

Министерство образования Российской Федерации
Тамбовский государственный технический университет

Ю. Ю. Громов, О. Г. Иванова, А. В. Лагутин

ИНФОРМАТИКА

Учебное пособие для студентов-иностранцев
подготовительного отделения

Тамбов • Издательство ТГТУ • 2002

УДК 007(075)
ББК $\text{з}973\text{я}73\text{-}1$
И74

Утверждено Редакционно-издательским советом университета

Р е ц е н з е н т

Зав. кафедрой прикладной информатики ТФ МГУКИ,
доктор технических наук, профессор
В. М. Тю т ю н н и к

Ю. Ю. Громов, О. Г. Иванова, А. В. Лагутин

И74 Информатика: Учебное пособие. Тамбов: Изд-во Тамб. гос. техн. ун-та,
2002. 92 с.

ISBN 5-8265-0088-3

Учебное пособие представляет собой часть учебно-методического комплекса по информатике и знакомит иностранных учащихся с основными понятиями данного курса. В первой части учебного пособия рассмотрены основные виды программного обеспечения персонального компьютера, подробно описана работа с ОС Windows и текстовым процессором Microsoft Word. Вторая часть посвящена основам алгоритмизации и алгоритмическим языкам. Пособие содержит адаптированные тексты и упражнения, как обеспечивающие ускоренное формирование навыков связной речи и чтения специальной литературы по курсу информатики, так и позволяющие студентам-иностранцам овладеть основами курса на русском языке. Содержание пособия соответствует программе по информатике на подготовительных факультетах для иностранных граждан и рассчитано на предварительное изучение вводного курса по данной дисциплине и основ научного стиля речи.

Учебное пособие предназначено для студентов-иностранцев, проходящих предвузовскую подготовку.

УДК 007(075)
ББК $\text{з}973\text{я}73\text{-}1$

ISBN 5-8265-0088-3

© Тамбовский государственный
технический университет (ТГТУ), 2002

© Громов Ю. Ю., Иванова О. Г., Лагутин А. В.
2002

Ю. Ю. Громов, О. Г. Иванова, А. В. Лагутин

ИНФОРМАТИКА



• ИЗДАТЕЛЬСТВО ТГТУ •

Учебное издание

ГРОМОВ Юрий Юрьевич,
ИВАНОВА Ольга Геннадьевна,
ЛАГУТИН Андрей Владимирович

ИНФОРМАТИКА

Учебное пособие

Редактор В. Н. Митрофанова
Компьютерное макетирование И. В. Евсеевой

Подписано к печати 24.09.2002.

Гарнитура Times New Roman. Формат 60 × 84/16. Бумага офсетная. Печать офсетная.

Объем: 5,35 усл. печ. л.; 5,28 уч.-изд. л.

Тираж 150 экз. С. 590.

Издательско-полиграфический центр ТГТУ
392000, Тамбов, Советская, 106, к. 14

СОДЕРЖАНИЕ

Часть 1 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА	3
1.1 Классификация программного обеспечения	3
1.2 Операционные системы	5
1.3 Структура операционной системы MS DOS	6
1.4 Операционные системы WINDOWS и UNIX	7
1.4.1 Операционные системы WINDOWS	7
1.4.2 Операционная система UNIX	8
1.5 Основы работы с ОС WINDOWS	9
1.6 Файловые менеджеры	19
1.7 Архиваторы	22
1.8 Транслятор, компилятор, интерпретатор	23
1.9 Системы программирования	24
1.10 Текстовый редактор	26
1.10.1 Стандартные инструменты текстового редактора	27
1.10.2 Работа с текстовым редактором Microsoft Word	28
1.11 Графический редактор	48
1.11.1 Стандартные инструменты графического редактора	49
1.12 Табличный процессор	51
1.13 Системы управления базами данных (СУБД)	53
1.14 Интегрированные пакеты программ	56
Часть 2 АЛГОРИТМЫ. АЛГОРИТМИЗАЦИЯ. АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ	58
2.1 Понятие алгоритма	58
2.2 Основные свойства алгоритма	59
2.3 Представление алгоритмов	59
2.4 Базовые структуры алгоритмов	64
2.5 Итерационные циклы	68
2.6 Вложенные циклы	70
2.7 Языки программирования	71
2.7.1 Классификация языков программирования	71
2.7.2 Машинные языки	72
2.7.3 Алгоритмические языки	74
2.8 Составляющие алгоритмического языка	75
2.8.1 Основные понятия	75
2.8.2 Стандартная функция	76
2.8.3 Арифметические выражения	78
2.8.4 Логические выражения	79
2.9 Упражнения	80
СПИСОК ЛИТЕРАТУРЫ	90

ЧАСТЬ 1

Программное обеспечение компьютера

Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах.

Программное обеспечение современных компьютеров включает миллионы программ – от игровых до научных.

Под *программным обеспечением* (Software) понимают совокупность программ, выполняемых вычислительной системой.

К программному обеспечению (ПО) относится также вся область деятельности по проектированию и разработке ПО, например:

- технология проектирования программ;
- методы тестирования программ;
- анализ качества работы программ;
- документирование программ;
- разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения.

Программное обеспечение – *неотъемлемая часть компьютерной системы*.

1.1 Классификация программного обеспечения

Все программы, работающие на компьютере (рис. 1), можно условно разделить на три категории:

1 прикладные программы, непосредственно обеспечивающие выполнение необходимых пользователям работ;

2 системные программы, выполняющие различные вспомогательные функции, например:

- управление ресурсами компьютера;
- создание копий используемой информации;
- проверка работоспособности устройств компьютера;
- выдача справочной информации о компьютере и др.;

3 инструментальные программные системы, облегчающие процесс создания новых программ для компьютера.

Быстрое развитие вычислительной техники и расширение применения компьютеров ускорили процесс развития программного обеспечения. Если раньше основных категорий ПО было несколько – операционные системы, трансляторы, пакеты прикладных программ, то сейчас ситуация сильно изменилась.

Развитие ПО пошло как вглубь (появились новые подходы к построению операционных систем, языков программирования и т.д.), так и



Рис. 1

вширь (прикладные программы перестали быть прикладными и приобрели самостоятельную ценность).

На сегодняшний день можно сказать, что сложились следующие группы программного обеспечения:

- операционные системы и оболочки;
- системы программирования (трансляторы, библиотеки подпрограмм, отладчики и т.д.);
- инструментальные системы;
- интегрированные пакеты программ;
- электронные таблицы;
- системы машинной графики;
- системы управления базами данных (СУБД);
- прикладное программное обеспечение.

Прикладная программа – это любая конкретная программа, помогающая решению какой-либо задачи.

Прикладные программы могут носить и общий характер, например, обеспечивать составление и печатание документов и т.п.

Прикладные программы могут использоваться либо автономно, то есть решать поставленную задачу без помощи других программ, либо в составе программных комплексов или пакетов.

Системные программы выполняются вместе с прикладными и служат для управления ресурсами компьютера – центральным процессором, памятью, вводом-выводом.

Это программы общего пользования, которые предназначены для всех пользователей компьютера. Системное программное обеспечение разрабатывается так, чтобы компьютер мог эффективно выполнять прикладные программы.

Среди системных программ особое место занимают *операционные системы*, которые обеспечивают управление ресурсами компьютера.

Важными классами системных программ являются также программы вспомогательного назначения – *утилиты*. Они расширяют и дополняют возможности операционной системы. Некоторые виды утилит:

- программы контроля, тестирования и диагностики, которые используются для проверки устройств компьютера и для обнаружения неисправностей; указывают причину и место неисправности;
- программы-драйверы, которые расширяют возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д.; с помощью драйверов возможно подключение к компьютеру новых устройств;

- программы-упаковщики (архиваторы), которые позволяют записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл;

- антивирусные программы, предназначенные для защиты от компьютерных вирусов;

Компьютерный вирус – это специально написанная небольшая по размерам программа, которая может "приписывать" себя к другим программам для выполнения каких-либо вредных действий – портит файлы, "засоряет" оперативную память и т.д.

- программы оптимизации и контроля качества дискового пространства;

- программы восстановления информации, форматирования, защиты данных;

- коммуникационные программы, организующие обмен информацией между компьютерами;

- программы для управления памятью, обеспечивающие более гибкое использование оперативной памяти;

- программы для записи CD-ROM, CD-R и многие другие.

Часть утилит входит в состав операционной системы, а другая часть функционирует независимо от нее, т.е. автономно.

1.2 Операционные системы

Операционная система – это комплекс системных программ для организации взаимодействия пользователя с компьютером и выполнение всех других программ.

, , , , .

Операционная система обычно хранится во внешней памяти компьютера – *на диске*. При включении компьютера она считывается с дисковой памяти и размещается в *ОЗУ*.

Этот процесс называется **загрузкой операционной системы**.

Функции операционной системы:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;

- передача информации между различными внутренними устройствами;

- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

Операционная система скрывает от пользователя сложные подробности взаимодействия с аппаратурой. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера.

В различных моделях компьютеров используют операционные системы с разными возможностями. Для их работы требуются разные ресурсы. Они предоставляют разную степень сервиса для программирования и работы с готовыми программами.

Пользователь взаимодействует с компьютером через внешний интерфейс, организуемый операционной системой. Он вводит задания и получает результаты их выполнения, работая в диалоговом режиме.

Существует два типа диалоговых интерфейсов – текстовый, как, например, в операционной системе MS DOS, и графический, характерный для операционных систем Windows. В графических интерфейсах информация и команды представляются в виде пиктограмм (значков), и пользователь выполняет необходимые действия, указывая на эти значки.

Для текстовых интерфейсов характерна работа с директивами, набранными в командной строке.

Анализ и исполнение команд пользователя, а также загрузку готовых программ из файлов в оперативную память и их запуск, осуществляет *командный процессор* операционной системы.

Для управления внешними устройствами компьютера используются специальные системные программы – *драйверы*. Драйверы стандартных устройств образуют в совокупности *базовую систему ввода-вывода* (BIOS), которая обычно заносится в постоянное ЗУ компьютера.

1.3 Структура операционной системы MS DOS

Операционная система MS DOS (Microsoft Disk Operating System) – самая распространенная ОС на 16-разрядных персональных компьютерах. Она состоит из следующих основных модулей:

- базовая система ввода/вывода (BIOS);
- блок начальной загрузки (Boot Record);
- модуль расширения базовой системы ввода/вывода (IO.SYS);
- модуль обработки прерываний (MSDOS.SYS);
- командный процессор (COMMAND.COM);
- утилиты MS DOS.

Места постоянного размещения этих модулей различны. Так, *базовая система ввода/вывода находится в постоянном запоминающем устройстве* (ПЗУ), а не на дисках, как все остальные модули.

Базовая система ввода/вывода (BIOS) выполняет наиболее простые и универсальные услуги операционной системы, связанные с осуществлением *ввода-вывода*. В функции BIOS входит также *автоматическое тестирование основных аппаратных компонентов* (оперативной памяти и др.) при включении машины и *вызов блока начальной загрузки DOS*.

Блок начальной загрузки (или просто *загрузчик*) – это очень короткая программа, единственная функция которой заключается в считывании с диска в оперативную память двух других частей DOS – модуля расширения базовой системы ввода/вывода и модуля обработки прерываний.

Модуль расширения базовой системы ввода/вывода дает возможность использования *дополнительных драйверов*, обслуживающих новые внешние устройства, а также *драйверов для нестандартного обслуживания внешних устройств*.

Модуль обработки прерываний реализует основные высокоуровневые услуги DOS, поэтому его и называют основным.

Командный процессор DOS обрабатывает команды, вводимые пользователем.

Утилиты DOS – это программы, поставляемые вместе с операционной системой в виде отдельных файлов. Они выполняют действия обслуживающего характера, например, разметку дискет, проверку дисков и т.д.

1.4 Операционные системы Windows и Unix

1.4.1 Операционные системы Windows

В настоящее время большинство компьютеров в мире работают под управлением различных версий операционной среды Windows фирмы Microsoft.

Windows NT (NT – англ. New Technology) – это операционная система, которая использует все возможности новейших моделей персональных компьютеров и работает без DOS. Windows NT – 32-разрядная ОС со встроенной сетевой поддержкой и развитыми многопользовательскими средствами. Она предоставляет пользователям многозадачность, многопроцессорную поддержку, секретность, защиту данных и многое другое. Эта операционная система очень удобна для пользователей, работающих в рамках локальной сети, для коллективных пользователей,

особенно для групп, работающих над большими проектами и обменивающимися данными.

Windows 95 представляет собой универсальную высокопроизводительную многозадачную 32-разрядную ОС нового поколения с графическим интерфейсом и расширенными сетевыми возможностями. *Windows 95* предоставляет пользователю широкие возможности работы с мультимедиа, обработки текстовой, графической, звуковой и видеоинформации.

, , , . *Windows 95* , .

Пользовательский интерфейс *Windows 95* прост и удобен. Эта операционная система предназначена для установки на компьютерах с процессором 486 или Pentium. Рекомендуемый размер оперативной памяти 32 – 128 Мбайт. После включения компьютера и выполнения тестовых программ BIOS операционная система *Windows 95* автоматически загружается с жесткого диска. После загрузки и инициализации системы на экране появляется рабочий стол, на котором размещены различные графические объекты. Пользовательский интерфейс спроектирован так, чтобы максимально облегчить усвоение этой операционной системы новичками и создать комфортные условия для пользователя.

Windows 98 отличается от *Windows 95* тем, что в ней операционная система объединена с браузером Internet Explorer. Кроме этого, в ней улучшена совместимость с новыми аппаратными средствами компьютера, она одинаково удобна как для использования на настольных, так и на портативных компьютерах.

Windows 2000 Professional – операционная система нового поколения для делового использования на самых разнообразных компьютерах – от портативных до серверов. Эта ОС является наилучшей для ведения коммерческой деятельности в Интернете. Она объединяет простоту использования в Интернете, на работе, в пути с надежностью, экономичностью и безопасностью.

1.4.2 Операционная система Unix

Операционная система Unix была создана в Bell Telephone Laboratories. Unix – многозадачная операционная система, способная обеспечить одновременную работу очень большого количества пользователей. Ядро ОС Unix написано на языке высокого уровня C. Это позволяет за считанные месяцы переносить ОС Unix на другие аппаратные платформы и достаточно легко вносить в нее серьезные изменения и дополнения. UNIX является первой действительно переносимой операционной системой. В многочисленные существующие версии UNIX

постоянно вносятся изменения. С одной стороны, это расширяет возможности системы, делает ее мощнее и надежнее, с другой – ведет к появлению различий между существующими версиями.

Сейчас существуют десятки операционных систем, которые можно объединить под общим названием UNIX. В основном, это коммерческие версии, выпущенные производителями аппаратных платформ для компьютеров своего производства.

Причины популярности UNIX:

- код системы написан на языке высокого уровня C, что сделало ее простой для понимания, изменения и переноса на другие платформы. Можно смело сказать, что UNIX является одной из наиболее открытых систем.

