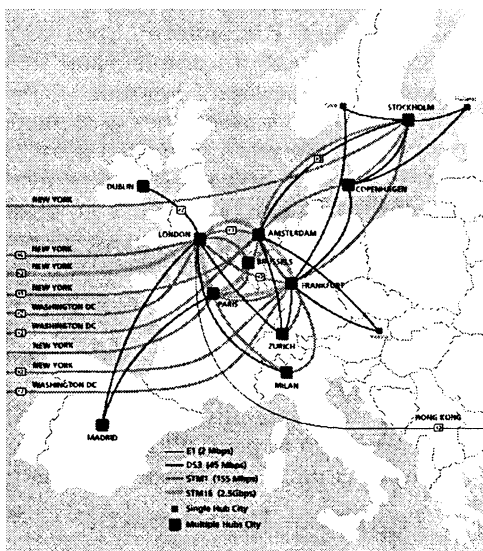


Рис. 7.1. Глобальная сеть Интернет

Внутри региональных сетей информация передается также преимущественно по оптоволоконным каналам с различной пропускной способностью (от 1 до 155 Мбит/с). В региональных сетях часто используются также выделенные линии (медные), а иногда (в пределах прямой видимости) и радиоканалы, пропускная способность которых может достигать 2 Мбит/с.

Для подключения отдаленных регионов наиболее экономически выгодным является подключение по спутниковым каналам, пропускная способность которых может достигать десятков мегабит в секунду.

**Рис. 7.2**  
Региональные сети  
Европы и Москвы



**Маршрут и скорость передачи данных.** После установления модемного соединения с сервером провайдера появляется информационная панель, на которой содержится информация о величине скорости соединения с Интернетом (в зависимости от выбранного модемами протокола связи она может быть различной, например, 57600 бит/с). Однако это число характеризует лишь максимально возможную скорость передачи данных между двумя компьютерами, реальная скорость может быть меньше.

Реальная скорость обмена данными между двумя компьютерами определяется их «удаленностью» друг от друга в Интернете и пропускной способностью линий связи. Удаленность компьютеров в Интернете определяется не величиной расстояния между ними, а количеством промежуточных серверов, через которые проходят данные.

Определить «удаленность» сервера Интернета от локального компьютера можно с помощью программы `tracert.exe`, которая входит в состав Windows. Для этого в режиме MS-DOS необходимо ввести команду `tracert` и доменное имя или IP-адрес сервера Интернета, в результате появится трассировка маршрута к выбранному серверу, содержащая количество серверов (переходов между ними), времена переходов в миллисекундах, а также доменные имена и IP-адреса промежуточных серверов.

**Практическое задание 7.9. «География Интернета-1».** Получить информацию с помощью команды `tracert` о маршруте и скорости прохождения информации от сервера Интернета, находящегося по адресу <http://iit.metodist.ru> (на нем размещен сайт «Информатика и информационные технологии»).



### География Интернета-1

- 1 Перейти в режим MS-DOS с помощью команды [Пуск-Сеанс MS-DOS].
- 2 В окне *Сеанс MS-DOS* ввести команду `tracert` и доменное имя или IP-адрес сервера Интернета.

```

Сеанс MS-DOS
-----
Microsoft(R) Windows 98
(C) Copyright Microsoft Corp 1981-1999.

C:\WIN98>tracert 213.171.37.202

Трассировка маршрута к 213.171.37.202 с максимальным числом переходов 30

  1  113 ms  108 ms  115 ms  BPH04.ntu.ru [212.30.161.24]
  2  106 ms  103 ms  102 ms  AccessPath-B3-1E1-0.ntu.ru [212.30.161.92]
  3  105 ms  102 ms  103 ms  Core14-0-9.ssslan.ntu.ru [195.34.52.66]
  4  104 ms  104 ms  97 ms  MT01-M9-pas118-185M.ntu.ru [212.30.172.106]
  5  105 ms  101 ms  106 ms  E1-M10-Moscow.Rostelecom.ru [195.161.2.121]
  6  105 ms  106 ms  102 ms  aa-qiga10-B-0.Moscow.Rostelecom.ru [195.161.0.5]

  7  105 ms  107 ms  103 ms  Cancor.Moscow.Rostelecom.ru [195.161.2.6]
  8  126 ms  126 ms  124 ms  213.171.37.61
  9  120 ms  122 ms  127 ms  213.171.37.202

Трассировка завершена.

C:\WIN98>
  
```

3 В окне программы просмотреть информацию о параметрах обмена данными с сервером Интернета (количество промежуточных серверов и время перехода от одного сервера к другому).

Трассировку передачи информации от заданного сервера Интернета на локальный компьютер с использованием более удобного интерфейса выполняет специальная программа AnalogX HyperTrace.

**Практическое задание 7.10. «География Интернета-2».** Получить информацию о параметрах соединения с сервером Интернета, находящимся по адресу <http://schools.keldysh.ru> с помощью программы AnalogX HyperTrace.



## География Интернета-2

- 1 Запустить программу AnalogX HyperTrace.
- 2 Ввести доменное имя или IP-адрес сервера Интернета. Щелкнуть по кнопке *Trace*.

The screenshot shows the AnalogX HyperTrace application window. The title bar reads "AnalogX HyperTrace (www.analogx.com)". The status bar indicates "Traceroute completed 9 hops / 120ms TTL...". The main area contains a table with the following data:

Hop	IP Address	Response	Machine Name	Loss
1	212.30.161.22	124ms	APARC2.mtu.ru	----
2	212.30.161.62	124ms	AccessPath-RS-BV12.mtu.ru	----
3	195.34.34.1	125ms	MTUInform-Xchange-1-E501.mtu.ru	----
4	195.34.33.238	120ms	MTU1-MTU2-FastEth10.mtu.ru	----
5	212.30.177.106	130ms	MTU1-M9-pes110.155M.mtu.ru	----
6	193.232.244.34	132ms	MSK-M9-Relain-1.Relain.ru	----
7	194.226.29.133	132ms	MSK-KIAE-Relain-3.Relain.ru	----
8	195.208.22.30	134ms	KIAM-1-Relain.Relain.ru	----
9	194.226.57.46	123ms	schools.keldysh.ru	----

At the bottom of the window, there is an "About" button with the text "schools.keldysh.ru" and a "Trace" button. Other buttons include "Copy", "Config", and "Done".

3 В окне программы просмотреть информацию о параметрах обмена данными с сервером Интернета.

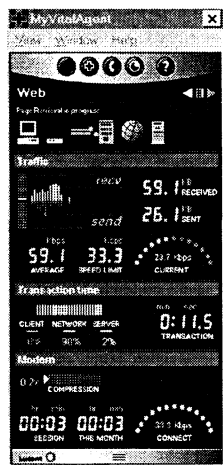
**Параметры текущего соединения.** В процессе работы в Интернете можно отслеживать параметры соединения с различными серверами в фоновом режиме с помощью специальных программ (NetMedic, Modemgrh и др.). Для каждого сервера, с которым производится обмен данными, эти программы дают в графической и числовой форме информацию о количестве переходов (hops) до сервера, максимальной и текущих скоростях передачи и приема данных, средней скорости передачи, величине загрузки процессора локального компьютера и др.

**Практическое задание 7.11. «VitalAgent».** Получить информацию о параметрах текущего соединения с Интернетом с помощью программы VitalAgent.



## Получение информации о параметрах соединения с сервером Интернета с помощью программы VitalAgent

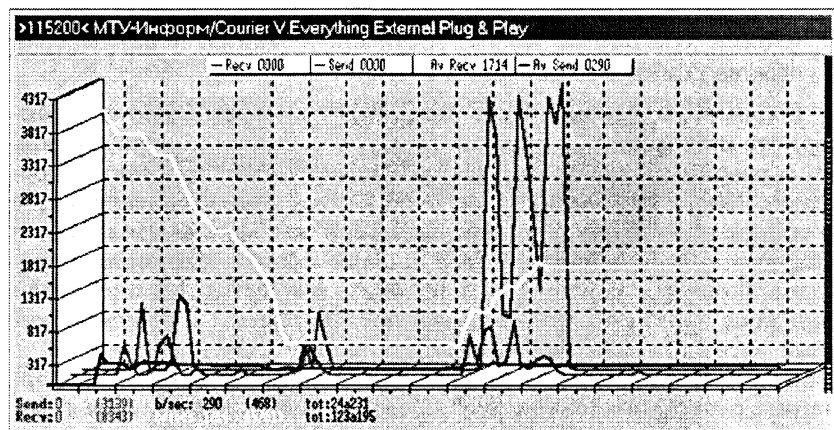
- 1 Запустить программу VitalAgent.  
Раскрыть окно программы, щелкнув по значку программы, появившемуся на панели задач в области индикаторов. Проанализировать информацию о различных параметрах обмена данными с сервером Интернета (скоростях передачи и получения данных, средней и предельной скорости и др.).



**Практическое задание 7.12. «Modemgrh».** Получить информацию о параметрах текущего соединения с Интернетом (количество переданных и принятых байтов за последнюю секунду, а также среднюю скорость приема и передачи за определенный период времени) с помощью программы Modemgrh.

## Получение информации о параметрах соединения с сервером Интернета с помощью программы Modemgrh

- 1 Запустить программу Modemgrh.
- 2 Раскрыть окно программы, щелкнув по значку программы, появившемуся на панели задач в области индикаторов.





Проанализировать графики, показывающие изменения скорости передачи и приема данных в зависимости от времени.

**Интернет-сканеры.** Тестирование скорости соединения с Интернетом можно проводить не только с помощью специальных программ, но и непосредственно в Интернете. Интернет-сканеры (DSL Reports, Microsoft, CNET, PC Pitstop и др.) показывают скорость обмена данными с сервером Интернета, на котором установлен сканер, в сравнении с типичными скоростями различных типов соединений.

**Практическое задание 7.13. «Интернет-сканеры».** Получить информацию о параметрах соединения с помощью Интернет-сканеров.

### Получение информации о параметрах соединения с помощью Интернет-сканеров



- 1 Последовательно установить соединения с Интернет-сканерами DSL Reports, Microsoft, CNET, PC Pitstop.

<b>kbps</b>	65	129	193	257	321	385	449	513	577
<b>k/sec</b>	8	16	24	32	40	48	56	64	72
<b>time</b>	21.3h	10.6h	7.1h	5.3h	4.2h	3.5h	3h	2.6h	2.3h
Some useful comparisons									
telephone quality voice									
<b>DOWN</b> 									
<b>UP</b> 									
33.6k modem download speed									
typical 56k modem download speed									
single channel ISDN									
dual channel ISDN									
IDSL									
150k Video Stream									
Fastest multiplayer game									
192k 'near CD quality' MP3									
300k Video Stream									

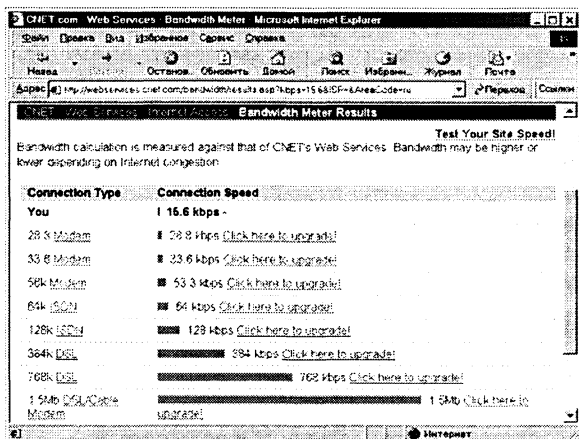
Интернет-сканер DSL Reports <http://www.dslreports.com/stest>

### Download Bandwidth Test Results

Text Block Size	50 KB
Download Time	16860 ms
Transmission Speed	24 Kb/s

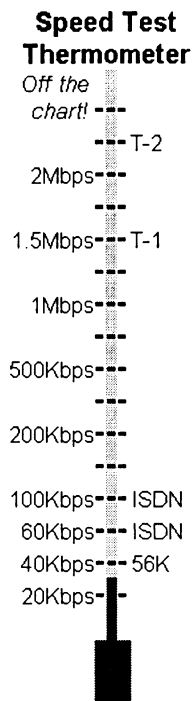
-  Repeat Last Test
-  New Test

Интернет-сканер PC Pitstop <http://www.pcpitstop.com/bandwidth.asp>



Интернет-сканер SNET  
<http://webservices.cnet.com/bandwidth>

Интернет-сканер Microsoft  
<http://computingcentral.msn.com/topics/bandwidth>



## Задания для самостоятельного выполнения

- Получить необходимые сведения и выбрать наиболее эффективного провайдера вашего региона, то есть провайдера, который обеспечивает наиболее качественное подключение к Интернету за минимальную цену.
- Определить скорость передачи данных при скорости модуляции 14400 бод, если происходит частотно-фазовая модуляция.
- Определить скорость передачи данных при скорости модуляции 9600 бод, если происходит амплитудно-частотно-фазовая модуляция.
- Соединение с Интернетом установлено по протоколу V.34+. Оценить величину минимального времени, которое потребуется для передачи графического файла в формате BMP размером 100×100 точек и палитрой из 65 536 цветов.
- Провести тестирование модема с использованием утилиты Модемы из Панели управления Windows. Определить протоколы модемной связи, которые он поддерживает.
- Провести тестирование модема с использованием AT-команд при помощи стандартной программы HyperTerminal.

- 7.7. Провести установку оптимальной скорости последовательного порта, к которому подключен модем.
- 7.8. Провести установку режима сжатия данных и коррекции ошибок для подключенного модема.
- 7.9. Позвонить с помощью терминальной программы по заданному телефонному номеру.
- 7.10. Просмотреть содержимое регистров вашего модема. В случаях частого обрыва модемного соединения увеличить значения регистров S9 и S10.
- 7.11. Провести трассировку соединения с заданным сервером (например, [www.afort.ru](http://www.afort.ru)) с использованием команды `tracert` и программы AnalogX HyperTrace.
- 7.12. Провести исследование параметров соединения с заданным сервером (например, [www.yandex.ru](http://www.yandex.ru)) с использованием программ VitalAgent и Modemgph.
- 7.13. Провести тестирование скорости соединения с помощью Интернет-сканеров DSL Reports, Microsoft, CNET и PC Pitstop.

## 7.2. Работа с электронной почтой

Для работы с электронной почтой используются почтовые программы, которые входят в состав некоторых офисных пакетов (StarOffice) и интегрированных систем для работы в Интернете. В Microsoft Internet Explorer входит почтовая программа Outlook Express, в Netscape Communicator — почтовая программа Messenger, в NeoPlanet — почтовая программа E-Mail Messages и так далее.

### 7.2.1. Настройка почтовой программы Outlook Express

Outlook Express обладает широкими возможностями по созданию, отправке, получению и переадресации почтовых сообщений, позволяет использовать в сообщениях различные кодировки, обеспечивает вложение в сообщение файлов, облегчает адресацию сообщений за счет использования адресной книги и др.

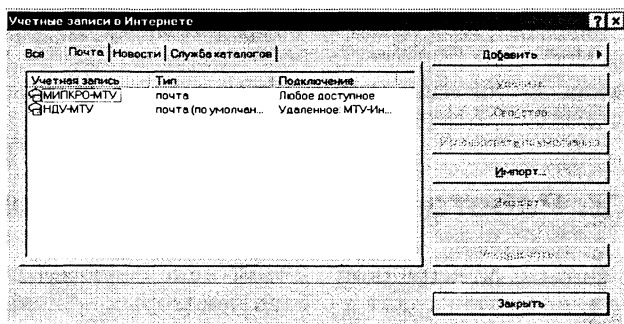
Многие пользователи часто имеют несколько почтовых адресов, и Outlook Express позволяет работать с несколькими почтовыми ящиками. Каждому почтовому ящику должна соответствовать своя *Учетная запись*, которую необходимо настроить. В процессе настройки *Учетной записи* требуется ввести данные, которые пользователь получает при регистрации почтового ящика: имя почтового сервера и почтового ящика, пароль доступа и имена серверов входящей и исходящей почты.

**Практическое задание 7.14. «Настройка учетной записи».** Настроить учетную запись на почтовый ящик, который был зарегистрирован на одном из серверов бесплатной почты (например на [www.aport.ru](http://www.aport.ru)).

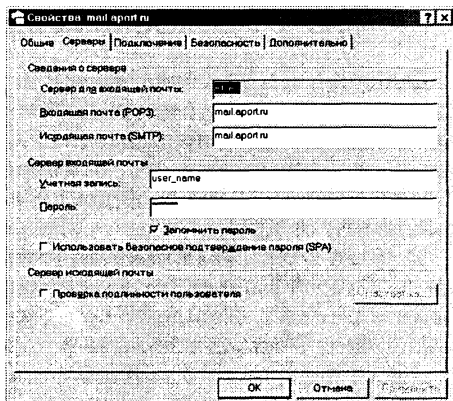


### Настройка учетной записи на новый почтовый ящик

- 1 Запустить Outlook Express. Ввести команду [Сервис-Учетные записи...].
- 2 На появившейся диалоговой панели *Учетные записи в Интернете* выбрать вкладку *Почта* и щелкнуть по кнопке *Добавить*. Из появившегося списка выбрать опцию *Почта*.



- 3 На последовательно появляющихся панелях *Мастера подключения к Интернету* ввести данные, полученные при регистрации почтового ящика на [www.aport.ru](http://www.aport.ru).
- 4 После завершения работы с *Мастером* на диалоговой панели *Учетные записи в Интернете* появится новая учетная запись *mail.aport.ru*.
- 5 Выбрать эту запись и щелкнуть по кнопке *Свойства*. На появившейся диалоговой панели *Свойства: mail.aport.ru* на вкладках *Общие* и *Серверы* проверить правильность и при необходимости отредактировать введенные регистрационные данные.



Почтовая программа Outlook Express может работать с новым почтовым ящиком.

**Настройка правил работы с почтой.** Достаточно часто при работе с почтой приходится с определенными группами однотипных сообщений производить одинаковые действия.

Для своевременного ответа на многочисленные поздравления с праздниками можно предусмотреть отправку автоматического ответа. Для упорядочения сообщений их можно помещать в определенные папки в соответствии с их тематикой, отправителем или получателем и так далее.

Для защиты от спама (надоедливых рекламных сообщений) их целесообразно удалять непосредственно на почтовом сервере, еще до доставки на локальный компьютер.

Автоматическое выполнение таких однотипных действий можно задать с помощью создания *правил для работы с почтой*.

**Практическое задание 7.15. «Защита от спама».** Создать правило работы с почтой, которое будет защищать от получения рекламных сообщений от одного из отправителей (например, FastCash@jackpot.com).



### Создание правила защиты от спама

- 1 Запустить Outlook Express. Ввести команду [Сервис-Правила для сообщений-Почта...].
- 2 На появившейся диалоговой панели *Правила для сообщений* выбрать вкладку *Правила для почты* и щелкнуть по кнопке *Создать*.
- 3 Появившаяся диалоговая панель *Создать правило для почты* содержит четыре окна. В первом окне необходимо выбрать условие применения правила, в данном случае *Искать сообщения, содержащие адресатов в поле "От"*.
- 4 Во втором окне необходимо выбрать действие для данного правила, в данном случае *Удалить с сервера*.

Создать правило для почты

Выборите условия и действия для правила, затем задайте необходимые величины в его описании.

1. Выберите условия для данного правила:

- Искать сообщения, содержащие адресатов в поле "От"
- Искать сообщения, содержащие заданные слова в поле "Тема"
- Искать сообщения, содержащие заданные слова
- Искать сообщения, содержащие адресатов в поле "Кому"
- Искать сообщения, содержащие адресатов в поле "Копия"

2. Выберите действия для данного правила:

- Ответить заданным сообщением
- Прекращение выполнения правил
- Не загружать из сервера
- Удалить с сервера

3. Описание правила (для справки щелкните по подчеркнутой величине)

Применить данное правило при получении сообщения  
Искать сообщения, содержащие FastCash@jackpot.com в поле "От"  
Удалить с сервера

4. Название правила

Правило для почты #1

ОК Отмена

- 5 В третьем окне необходимо уточнить порядок действия данного правила. В условии выполнения ввести почтовый адрес конкретного адресата (в данном случае `FastCash@jackpot.com`).
- 6 Четвертое окно содержит номер созданного правила. После щелчка по кнопке *ОК*, правило вступает в действие.

Сообщения, поступающие от `FastCash@jackpot.com`, будут удаляться уже на почтовом сервере.

## 7.2.2. Работа с Web-почтой

**Регистрация почтового ящика Web-почты.** Существует достаточно много Интернет-серверов (`www.aport.ru`, `www.rambler.ru`, `www.yandex.ru`, `www.mail.ru` и др.), которые предоставляют возможность бесплатно зарегистрировать почтовый ящик и пользоваться им. Важно, что с такой Web-почтой можно работать с использованием Web-интерфейса, то есть непосредственно из браузера.

Для регистрации почтового ящика необходимо загрузить в браузер домашнюю страницу сервера, предоставляющего почтовые услуги, и активизировать ссылку (*Зарегистрироваться*, *Получить адрес* и так далее). После этого будет предъявлено лицензионное соглашение по использованию почтового сервиса, после принятия которого пользователь может выбрать имя своего почтового ящика и пароль доступа к нему (иногда требуется также заполнить анкету).

В некоторых случаях пользователь может выбрать не только имя почтового ящика, но и доменное имя почтового сервера, например, на сервере `www.aport.ru` можно выбрать следующие варианты почтового адреса: `Имя@aport2000.ru`, `Имя@aport.ru`, `Имя@omen.ru`, `Имя@atrus.ru` и `Имя@au.ru`.

**Практическое задание 7.16.** «Регистрация почтового ящика». Зарегистрировать почтовый ящик на сервере `www.aport.ru`.



### Регистрация почтового ящика на сервере `www.aport.ru`

- 1 Загрузить в браузер домашнюю страницу сервера `www.aport.ru`. Активизировать ссылку *Получить адрес*.
- 2 На появившейся странице с соглашением об использовании почтового сервиса щелкнуть по кнопке *Согласен*.
- 3 На появившейся странице *Регистрация* выбрать имя почтового сервера, например `Имя@aport.ru`.
- 4 В текстовое поле *Имя* ввести имя почтового ящика (например, `ndu`). В текстовые поля *Пароль* и *Подтверждение пароля* ввести пароль доступа к нему.

Регистрация - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Назад Вперед Останов... Обновить Домой Поиск Избранн... Журнал Почта

Адрес http://webmail.aort.ru/scripts/signup

**aort2000.ru**

Приветствуем Вас в системе регистрации бесплатного почтового ящика.

Регистрация адреса:  Имя@aort2000.ru  Имя@aort.ru  Имя@open.ru  Имя@at.ru

ИМЯ

ПАРОЛЬ

ПОДТВЕРЖДЕНИЕ ПАРОЛЯ

- Имя может содержать от 2 до 15 символов (латинских букв, цифр или знаков "\_", "-", ".")
- Пароль может содержать от 4 до 20 знаков
- Во избежание различных проблем, связанных с кодировками, рекомендуем Вам не использовать русские буквы в пароле

- Возможно, что выбранные вами имя почтового ящика и пароль уже заняты. Для ускорения процедуры их подбора нажать на кнопку *Проверить занятость имени*.
- Щелкнуть по кнопке *Далее >>*.
- Если уникальные имя и пароль выбраны правильно, появится страница с зарегистрированным именем почтового ящика и анкетой. Поля, помеченные значком #, должны быть заполнены обязательно. Ввод секретного вопроса и ответа на него позволит пользователю получить с сервера пароль в случае его утери.

Регистрация - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Назад Вперед Останов... Обновить Домой Поиск Избранн... Журнал Почта

Адрес http://webmail.aort.ru/scripts/signup

**Поздравляем!**  
Адрес nku@aort.ru создан.

Ваше регистрационное имя: nku (регистрационный адрес nku@aort2000.ru).

Обязательная информация помечена знаком #.

#

или введите свой вопрос

Ответ на секретный вопрос #

Фамилия

Имя

Отчество

E-mail для связи

Адрес домашней страницы

Пол #

Дата рождения #

Регион #

Статус в Интернете



8 После заполнения анкеты нажать кнопку *Сохранить*. Если анкета заполнена правильно, регистрация будет успешно завершена.

**Создание и отправка писем.** Почта с Web-интерфейсом обладает богатыми функциональными возможностями.

Возможно использование адресной книги и отправка почты сразу по нескольким адресам, причем для этого достаточно в соответствующих полях формы отправки письма указывать адреса, разделяя их запятыми (допустимо обычно не больше 5 адресатов в одном письме).

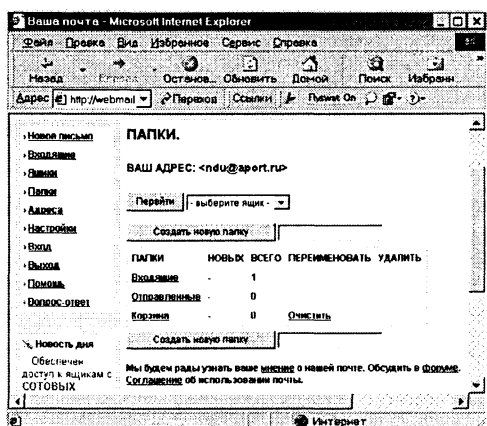
Возможно также вложение в сообщение файлов, необходимо только иметь в виду, что хотя число прикрепляемых к одному письму файлов не ограничено, общий размер письма и прикрепленных файлов не должен превышать 2 Мб.

**Практическое задание 7.17.** «Создание и отправка письма». Создать и отправить письмо с использованием почты с Web-интерфейсом.



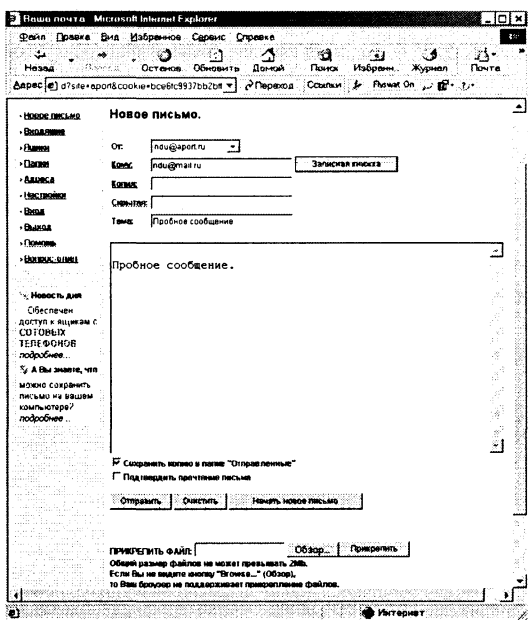
### Создание и отправка письма с использованием почтового ящика на сервере [www.aport.ru](http://www.aport.ru)

- 1 Загрузить в браузер начальную страницу сервера [www.aport.ru](http://www.aport.ru). Ввести в текстовые поля *Имя* и *Пароль* регистрационные данные.
- 2 Загрузится основная страница вашей почты *Папки*, которая имеет ссылки на папки *Входящие* (содержит полученные сообщения), *Отправленные* (содержит отправленные сообщения) и *Корзина* (содержит удаленные сообщения).



- 3 В левой части окна расположены ссылки, позволяющие создать новое письмо, ввести адрес в адресную книгу, провести настройку почты и так далее.

- 4 Для создания нового письма необходимо активизировать ссылку *Новое письмо*.
- 5 В текстовом поле *От:* появится адрес отправителя (ваш адрес), в поле *Кому:* необходимо ввести адрес получателя письма. Если адрес имеется в адресной книге, то можно щелкнуть по кнопке *Записная книжка* и выбрать необходимый адрес.
- 6 В поле *Тема:* ввести тему письма, а в ниже расположенном поле ввести текст сообщения.
- 7 Если к письму необходимо прикрепить файл, в поле *Прикрепить файл:* ввести путь к файлу или воспользоваться для выбора файла кнопкой *Обзор*. После выбора файла щелкнуть по кнопке *Прикрепить*.
- 8 Для отправки письма щелкнуть по кнопке *Отправить*.



**Настройка Web-почты.** Почта с Web-интерфейсом позволяет осуществить настройку работы многих параметров почтового ящика. Такие настройки позволяют сделать процесс создания писем более быстрым и удобным.

Для удобства абонентов, получателей ваших писем, можно ввести имя, которое будет указываться перед почтовым адресом во всех отправляемых вами письмах.

В конце письма обязательно должна быть подпись, в данном случае имеется в виду не ваш автограф, а заключительная фраза письма, зависящая от ваших отношений с абонентом. На-

пример, при официальной переписке будет уместно завершить письмо фразой «С уважением и надеждой на плодотворное сотрудничество, Имя, Отчество, Фамилия, ученое звание, должность», при деловой — «С пожеланием успехов, Имя, Фамилия», при дружественной «Твой Имя» и так далее.

В некоторых случаях при ответе на письмо полезно процитировать его в ответе полностью или частично. Это позволит вашему абоненту легче вспомнить о чем шла речь в его первоначальном послании. Однако цитированием не надо злоупотреблять, так как это может создать психологический дискомфорт в общении.

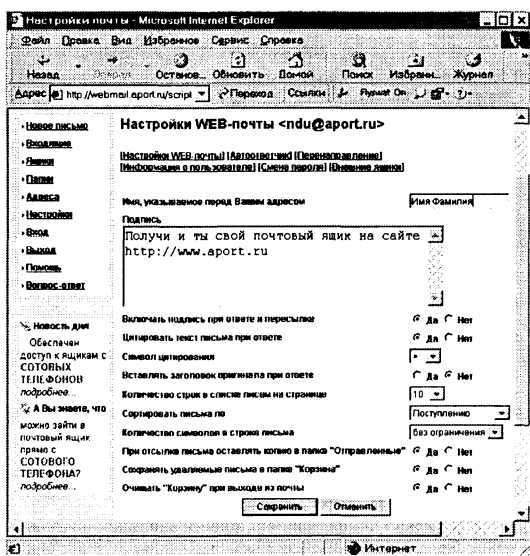
Можно настроить также такие параметры работы почты, как сортировка писем по различным параметрам, режим их удаления и другие.

**Практическое задание 7.18. «Настройка Web-почты».** Настроить правила работы почты с Web-интерфейсом.



## Настройка Web-почты

- 1 Загрузить в браузер начальную страницу сервера [www.aport.ru](http://www.aport.ru). Ввести в текстовые поля *Имя* и *Пароль* регистрационные данные.



- 2 На появившейся странице вашей почты активизировать ссылку *Настройки*. Появится страница *Настройки Web-почты*.

- 3 В текстовое поле *Имя*, указываемое перед Вашим адресом ввести *Имя\_Фамилию*.
- 4 В текстовом поле *Подпись* текст, заданный по умолчанию, заменить на заключительную фразу письма. В переключателе *Включать подпись при ответе и пересылке* выбрать опцию *Да*.
- 5 В переключателе *Цитировать текст письма при ответе* выбрать опцию *Да*. С помощью раскрывающегося списка *Символ цитирования* выбрать символ.
- 6 Установить с помощью переключателей и раскрывающихся списков другие настройки. Щелкнуть по кнопке *Сохранить*.

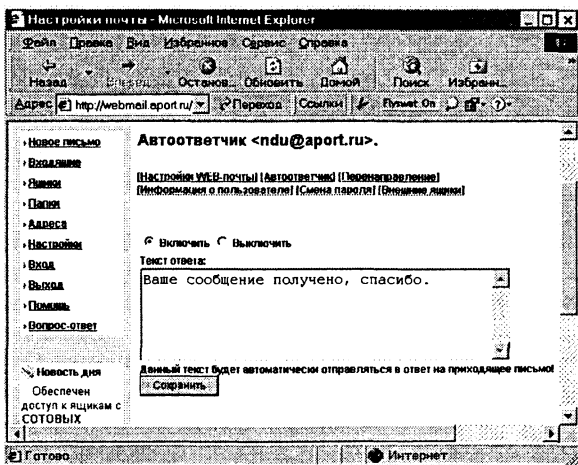
**Автоответчик.** В ряде случаев бывает необходимо задействовать функцию автоматического ответа на приходящие письма (например, в случае временного отсутствия доступа к Интернету). Рассмотрим включение автоответчика на примере Web-почты.

**Практическое задание 7.19. «Настройка Автоответчика».** Настроить *Автоответчик* почты с Web-интерфейсом.



### Настройка Автоответчика

- 1 На странице *Настройки Web-почты* активизировать ссылку *Автоответчик*. Появится страница *Автоответчик*.



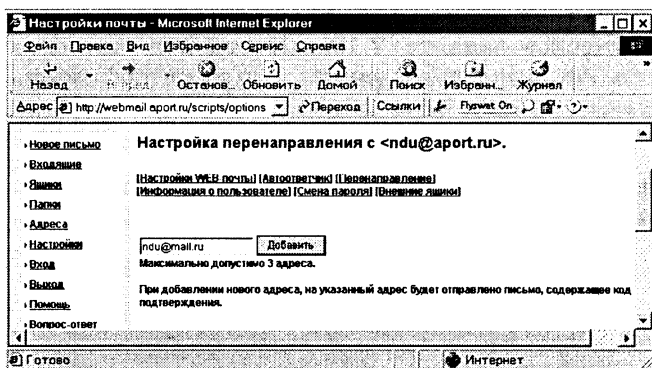
- 2 В переключателе выбрать опцию *Включить*.
- 3 В окне *Текст ответа*: ввести текст ответного сообщения. Щелкнуть по кнопке *Сохранить*.

**Перенаправление сообщений.** В ряде случаев оказывается необходимым перенаправлять почтовые сообщения на другой почтовый ящик. Например, присылаемые сообщения содержат информацию, которая может оказаться полезной вашим коллегам по разработке проекта, друзьям, родственникам и т. д.

**Практическое задание 7.20. «Перенаправление сообщений».** Обеспечить перенаправление входящих сообщений на другой почтовый ящик.

### ■ Перенаправление сообщений

- 1 На странице *Настройки Web-почты* активизировать ссылку *Перенаправление*. Появится страница *Настройка перенаправления*.
- 2 В текстовом окне ввести адреса почтовых ящиков (не более трех), на которые требуется перенаправлять почту. Щелкнуть по кнопке *Добавить*.



Приходящие почтовые сообщения будут перенаправляться по указанным адресам.

**Подключение внешних почтовых ящиков.** Пользователь может иметь несколько почтовых ящиков на различных почтовых серверах. Удобно иметь к ним доступ из одной почтовой системы или программы.

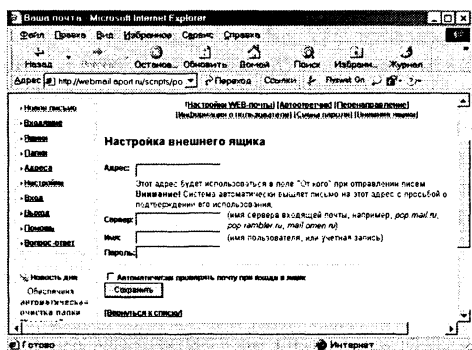
В практическом занятии 6.14 было рассмотрено, как можно настроить почтовую программу Outlook Express на работу с несколькими почтовыми ящиками, теперь рассмотрим как можно это сделать при использовании Web-почты. После подключения внешних почтовых ящиков с ними можно работать так же, как с почтовым ящиком, зарегистрированным на данной почтовой системе.

## Практическое задание 7.21. «Подключение внешних почтовых ящиков».



### Подключение внешних почтовых ящиков

- 1 На странице *Настройки Web-почты* активизировать ссылку *Внешние ящики*. На появившейся странице *Внешние ящики* активизировать ссылку *Подключить*.
- 2 На странице *Настройка внешнего ящика* в поле *Адрес:* ввести почтовый адрес, в поля *Сервер:*, *Имя:*, *Пароль:* ввести данные, полученные при регистрации почтового ящика.



- 3 Щелкнуть по кнопке *Сохранить*.

## Задания для самостоятельного выполнения

- 7.14. Зарегистрировать почтовый ящик на одном из серверов бесплатной почты ([www.aport.ru](http://www.aport.ru), [www.rambler.ru](http://www.rambler.ru), [www.yandex.ru](http://www.yandex.ru), [www.mail.ru](http://www.mail.ru) или др.).
- 7.15. Настроить почтовую программу Outlook Express на получение почты с зарегистрированного почтового ящика.
- 7.16. Настроить почтовую программу Outlook Express на автоматический ответ на письма, содержащие поздравления с праздниками.
- 7.17. Настроить почтовую программу Outlook Express на защиту от спама (рекламных сообщений, приходящих с определенного адреса).
- 7.18. С помощью Web-почты создать и послать сообщение с вложенным файлом.
- 7.19. Настроить Web-почту на автоматический ответ на приходящие сообщения.
- 7.20. Настроить Web-почту на переадресацию сообщения на другой почтовый ящик.
- 7.21. Используя Web-почту, подключить внешний почтовый ящик.
- 7.22. Создать и послать почтовое сообщение с вложенным файлом с помощью почтовых клиентов, входящих в браузеры: Сибкон Коммуникатор (Почта), NeoPlanet (E-Mail Messages) и Opera.

## 7.3. Настройка браузеров

**Браузеры.** Глобальная компьютерная сеть Интернет включает в себя более 140 миллионов серверов, на которых размещены миллиарды Web-страниц. Web-страницы с помощью гиперссылок связаны между собой и образуют Всемирную паутину. Для путешествий по Всемирной паутине используются специальные программы — браузеры. Браузер загружает Web-страницу и отображает ее в соответствии с тэгами языка разметки гипертекста (HTML). Браузеры существуют для всех операционных систем: Microsoft Internet Explorer, NeoPlanet, Opera (Windows), Netscape Communicator (Windows, Linux).

**Загрузка Web-страниц.** Для просмотра в браузере определенной Web-страницы в текстовое поле *Адрес:* необходимо ввести адрес Web-страницы (URL — Universal Resource Locator). URL включает в себя протокол доступа, доменное имя или IP-адрес сервера и путь к Web-странице. Например, URL для начальной страницы сайта «Информатика и информационные технологии» можно записать в следующих формах:

`http://schools.keldysh.ru/info2000/index.htm`

`http://194.226.57.46/info2000/index.htm`

Если начальная страница сайта имеет имя `index.htm`, то в URL ее имя можно не указывать (по умолчанию в браузер из указанного каталога загружается страница с именем `index.htm`).