- UNIX – многозадачная многопользовательская система. Один мощный сервер может обслуживать запросы большого количества пользователей. Кроме того, система способна выполнять большое количество различных функций, работать, как вычислительный сервер, как сервер базы данных, как сетевой сервер, поддерживающий важнейшие сервисы сети и т.д.

- Наличие стандартов. Несмотря на разнообразие версий UNIX, основой всего семейства являются принципиально одинаковая архитектура и ряд стандартных интерфейсов. Для администратора переход на другую версию системы не составит большого труда, а для пользователей он может и вовсе оказаться незаметным.

- Простой, но мощный интерфейс.

- Использование единой файловой системы.

- Очень большое количество приложений, в том числе свободно распространяемых, начиная от простейших текстовых редакторов и заканчивая мощными системами управления базами данных.

1.5 Основы работы с ОС WINDOWS

ОС Windows рассчитана на работу с мышью; альтернативным способом управления является клавиатура. На экране указатель мыши чаще всего имеет вид широкой стрелки, направленной влево, которая перемещается на экране при движении мыши.

Основные действия с использованием мыши:

- **щелчок мышью** – кратковременное нажатие кнопки мыши (по умолчанию – левой);

- **двойной щелчок** – двойное кратковременное нажатие кнопки мыши;

- перетягивание – **перемещение мыши при нажатой левой кнопке.**

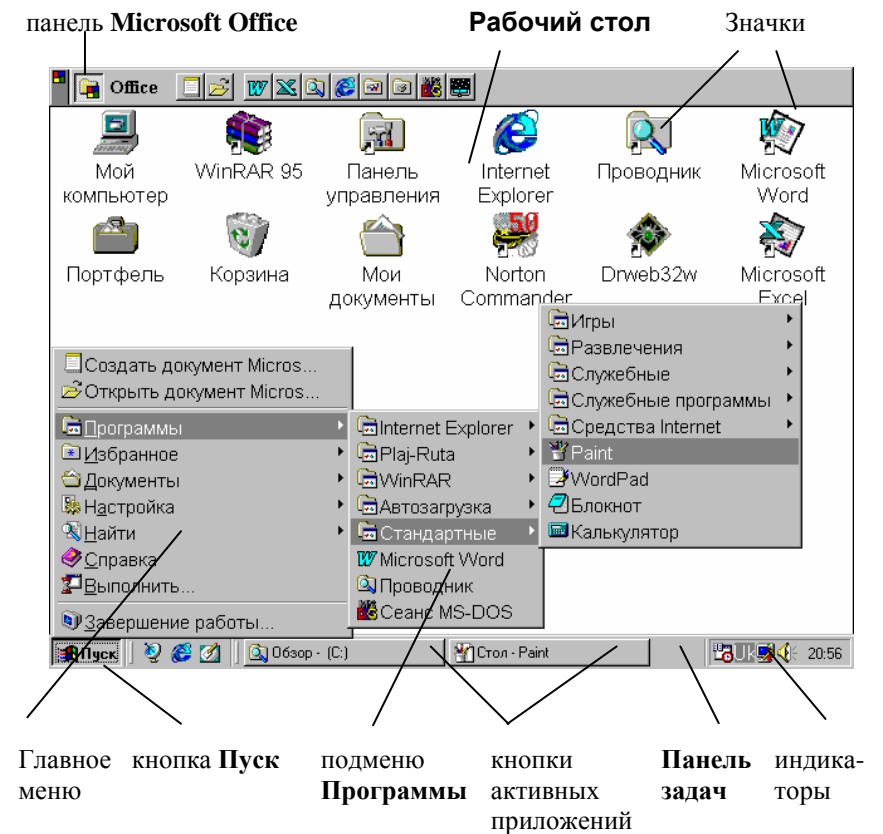



Рис. 2

Основную часть экрана Windows (рис. 2) занимает **Рабочий стол**. На нем располагаются значки **объектов**: папок, дисков, программ и др. Если значок имеет метку , то это значок **ярлыка**. **Ярлык** – ссылка на объект, расположенный не на Рабочем столе, а в какой-то папке. *При двойном щелчке на значке объекта открывается окно этого объекта.* При двойном щелчке на ярлыке открывается окно объекта, на который ссылается ярлык.

Окно – прямоугольная область экрана, в которой выполняются различные Windows-программы. Каждая программа имеет свое окно. Все окна имеют одинаковый состав и структуру.

В состав окна входят следующие элементы (рис. 3):

1 – **заголовок** – верхняя строка окна, в которой находится имя программы или имя окна;

2 – **кнопка сворачивания окна**;

3 – **кнопка восстановления окна** (ее вид зависит от состояния окна);

4 – **кнопка закрытия окна**;

5 – **кнопка системного меню** – открывает системное меню окна;

6 – **строка меню** – содержит команды для управления окном;

7 – **панель инструментов** – содержит кнопки, вызывающие наиболее часто употребляемые команды;

8 – **полосы прокрутки** – позволяют просматривать содержимое окна;

9 – **рабочее поле** – пространство для размещения объектов (текста, рисунков, значков и пр.) и работы с ними;

10 – **строка состояния** – полоса, на которой расположены индикаторы состояния;

11 – **рамка окна**;

Окно может существовать в трех состояниях:

- **полноэкранное** – окно развернуто на весь экран;
- **нормальное** – окно занимает часть экрана;
- **свернутое** – окно в свернутом состоянии.

Изменение состояния окон:

- **сворачивание окна** – щелчок мыши на кнопке сворачивания;
- **из свернутого в предыдущее состояние** – щелчок мыши на кнопке активного приложения на панели задач;
- **из полноэкранного в нормальное и обратно** – щелчок мыши на кнопке восстановления окна.

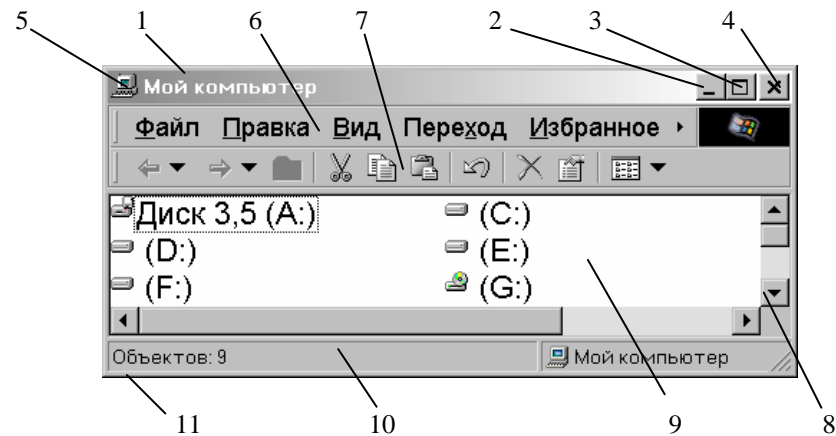


Рис. 3






Для вызова команды из меню необходимо навести указатель мыши на пункт меню и нажать кнопку мыши. Меню откроется и для выбора из него необходимой команды следует щелкнуть мышью на соответствующем пункте. Если после имени команды стоит многоточие, то после ее выбора появится диалоговое окно. Серым цветом изображены команды, которые в данный момент недоступны. Чтобы закрыть меню без выбора команды, необходимо щелкнуть мышью вне области меню или нажать клавишу **Esc**.

Одно из окон является **активным**. Заголовок активного окна выделяется темным цветом, окно выходит на передний план, в нем находится курсор.


Изменение размера окна (в нормальном состоянии) осуществляется перетягиванием рамок окна при нажатой кнопке мыши.

Перемещение окна (в нормальном состоянии) осуществляется перетягиванием заголовка окна при нажатой кнопке мыши.


Полосы прокрутки появляются когда содержимое окна не помещается в его видимой части. Для просмотра содержимого окна есть несколько вариантов:

- щелкать на кнопках  и  на полосах прокрутки;
- перетягивать прямоугольник прокрутки ;
- щелкать мышью между прямоугольником прокрутки и кнопками , .

Закрытие окон: для завершения работы с приложением (программой) необходимо закрыть его окно. Активное окно можно закрыть одним из следующих способов:

- щелкнуть на кнопке закрытия окна ;
- набрать на клавиатуре комбинацию **Alt + F4**;
- выбрать команду **Выход** из меню **Файл**;
- выбрать команду **Закрыть** в системном меню окна (щелкнуть на кнопке системного меню).

Специальные папки Рабочего стола:

Мой компьютер – специальная папка, которая  просматривать содержимое дисков компьютера и выполнять различные операции с файлами и папками (запуск программ, копирование, перемещение, удаление файлов, создание папок и др.).

Корзина – специальная папка, содержащая список всех удаленных файлов и папок.

Портфель – системная папка, которая используется для согласования копий документов, обрабатываемых в различных компьютерах.

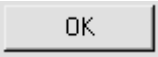




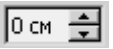
Сетевое окружение – специальная папка, которая используется для просмотра содержимого дисков компьютеров, подключенных к локальной сети, и выполнения различных операций на них.

Панель задач предназначена для запуска приложений и перехода между ними. По умолчанию она находится в нижней части экрана. Панель задач содержит кнопку **Пуск**, кнопки со значками активных приложений и индикаторы. **Активным приложением** называется программа, которая запущена на выполнение. При щелчке на кнопке **Пуск** появляется **Главное меню**. Если пункт меню отмечен стрелкой **▸**, то при наведении на него указателя мыши откроется подменю. Для запуска приложения необходимо щелкнуть на его имени. После этого откроется окно приложения, а на панели задач появится кнопка с его значком. Для перехода между активными приложениями следует щелкать соответствующие кнопки на панели задач.

Также для перехода между активными приложениями используется комбинация **клавиш Alt + Tab**. Нажав и удерживая клавишу **Alt**, следует нажать клавишу **Tab**. В центре экрана появится окно со значками активных приложений. Продолжая удерживать клавишу **Alt** необходимо нажимать **Tab**, пока нужный значок не будет выделен рамкой, после чего отпустить **Alt**.

Диалоговые окна (рис. 4) появляются в случае, когда Windows необходима дополнительная информация для выполнения команды. Они содержат такие элементы:

Запуск программ

	– кнопка закрытия окна с сохранения всех измененных параметров;
	– кнопка закрытия окна без сохранения измененных параметров;
	– кнопка сохранения всех измененных параметров без закрытия окна;
	– кнопка закрытия окна, когда измененные параметры уже сохранены;
	– поле ввода – ограниченная прямоугольной рамкой область, в которую пользователь может вводить с клавиатуры текст; чтобы ввести данные в поле необходимо сначала щелкнуть в нем мышью;
	– счетчик – поле с двумя кнопками справа; можно щелкнуть в него мышью и набрать значения параметра на клавиатуре или щелкать на кнопках: для увеличения параметра ▲ , для уменьшения ▼ ;
	– поле списка – содержат список объектов, доступных для выбора; если содержимое списка не по-

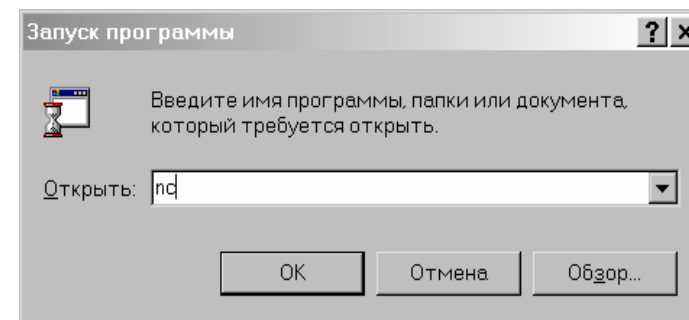
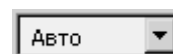


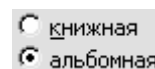
Рис. 4



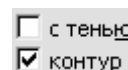
мещается в видимую часть, то появляются полосы прокрутки для просмотра длинных списков; чтобы выбрать объект следует щелкнуть на нем мышью;



– **поле скрытого списка** в видимой части имеет только значение текущего параметра, для их открытия необходимо щелкнуть мышью на кнопке ▼ ;




– **переключатели** – круги с черной точкой или без нее, предназначенные для выбора одного из взаимо-исключающих режимов;



– **флажок** – квадратное поле индикатора с меткой ✓ внутри или без нее, используется для включения/выключения режима (его имя написано рядом), который может находиться во включенном или отключенном состоянии



– **кнопка контекстной справки**, для вызова контекстной справки следует щелкнуть ее, а потом неизвестный элемент.

Корзина – специальная папка, содержащая список удаленных файлов и папок. Для восстановления удаленных файлов и папок их необходимо выделить и в пункте меню **Файл** выбрать команду **Восстановить**. Для удаления файлов и папок их следует выделить и нажать клавишу **Delete** или кнопку . Для удаления из папки **Корзина** всех файлов и папок используется команда **Очистить корзину** из пункта меню **Файл**.

Подготовка к выключению компьютера:

- щелкнуть кнопку **Пуск**;
- выбрать пункт **Завершение работы**;
- в появившемся диалоговом окне выбрать пункт **выключить компьютер**;
- щелкнуть кнопку **ОК**;
- после того, как на экране появится сообщение "**Теперь питание компьютера можно отключить.**" можно выключать компьютер.

Действия при "зависании" компьютера

Иногда во время работы компьютер "зависает", т.е. перестает реагировать на нажатие клавиш. В этом случае необходимо:

- одновременно нажать клавиши **Ctrl + Alt + Delete**;

- в окне **Завершение работы программы** (рис. 10) выделить имя "зависшего" приложения и щелкнуть кнопку **Завершить задачу**;
- если окно **Завершение работы программы** не появилось, нажать кнопку **RESET** на системном блоке;
- если после нажатия кнопки **RESET** компьютер продолжает "зависать" его необходимо выключить и через 30 – 40 с включить.

Проводник Windows

Ярлык программы **Проводник** (Explorer) чаще всего имеет вид, показанный на рис. 5.

В левой части окна программы (рис. 6) отображается дерево папок, в правой – содержимое открытой (текущей) папки. Рядом с именем каждого объекта (диска, папки, файла) стоит его значок, который указывает на тип объекта.



Проводник

Рис. 5

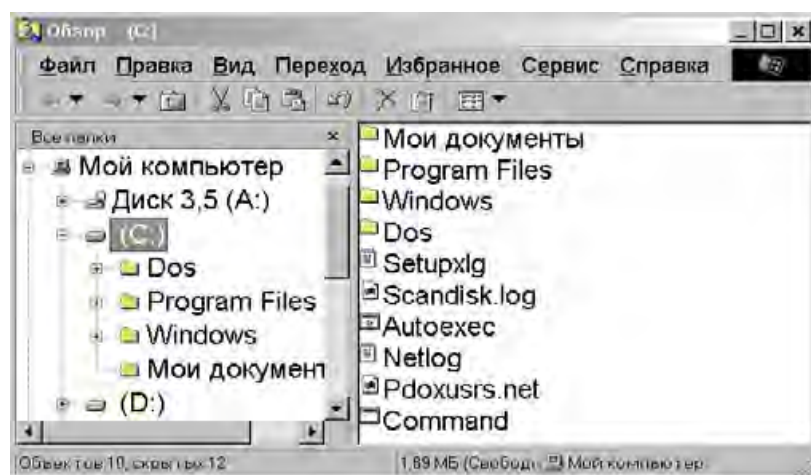


Рис. 6



– значок дисковогода;



– значок диска винчестера;



– значок CD-ROM дисковогода;



– значок папки;





– значок открытой папки (текущей папки);







– папка содержит вложенные папки (подкаталоги) и их структура может быть показана;



– папка содержит вложенные папки и их структура показана.

Нажимая мышью значки ,  можно **показывать и скрывать структуру папок**. Для **открытия папки** следует один раз щелкнуть по значку папки в левой части или дважды в правой. Двойной щелчок по значку папки в левой части открывает папку и показывает ее структуру.

Для **перехода на один уровень вверх** (в папку, которая содержит текущую папку) используется кнопка  на панели инструментов или клавиша **Backspace**. Чтобы перейти на один и более уровней вверх или на другой диск, можно использовать поле списка на панели инструментов **Адрес**. Перейти к предыдущей папке, которая была открыта, можно с помощью кнопки , а возвратиться – .


Для **запуска программы** следует выполнить двойной щелчок на имени файла, который ее содержит. Если дважды щелкнуть по имени любого другого файла, то загрузится программа, которая обрабатывает этот файл, а в ее окне откроется этот файл. Например, если дважды щелкнуть по файлу  **Договор** (документ Microsoft Word), то запустится программа Microsoft Word и в ее окне документ **Договор**.


Выделение файлов и папок



- Для выделения одного файла достаточно щелкнуть мышью на его имени.
- Для выделения блока файлов необходимо щелкнуть в одном углу блока и растянуть выделение до противоположного угла.
 - Для выделения нескольких файлов подряд – выделить имя первого файла, нажать клавишу **Shift** и выделить последний файл из группы.
 - Для выделения нескольких файлов, расположенных отдельно – нажать **Ctrl** и щелкать по именам нужных файлов.
 - Для выделения всех файлов в папке можно использовать команду **Выделить все** меню **Правка**.

Копирование файлов и папок

Первый способ:

- выделить файлы и папки;
- щелкнуть кнопку .

- открыть папку, в которую необходимо скопировать файлы;
- щелкнуть кнопку .

Перемещение файлов выполняется так же, только вместо кнопки  используется кнопка .


Второй способ:


- выделить файлы и папки;
- в левой части окна показать папку, в которую необходимо скопировать файлы;
- перетянуть выделение из правой части окна на имя нужной папки.


Если при перемещении рядом с указателем мыши появится +, то будет выполнена операция **копирования**. Для **перемещения** следует выполнять эту операцию при нажатой клавише **Shift**. Если рядом с указателем мыши знака + не будет, то файлы и папки будут **перемещены**, а для **копирования** необходимо перетягивать их при нажатой клавише **Ctrl**.


Для **переименования файла или папки** следует щелкнуть два раза на имени с большим интервалом. В появившейся рамке с курсором, можно отредактировать имя и нажать **Enter**. Также для переименования можно щелкнуть правой кнопкой мыши на имени файла или папки и в контекстном меню выбрать команду **Переименовать**.

Для **создания папки** необходимо в меню **Файл** или в контекстном меню выбрать пункт **Создать**, а затем пункт **Папка**. В появившейся рамке, ввести имя папки и нажать **Enter**.

Для **создания ярлыка программы** необходимо перетянуть файл, который запускает эту программу (файл с расширением **exe** или **com**) в левую часть окна **Проводник** на имя нужной папки. При перетягивании рядом с указателем мыши появится значок .

Для **удаления файлов и папок** их нужно выделить и нажать клавишу **Delete** или кнопку .

Для **отмены последней операции** необходимо нажать кнопку  или выбрать соответствующий пункт в меню **Правка** или контекстном меню.

Каждый объект (файл, папка, диск и пр.) имеет набор параметров. Для их просмотра и изменения следует выделить этот объект и щелкнуть кнопку  или в контекстном меню объекта выбрать пункт **Свойства**.

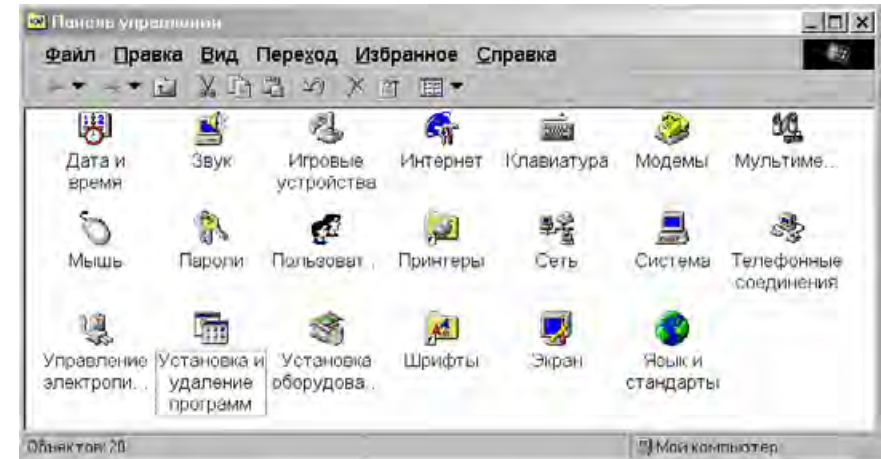


Рис. 7

Панель управления

Панель управления (рис. 7) – один из командных центров Windows, она содержит окна, в которых устанавливаются параметры Windows. Например, с помощью окна **Дата/время** изменяется дата и время на системных часах компьютера. Для вызова Панели управления необходимо:

- щелкнуть кнопку **Пуск**;
- выбрать пункт **Настройка**;
- выбрать пункт **Панель управления**.

Настройка экрана



Экран

Рис. 8

Для изменения параметров экрана следует в Панели управления дважды щелкнуть значок на рис. 8.

Вкладыш **Фон** служит для установления фона **Рабочего стола**. В поле списка **Рисунок рабочего стола** можно выбрать стандартный рисунок. Для выбора других рисунков используется кнопка **Обзор**.

Вкладыш **Заставка** используется для выбора параметров заставки. **Программа-заставка** – программа, которая служит для продления срока эксплуатации монитора. Если несколько минут пользователь не использует клавиатуру и мышь, на экране появляется заставка. В поле списка **Заставка** можно выбрать одну из программ-заставок, а в поле **Интервал** выбирается время от последнего нажатия клавиши до начала работы программы. Для изменения параметров работы программы используется кнопка **Настройка**.

Вкладыш **Оформление** содержит поля, в которых устанавливается цвет, размер и шрифт элементов окон. В поле списка **Элемент** или щелчком на образце в окне следует выбрать элемент окна, а в полях справа – его параметры. В поле списка **Схема** можно выбрать стандартные сочетания цветов.

Во вкладыше **Настройка** в рамке **Область экрана** выбирается разрешающая способность монитора – количество точек на экране. Дополнительные параметры экрана устанавливаются после щелчка на кнопке **Дополнительно**. В поле списка **Размер шрифта** можно выбрать размер шрифта элементов экрана. Если установить флажок **Вывести значок настройки на панель задач**, то на панели задач появится значок окна настройки экрана. Двойной щелчок на значке вызывает окно настройки экрана. Нажатие правой кнопки мыши на этом значке вызывает меню, в котором можно выбрать разрешающую способность монитора.

1.6 Файловые менеджеры

Давным-давно, когда никаких графических оболочек не существовало, а была лишь командная строка системы DOS, Питер Нортон создал программу, позволяющую работать в файловой системе без специальных команд, через систему меню и "горячих" клавиш – Norton Commander или NC. После NC стали появляться альтернативные программы, отличающиеся большими возможностями и функциональностью, например, программа FAR.

В любом файловом менеджере (ФМ) можно выполнить команды операционной системы, вводимые в *командной строке*.

Для запуска ФМ в ОС Windows достаточно щелкнуть на значке программы. После запуска ФМ появляются две панели: левая и правая, в которых видны имена файлов и каталогов (рис. 9).

Активной считается панель, в которой находится цветной прямоугольник (*указатель*).

Выход из ФМ осуществляется нажатием клавиши F10-выход или с помощью кнопки закрытия окна [x].

Каталог – это место на диске, имеющее имя, внутри которого могут находиться программы или файлы, объединенные по какому-либо признаку, а также другие каталоги.

Например: каталог *Program files* – в Windows служит для размещения установленных программ, *DOS* – место на диске, где хранятся файлы, относящиеся к операционной системе MSDOS.

Каждый каталог имеет свое имя.

В нижней части экрана находится *подсказка* по функциональным клавишам F1-F10.

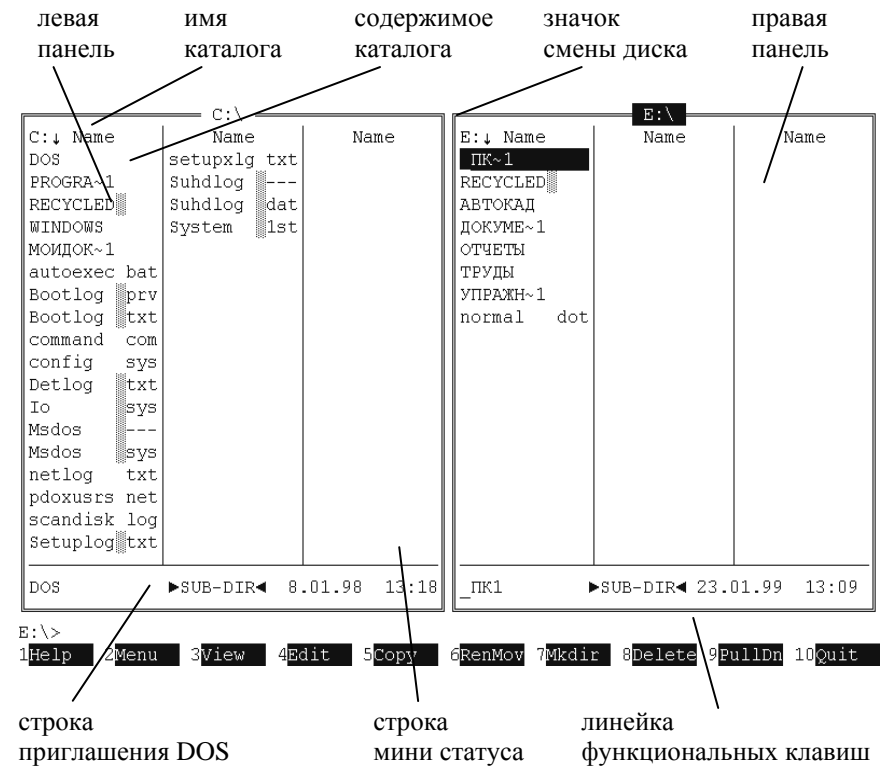


Рис. 9

Между подсказкой и окнами расположена *командная строка*, в которой появляется приглашение *ОС*, состоящее из имени текущего диска и знака \>, например: *C:\>*.

Для запуска программ (приложений) нужно выполнить следующие шаги:

1. войти в каталог, где расположена программа,
2. установить цветной указатель на имя нужного файла,
3. нажать клавишу <ENTER>.

Большинство ФМ заменяют трудоемкий процесс написания команд выбором из возможных вариантов с помощью:

- сочетания клавиш (горячие клавиши),
- функциональные клавиши,
- подсказка и верхнее меню.

:

F1- помощь по программе,

F2-меню пользователя (часто выполняемые операции можно вывести в это меню и выполнять нажатием одной клавиши),

F3-просмотр файла,

F4-редактирование (изменение) файла,

F5-копирование (перезапись с одного диска на другой, из одного каталога в другой, на печатающее устройство),

F6-переименование (изменение имени) или перенос,

F7-создание нового каталога,

F8-удаление файла или каталога,

F9-переход в старшее меню (установка режимов работы ФМ),

F10-выход (завершение работы).

При нажатии любой функциональной клавиши появляется диалоговое окно, в котором сообщается о режиме работы и предлагается выбрать нужные действия из нескольких вариантов. Выбор действия возможен как с клавиатуры (клавиши управления курсором и <Tab>), так и мышью.

Выбор одного файла производится с помощью клавиш управления курсором или с помощью "мышки". При этом вы устанавливаете цветной указатель на имя файла.

Переход из левой панели (окна) в правую (и наоборот) осуществляется нажатием клавиши <TAB>.

Вход в каталог производится установкой цветного указателя на название каталога и нажатия клавиши <ENTER> или двойного щелчка левой кнопки мышки.

Для выхода из каталога устанавливаем "прямоугольник" (указатель) в верхнюю часть списка на знаки ".." и нажимаем клавишу <ENTER>.

Выбор группы файлов можно реализовать несколькими способами:

1 Установив с помощью клавиш управления курсором цветной указатель на нужный файл, нажать <INSERT>, на экране этот файл поменяет цвет (окрасится) и так далее для других файлов, которые вы хотите выбрать.

2 Нажатием клавиши **серый "+"** можно выбрать или все файлы, находящиеся на диске или в каталоге, или же только некоторые из них, согласно шаблону (маске).

3 Нажатие клавиши **серый "-"** отменит выбор файлов.

4 Выделение файлов чаще всего используются при копировании, переносе, удалении или установке атрибутов.

Выбор текущего диска производится нажатием клавиш <Alt> + <F1> для левой панели и <Alt> + <F2> для правой панели соответственно.

В диалоговом окне появятся имена доступных дисков. Нужный диск выбирается клавишами управления курсором или нажатием буквы с именем диска.

Если вы не хотите выбирать диск, отмените эту операцию, нажав <ESC>.

Наиболее часто используемые сочетания клавиш:

CTRL + Y – удаление строки;

CTRL + O – убрать панели с экрана;

CTRL + E – вывести в командной строке предыдущую(ие) команды, которые набирались в командной строке;

CTRL + L – выдача в соседнем окне информации о текущем диске;

CTRL + F1 – убрать левую панель;

CTRL + F2 – убрать правую панель;

ALT + F8 – просмотр списка ранее дававшихся команд и выбор любой из них;

ALT + F7 – поиск файла на диске по начальной букве или полному имени;

ALT + F10 – дерево каталогов;

SHIFT + F4 – редактирование файла, при этом имя файла запрашивается. Этот режим можно использовать для создания (открытия) нового файла;

SHIFT + F9 – запись текущего состояния панелей (сохранение настроек).

Команды используют принцип "выключателя": первое нажатие включает тот или иной режим, повторное нажатие отменяет его (возвращает в исходное состояние).

1.7 Архиваторы

Несмотря на то, что технологии хранения информации постоянно совершенствуются, сами данные вырастают в пропорциональных объемах (видео, графика, звук). Для того чтобы уменьшить объем занимаемого этими данными места, необходимо использовать различные методы компрессии (сжатия, архивации) данных. Существуют программы, которые выполняют эти задачи.

Программы, позволяющие сжимать объём файлов называются **архиваторами**.