**Практическое задание 7.22.** «Процесс загрузки Web-страниц». Проанализировать последовательность процесса загрузки начальной страницы сайта «Информатика и информационные технологии» в браузере Internet Explorer, информация о котором отображается в левом нижнем углу браузера.



### Процесс загрузки Web-страницы

1 Сначала осуществляется поиск заданного пользователем доменного имени сервера в иерархической базе данных доменной системы имен (Domain Name System), которая хранится на DNS-серверах:

🔍 Поиск узла: schools.keldysh.ru


2 Если в базе данных DNS имя сервера найдено, начинается поиск самого сервера и ожидание отклика от него:

🔍 Узел найден. Ожидается ответ...

3 Если сервер найден, производится соединение с сервером, при этом отображается его IP-адрес:

🔍 Соединение с узлом: 194.226.57.46

- 4 Производится загрузка Web-страницы и связанных с ней мультимедийных объектов (графических и звуковых файлов). Выводится информация о количестве файлов, оставшихся незагруженными.

 (Осталось: 6) Загрузка рисунка <http://schools.keldysh.ru/info2000/images/back3.jpg...>

Если Web-страница размещена на локальном компьютере, то для ее загрузки в браузер в текстовое поле *Адрес:* необходимо ввести полный путь к файлу Web-страницы. Например, для Web-страницы размещенной на CD-ROM:

E:\practicum\Internet\CP-1251.htm

Загрузить нужную Web-страницу можно также воспользовавшись командой меню браузера [Файл-Открыть...].


**Установка правильной кодировки при просмотре Web-страниц.** Web-страницы представляют собой текстовые файлы с HTML-тэгами, поэтому при просмотре русскоязычных страниц могут возникать проблемы с кодировками кириллицы. Существуют шесть различных кодировок кириллицы CP-866 (MS-DOS), CP-1251 (Windows), KOI8-R (Unix), Mac (Mac OS), ISO и Unicode. Браузеры могут отображать Web-страницы, созданные в любой кодировке.

Если Web-страница содержит в заголовке специальный мета-тэг, информирующий браузер об используемой кодировке, то браузер автоматически настраивается на эту кодировку:

```
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1251">
```

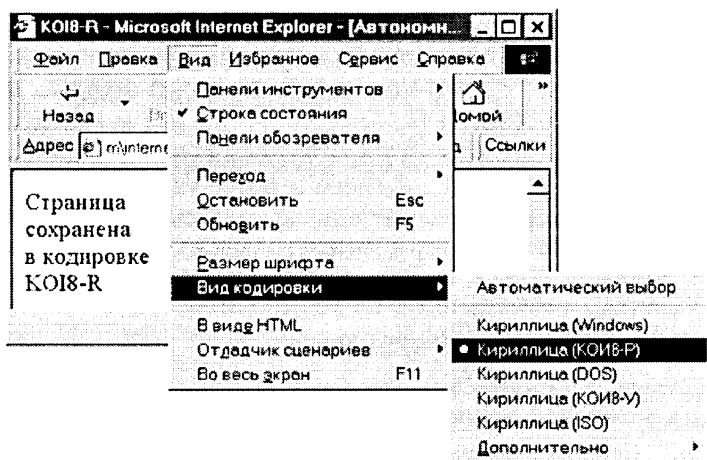
Если такого мета-тэга нет, то необходимую кодировку можно установить с помощью меню браузера.

**Практическое задание 7.23.** «Установка кодировки Web-страниц». На CD-ROM в каталоге \practicum\Internet\ хранятся пять Web-страниц в различных кодировках (CP-1251.html, CP-866.html, KOI8-R.html, ISO.html и Unicode.html). Последовательно открыть в браузере Web-страницы и настроить правильную кодировку.

 **Установка правильной кодировки при просмотре Web-страницы с помощью браузера Internet Explorer**

- 1 В браузере Internet Explorer последовательно открыть перечисленные выше Web-страницы. Выбрать для каждой из них правильную кодировку с помощью команды [Вид-Вид кодировки-Кириллица (...)].






- 2 Если текст страницы читается нормально, оставить выбранную кодировку, в противном случае повторить пункт 1.

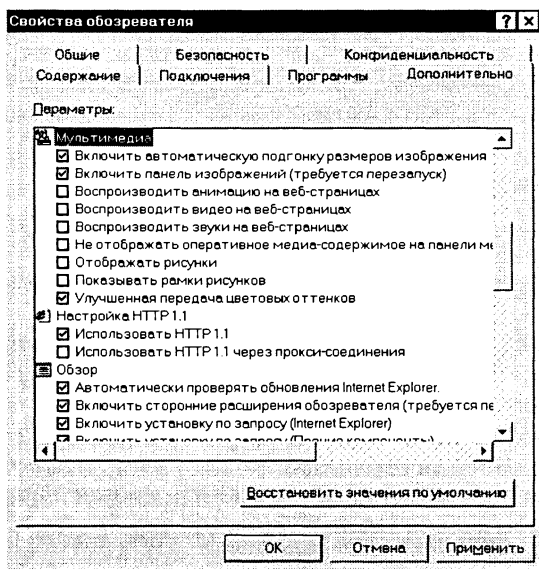
**Ускорение загрузки Web-страниц.** Для ускорения загрузки в браузер Web-страниц можно использовать несколько различных методов. При малой скорости соединения с Интернетом (менее 14 400 бит/с) целесообразно отключить в настройках браузера загрузку мультимедийных объектов Web-страниц (графических, анимационных, видео и звука). В результате Web-страницы будут загружаться значительно быстрее, однако информация на них будет представлена только в текстовом виде.

**Практическое задание 7.24.** «Ускорение загрузки Web-страниц». Ускорить загрузку Web-страниц отключением мультимедийных объектов.

 **Ускорение загрузки Web-страниц в браузере Internet Explorer** путем отключения загрузки мультимедийных объектов

- 1 Ввести команду [Сервис-Свойства обозревателя...]. На появившейся диалоговой панели *Свойства обозревателя* выбрать вкладку *Дополнительно*. В списке свойств группы *Мультимедиа* снять флажки с пунктов, задающих отображение мультимедийных объектов на Web-страницах.

Web-страницы начнут загружаться быстрее, но без графики и звука.



**Хранение копий Web-страниц в кэше локального компьютера.** В процессе просмотра в браузере Web-страниц их копии и связанные с ними мультимедийные файлы сохраняются в кэш-памяти локального компьютера (специальной области памяти, предназначенной для временного хранения файлов). Кэш-память может быть организована как в оперативной памяти локального компьютера, так и на его жестком диске.

Загрузка копий Web-страниц из кэш-памяти локального компьютера существенно ускоряет их просмотр, однако в этом случае может возникнуть ситуация, когда Web-страница в Интернете изменилась, а в браузере мы будем просматривать ее устаревшую копию.

Для загрузки Web-страницы непосредственно из Интернета необходимо в меню браузера щелкнуть по кнопке *Обновить*.

С помощью настроек браузера можно установить различные режимы обновления Web-страницы. При установке опции обновления *Никогда* всегда будет вестись просмотр копий Web-страниц из кэш-памяти. При установке опции *При каждом запуске браузера* будет проводиться проверка обновления страницы при каждом новом запуске окна браузера. Наконец, при установке опции *При каждом просмотре страницы* страница всегда будет загружаться из Интернета.

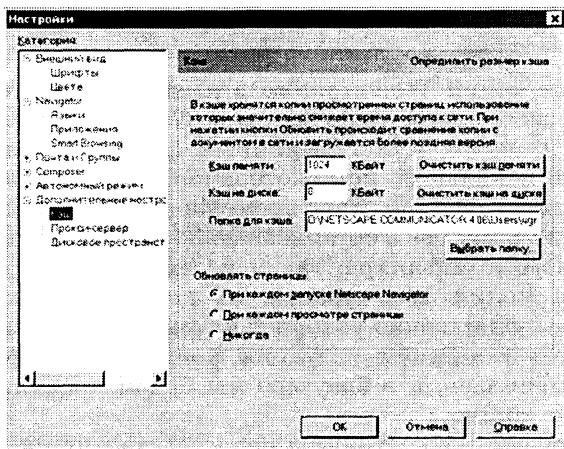
Объем кэша оперативной памяти и кэша жесткого диска можно регулировать, а в случае нехватки памяти или места на жестком диске кэш-память можно очистить.

В случае использования браузера Internet Explorer папка кэш-памяти жесткого диска расположена по адресу C:\Windows\Temporary Internet Files. При использовании других браузеров папка кэш-памяти расположена в ином каталоге.

**Практическое задание 7.25.** «Настройка кэша браузера Сибкон Коммуникатор».

### Настройка кэша браузера Сибкон Коммуникатор

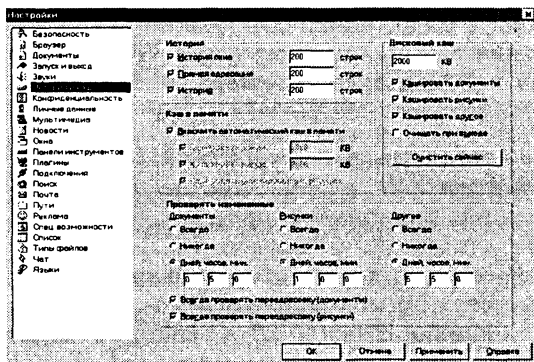
- 1 В браузере Netscape Communicator ввести команду [Правка-Настройки].
- 2 На появившейся диалоговой панели *Настройки* в окне *Категория:* раскрыть пункт *Дополнительные настройки* и выбрать опцию *Кэш*.
- 3 На появившейся вкладке *Кэш* установить требуемые параметры обновления страницы, объема кэша памяти и кэша на диске, а также места нахождения папки для кэша на диске.



**Практическое задание 7.26.** «Настройка кэша браузера Орега».

### Ускорение загрузки Web-страниц в браузере Орега за счет использования кэша

- 1 В браузере Орега ввести команду [Файл-Настройки...].
- 2 На появившейся диалоговой панели *Настройки* в окне выбрать опцию *История и кэш*.
- 3 На появившейся вкладке *Кэш* установить требуемые параметры кэша памяти, дискового кэша и режима обновления Web-страниц и отдельных их элементов.



**Интернет-утилита NetSonic.** Интернет-утилита NetSonic встраивается в браузеры и ускоряет загрузку Web-страниц в процессе путешествий по Всемирной паутине за счет сохранения в кэше программы отдельных элементов (текст, графика и так далее) с каждой посещенной Web-страницы. Когда вы возвращаетесь к уже просмотренной Web-странице, NetSonic загружает из Интернета не страницу целиком, а только ее обновленные элементы, что существенно ускоряет процесс загрузки.

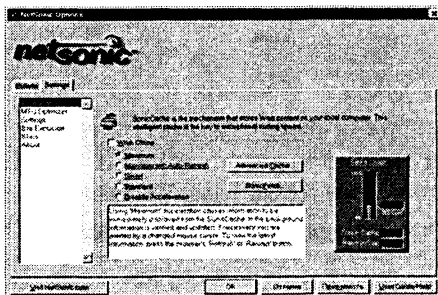
Кэш программы NetSonic размещается в папке icache, расположенной в папке NetSonic, размер кэша, а также другие параметры можно изменять. После установки программы NetSonic она выполняется в фоновом режиме, а ее значок появляется на панели задач в области индикаторов.

**Практическое задание 7.27. «Ускорение загрузки Web-страниц с помощью программы NetSonic».**



**Ускорение загрузки Web-страниц с помощью программы NetSonic**

- 1 Установить с диска Интернет-утилиту NetSonic.
- 2 Щелкнуть по значку программы на панели задач.
- 3 На появившейся диалоговой панели NetSonic установить требуемые параметры: объем кэша, место его нахождения и так далее.



**Сохранение Web-страниц.** В процессе путешествий по Всемирной паутине важные и интересные Web-страницы полезно сохранять на локальном компьютере. Можно выбрать различные варианты сохранения Web-страниц:

- сохранение страницы в формате *HTML* приведет к сохранению самой страницы, но при этом не сохраняются связанные с ней рисунки, звуковые и прочие файлы;
- сохранение страницы в формате *TXT* приведет к сохранению самой страницы в текстовом формате;
- сохранение страницы в формате *Web-страница полностью* (доступно в Internet Explorer 5.0 и выше) приведет к сохранению не только самой страницы, но и связанных с ней рисунков, звуковых и прочих файлов в отдельной папке.

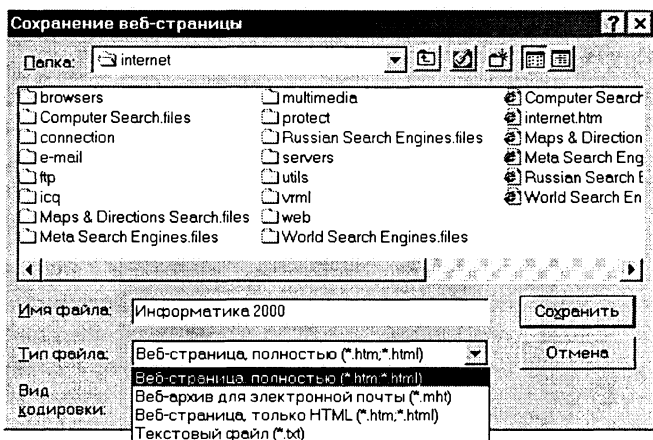
Можно сохранить как Web-страницу полностью, так и отдельную ее часть: текст, изображения или ссылки. Для этого необходимо щелкнуть по выбранному для сохранения объекту правой кнопкой мыши и выбрать в контекстном меню опции *Сохранить объект как...* или *Сохранить рисунок как...* и выбрать затем папку на локальном компьютере, где будет произведено сохранение элемента Web-страницы.

### Практическое задание 7.28. «Сохранение Web-страниц».



#### Сохранение Web-страниц

- 1 Загрузить в браузер Internet Explorer Web-страницу.
- 2 Ввести команду [Файл-Сохранить как...].
- 3 На появившейся диалоговой панели *Сохранение веб-страницы* в поле *Тип файла:* выбрать формат сохранения Web-страницы (например, *Веб-страница полностью (\*.htm, \*.html)*).



- 4 Будет сохранена Web-страница и связанные с ней файлы в отдельной папке.

**Печать Web-страниц.** В процессе просмотра в браузере можно выводить на печать как Web-страницы целиком, так и отдельные их элементы (рисунки, таблицы и т.д.). Для печати страницы целиком необходимо ввести команду [Файл-Печать...] и на появившемся диалоговом окне *Печать* выбрать нужные опции печати. Для печати элемента Web-страницы необходимо щелкнуть по выбранному для печати объекту правой кнопкой мыши и выбрать в контекстном меню опцию *Печать объекта*.

### Задания для самостоятельного выполнения

- 7.23. Загрузить в различных браузерах начальную Web-страницу сайта «Информатика и информационные технологии» из Интернета. Ввести URL страницы с использованием доменного имени (it.metodist.ru) и IP-адреса (213.171.37.202). Проанализировать последовательность загрузки.
- 7.24. Загрузить Web-страницу index.htm с локального компьютера (из корневого каталога CD-ROM). Воспользоваться для этого полем *Адрес:* и командой меню браузера [Файл-Открыть...]
- 7.25. Просмотреть Web-страницы CP-866.html, CP-1251.html, KOI8-R.html, ISO.html, Unicode.html, находящиеся на CD-ROM в каталоге \practicum\Internet в различных браузерах, установив правильную кодировку.
- 7.26. В целях ускорения загрузки Web-страниц в различных браузерах отменить загрузку мультимедиа-объектов.
- 7.27. Установить в различных браузерах параметры кэша, которые максимально ускоряют загрузку Web-страниц.
- 7.28. В различных браузерах произвести установку оптимального размера кэша (для имеющейся конфигурации компьютера).
- 7.29. В различных браузерах произвести очистку кэша памяти и дискового кэша.
- 7.30. Произвести настройку Интернет-утилиты NetSonic. Просмотреть в браузере последовательно несколько Web-страниц, а затем просмотреть эти страницы повторно. Обратит внимание на ускорение повторного процесса загрузки.
- 7.31. Загрузить в браузер Web-страницу и сохранить ее в различных форматах.
- 7.32. Сохранить имеющийся на Web-странице рисунок в отдельном файле.
- 7.33. Загрузить в браузер Web-страницу и распечатать ее целиком.
- 7.34. Загрузить в браузер Web-страницу с рисунком и распечатать рисунок.

## 7.4. Работа с серверами файловых архивов и Web-серверами

### 7.4.1. Загрузка файлов из файловых архивов с помощью браузеров

На десятках тысяч серверов файловых архивов хранятся сотни миллионов свободно распространяемых (freeware) и условно бесплатных (shareware) программ.

Для удобства пользователей некоторые серверы файловых архивов имеют Web-интерфейс, что позволяет работать с ними с использованием браузеров. Для упрощения поиска программы обычно тематически сгруппированы по разделам с помощью иерархического каталога. Возможен также быстрый поиск программ путем ввода их имени в поле поиска.

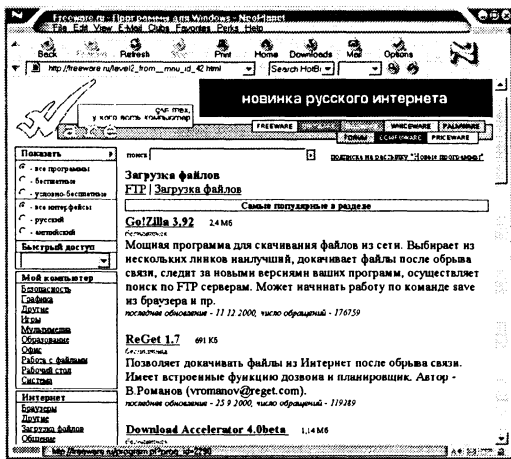
Так как браузеры являются интегрированными системами для работы с различными информационными ресурсами Интернета (Всемирная паутина, файловые архивы, электронная почта и др.), они включают в себя менеджеры загрузки файлов (Download Manager).

В российском Интернете (Рунете) одним из наиболее посещаемых серверов файловых архивов является сервер [www.freeware.ru](http://www.freeware.ru). Сервер содержит несколько тысяч бесплатных и условно-бесплатных программ для различных компьютерных платформ (Windows, Unix, Mac и др.).

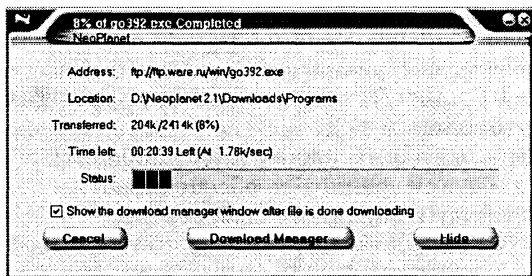
**Практическое задание 7.29. «Загрузка файлов с использованием браузера».** Произвести загрузку файла с использованием возможностей браузера NeoPlanet с сервера [www.freeware.ru](http://www.freeware.ru). Для загрузки можно выбрать файл мощного специализированного менеджера загрузки файлов Go!Zilla.

#### **Загрузка файла с сервера файлового архива с помощью браузера NeoPlanet**

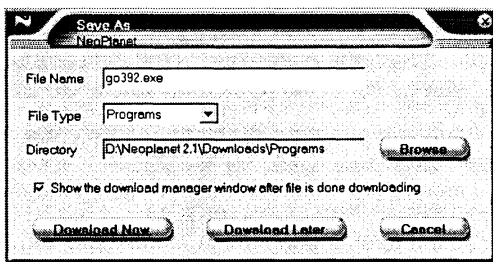
- 1 Загрузить в браузер NeoPlanet начальную страницу сервера Freeware.ru, находящуюся по адресу: <http://www.freeware.ru>.
- 2 Выбрать тематический каталог, например, Интернет\Загрузка файлов. В окне браузера появится перечень программ с краткой аннотацией их размера, статуса и назначения.
- 3 Активизировать ссылку на требуемый файл (в данном случае Go!Zilla 4.1). Появится окно с более подробной информацией о программе и ссылками на домашнюю страницу программы и файловый архив. Активизировать ссылку на файловый архив [Скачать с freeware.ru](http://www.freeware.ru).



- 4 На появившемся диалоговом окне менеджера загрузки файлов (Download Manager) выбрать папку локального компьютера, в которую будет загружаться файл `gozilla.exe`. Щелкнуть по кнопке *Download Now*.

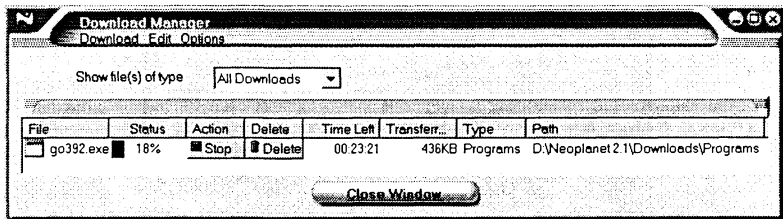


- 5 Появится информационное окно, показывающее откуда и куда загружается файл, объем загруженной части файла и его общий объем, оставшееся время загрузки и скорость загрузки, а также графический индикатор процесса загрузки. Щелчком по кнопке *Download Manager* можно вызвать окно менеджера загрузки файлов.





- 6 Окно менеджера загрузки файлов предоставляет полную информацию о процессе загрузки, а также позволяет изменять параметры загрузки с помощью меню *Options*.



### 7.4.2. Загрузка файлов из файловых архивов с помощью специализированных менеджеров загрузки файлов (Download Manager)


Специализированные менеджеры загрузки файлов (Go!Zilla, ReGet и др.) обладают гораздо большими возможностями по сравнению с менеджерами загрузки файлов, интегрированными в браузеры. В частности, они позволяют продолжить загрузку файла после разрыва соединения с сервером или подключения к Интернету, обеспечивают доступ к файлам как по FTP (File Transport Protocol), так и по HTTP (Hyper Text Transport Protocol) протоколам и т. д.

Go!Zilla является одним из наиболее удобных и эффективных менеджеров загрузки файлов и обеспечивает:

- загрузку файла, выбранного в браузере;
- повышенную скорость загрузки за счет использования нескольких соединений с удаленным сервером (одновременно загружаются несколько частей файла);
- поиск выбранного для загрузки файла по серверам файловых архивов и др.

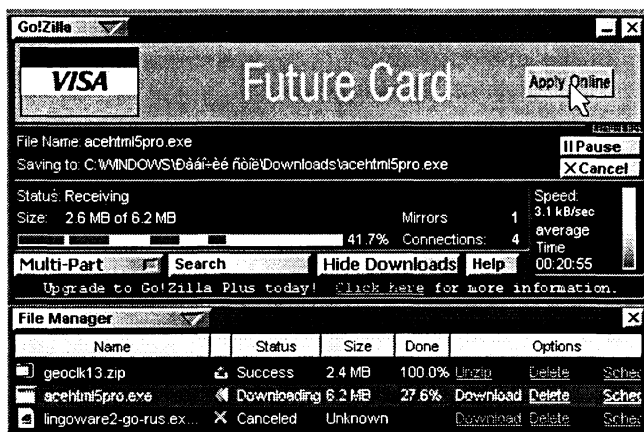
Подробная информация о процессе загрузки файла (текущая и средняя скорость загрузки, процент выполнения загрузки, ориентировочное время загрузки и др.) предоставляется пользователю в числовом и графическом виде.

**Практическое задание 7.30.** «Загрузка файлов с использованием менеджера загрузки файлов». Произвести загрузку файла с помощью менеджера загрузки файлов Go!Zilla.

 **Загрузка файла с сервера файлового архива с помощью менеджера загрузки файлов Go!Zilla**

- 1 В браузере активизировать ссылку на требуемый файл. Начнется процесс загрузки файлов и появится окно менеджера

загрузки файлов Go!Zilla, в котором будет отображаться в числовом и графическом виде информация о процессе загрузки: куда загружается файл, объем загруженной части файла и его общий объем, оставшееся время загрузки, текущая и средняя скорости загрузки, а также графический индикатор процесса загрузки.



- 2 Щелчком по кнопке *Go!Zilla* можно вызвать меню, которое позволяет изменять параметры загрузки.

### 7.4.3. Обмен файлами с FTP-серверами

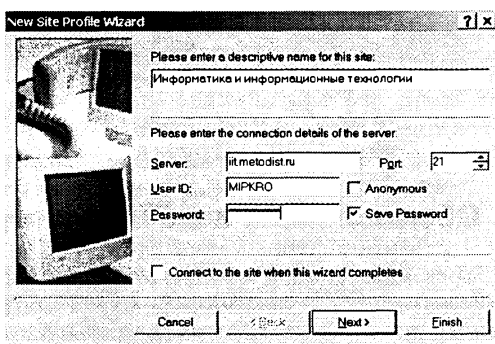
FTP-серверы — это серверы, с которыми может производиться обмен файлами по FTP-протоколу. FTP-серверы по своему функциональному назначению могут являться как серверами файловых архивов, так и Web-серверами, на которых размещаются Web-сайты. С серверов файловых архивов производится загрузка файлов (Download) на локальный компьютер, наоборот, на Web-серверы производится передача файлов (Upload) с локального компьютера в процессе публикации Web-сайтов. Обмен файлами (загрузка и передача) с FTP-серверами производится с помощью специализированных программ — FTP-клиентов (AceFTP, CuteFTP и др.).

Доступ к серверам файловых архивов для загрузки файлов на локальный компьютер обычно является анонимным и не требует ввода имени пользователя (username) и пароля (password). Наоборот, доступ к Web-серверам с целью передачи файлов на удаленный сервер в процессе публикации Web-сайта требует идентификации пользователя, то есть ввода имени пользователя и пароля.

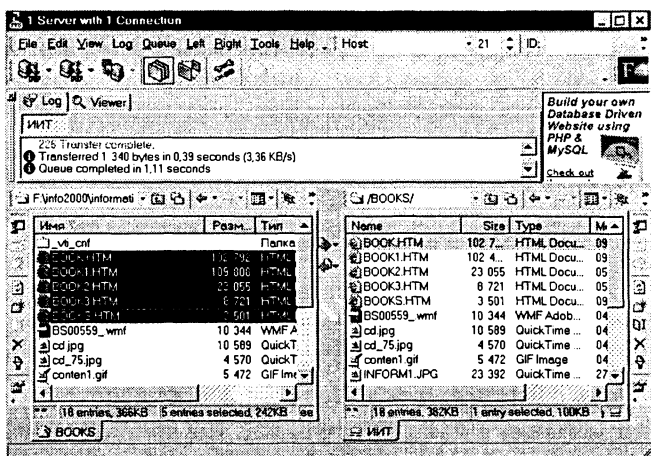
Одним из наиболее удобных свободно распространяемых FTP-клиентов является AceFTP. Включает в себя *Site Manager* (менеджер сайтов), позволяющий создать список серверов, с которыми планируется работа. Представляет в удобном для пользователя виде каталоги локального и удаленного компьютера, поддерживает технологию Drag&Drop, обеспечивает продолжение загрузки файла после обрыва соединения и др. В процессе передачи файла в нижней строке окна приложения отображается необходимая информация: процент переданного объема файла, скорость передачи, оставшееся время и др.

**Практическое задание 7.31. «Передача файлов на Web-сервер с локального компьютера».** Произвести передачу обновленных Web-страниц с помощью FTP-клиента AceFTP на Web-сервер.

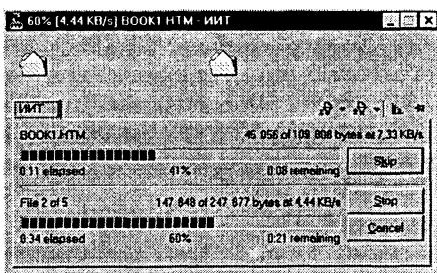
- 1 Запустить FTP-клиент AceFTP.
- 2 Открыть *Менеджер сайтов*, щелкнув по кнопке *Connect*. Появится диалоговое окно *AceFtp Site Profiles-Sites*.



- 3 Создать *Новое соединение*, для этого ввести команду [File>Create-New Site Profile].  
В появившемся диалоговом окне ввести в соответствующие текстовые поля: имя нового соединения, адрес сервера, имя пользователя (UserID) и пароль (Password).
- 4 Установить соединение с сервером, щелкнув в окне *AceFtp Site Profiles-Sites* по имени нужного соединения.
- 5 После установки соединения с сервером в левой панели окна приложения отобразится каталог локального компьютера, а в правой панели — каталог удаленного сервера.  
Для передачи файлов с локального компьютера на сервер необходимо их выбрать и перетащить мышью в правую панель диалогового окна (папку сервера).



- 6 Процесс загрузки каждого файла (скорость передачи, оставшееся время и др.) отображается с помощью информационной панели.



#### 7.4.4. Загрузка Web-страниц и Web-сайтов с помощью off-line браузеров

Далеко не всегда российский пользователь Интернета может позволить себе изучение Web-документов при постоянном подключении к сети Интернет (on-line). Для быстрой загрузки Web-сайтов с целью их дальнейшего неспешного просмотра в автономном режиме используются специальные программы — off-line браузеры.

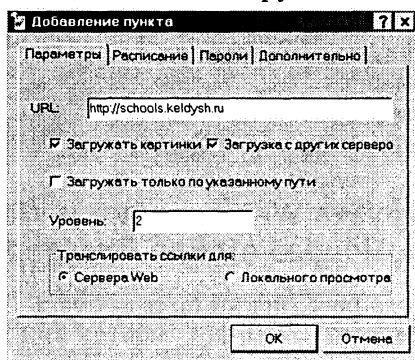
Off-line браузеры (например, Web Downloader) позволяют загружать на локальный компьютер Web-сайты целиком или отдельные части сайта по выбору. Пользователь может установить необходимую «глубину» загрузки Web-сайта (количество вложенных каталогов), загрузку связанных со страницами мультимедиа-файлов, загрузку по гиперссылкам Web-страниц с других Web-серверов и так далее. Имеется возможность про-

должения загрузки сайта после разрыва соединения и обновления ранее загруженных сайтов. Некоторые off-line браузеры (например, WebZip) позволяют одновременно с загрузкой файлов выполнять их архивацию.

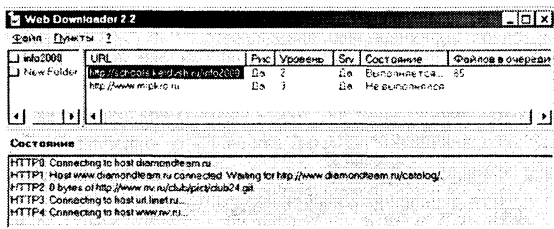
**Практическое задание 7.32. «Загрузка сайта с использованием off-line браузера».** Произвести загрузку Web-сайта (например, «Информатика и информационные технологии») на локальный компьютер с помощью off-line браузера Web Downloader.

### Загрузка Web-сайта на локальный компьютер с помощью off-line браузера Web Downloader

- 1 Запустить off-line браузер Web Downloader.
- 2 Ввести команду [Файл-Настройки]. На появившемся диалоговом окне *Настройки* выбрать вкладку *Загрузка*. В текстовое поле *Каталог*: ввести путь к папке на локальном компьютере, в которую будет загружаться Web-сайт.
- 3 Если соединение с Интернетом производится через прокси-сервер, на вкладке *Прокси сервер* установить его тип.
- 4 Ввести команду [Пункты-Добавить]. На появившемся диалоговом окне *Добавление пункта* на вкладке *Параметры* в текстовое поле *URL*: ввести URL загружаемого Web-сайта.



- 5 Если Web-сайт должен быть загружен целиком (с мультимедиа-файлами и Web-страницами, находящимися на других серверах), установить флажки в пунктах *Загрузить картинки* и *Загрузка с других серверов*.
- 6 В поле *Уровень*: установить количество вложенных каталогов Web-сайта, которые будут загружаться.
- 7 Информация о добавленных пунктах отображается в правой части диалогового окна программы. Из списка пунктов выбрать сайт для загрузки щелчком мыши или нажатием клавиши *Insert*.



- 8 Для начала загрузки выбранного сайта ввести команду [Пункты-Выполнить] или нажать клавишу {F5}.
- 9 В нижней части диалогового окна программы отображается процесс загрузки. В целях ускорения загрузки сайта одновременно производится загрузка пяти различных входящих в него файлов. Количество одновременно загружаемых файлов можно изменять на вкладке *Интернет* диалоговой панели *Настройки*, доступ к которой можно осуществить с помощью команды [Файл-Настройки-Интернет].

### Задания для самостоятельного выполнения

- 7.35. Загрузить из Интернета с помощью менеджеров загрузки файлов Go!Zilla и Reget их обновленные версии.
- 7.36. Найти на серверах файловых архивов и загрузить другие менеджеры загрузки файлов.
- 7.37. Загрузить из Интернета с помощью FTP-клиентов AceFTP и CuteFTP их обновленные версии.
- 7.38. Найти на серверах файловых архивов и загрузить файлы других FTP-клиентов.
- 7.39. Загрузить из Интернета с помощью off-line браузеров Web-сайты.

## 7.5. Поиск информации в Интернете

Для поиска информации в Интернете используются специальные поисковые серверы, которые содержат постоянно обновляемую информацию о Web-сайтах, файлах, e-mail-адресах и других информационных ресурсах Интернета. Различные поисковые серверы могут использовать различные механизмы поиска, хранения и предоставления пользователю информации.

Поисковые серверы Интернета можно разделить на две группы: поисковые системы общего назначения и специализированные поисковые системы.

**Поисковые системы общего назначения** позволяют находить документы во Всемирной паутине *по ключевым словам*, которые вводятся в поле поиска, и/или с помощью иерархической системы *каталогов*.

**Поисковые машины.** Автоматическое заполнение баз данных поисковых систем производится роботизированными системами (Search Engines). Специальные программы-роботы, которые называют пауками (spiders), постоянно обходят Интернет в поисках новой информации, которую они вносят в базу данных. База данных содержит URL-адреса и проиндексированную информацию, связанную с этими адресами.

Мощная поисковая машина обходит весь Интернет за несколько дней. При этом составляется новый индекс — опись доступных ресурсов. При каждом новом цикле обхода индекс обновляется, и старые недействительные адреса удаляются.

При поиске по ключевым словам в поле поиска вводится одно или несколько ключевых слов, которые, по мнению пользователя, являются главными для искомого документа. Можно также использовать сложные запросы, использующие логические операции, шаблоны и т. д.

**Каталоги.** Каталоги составляются редакторами, просматривающими каждый новый сайт до его включения в индекс. В процессе наполнения каталогов могут участвовать и сами разработчики сайтов при их регистрации в поисковой системе. Каталоги обычно организованы в соответствии с предметной классификацией.

Качество систематизации сайтов при этом выше, чем у поисковых машин, но люди не могут успеть за темпами расширения и изменения Интернета: индекс популярного каталога Yahoo! содержит лишь немногим более миллиона записей.

Интерфейс каталогов содержит *список разделов*, в которые сгруппированы ссылки на URL-адреса наиболее важных документов. Каждая ссылка обычно аннотирована, т. е. содержит краткое описание содержания документа.

Наиболее популярными поисковыми системами общего назначения в русскоязычном Интернете (Рунете) являются Яндекс ([www.yandex.ru](http://www.yandex.ru)), Rambler ([www.rambler.ru](http://www.rambler.ru)) и Апорт ([www.aport.ru](http://www.aport.ru)), а в англоязычном — Yahoo! ([www.yahoo.com](http://www.yahoo.com)).

Важными показателями качества поискового сервера являются:

- объем базы данных (количество документов);
- скорость обхода Интернета программным роботом (с этим связана скорость обновления информации в базе данных);
- алгоритм индексации (только по ключевым словам Web-страницы или по всему тексту);
- возможности расширенного поиска с использованием языка запросов;
- структура системы каталогов.

Рис. 7.3  
Поисковые серверы  
Яндекс, Rambler и Апорт

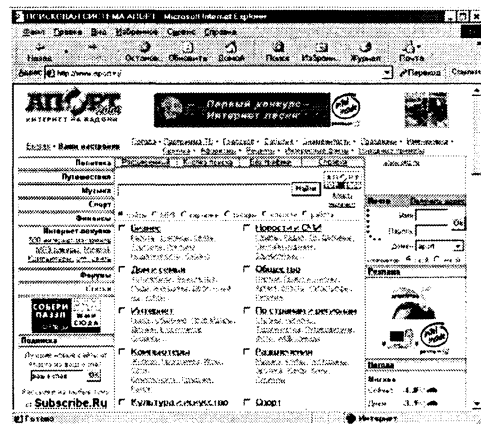
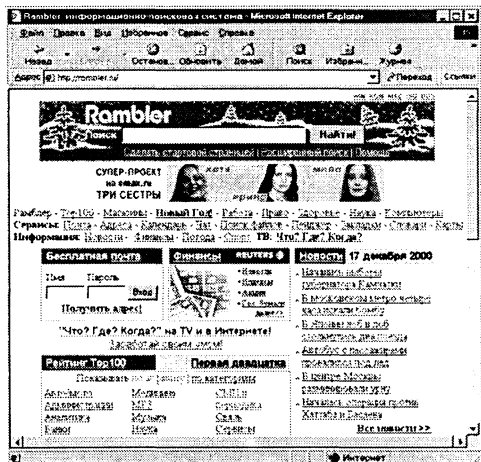




Таблица 7.1

### Сравнительные характеристики поисковых серверов Рунета

	Яндекс	Rambler	Апорт
Объем базы на начало 2001 года	Более 31 миллиона документов	Более 12 миллионов документов	Более 14 миллионов документов
Наличие сервисов	Система объединяет поисковую машину и каталог		
Каталоги	10 основных разделов и 7 комбинированных	56 разделов	14 разделов верхнего уровня, каталог ресурсов по регионам
Язык запросов	Логическое И		
	пробел, &	AND, &, пробел	И, AND, пробел
	Логическое ИЛИ		
		OR,	ИЛИ, OR,
Язык документа	выбор: любой, кириллица, латиница	выбор: любой, русский, английский	выбор: русский, английский

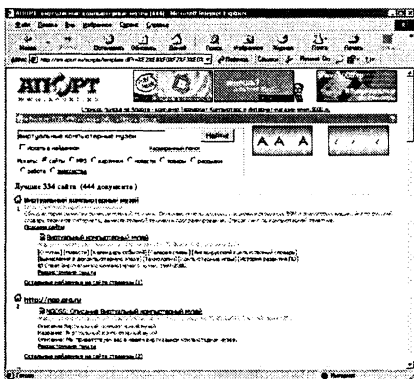
**Практическое задание 7.33. «Виртуальные компьютерные музеи».** Найти в Интернете сайты, посвященные истории развития вычислительной техники и компьютеров.