Программы, возвращающие файлы из сжатого состояния называются **разархиваторами**.

Наиболее распространены универсальные архиваторы, позволяющие как помещать файлы в архив, так и извлекать их. Например: ARJ, ZIP, RAR и др.

Наиболее популярным средством для решения подобных задач в ОС Windows считается программа WinZip, пример рабочего окна которой приведен на рис. 10.

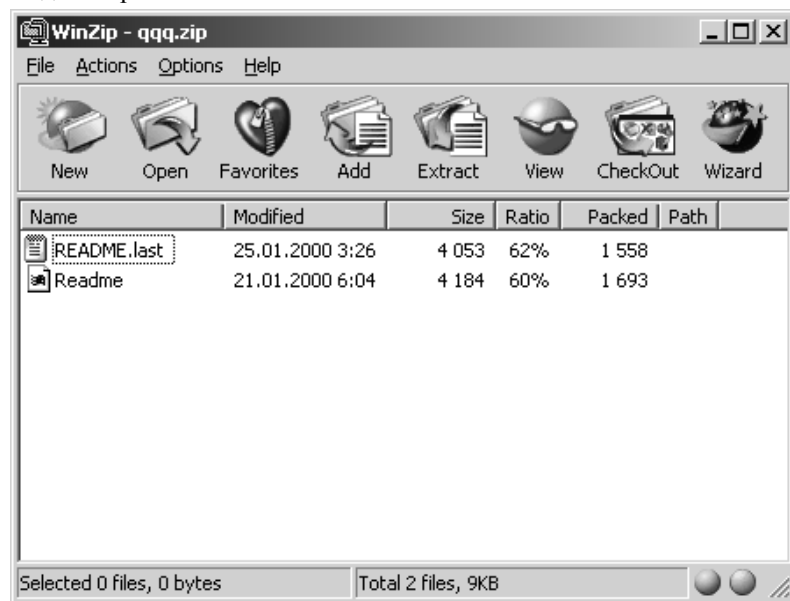


Рис. 10

Архивный файл – набор из одного или нескольких файлов, помещенных в сжатом виде в единый файл, имеющий расширение по имени программы архиватора, например: photo4.arj – архив файлов фотоизображений, созданный с помощью архиватора *arj*.

Применение архиваторов позволяет сэкономить место при хранении и переносе архивных копий, а также объединять файлы, относящиеся к одной программе, в один архив. При этом возможна архивация вложенных каталогов.

1.8 Транслятор, компилятор, интерпретатор

Транслятор (англ. *translator* – переводчик) – это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются.

Компилятор (англ. *compiler* – составитель, собиратель) читает всю программу *целиком*, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор (англ. *interpreter* – истолкователь, устный переводчик) переводит и выполняет программу *строка за строкой*.

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново *переводиться* на машинный язык при каждом очередном запуске программы.

Откомпилированные программы работают быстрее, но *интерпретируемые* проще исправлять и изменять.

Каждый конкретный язык ориентирован либо на компиляцию, либо на интерпретацию – в зависимости от того, для каких целей он создавался. Например, Паскаль обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью *компилятора*.

С другой стороны, Бейсик создавался как язык для начинающих программистов, для которых построчное выполнение программы имеет неоспоримые преимущества.

Иногда для одного языка имеется *и компилятор, и интерпретатор*. В этом случае для разработки и тестирования программы можно воспользоваться интерпретатором, а затем откомпилировать отлаженную программу, чтобы повысить скорость ее выполнения.

1.9 Системы программирования

Система программирования – это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям **мощные и удобные средства разработки программ**. В них входят:

- компилятор или интерпретатор;
- интегрированная среда разработки;
- средства создания и редактирования текстов программ;
- обширные библиотеки стандартных программ и функций;
- отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- "дружественная" к пользователю диалоговая среда;

- многооконный режим работы;
- мощные графические библиотеки; утилиты для работы с библиотеками;
- встроенная справочная служба;
- другие специфические особенности.

Популярные системы программирования – ***Turbo Basic, Quick Basic, Turbo Pascal, Turbo C.***

В последнее время получили распространение системы программирования, ориентированные на создание *Windows-приложений*:

- пакет ***Borland Delphi*** (Дельфи) – блестящий наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.

- пакет ***Microsoft Visual Basic*** – удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Содержит инструментарий для создания *диаграмм и презентаций*.

- пакет ***Borland C++*** – одно из самых распространённых средств для разработки DOS и Windows приложений.

В табл. 1 для иллюстрации приведены на языках Бейсик, Паскаль и Си программы решения одной и той же простой задачи – вычисления суммы S элементов одномерного массива $A = (a_1, a_2, \dots, a_n)$.

Язык Паскаль был разработан в 1970 г. **Н. Виртом** как язык обучения студентов программированию. Паскаль вырабатывает навыки соблюдения хорошего строгого стиля программирования, упрощающего разработку сложных программ (табл. 1).

Таблица 1

Программа на Бейсике	Программа на Паскале	Программа на СИ
<pre> INPUT "N = "; N : DIM A(N) FOR I = 1 TO N PRINT "A("; I; ") ="; INPUT A(I) NEXT I S = 0 FOR I = 1 TO N S = S + A(I) NEXT I </pre>	<pre> Program Summa; Type Mas = Array [1 .. 100] of Real; Var A : Mas; i, n: Integer; S : Real; BEGIN Write('n = '); ReadLn(n); For i : = 1 to n do begin </pre>	<pre> # include <stdio.h> # include <conio.h> main() { float a[100], s; int i, n; clrscr(); printf("n="); scanf("%i", &n); for (i = 1; i <= n; i++) { printf("a[%i]=", i); scanf("%f", &a[i]);} </pre>

PRINT "Сумма ="; S END	Write('A', i, ' ') = '); ReadLn(A [i]); end; S := 0; For i := 1 to n do S := S + A[i]; WriteLn('S = ', S:8:2); END.	s = 0; for (i = 1; i <= n; i++) s = s + a[i]; printf("s = % f \ n", s); return 0; }
---------------------------	---	--

В своем первоначальном виде Паскаль имел довольно ограниченные возможности, но расширенный вариант этого языка – **Turbo Pascal**, является очень мощным языком программирования. Интегрированная оболочка Turbo Pascal, разработанная фирмой **Borland** (ныне Inprise), включающая в себя редактор, компилятор, компоновщик и отладчик, вместе с интерактивной справочной системой сделали разработку программ на Паскале делом простым и приятным.

Язык Си разработан **Деннисом Ритчи** в 1972 г. как язык, пригодный для программирования новой операционной системы UNIX.

Операционные системы ради повышения скорости работы традиционно писались на языке низкого уровня – ассемблере, но язык Си настолько хорошо зарекомендовал себя, что на нем было написано более 90 % всего кода ОС UNIX. Язык СИ обрел популярность как, так называемый, язык среднего уровня, в котором удобство, краткость и мобильность языков высокого уровня сочетаются с возможностью непосредственного доступа к аппаратуре компьютера, что обычно достигаются только при программировании на языке Ассемблера.

Си не очень прост в изучении и требует тщательности в программировании, но позволяет создавать сложные и весьма эффективные программы.

1.10 Текстовый редактор

Текстовый редактор – это программа, используемая специально для ввода и редактирования текстовых данных.

Этими данными могут быть программа или какой-либо документ или книга. Редактируемый текст выводится на экран, и пользователь может в диалоговом режиме вносить в него свои изменения.

, :

- редактирование строк текста;
- возможность использования различных шрифтов символов;

- копирование и перенос части текста с одного места на другое или из одного документа в другой;
- контекстный поиск и замена частей текста;
- задание произвольных межстрочных промежутков;
- автоматический перенос слов на новую строку;
- автоматическая нумерацию страниц;
- обработка и нумерация сносок;
- выравнивание краев абзаца;
- создание таблиц и построение диаграмм;
- проверка правописания слов и подбор синонимов;
- построение оглавлений и предметных указателей;
- распечатка подготовленного текста на принтере в нужном числе экземпляров и т.п.

Возможности текстовых редакторов различны – от программ, предназначенных для подготовки небольших документов простой структуры, до программ для набора, оформления и полной подготовки к типографскому изданию книг и журналов (издательские системы).

Наиболее известный текстовый редактор – *Microsoft Word*. Полнофункциональные издательские системы – *Microsoft Publisher*, *Corel Ventura* и *Adobe PageMaker*. Издательские системы незаменимы для компьютерной верстки и графики. Они значительно облегчают работу с многостраничными документами, имеют возможности автоматической разбивки текста на страницы, расстановки номеров страниц, создания заголовков и т.д. Создание макетов любых изданий – от рекламных листков до многостраничных книг и журналов – становится очень простым, даже для новичков.

1.10.1 Стандартные инструменты текстового редактора

При работе с редактором текстов роль бумаги играет экран компьютера, а роль карандаша и резинки – **курсор**: мигающая вертикальная полоска |. Он указывает место, в которое будет вводиться текст. Для его перемещения используются клавиши управления курсором или мышь. Для перемещения курсора с помощью мыши следует установить указатель мыши в нужную позицию и щелкнуть клавишей мыши.

Курсор можно перемещать в любое место экрана. Текст можно просматривать ("прокручивать"), используя клавиши управления курсором (это уже известные вам четыре клавиши с нарисованными стрелочками).

Процесс просмотра информации (в частности, текста) на компьютере с помощью прокрутки, называется "**скроллингом**".

Текст можно раздвигать, вставляя новые слова. Можно стирать отдельные буквы и переставлять целые абзацы, автоматически заменять во всем тексте одно слово другим.

Для работы с текстовым редактором используются стандартные *клавиши редактирования*.

Используя клавиши *вставки* <Insert> и *удаления* символов, необходимо иметь ввиду следующее. В электронном тексте присутствуют невидимые символы, которые воспринимаются компьютером точно так же, как и самые обычные буквы.

Один из таких символов – это *пробел*. Невидимым символом является и *символ табуляции*. Третий невидимый символ – "*перевод строки*". Он вставляется в текст тогда, когда нажимается клавиша <ENTER>. Многие редакторы самостоятельно делают перенос слов на новую строку и трактуют этот символ, как начало нового абзаца.

Большинство редакторов текстов автоматически следят за длиной строки и в нужный момент делают либо перенос слова на новую строку, либо его переброску (то есть слово целиком перемещается на новую строку). Многие редакторы текстов умеют автоматически разбивать текст на страницы и нумеровать их. Они следят за размером полей и выравнивают текст.

Обычно можно задать четыре типа выравнивания:

- *по левой* границе (обычный тип для не очень больших деловых документов);
- по обеим границам "*по ширине*" (для больших "читаемых" текстов);
- *по правой* границе (как правило, для специальных оформительских целей);
- *по центру* (обычно применяется для заголовков).

В случае выравнивания по обеим границам текстовый редактор равномерно растягивает пустые места между словами (добавляет между ними пробелы), добиваясь красивого расположения текста.

, :

• стандартная машинописная страница имеет размеры 210 × 297 мм. Такой формат называется А4;

• страница может иметь вертикальную ориентацию, именуемую "*PORTRET*" или (после поворота на 90°), горизонтальную или альбомную, так называемую "*LANDSHAFT*";

• стандартная строка на такой странице в книжной ориентации содержит примерно 62 – 65 символов при использовании шрифта, похожего на шрифт пишущей машинки;

• необходимо помнить и о полях: 2,5 – 3 см слева, и 2 см справа и сверху, 1,5 – 2 см снизу.

Для создания разнообразных надписей используются **шрифты** различной формы. Шрифты имеют размер, называемый *кеглем* (это цифры 10, 12 и т.д., рядом с названием шрифта).

Изменяя размер и форму шрифта, можно добиваться самых различных эффектов. Однако наличие на одной странице большого количества шрифтов ухудшает восприятие текста. Выбор шрифта очень сильно зависит от характера текста и предполагаемого читателя.

Текстовый редактор позволяет вставлять внутрь документа таблицы, оперативно управляя количеством строк и столбцов. При необходимости горизонтальные и вертикальные линии можно сделать видимыми, а часть ячеек выделить штриховкой или цветом.

1.10.2 Работа с текстовым редактором Microsoft Word

Более подробно рассмотрим работу с текстовым редактором на примере наиболее распространенного продукта – редактора Microsoft Word 2000.

Значок Microsoft Word 2000 представлен на рис. 11.

Многооконая организация Microsoft Word позволяет одновременно работать с несколькими документами, каждый из которых расположен в своем окне. При введении и редактировании текста пользователь работает с активным документом в активном окне. Для перехода к окну другого документа необходимо щелкнуть на его имени на панели задач или в меню **Окно**, которое содержит список открытых документов.

Под заголовком окна находится строка меню, через которую можно вызвать любую команду Microsoft Word. Для открытия меню необходимо щелкнуть мышью на его имени (рис. 12).



Рис. 11

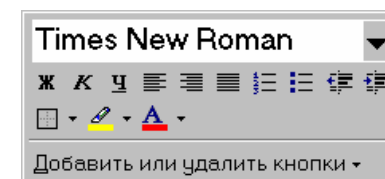


Рис. 13

, . , . - , . , , - (. 13).

Обычно под строкой меню находятся две панели инструментов – **Стандартная** и **Форматирование**.

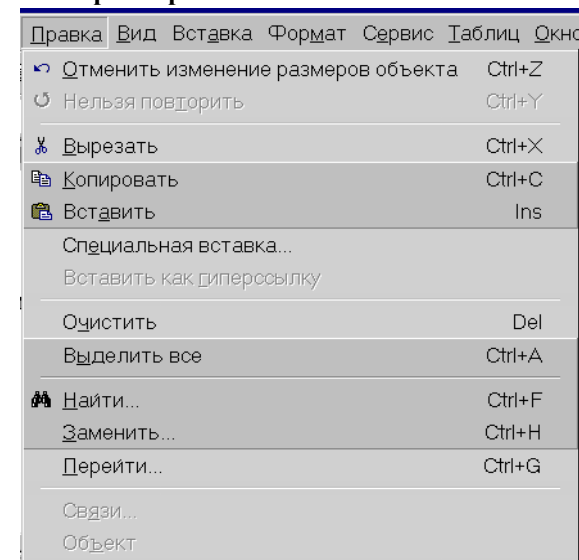


Рис. 12

Горизонтальная координатная линейка расположена над рабочим полем, **вертикальная** – слева от рабочего поля. С их помощью можно устанавливать поля страниц, абзацные отступы, изменять ширину столбцов и устанавливать позиции табуляции. По умолчанию координатная линейка градуирована в сантиметрах.

Для завершения работы с Microsoft Word необходимо закрыть окно программы (кнопка закрытия окна **X**, или комбинация клавиш **Alt + F4**).

Операции с документами



Для создания нового документа следует в меню **Файл** выбрать команду **Создать**. В открывшемся диалоговом окне (рис. 15) выбрать сначала вкладку, а затем шаблон, на основе которого будет создан документ, после чего щелкнуть кнопку **ОК**. Обычные документы создаются на основе шаблона **Новый документ**. Для создания документа на основе шаблона **Новый документ** используется кнопка .