#### Поиск русскоязычных сайтов с использованием системы Апорт

- 1 Ввести в поле поиска системы Апорт ключевые слова «виртуальные компьютерные музеи».
- 2 Пользователю будет представлен список со ссылками на 334 сайта, содержащие виртуальные компьютерные музеи. Каждая ссылка включает также краткую аннотацию содержания сайта.

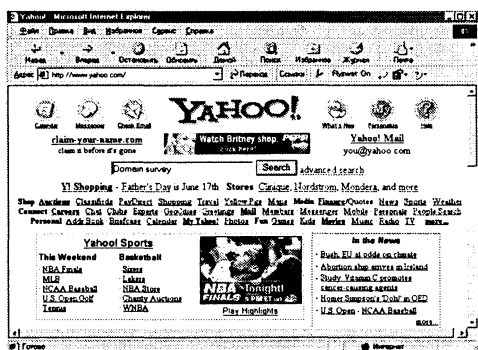
Активизация любой из ссылок позволяет посетить один из виртуальных компьютерных музеев.



**Практическое задание 7.34. «Статистика Интернета».** Найти в Интернете сайты, содержащие статистические данные о развитии Интернета.

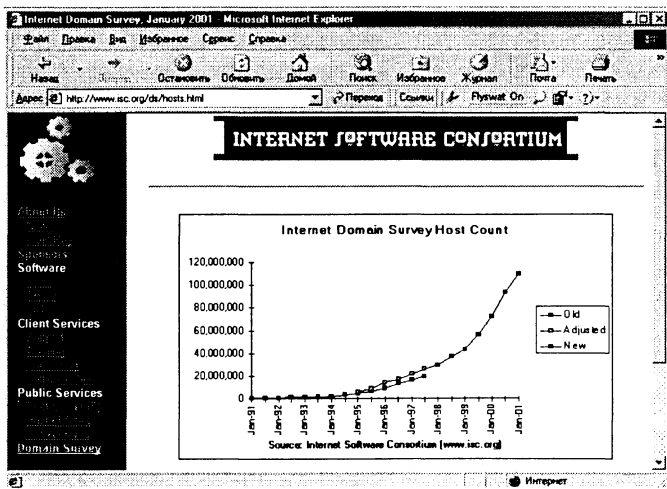
### Поиск англоязычных сайтов с использованием системы Yahoo!

- 1 Ввести в поле поиска системы Yahoo! ключевые слова «domain survey».
- 2 Пользователю будет представлен список со ссылками на 7 сайтов, содержащих статистическую информацию об Интернете.



После активизация первой ссылки *Internet Domain Survey* произойдет переход на сервер Internet Software Consortium ([www.isc.com](http://www.isc.com)).

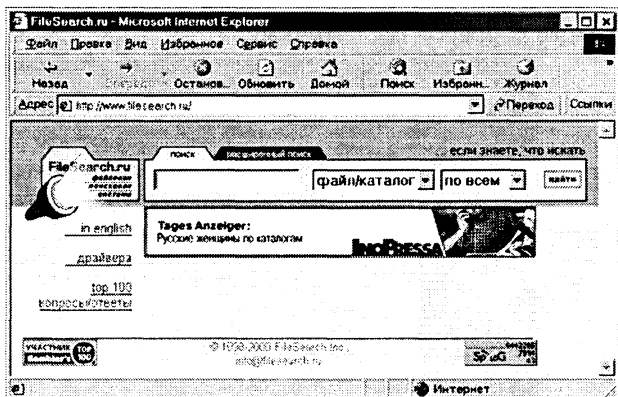
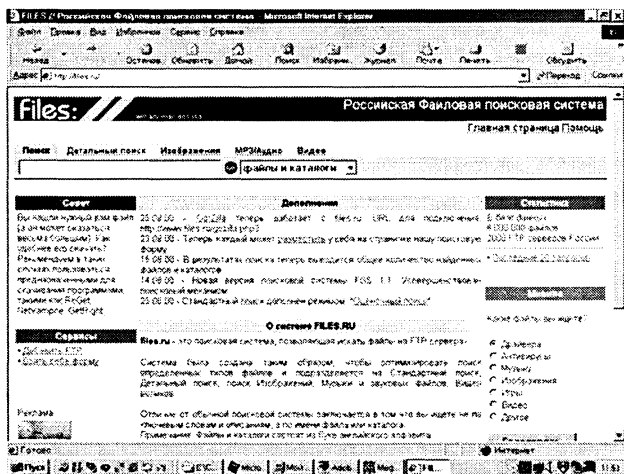
На нем можно ознакомиться с данными о росте количества серверов Интернета по годам и количественном распределении серверов по доменам.



**Специализированные поисковые системы** позволяют искать информацию в других информационных «слоях» Интернета. Существуют специализированные поисковые системы, предназначенные для поиска файлов на серверах файловых архивов, а также системы поиска адресов электронной почты.

Российская файловая поисковая система ([www.files.ru](http://www.files.ru)) и Файловая поисковая система ([www.filesearch.ru](http://www.filesearch.ru)) являются специализированными поисковыми системами, в базах данных которых хранятся сведения о 6 миллионах файлов, размещенных на 2 тысячах серверов файловых архивов Рунета.

Рис. 7.4  
Системы поиска файлов

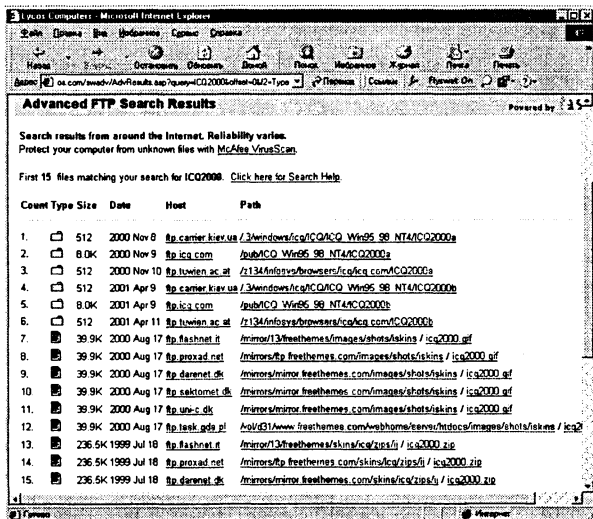


**Практическое задание 7.35. «Поиск файла».** Найти файл программы интерактивного общения ICQ.



## Поиск файла с использованием системы FTP Search

- 1 Ввести в поле поиска системы FTP Search имя файла ICQ. Существует возможность уточнить параметры поиска файла (тип файла, домен и др.) и параметры вывода списка найденных ссылок (количество, формат и др.).
- 2 Пользователю будет представлен список со ссылками на FTP-серверы, хранящие файл с заданным именем.



Активизация любой из ссылок приводит к началу процесса загрузки файла на локальный компьютер.

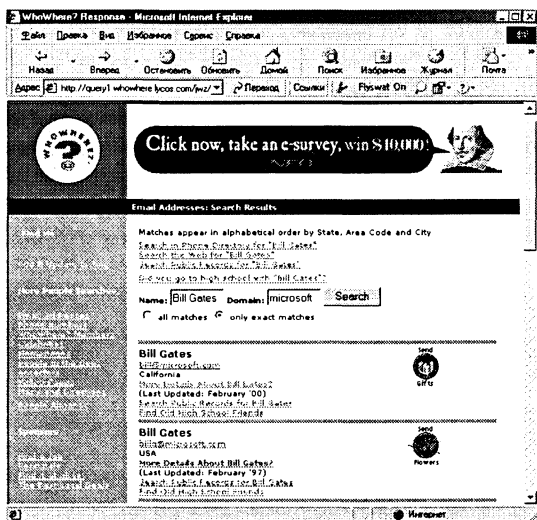
WhoWhere? ([www.whowhere.lycos.com](http://www.whowhere.lycos.com)) является специализированной поисковой системой, позволяющей искать адрес электронной почты по имени человека или, наоборот, имя человека, хозяина определенного адреса электронной почты. В окно поиска необходимо ввести имя и фамилию, после чего система осуществит поиск e-mail-адреса в базе данных, содержащей абонентов, зарегистрированных в системе.

**Практическое задание 7.36.** «Поиск адреса электронной почты». Найти адрес электронной почты по имени и фамилии абонента.



## Поиск e-mail с использованием системы WhoWhere?

- 1 Ввести в поле поиска системы WhoWhere? имя и фамилию. Можно также ввести предполагаемый домен, в котором находится почтовый сервер.



Будет выведен список однофамильцев, о каждом из которых можно получить дополнительную информацию, введенную абонентом при регистрации в системе.

### Задания для самостоятельного выполнения

- 7.40. Найти с помощью поисковых систем Rambler и Яндекс в Интернете сайты виртуальных компьютерных музеев.
- 7.41. Найти статистическую информацию о росте российской части Интернета (Рунета).
- 7.42. Найти сайты, содержащие информацию о ценах на компьютеры и комплектующие.
- 7.43. Найти сайты, содержащие информацию о провайдерах Интернета вашего региона.
- 7.44. Найти серверы, с которых можно скачать свободно распространяемое (freeware) и условно-бесплатное (shareware) программное обеспечение.
- 7.45. Найти ответы на вопросы, которые задавались на Кубке России по поиску в Интернете, проводившемся на поисковом сервере Яндекс:
  1. Какова преобладающая глубина Белого озера (Вологодская область)?
  2. Сколько весит золотник (в граммах)?
  3. Сколько кантонов, объединившись, создали Швейцарию?
  4. В каком возрасте (по мнению историков) умер фараон Тутанхамон?
  5. Какое полное имя было у Остапа Бендера?
  6. Сколько куполов на соборе Василия Блаженного на Красной площади?

7. На каком этаже в Эрмитаже висят картины импрессионистов?
  8. Что означает слово «Lego» (название известной компании по производству игрушек)?
  9. Когда изобрели компьютерную мышь?
  10. Подойдет ли бытовой блок питания (9 или 12 вольт постоянного тока на выходе) для модема Sportster?
- 7.46. Найти файл программы интерактивного общения ICQ с помощью российских поисковых систем. Сравнить результаты поиска.
- 7.47. Зарегистрировать свой почтовый адрес в системе WhoWhere?. Провести его поиск с целью проверки правильности регистрации.

## 7.6. Интерактивное общение в Интернете

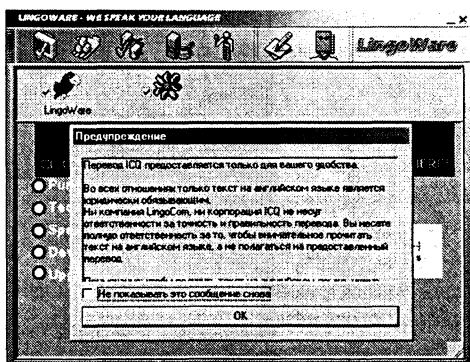
Обмен информацией в Интернете может реализовываться в различных формах: электронная почта, *chat* (обмен сообщениями в реальном времени), Интернет-телефония, пересылка файлов и др. Наиболее популярной системой интерактивного общения в настоящее время является ICQ, на серверах которой уже зарегистрированы более 160 миллионов пользователей.

**Практическое задание 7.37. «Интерактивное общение с использованием системы ICQ».** Зарегистрироваться в системе ICQ, настроить систему и реализовать различные формы интерактивного общения.

Существуют различные версии программы интерактивного общения ICQ и различные способы ее русификации. Воспользуемся универсальной программой русификации LingoWare, которая позволяет русифицировать различные программы, скачивая для этого словари с сервера компании LingoCom ([www.lingoware.com](http://www.lingoware.com)).

### Интерактивное общение с использованием системы ICQ

- 1 Запустить программу LingoWare. Скачать с сервера в Интернете словарь для русификации ICQ2001b. Запустить программу ICQ2001b. Интерфейс программы будет автоматически русифицирован.



Для интерактивного общения с пользователями глобальной системы ICQ необходимо зарегистрироваться и получить уникальный идентификационный номер ICQ#.

- 2 Щелкнуть по кнопке *МОЯ-ICQ* и ввести команду [Регистрация на ICQ-Регистрация нового пользователя].

Будет произведено соединение с сервером ICQ и начат процесс регистрации.

В процессе регистрации необходимо указать фамилию, имя, e-mail и другие данные о себе (по желанию).

Выбрать пароль и получить регистрационный номер ICQ#.



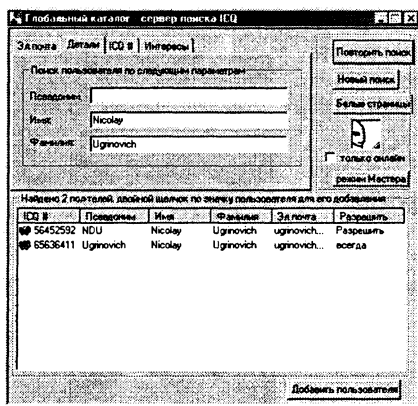
В принципе можно начать общение с любым зарегистрированным пользователем системы ICQ, однако для удобства рекомендуется создать *Мой список контактов*, в который включить наиболее часто вызываемых абонентов. Абонентов списка можно группировать по различным принципам, например, *Семья*, *Друзья*, *Сослуживцы* и т. д.

- 3 Щелкнуть по кнопке *Пригласить/Добавить пользователя*.

- 4 На диалоговой панели ввести e-mail, или фамилию и имя, или ICQ# абонента и щелкнуть по кнопке *Поиск*.

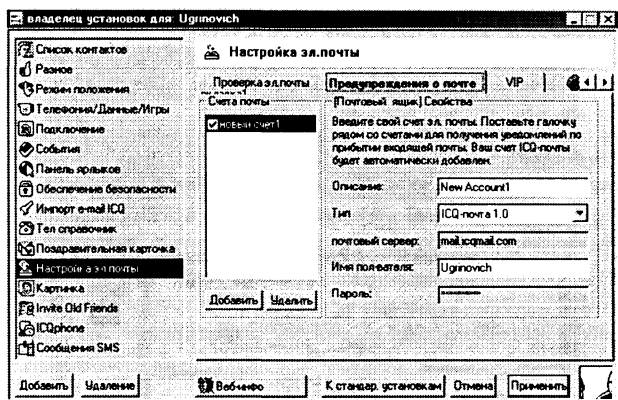
Произойдет соединение с сервером ICQ и начнется поиск в базе данных абонентов системы.

Если абонент будет найден, то ему будет послан запрос на включение в ваш список контактов.



Далее необходимо произвести настройку различных сервисов системы (электронной почты, Интернет-телефонии, пересылки файлов и др.).

- 5 Щелкнуть по кнопке *ICQ* и выбрать пункт меню *Настройка*. В левой части появившейся диалоговой панели выбрать пункт, например, *Настройка эл. почты*. В правой части произвести настройку параметров работы с ICQ-почтой.

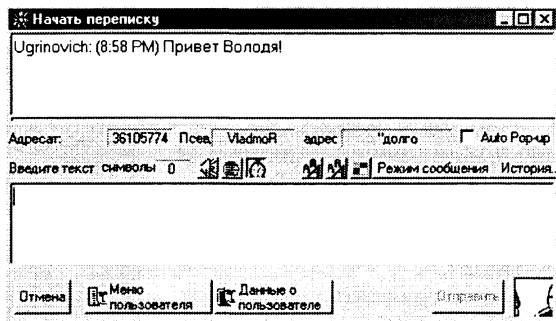


После настройки ICQ-программа готова к реализации интерактивного общения с выбранным абонентом.

- 6 Для выбора абонента щелкнуть по его имени в списке контактов.

В нижней части появившегося окна *Начать переписку* ввести текст сообщения и щелкнуть по кнопке *Отправить*.

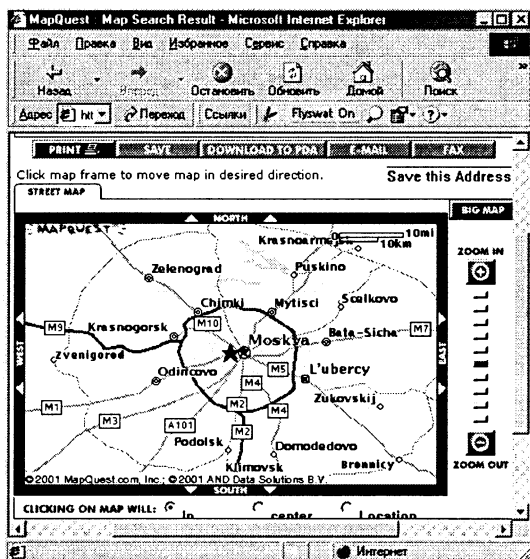
В верхней части окна будут отображаться отправленные и полученные сообщения.



ICQ-система предоставляет возможность ознакомиться с местом жительства абонентов с помощью интерактивных географических карт.



- 7 В окне *Начать переписку* щелкнуть по кнопке *Данные о пользователе*.  
На появившейся панели *Инфо о пользователе*: щелкнуть по значку *Дом* и кнопке *Показать карту*.
- 8 В окно браузера будет загружена интерактивная карта местности, где проживает абонент. Возможно изменение масштаба карты в широких пределах и перемещение по параллелям и меридианам.

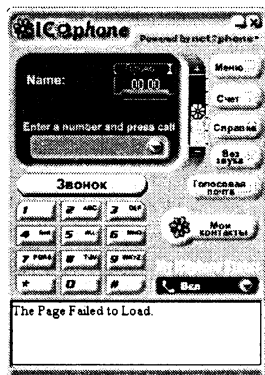


ICQ-система включает в себя функцию Интернет-телефонии (для этого на компьютере должна быть установлена звуковая плата, а к ней подключены микрофон и наушники).

- 9 Щелкнуть по кнопке *Услуги*.

В появившемся меню ввести команду [ICQphone-Запуск программы ICQphone].

С помощью появившейся панели *ICQphone* можно осуществить набор телефонного номера и начать телефонный разговор (необходимо помнить, что звонок платный).



## Задания для самостоятельного выполнения

- 7.48. Зарегистрироваться в системе ICQ. Сформировать контактный лист, произвести настройки различных сервисов системы. Осуществить обмен сообщениями, электронными письмами и файлами с выбранным абонентом. Ознакомиться с картами местностей, где проживают абоненты.

## 7.7. Потокковое аудио- и видео в Интернете

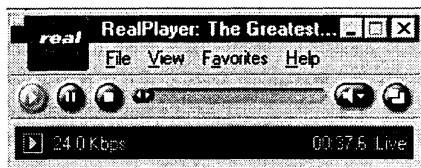
Передача в Интернете потокковой аудио- и видео информации (радио и телепрограмм, изображений от LiveCam и др.) реализуется в различных форматах (Real8, Windows Media Technology 7 и MP3) и на различных скоростях передачи информации (20 Кбит/с и выше). Для прослушивания или просмотра информации на локальном компьютере используются специальные проигрыватели RealPlayer, Windows Media Player, WinAmp, MusicMatch Jukebox и др.

**Практическое задание 7.38. «Прием Интернет-радио».** Осуществить прием Радио-101 с использованием проигрывателя RealPlayer.



### Прием Радио-101

- 1 Открыть в браузере сайт «Радио-101» ([www.101.ru](http://www.101.ru)) и выбрать формат *Real8*. Произойдет автоматический запуск проигрывателя RealPlayer и начнется прослушивание прямого радиоэфира.



**Практическое задание 7.39 «Прием Интернет-ТВ».** Осуществить прием телеканала «Культура» с использованием проигрывателя Windows Media Player.



### Прием телеканала Культура

- 1 Открыть в браузере сайт «Государственное телевидение России» ([www.rtr.ru](http://www.rtr.ru)) и активизировать ссылку *ТК Культура*. Произойдет автоматический запуск проигрывателя Windows Media Player и начнется просмотр прямого телеэфира.



### Задания для самостоятельного выполнения

- 7.49. Осуществить прием прямого радиоэфира в формате MP3 с использованием проигрывателя WinAmp.
- 7.50. Осуществить прием телеканала с использованием проигрывателя MusicMatch Jukebox.
- 7.51. Осуществить поиск в Интернете адресов LiveCam и ознакомиться с городами России и мира.

## 7.8. Покупки в Интернете

В российском Интернете существуют сотни магазинов, которые предоставляют возможность покупки десятков тысяч наименований товаров (книг, компьютеров, автомобилей и так далее). Магазин предоставляет пользователю каталог товаров, в котором содержатся необходимые для покупателя сведения о товаре (технические характеристики, внешний вид, цена и так далее). Покупатель может выбрать товары и сделать заказ на приобретение и доставку.

**Практическое задание 7.40 «Покупка книг».** Осуществить выбор и заказ книг в Интернет-магазине «Библио-Глобус».

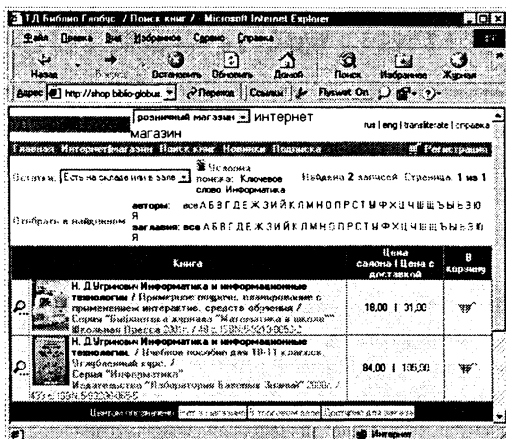


### Покупка книг в Интернет-магазине

- 1 Открыть в браузере сайт Интернет-магазина «Библио-Глобус» ([shop.biblio-globus.ru](http://shop.biblio-globus.ru)) и в поле поиска ввести ключевое слово, например, *Информатика*.
- 2 Будет выведен список из нескольких десятков названий книг.

Для уточнения поиска можно указать первую букву фамилии автора (например, *У*) или названия книги. Будет выведен новый список.

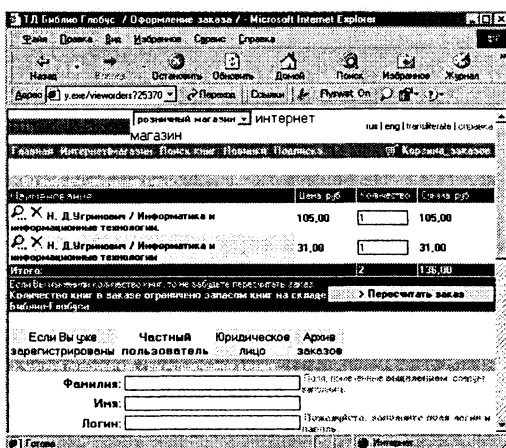
Для заказа книги достаточно щелкнуть по значку *В корзину*.



3 Для подтверждения заказа необходимо осуществить щелчок по ссылке *Корзина заказов*.

Будет выведен список заказанных книг и общая цена покупки.

Для оформления заказа необходимо заполнить форму, в которой указываются данные и адрес покупателя, способы оплаты и доставки книг.



### Задания для самостоятельного выполнения

7.52. Осуществить поиск в Интернет-магазинах комплектующих для компьютера.

7.53. Осуществить поиск в Интернет-магазинах аудио компакт-дисков.

# Глава 8

---

## Разработка Web-сайтов

---

В процессе изучения темы  
рекомендуется установить:

CD-ROM 

- Web-редакторы FrontPage Express (входит в состав Microsoft Internet Explorer 4), StarOffice Writer (входит в состав StarOffice), AceHTML, 1st Page 2000;
- браузер Сибкон Коммуникатор.

### 8.1. HTML — язык разметки гипертекста

**Структура Web-страницы.** Большая часть тэгов образует контейнер, состоящий из открывающего и закрывающего тэгов. Тэги можно набирать как заглавными, так и строчными буквами.

Web-страница помещается в контейнер `<HTML></HTML>` и состоит из двух частей: заголовка и отображаемого в браузере содержания.

Заголовок страницы помещается в контейнер `<HEAD></HEAD>`. Заголовок содержит название страницы, которое помещается в контейнер `<TITLE></TITLE>` и при просмотре отображается в верхней строке окна браузера.

Также в заголовок помещаются не отображаемые при просмотре мета-тэги, задающие кодировку страницы для ее правильного отображения в браузере, а также содержащие описание и ключевые слова страницы, которые в первую очередь просматривают роботы поисковых систем.

Отображаемое в браузере содержание страницы помещается в контейнер `<BODY></BODY>`.

**Практическое задание 8.1. «Основные тэги HTML».** Создать Web-страницу, знакомящую с основными тэгами HTML.

 **Создание Web-страницы «Основные тэги HTML»**

- 1 Запустить текстовый редактор Блокнот командой [Пуск-Программы-Стандартные-Блокнот].

## 2 Ввести HTML-код страницы.

```
<HTML>
<HEAD>
<TITLE>Первое знакомство с тэгами HTML</TITLE>
</HEAD>
<BODY>

</BODY>
</HTML>
```

## 3 Ввести команду [Файл-Сохранить]. Файлу Web-страницы присвоить имя index.htm.

## 4 Запустить браузер и открыть созданный файл командой [Файл - Открыть]. В заголовке окна браузера высвечивается название Web-страницы *Первое знакомство с тэгами HTML*.

**Заголовки.** Внести в текст страницы тэги заголовков различных уровней (размеров).

```
5 <H1>Заголовок первого уровня</H1>
   <H2>Заголовок второго уровня</H2>
   <H3>Заголовок третьего уровня</H3>
   <H4>Заголовок четвертого уровня</H4>
   <H5>Заголовок пятого уровня</H5>
   <H6>Заголовок шестого уровня</H6>
```

**Форматирование шрифта.** Внести в текст страницы тэги, определяющие начертание шрифта. Отделить этот фрагмент от остального текста с помощью горизонтальных разделительных линий.

```
6 <HR>
   Обычный текст
   <B>Жирный</B>
   <I>Курсив</I>
   <U>Подчеркнутый</U>
   <B><I><U>Жирный подчеркнутый курсив </B></I></U>
   <TT>Равноширинный</TT>
   <EM>Выделение</EM>
   <STRONG>Усиленное выделение</STRONG>
   <HR>
```

**Списки.** Внести в текст страницы тэги, задающие списки нумерованные и нenumерованные, а также списки определений.

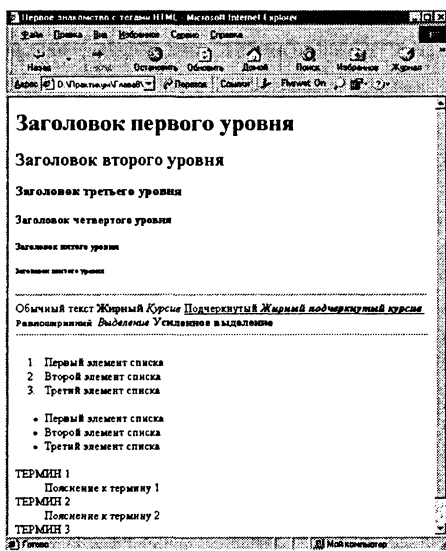
```

7 <OL>
  <LI>Первый элемент списка</LI>
  <LI>Второй элемент списка</LI>
  <LI>Третий элемент списка</LI>
</OL>
<UL>
  <LI>Первый элемент списка</LI>
  <LI>Второй элемент списка</LI>
  <LI>Третий элемент списка</LI>
</UL>
<DL>
  <DT>ТЕРМИН 1</DT>
  <DD>Пояснение к термину 1</DD>
  <DT>ТЕРМИН 2</DT>
  <DD>Пояснение к термину 2</DD>
  <DT>ТЕРМИН 3</DT>
  <DD>Пояснение к термину 3</DD>
</DL>

```

**Внесение изменений и дополнений в Web-страницу.** В процессе создания Web-страницы приходится добавлять новые теги и просматривать получаемый результат.

8 Активизировать Блокнот с открытой в нем редактируемой Web-страницей. Внести в содержимое страницы необходимые изменения и сохранить новый вариант страницы, выполнив команду меню [Файл-Сохранить].



9 Активизировать браузер с открытым в нем предыдущим вариантом страницы. Щелкнуть по кнопке *Обновить*. В окне браузера отобразится обновленная Web-страница.

**Мета-тэги.** В раздел заголовка Web-страницы могут быть добавлены информационные одиночные тэги <META>, имеющие атрибуты NAME, HTTP-EQUIV и CONTENT.

Мета-тэг может информировать браузер о кодировке Web-страницы:

```
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1251">
```

Мета-тэги используются поисковыми системами для индексирования содержания, ключевых слов и автора Web-страницы:

```
<meta name="Description" content="">
<meta name="Keywords" content="">
<meta name="Author" content="">
```

10 Включить мета-тэги в заголовок страницы index.htm, присвоив атрибуту CONTENT необходимые значения.

На CD-ROM готовая Web-страница хранится в файле page8-1.htm в каталоге \practicum\html\.

### **Задания для самостоятельного выполнения**

- 8.1. Разработать Web-страницу, демонстрирующую возможности форматирования текста и создание списков различных типов с использованием Web-редактора (например, Web-мастера, входящего в Сибкон Коммуникатор).
- 8.2. Создать заготовку страницы с использованием Web-редактора АсеHTML. Обратит внимание на наличие мета-тэгов.
- 8.3. Ознакомиться с форматом HTML-тэгов с помощью справочной системы по HTML.

## **8.2. Цветовая схема, вставка изображений, гиперссылок и таблиц**

Web-сайты обычно являются мультимедийными и интерактивными, то есть содержат изображения и гиперссылки. При форматировании текста, вставке изображений и гиперссылок и так далее используются тэги с атрибутами.

Атрибуты и присвоенные им значения записываются внутри открывающего тэга. В одном тэге можно использовать несколько атрибутов, разделенных между собой пробелами, а одни и те же атрибуты могут использоваться в различных тэгах.



**Практическое задание 8.2. «Московский Кремль».** Создать Web-сайт, рассказывающий о башнях Московского Кремля.

Сайт рекомендуется хранить в отдельной, специально созданной папке Kremlin. Здесь будут находиться как Web-страницы, так и графические файлы. Файл начальной страницы сайта обычно имеет имя index.htm, в этом случае для начала просмотра сайта в адресе достаточно указать путь к папке, где он хранится.

Начальная страница «Башни Московского Кремля» будет содержать информацию о башнях Московского Кремля: история создания, описания отдельных башен, иллюстрации и т. д. Вторая страница «Кремлевские Куранты» рассказывает о Кремлевских Курантах. Страницы связаны между собой гиперссылками.

Создадим начальную Web-страницу, в процессе работы над которой познакомимся с применением атрибутов тэгов для вставки изображений, гиперссылок, таблиц и карт.



### Создание Web-сайта «Московский Кремль»

- 1 Запустить текстовый редактор Блокнот командой [Пуск - Программы - Стандартные - Блокнот].
- 2 Дать странице название «Башни Московского Кремля», между тэгами <BODY> и </BODY> набрать текст о Московском Кремле или скопировать его из текстового файла kremlin.doc, который находится на CD-ROM в каталоге \practicum\html\Kremlin\.
- 3 Отформатировать текст (разметить абзацы и расставить заголовки) и сохранить в файле index.htm в папке Kremlin.
- 4 Просмотреть полученную черно-белую страницу в браузере:

The screenshot shows the Microsoft Internet Explorer browser window. The title bar reads 'Башни Московского Кремля - Microsoft Internet Explorer'. The address bar contains 'kremlin.htm'. The main content area displays the following text:

**Башни Московского Кремля**

Современный Кремль построен в конце XV - начале XVI в. Непрístupная крепость со всех сторон была окружена водой. с юга - Москва-река, с севера и запада - река Неглинная, с востока - ров глубиной 10 м и шириной 32 м, выложенный белым камнем.

Протяженность укреплений более 2-х километров, высота от 5 до 19 м, толщина от 3,5 до 6,5 м. Из стен выступают вперед 18 боевых башен. Расстояние между ними позволяло простреливать весь защищаемый периметр. Первоначально башни сверху были плоскими, только навесы прикрывали воинов от дождя и снега.

The browser interface includes a menu bar (Файл, Правка, Вид, Избранное, Сервис, Справка), a toolbar with navigation buttons (Назад, Вперед, Остановить, Обновить, Домой, Поиск, Избранное, Журнал), and a status bar at the bottom showing 'Готово' and 'Мой компьютер'.

**Атрибуты задания цветовой схемы (цвета фона, текста и гиперссылок).** Цвет на Web-странице задают либо его названием, либо числовым шестиразрядным шестнадцатеричным кодом #RRGGBB (первые два разряда задают интенсивность красного цвета, вторые — зеленого и третьи — синего).

Значение яркости цвета может меняться от минимальной 00 до максимальной FF. В таблице приведены примеры некоторых цветов:

цвет	код	название	цвет	код	название
черный	#000000	black	фиолетовый	#FF00FF	magenta
белый	#FFFFFF	white	бирюзовый	#00FFFF	cyan
красный	#FF0000	red	желтый	#FFFF00	yellow
зеленый	#00FF00	lime	золотой	#FFD800	gold
синий	#0000FF	blue	оранжевый	#FFA500	orange
серый	#808080	gray	коричневый	#A82828	brown

Основную цветовую схему Web-страницы можно задать в тэге <BODY> с помощью атрибутов:

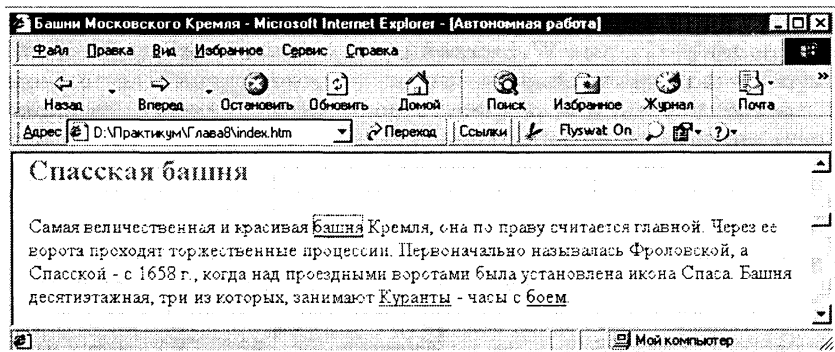
Цвет фона	BGCOLOR="#RRGGBB"
Текстура фона	BACKGROUND="file_name"
Цвет текста	TEXT="#RRGGBB"
Цвет текста ссылки	LINK="#RRGGBB"
Цвет текста активной ссылки	ALINK="#RRGGBB"
Цвет текста просмотренной ссылки	VLINK="#RRGGBB"

При использовании текстуры, закрывающей собой всю площадь страницы, применение однотонного фона кажется излишним. Однако рисунки загружаются несколько медленнее, чем текст. Все это время посетители страницы будут видеть цвет фона, заданный атрибутом BGCOLOR. Поэтому для фона указывают цвет, совпадающий с основным тоном фонового рисунка.

Чтобы текст хорошо читался, цвета на странице подбирают контрастирующие по яркости: пастельный фон — темный текст, или темный фон — светлый текст. Нежелательны буквы белого цвета — они могут оказаться невидимыми при печати страницы на принтере.

**5** Для оформления страницы о Московском Кремле использовать следующую цветовую схему:

```
<BODY BGCOLOR="#FFFFCC" BACKGROUND="fon.png"
TEXT="#993300" LINK="#00FF00" ALINK="#FF0000"
VLINK="#00FF00">
```



**Вставка изображений.** Для размещения на Web-страницах используются графические файлы форматов GIF, JPEG и PNG.

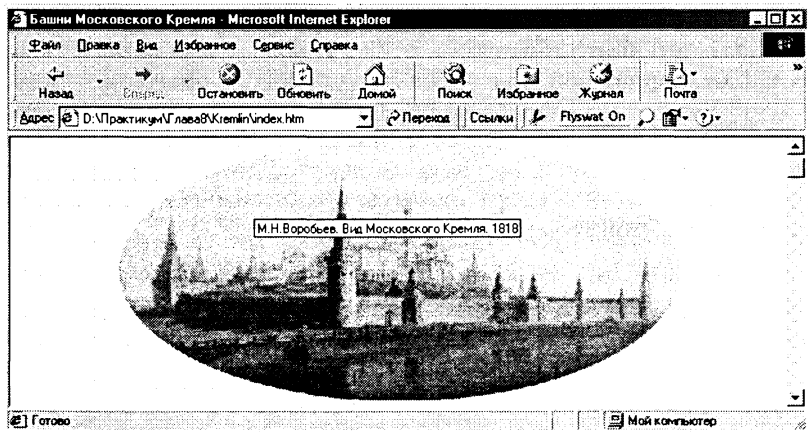
Изображения помещаются на Web-страницу тэгом `<IMG>` с атрибутом SRC, сообщаящим браузеру имя и местоположение графического файла.

```
<IMG SRC="image_name">
```

Выравнивание рисунка по горизонтали в этом случае задается в тэге `<P>`. К изображению с помощью атрибута `<ALT>`, можно добавить название, которое появляется вместо иллюстрации (если по какой-либо причине графика не выводится в окне браузера), или в форме всплывающий строки, при указании на рисунок мышью.