Рис. 14

Для открытия существующего документа необходимо в меню **Файл** выбрать команду **Открыть** или щелкнуть кнопку , после чего откроется диалоговое окно **Открытие документа** (рис. 16). В поле **Папка** выбирается диск, на

котором находится нужный документ. В поле, которое расположен ниже, выбрать (двойным щелчком) папку с документом и сам документ. Документы Microsoft Word имеют расширение doc и значки, показанные на рис. 14.

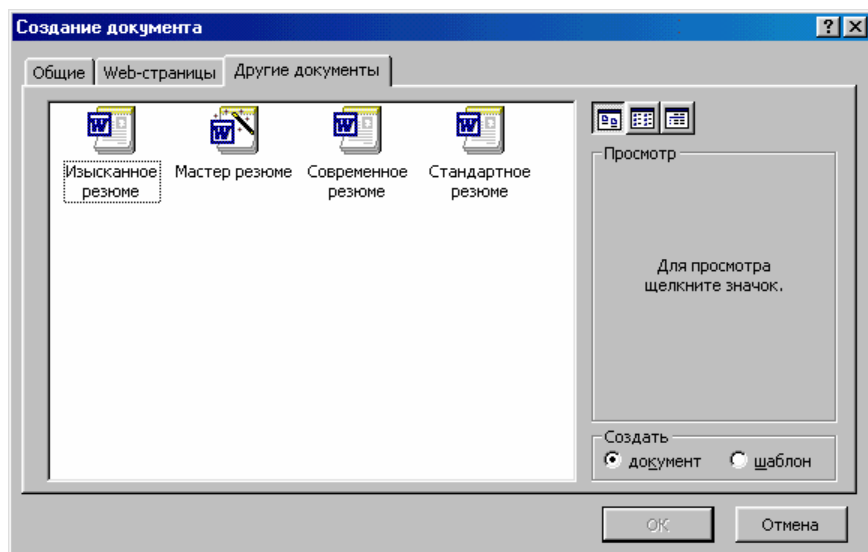


Рис. 15

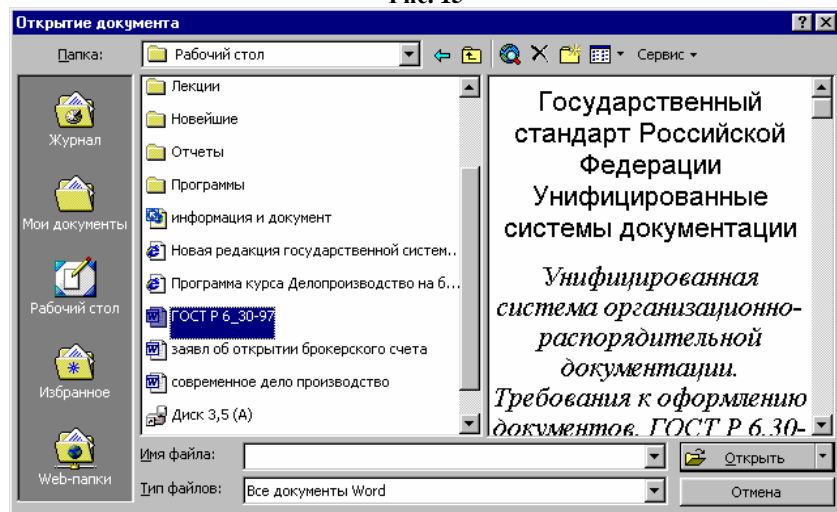



Рис. 16

По умолчанию в поле списка выводятся только файлы с документами Microsoft Word. Для вывода других типов файлов или всех файлов необходимо выбрать соответствующий тип в поле **Тип файлов**.

Для сохранности документа необходимо вызывать команду **Сохранить** меню **Файл** или щелкнуть кнопку .

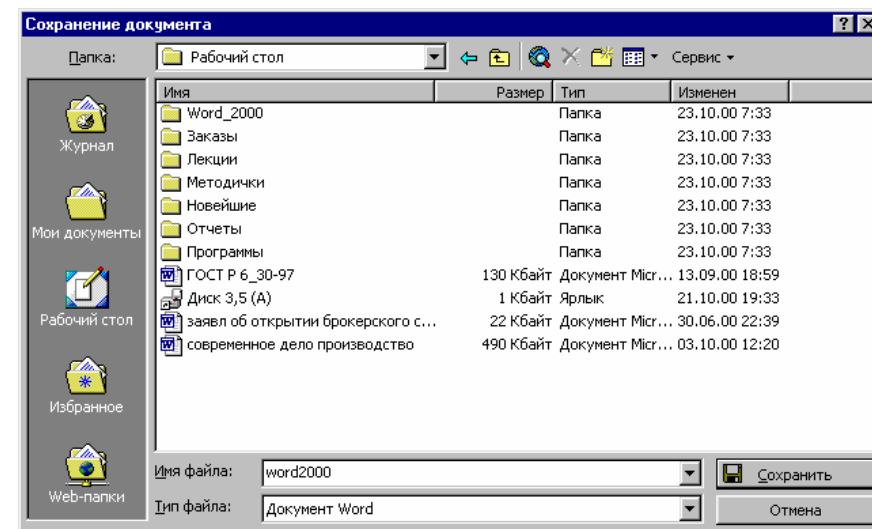



Рис. 17

При первом сохранении появится диалоговое окно **Сохранение документа** (рис. 17). В поле **Папка** следует выбрать диск, в поле, расположенном ниже – папку, в которой необходимо сохранить документ. В поле **Тип файла** – формат, в котором будет сохранен документ. В поле **Имя файла** – ввести имя файла документа и нажать кнопку **Сохранить**.

При повторном сохранении диалоговое окно **Сохранение документа** не выводится, документ автоматически сохраняется в том же файле. Чтобы сохранить документ под другим именем или в другой папке, следует в меню **Файл** выбрать команду **Сохранить как**, после чего появляется окно **Сохранение документа**.

Для закрытия документа необходимо выбрать в меню **Файл** команду **Закрыть** или щелкнуть кнопку  окна документа.

Работа с текстом

Ввод текста

Курсор указывает место, в которое будет вводиться текст. Достигнув края страницы, курсор автоматически переходит в начало следующей строки. Для перехода в начало следующего абзаца следует нажать **Enter**.

Существует два режима ввода текста – вставки и замены. **В режиме вставки** при вводе новых символов, текст, содержащийся в документе, перемещается вправо от места ввода. **В режиме замены** старый текст заменяется новым.

Выделение фрагмента текста

Прежде чем выполнить какую-нибудь операцию над фрагментом текста, его необходимо выделить одним из следующих способов:

установить указатель мыши в левое поле (он превратится в стрелку, направленную вправо), при нажатии клавиши мыши выделится одна строка, при двойном нажатии – абзац, при тройном – весь документ;

установить указатель мыши в левое поле напротив первой строки фрагмента, нажать клавишу мыши и, не отпуская ее, растянуть выделение на весь фрагмент;

установить указатель мыши в начале фрагмента, нажать клавишу мыши и, не отпуская ее, растянуть выделение на весь фрагмент;

для выделения одного слова достаточно дважды щелкнуть на нем мышью;

для выделения одного абзаца можно сделать в нем тройной щелчок;

для выделения одного предложения следует нажать клавишу **Ctrl** и щелкнуть мышью в предложении;

для выделения всего текста следует нажать клавишу **Ctrl** и щелкнуть мышью в левом поле;

чтобы выделить фрагмент текста с помощью клавиатуры, необходимо установить курсор в начало фрагмента и, нажав клавишу **Shift**, клавишами управления курсором растянуть выделение на весь фрагмент.

Снять выделение можно щелчком мыши в любом месте текста. При выделении нового фрагмента предыдущее выделение снимается.

Редактирование текста

Символ справа от курсора удаляется клавишей **Delete**, символ слева от курсора – клавишей **Backspace**. Для удаления фрагмента текста следует выделить его и нажать клавишу **Delete**. Если выделить фрагмент текста и набрать на клавиатуре новый текст, он вставится вместо выделенного фрагмента.

Чтобы разделить абзац на два, необходимо установить курсор в предполагаемый конец первого абзаца и нажать клавишу **Enter**.


Собрать два абзаца в один можно двумя способами:


установить курсор за последним символом первого абзаца и нажать клавишу **Delete**;

установить курсор перед первым символом второго абзаца и нажать клавишу **Backspace**.

При нажатии клавиши **Enter** в текущую позицию курсора вставляется непечатаемый символ ¶. Для включения/отключения режима просмотра непечатаемых символов используется кнопка ¶. Непечатаемые символы удаляются как обычные, клавишами **Delete** и **Backspace**.

Отмена операций над текстом

Для отмены последней операции редактирования необходимо в меню **Правка** выбрать команду **Отменить ...** или щелкнуть кнопку . Если щелкнуть на стрелке ▼ рядом с этой кнопкой, то откроется список операций, выполненных в текущем сеансе. Щелкнув на имени одной операции, можно отменить ее и все операции выполненные после нее.

Чтобы вернуть последнюю отмененную операцию, следует в меню **Правка** выбрать команду **Повторить ...** или щелкнуть кнопку . Для просмотра списка отмененных операций следует щелкнуть на стрелке ▼ рядом с этой кнопкой.

КОПИРОВАНИЕ ТЕКСТА

Для копирования фрагмента текста необходимо:

выделить фрагмент текста;

щелкнуть кнопку  или выбрать в меню **Правка** команду **Копировать**;



установить курсор в место, куда следует вставить фрагмент;

щелкнуть кнопку  или выбрать в меню **Правка** команду **Вставить**.


В процессе этой операции копия выделенного фрагмента текста помещается в буфере промежуточного хранения **Clipboard**, а потом попадает в документ. Вставлять фрагмент из буфера можно сколько угодно раз, но после копирования в буфер нового фрагмента текста, предыдущий фрагмент удаляется.

Перемещение текста

Для перемещения фрагмента текста необходимо:

- ★ • выделить фрагмент текста;
- ★ • щелкнуть кнопку  или выбрать в меню **Правка** команду **Вырезать**;
- ★ • установить курсор в место, куда следует вставить фрагмент;
- ★ • щелкнуть кнопку  или выбрать в меню **Правка** команду **Вставить**.

Переместить фрагмент текста можно следующим способом:

- ★ • выделить фрагмент текста;
- ★ • перетянуть выделение в нужное место (место вставки указывает знак ).

Если при перетягивании выделения держать нажатой клавишу **Ctrl** (при этом возле указателя мыши появится знак +), то фрагмент будет скопирован.

Буфер обмена

В Microsoft Word 2000 существует **буфер обмена** на 12 ячеек, с помощью которого можно копировать фрагменты таблицы не только в

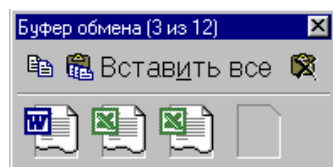






Рис. 18

пределах Word но и в другие приложения, например, в Microsoft Excel. Для выведения панели буфера обмена (рис. 18) необходимо в меню **Вид** выбрать – **Панели инструментов**, потом – **Буфер обмена**. Для копирования фрагмента в буфер его необходимо выделить и нажать кнопку . Для вставки фрагмента из буфера (в позицию курсора) необходимо клацнуть по значку фрагмента. Например, если фрагмент скопирован из Microsoft Word, то он будет иметь значок . Для вставки всех фрагментов из буфера одновременно используется кнопка . Для очистки буфера следует нажать кнопку . При копировании двух фрагментов подряд панель **Буфер обмена** появляется автоматически.

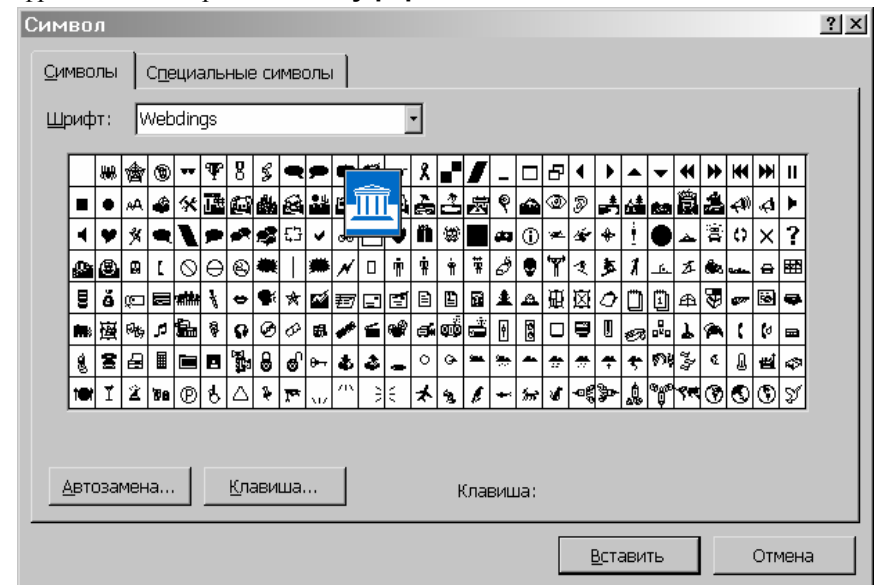


Рис. 19



Вставка символа

Для вставки в текст символа, отсутствующего на клавиатуре, необходимо:

- установить курсор в позицию, в которую следует вставить символ;
- в меню **Вставка** выбрать команду **Символ**;

- в диалоговом окне **Символ** (рис. 19) выбрать вкладыш **Символы**;
- в поле **Шрифт** выбрать тип шрифта;
- щелкнуть мышью нужный символ в таблице;
- щелкнуть кнопку **Вставить**;
- для завершения работы с окном **Символ** – щелкнуть кнопку **Заккрыть**.

Форматирование текста

Это процесс установления параметров фрагмента текста, которые определяют внешний вид текста в этом фрагменте. Перед изменением параметров фрагмент текста следует выделить. Если фрагмент текста не будет выделен, то изменяться будут текущие параметры (параметры текста, который будет вводиться с текущей позиции).

Для изменения параметров символов используется команда **Шрифт** меню **Формат**, которая вызывает диалоговое окно **Шрифт** (рис. 20). Вкладыш **Шрифт** используется для установления параметров шрифта.