6 Вставить в начале страницы картину М. Н. Воробьева «Вид Московского Кремля. 1818»:

```
<P ALIGN="center"> <IMG SRC="kremlin.gif"
WIDTH="515" HEIGHT="240" ALT="М.Н.Воробьев. Вид
Московского Кремля. 1818"></P>
```

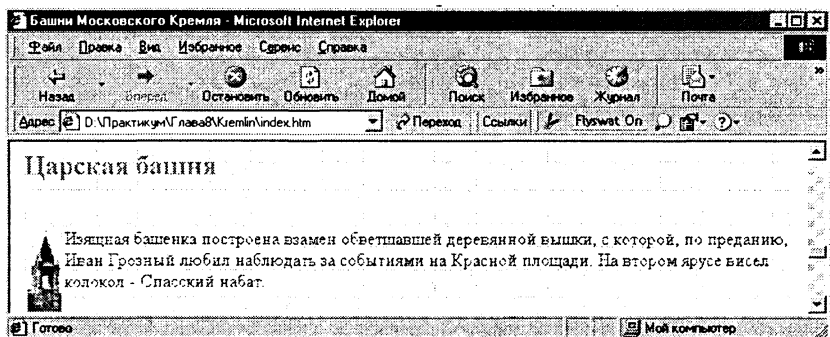


В тэге IMG с помощью атрибутов WIDTH (ширина) и HEIGHT (высота) можно указать размер выводимой картинки (в пикселях).

Атрибут ALIGN выравнивает текст по верхнему краю, середине или нижнему краю изображения, справа или слева от него с помощью значений: top, bottom, middle, left или right.

7 Разместить изображение Царской башни определенного размера слева от текста:

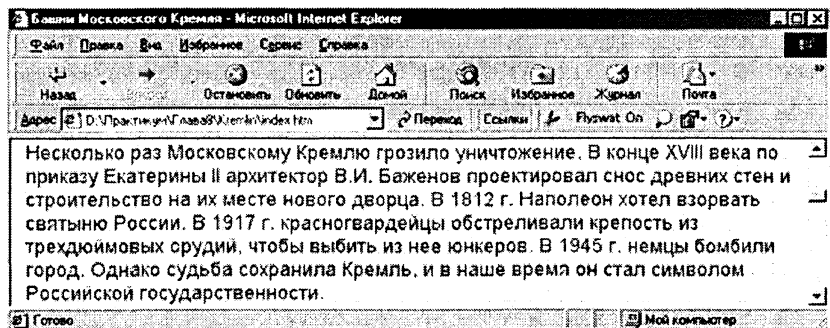
```
<IMG SRC="carsk.gif" WIDTH="30" HEIGHT="75"
ALIGN="left">
```



**Форматирование текста.** Для выделения фрагментов текста используется тэг <FONT>. Атрибут FACE определяет гарнитуру шрифта, атрибут COLOR — цвет и атрибут SIZE — размер символов.

8 Выделим часть текста более крупным шрифтом и цветом:

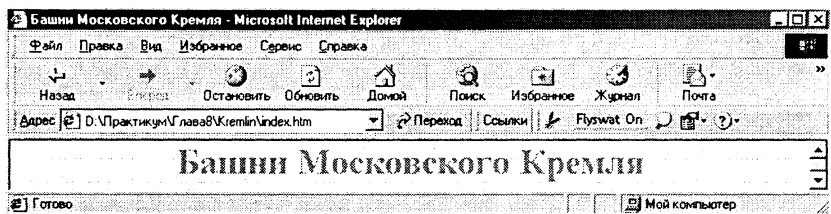
```
<FONT SIZE="4" COLOR="#FF0066" FACE="Arial">
Несколько раз Московскому Кремлю грозило унич-
тожение... </FONT></P>
```



Выравнивание текста по горизонтали задает атрибут ALIGN. Его возможные значения: left — выравнивание по левому краю, center — по центру и right — по правому краю.

9 Выравнивание можно добавлять к тэгам, задающим заголовки, абзацы, или размещающим информацию в таблице. Выровнять по центру заголовков страницы.

```
<H1 ALIGN="center"><FONT COLOR="#FF6600">
Башни Московского Кремля</FONT></H1>
```



**Различные виды гиперссылок.** Связать Web-страницу с другими документами можно с помощью универсального тэга <A> и его атрибута HREF, указывающих в каком файле хранится вызываемый ресурс.

```
<A HREF="file_name">Указатель ссылки</A>
```

file\_name — путь к файлу или его URL-адрес в Интернете. Если вызываемый документ размещается в той же папке, что и Web-страница, то можно указывать только имя файла.

Указатель ссылки в окне браузера выделяется подчеркиванием и особым цветом. При указании на него мышью ее курсор превращается в значок «рука». Щелчок мыши по указателю вызывает переход на документ, указанный в гиперссылке.

Используем различные значения атрибута HREF для реализации различных реакций браузера:

10 Ссылка на Web-страницу локального компьютера

```
<A HREF="clock.htm">Куранты</A>
```

**Ссылка на Web-страницу, размещенную в Интернете**

```
<A HREF="http://www.moskva.ru">Сайт о Москве</A>
```

**Вставка изображения**

```
<A HREF="spassk.jpg">Башня</A>
```

**Запуск проигрывателя звукового файла**

```
<A HREF="strike.wav">Бой часов</A>
```

## Сохранение файла на локальном компьютере

```
<A HREF="kremlin.zip">Скачать файл</A>
```

## Создание бланка письма с заполненным адресом получателя

```
<A HREF="mailto:your_name@freemail.ru">Создать письмо</A>
```

В качестве указателя ссылки можно использовать не только текст, но и изображение. В этом случае указатель ссылки задается с помощью тэга <IMG>.

Пусть в нашем проекте при щелчке мышью по маленькому рисунку Царской башни вызывается ее крупная фотография. Такой прием часто используют для снижения времени загрузки страницы. Посетитель увидит уменьшенные копии рисунков, а при желании сможет загрузить полномасштабное изображение.

**11**

```
<A HREF="carsk.jpg"><IMG SRC="carsk.gif" WIDTH="30" HEIGHT="75"></A>
```

**Якоря. Гиперссылки на якоря.** Если создаваемая страница очень большая, то в ее тексте можно расставить якоря (иначе их еще называют метки или закладки). Они помечают смысловые разделы и помогают быстро перемещаться по документу.

**12** Поместить метку `top` в начало страницы. Для этого атрибуту `NAME` тэга необходимо присвоить определенное значение (имя метки):

```
<A NAME="top"></A>
```

**13** В конец страницы поместить указатель ссылки на метку. Для этого атрибуту `HREF` тэга необходимо присвоить значение (имя метки):

```
<A HREF="#top">На начало страницы</A>
```

Теперь, находясь в конце страницы, для перемещения в начало страницы достаточно осуществить щелчок по указателю гиперссылки *На начало страницы*.

Переключаться на якорь по гиперссылке можно не только внутри текущей Web-страницы, но и с другой Web-страницы.

**14** Установить метку на главной странице сайта `index.htm` в начале описания Спасской башни:

```
<A NAME="anchor1"></A>
```

- 15 Создать страницу «Кремлевские Куранты» и сохранить ее в файле `clock.htm` в папке `Kremlin`.



- 16 На странице `clock.htm` добавить гиперссылку перехода на главную страницу на метку `anchor1`, установленную в абзаце о Спасской башне:

```
<A href= "index.htm#anchor1">
```

```
На главную страницу</A>
```

Теперь после перехода со страницы «Башни Московского Кремля» на страницу «Кремлевские куранты» можно вернуться обратно на тот же абзац, с которого был осуществлен первый переход.

**Ссылки с использованием карт-изображений (Map Image).** Для создания гиперссылок можно использовать карты-изображения, на которых выделены области — указатели гиперссылок. С каждой областью изображения связывают переход на определенный объект (внешнюю Web-страницу или внутренний якорь).

В качестве основы для создания карты-изображения можно выбрать любое изображение. Чтобы преобразовать изображение в карту, в тэг `<IMG>` добавляется атрибут `USEMAP`, значением которого является ссылка на имя описания конфигурации областей на карте. Для рисования рамки вокруг карты-изображения используется атрибут `BORDER`, значением которого является толщина рамки в пикселях.

- 17** Выбрать в качестве основы для карты-изображения схему Московского Кремля (файл схема.png, которая хранится на CD-ROM в каталоге \practicum\html\Kremlin\):

```
<img SRC="схема.png" WIDTH="265" HEIGHT="330"  
BORDER="1" USEMAP="#Kremlin" ALIGN="left"  
ALT="Схема Московского Кремля">
```

Для описания конфигурации областей карты-изображения используется тэг <MAP>, единственным атрибутом которого является NAME. Значение атрибута NAME должно соответствовать значению атрибута USEMAP, ранее заданному в тэге <IMG>.

- 18** Тэгу NAME присвоить имя конфигурации карты-изображения:

```
<MAP NAME="Kremlin">  
</MAP>
```

Форма и геометрические размеры областей на карте-изображении, а также адрес гиперссылки задаются тэгами <AREA> с атрибутами. Атрибут SHAPE определяет форму области и может принимать значения rect — прямоугольник, circle — круг и poly — многоугольник.

Атрибут COORDS задает координаты каждой области в пикселях. Значением атрибута является список координат точек, определяющих область, разделенных запятыми. Для прямоугольника задаются координаты левого верхнего и правого нижнего углов, для круга — координаты центра и радиус, для многоугольника — координаты вершин.

Для задания адреса гиперссылки используется атрибут HREF, а для вывода альтернативного текста атрибут ALT.

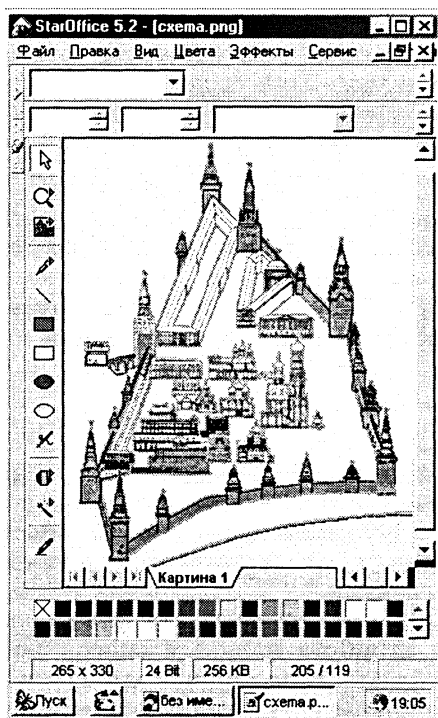
- 19** Выбрать в качестве областей прямоугольные изображения башен, в контейнер ввести координаты 20 областей (изображений башен), адреса ссылок (якоря) и альтернативные тексты:

```
<area shape="rect" coords="175,95,183,119"  
href="#anchor9" alt="Сенатская">  
<area shape="rect" coords="129,21,148,87"  
href="#anchor2" alt="Никольская">  
<area shape="rect" coords="103,3,116,57"  
href="#anchor3" alt="Угловая Арсенальная">
```

- 20** Координаты областей можно узнать, открыв рисунок в каком-либо графическом редакторе, например, StarOffice Draw.

Достаточно навести курсор на интересующую точку, и в строке состояния высветятся координаты курсора.





Активизация гиперссылок будет вызывать переход на абзац текста, относящийся к соответствующим башням (начало каждого раздела заранее помечено якорем).

**Таблицы.** Таблицу формируют несколько различных тэгов. Таблица задается контейнером `<TABLE>`, внутри которого содержится описание структуры таблицы и ее содержания.

Любая таблица состоит из строк, которые задаются контейнером `<TR>` (Table Row), в который помещается описание ячеек.

Формат ячеек и их содержание помещается в контейнер `<TD>` (Table Data), а заголовки ячеек в контейнер `<TH>` (Table Header).

Толщина разделительных линий в таблице задается с помощью атрибута `BORDER`, а для выравнивания информации внутри ячейки по горизонтали применяют атрибут `ALIGN`.

**21** Создать таблицу и внести в нее данные о башнях Кремля:

```
<TABLE border="1">
<TR>
<TH>Башня</TH>
```

```

<TH>Год сооружения</TH>
<TH>Архитектор</TH>
<TH>Высота</TH>
</TR>
<TR>
<TD>Спасская</TD>
<TD align="center">1491</TD>
<TD align="center">П.Соларио</TD>
<TD align="center">71</TD>
</TR>
... ..
<TD>Беклемишевская</TD>
<TD align="center">1487</TD>
<TD align="center">М.Руффо</TD>
<TD align="center">46</TD>
</TR>
</TABLE>

```



Готовый сайт «Московский Кремль» хранится на CD-ROM в каталоге \practicum\html\Kremlin\.

### Задания для самостоятельного выполнения

- 8.4. Разработать Web-сайт «Виртуальный зоопарк», содержащий интересные факты из жизни «братьев наших меньших», а также шутки про животных. Для создания сайта воспользоваться Web-редактором (например, FrontPage Express, входящим в состав Microsoft Internet Explorer 4), а также текстовыми материалами, фотографиями и рисунками животных, хранящимися на CD-ROM.

- 8.5. Воспользоваться для создания карты-изображения Web-редактором (например, AceHTML).
- 8.6. Разработать Web-сайт «Устройство компьютера».

## 8.3. Интерактивные формы на Web-страницах

Интерактивные формы предназначены для сбора информации от посетителей Web-страницы. Внешне форма выглядит, как анкета, в которую посетитель может вписать свой текст или выбрать значения из предлагаемых перечней.

**Формы.** Форма помещается в специальный контейнер `<FORM></FORM>`, внутри которого располагаются все тэги элементов формы. Там же обязательно находится кнопка, отправляющая заполненную анкету на сервер для последующей обработки данных.

Поля формы создаются с помощью тэгов `<INPUT>`, `<SELECT>` и `<TEXTAREA>`, в которых с помощью атрибутов задаются параметры полей. С помощью атрибута `TYPE` задается тип поля, атрибут `NAME` присваивает полю имя, а атрибут `VALUE` содержит его значение. Для различных конкретных типов поля существуют также дополнительные атрибуты.

При обработке формы на сервер передаются пары `NAME=VALUE`, где `VALUE` — значение соответствующего атрибута, установленное пользователем или заданное по умолчанию.

**Текстовые поля.** Текстовое поле создается тэгом:

```
<INPUT TYPE="text" NAME="regist" VALUE="">
```

Атрибут `VALUE` принимает такое значение, которое задается пользователем при заполнении поля. Если пользователь ввел в поле текст «Фамилия», то на сервер будет передано `regist=Фамилия`.

**Флажки.** Флажки могут объединяться в группы присвоением атрибутам `NAME` всех флажков одинакового значения. Для установки флажка по умолчанию используется атрибут `CHECKED`. Группа флажков создается тэгами:

```
<INPUT TYPE="checkbox" NAME="chb1" VALUE="1" CHECKED>
```

```
<INPUT TYPE="checkbox" NAME="cnb1" VALUE="2">
```

На сервер передается значение атрибута `VALUE` флажка, установленного пользователем. Если пользователем установлены несколько флажков, то на сервер будут переданы значения уставленных флажков группы через запятую `chb1=1,2`.

**Переключатели.** Переключатели существуют только в составе группы, что обеспечивается присвоением атрибутам NAME всех переключателей одинакового значения. Для установки переключателя по умолчанию используется атрибут CHECKED. Группа переключателей создается тэгами:

```
<INPUT TYPE="radio" NAME="rad1" VALUE="1" CHECKED>  
<INPUT TYPE="radio" NAME="rad1" VALUE="2">
```

На сервер передается значение атрибута VALUE переключателя, установленного пользователем. Если пользователем установлен первый переключатель, то на сервер будет передано rad1=1.

**Списки.** Списки предоставляют пользователю выбор элементов в форме ниспадающего меню (значение атрибута SIZE=1) или списка прокрутки. Список помещается в контейнер:

```
<SELECT NAME="list" SIZE=N>  
<OPTION SELECTED>Первый  
<OPTION> Второй  
<OPTION> Третий  
<SELECT>
```

На сервер передается значение атрибута OPTION, выбранного пользователем или установленного по умолчанию атрибутом SELECTED. Если пользователем выбран третий элемент списка, то на сервер будет передано list=Третий.

**Текстовая область.** Текстовая область представляет собой текстовое поле с заданным количеством строк (значение атрибута ROWS) и столбцов (значение атрибута COLS). Создается тэгом:

```
<TEXTAREA NAME="resume" ROWS=M COLS=N>  
Текст по умолчанию  
</TEXTAREA>
```

На сервер передается содержимое поля, если пользователь не изменял текст, то resume=Текст по умолчанию.

**Кнопки.** На форме должны присутствовать кнопки, которые реализуют отправку данных из формы для обработки на сервер и очистку формы от введенных данных.

Кнопка отправки данных формы реализуется с помощью тэга:

```
<INPUT TYPE="submit" VALUE="Отправить">
```

Кнопка очистки данных формы реализуется с помощью тэга:

```
<INPUT TYPE="reset" VALUE="Очистить">
```

**Практическое задание 8.3. «Web-страница с формой».** Создать Web-страницу с анкетой для абитуриентов с использованием Web-редактора FrontPage Express.

Во FrontPage Express форма создается автоматически при добавлении на Web-страницу любого элемента формы. На странице редактора область формы выделяется пунктирным прямоугольником, внутри которого появляется первый элемент.

Добавляются элементы формы на страницу или с помощью команды меню [Вставка-Поле формы...], или специальными кнопками на *Панели инструментов полей формы*:



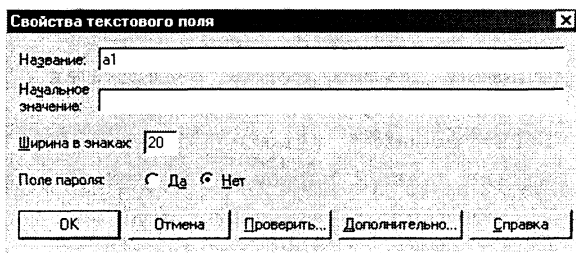
Рис. 8.1. Панель инструментов полей формы

Для каждого размещенного на странице элемента формы необходимо настроить его свойства (задать значения атрибутам NAME, VALUE и др.). Для этого необходимо выделить элемент формы и ввести команду [Правка-Свойства поля формы] с помощью меню приложения или контекстного меню. В результате откроется специальное окно свойств элемента, в котором необходимо произвести установки значений.



### Создание Web-страницы «Анкета абитуриента»

- 1 Запустить Web-редактор FrontPage Express.
- 2 Разместить на форме четыре *Текстовых поля* для ввода фамилии, имени, отчества и возраста.
- 3 Последовательно вызвать окно свойств полей и задать им названия (присвоить атрибуту NAME значения a1, a2, a3, a4).



- 4 Разместить на форме три *Переключателя* для определения наличия медали.

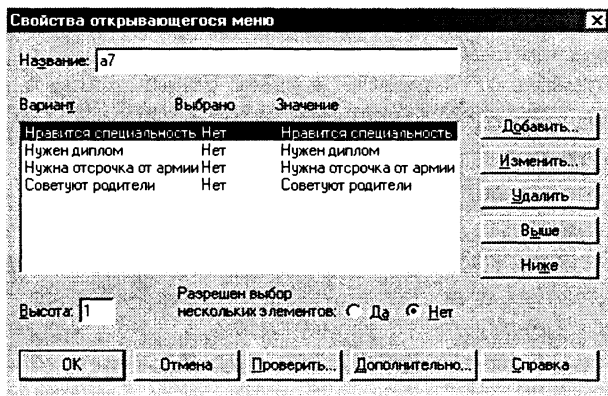
Последовательно вызвать окно свойств переключателей и задать им одинаковое имя группы (присвоить атрибуту NAME значение a5) и значения (присвоить атрибуту VALUE значения Золотая, Серебряная, Нет).

- 5 Добавить четыре элемента типа *Флажок* для выбора предметов, по которым в аттестате стоит оценка «Отлично».

Задать им одинаковое имя группы (присвоить атрибуту NAME значение a6) и значения (присвоить атрибуту VALUE значения: Математика, Информатика, Физика, Химия).

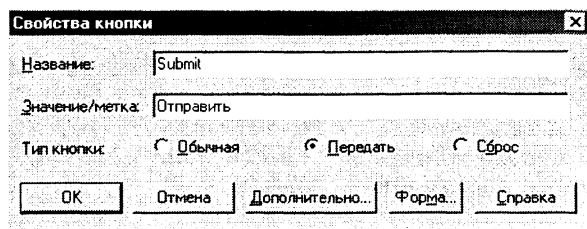
- 6 Добавить элемент *Список*.

В окне свойств ввести его название (NAME=a7) и, последовательно щелкая по кнопке *Добавить*, ввести элементы списка (значения тэга OPTION).



- 7 Добавить *Многострочное текстовое поле* для внесения произвольного послания от посетителя страницы. Ввести его название (NAME=a8) и установить размер (присвоить значения атрибутам ROWS и COLS).

- 8 Поместить на форму кнопки *Очистить* и *Отправить*. Установить для них тип кнопки (присвоить атрибуту TYPE значения reset и submit).



После размещения на форме полей различных типов следует определить способ обработки введенных пользователем данных. Для этого необходимо присвоить значения атрибутам ACTION и METHOD тэга <FORM>.

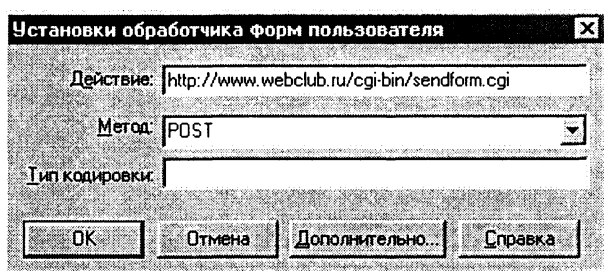
Значением атрибута ACTION является URL-адрес программы, которая будет обрабатывать данные из формы, а значениями атрибута METHOD являются либо get (по умолчанию), либо post, которые определяют форму пересылки данных.

```
<FORM ACTION="URL" METHOD="post">
```

Для обработки форм на сервере размещают специальные программы (такая программа, например, размещена на сайте клуба Web-мастеров по адресу [www.webclub.ru](http://www.webclub.ru)).

9 В контекстном меню формы выбрать пункт *Свойства формы...*

В окне *Свойства формы* выбрать в раскрывающемся списке пункт *Специальный сценарий ISAPI, NSAPI или CGI* и щелкнуть по кнопке *Настройка...*

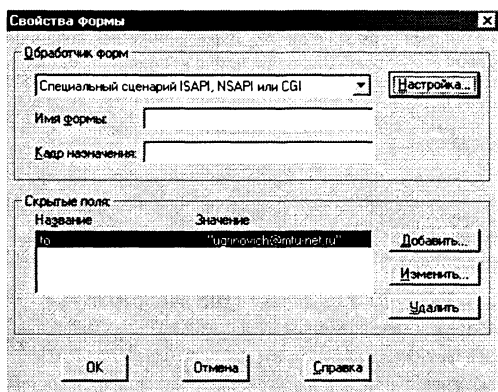


10 В появившемся окне *Установки обработчика форм пользователя* в поле *Действие:* вписать полный URL-адрес программы-обработчика на сайте Web-мастеров <http://www.webclub.ru/cgi-bin/sendform.cgi>  
В раскрывающемся списке *Метод:* выбрать строку *POST*.

Для того чтобы программа-обработчик смогла прислать нам результат обработки анкеты по электронной почте, необходимо в форме указать свой адрес. Это делается с помощью скрытых полей:

11 В окне *Свойства формы* в разделе *Скрытые поля* нажать кнопку *Добавить...*

В окне *Имя и значение* в поле *Название:* ввести *to* (то есть *Кому*), в поле *Значение:* ввести *your\_name@server\_name* (ваш адрес электронной почты).



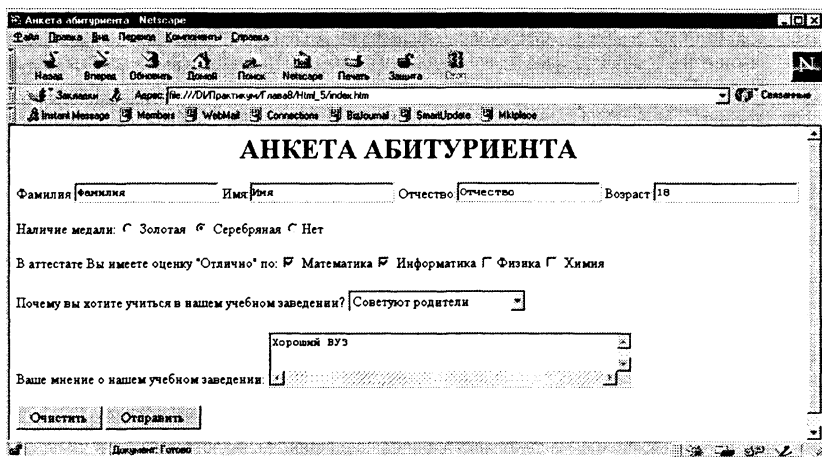
12 Сохранить полученную Web-страницу в файле anketa.htm.

Теперь Web-страница с формой полностью готова и можно приступить к ее апробации.

Готовая Web-страница «Анкета абитуриента» хранится на CD-ROM в файле page8-3.htm в каталоге \practicum\html\.

13 Запустить браузер, например, Сибкон Коммуникатор и открыть Web-страницу с формой.

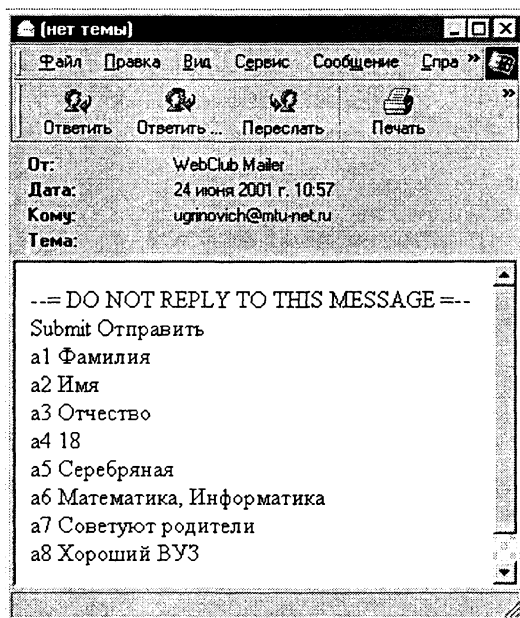
Заполнить анкету и щелкнуть по кнопке *Отправить*.



14 Программа-обработчик, обработав форму, пришлет письмо, содержащее имена элементов и выбранные значения.

Информация по каждому элементу располагается в отдельной строке, причем строки отсортированы не по порядку пунктов в анкете, а по алфавиту.





### Задания для самостоятельного выполнения

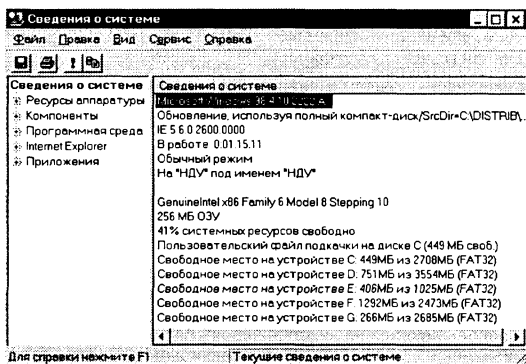
- 8.7. Разработать Web-страницу с анкетой для поступающих в десятый класс вашего учебного заведения (воспользоваться Web-редактором StarOffice Writer).
- 8.8. Разработать Web-страницу с шутливой анкетой для поступающих в частную школу «Золотой ключик» (воспользоваться Web-редактором 1st Page 2000).
- 8.9. Разработать Web-страницу с тестом по информатике.

# Ответы и указания по выполнению заданий

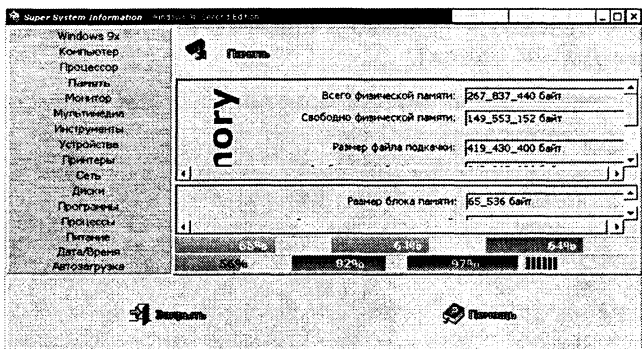
## Ответы к главе 1

### Компьютер и программное обеспечение

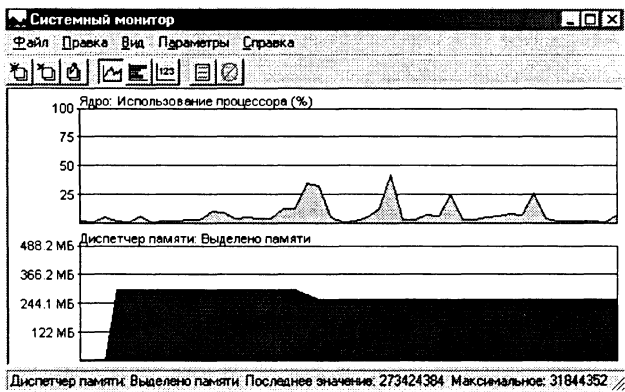
- 1.2. Запустить служебную программу Сведения о системе. Выбрать в левой части окна приложения тип устройств, в правой появится информация об этих устройствах.



- 1.3. Запустить программу Super System Information. Выбрать в левой части окна приложения тип устройств, в правой появится информация об этих устройствах.

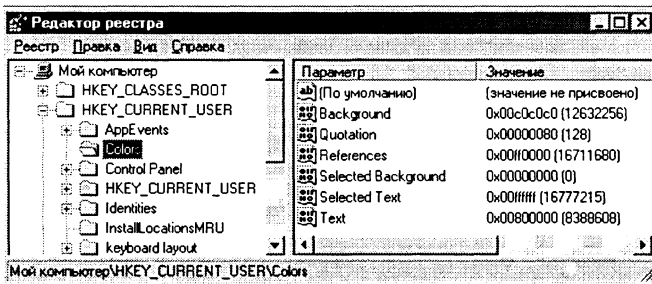


- 1.6. Запустить служебную программу Системный монитор. Ввести команду [Правка-Добавить показатель...] и выбрать, например, *Использование процессора (%)* и *Выделено памяти (Мбайт)*.



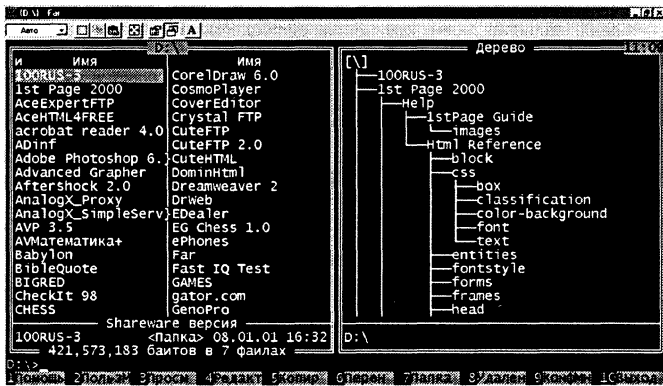
1.11 Найти в папке Windows файл редактора реестра Regedit.exe и запустить его.

В левой части окна приложения выбрать запись, в правой части появятся конфигурационные параметры.



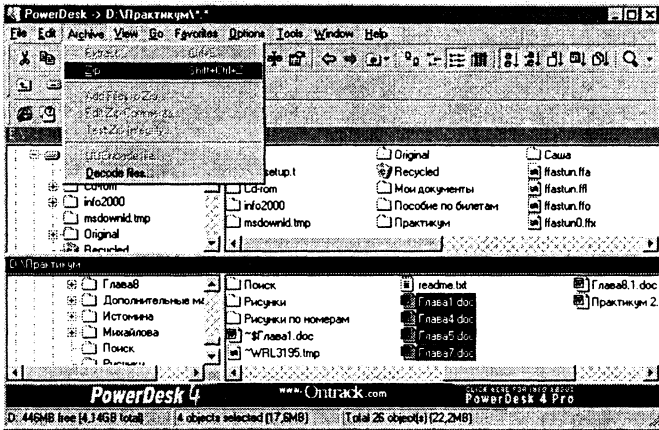
1.12. Запустить файловый менеджер FAR.

Для просмотра файловой системы ввести команду [Правая-Дерево папок].



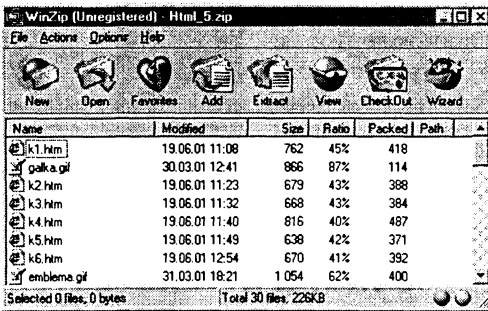
1.13. Запустить файловый менеджер PowerDesk.

Для архивирования файлов выделить их и ввести команду [Archive-Zip].



1.14. Запустить архиватор WinZip.

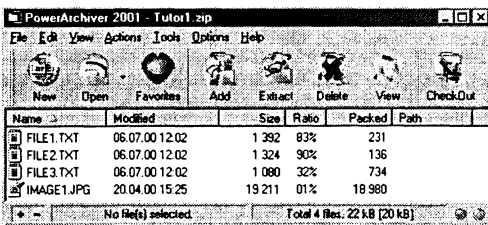
Для разархивирования выделить файл архива, в окне появится список входящих в архив файлов с указанием степени сжатия.



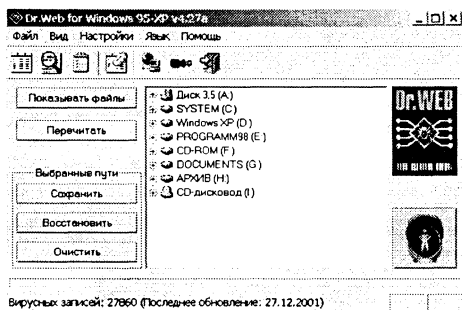
1.15. Ввести команду [Action-Extract...].

Запустить архиватор PowerArchiver.

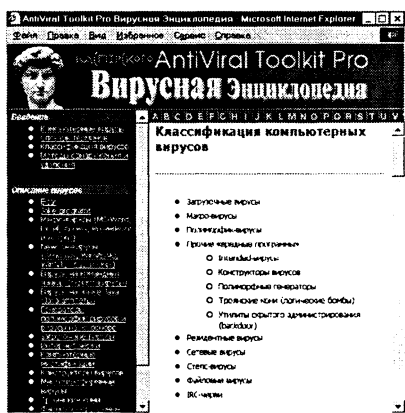
Для разархивирования выделить файл архива и ввести команду [Action-Extract...].



- 1.16. Запустить полифаг-сканер Dr.Web.  
Произвести настройки режима проверки и лечения. Ввести команду [Настройки-Изменить установки]. Ввести команду [Файл-Начать проверку].



- 1.17. Вирусная энциклопедия создана в форме Web-сайта, поэтому для работы с ней необходимо загрузить в браузер начальную страницу, находящуюся в файле index.htm. Для ознакомления с типами вирусов или конкретными вирусами необходимо активизировать соответствующие ссылки.

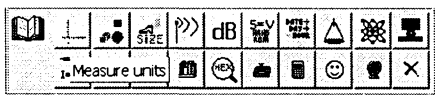


## Ответы к главе 2

### Кодирование информации. Системы счисления

- 2.1. 3 бита.  
2.2. 6 бит.  
2.3. 7 бит.  
2.4. 5 бит.  
2.5. а) 2 бита; б)  $\approx 1,9$  бит.  
2.6. 9 (5 для определения числа и 4 для определения месяца).  
2.7. а) 5 Кбайт = 5120 байт = 40 960 бит;  
б) 1,5 Кбайта = 1 536 байт = 12288 бит;  
в) 1 Кбайт =  $2^{10}$  байт =  $2^{13}$  бит;  
г) 1,5 Гбайта = 1536 Мбайт = 1 572 864 Кбайт;  
д) 512 Кбайт =  $2^{19}$  байт =  $2^{22}$  бит.

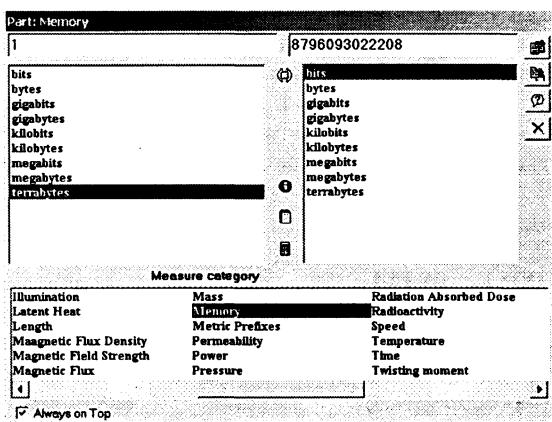
- 1 Запустить программу и щелкнуть по кнопке *Measure Units* (Единицы измерения).



- 2 В нижней части панели *Part: Memory* выбрать раздел *Memory*.

В левом верхнем текстовом окне ввести число и в левом списке выбрать единицу измерения.

В правом списке выбрать единицу измерения результата, в правом верхнем поле будет выведен результат.



- 2.8. а)  $x = 7$ ; б)  $x = 8$ .
- 2.9. 256 символов.
- 2.10. 8 бит  $\times$  100  $\approx$  100 байт.
- 2.11. 16 бит  $\times$  50  $\times$  40  $\times$  5  $\times$  4  $\approx$  78 Кбайт.
- 2.12. а) 6 тысяч; б) 6 сотен; в) 6 единиц; г) 6 десятков.
- 2.13. III (1+1+1=3) < 111.
- 2.14. а) M(1000)CM(1000-100)XC(100-10)IX(10-1)  $\rightarrow$  1999;  
б) 988;  
в) 1147.
- 2.16. Один из возможных способов решения:  
VI + V=XI                      XI-V=VI  
VI =IX-III                      VIII +II=X

2.17.

Система счисления	Осно- вание	Цифры
шестнадцатеричная	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
десятичная	10	0,1,2,3,4,5,6,7,8,9
восьмеричная	8	0,1,2,3,4,5,6,7
двоичная	2	0,1

2.18. Система счисления	Основание	Разряды (степени)				
десятичная	10	10000	1000	100	10	1
восьмеричная	8	4096	512	64	8	1
двоичная	2	16	8	4	2	1

2.19. е)  $A_{16}=1A3,5C1=1 \cdot 16^2 + A \cdot 16^1 + 3 \cdot 16^0 + 5 \cdot 16^{-1} + C \cdot 16^{-2} + 1 \cdot 16^{-3}$ .

2.20. б)  $A_{16}=A1,75$ .

2.21. а) нет; б) нет; в) да; г) нет.

2.22. Так как в записи одного из чисел есть цифра 7, то минимально возможное основание — 8.  $127_8=87_{10}$ ,  $222_8=146_{10}$ ,  $111_8=73_{10}$ .

2.23.  $10101_2=21_{10}$ ;  $10101_8=4161_{10}$ ;  $10101_{16}=65793_{10}$ .

2.24. Исходное число  $ab3=a \cdot 100 + b \cdot 10 + 3$ ,

новое число  $3ab=3 \cdot 100 + a \cdot 10 + b$ .

По условию  $3 \cdot 100 + a \cdot 10 + b = 3 \cdot (a \cdot 100 + b \cdot 10 + 3) + 1$ ,

$$3 \cdot 100 + a \cdot 10 + b = 3 \cdot a \cdot 100 + 3 \cdot b \cdot 10 + 10,$$

$$3 \cdot 100 + a \cdot 10 + b = 3 \cdot a \cdot 100 + (3 \cdot b + 1) \cdot 10 + 0.$$

Учитывая, что  $a$  и  $b$  — десятичные цифры, имеем:  $a=1$  и  $b=0$ .

Исходное число 103.

2.25.  $142857_{10}$ .

2.26. а)  $110011_2$ ; б)  $111_4$ .

2.27. Не существует, так как  $12_8 + 11_{16} = 11011_2$  ( $10 + 17 = 27$ ).

2.28.  $111_2 = 7_{10}$ ,  $777_8 = 191_{10}$ ,  $FFF_{16} = 4095_{10}$ .

2.29.  $2 \times 2 = 10_2 \times 10_2 = 100_2$ ;

$$6 \times 6 = 6_8 \times 6_8 = 44_8$$
;

$$4 \times 4 = 4_8 \times 4_8 = 20_8.$$

2.30. а) 45, 46, 47, 48; б) 12, 13, 14, 15, 16; в) 40, 41, ..., 47, 48.

2.31. 27 учеников.

2.32.  $q=8$ .

2.33. Составим уравнение:  $q^2 = 3 \cdot q + 3 + 2 \cdot q + 2 + q + 6 + 5$ ,  
 $q^2 - 6 \cdot q - 16 = 0$ ,  $q=8$ .

2.34.  $q=2$ .

2.35. Может быть, если все данные приведены в двоичной системе счисления.

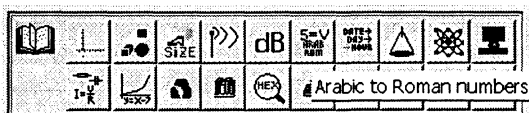
2.36. 9 дней, 512 листьев.

2.37. а) 101; г) 11001;

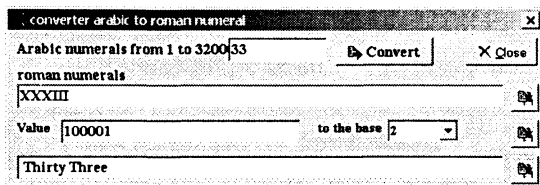
б) 111; д) 100000;

в) 1100; е) 100001.

1 Запустить программу и щелкнуть по кнопке *Arabic to Roman numbers*.



- 2 В верхнее поле ввести десятичное число и щелкнуть по кнопке *Convert*. В текстовом поле *Value* появится двоичное число.



2.38.

Двоичная	Восьмеричная	Десятичная	Шестнадцатеричная
101010	52	42	2A
1010111	127	87	57
100001101	415	269	10D
10011011	233	155	9B

2.39.

Двоичная	Восьмеричная	Десятичная	Шестнадцатеричная
0,101	0,5	0,625	0,A
0,11	0,6	0,75	0,C
0,001	0,1	0,125	0,2
0,01	0,2	0,25	0,4

2.40.

Двоичная	Восьмеричная	Десятичная	Шестнадцатеричная
111101,1	75,4	61,5	3D,8
10011011,101	233,5	155,625	9B,A
101110,1001	56,44	46,5625	2E,9
1011001,1011	131,52	89,65625	59,B

- 2.41. а)  $10111_2$ ; г)  $101_2$ ; ж)  $1111110_2$ ; к)  $101_2$ ;  
 б)  $112_8$ ; д)  $44_8$ ; з)  $2025_8$ ; л)  $3_8$ ;  
 в)  $146_{16}$ ; е)  $18_{16}$ ; и)  $6739_{16}$ ; м)  $3_{16}$ .

- 2.42. Переписав условие в десятичной системе. получим:  
 а)  $12 \times 3 - 4 = 32$ ; б)  $12 : 2 - 2 = 4$ ; в)  $12 \times 2 \times 2 = 48$ ; г)  $12 - 2 : 2 = 11$ ;  
 д)  $12 - 3 \times 4 = 0$ .

- 2.43. а)  $11_{10}$ ; б)  $700_8, 448_{10}$ ; в)  $B0_{16}, 176_{10}$ ; г)  $110_2, 6_{10}$ .

- 2.44. а)  $9_{10}$ ; б)  $55_8, 45_{10}$ ; в)  $99_{16}, 153_{10}$ ; г)  $101_2, 5_{10}$ .

- 2.45. а)  $101101_2, 101110_2, 101111_2, 110000_2$ ;

б)  $14_8, 15_8, 16_8, 17_8, 20_8$ ;

в)  $28_{16}, 29_{16}, 2A_{16}, 2B_{16}, 2C_{16}, 2D_{16}, 2E_{16}, 2F_{16}, 30_{16}$ .

- 2.46. а)  $10_{10}$ ; б)  $4000_{10}$ .

- 2.47. а)  $100_{10}$ ; б)  $100_{10}$ .



- 2.48. Сложив исходные числа столбиком, получим:  
 $1717171717_8 \rightarrow F3CF3CF_{16}$ . Пятая цифра слева равна 3.
- 2.49. а) 
$$\begin{array}{r} 5255 \\ +4327 \\ \hline 11604 \end{array}$$
      б) 
$$\begin{array}{r} 1536 \\ -642 \\ \hline 674 \end{array}$$
  
 восьмеричная      восьмеричная

2.51.

Десятичные числа	Прямой код	Обратный код	Дополнительный код
-10	0000000000001010	111111111110101	111111111110110
-100	0000000001100100	111111110011011	111111110011100
-1000	0000001111101000	111110000010111	111110000011000
-10000	0010011100010000	1101100011101111	1101100011110000

2.52.

Десятичные числа	Компьютерное представление	
	целые неотрицательные числа	целые числа со знаком
255	11111111	0000000011111111
-255	отсутствует	1111111100000001
32768	отсутствует	отсутствует
-32768	отсутствует	1000000000000000

2.53.

Компьютерное представление	Максимальное значение	Минимальное значение
целые неотрицательные числа	$2^8 - 1 = 255$	0
целые числа со знаком	$2^{15} - 1 = 32767$	$-2^{15} = -32768$
большое целое число со знаком	$2^{31} - 1 = 2147483647$	$-2^{31} = -2147483648$

- 2.54. Представим положительное число в прямом, а отрицательное число в дополнительном коде:

Десятичное число	Прямой код	Обратный код	Дополнительный код
20	000000000010100		
-60	000000000111100	111111111000011	111111111000011 +000000000000001 111111111000100

Сложим прямой код положительного числа с дополнительным кодом отрицательного числа. Получим результат в дополнительном коде:

20-60			111111111011000
-------	--	--	-----------------

Переведем полученный дополнительный код в десятичное число:

1. Инvertируем дополнительный код: 0000000000100111.
2. Прибавим к полученному коду 1 и получим модуль отрицательного числа:

$$\begin{array}{r} + 0000000000100111 \\ 0000000000000001 \\ \hline 0000000000101000 \end{array}$$

3. Переведем в десятичное число и припишем знак отрицательного числа:  $-40$ .

- 2.55. а)  $0,217934 \cdot 10^3$ ; б)  $0,75321 \cdot 10^5$ ; в)  $0,100101 \cdot 10^2$ ; г)  $0,200450 \cdot 10^6$ .

- 2.56. Максимальное значение порядка числа составит  $111111111_2 = 1023_{10}$ , и, следовательно, максимальное значение числа составит:

$$2^{1023} = 8,988465674311579538646525953945 \cdot 10^{307}$$

Точность вычислений определяется количеством разрядов, отведенных для хранения мантиссы чисел. Максимальное значение положительной мантиссы равно:

$$2^{52} - 1 \approx 2^{52} = 2^{(10 \cdot 5,2)} \approx 1000^{5,2} = 10^{(3 \cdot 5,2)} = 10^{15,6}$$

Таким образом, максимальное значение чисел двойной точности с учетом возможной точности вычислений составит  $8,98846567431157 \cdot 10^{307}$  (количество значащих цифр десятичного числа в данном случае ограничено 15–16 разрядами).

2.57.	Сложение	Вычитание	Умножение	Деление
	$0,10000 \cdot 2^2$	$0,10000 \cdot 2^2$	$0,10000 \cdot 2^2$	$0,10000 \cdot 2^2$
	$+ 0,00001 \cdot 2^2$	$- 0,00001 \cdot 2^2$	$\times 0,00001 \cdot 2^2$	$: 0,00001 \cdot 2^2$
	$0,10001 \cdot 2^2$	$0,01111 \cdot 2^2$	$0,000001 \cdot 2^4$	$= 10000 \cdot 2^0$
		$= 0,1111 \cdot 2^1$	$= 0,1 \cdot 2^{-1}$	$= 0,1 \cdot 2^5$

- 2.58. а) 50 61 73 73 77 6F 72 64;  
 б) 57 69 6E 64 6F 77 73;  
 в) 4E 6F 72 74 6F 6E 20 43 6F 6D 6D 61 6E 64 65 72.

- 2.59. а) Tornado;  
 б) I love you;  
 в)  $2 \cdot X + Y = 0$ .

- 2.60. а) 55 70 20 26 20 44 6F 77 6E — Up & Down;  
 б) 49 42 4D — IBM;  
 в) 45 6E 74 65 72 — Enter.

- 2.61. а) 57 6F 72 64 — Word;  
 б) 44 4F 53 — DOS;  
 в) 50 61 69 6E 74 62 72 75 73 68 — Paintbrush.

- 2.62. КОИ8-Р: E2 E9 F3;  
 CP1251: C1 C8 D1;

CP866: 81 88 91;  
 Mac: 81 88 91;  
 ISO: V1 V8 C1.

2.63. КОИ8-Р: ДХЯЙ;

CP866: фшёь;

Mac: диск;

ISO: фшёь.

2.64.  $25 \times 80 = 2000$  байт.

2.65. В два раза.

2.66. 11 байт, 22 байта.

2.67. 16, 256, 65536, 16777216, 4294967296 цветов.

2.68. 100 бит.

2.69. 100 байт.

2.70. В 4 раза.

2.71. В 8 раз.

2.72. 120.

2.73. 256 цветов.

2.74. а) 0,75; б) 0,75; в) 0,75; г) 0,75; д) 0,8.

2.75.  $1280 \times 1024$ .

2.76.

Разрешающая способность экрана	Глубина цвета (бит на точку)				
	4	8	16	24	32
640×480	150 Кбайт	300 Кбайт	600 Кбайт	900 Кбайт	1,2 Мбайта
800×600	234 Кбайта	469 Кбайт	938 Кбайт	1,4 Мбайта	1,8 Мбайта
1024×768	384 Кбайта	768 Кбайт	1,2 Мбайта	2,25 Мбайта	3 Мбайта
1280×1024	640 Кбайт	1,25 Мбайта	2,5 Мбайта	3,75 Мбайта	5 Мбайт

2.77. Достаточно.

2.78.  $640 \times 480$  (4, 8, 16, 24 бит),  $800 \times 600$  (4, 8, 16 бит),  $1024 \times 768$  (4, 8 бит),  $1280 \times 1024$  (4 бита).

2.80.

Название цвета	Интенсивность		
	Красный	Зеленый	Синий
Черный	00	00	00
Красный	FF	00	00
Зеленый	00	FF	00
Синий	00	00	FF
Белый	FF	FF	FF

2.81.  $\approx 398$  Мбайт.

2.82. а) 940 Кбайт;

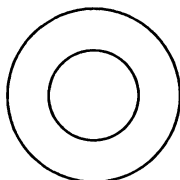
б) 2,8 Мбайта.

- 2.83. а) качество CD; б) качество радиотрансляции.
- 2.84. а) 10 с;  
б) 1,5 мин.
- 2.85. а)  $\approx 3$  минуты;  
б)  $\approx 7,6$  секунды.
- 2.86. В 2 раза.
- 2.87. а) на гибких и жестких магнитных дисках;  
б) на жестких магнитных дисках;  
в) на оптических дисках.
- 2.88.  $64 \text{ байта} \cdot 32 \cdot 4 = 8192 \text{ байта} = 8 \text{ Кбайт}$ .
- 2.89. Каждая графическая страница содержит  $4 \text{ бита} \cdot 320 \cdot 640 = 819200 \text{ бит} = 100 \text{ Кбайт}$  информации, каждая текстовая —  $64 \text{ байта} \cdot 32 = 2048 \text{ байт} = 2 \text{ Кбайта}$ . Пусть  $X$  — число страниц с графикой, тогда  $8X$  — число страниц с текстом. Составим уравнение:  $100X + 16X = 20480$ . Решив уравнение, получим  $X = 176$ . Таким образом, на жестком магнитном диске объемом 20 Мб можно разместить 1584 страницы энциклопедии (176 графических и 1408 текстовых).
- 2.90. а) 2847; б) 2847; в) 949.
- 2.91. Нет. На дискету можно записать только 86 файлов указанного размера.
- 2.92. Можно.
- 2.93. Одинаковое количество файлов — 65536, так как текстовый файл будет занимать на диске один кластер, а количество кластеров на дисках при использовании FAT16 одинаково.
- 2.94. Тем большее количество файлов, чем больше объем диска, так как объем кластера при использовании FAT32 постоянен:  
а) 51200; б) 524288; в) 5242880.

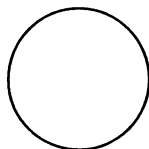
## Ответы к главе 3

### Основы логики и логические основы компьютера

3.1.



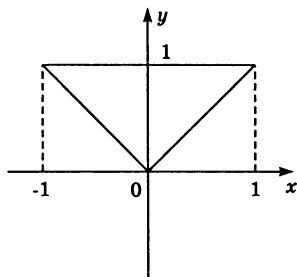
а)



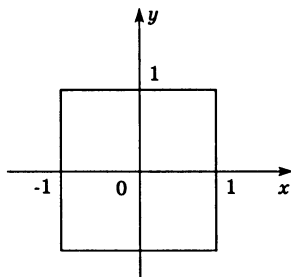
б)

- 3.2. Субъекты: «Сканер», «Луна», «Атом».  
 Предикаты: «устройство ввода информации», «спутником Земли», «ядра и электронов».  
 Связки: «это», «является», «состоит из».
- 3.4. а)  $X < -3$  или  $X > 8$ ;  
 б)  $-3 \leq X \leq 8$ .
- 3.5. б)  $X < A$  или  $X > B$ ;      в) не ( $X \leq B$  или  $X \geq A$ ).

3.6.



а)



б)

- 3.7. 1)  $A = \{\text{Число } 376 \text{ четное}\}$ ,  $B = \{\text{Число } 376 \text{ трехзначное}\}$ ,  $A \& B$ .  
 2)  $A = \{\text{Солнце движется вокруг Земли}\}$ ,  $\bar{A}$ .  
 3)  $A = \{\text{Сумма цифр числа делится на } 3\}$ ,  $B = \{\text{Число делится на } 3\}$ ,  $A \Rightarrow B$ .  
 4)  $A = \{\text{Число } 15 \text{ делится на } 3\}$ ,  $B = \{\text{Сумма цифр числа } 15 \text{ делится на } 3\}$ ,  
 $A \Leftrightarrow B$ .

3.8.

В естественном языке	В логике
... и ...	$\&$
... или ...	$\vee$
Неверно, что ...	$\bar{A}$
... в том и только в том случае ...	$\Leftrightarrow$
... если ..., то ...	$\Rightarrow$
... тогда и только тогда, когда ...	$\Leftrightarrow$
... не ...	$\bar{A}$

- 3.9. а) 1; б) 1; в) 1; г) 0; д) 1; е) 1; ж) 1; з) 0; и) 0.
- 3.10. б, г.
- 3.11. а) 1; б) 1.
- 3.12.  $A = 0$ ;  $B = 1$ ;  $C = 0$ ;  $D = 1$ .  
 а) 0; б) 1; в) 0; г) 0.
- 3.13.  $A = 0$ ,  $B = 1$ .
- 3.14. а) 110; б) 1010; в) 111111.
- 3.15. 11011101, 1111101110010, 10101111, 1011100.

3.18. 2) (A или B) и (A или C).

- 3.20. а)  $F_3(X, Y) = X \& \bar{Y}$ ; б)  $F_5(X, Y) = \bar{X} \& Y$ ;  
 в)  $F_7(X, Y) = \bar{X} \& Y \vee X \& \bar{Y}$ ; г)  $F_{10}(X, Y) = \bar{X} \& \bar{Y} \vee X \& Y$ ;  
 д)  $F_{11}(X, Y) = \bar{Y}$ ; е)  $F_{12}(X, Y) = \bar{X} \vee Y$ ;  
 ж)  $F_{13}(X, Y) = \bar{X}$ ; з)  $F_{14}(X, Y) = X \vee \bar{Y}$ .

3.21. Лист *Логические функции* в файле log.sdc (на CD-ROM хранится в каталоге \practicum\inftech\logica).

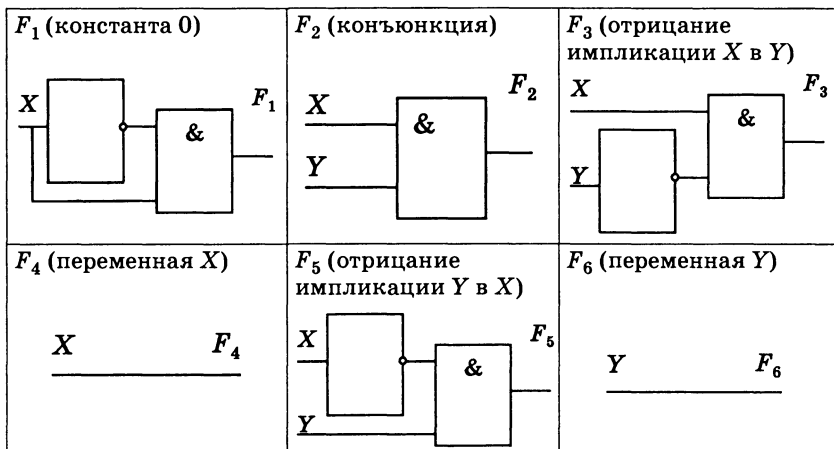
	X	Y	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1																		
2	0	0	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
3	0	1	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
4	1	0	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE
5	1	1	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE

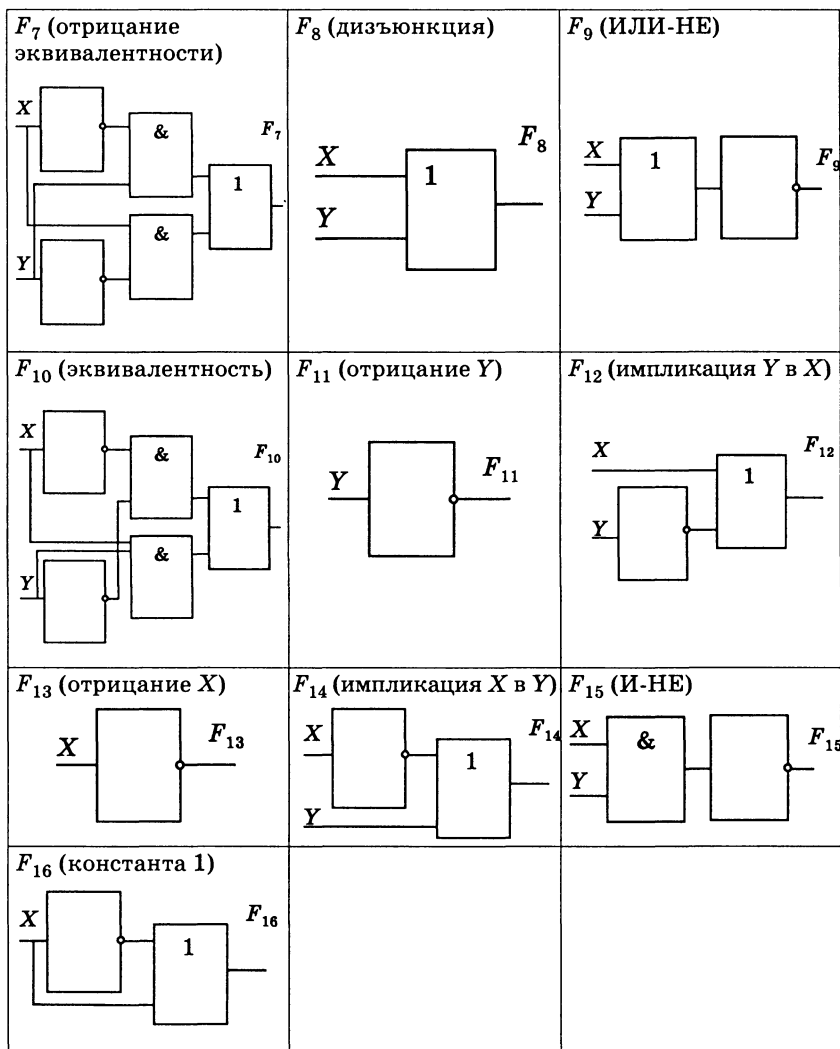
3.22. 2).

- 3.23. а) переместительному;  
 б) сочетательному;  
 в) аналога нет;  
 г) распределительному.

- 3.26. а)  $A \vee (A \& B) = (A \vee A) \& (A \vee B) = 1 \& (A \vee B) = A \vee B$ ;  
 б)  $A \& (A \vee B) = (A \& A) \vee (A \& B) = 1 \vee (A \& B) = A \& B$ ;  
 в)  $(A \vee B) \& (B \vee A) \& (C \vee B) = A \& (B \vee B) \& (C \vee B) = A \& 1 \& (C \vee B) = A \& (C \vee B)$ .

3.27.





3.28. Таблица сложения одnorазрядных двоичных чисел  $X$  и  $Y$  с учетом переноса в старший разряд выглядит следующим образом:

Слагаемые		Перенос	Сумма
$X$	$Y$	$P$	$S$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Из этой таблицы сразу видно, что перенос можно реализовать с помощью операции логического умножения:

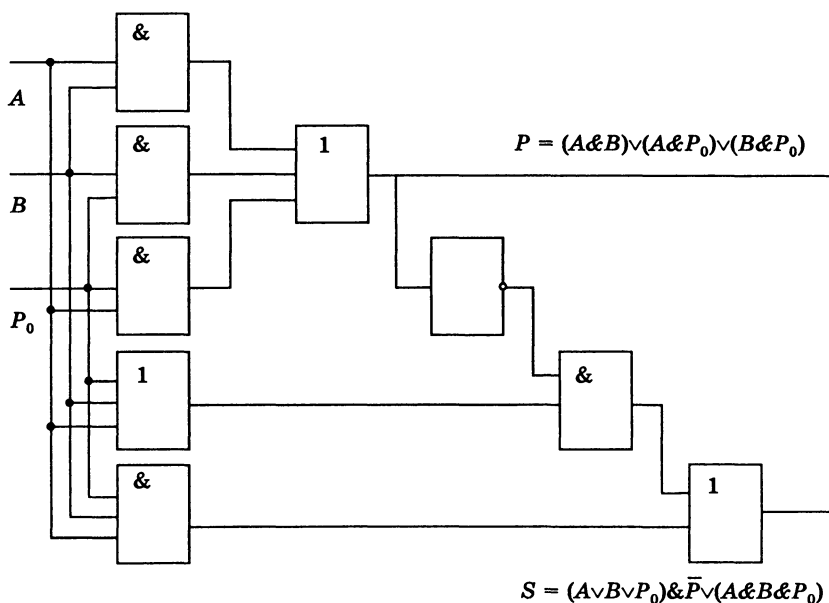
$$P = F_1(X, Y) = X \& Y.$$

Получим теперь формулу для вычисления суммы. Значения суммы более всего совпадают с результатом операции логического сложения (кроме случая, когда на входы подаются две единицы, а на выходе должен получиться нуль).

Нужный результат достигается, если результат логического сложения умножить на инвертированный перенос. Таким образом, для определения суммы можно применить следующее логическое выражение:

$$S = F_2(X, Y) = (X \vee Y) \& (\overline{X \& Y}).$$

3.31.



3.32. Для построения одноразрядного сумматора двоичных чисел необходимо 9 базовых логических элементов. Следовательно, для построения 64-разрядного необходимо  $9 \cdot 64 = 576$  элементов.

3.33. Количество базовых логических элементов в триггере необходимо умножить на количество бит в ячейке оперативной памяти и умножить на количество ячеек:  
 $4 \cdot 8 \cdot 64 \cdot 1024 \cdot 1024 = 2\,147\,483\,648.$

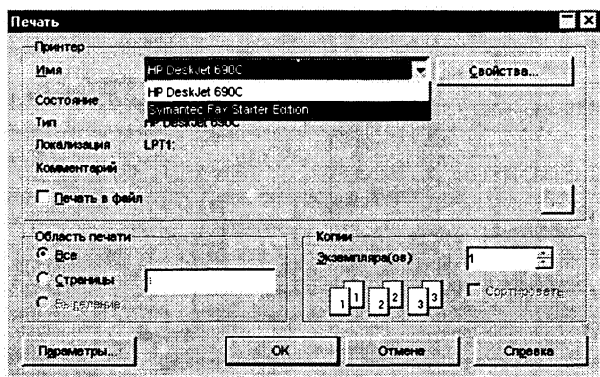


## Ответы к главе 4

### Информационные технологии

#### 4.1. Создание и отправка факса.

- 1 Ввести команду [Создать-Автопилот-Факс...] и в процессе создания собственного шаблона факса следовать указаниям диалоговых окон.
- 2 Ввести команду [Сохранить как...] и выбрать формат сохранения документа. Отредактировать документ.
- 3 Если к компьютеру подключен модем, то отправить факс непосредственно из StarOffice Writer.
- 4 Ввести команду [Файл-Печать...]. На появившейся панели *Печать* в раскрывающемся списке *Имя* выбрать *Symantec Fax Starter Edition* и щелкнуть по кнопке *OK*. Начнется процедура отправки факса.



#### 4.2. Создание расписание уроков на основе шаблона *Расписание занятий*.

- 1 Ввести команду [Файл-Создать-Из шаблонов...].
  - 2 На панели *Создать* в окне *Категории*: выбрать набор шаблонов (*Education*), а в окне *Шаблоны* выбрать конкретный шаблон (*Расписание занятий*).
  - 3 Сохранить шаблон как документ и отредактировать его.
- #### 4.3. Создание вкладышей для кассеты и компакт-диска и наклейки для дискеты на основе шаблонов.
- 1 Ввести команду [Файл-Создать-Из шаблонов...].
  - 2 На панели *Создать* в окне *Категории*: выбрать набор шаблонов (*Miscellaneous*), а в окне *Шаблоны* последовательно выбрать шаблоны (*Кассета*, *Компакт-диски* и *Наклейка для дискет*).
  - 3 Сохранить шаблоны как документы и отредактировать их.

4.4. Открытие и форматирование текстового документа.

1 Открыть файл text1.txt и отформатировать текст документа.

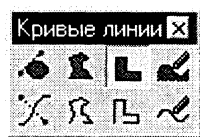
4.5. Получение изображения значка *Мой компьютер*.

1 Повторить последовательность действий, подробно описанную в практическом задании 4.4.

4.7. Построение блок-схем «ветвление».

4.8. Построение блок-схем «цикл».

1 Для рисования элементов блок-схем «условие» и «счетчик» использовать отрывную панель инструментов *Кривые линии* в StarOffice Draw.



4.9. Рисование структурной схемы компьютера.

1 Для отображения потоков информации между отдельными устройствами компьютера использовать отрывную панель инструментов *Соединительная ломаная линия* в StarOffice Draw.



4.10. Файл презентации «Путешествие во времени» Tour.ppt находится в каталоге \practicum\inftech\inpress\.

Можно предложить следующее содержание презентации путешествия во времени:

№ Содержание слайда

1 *Хронотур «Вчерашний XX век»*

Маршрут фирмы  
«Вперед в прошлое»

2 *История транспорта*

На слайд поместить изображения автомобиля, поезда, самолета и космического корабля.

3 *Музей компьютеров*

На слайд поместить фотографии компьютеров.

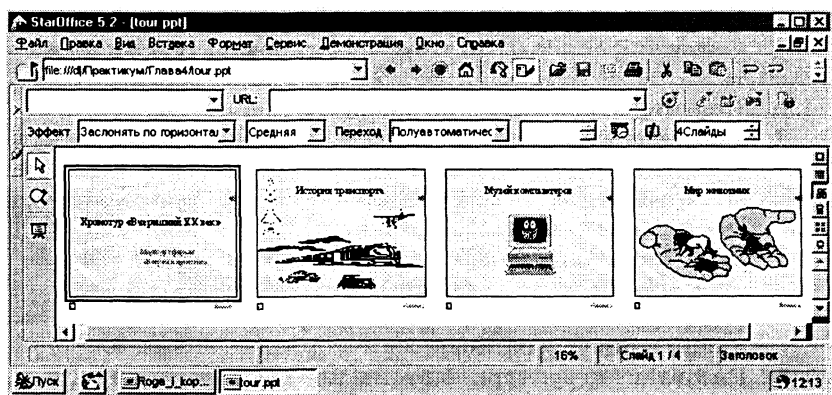
4 *Мир животных*

На слайд поместить несколько изображений животных.

Эффекты

Использовать эффекты анимации как при переходах слайдов, так и при появлении объектов на слайдах.

Установить режим непрерывной (циклической) демонстрации презентации.



- 4.11. Файл презентации «Рога и копыта» *Roga\_and\_kopyta.ppt* находится в каталоге `\practicum\inftech\inpress\`. Можно предложить следующее содержание презентации к выступлению Остапа Бендера:

#### № Содержание слайда

1 *Черноморское отделение Арбатовской конторы ПО ЗАГОТОВКЕ РОГОВ И КОПЫТ*

2 *Структура ТОО «Рога и копыта»*

Включает в себя Арбатовскую контору, Черноморское отделение и филиалы.

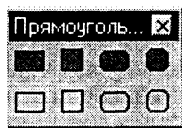
3 *Организационная диаграмма отделения*  
Описывает кадровую структуру фирмы во главе с О. Бендером.

4 *Доходы и расходы отделения*

Доходы (взнос Корейко и кредит) и многочисленные расходы (зарплата, командировочные и так далее).

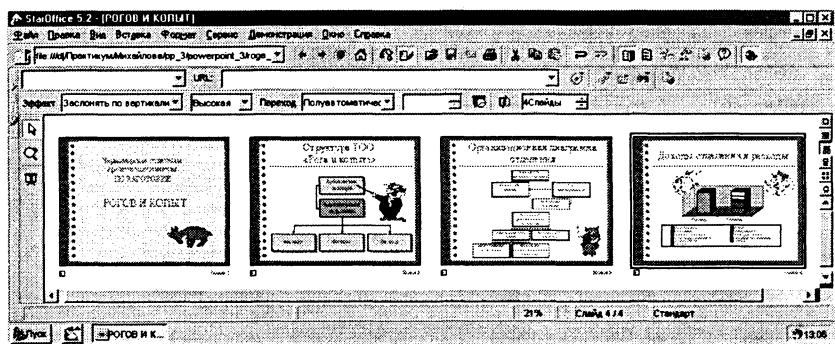
#### Эффекты

На слайдах 2 и 3 с помощью панелей инструментов *Прямоугольники*



и *Соединительная ломаная линия* нарисовать структурные схемы.

В электронных таблицах StarOffice Calc создать диаграммы доходов и расходов. Создать слайд с разметкой *Диаграмма* и поместить диаграммы на слайд с помощью команды [Вставка-Диаграмма].



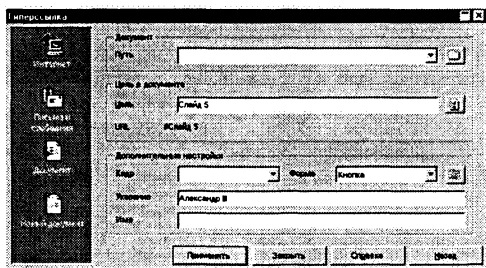
4.12. Файл презентации «Три Александра» Tri\_Aleksandra.ppt находится в каталоге \practicum\inftech\inpress\.

Можно предложить следующее содержание презентации к докладу на конференции по истории России XIX века:

№	Содержание слайда	Эффекты
1	Три Александра	

Для вставки гиперссылки необходимо ввести команду [Вставка-Гиперссылка]. Появится диалоговая панель *Гиперссылка*, на которой необходимо указать форму указателя ссылки, ее цель и др.

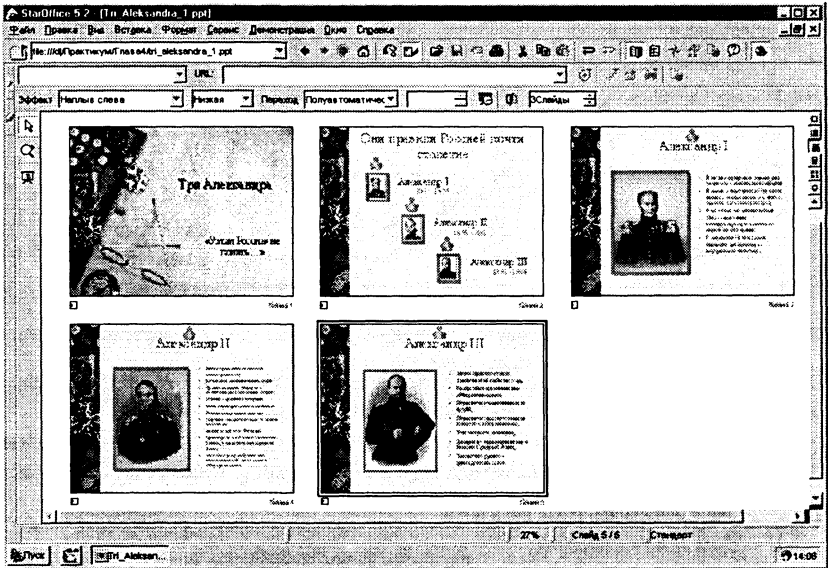
- |   |  |
|---|--|
| 2 | <p><i>Они правили<br/>Россией почти<br/>столетие</i></p> <p>Александр I<br/>1801–1825</p> <p>Александр II<br/>1855–1881</p> <p>Александр III<br/>1881–1894</p> |
|---|--|



Второй слайд является центральным слайдом презентации, с него возможны переходы по гиперссылкам, размещенным на кнопках, на слайды 3, 4 и 5.

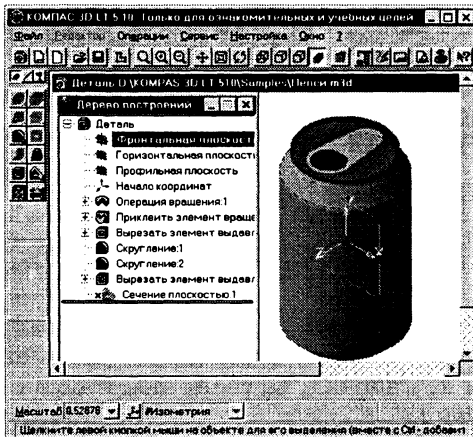
- |   |               |
|---|---------------|
| 3 | Александр I   |
| 4 | Александр II  |
| 5 | Александр III |

На слайдах размещаются портреты императоров, годы царствования и краткие сведения об их деятельности.



#### 4.13. Ознакомление с возможностями САПР КОМПАС-3D.

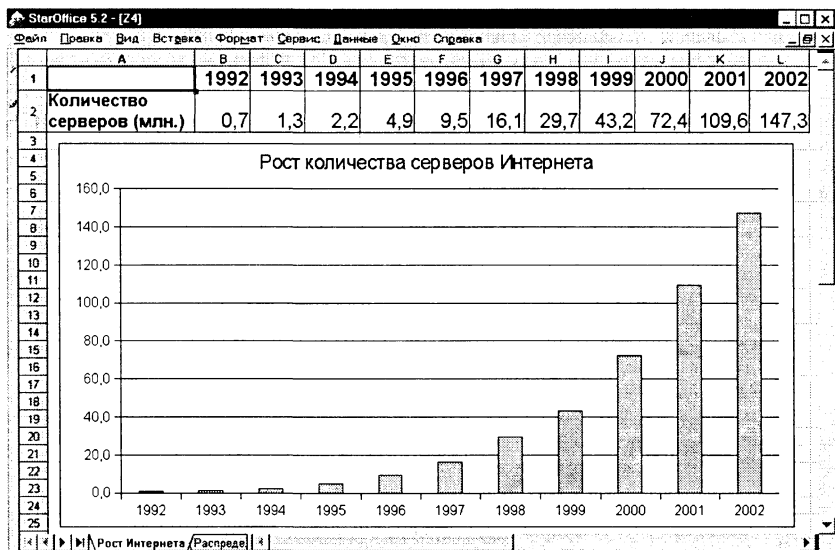
- 1 Запустить КОМПАС-3D. Щелкнуть по кнопке *Новая деталь*. Ввести команду [Файл-Открыть...]. С помощью стандартного диалогового окна *Выберите файл для открытия* выбрать файл, например, пепси.m3d.