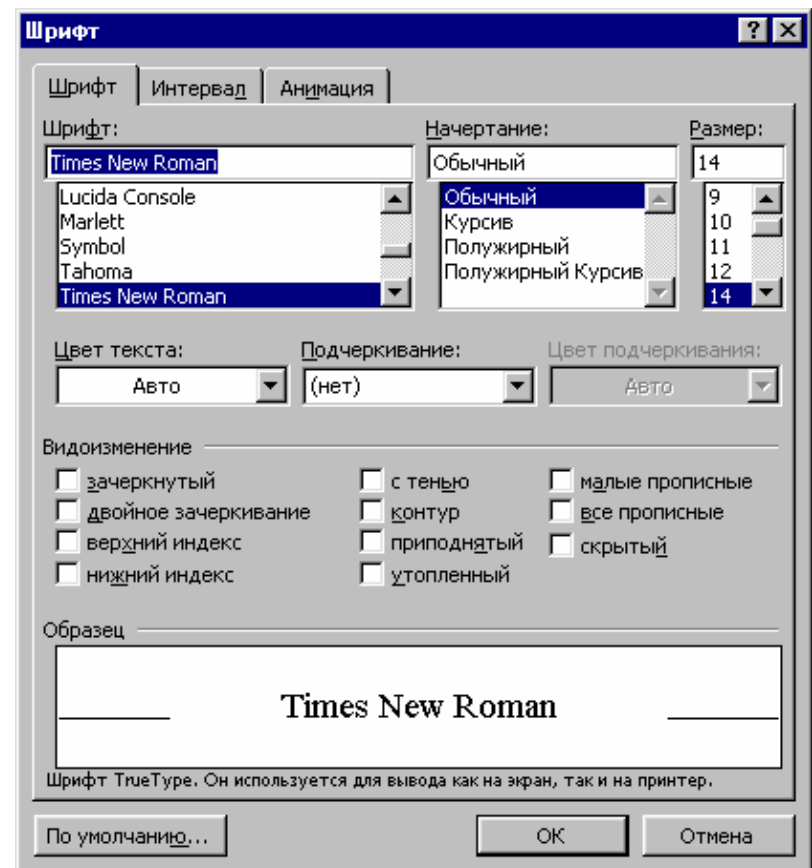


Рис. 20



В поле **Шрифт** выбирается тип шрифта (шрифты типа **TrueType** выглядят одинаково на экране и на печати, рядом с их именем установлены значки **T**).

В поле **Начертание** выбирается начертание шрифта:

Обычный – обычное начертание;

Курсив – *курсивное начертание*;

Полужирный – **жирное начертание**;

Полужирный Курсив – ***жирное курсивное начертание***.

В поле **Размер** – размер шрифта в пунктах (1 пункт = 0,375 мм).

В поле **Подчеркивание** – тип линии подчеркивания.

В поле **Цвет** – цвет символов.

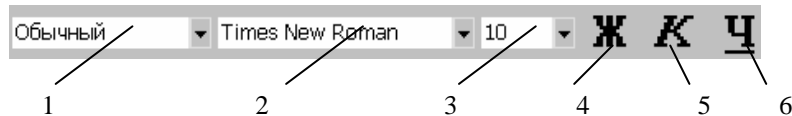


Рис. 21

Установить параметры шрифта можно также с помощью панели **Форматирование** (рис. 21): 1 – стиль форматирования; 2 – тип шрифта; 3 – размер шрифта; 4 – жирное начертание; 5 – курсивное начертание; 6 – подчеркивание одинарной линией.

Для изменения интервала и положения символов используется вкладка **Интервал** диалогового окна **Шрифт**. В поле **Масштаб** выбирается степень растяжения или сжатия символов.

В поле **Интервал** устанавливается межсимвольный интервал:

Обычный – обычный интервал;

Разреженный – расстояние между символами увеличивается до значения, указанного в поле **на**;

Уплотненный – расстояние между символами уменьшается до значения, указанного в поле **на**.

В поле **Смещение** устанавливается вертикальное положение символов:

Нет – обычное положение;

Вверх – символы располагаются выше базовой линии на величину, указанную в поле **на**;

Вниз – символы располагаются ниже базовой линии на величину, указанную в поле **на**.

Для установления параметров абзаца используется команда **Абзац** из меню **Формат**. После выбора этой команды появляется диалоговое окно **Абзац** (рис. 22). Для установления абзацных отступов и интервалов необходимо выбрать вкладку **Отступы и интервалы**.

В поле **Выравнивание** устанавливается способ выравнивания абзаца:

По левому краю – абзац выравнивается по левому полю страницы.

По центру – абзац центрируется между левым и правым полем страницы.

По правому краю – абзац выравнивается по правому полю страницы.

По ширине – абзац выравнивается по обеим полям страницы.

В полях **слева** и **справа** устанавливаются расстояния от левого и правого полей до границ абзаца.

В поле **первая строка** – вид отступа первой строки абзаца:

(нет) – отступ отсутствует.

Отступ – красная строка, расстояние указывается в поле **на**.

Выступ – отрицательный отступ, расстояние указывается в поле **на**.

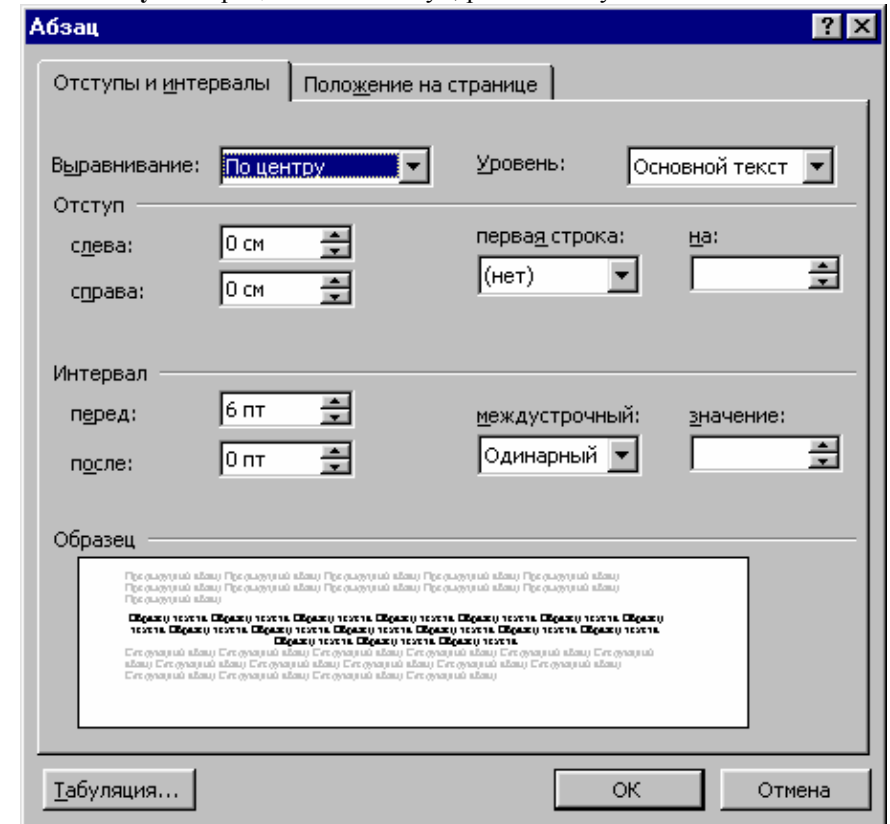


Рис. 22

В полях **перед** и **после** – расстояния соответственно перед первой строкой абзаца и после последней строки абзаца.

В поле **междустрочный** – интервал между строками внутри абзаца:

Одинарный – интервал, стандартный для данного типа шрифта.

Полуторный – интервал в 1,5 раза больше стандартного.

Двойной – интервал в 2 раза больше стандартного.

Минимум – интервал не менее указанного в поле **значение**.

Точно – интервал, равный указанному в поле **значение**.

Множитель – интервал, равный стандартному, умноженному на значение, указанное в поле **значение**.

Устанавливать тип выравнивания можно также с помощью кнопок (рис. 23).



Рис. 23



★ Рис. 24

На горизонтальной координатной линейке (рис. 24) находятся: маркер первой строки (1), маркер левой (2) и правой (3) границ абзаца. Перетягивая их с помощью мыши, можно изменять соответствующие параметры абзаца.

Упорядочение списков

Microsoft Word позволяет быстро составлять списки с пометками, нумерацией и многоуровневые списки с нумерацией. Элементом списка считается абзац текста. Для создания списка необходимо выделить абзацы, которые следует сделать элементами списка или установить курсор в тот абзац, с которого будет начинаться список. Затем вызвать команду **Список** из меню **Формат**, которая вызывает диалоговое окно **Список** (рис. 25).

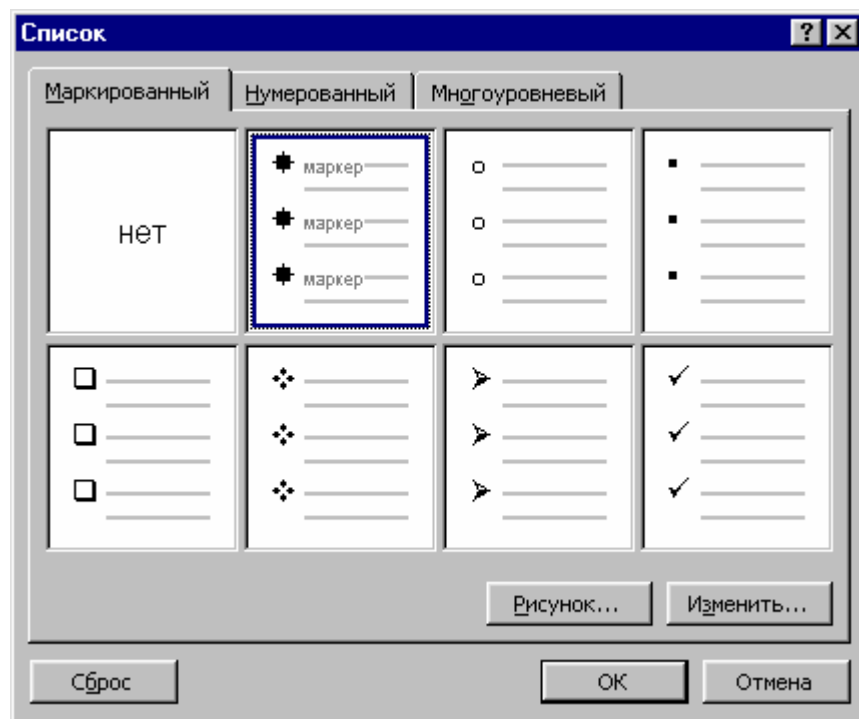




Рис. 25

Для создания списка с пометками необходимо выбрать вкладку **Маркированный**. Каждый элемент списка с пометками выделяется с помощью небольшого значка, расположенного слева от самого элемента. Среди предложенных вариантов пометок следует выбрать подходящий (щелкнуть на нем мышью) и щелкнуть кнопку **ОК**.

Для изменения вида пометки можно воспользоваться кнопкой **Изменить**. Появится окно **Изменение маркированного списка**, в котором содержатся дополнительные пометки. При нажатии кнопки **Маркер** появляется диалоговое окно **Символ**, в котором можно выбрать любой из символов в качестве пометки списка. В рамке **Положение маркера** задается расстояние от левого края абзаца до пометки. В рамке **Положение текста** определяется расстояние от левого края абзаца до левого края текста в списке.

Для создания списков с нумерацией используется вкладка **Нумерованный** диалогового окна **Список**. Среди предложенных вариантов нумерации списка необходимо выбрать подходящий, щелкнуть **ОК** и список будет создан. Когда курсор ввода находится в списке, каждое нажатие **Enter** создает новый пронумерованный элемент списка. При добавлении нового элемента в список или удалении элемента, номера в списке **корректируются автоматически**.

Быстро создать списки с пометками и нумерацией можно с помощью кнопок  и . Для создания списка с несколькими уровнями вложенности используется вкладка **Многоуровневый** диалогового окна **Список**.

Оформление страниц документа

Установление параметров страницы

Для установления параметров страницы используется команда **Параметры страницы** меню **Файл**, которая вызывает диалоговое окно **Параметры страницы**.

Для установления полей страницы используется вкладка **Поля** (рис. 26), в окнах которого можно установить:

Верхнее – верхнее поле страницы.

Нижнее – нижнее поле страницы.

Левое – левое поле страницы.

Правое – правое поле страницы.

В рамке **Образец** показан внешний вид страницы с выбранными параметрами. Если страница должна иметь зеркальные поля, необходимо включить флажок **Зеркальные поля**. В результате вместо полей **Правое** и **Левое** появятся поля **Внутри** и **Снаружи**.

В поле **Переплет** устанавливается ширина поля подшивки.

В рамке **От края до колонтитула** устанавливается расстояние:

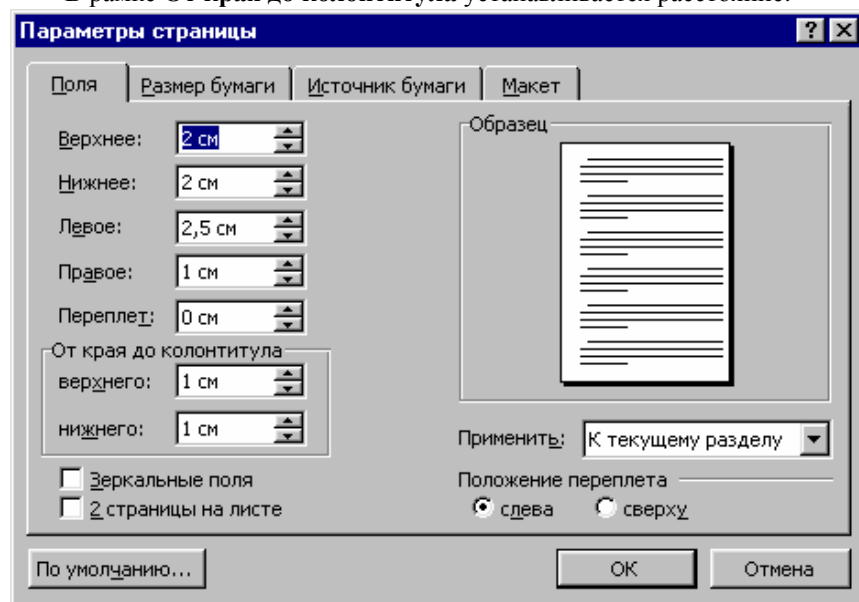


Рис. 26

Верхнего – от верхнего края страницы до верхнего колонтитула.

Нижнего – от нижнего края страницы до нижнего колонтитула.

Следует отметить к какой части документа относятся избранные параметры, выбрав нужное значение в поле **Применить**:

Ко всему документу – параметры используются в всем документе.

К концу документа – параметры используются для той части документа, которая расположена ниже курсора.

Установить поля страницы можно также с помощью координатных линеек в режиме **Разметка страниц**. В этом режиме на экране присутствуют и вертикальная, и горизонтальная координатные линейки. На координатных линейках поля страницы обозначены серым цветом. Необходимо установить указатель мыши на границу серого и белого участка (он будет иметь вид двунаправленной стрелки) и перетянуть ее в нужное место.

Вкладыш **Размер бумаги** (рис. 27) содержит поле **Размер бумаги**, в котором можно выбрать размер страницы документа. Если необходимые размеры в списке отсутствуют, то в поле **Ширина** и **Высота** можно ввести соответствующие значения ширины и высоты страницы.

В рамке **Ориентация** выбирается ориентация страницы. Переключатель **книжная** означает вертикальную ориентацию страницы, **альбомная** – горизонтальную.