#### 4.14. Диаграмма роста количества серверов Интернета (лист *Рост Интернета* в файле *Z.xls*, который находится в каталоге `practicum\inftech\calc\`).

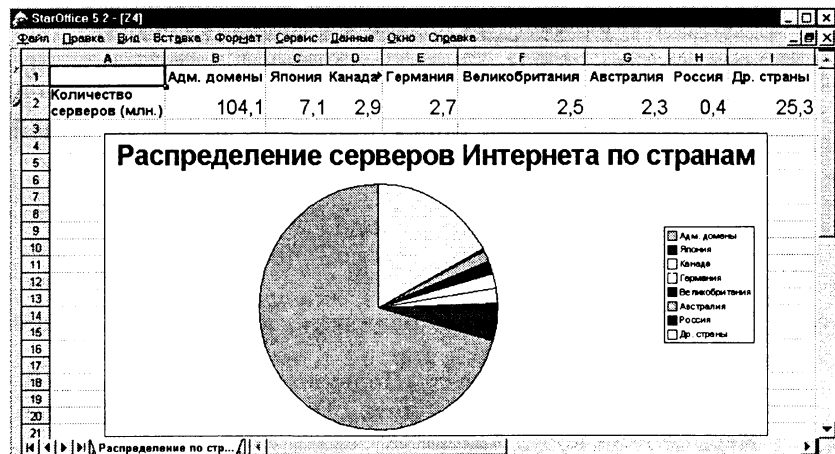
- 1 Ввести данные о количестве серверов Интернета по годам (смотри данные по адресу <http://www.isc.com>).

- Ввести команду [Вставка-Диаграмма...]. На панелях Автоформат диаграммы выбрать Ряд данных в: Строки и тип диаграммы Гистограмма.



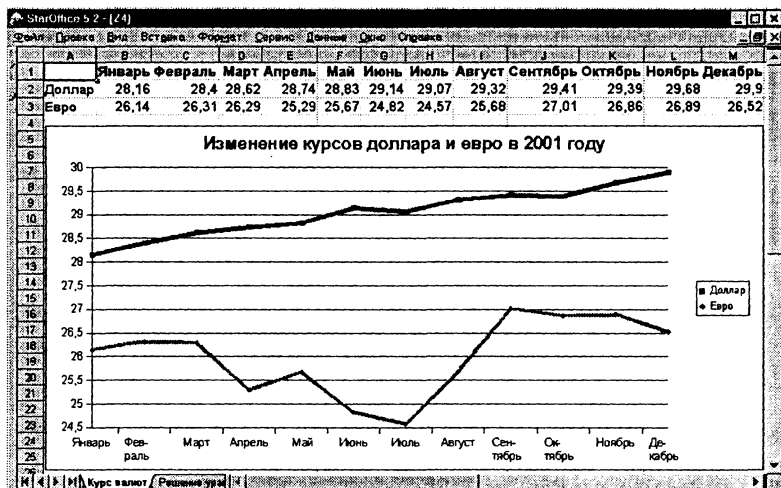
4.15. Диаграмма распределения серверов Интернета по странам (лист Распределение по странам в файле Z.xls).

- Ввести данные о распределении серверов Интернета по странам (смотри данные по адресу <http://www.isc.com>).
- Ввести команду [Вставка-Диаграмма...]. На панелях Автоформат диаграммы выбрать Ряд данных в: Строки и тип диаграммы Круговая.



4.16. Графики изменения курсов доллара и евро к рублю за 2001 год (лист *Курсы валют* в файле *Z.xls*).

- 1 Найти в Интернете и ввести данные об изменении курсов валют по месяцам.
- 2 Ввести команду [Вставка-Диаграмма...]. На панелях *Автоформат диаграммы* выбрать *Ряд данных в: Строки* и тип диаграммы *Линии*.



## Ответы к главе 5

### Объектно-ориентированное программирование на языке Visual Basic

Готовые проекты хранятся в каталоге  
practicum\VB\Projects\ в отдельных папках



5.1. Проект «Перемещение кнопок по форме» хранится в папке *rgjZ5\_1*.

- 1 Изменить код событийных процедур. Код событийной процедуры `cmd3_Click()`:

```
Private Sub cmd3_Click()
cmd3.Left = frm1.Width - cmd3.Width
cmd3.Top = frm1.Height - cmd3.Height
End Sub
```

5.2. Проект «Перемещение надписи» хранится в папке *rgjZ5\_2*.

- 1 Модернизировать проект «Перемещение кнопок по форме». Поместить на форму метку `lbl1` с надписью, например, *Перемещение надписи*.

- 2 Изменить код событийных процедур. Код событийной процедуры cmd3\_Click():

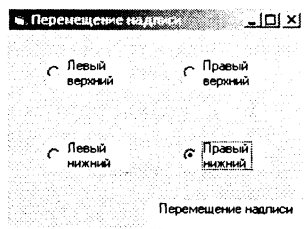
```
Private Sub cmd3_Click()
    lbl1.Left = frm1.Width - lbl1.Width
    lbl1.Top = frm1.Height - lbl1.Height
End Sub
```

- 5.3. Проект «Перемещение надписи-2» хранится в папке prjZ5\_3.

- 1 Модернизировать проект «Перемещение надписи». Удалить кнопки и поместить на форму четыре переключателя (OptionButton). Присвоить им имена opt1, opt2, opt3 и opt4.

- 2 Создать для каждого из переключателей событийную процедуру. Код событийной процедуры opt3\_Click():

```
Private Sub opt3_Click()
    lbl1.Left = frm1.Width -
    lbl1.Width
    lbl1.Top = frm1.Height -
    lbl1.Height
End Sub
```



- 5.4. Проект «Перемещение надписи-3» хранится в папке prjZ5\_4.

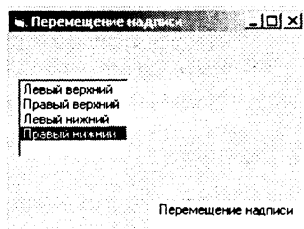
- 1 Модернизировать проект «Перемещение надписи-2». Удалить переключатели и поместить на форму список (ListBox).

- 2 Активизировать список lst1 и с помощью окна *Свойства объекта* установить для свойства List перечень значений:

- Левый верхний;
- Правый верхний;
- Левый нижний;
- Правый нижний.

- 3 Создать событийную процедуру lst1\_DblClick(), которая обеспечит перемещение надписи с помощью двойного щелчка по элементу списка. Для реализации перемещений в различные углы формы использовать конструкцию выбора **Select Case**:

```
Private Sub lst1_DblClick()
    Select Case lst1.Text
```





```

Case Is = "Левый верхний"
lb11.Top = 0
lb11.Left = 0
Case Is = "Правый верхний"
lb11.Left = frm1.Width - lb11.Width
lb11.Top = 0
Case Is = "Левый нижний"
lb11.Left = 0
lb11.Top = frm1.Height - lb11.Height
Case Is = "Правый нижний"
lb11.Left = frm1.Width - lb11.Width
lb11.Top = frm1.Height - lb11.Height
End Select
End Sub

```

5.5. Проект «Перемещение формы на экране» хранится в папке prjZ5\_5.vbp.

- 1 В углах формы разместить четыре кнопки.
- 2 Присвоить свойству Name для каждой кнопки следующие значения:

```

cmdUpLeft      — для левой верхней;
cmdDownLeft    — для левой нижней;
cmdUpRight     — для правой верхней;
cmdDownRight   — для правой нижней.

```

- 3 Использовать процедуру Form\_Load() для установки значений свойства Caption:

```

Private Sub Form_Load()
cmdDownLeft.Caption = "Левый нижний угол"
cmdDownRight.Caption = "Правый нижний угол"
cmdUpRight.Caption = "Правый верхний угол"
cmdUpLeft.Caption = "Левый верхний угол"
End Sub

```

- 4 Создать процедуры обработки событий Click для перемещения формы на экране монитора для каждой командной кнопки.

Так, при щелчке по кнопке cmdDownLeft необходимо, чтобы нижний край формы совместился с нижним краем экрана. Тогда свойство Top формы будет равно высоте экрана минус высота формы, то есть

```
frm1.Top = Screen.Height - frm1.Height
```

Свойство Left формы должно быть равно координате левого края экрана, то есть

```
frm1.Left = 0
```

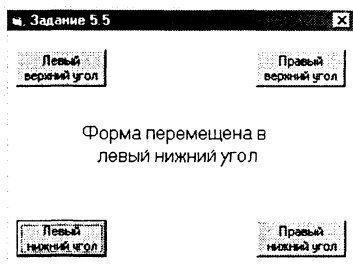
- 5 Для вывода текста на метке необходимо присвоить свойству Caption соответствующее значение текста:

```
lblVar.Caption="Форма перемещена в левый
нижний угол"
```

- 6 Код обработки события Click для кнопки cmdDownLeft будет выглядеть следующим образом:

```
Private Sub cmdDownLeft_Click()
frm1.Top = Screen.Height - frm1.Height
frm1.Left = 0
lblVar.Caption="Форма перемещена в левый
нижний угол"
End Sub
```

- 7 Составьте коды обработки события Click для других кнопок самостоятельно.



5.6. Проект хранится в папке prjZ5\_6.vbp.

- 1 Открыть проект prjZ5\_5.vbp. Для этого ввести команду [File-Open Project...] и с помощью диалоговой панели *Open Project* выбрать проект.
- 2 Изменить код событийных процедур для каждой командной кнопки. Код процедуры для кнопки cmdDownLeft будет иметь следующий вид:

```
Private Sub cmdDownLeft_Click()
frm1.Move 0,Screen.Height - frm1.Height
End Sub
```

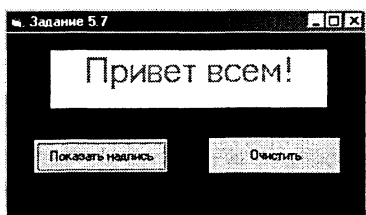
5.7. Проект «Видимость метки» хранится в папке prjZ5\_7.vbp.

- 1 Поместить на форму метку и две кнопки.
- 2 В событийную процедуру Click для первой кнопки включить строку:

```
lbl1.Visible = True
```

- 3 В событийную процедуру Click для второй кнопки включить строку:

```
lbl1.Visible = False
```



5.8. Проект «Видимость формы» хранится в папке `prjZ5_8.vbp`.

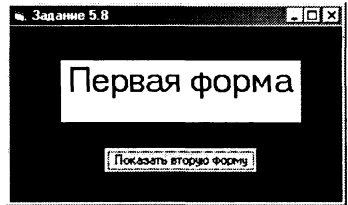
1 Добавить в проект вторую форму командой [Project-Add form].

2 В событийную процедуру щелчка по кнопке на первой форме включить строки:

```
frm1.Visible = False
frm2.Visible = True
```

3 В событийную процедуру щелчка по кнопке на второй форме включить строки:

```
frm1.Visible = True
frm2.Visible = False
```



5.9. Проект хранится в папке `prjZ5_9.vbp`.

1 Открыть проект `prj2.vbp`. Разместить на каждой форме по две кнопки и связать их с событийными процедурами перехода на другие формы.

2 Программный код для третьей формы:

```
Private Sub
cmd1_Click()
frm1.Visible = True
frm2.Visible = False
frm3.Visible = False
End Sub
Private Sub
cmd2_Click()
frm1.Visible = False
frm2.Visible = True
frm3.Visible = False
End Sub
```



5.10. Проект хранится в папке `prjZ5_10.vbp`.

1 Открыть проект `prjZ5_9.vbp`. Разместить на второй форме кнопку и ввести событийную процедуру для вывода изображения:

```
Private Sub cmdScan_Click()
img1.Picture = LoadPicture ("E:\practicum\
VB\Projects\project2\scanner.jpg")
End Sub
```

2 Разместить на третьей форме кнопку и ввести событийную процедуру для вывода изображения и изменения надписи на метке:

```

Private Sub cmdCam_Click()
pic1.Picture = LoadPicture
("E:\practicum\VB\Projects\project2\camera.jpg")
lbl1.Caption = "Цифровая камера"
End Sub

```



5.11. Проект «Собери картинку» хранится в папке `prjZ5_11.vbp`.

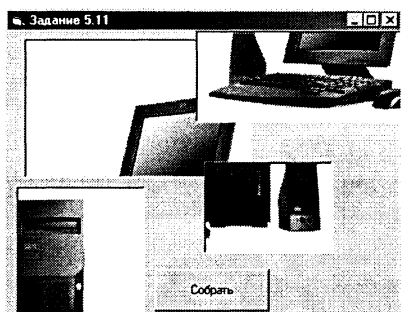
- 1 С помощью графического редактора (например, StarOffice Image) изображение (например, компьютера) произвольно разделить на несколько прямоугольников (например, четыре) и сохранить в отдельных файлах.
- 2 Поместить на форму произвольным образом четыре элемента `PictureBox` произвольного размера. Присвоить значению свойства `AutoSize` значение **True**.
- 3 Создать событийную процедуру `Load_Form`, которая помещает фрагменты изображения в графические окна:

```

Private Sub Form_Load()
pic1.Picture = LoadPicture
("E:\practicum\VB\Projects\Z5_11\comp1.gif")
pic2.Picture = LoadPicture
("E:\practicum\VB\Projects\Z5_11\comp2.gif")
pic3.Picture = LoadPicture
("E:\practicum\VB\Projects\Z5_11\comp3.gif")
pic4.Picture = LoadPicture
("E:\practicum\VB\Projects\Z5_11\comp4.gif")
End Sub

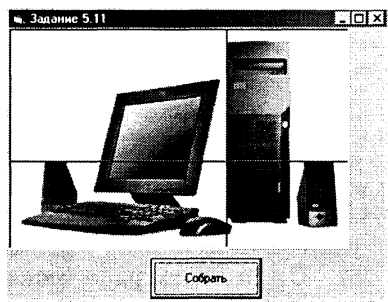
```

- 4 Запустить проект на выполнение. Фрагменты изображения беспорядочно размещены на форме.



- 5 Поместить на форму кнопку и создать для нее событийную процедуру, которая задает такие координаты для каждого графического окна, чтобы получилась целое первоначальное изображение.

Запустить проект на выполнение и щелкнуть по кнопке *Собрать*.



### 5.12. Проект «Максимальный элемент» хранится в папке rjZ5\_12.vbp.

- 1 Создать проект, в котором использовать для заполнения массива событийную процедуру из проекта «Сортировка числового массива».
- 2 Создать событийную процедуру поиска максимального элемента в массиве:

```
Private Sub cmdMax_Click()
    bytMax = bytA(1)
    bytN = 1
    For bytI = 2 To 10
        If bytA(bytI) > bytMax Then bytMax = _
            bytA(bytI): bytN = bytI
    Next bytI
    Print "Максимальный элемент "; bytMax; "
    его индекс"; bytN
End Sub
```

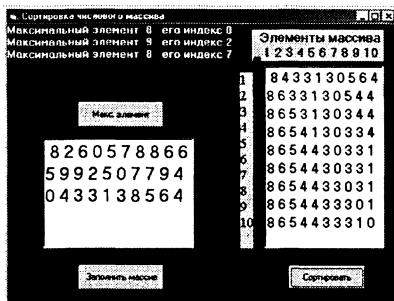
- 3 Запустить проект.

Последовательно щелкнуть по кнопкам *Заполнить массив* и *Макс. элемент*. Процедуру можно повторить несколько раз.



5.13. Проект хранится в папке prjZ5\_13.vbp.

- 1 Открыть файл prj3.vbp и изменить его так, чтобы производилась сортировка не по возрастанию, а по убыванию. Использовать в качестве общей процедуры процедуру поиска максимального элемента.



5.14. Проект «Пузырьковая сортировка» хранится в папке prjZ5\_14.vbp.

- 1 Поместить на форму две кнопки cmdDim и cmdSort и два текстовых поля txtDim и txtSort.
- 2 Событийную процедуру заполнения массива случайными числами и вывода значений элементов массива в текстовое окно cmdDim взять из проекта 5.3.
- 3 «Пузырьковый» метод основан на последовательном перемещении больших элементов к верхнему краю массива (подобно тому, как всплывают пузырьки воздуха). В массиве последовательно просматриваются элементы с 1-го до предпоследнего. При этом для каждого  $i$ -го элемента массив просматривается от  $(i+1)$ -го элемента до последнего. Если данный  $(i$ -й) элемент окажется больше одного из следующих за ним, они меняются местами. Вывод отсортированного массива осуществить в текстовое окно txtSort.

Программный код событийной процедуры сортировки:

```
Private Sub cmdSort_Click()
For bytI = 1 To 10
For bytJ = bytI + 1 To 10
If bytA(bytI) > bytA(bytJ) Then bytR =
bytA(bytI): bytA(bytI) = bytA(bytJ):
```

```

bytA(bytJ) = bytR
Next bytJ
txtSort.Text = txtSort.Text +
Str(bytA(bytI))
Next bytI
End Sub

```

- 4 Запустить проект. Последовательно щелкнуть по кнопкам *Заполнить массив* и *Сортировать*. Такую процедуру выполнить несколько раз.



- 5.15. Проект хранится в папке prjZ5\_15.vbp.

- 1 Открыть файл проекта prj4.vbp. Поместить на форму кнопку cmdSave и создать для нее событийную процедуру сохранения отсортированного массива strC в файле sort.txt. Для записи данных в файл, например на диск A:, использовать оператор **Write**:

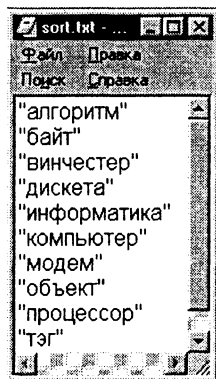
```

Private Sub cmdSave_Click()
Open "A:\sort.txt" For Output As #3
For bytI = 1 To 10
Write #3, strC(bytI)
Next bytI
Close #3
End Sub

```

- 2 Запустить проект на выполнение и последовательно щелкнуть по кнопкам: *Считать алфавит*, *Считать термины*, *Сортировать*, *Показать*, *Сохранить в файле*.

Будет создан текстовый файл, содержащий значения элементов отсортированного строкового массива. Замените оператор **Write** на оператор **Print**. Сравните полученные файлы данных между собой.



5.16. Проект «Словарь компьютерных терминов» хранится в папке prjZ5\_16.vbp.

- 1 Создать текстовые файлы, содержащие компьютерные термины (term.txt) и их определения (description.txt). Удобно в списках терминов и определений каждый элемент данных начинать с новой строки (в этом случае в качестве разделителя данных вместо запятой будет использоваться перевод строки). Порядки следования терминов и их определений должны быть одинаковыми.
- 2 Поместить на форму кнопки cmdTerm, cmdDesc и cmdVoc и два текстовых поля txtTerm (для ввода терминов) и txtDesc (для вывода определений).
- 3 Создать событийные процедуры cmdTerm\_Click() и cmdDesc\_Click(), которые обеспечивают считывание данных из файлов в строковые массивы.

Создать событийную процедуру cmdVoc\_Click(), которая сравнивает в цикле введенный пользователем в поле txtTerm термин с терминами, хранящимися в символьном массиве strA.

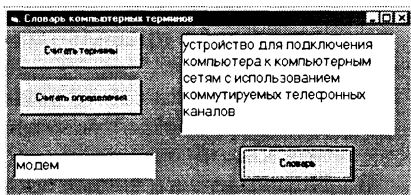
В случае совпадения, в поле txtDesc выводится соответствующее термину определение, в противном случае в окне появляется текст сообщения об отсутствии данного термина в словаре.

```
Private Sub cmdVoc_Click()
For bytI = 1 To 10
If txtTerm.Text = strA(bytI)
Then txtDesc.Text = strB(bytI): bytI = 11
Else txtDesc.Text = "Такого термина
в словаре нет"
Next bytI
End Sub
```

- 4 Для того чтобы нельзя было изменить текст в окне вывода определений txtDesc, для свойства Locked установить значение True.

- 5 Запустить проект и щелкнуть по кнопкам: *Считать термины*, *Считать определения*. Ввести в окно термина термин и щелкнуть по кнопке *Словарь*.

Процедуру можно повторять многократно.



5.17. Проект «Русско-английский и англо-русский словарь» хранится в папке prjZ5\_17.vbp.

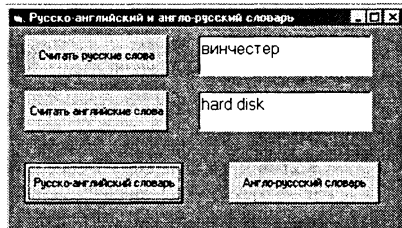


- 1 Модернизировать предыдущий проект 5.16 «Словарь компьютерных терминов». Создать текстовые файлы русских слов (russian.txt) и английских слов (english.txt). Порядки следования слов и их переводов должны быть одинаковыми.
- 2 Поместить на форму кнопки cmdRussian, cmdEnglish, cmdRE и cmdER и два текстовых поля txtRussian (для ввода и вывода русских слов) и txtEnglish (для ввода и вывода английских слов).
- 3 Создать событийные процедуры cmdRussian\_Click() и cmdEnglish\_Click(), которые обеспечивают считывание данных из файлов в строковые массивы.
- 4 Создать событийные процедуры cmdRE и cmdER, которые обеспечивают перевод:

```
Private Sub cmdRE_Click()
For bytI = 1 To 10
If txtRussian.Text = strA(bytI) _
Then txtEnglish.Text=strB(bytI) : bytI=11 _
Else txtEnglish.Text = "Такого термина
в словаре нет"
Next bytI
End Sub
```

- 5 Запустить проект. Щелкнуть по кнопкам *Считать русские слова* и *Считать английские слова*.

Ввести слово в одно из полей и щелкнуть по соответствующей кнопке для получения перевода.



### 5.18. Проект «Контроль знаний» хранится в папке rgjZ5\_18.vbr.

- 1 Для проверки знаний, например по информатике, создать текстовые файлы, содержащие 10 вопросов (problem.txt) и 10 правильных ответов (answer.txt).
- 2 Поместить на форму кнопки cmdProbl и cmdAnsw и создать для них событийные процедуры считывания вопросов в строковый массив strA и правильных ответов в строковый массив strB.
- 3 Поместить на форму кнопку cmdCont и создать для нее событийную процедуру проверки знаний. Вопросы вывести в поле txtProbl, а их номера в поле txtNum. Запрос ответов осуществить с помощью функции InputBox и сохранить их в строковом массиве strC.

Правильные ответы отображать в поле txtAnsw1, а ответы, введенные учащимися, в поле txtAnsw2.

Последовательно в цикле сравнить с помощью условного оператора ответы, введенные учащимися (значения массива strC), с правильными ответами, хранящимися в строковом массиве strB.

Реализовать вывод соответствующих сообщений в поле txtMsg и подсчет количества ошибок с помощью переменной bytN и вывод ее значения в поле txtError.

С помощью оператора **Select Case** осуществить выставление оценки и ее вывод в поле txtBall.

```

Private Sub cmdCont_Click()
For bytK = 1 To 10
    txtProbl.Text = strA(bytK)
    txtNum.Text = Str(bytK)
    strC(bytK)=InputBox("Введите ответ", "Ответ")
If strB(bytK) = strC(bytK)
Then txtMsg.Text = "Правильно!"
Else txtMsg.Text = "Неправильно!": bytN =
    bytN + 1
    txtError.Text = Str(bytN)
    txtAnsw1.Text = strX
    txtAnsw2.Text = strB(bytK)
    MsgBox "Следующий вопрос?"
Next bytK
Select Case bytN
Case 0
    txtBall.Text = "Отлично"
Case 1
    txtBall.Text = "Хорошо"
Case 2
    txtBall.Text = "Удовлетворительно"
Case Is > 2
    txtBall.Text = "Неудовлетворительно"
End Select
End Sub
    
```

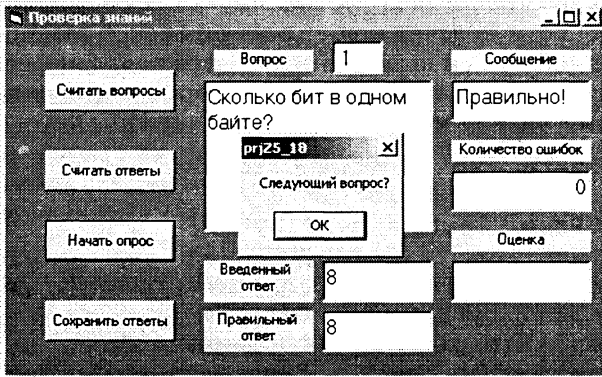
4 Поместить на форму кнопку cmdSave и создать для нее событийную процедуру сохранения ответов учащихся (строковый массив strC) в файле student.txt.

5 Запустить проект и щелкнуть по кнопкам: *Считать вопросы, Считать ответы и Начать опрос.*

В окне вопросов последовательно будут появляться вопросы, а ввод ответов будет осуществляться с помощью окна функции InputBox.

В соответствующих окнах будут появляться введенный и правильный ответы, сообщения и количество ошибок. После ответа на последний вопрос в окне *Оценка* появится результирующая отметка.

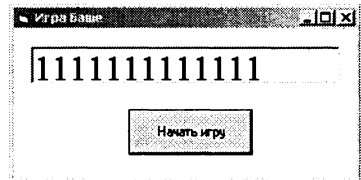
Для сохранения ответов учащихся в файле щелкнуть по кнопке *Сохранить ответы*.



5.19. Проект «Игра Баше-2» хранится в папке `prjZ5_19.vbp`.

- 1 Поместить на форму текстовое поле `txt1`. Заменить вывод количества предметов с помощью оператора `Print`, на вывод определенного количества символов в текстовое поле с помощью цикла со счетчиком:

```
For bytL = 1 To bytN
txt1.Text = txt1.Text
+ "1"
Next bytL
```



5.20. Проект «Игра Баше-3» хранится в папке `prjZ5_20.vbp`.

- 1 Поместить после цикла со счетчиком строки:

```
If bytN = 1 Then MsgBox "После хода
компьютера остался 1 предмет.
Вы проиграли!", , "Игра Баше": End
strX = InputBox("Какое количество предметов
Вы забираете?", "Ваш ход!")
bytN = bytN - Val(strX)
If bytN = 1 Then MsgBox "После Вашего хода
остался 1 предмет. Компьютер проиграл!!", ,
"Игра Баше": End
```

5.21. Проект «Усовершенствованный мультисистемный калькулятор» хранится в папке `prjZ5_21`.

1 Открыть файл проекта `prj6.vbp`. Поместить на форму три текстовых поля `txt1Hex`, `txt2Hex`, `txt3Hex`, две кнопки `cmdDecHex` и `cmdHexDec` и метку `lblHex` с надписью *Шестнадцатеричная система*.

2 Ввести программный код событийной процедуры `cmdDecHex_Click()` перевода десятичных чисел в шестнадцатеричные:

```
Private Sub cmdDecHex_Click()
    txt1Hex.Text = Hex(Val(txt1Dec.Text))
    txt2Hex.Text = Hex(Val(txt2Dec.Text))
    txt3Hex.Text = Hex(Val(txt3Dec.Text))
End Sub
```

3 Ввести программный код событийной процедуры `cmdHexDec_Click()` перевода шестнадцатеричных чисел в десятичные:

```
Private Sub cmdHexDec_Click()
    txt1Dec.Text = Val("&H" + txt1Hex.Text)
    txt2Dec.Text = Val("&H" + txt2Hex.Text)
    txt3Dec.Text = Val("&H" + txt3Hex.Text)
End Sub
```

4 Ввести десятичные числа и умножить их.

Перевести числа в шестнадцатеричную систему, щелкнув по кнопке *Dec-Hex*.

Ввести шестнадцатеричные числа и перевести их в десятичную систему, щелкнув по кнопке *Hex-Dec*.



5.22. Проект «Инженерный калькулятор» хранится в папке `prjZ5_22`.

1 Открыть проект `prj6.vbp`. Удалить с формы три текстовых поля `txt1Oct`, `txt2Oct`, `txt3Oct`, две кнопки `cmdDecOct` и `cmdOctDec`, две метки `lblDec` и `lblOct`.

2 Удалить из программного кода событийные процедуры `cmdDecOct_Click()` и `cmdOctDec_Click()`.

3 Добавить на форму девять кнопок `cmdSin`, `cmdCos`, `cmdTan`, `cmdAtn`, `cmdSt`, `cmdSqr`, `cmdRnd`, `cmdLog` и `cmdExp`.

4 Для каждой из этих кнопок создать событийные процедуры, реализующие вычисление соответствующих функ-

ций: синус, косинус, тангенс, арктангенс, возведение в степень, квадратный корень, случайное число, натуральный логарифм и показательная функция.

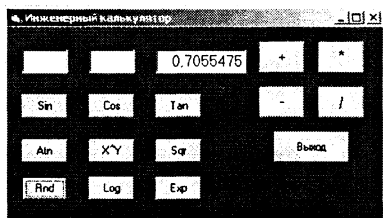
Например, для получения случайного числа событийная процедура `cmdRnd_Click()` примет вид:

```
Private Sub cmdSt_Click()
txt3Dec.Text = Rnd
End Sub
```

- Ввести самостоятельно программный код других событийных процедур с использованием встроенных функций языка Visual Basic: `Sin(x)`, `Cos(x)`, `Tan(x)`, `ATn(x)`, `X^Y`, `Sqr(x)`, `Log(x)` и `Exp(x)`.

- Запустить проект на выполнение.

Например, для получения случайного числа щелкнуть по кнопке *Rnd*.



### 5.23. Проект «Подсчет цифр в числе» хранится в папке `prjZ5_23`.

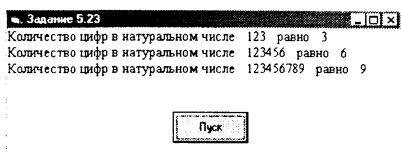
- Поместить на форму кнопку `cmd1` и создать для нее событийную процедуру.
- Ввод чисел осуществить с использованием функции ввода данных `InputBox`. Для преобразования строкового значения функции в десятичное число использовать функцию `Val`.
- Количество цифр в числе равно количеству выполнения тела цикла, в котором десятичное число делится на 10 и выделяется целая часть частного от деления. Для выделения целой части воспользоваться встроенной функцией `Int(x)`, которая возвращает целую часть числа:

```
Dim intDec, intD, intA As Integer, bytK As Byte
Private Sub cmd1_Click()
intDec = Val(InputBox("Введите натуральное десятичное число", "Ввод десятичного натурального числа", 1))
intD = intDec
bytK = 0
Do While intD > 0
intD = Int(intD / 10)
bytK = bytK + 1
```

**Loop**

```
Print "Количество цифр в натуральном числе";
intDec; "равно"; bytK
End Sub
```

- 4 Запустить проект и щелкнуть по кнопке *Пуск*.

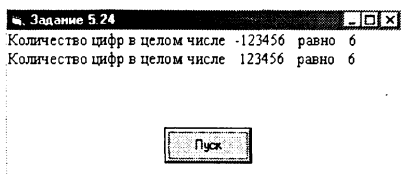


5.24. Проект хранится в папке `prjZ5_24`.

- 1 Открыть проект `prjZ5_23.vbp` и внести дополнения в событийную процедуру. Применить функцию `Abs(x)` для выделения модуля числа:

```
intD = Abs(intD)
```

- 2 Запустить проект на выполнение и щелкнуть по кнопке *Пуск*. В появившемся окне функции ввода `InputBox` ввести число.

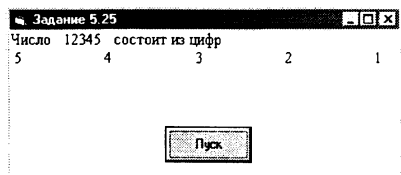


5.25. Проект «Печать цифр числа» хранится в папке `prjZ5_25`.

- 1 Модернизировать проект `prjZ5_24.vbp`. В событийной процедуре для выделения остатка от деления числа  $X$  на число  $Y$  использовать оператор  $X \text{ Mod } Y$ .

Для печати по зонам в качестве разделителя списка печати использовать запятую:

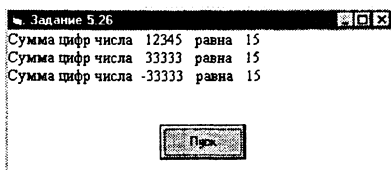
```
Do While intD > 0
bytB = intD Mod 10
Print bytB,
intD = Int(intD/10)
Loop
```



5.26. Проект «Сумма цифр числа» хранится в папке `prjZ5_26`.

- 1 Модернизировать проект `prjZ5_25.vbp`. В программном коде цикла значениями переменной `bytB` являются цифры числа, а их сумма накапливается в переменной `bytS`:

```
Do While intD > 0
bytB = intD Mod 10
bytS = bytS + bytB
intD = (intD - bytB) / 10
Loop
```



5.27. Проект «Числа без одинаковых цифр» хранится в папке prjZ5\_27.

- 1 В событийной процедуре используем вложенные циклы и неполное ветвление, которое обеспечивает в случае совпадения цифр переход к следующей цифре:

```
Dim intI, intJ As Byte
Private Sub cmd1_Click()
For intI = 1 To 9
For intJ = 0 To 9
If intJ = intI Then intJ = intJ + 1
Print intI * 10 + intJ;
Next intJ
Next intI
End Sub
```



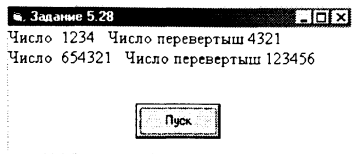
5.28. Проект «Перевертыш чисел» хранится в папке prjZ5\_28.

- 1 В первом цикле с предусловием заполнить массив цифрами числа, а во втором цикле со счетчиком обеспечить обратный порядок их следования в числе. Программный код проекта будет следующий:

```
Dim intN, intN1, intN2 As Integer, bytI,
bytJ As Byte, sngA(1 To 10) As Single
Private Sub cmd1_Click()
intN = Val(TextBox("Введите целое
десятичное число", "Ввод числа", 1))
intN2 = intN
bytI = 0
Do While intN > 0
bytI = bytI + 1
sngA(bytI) = intN Mod 10
intN = Int(intN / 10)
Loop
intN1 = sngA(1)
For bytJ = 2 To bytI
intN1 = intN1 * 10 + sngA(bytJ)
Next bytJ
```

```
Print "Исходное число"; intN2; "Перевертыш";
intN1
End Sub
```

- 2 Запустить проект, щелкнуть по кнопке *Пуск* и ввести число.



5.29. Проект «Перевод десятичного числа в двоичную систему счисления» хранится в папке `prjZ5_29`.

- 1 Реализовать перевод натурального десятичного числа в двоичное число с использованием известного алгоритма последовательного деления числа на основание новой системы счисления.

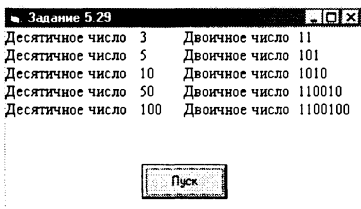
Реализовать последовательное деление введенного числа на 2 в цикле с предусловием `intD > 0`, который будет выполняться до тех пор, пока частное от деления не станет дробным числом.

В переменной `intA` хранить остатки деления. С помощью оператора ветвления преобразовать числовое значение остатка (переменная `intA`) в строковую переменную `strBin`, хранящую двоичное число.

Программный код проекта будет следующий:

```
Dim intDec, intD, intA As Integer, strBin As String
Private Sub cmd1_Click()
intDec = Val(InputBox("Введите натуральное десятичное число", "Десятичное число", 1))
strBin = " "
intD = intDec
Do While intD > 0
intA = intD Mod 2
If intA = 1
Then strBin = "1" + strBin
Else strBin = "0" + strBin
intD = Int(intD / 2)
Loop
Print "Десятичное число"; intDec, "Двоичное число"; strBin
End Sub
```

- 2 Запустить проект, щелкнуть по кнопке *Пуск* и ввести число.





5.30. Проект «Усовершенствованный строковый калькулятор» хранится в папке `prjZ5_30`.

5.31. Проект «Подсчет количества экземпляров символа в тексте» хранится в папке `prjZ5_31`.

- 1 На форме разместить кнопку `cmd1` и создать для нее событийную процедуру подсчета символов в тексте.

Ввод текста и искомого символа производится с помощью функции ввода `MsgBox`. и ее значения присваиваются переменным `strA` и `strB`.

В цикле со счетчиком, выполнение которого производится столько раз, сколько символов в тексте (значение функции `Len(strA)`), последовательно вырезаются символы из введенного текста и с помощью неполного условного оператора сравниваются с введенным искомым символом. В случае выполнения условия значение счетчика (переменной `bytN`) увеличивается на единицу:

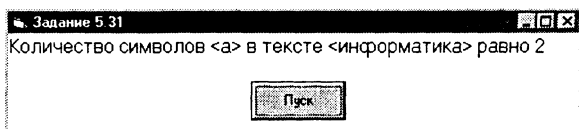
```
Dim strA, strB, strX As String, bytI, bytN
As Byte
```

```
Private Sub cmd1_Click()
strA = InputBox("Введите текст",
"Ввод текста")
strX = InputBox("Введите искомый символ",
"Ввод символа")
bytN = 0
```

```
For bytI = 1 To Len(strA)
strB = Mid$(strA, bytI, 1)
If strB = strX Then bytN = bytN + 1
Next bytI
```

```
Print "Количество символов <" & strX;
"> в тексте <" & strA; "> равно"; bytN
End Sub
```

- 2 После запуска проекта и щелчка по кнопке *Пуск* ввести текст и искомый символ.



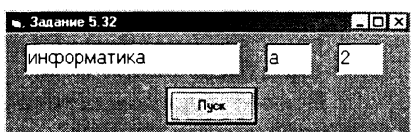
5.32. Проект «Подсчет количества экземпляров символа в тексте-2» хранится в папке `prjZ5_32`.

- 1 Открыть проект `prjZ5_31.vbp` и на форме разместить три текстовых поля: для ввода текста `txtText`, для ввода символа `txtSim` и для вывода количества символов в тексте `txtNum`.

Вместо строковых переменных использовать свойство текстовых полей txtText.Text, txtSim.Text, txtNum.Text:

```
Dim strB As String, bytI, bytN As Byte
Private Sub cmd1_Click()
    bytN = 0
    For bytI = 1 To Len(txtText.Text)
        strB = Mid$(txtText.Text, bytI, 1)
        If strB = txtSim.Text Then bytN = bytN + 1
    Next bytI
    txtNum.Text = bytN
End Sub
```

- 2 Запустить проект и ввести в первое текстовое поле текст, а во второе — символ. Щелкнуть по кнопке *Пуск*.



5.33. Проект «Замена символов в тексте» хранится в папке prjZ5\_33.

- 1 На форме разместить кнопку cmd1 и создать для нее событийную процедуру замены в тексте одного символа на другой.

Ввод текста, первого и второго символов производится с помощью функции ввода MsgBox и ее значения присваиваются переменным strA, strX и strY.

В цикле со счетчиком, количество выполнения которого равно количеству символов в тексте (значение функции Len(strA)), последовательно вырезаются символы из введенного текста и с помощью неполного условного оператора сравниваются с введенным искомым символом. В случае выполнения условия символы заменяются с использованием функции Mid\$:

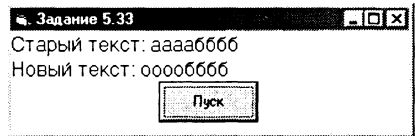
```
Dim strA, strB, strX, strY As String, bytI
As Byte
Private Sub cmd1_Click()
    strA = InputBox("Введите текст",
    "Ввод текста")
    strX = InputBox("Введите символ, который
    необходимо заменить", "Ввод первого символа")
    strY = InputBox("Введите символ, который
    необходимо вставить", "Ввод второго символа")
    Print "Старый текст: "; strA
    For bytI = 1 To Len(strA)
        strB = Mid$(strA, bytI, 1)
        If strB = strX Then Mid$(strA, bytI, 1) = strY
```

```

Next bytI
Print "Новый текст: "; strA
End Sub

```

- 2 Запустить проект и ввести в первое поле текст, во второе — заменяемый символ, а в третье — новый символ. Щелкнуть по кнопке *Пуск*.



5.34. Проект «Замена символов в тексте-2» хранится в папке `prjZ5_34`.

- 1 Открыть проект `prjZ5_33.vbp` и на форме разместить три текстовых поля: для ввода текста `txtText`, для ввода первого символа `txtSim1` и для ввода второго символа `txtSim2`.

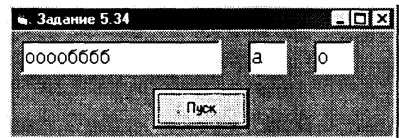
Вместо строковых переменных использовать свойство текстовых полей `txtText.Text`, `txtSim1.Text`, `txtSim2.Text`:

```

Dim strA, strB As String, bytI As Byte
Private Sub cmd1_Click()
strA = txtText.Text
For bytI = 1 To Len(txtText.Text)
strB = Mid$(txtText.Text, bytI, 1)
If strB = txtSim1.Text Then
Mid$(strA, bytI, 1) = txtSim2.Text
Next bytI
txtText.Text = strA
End Sub

```

- 2 Запустить проект и ввести в первое окно текст, во второе окно заменяемый символ и в третье окно — новый символ.



После щелчка по кнопке

*Пуск* в первом окне появится измененный текст.

5.35. Проект «Слово-перевертыш» хранится в папке `prjZ5_35`.

- 1 На форме разместить кнопку `cmd1` и создать для нее событийную процедуру преобразования введенного слова в слово с обратной последовательностью следования составляющих его символов.

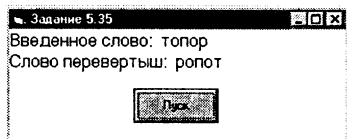
Ввод слова производится с помощью функции ввода `MsgBox`, и ее значение присваивается переменной `strA`.

В цикле со счетчиком, количество выполнения которого равно количеству символов в слове, последовательно вырезаются в прямой последовательности (слева направо)

символы из введенного слова и в обратной последовательности (справа налево) присваиваются переменной `strX`, значением которой после завершения цикла будет слово-перевертыш:

```
Dim strA, strB, strX As String,
bytI As Byte
Private Sub cmd1_Click()
strA = InputBox("Введите слово",
"Ввод слова")
Print "Введенное слово: "; strA
strX = " "
For bytI = 1 To Len(strA)
strB = Mid$(strA, bytI, 1)
strX = strB + strX
Next bytI
Print "Слово перевертыш: "; strX
End Sub
```

- 2 Запустить проект и щелкнуть по кнопке *Пуск*. Ввести в появившемся диалоговом окне слово.



5.36. Проект «Слово-перевертыш-2» хранится в папке `prjZ5_36`.

- 1 Открыть проект `prjZ5_35.vbp` и на форме разместить два текстовых поля: для ввода слова `txt1`, для вывода слова-перевертыша `txt2`.  
Вместо строковых переменных использовать свойство текстовых полей `txt1.Text` и `txt2.Text`:

```
Dim strB As String, bytI As Byte
Private Sub cmd1_Click()
txt2.Text = " "
For bytI = 1 To Len(txt1.Text)
strB = Mid$(txt1.Text, bytI, 1)
txt2.Text = strB + txt2.Text
Next bytI
End Sub
```

- 2 Запустить проект и ввести в первое поле слово. Щелкнуть по кнопке *Пуск*, слово-перевертыш появится во втором окне.

5.37. Проект «Шифровка и дешифровка-2» хранится в папке `prjZ5_37`.

- 1 Открыть проект `prj8.vbp` и в программном коде заменить циклы со счетчиком, которые обеспечивают последовательное вырезание символов из секретного текста, на циклы с послеусловием:

```

Dim bytI, bytJ, bytZ(1 To 200) As Byte,
strKF, strA, strB As String
Private Sub cmdCod_Click()
strKF = "электронно-вычислительные машины,
компьютеры (железо) позволяют находить
значения функций, а ещё свойства готовых
объектов"
bytI = 1
Do
strA = Mid$(txtSecrtext.Text, bytI, 1)
For bytJ = 1 To Len(strKF)
If Mid$(strKF, bytJ, 1) = strA Then
bytZ(bytI) = bytJ: bytJ = Len(strKF) + 1
Next bytJ
txtCod.Text = txtCod.Text + Str(bytZ(bytI))
bytI = bytI + 1
Loop While bytI <= Len(txtSecrtext.Text)
End Sub
Private Sub cmdDecod_Click()
bytI = 1
Do
strB = Mid$(strKF, bytZ(bytI), 1)
txtDecod.Text = txtDecod.Text + strB
bytI = bytI + 1
Loop While bytI <= Len(txtSecrtext.Text)
End Sub

```

5.38. Проект «Текстовый редактор-2» хранится в папке prjZ5\_38.

- 1 Открыть проект prj9.vbp и ввести команду [Tools-Menu Editor...].

На появившейся диалоговой панели *Menu Editor* создать заголовок первого уровня. В поле *Caption* ввести надпись *Файл* и в поле *Name* имя объекта *mnuFile*.

Для создания пункта меню уровня команд щелкнуть по кнопке со стрелкой вправо и ввести надпись *Выход* и имя *mnuExit*.

Ввести событийную процедуру для пункта меню *Выход*:

```

Private Sub mnuExit_Click()
End
End Sub

```

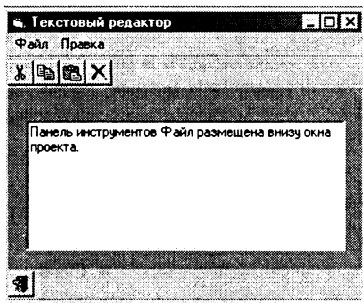
5.39. Проект «Текстовый редактор-3» хранится в папке prjZ5\_39.

- 1 Открыть проект prjZ5\_38.vbp. В режиме конструирования проекта поместить на форму управляющий элемент *Панель инструментов* и присвоить ему имя *tbrToolbar2*.

- 2 Активизировать свойство *Custom*. На появившейся диалоговой панели *Property Pages* выбрать вкладку *Buttons*. Щелкнуть по кнопке *Insert Button*. В поле *Key*: ввести имя кнопки панели инструментов *Exit*.
- 3 В режиме конструирования проекта дважды щелкнуть по панели инструментов проекта, будет создана пустая заготовка событийной процедуры. Ввести код процедуры, которая будет обеспечивать выход из проекта:

```
Button.Key = "Exit"
End
```

- 4 Поместить на форму управляющий элемент *Список изображений* (*ImageList*) и присвоить ему имя *imlToolbar2*.
- 5 Активизировать свойство *Custom* и на появившейся диалоговой панели *Property Pages* выбрать вкладку *Images*. Щелкнуть по кнопке *Insert Picture...* и выбрать графический файл (*exit.bmp*) для размещения на кнопке *Exit*.
- 6 Для элемента *tbrToolbar2* активизировать свойство *Custom*.  
На вкладке *General* в окне *ImageList* выбрать элемент *imlToolbar2*.
- 7 Для размещения панели инструментов *Файл* в определенном месте окна проекта для свойства *Align* установить требуемое значение, например, для размещения в нижней части окна выбрать значение *vbAlignBottom*.
- 8 Запустить проект и с помощью панели инструментов *Файл* выйти из проекта.



5.40. Проект «Текстовый редактор-3» хранится в папке *prjZ5\_40*.

- 1 Открыть проект *prjZ5\_39.vbp*. В режиме [View-Object] добавить с помощью редактора *Menu Editor* в меню *Правка* команды *Найти* (*mnuSearch*) и *Заменить* (*mnuReplace*).
- 2 Создать для команды *mnuSearch* событийную процедуру, которая обеспечивает поиск в выделенном фрагменте текста заданной последовательности символов. Для ввода

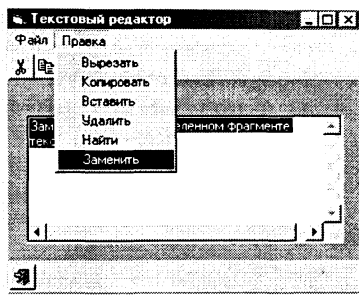
подстроки использовать функцию `InputBox`, а для поиска ее места в выделенном фрагменте текста функцию `InStr`. Воспользоваться свойством `SelStart` текстового поля для установки курсора в заданную позицию:

```
Private Sub mnuSearch_Click()
strA = InputBox("Введите подстроку", "Поиск подстроки")
frm1.txtEdit.SelStart =
InStr(frm1.txtEdit.SelText, strA) - 1
End Sub
```

- 3 Создать для команды *mnuReplace* событийную процедуру, которая обеспечивает поиск в выделенном фрагменте текста заданной последовательности символов и заменяет ее на другую заданную последовательность. Для выделения фрагмента текста требуемой длины воспользоваться свойством `SelLength` текстового поля:

```
Private Sub mnuReplace_Click()
strA = InputBox("Введите подстроку",
"Поиск подстроки")
strB = InputBox("Введите подстроку",
"Замена подстроки")
frm1.txtEdit.SelStart =
InStr(frm1.txtEdit.SelText, strA) - 1
frm1.txtEdit.SelLength = Len(strA)
frm1.txtEdit.SelText = strB
End Sub
```

- 4 Добавить в панель инструментов `tbrToolbar` две кнопки `Search` и `Replace`, а в список изображений `imlToolbar` — изображения из файлов `search.bmp` и `replace.bmp`. Синхронизировать кнопки и изображения с использованием свойства *Custom*.
- 5 В событийную процедуру панели инструментов `tbrToolbar_ButtonClick` добавить варианты **Case Is** = "Search" и **Case Is** = "Replace", которые должны содержать программные коды, аналогичные событийным процедурам одноименных команд меню.
- 6 Установить для значения свойства текстового поля *ScrollBars* значение *Both*. Появятся горизонтальная и вертикальная линейки прокрутки.
- 7 Запустить проект. Ввести текст, выделить его и ввести, например, команду [Правка-Заменить]. Ввести с помощью появляющихся диалоговых панелей функции `InputBox` значения двух подстрок.



5.41. Проект «Логические функции» хранится в папке `prjZ5_41`.

- 1 Поместить на форму три массива текстовых полей, по четыре элемента в каждом массиве: `txtA(number)` и `txtB(number)` — для ввода логических аргументов и `txtC(number)` — для вывода значений логической функции. Технология создания массива для любого управляющего элемента одинакова. Сначала на форму помещается один экземпляр управляющего элемента и вводится команда *Copy*. После ввода команды *Paste* появляется информационное окно, которое предлагает либо согласиться с созданием массива управляющих элементов, либо отказаться.
- 2 Присвоить свойству `Text` текстовых полей логических аргументов требуемые наборы логических значений.
- 3 Определить логические переменные:

```
Dim blnA, blnB, blnC As Boolean
```

- 4 Поместить на форму пять кнопок `cmdAnd`, `cmdOr`, `cmdNot`, `cmdXor` и `cmdEqv` и для каждой кнопки создать событийную процедуру для получения таблицы истинности соответствующей логической функции.

Событийная процедура для функции логического умножения `cmdAnd_Click()` запишется следующим образом:

```
Private Sub cmdAnd_Click()  
For bytI = 0 To 3  
    blnA = txtA(bytI)  
    blnB = txtB(bytI)  
    blnC = blnA And blnB  
    txtC(bytI) = blnC  
Next bytI  
End Sub
```

- 5 Создать аналогичные событийные процедуры для других логических функций.



- 6 Запустить проект на выполнение.

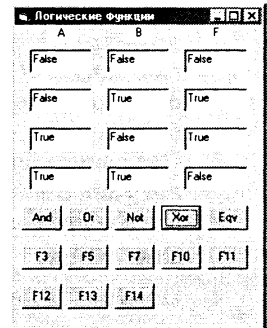
Щелкнуть по любой кнопке (например, *And*) и получить таблицу истинности соответствующей логической функции.



- 5.42. Проект «Логические функции-2» хранится в папке `prjZ5_42`.

- 1 Открыть проект `prjZ5_41.vbp` и дополнить его. Добавить на форму восемь кнопок и создать для них соответствующие событийные процедуры (смотри таблицу 3.1).
- 2 Например, событийная процедура для логической функции  $F_7$  запишется следующим образом:

```
Private Sub _cmdF7_Click()
For bytI = 0 To 3
blnA = txtA(bytI)
blnB = txtB(bytI)
blnC = (Not blnA And _
blnB) Or (blnA And _
Not blnB)
txtC(bytI) = blnC
Next bytI
End Sub
```



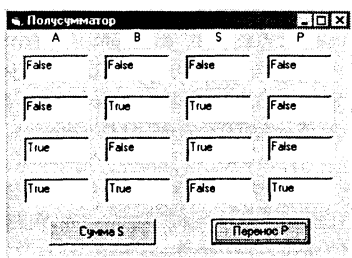
- 5.43. Проект «Полусумматор» хранится в папке `prjZ5_43`.

- 1 Поместить на форму четыре массива текстовых полей, по четыре элемента в каждом массиве: `txtA(number)` и `txtB(number)` — для ввода логических аргументов и `txtS(number)` — для вывода значений суммы и `txtP(number)` — для вывода значений переноса.
- 2 Поместить на форму две кнопки `cmdS` и `cmdP` и создать для них событийные процедуры (смотри задание 3.28):

```
Private Sub cmdS_Click()
For bytI = 0 To 3
blnA = txtA(bytI)
blnB = txtB(bytI)
blnP = blnA And blnB
blnS = (blnA Or blnB) And Not blnP
txtS(bytI) = blnS
```

```
Next bytI
End Sub
```

- 3 Поместить на форму четыре метки lblA, lblB, lblS и lblP. Запустить проект и щелкнуть по кнопкам.



5.44. Проект «Сумматор» хранится в папке prjZ5\_44.

- 1 Открыть проект prjZ5\_43.vbp и дополнить его. Поместить на форму дополнительный массив текстовых полей txtP0 (number) для ввода значений переноса из младшего разряда.

Количество элементов во всех массивах текстовых полей увеличить до восьми.

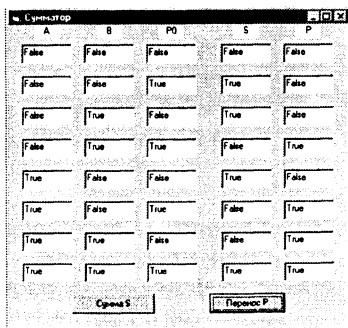
- 2 Создать событийные процедуры для определения суммы и переноса (смотри задания 3.29 и 3.30).

Процедура для суммы:

```
Private Sub cmdS_Click()
For bytI = 0 To 7
blnA = txtA(bytI)
blnB = txtB(bytI)
blnP0 = txtP0(bytI)
blnP = (blnA And blnB) Or (blnA And blnP0)
Or (blnB And blnP0)
blnS = (blnA Or blnB Or blnP0) And Not
blnP Or (blnA And blnB And blnP0)
txtS(bytI) = blnS
Next bytI
End Sub
```

- 3 Запустить проект и щелкнуть по кнопкам *Сумма S* и *Перенос P*.

Будут получены таблицы истинности для логических функций, которые используются для определения суммы и переноса в старший разряд.



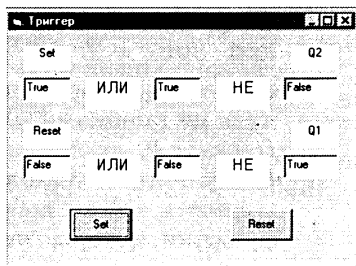
### 5.45. Проект «Триггер» хранится в папке prjZ5\_45.

- 1 Поместить на форму шесть текстовых полей: txtSet и txtReset — для ввода начальных логических значений на входах триггера, txtOr1Out и txtOr2Out — для визуального контроля промежуточных логических значений, txtQ2 и txtQ1 для вывода состояния выходов триггера.
- 2 Поместить на форму восемь меток: lblOr1, lblOr2, lblNot1, lblNot2 — для обозначения составляющих триггер логических элементов, lblSet, lblReset, lblQ1, lblQ2 — для обозначения входов и выходов триггера.
- 3 Определить логические переменные:
 

```
Dim blnS, blnR, BlnOr1Out, BlnOr2Out,
    blnNot1Out, blnNot2Out As Boolean
```
- 4 Поместить на форму две кнопки cmdSet и cmdReset и создать для них событийные процедуры. Процедура установки значения триггера:

```
Private Sub cmdSet_Click()
    txtSet.Text = True
    txtReset.Text = False
    blnS = txtSet.Text
    blnR = txtReset.Text
    BlnOr1Out = blnS Or blnNot2Out
    blnNot1Out = Not BlnOr1Out
    BlnOr2Out = blnNot1Out Or blnR
    blnNot2Out = Not BlnOr2Out
    txtOr1Out = BlnOr1Out
    txtOr2Out = BlnOr2Out
    txtQ2 = blnNot1Out
    txtQ1 = blnNot2Out
End Sub
```

- 5 Запустить проект. Установить триггер, ввести в поле Set значение **True**, проследить за установкой значений на элементах триггера. Сбросить триггер, ввести в поле Reset значение **True**.



### 5.46. Проект «Графический редактор-2» хранится в папке prjZ5\_46.

- 1 Модернизировать проект «Графический редактор» (файл prj12.vbp). Для ввода координат графических примитивов

поместить на форму четыре текстовых поля: txtX1, txtY1, txtX2 и txtY2. Присвоить свойству Text значение 0.

- 2 Поместить на форму четыре текстовых поля для ввода числовых кодов цвета: txtC (для ввода аргумента функции QColor()) и txtR, txtG, txtB (для ввода аргументов функции RGB()). Присвоить свойству Text значение 0.
- 3 Поместить на форму четыре текстовых поля для ввода параметров рисования окружности: txtRad (радиус), txtAng1 и txtAng2 (начальный и конечный угол дуги) и txtAsp (степень сжатия).
- 4 Поместить на форму двенадцать меток для обозначения текстовых полей.
- 5 Изменить событийные процедуры рисования примитивов так, чтобы параметры считывались из текстовых окон. Предусмотреть с помощью оператора условного перехода возможность выбора использования функции QColor() или функции RGB() для задания цвета. Например, событийная процедура рисования окружности запишется следующим образом:

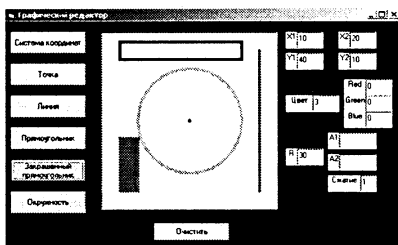
```
Private Sub cmdCircle_Click()
    bytX1 = Val(txtX1)
    bytY1 = Val(txtY1)
    bytRad = Val(txtRad)
    sngA1 = Val(txtAng1)
    sngA2 = Val(txtAng2)
    sngAsp = Val(txtAsp)
    If Val(txtC.Text) > 0
    Then lngC = QColor(Val(txtC))
    Else: bytR = Val(txtR): bytG = Val(txtG):
    bytB = Val(txtB): lngC = RGB(bytR, bytG,
    bytB)
    picPaint.Circle (bytX1, bytY1), bytRad,
    lngC, sngA1, sngA2, sngAsp
End Sub
```

- 6 Поместить на форму кнопку txtCls и создать событийную процедуру очистки графического поля:

```
Private Sub txtCls_Click()
    picPaint.Cls
End Sub
```

- 7 Запустить проект. Ввести в поля числовые значения (значения углов измеряются в радианах и должны находиться в интервале от  $-2\pi$  до  $2\pi$ ).

Щелкнуть по кнопкам, в графическом поле будут нарисованы графические примитивы.



5.47. Проект «Графический редактор-3» хранится в папке `prjZ5_47`.

- 1 Открыть проект `prjZ5_46.vbr`. Удалить с формы все текстовые поля и метки.
- 2 Поместить на форму четыре метки `lblX1Rez`, `lblY1Rez`, `lblX2Rez` и `lblY2Rez` для вывода значений координат первой и второй точек, заданных щелчком мыши.
- 3 Создать событийную процедуру `picPaint_MouseDown` (событие — нажатие кнопки мыши), реализующую считывание координат точек после нажатия кнопок мыши и присвоение их значений переменным `sngX` и `sngY`.

Нажатие одной из кнопок мыши присваивает переменной `Button` числовые значения (1 — левая кнопка, 2 — правая кнопка и 4 — центральная кнопка).

Для присвоения числовых координат двух точек значениям переменных `sngX1`, `sngY1`, `sngX2` и `sngY2` используем оператор **Select Case**.

Выведем полученные значения на форму с помощью меток `lblX1Rez`, `lblY1Rez`, `lblX2Rez` и `lblY2Rez`:

```
Private Sub picPaint_MouseDown(Button As Integer, Shift As Integer, sngX As Single, sngY As Single)
```

```
Select Case Button
```

```
Case Is = 1
```

```
sngX1 = sngX: sngY1 = sngY:  
lblX1Rez = sngX1: lblY1Rez = sngY1
```

```
Case Is = 2
```

```
sngX2 = sngX: sngY2 = sngY:  
lblX2Rez = sngX2: lblY2Rez = sngY2
```

```
End Select
```

```
End Sub
```

- 4 Изменить событийные процедуры рисования графических примитивов, для простоты задать цвет примитивов с помощью цветовых констант. Например, событийная

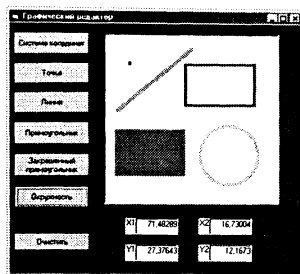
процедура рисования окружности примет следующий вид:

```
Private Sub cmdCircle_Click()
picPaint.Circle (sngX1, sngY1), sngX2, vbCyan
End Sub
```

5 Запустить проект.

Нажать левую, а затем правую кнопки мыши, на метках появятся значения координат точек.

Щелкнуть по любой кнопке, на графическом поле появится соответствующий графический примитив.



5.48. Проект «Графический редактор-4» хранится в папке rjZ5\_48.

- 1 Открыть проект rjZ5\_47.vbp. Удалить с поля все кнопки.
- 2 С помощью редактора меню *Menu Editor* создать меню и ввести пункты:

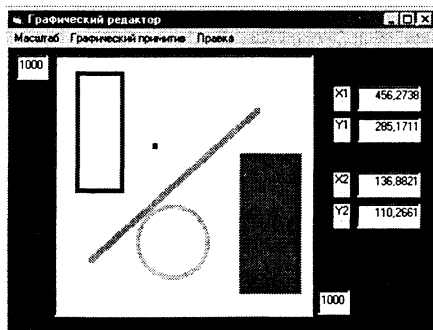
<i>Масштаб</i>	<i>Графический примитив</i>	<i>Правка</i>
....100*100	....Точка	....Очистить
....1000*1000	....Линия	
	....Прямоугольник	
	....Закрашенный прямоугольник	
	....Окружность	

- 3 Поместить на форму текстовые поля txtX и txtY для вывода максимальных координат выбранного масштаба. Создать событийные процедуры для пунктов меню *Масштаб*. Процедура для пункта *1000\*1000*:

```
Private Sub mnul000_Click()
picPaint.Scale (0, 1000)-(1000, 0)
txtX = 1000
txtY = 1000
End Sub
```

- 4 Внести изменения в названия процедур рисования графических примитивов (вместо имен кнопок должны быть имена пунктов меню).
- 5 Запустить проект вполне работоспособного графического редактора.

Задать с помощью мыши координаты точек и с помощью меню выбрать масштаб и графический примитив.



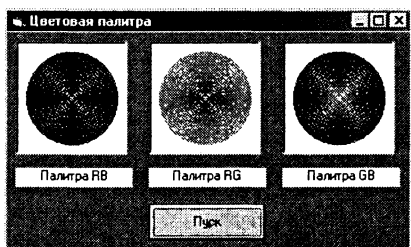
5.49. Проект «Палитра цветов» хранится в папке `prjZ5_49`.

- 1 Поместить на форму три графических окна `picColor1`, `picColor1` и `picColor1` и кнопку `cmdStart`.
- 2 Создать событийную процедуру, которая задает масштаб для каждого окна и рисует в цикле окружности с увеличивающимся радиусом и изменяющимся цветом:

```
Private Sub cmdStart_Click()
picColor1.Scale (-300, 300)-(300, -300)
picColor2.Scale (-300, 300)-(300, -300)
picColor3.Scale (-300, 300)-(300, -300)
For intI = 0 To 255
picColor1.Circle (0,0), intI,
RGB(255 - intI, 0, intI)
picColor2.Circle (0,0), intI,
RGB(255 - intI, intI, 0)
picColor3.Circle (0,0), intI,
RGB(0, 255 - intI, intI)
Next intI
End Sub
```


- 3 Запустить проект.

В трех графических окнах будут представлены палитры попарно изменяющихся базовых цветов (RB, RG и GB).



5.50. Файл проекта «Выбор цвета» `prjZ5_50.vbp`.

- 1 Поместить на форму графическое поле `picColor` для просмотра выбранного цвета и текстовое поле `txtColor` для вывода его названия.

- 2 Для вывода раскрывающегося списка цветовых констант поместить на форму *комбинированное поле* (ComboBox) и дать ему имя cmb1. 

Задать свойству Style значение Dropdown List (раскрывающийся список).

Внести в список перечень цветовых констант с использованием свойства List.

- 3 Создать событийную процедуру cmb1\_Click(), реализующую задание цвета с помощью цветовых констант путем щелчка по цветовой константе в раскрывающемся списке.

Отображение цвета в графическом и текстовом полях осуществить с помощью инструкции выбора **Select Case**:

```
Private Sub cmb1_Click()
Select Case cmb1.Text
Case Is = "vbBlack"
picColor.BackColor = vbBlack
txtColor = "черный"
Case Is = "vbBlue"
picColor.BackColor = vbBlue
txtColor = "синий"
Case Is = "vbGreen"
picColor.BackColor = vbGreen
txtColor = "зеленый"
Case Is = "vbCyan"
picColor.BackColor = vbCyan
txtColor = "голубой"
Case Is = "vbRed"
picColor.BackColor = vbRed
txtColor = "красный"
Case Is = "vbMagenta"
picColor.BackColor = vbMagenta
txtColor = "сиреневый"
Case Is = "vbYellow"
picColor.BackColor = vbYellow
txtColor = "желтый"
Case Is = "vbWhite"
picColor.BackColor = vbWhite
txtColor = "белый"
End Select
End Sub
```

- 4 Поместить на форму текстовое поле txtQBC для ввода числового кода цвета. Создать событийную процедуру txtQBC\_Change() (событие – изменение содержания



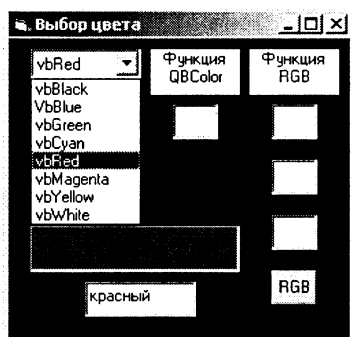
поля, например, ввод нового числового кода), реализующую выбор цвета с использованием функции QBColor:

```
Private Sub txtQBC_Change()
picColor.BackColor =
QBColor(Val(txtQBC.Text))
txtColor = ""
End Sub
```

- 5 Поместить на форму три текстовых поля txtRed, txtGreen и txtBlue для ввода числовых кодов базовых цветов и кнопку cmdRGB. Создать событийную процедуру cmdRGB\_Click(), реализующую выбор цвета с использованием функции RGB:

```
Private Sub cmdRGB_Click()
picColor.BackColor = RGB(Val(txtRed.Text),
Val(txtGreen.Text), Val(txtBlue.Text))
txtColor = ""
End Sub
```

- 6 Запустить проект.  
Выбрать цвет из списка *Константы* двойным щелчком.  
Выбрать цвет путем ввода числового кода в поле *Функция QBColor*.  
Выбрать цвет путем ввода числовых кодов в поля *Функция RGB*.



- 5.51. Проект «Треугольник» хранится в папке prjZ5\_51.

- 1 Открыть проект prjZ5\_48.vbp. Поместить на форму четыре метки lblX3Rez и lblY3Rez — для отображения координат третьей точки и lblX3 и lblY3 — для пояснительных надписей.
- 2 Для получения координат трех точек (вершин треугольника) нажатием кнопки мыши введем событийную процедуру picPaint\_MouseDown. Нажатие одной из кнопок мыши одновременно с нажатием одной из клавиш {Shift}, {Ctrl} или {Alt} присваивает переменной Shift числовые значения (1 — Shift, 2 — Ctrl и 4 — Alt).

```
Private Sub picPaint_MouseDown(Button
As Integer, Shift As Integer,
sngX As Single, sngY As Single)
Select Case Shift
```

```

Case Is = 1
sngX1 = sngX: sngY1 = sngY :
lblX1Rez = sngX1 : lblY1Rez = sngY1
Case Is = 2
sngX2 = sngX: sngY2 = sngY :
lblX2Rez = sngX2: lblY2Rez = sngY2
Case Is = 4
sngX3 = sngX: sngY3 = sngY :
lblX3Rez = sngX3: lblY3Rez = sngY3
End Select
End Sub

```

- 3 Добавить на форму четыре метки lblPerRez и lblSqRez — для вывода результатов вычисления периметра и площади треугольника и lblPer и lblSq — для поясняющих надписей.
- 4 С помощью редактора меню *Menu Editor* изменить меню:

<i>Масштаб</i>	<i>Треугольник</i>	<i>Правка</i>
....100*100	....Построить	....Очистить
....1000*1000	....Периметр	
	....Площадь	

- 5 Создать событийную процедуру построения треугольника по заданным координатам его вершин:

```

Private Sub mnuTrDraw_Click()
picPaint.Line (sngX1, sngY1)-(sngX2, sngY2)
picPaint.Line (sngX1, sngY1)-(sngX3, sngY3)
picPaint.Line (sngX2, sngY2)-(sngX3, sngY3)
End Sub

```

- 6 Создать событийную процедуру вычисления длины сторон треугольника и его периметра:

```

Private Sub mnuTrPer_Click()
sngL1 = Sqr((sngX2-sngX1)^2+(sngY2-sngY1)^2)
sngL2 = Sqr((sngX3-sngX1)^2+(sngY3-sngY1)^2)
sngL3 = Sqr((sngX2-sngX3)^2+(sngY2-sngY3)^2)
sngPer = sngL1 + sngL2 + sngL3
lblPerRez = sngPer
End Sub

```

- 7 Создать событийную процедуру вычисления площади треугольника по формуле Герона:

```

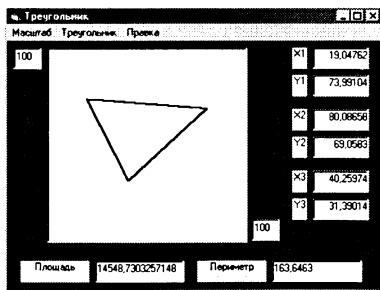
Private Sub mnuTrSq_Click()
lblSqRez = Sqr(sngPer * (sngPer - sngL1) *
(sngPer - sngL2) * (sngPer - sngL3))
End Sub

```

## 8 Запустить проект.

Последовательно ввести координаты трех вершин треугольника одновременным нажатием кнопки мыши и клавиш  $\{Shift\}$ ,  $\{Ctrl\}$  и  $\{Alt\}$ .

В меню *Треугольник* ввести команды *Построить*, *Периметр* и *Площадь*.



## 5.52. Проект «График функции-2» хранится в папке prjZ5\_52.

- 1 Открыть проект prj13.vbp. Поместить на форму поле txtX для ввода максимального значения аргумента X. Внести изменения в программный код рисования оси X:

```
'Ось X
picGraph.Line (-Val(txtX), 0)-(Val(txtX), 0)
For intJ = -Val(txtX) To Val(txtX) Step
Val(txtX) / 10
picGraph.PSet (intJ, 0)
picGraph.Print Int(intJ)
Next intJ
```

- 2 Для автоматической корректировки масштаба по оси Y необходимо определить максимальное и минимальное значение функции на заданном интервале значений аргументов. Для этого необходимо сначала занести в массив значения функции в цикле с заданным шагом изменения аргумента:

```
'заполнение массива
intI = 1
intX = -Val(txtX)
Do While intX < Val(txtX)
intY(intI) = intX ^ 3
intX = intX + 0.01
intI = intI + 1
Loop
```

- 3 Определить значения минимального и максимального значений заполненного массива (смотри проект 5.4):

```
'поиск минимального элемента
intMin = intY(1)
For intI = 2 To 10000
If intY(intI) < intMin Then intMin =
intY(intI): intN = intI
Next intI
```

- 4 Внести изменения в программный код установки масштаба с учетом введенного максимального значения аргумента и вычисленных минимального и максимального значений функции:

```
'Установка масштаба
picGraph.Scale (-Val(txtX),
  intMax)-(Val(txtX), intMin)
```

- 5 Внести изменения в программный код построения графика функции, рисования осей координат и печати значений их шкал:

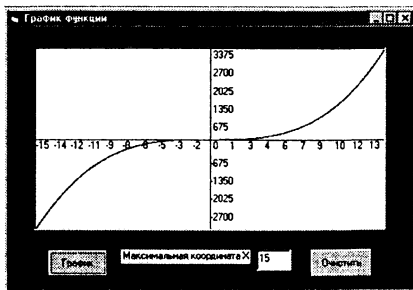
```
'Построение графика
intI = 1
intX = -Val(txtX)
Do While intX < Val(txtX)
intY(intI) = intX ^ 3
picGraph.PSet (intX, intY(intI))
intX = intX + 0.01
intI = intI + 1
Loop
'Ось X
picGraph.Line (-Val(txtX), 0)-(Val(txtX), 0)
For intJ = -Val(txtX) To Val(txtX) Step
Val(txtX) / 10
picGraph.PSet (intJ, 0)
picGraph.Print Int(intJ)
Next intJ
'Ось Y
picGraph.Line (0, intMax)-(0, intMin)
For intJ = intMin To intMax Step intMax/5
picGraph.PSet (0, intJ)
picGraph.Print Int(intJ)
Next intJ
```

- 6 Поместить на форму кнопку cmdCls и создать событийную процедуру очистки графического поля.

- 7 Запустить проект.

Ввести в поле максимальное значение аргумента и щелкнуть по кнопке *График*.

Перед построением функции для другого максимального значения аргумента щелкнуть по кнопке *Очистить*.



5.53. Проект «Стрелочные часы» хранится в папке prjZ5\_53.

- 1 Поместить на форму графическое поле picWatch, кнопку cmdStart, текстовое поле txtHour для ввода количества оборотов стрелки (продолжительности работы часов), ползунков sld1 и кнопку cmdSld для регулировки скорости хода часов. С помощью окна *Свойства объекта* для увеличения диапазона регулировки скорости хода часов установить для объекта sld1 значение свойства *max* = 1000.
- 2 Создать событийную процедуру регулировки скорости хода часов:

```
Private Sub cmdSld_Click()  
dblS = sld1.Value  
End Sub
```

- 3 Создать событийную процедуру хода часов (анимацию движения стрелок):

```
Private Sub cmdStart_Click()  
  'Масштаб  
  picWatch.Scale (-6, 6)-(6, -6)  
  'Завод часов  
  dblHour = Val(txtHour)  
  'Циферблат  
  picWatch.Circle (0, 0), 5  
  For bytN = 1 To 12  
  picWatch.FillStyle = 0 :  
  picWatch.FillColor = vbYellow  
  picWatch.Circle (3 * Sin(6.28 / 12 * bytN),  
  3 * Cos(6.28 / 12 * bytN)), 0.35  
  picWatch.PSet (4 * Sin(6.28 / 12 * bytN) -  
  0.5, 4 * Cos(6.28 / 12 * bytN) + 0.3),  
  vbWhite  
  picWatch.Print bytN  
  Next bytN  
  'Анимация движения стрелок  
  For dblN = 1 To 360 * dblHour  
  picWatch.Circle (0, 0), 0.2  
  picWatch.Line (0, 0)-(2.5 * Sin(6.28 / 360  
  * dblN), 2.5 * Cos(6.28 / 360 * dblN))  
  picWatch.Line (0, 0)-(2 * Sin(6.28 / 21600  
  * dblN), 2 * Cos(6.28 / 21600 * dblN))  
  picWatch.Line (0, 0)-(1.5 * Sin(6.28 /  
  1296000*dblN), 1.5 * Cos(6.28/1296000*dblN))  
  'Цикл задержки  
  For dblT = 1 To 1000 * dblS  
  Next dblT  
  picWatch.Line (0,0)-(2.5*Sin(6.28/360*dblN),  
  2.5 * Cos(6.28 / 360 * dblN)), vbWhite
```

```

picWatch.Line (0, 0)-(2 * Sin(6.28 / 21600
* dblN),
2 * Cos(6.28 / 21600 * dblN)), vbWhite
picWatch.Line (0, 0)-(1.5 * Sin(6.28 /
1296000 * dblN), 1.5 * Cos(6.28 / 1296000 *
dblN)), vbWhite
Next dblN
'Положение стрелок в момент остановки часов
picWatch.Line (0, 0)-(2.5 * Sin(6.28 / 360
* dblN), 2.5 * Cos(6.28 / 360 * dblN))
picWatch.Line (0, 0)-(2*Sin(6.28/21600*dblN),
2 * Cos(6.28 / 21600 * dblN))
picWatch.Line (0, 0)-(1.5 * Sin(6.28 /
1296000 * dblN),1.5*Cos(6.28/1296000*dblN))
picWatch.Circle (0, 0), 0.2
End Sub

```

4 Запустить проект.