Способ вертикального выравнивания текста на странице выбирается в поле **Вертикальное выравнивание**:

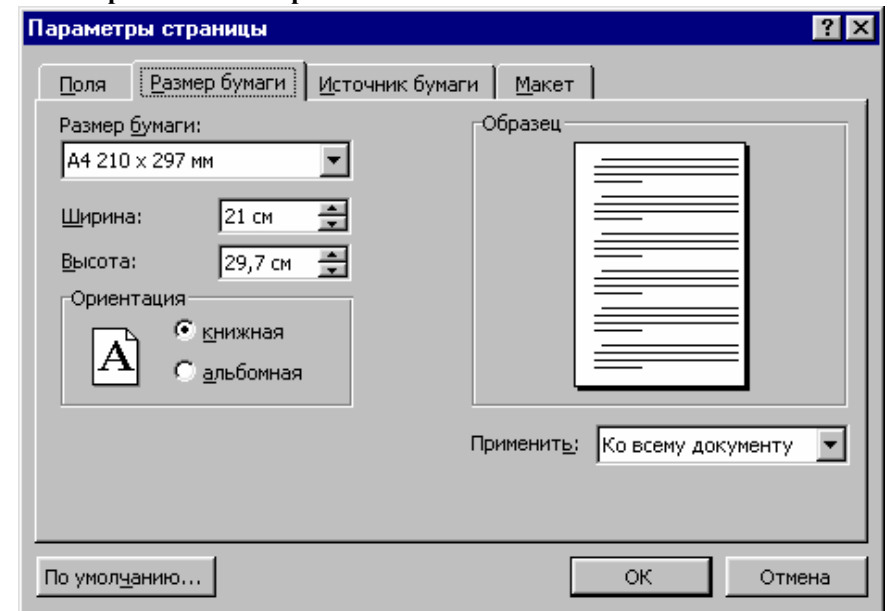


Рис. 27

По верхнему краю – текст выравнивается по верхнему полю страницы.

По центру – текст центрируется между верхним и нижним полем страницы.

По высоте – текст распределяется между верхним и нижним полем (применяется только к полным страницам).

Нумерация страниц

Для вставки номеров страниц необходимо вызывать команду **Номера страниц** меню **Вставка**, которая вызывает окно **Номера страниц** (рис. 28).

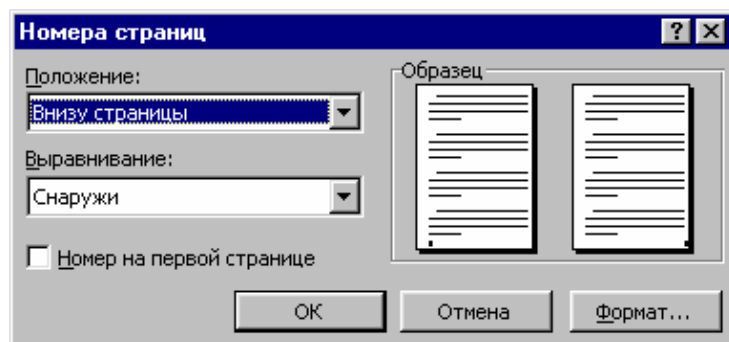


Рис. 28

В поле **Положение** следует выбрать расположение номера на странице:

Вверху страницы – номер страницы располагается вверху (вставляется в верхний колонтитул).

Внизу страницы – номер страницы располагается внизу (вставляется в нижний колонтитул).

В поле **Выравнивание** – расположение номера страницы относительно полей страницы:

Слева – номер страницы располагается у левого края страницы.

От центра – номер страницы располагается по центру страницы.

Справа – номер страницы располагается у правого края страницы.

Внутри – номер страницы располагается с внутреннего края страницы (доступен, если документ имеет зеркальные поля).

Снаружи – номер страницы располагается у внешнего края страницы (доступен, если документ имеет зеркальные поля).

Если снять флажок **Номер на первой странице**, то на первой странице номер не будет проставлен.

Кнопка **Формат** вызывает диалоговое окно **Формат номера страницы** (рис. 29), в котором задается формат нумерации. В поле **Формат номера** выбирается тип нумерации (арабские или римские цифры, буквы латинского алфавита).


В рамке **Нумерация страниц** устанавливается начало нумерации:

продолжить – нумерация страниц текущего раздела начинается с числа, следующего за номером последней страницы предыдущего раздела;
начать с – нумерация начинается с числа, указанного в поле справа.

Если включить флажок **Включить номер главы**, к номеру страницы будет добавлен номер главы или раздела документа. Установив все параметры, следует щелкнуть **ОК**, после чего снова появится окно **Номера страниц**. Здесь также необходимо щелкнуть **ОК** и все страницы документа будут пронумерованы.

Печать документов

Перед распечаткой документа, можно просмотреть на экране, как он будет выглядеть на печати. Для этого

необходимо перейти в режим предварительного просмотра с помощью команды **Предварительный просмотр** меню **Файл** или кнопки . Чтобы увеличить изображение следует навести указатель мыши (он будет иметь вид лупы с плюсом), на нужный фрагмент и нажать кнопку мыши. Указатель мыши примет вид лупы с минусом и если нажать кнопку мыши, то изображение уменьшится. Выйти из режима предварительного просмотра можно с помощью кнопки **Заккрыть** или клавиши **Esc**.

Для распечатки документа используется команда **Печать** меню **Файл**. В диалоговом окне **Печать** (рис. 30) в поле **имя** нужно выбрать принтер, если возможно печатать на нескольких принтерах.

В рамке **Страницы** задается диапазон страниц, которые будут напечатаны:

все – напечатается весь документ;

текущая – напечатается страница, в которой находится курсор;

выделенный фрагмент – напечатается только выделенный фрагмент документа;

номера – напечатается указанный набор страниц. Например: чтобы напечатать страницы 1, 5, 11, 12, 13, необходимо ввести в поле слева: 1, 5, 11 – 13.

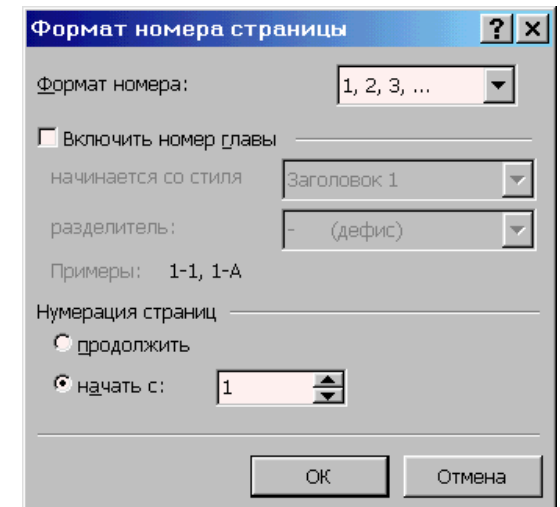


Рис. 29

Для распечатки документа используется команда **Печать** меню **Файл**. В диалоговом окне **Печать** (рис. 30) в поле **имя** нужно выбрать принтер, если возможно печатать на нескольких принтерах.

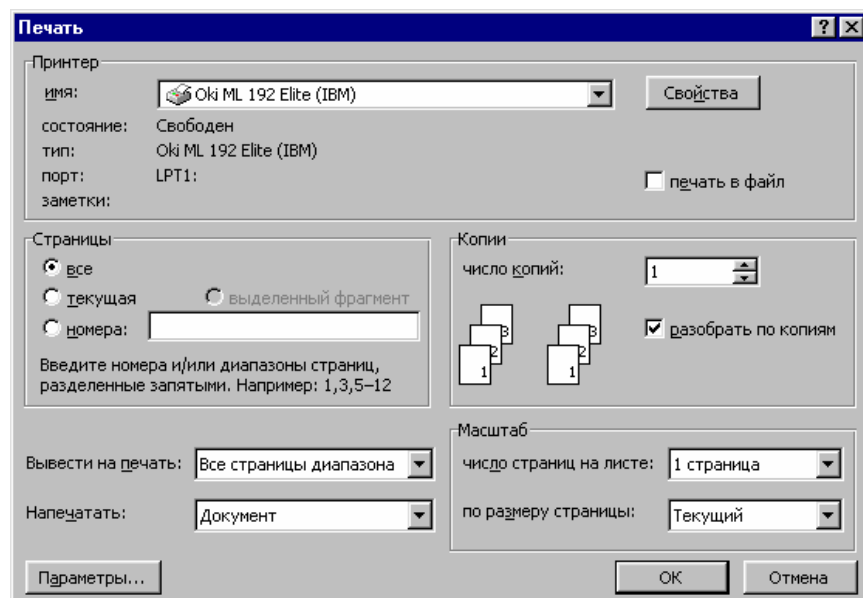


Рис. 30

В рамке **Страницы** задается диапазон страниц, которые будут напечатаны:

все – напечатается весь документ;

текущая – напечатается страница, в которой находится курсор;

выделенный фрагмент – напечатается только выделенный фрагмент документа;


номера – напечатается указанный набор страниц. Например: чтобы напечатать страницы 1, 5, 11, 12, 13, необходимо ввести в поле слева: 1, 5, 11 – 13.

В поле **Копии** указывается количество копий. Чтобы распечатать целиком первую копию, потом вторую и т.д. следует включить флажок **разобрать по копиям**. Для распечатки многостраничного документа с двух сторон каждого листа можно включить режим вывода на печать только четных или нечетных страниц. В списке **Вывести на печать** можно выбрать одно из значений:

Все страницы диапазона – напечатать весь диапазон страниц.


Нечетные страницы – только нечетные страницы из указанного диапазона.

Четные страницы – только четные страницы из указанного диапазона.

Для печати одной копии всего документа достаточно щелкнуть кнопку .

Вставка графических объектов


Microsoft Word позволяет вставлять в документ графические объекты, созданные как в других программах, так и с помощью собственной панели рисования. Объекты можно копировать и вставлять в любое место документа. При добавлении рисунка в документ он присоединяется к окружающему тексту. Если абзац, который содержит рисунок, передвигается вверх или вниз по странице, рисунок передвигается вместе с ним.

Вызвать панель **Рисование** можно через пункт **Панели инструментов** меню **Вставка** или нажав кнопку . При этом следует перейти в режим **Разметка страниц**. С помощью кнопок панели **Рисование** можно изображать линии, стрелки, эллипсы, прямоугольники, круги, дуги, сектора и различные кривые. Графический объект можно залить цветом или узором, изменить форму, зеркально отразить или повернуть, изменить цвет и тип линий, добавить к ним стрелки.

Для вставки графического объекта, созданного в другой программе, необходимо установить курсор в позицию, где должен находиться объект и в меню **Вставка** выбрать пункт **Рисунок**, затем пункт **Из файла**. В появившемся окне (рис. 31), в поле **Папка** выбрать диск, а в поле, расположенном ниже – папку, в которой находится файл с рисунком. Если щелкнуть на имени файла, содержащего рисунок, в рамке слева будет представлено его изображение. После нажатия кнопки **ОК** выбранный рисунок будет вставлен в документ. Для вставки рисунков, поставляемых с Microsoft Word, следует после пункта **Рисунок** выбрать пункт **Картинки**.

Чтобы изменить размеры рисунка, следует щелкнуть на нем мышью, после чего вокруг него появятся маркеры размера. Перетягивая угловые маркеры мышью, можно изменять размеры рисунка при сохранении его пропорций. При перетягивании других маркеров будет изменяться ширина или длина рисунка. Для перемещения рисунка его необходимо перетянуть мышью. При перемещении рисунка за границы видимости, экран продвинется в том же направлении.

Если щелкнуть мышью на рисунке, по умолчанию появится панель Настройка изображения с кнопками для изменения параметров рисунка. Эту панель можно вызывать также с помощью контекстного меню.

Чтобы изменить параметры рисунка, необходимо вызвать окно **Формат рисунка (Формат объекта)**. Для этого необходимо выбрать соответствующий пункт в меню Вид или щелкнуть кнопку  панели **Настройка изображения**. Например, чтобы текст мог располагаться слева или справа от рисунка следует выбрать вкладку **Положение** и там – вид обтекания.

Для удаления рисунка его следует выделить и нажать клавишу **Delete**.

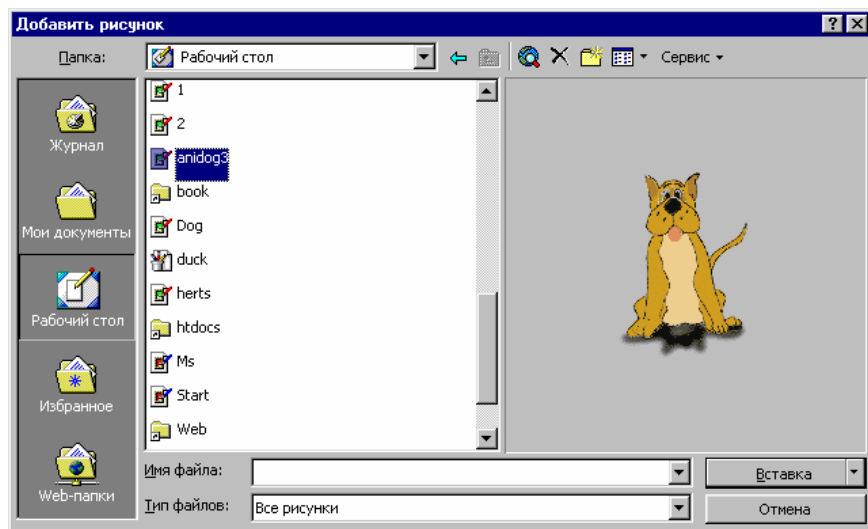




Рис. 31





Вставка таблиц



Для вставки в документ таблицы необходимо установить курсор в месте, где должна начинаться таблица и в меню **Таблица** выбрать пункт **Добавить**, затем **Таблица**. В появившемся диалоговом окне следует ввести число столбцов и строк и щелкнуть **ОК**. Для вставки таблицы также можно щелкнуть кнопку . В открывшемся окне нужно растянуть выделение на необходимое число ячеек и нажать клавишу мыши. Все команды для работы с таблицами находятся в меню **Таблица**.

Передвижение по таблице осуществляется с помощью указателя мыши или клавиш: **↑**, **↓**, **←**, **→**, **Tab** (на ячейку вправо), **Shift + Tab** (на ячейку влево). Каждая ячейка таблицы рассматривается как абзац, и

данные в ячейках форматируются как абзацы текста. Для работы с таблицами удобно пользоваться панелью инструментов Таблицы и границы, которую можно вывести на экран с помощью кнопки .

При наведении указателя мыши на верхнюю линию таблицы, он превращается в черную стрелку. Если в этот момент щелкнуть мышью, то выделится один столбец. Перетягивая мышью черную стрелку, можно выделить сразу несколько столбцов. Строки таблицы выделяются как строки обычного текста. Для выделения нескольких смежных ячеек необходимо щелкнуть мышью в одну ячейку фрагмента и растянуть выделение на остальные.





Когда курсор ввода находится в таблице, на координатных линейках появляются знаки границ столбцов  и строк . При перетягивании этих знаков изменяются размеры соответствующих столбцов и строк. С помощью скрытого списка  можно выбрать тип выравнивания текста в ячейках. Кнопка  служит для изменения направления текста в выделенных ячейках.