«Завести» часы, ввести в текстовое поле число, определяющее продолжительность хода часов.

С помощью ползунка установить скорость хода часов.

Щелкнуть по кнопкам *Регулировка* и *Пуск*.

Настроить скорость хода часов (скорость анимации), которая зависит от быстродействия компьютера, и поэтому положение ползунка, реализующего точный ход, будет на разных компьютерах различным.

5.54. Проект «Стрелочные и электронные часы» хранится в папке *prjZ5\_54*.

- 1 Модернизировать проект «Стрелочные часы» (файл *prjZ5\_53.vbp*). Для создания электронных часов использовать функцию *Time\$*, которая возвращает значение текущего времени типа **String**, которое можно вывести в текстовое поле. Значение времени представляется в виде тройки чисел *#Часы:Минуты:Секунды#*, разделенных знаком «:».

Для периодического обновления значения времени используем объект *Timer*. Объект *Timer* не отображается на форме в процессе выполнения программы и выполняет всего одну функцию — проверяет показания системных часов по событию *Timer*.

Периодичность события *Timer* может быть задана в свойстве *Interval*, измеряемом в миллисекундах (может изменяться от 0 до 65535). Для того чтобы событие *Timer* происходило каждую секунду, необходимо свойству *Interval* присвоить значение 1000.

Разместить на форме объект `Timer` и присвоить ему имя `tmr1`, а свойству `Interval` – значение `1000`. Поместить на форму текстовое поле `txtTime`. Ввести событийную процедуру `tmr1_Timer()`:



```
Sub tmr1_Timer()
txtTime.Text = Time$
End Sub
```

- 2 Синхронизировать стрелочные часы с электронными часами. Связать рисование стрелок с событиями таймера, для этого разместить на форме объект `Timer` и присвоить ему имя `tmr2`, а свойству `Interval` — значение `1000`. Ввести событийную процедуру `tmr2_Timer()`:

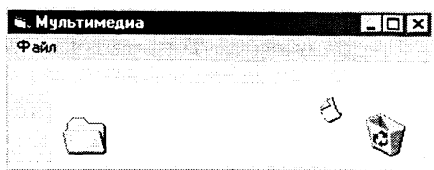
```
Sub tmr2_Timer()
dblN = 1 + dblN
picWatch.Circle (0, 0), 0.2
picWatch.Line (0, 0)-(2.5 * Sin(6.28 * dblN / 60), 2.5 * Cos(6.28 * dblN / 60))
picWatch.Line (0, 0)-(2 * Sin(6.28 * dblN / 3600), 2 * Cos(6.28 * dblN / 3600))
picWatch.Line (0, 0)-(1.5 * Sin(6.28 * dblN / 216000), 1.5 * Cos(6.28 * dblN / 216000))
For dblT = 1 To 50000000
Next dblT
picWatch.Line (0, 0)-(2.5 * Sin(6.28 * dblN / 60), 2.5 * Cos(6.28 * dblN/60)), vbWhite
picWatch.Line (0, 0)-(2 * Sin(6.28 * dblN / 3600), 2 * Cos(6.28 * dblN/3600)), vbWhite
picWatch.Line (0,0)-(1.5*Sin(6.28*dblN/216000),1.5*Cos(6.28*dblN/216000)),vbWhite
End Sub
```

5.55. Проект «Мультимедиа-2» хранится в папке `prjZ5_55`.

- 1 Открыть проект `prj15.vbp`. Удалить с формы кнопки. С помощью редактора меню *Menu Editor* создать меню. Внести изменения в событийные процедуры:

```
Private Sub mnuOpen_Click()
dlg1.ShowOpen
anim1.Open dlg1.filename
End Sub
Private Sub mnuPlay_Click()
anim1.Play
End Sub
Private Sub mnuClose_Click()
anim1.Close
End Sub
Private Sub mnuExit_Click()
End
End Sub
```

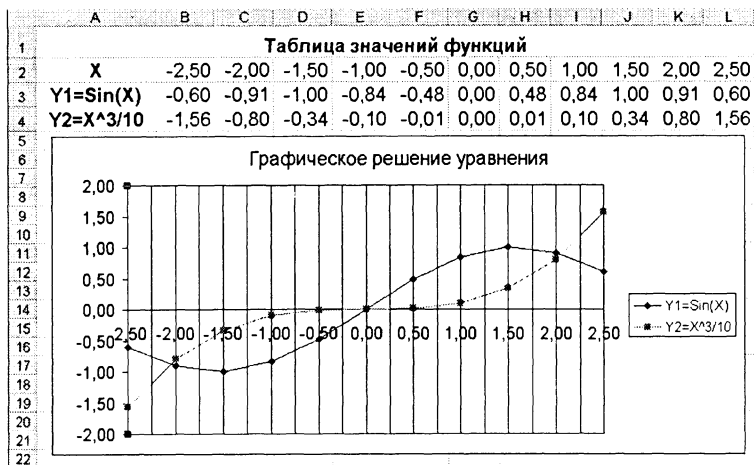
- 2 Запустить проект и открыть с помощью меню AVI-файл. Завершить работу с проектом с помощью меню.



## Ответы к главе 6

### Моделирование и формализация

- 6.1. Графическое решение уравнения  $x^3/10 = \sin x$  в файле Z.xls в каталоге \practicum\inftech\calc\ (лист *Решение уравнений*).
- 1 Ввести формулы функций и заполнить таблицу значений функций на интервале от  $-2,5$  до  $2,5$  с шагом  $0,5$ .
  - 2 Ввести команду [Вставка-Диаграмма...]. На панелях *Автоформат диаграммы* выбрать *Ряд данных в: Строки* и тип диаграммы *Линии с символами*. Будут построены два графика функций, координаты их точек пересечения и будут корнями уравнения:  $x_1 \approx -2$ ,  $x_2 \approx 0$ ,  $x_3 \approx 2$ .

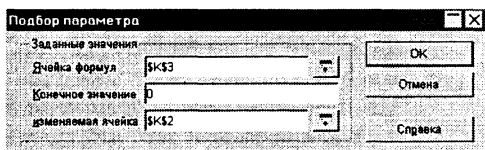


- 6.2. Решение уравнения  $x^3/10 = \sin x$  с заданной точностью в файле Z.xls в каталоге \practicum\inftech\calc\ (лист *Подбор параметра*).
- Метод *Подбор параметра* используется при поиске значения аргумента функции, который обеспечивает требуемое значение функции. При подборе параметра изменяется значение в ячейке аргумента функции до тех пор, пока



значение в ячейке самой функции не возвращает нужный результат. Точность подбора зависит от заданной точности представления чисел в ячейках таблицы.

- 1 Ввести формулы функций и заполнить таблицу значений функций на интервале от  $-2,5$  до  $2,5$  с шагом  $0,5$ . Установить точность представления чисел в ячейках с точностью 4 знака после запятой.
- 2 Построить графики функций. По графикам грубо приближенно можно определить, что уравнение имеет корни  $x_1 \approx -2$ ,  $x_2 \approx 0$ ,  $x_3 \approx 2$ .
- 3 Выделить ячейку, содержащую значение функции наиболее близкое к нулю, например,  $\$K\$3$ . Ввести команду [Сервис-Подбор параметра...].
- 4 На панели *Подбор параметра* в поле *Конечное значение* ввести требуемое значение функции (в данном случае 0).



В поле *изменяемая ячейка* ввести адрес ячейки  $\$K\$2$ , в которой будет производиться подбор значения аргумента.

- 5 На панели *Результат подбора параметра* будет выведена информация о величине подбираемого и подобранного значения.
- 6 В ячейке аргумента K2 появится подобранное значение  $2,0648$ . Повторить подбор параметра для ячейки значения функции S3. В ячейке аргумента C2 появится подобранное значение  $-2,0648$ .

	A	B	C	D	E	F	G	H	I	J	K	L
1		Таблица значений функций										
2	X	-2,5000	-2,0648	-1,5648	-1,0648	-0,5000	0,0000	0,5000	1,0000	1,5000	2,0648	2,5648
3	Y=X*3/10-Sin(x)	-0,9640	0,0002	0,6169	0,7540	0,4669	0,0000	-0,4669	-0,7415	-0,6600	-0,0002	1,1417
4												
5												



- 7 Таким образом, корни уравнения с точностью до четырех знаков после запятой найдены:  $x_1 \approx -2,0648$ ,  $x_2 \approx 0,0000$ ,  $x_3 \approx 2,0648$ .
- 6.3. Проект «Бросание кубика» хранится в папке \practicum\VB\projects\Z6\_3\.
- 1 Поместить на форму графическое поле, в котором на числовой оси  $X$  на отрезке от 0 до 6 будет отображаться процесс случайной генерации точек. Падению кубика на каждую из шести граней поставим в соответствие одинаковые числовые интервалы.
  - 2 Поместить на форму текстовое поле txtN для ввода числа генерируемых точек и поля txt1, txt2, txt3, txt4, txt5 и txt6 для вывода числа точек, попавших в каждый из числовых интервалов.
  - 3 Поместить на форму кнопку и создать для нее событийную процедуру, которая обеспечивает ввод количества генерируемых точек в переменную lngN, генерацию случайных точек и подсчет в переменных lng1, lng2, lng3, lng4, lng5 и lng6 количества точек, попавших в каждый из числовых интервалов:

```

Dim dblX As Double, I, lngN, lng1, lng2,
lng3, lng4, lng5, lng6 As Long
Private Sub cmd1_Click()
lng1 = 0
lng2 = 0
lng3 = 0
lng4 = 0
lng5 = 0
lng6 = 0
lngN = txtN.Text
pic1.Cls
pic1.Scale (0, 1)-(6, -1)
'Генерация точек
For I = 1 To lngN
dblX = 6 * Rnd
pic1.PSet (dblX, 0)
Select Case dblX
Case 0 To 1
lng1 = lng1 + 1
Case 1 To 2
lng2 = lng2 + 1
Case 2 To 3
lng3 = lng3 + 1
Case 3 To 4
lng4 = lng4 + 1

```

```

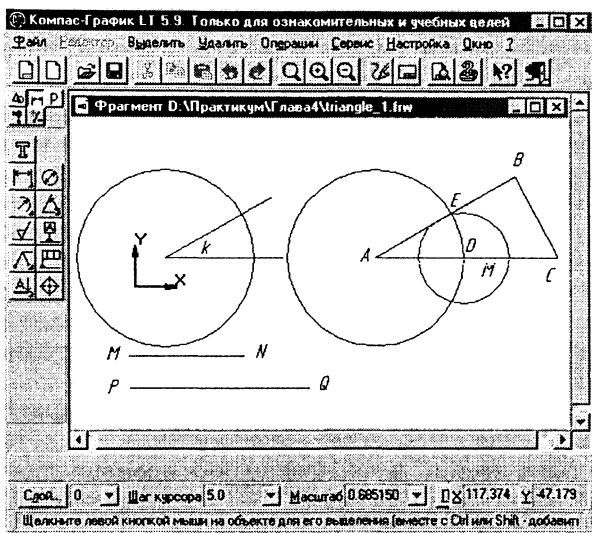
Case 4 To 5
lng5 = lng5 + 1
Case 5 To 6
lng6 = lng6 + 1
End Select
Next I
txt1.Text = lng1
txt2.Text = lng2
txt3.Text = lng3
txt4.Text = lng4
txt5.Text = lng5
txt6.Text = lng6
End Sub

```

- 4 Ввести количество генерируемых точек. После щелчка по кнопке *Пуск* в графическом поле будет отображен процесс генерации случайных точек, а в текстовые поля выведены количества выпадений кубика на каждую из шести граней.

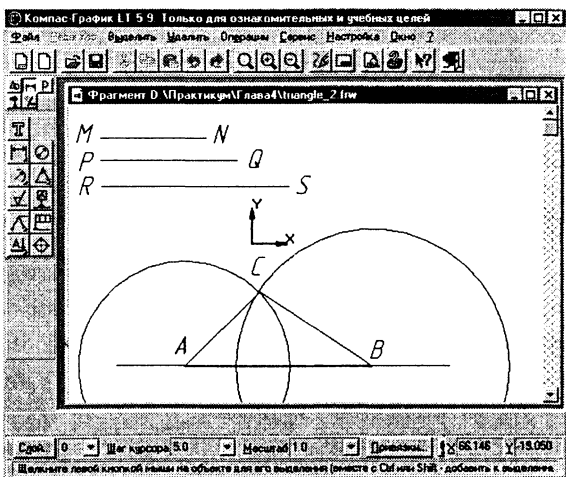


- 5 При увеличении количества генерируемых точек можно наблюдать все меньшее различие в количествах падения кубика на различные грани.
- 6.4. а) Файл `triangle_1.frw` находится в каталоге `\practicum\inf-tech\cad\`.
- 1 Алгоритм выполнения заданного построения следующий:
1. Построить угол  $A$ , равный заданному углу  $K$  (смотри практическое задание 4.9).
  2. Отложить на сторонах угла отрезки, равные заданным отрезкам  $MN$  и  $PQ$ , обозначить их  $AB$  и  $BC$ .
  3. Соединить отрезком точки  $B$  и  $C$ . Треугольник построен.



6.4. б) Файл triangle\_2.frw находится в каталоге \practicum\inf-tech\cad\.

1. Провести прямую и отложить на ней отрезок, равный по длине заданному  $RS$ . Обозначить концы отрезка буквами  $A$  и  $B$ .
2. Построить окружность с центром в точке  $A$  и радиусом, равным длине заданного отрезка  $MN$ .
3. Построить окружность с центром в точке  $B$  и радиусом, равным длине заданного отрезка  $PQ$ .
4. Построить отрезки  $AC$  и  $BC$ . Треугольник построен.



6.5. Компьютерная модель «Бросание тела под углом к горизонту» в файле calc.xls в каталоге \practicum\inftech\calc\ (лист *Бросание тела под углом*).

Вернемся к третьему этапу создания и исследования модели движения тела, брошенного под углом к горизонту. Преобразуем теперь формальную модель в компьютерную с использованием электронных таблиц StarCalc.

Выделим в таблице определенные ячейки для ввода значений начальной скорости  $v_0$  и угла  $\alpha$  и вычислим по формулам 5.1 значения координат тела  $x$  и  $y$  для определенных значений времени  $t$  с заданным интервалом.

Для преобразования значений углов из градусов в радианы используем функцию РАДИАНЫ( ).

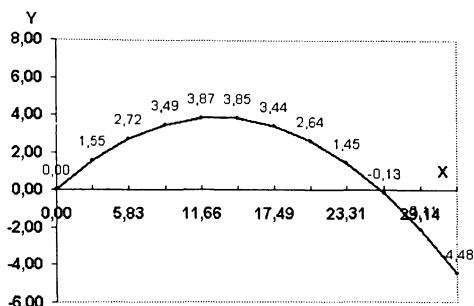
- 1 Для ввода начальной скорости будем использовать ячейку В1, а для ввода угла — ячейку В2.
- 2 Введем в ячейки А5:А16 значения времени с интервалом в 0,2 с.
- 3 В ячейки В5 и С5 введем формулы:  
 $=\$B\$1*\text{COS}(\text{РАДИАНЫ}(\$B\$2))*A5$   
 $=\$B\$1*\text{SIN}(\text{РАДИАНЫ}(\$B\$2))*A5-4,9*A5*A5$

	А	В	С
1	$V_0 =$	17,0 м/с	
2	$\alpha =$	31,0 град	
3			
4	$t$	$x = v_0 \cdot \cos\alpha \cdot t$	$y = v_0 \cdot \sin\alpha \cdot t - g \cdot t^2 / 2$
5	0,00	0,00	0,00
6	0,20	2,91	1,55
7	0,40	5,83	2,72
8	0,60	8,74	3,49
9	0,80	11,66	3,87
10	1,00	14,57	3,85
11	1,20	17,49	3,44
12	1,40	20,40	2,64
13	1,60	23,31	1,45
14	1,80	26,23	-0,13
15	2,00	29,14	-2,11
16	2,20	32,06	-4,48

- 4 Скопируем формулы в ячейки В6:В16 и С6:С16 соответственно.

Визуализируем модель, построив график зависимости координаты  $y$  от координаты  $x$  (траекторию движения тела).

- 5 Построим диаграмму типа *График*, в которой используется в качестве категории диапазон ячеек В5:В16, а в качестве значений — диапазон ячеек С5:С16.



Исследуем модель и определим с заданной точностью 0,1 градуса значения диапазона изменений угла, которые обеспечивают попадание в площадку, находящуюся на расстоянии 25 м и длиной 2 м, при заданной начальной скорости 17 м/с. Воспользуемся для этого методом *Подбор параметра*.

6 Установить для ячеек точность один знак после запятой.

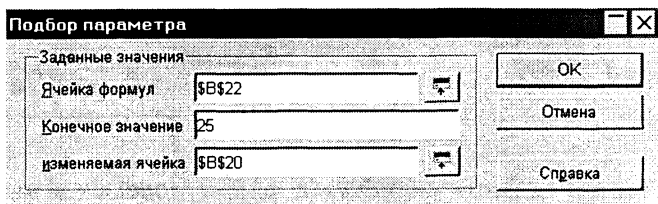
7 Ввести в ячейки В19 и В20 значения начальной скорости  $v_0 = 17$  м/с и угла  $\alpha = 31$  град, а в ячейку В22 формулу для вычисления координаты X мячика для заданных начальных условий:  

$$=B19^2 * SIN(РАДИАНЫ(2 * B20)) / 9,81$$

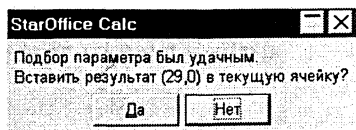
19	$V_0 =$	17,0 м/с
20	$\alpha =$	31,0 град
21		
22	X =	26,0 м

8 Выделить ячейку В22 и ввести команду [Сервис-Подбор параметра...].

На появившейся диалоговой панели ввести в поле *Конечное значение* координату ближнего края площадки — 25. В поле *изменяемая ячейка* ввести адрес ячейки, содержащей значение угла (в данном случае \$B\$20).

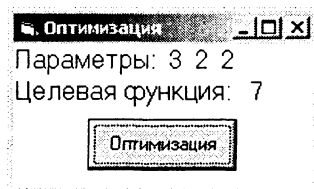


9 После щелчка по кнопке *Да* на появившейся панели *StarOffice Calc* в ячейку В20 будет записано значение 29,0.



- 10 Повторить процедуру подбора параметра для попадания в дальний край площадки, в ячейке B20 получим значение 33,2. Таким образом, существует диапазон значений угла бросания от 29,0 до 33,2 градусов, в котором обеспечивается попадание в площадку.
- 11 Повторить процедуру определения диапазона углов при начальном значении 55 град, получим значения предельных углов 56,8 и 61,0 градуса. С учетом точности вычислений данные для обоих диапазонов углов подтверждают результаты, полученные при исследовании компьютерной модели на языке Visual Basic.
- 6.7. Интерактивная карта России находится по адресу <http://nakarte.ru>.
- 6.8. Геоинформационная модель в файле mapstats.xls хранится в каталоге \textbook\Excel\ (лист *Страны мира*).
- 6.11. Проект «Оптимизационное моделирование» хранится в каталоге \practicum\VB\Projects\Z6\_11\.
- 1 Наборы параметров, при которых выполняются условия ограничений, можно найти перебором вариантов с помощью вложенных циклов:

```
Private Sub cmd1_Click()
For X1 = 0 To 100
For X2 = 0 To 100
For X3 = 0 To 100
If 3 * X1 + 2 * X2 + 1 * X3 = 15 And 1
* X1 + 2 * X2 + 4 * X3 = 15 Then
Print "Параметры: "; X1; X2; X3: Print
"Целевая функция: "; X1 + X2 + X3
Next X3
Next X2
Next X1
End Sub
```



- 6.12. Готовое задание хранится в файле Model.xls в каталоге \practicum\inftech\calc\.
- 6.13. Проект «Управление с автоматической обратной связью» хранится в каталоге \practicum\VB\Projects\Z6\_13\.

## Ответы к главе 7

### Информационные ресурсы Интернета

7.2. 28 800 бит/с.

7.3. 28 800 бит/с.

7.4. Время передачи графического файла.

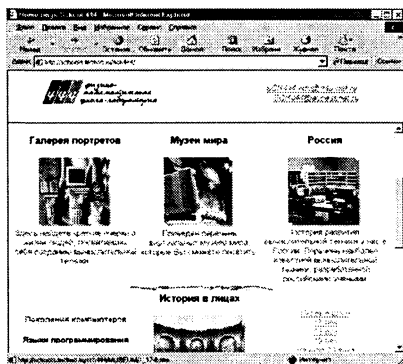
1 Для определения информационного объема графического файла необходимо количество бит на одну точку умножить на количество точек изображения:

$$16 \text{ бит} \cdot 10000 = 160\,000 \text{ бит.}$$

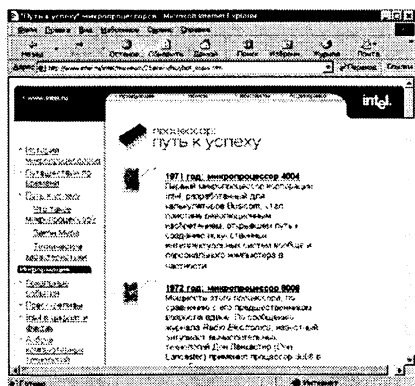
2 Минимальное время передачи будет достигаться при максимально возможной для этого соединения скорости. Для определения времени передачи необходимо информационный объем графического файла разделить на максимальную скорость передачи информации по каналу:

$$160\,000 \text{ бит} : 33\,200 \text{ бит/с} \sim 5 \text{ с.}$$

7.40.



<http://schools.techno.ru/sch444>



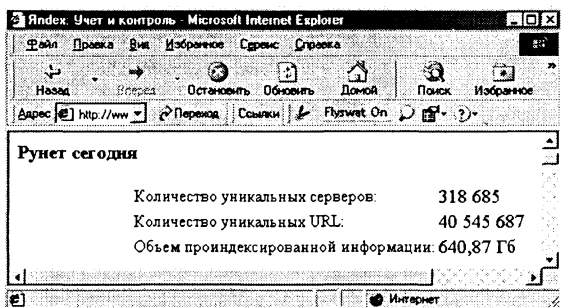
<http://www.intel.ru>



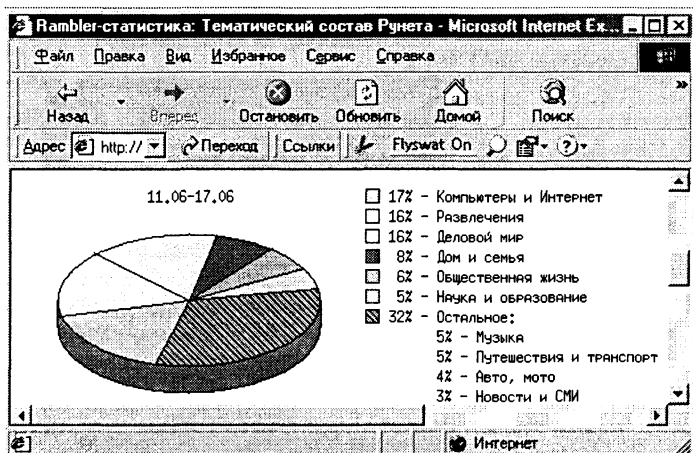


<http://www.computer-museum.ru>

7.41.

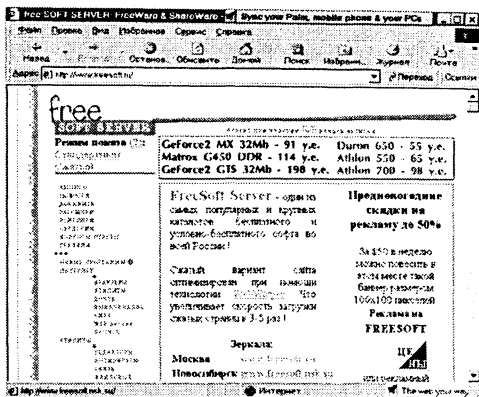


<http://www.yandex.ru/chisla.html>

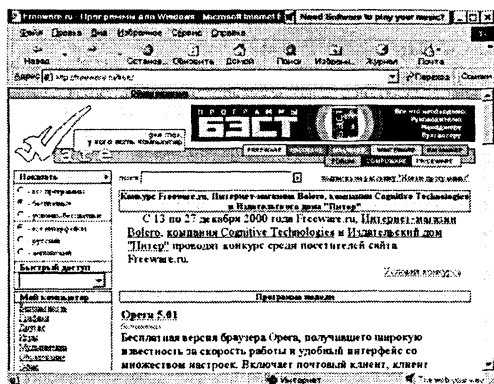


<http://rumetrica.rambler.ru/data/runet/2/>

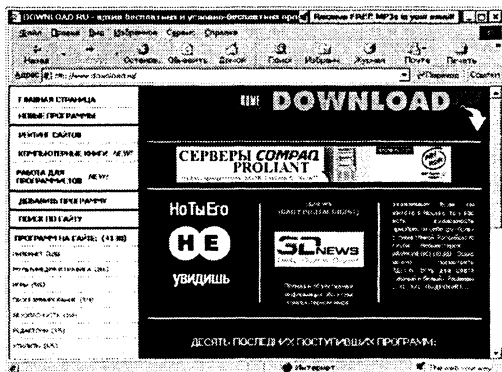
4.



<http://www.freesoft.ru>



<http://www.freeware.ru>



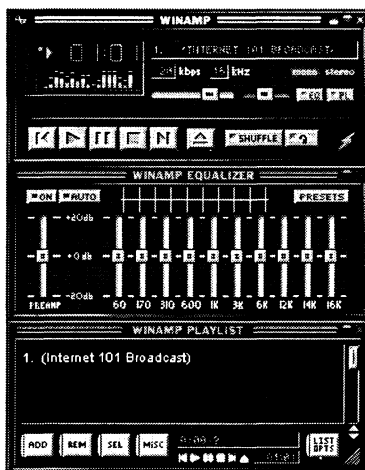
<http://www.download.ru>

7.45.

- 1 5,5 м  
<http://www.sailing.ru/books/mo/nextpart.shtml>
- 2 4,266 г  
<http://www.calc.ru/index.php?module=weight>
- 3 3  
<http://app.ricor.ru/datatour/country/058/058.htm>
- 4 18 лет  
<http://www.tur-gid.ru/news.phtml?id=795&city=2&year=2000&month=11&mday=9>
- 5 Остап-Ибрагим-Берта-Мария-Бендер-бей  
<http://teacher.fio.ru/news.php?n=3128&c=438>
- 6 10  
<http://mega.kemerovo.su/WEB/HTML/2615.HTM>
- 7 3  
[http://www.hermitage.ru/html\\_Ru/08/hm88\\_0\\_2.html](http://www.hermitage.ru/html_Ru/08/hm88_0_2.html)
- 8 Увлекательная игра  
<http://www.lego.ru/company/history.shtml>
- 9 1968 г.  
<http://www.ibo.ru/online/legal-deal/news/2000/10/2/>
- 10 Нет  
<http://www.usrmodem.ru/forum/messages/164/354.html?>

7.48. <http://web.icq.com>

7.49. Открыть в браузере сервер Радио-101 ([www.101.ru](http://www.101.ru)) и выбрать формат WinAmp. Произойдет автоматический запуск проигрывателя WinAmp и начнется прослушивание прямого радиоэфира. Проигрыватель позволяет с помощью графического эквалайзера настраивать частотную характеристику воспроизводимого звука.



- 7.50. Запустить проигрыватель MusicMatch Jukebox.  
 Ввести команду [Options-Settings] и установить флажок *Проигрыватель по умолчанию для файлов формата WMT7*.



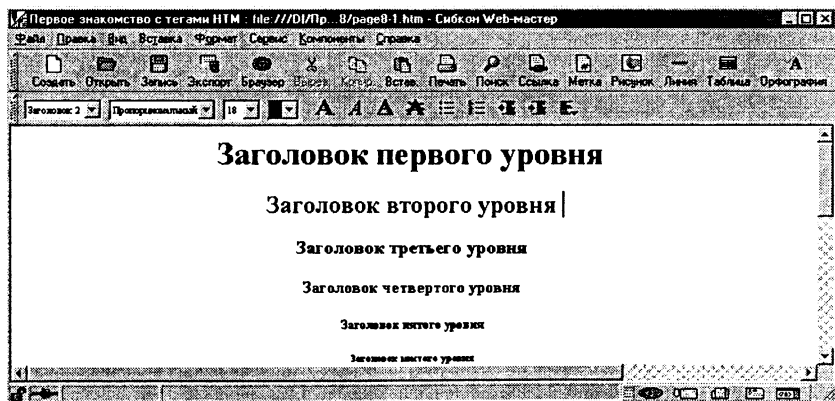
Открыть в браузере сервер «Государственное телевидение России» ([www.ptp.ru](http://www.ptp.ru)) и активизировать ссылку *ТК Вести*.

- 7.51. <http://www.guzei.com/radio/lc.html>

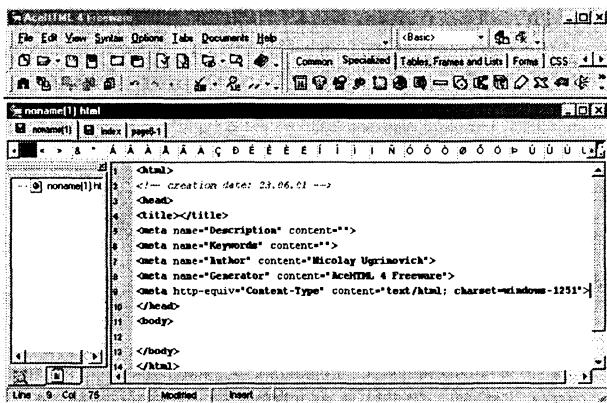
## Ответы к главе 8

### Разработка Web-сайтов

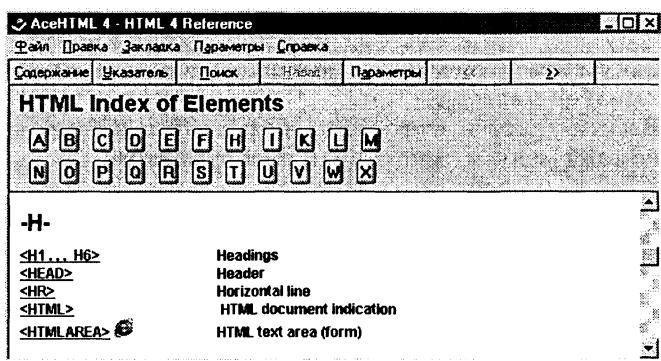
- 8.1. Запустить Web-мастер командой [Сибкон Коммуникатор-Web-мастер].  
 Ввести текст и отформатировать его с использованием команд меню и/или панели инструментов.



- 8.2. Запустить Web-редактор АсеHTML.  
 Ознакомиться с HTML-кодом заготовки Web-страницы, обратить особое внимание на формат мета-тэгов.



- 8.3. Открыть в браузере справочную систему по HTML. Начальная страница системы `index.html` находится на CD-ROM в каталоге `\soft\internet\web\html4spec\`.



В окне появившейся панели активизировать тэг и ознакомиться с его форматом.

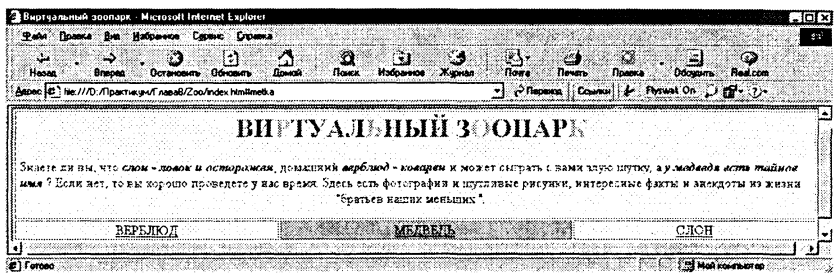
- 8.4. Папка сайта «Виртуальный зоопарк» Zoo находится на CD-ROM в каталоге `\practicum\html\`. Продумать структуру сайта. Например, сайт может состоять из семи страниц, образующих трехуровневую иерархическую структуру. Главная страница сайта `index.txt` содержит ссылки на три страницы второго уровня *Верблюд*, *Медведь*, *Слон*: Верблюд  
Файл страницы — `camel.htm`, рисунка — `camel_foto.gif`, ссылка на страницу-шутку третьего уровня `camel_shut-ka.htm`.

### Медведь

Файл страницы — bear.htm, рисунка — bear\_foto.gif, ссылка на страницу-шутку третьего уровня bear\_shotka.htm.

### Слон

Файл страницы — elephant.htm, рисунка — elephant\_foto.gif, ссылка на страницу-шутку третьего уровня elephant\_shotka.htm.



8.5. Запустить Web-редактор АсеHTML и открыть Web-страницу, в которую будет вставлена карта-изображение.

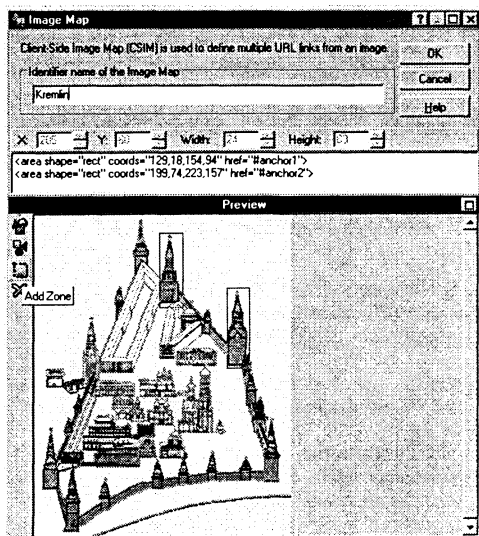
Ввести команду

[Tabs-Specialized-Image Map].

На появившейся панели *Image Map* ввести имя карты-изображения.

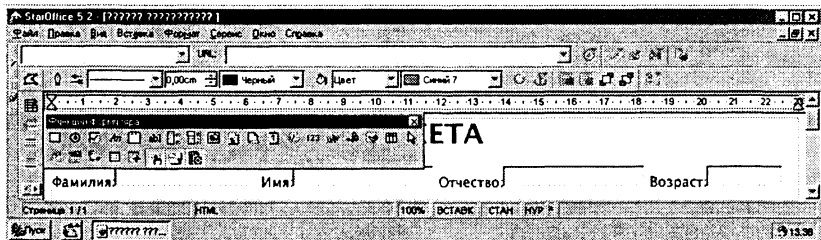
Выбрать графический файл для ее основы щелчком по кнопке *Insert an Image*.

Выделить область-ссылку на изображении мышью.

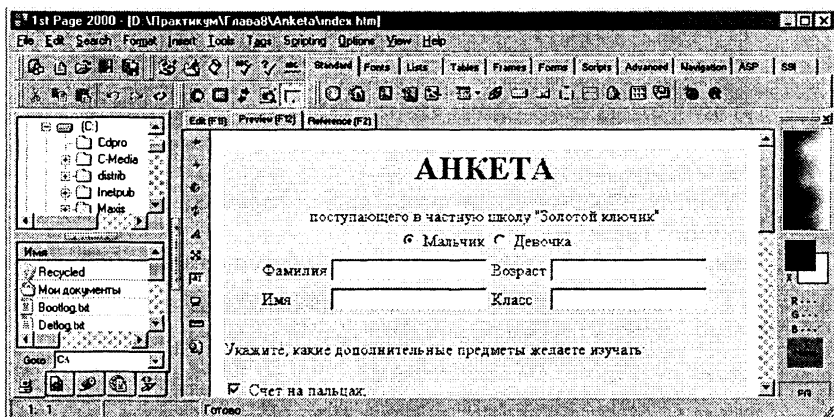


Щелкнуть по кнопке *Add Zone* и на появившейся панели задать адрес ссылки.

### 8.7. Запустить Web-редактор StarOffice Writer. Создать анкету.



### 8.8. Запустить Web-редактор 1st Page 2000. Воспользоваться материалами, хранящимися на CD-ROM в папке Anketa в каталоге \practicum\html\.



Учебное издание

**Угринович Николай Дмитриевич,  
Босова Людмила Леонидовна,  
Михайлова Наталья Ивановна**

**Практикум по информатике и информационным технологиям**  
Учебное пособие для общеобразовательных учреждений

Издание 2-е, исправленное

Художник *Н. Лозинская*  
Компьютерная верстка *В. Носенко*

Подписано в печать 31.05.04. Формат 60x90 <sup>1</sup>/<sub>16</sub>.  
Гарнитура Школьная. Бумага офсетная. Печать офсетная.  
Усл. печ. л. 25,0. Тираж 10000 экз. Заказ 1992

Издательство «Бином. Лаборатория знаний»  
Телефон (095) 955-03-98. E-mail: lbz@aha.ru

Отпечатано с готовых диапозитивов  
в полиграфической фирме «Полиграфист».  
160001, г. Вологда, ул. Челюскинцев, 3