Чтобы объединить несколько ячеек в одну, следует выделить их и вызвать команду **Объединить ячейки** меню **Таблица** или щелкнуть кнопку  на панели Таблицы и границы. Для разделения одной ячейки на несколько следует установить в ней курсор и выбрать пункт **Разбить ячейки** меню **Таблица** или щелкнуть кнопку .



Для вставки элементов таблицы (строк, столбцов, ячеек) необходимо выделить элементы, на месте которых необходимо вставить новые и в меню **Таблица** выбрать команду **Добавить**, потом – необходимый пункт (**Столбцы слева**, **Столбцы справа**, **Строки выше**, **Строки ниже**, **Ячейки**). Для удаления элементов таблицы следует выделить их и в меню **Таблица** выбрать пункт **Удалить**, потом – необходимый пункт (**Таблица**, **Столбцы**, **Строки**, **Ячейки**).


По умолчанию линии сетки таблицы имеют толщину 0,5 пт. Изменить толщину и вид линий сетки можно несколькими способами.

Первый способ:

- выделить ячейки, оформление которых нужно изменить;
- в поле **Тип линии**  на панели Таблицы и границы выбрать тип линии;
- в поле **Толщина линии**  – толщину линии;
- если щелкнуть кнопку , появится палитра цветов, в которой можно выбрать цвет оформления;
- открыть скрытый список  и выбрать вид оформления.

Второй способ:

- в полях панели **Таблицы и границы** выбрать тип, толщину и цвет линии;
- щелкнуть кнопку ;
- указателем мыши, который примет вид карандаша, указать начало линии и растянуть ее до конечной точки;
- после нажатия кнопки  указателем мыши можно стирать линии оформления.

Чтобы залить ячейки цветом необходимо выделить их и в скрытом списке  ▼ выбрать цвет.

1.11 Графический редактор

Графический редактор – это программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений. Предоставляет возможности рисования линий, кривых, раскраски областей экрана, создания надписей различными шрифтами и т.д.

Большинство редакторов позволяют обрабатывать изображения, полученные с помощью *сканеров*, а также выводить картинки в таком виде, чтобы они могли быть включены в документ, подготовленный с помощью текстового редактора.

Некоторые редакторы позволяют получать изображения трёхмерных объектов, их сечений, разворотов, каркасных моделей и т.п.

Любой современный графический редактор снабжается обширной библиотекой рисунков (либо сразу в комплекте, либо за отдельную плату). Мало того, многие фирмы занимаются исключительно тем, что разрабатывают и продают картинки самой разнообразной тематики для графических редакторов.

Похоже, нет ни одного предмета, рисунка которого не было бы в графических библиотеках. Здесь и самые разнообразные здания, и множество видов животных и растений (как живых, так и вымерших), и все инструменты, которыми пользуется человек, – начиная от швейной иголки и кончая громадными экскаваторами.

Графические редакторы снабжены спецэффектами – набором различных преобразований (фильтров). Используя спецэффекты, можно перевернуть фрагмент картинки, перекосить его, зеркально отразить, оставить от него только тень, изменить контур, или растянуть (сжать), превратить в камень или ветер, одним словом, изменить иногда до

неузнаваемости, картинку или ее фрагмент. С их помощью можно, нарисовав всего одно лишь деревце, создать за пару минут целый лес.

Ну и, наконец, графический редактор позволяет очень легко переносить фрагменты из одних картинок в другие, komponуя новые изображения из старых.

Умело используя возможности буфера обмена, и не забывая о спецэффектах и библиотеках стандартных рисунков, можно создавать громадное количество оригинальных картинок, совершенно не умея рисовать.

Среди графических редакторов пользуется известностью **Corel DRAW!** – мощный графический редактор с функциями создания публикаций, снабжённый инструментами для редактирования графики и трёхмерного моделирования.

1.11.1 Стандартные инструменты графического редактора

Художник делает всю работу по созданию графики с помощью карандаша, линейки, циркуля, ластика, кисточек, красок и других привычных инструментов.

Стандартные инструменты позволяют выполнять аналогичные операции в графическом редакторе.

Всего лишь одним нажатием на клавишу мыши можно выбрать любой цвет, представленный в **палитре** редактора, или нарисовать любую фигуру с помощью карандаша или кисти (рис. 32).

Роль карандаша играет **графический курсор**. Он может выглядеть по-разному – в виде стрелки, крестика или точки. Курсор можно перемещать по экрану с помощью мышки или графического планшета. При этом в специально отведенном месте экрана можно увидеть два изменяющихся числа. Это **координаты курсора**. Они показывают, на каком пикселе находится курсор (пиксель – одна светящаяся точка). Перемещая курсор, легко выяснить, где находится начало системы координат и куда направлены оси.

Карандаш (кисточка, мел и т.д.) оставляет за собой цветную линию. При этом следует сначала выбрать инструмент, а затем нажать левую кнопку мыши и, не отпуская ее, рисовать на холсте.

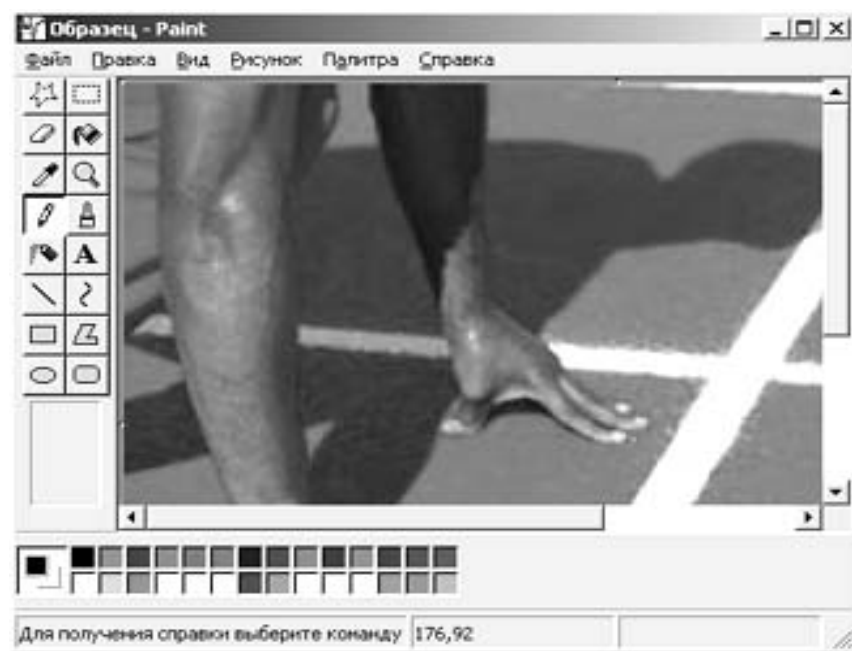


Рис. 32

Роль карандаша играет *графический курсор*. Он может выглядеть по-разному – в виде стрелки, крестика или точки. Курсор можно перемещать по экрану с помощью мышки или графического планшета. При этом в специально отведенном месте экрана можно увидеть два изменяющихся числа. Это *координаты курсора*. Они показывают, на каком пикселе находится курсор (пиксель – одна светящаяся точка). Перемещая курсор, легко выяснить, где находится начало системы координат и куда направлены оси.

Карандаш (кисточка, мел и т.д.) оставляет за собой цветную линию. При этом следует сначала выбрать инструмент, а затем нажать левую кнопку мыши и, не отпуская ее, рисовать на холсте.

Цвет выбирается в *палитре*, расположенной снизу или сбоку экрана. Выбор *толщины линии* необходим не только для карандаша, но и при рисовании стандартных фигур (графических примитивов).

Окружность – первая стандартная фигура. После того как вы поставили курсор на эту кнопочку и нажали левую клавишу мышки, поставьте курсор в центр будущей окружности, и, не отпуская левую кнопку, сдвиньте мышь. Этим действием выбирается радиус. Как вы скоро убедитесь, этим инструментом легко рисуются не только окружности, но и *эллипсы*.

Для рисования **прямоугольника** поставьте курсор в левый верхний угол будущего прямоугольника, не отпуская левую клавишу мыши, сдвиньте мышь. Этим действием выбирается размер прямоугольника. После того как размер выбран, отпустите кнопку мышки.

Резинка-ластик – выбрав этот инструмент, вы можете стирать ненужные фрагменты рисунка.

Резиновая линия – позволяет провести прямую линию. Отметьте мышкой начальную точку, растяните прямую линию до конечной точки и снова нажмите левую клавишу мышки.

Распылитель – разбрызгивает краску (как будто из аэрозольного баллончика).

Заливка – заливает выбранным цветом замкнутый контур. Если в контуре фигуры есть дырочка хотя бы в один пиксель, то краска разольется по всему рисунку.

С помощью инструмента **Текстура** – можно залить не только сплошным цветом, но и некоторым рисунком (фактурой).

Например, сделать заливку цветными попугаями или зелеными листочками. Стандартных видов текстур довольно много. Многие программы позволяют создавать заливки с переходом цветов, чаще называемые **градиентными**.

Инструмент Zoom (увеличительное стекло или лупа) – после нажатия на эту кнопку появится небольшой прямоугольник. Наложите его на часть рисунка, которую хотелось бы увеличить, и нажмите левую клавишу мыши. Это особенно важно проделать перед заливкой. С помощью лупы легко отыскать малейшие дырочки в контуре.

Любую ошибку можно исправить кнопкой **Отмена** или **стрелка влево** (возврат на один шаг назад). Количество шагов разное в различных программах. Например, можно тут же отменить заливку, если вам не понравился узор, которым вы покрасили нарисованный воздушный шарик.

Файл (пункт меню) – в этом разделе содержатся команды загрузки уже готовых рисунков и сохранение графики, над которой вы работаете. Наиболее распространены следующие форматы графических файлов: JPG, BMP, WMF, PDF, GIF, PNG.

1.12 Табличный процессор

Табличный процессор – это комплекс взаимосвязанных программ, предназначенный для обработки электронных таблиц.

Электронная таблица – это компьютерный эквивалент обычной таблицы, состоящей из строк и граф, на пересечении которых располагаются клетки, в которых содержится числовая информация, формулы или текст (рис. 33).

	A	B	C	D	E	F
1	Vertrod #V1					
2	Foil Pouch 3630				Foil Pouch 3630	
3	Dwell #2	Dwell #3	Dwell #4	Dwell #5	Dwell #2	Dwell #3
4	1/30/95	1/30/95	1/30/95	1/30/95	1/31/95	1/31/95
5	17.85	22.9	23.65	24.8	19.15	23.1
6	19	26.4	22.6	27.5	18.6	21.1
7	13.15	25.45	26.9	25.9	19.01	24.4
8	19.05	24.55	27.65	28.2	19.25	24.4
9	14.1	25.4	29.1	21.45	17.65	22.1
10	27.8	21.45	27.4	23.55	22.02	21.1
11	36.5	21.2	23.3	20.65	21.9	26.1
12	24.25	26.3	24.8	20.5	19.75	23.1
13	15.05	23.55	23.6	23.9	14.65	22.1
14	24.9	25	24.45	25.1	17.9	22.1
15	26.55	27.35	20.25	24.1	15.85	21.1
16	30.7	22.45	24.55	22.95	15.7	21.1
17	31.3	26.35	25.6	27.3	13.25	24.1

★ Рис. 33

Графам и строкам можно присваивать наименования. Экран монитора считается окном, через которое можно рассматривать таблицу целиком или по частям.

Табличные процессоры представляют собой удобное средство для проведения бухгалтерских и статистических расчетов. В каждом пакете имеются сотни встроенных математических функций для статистической обработки данных. Кроме того, имеются средства для связи таблиц между собой, создания и редактирования электронных баз данных.

Специальные средства позволяют автоматически получать и распечатывать настраиваемые отчеты с использованием десятков различных типов таблиц, графиков, диаграмм, снабжать их комментариями и графическими иллюстрациями.

Табличные процессоры имеют встроенную справочную систему, предоставляющую пользователю информацию по конкретным командам меню и другие справочные данные.

Самые популярные табличные процессоры – *Microsoft Excel (Эксель)* и *Lotus 1 – 2 – 3*.

Значение в числовой клетке таблицы может быть либо записано, либо рассчитано по соответствующей формуле; в формуле могут присутствовать обращения к другим клеткам.

Каждый раз при изменении значения в клетке таблицы в результате записи в нее нового значения с клавиатуры *пересчитываются* также значения во всех тех клетках, в которых стоят величины, зависящие от данной клетки.

В *Microsoft Excel* автоматизированы многие операции, специальные шаблоны помогают создавать отчёты, импортировать данные и многое другое.

Lotus 1 – 2 – 3 – профессиональный процессор электронных таблиц. Широкие графические возможности и удобный интерфейс пакета позволяют быстро ориентироваться в нём. С его помощью можно создать любой финансовый документ, отчёт для бухгалтерии, составить бюджет, а затем разместить все эти документы в базах данных.

1.13 Системы управления базами данных (СУБД)

База данных – это один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

В базе данных предприятия, например, может храниться:

- вся информация о рабочих и служащих предприятия;
- сведения о материальных ценностях;
- данные о поступлении сырья и комплектующих;
- сведения о запасах на складах;
- данные о выпуске готовой продукции;
- приказы и распоряжения дирекции и т.п.

Даже небольшие изменения какой-либо информации могут приводить к значительным изменениям в разных других местах.

Пример. Издание приказа о повышении в должности одного работника приводит к изменениям не только в досье работника, но и к изменениям в списках отдела, в котором он работает, в ведомостях на зарплату, в графике отпусков и т.п.

Базы данных используются под управлением *систем управления базами данных (СУБД)* (рис. 34).

Система управления базами данных (СУБД) – это система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ пользователей.

Системы управления базами данных позволяют *объединять* большие объемы информации и *обрабатывать* их, *сортировать*, *делать выборки* по определённым критериям и т.п.

Современные СУБД дают возможность включать в них не только *текстовую* и *графическую* информацию, но и звуковые фрагменты и даже видеоклипы.

Простота использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями.

СУБД обеспечивают *правильность*, *полноту* и *непротиворечивость* данных, а также *удобный доступ* к ним.

Популярные СУБД – ***FoxPro, Access for Windows, Paradox.***

Для менее сложных применений вместо СУБД используются ***информационно-поисковые системы (ИПС)***, которые выполняют следующие функции:

- *хранение* большого объема информации;
- быстрый *поиск* требуемой информации;
- *добавление, удаление* и *изменение* хранимой информации;
- *вывод* ее в удобном для человека виде.

Как же организована база данных? В ней содержатся сведения о большом количестве однотипных объектов. При этом для каждого из объектов существенными являются значения лишь некоторых признаков.

Что понимается под словами "признак" и "значение признака"? Рассмотрим базу данных библиотеки (табл. 2):

Таблица 2

Инв. №	Фамилия, имя, отчество автора	Произведение	Год издания	Количество страниц
1054	Чехов А. П.	Дом с мезонином	1983	320
1298	Куприн А. И.	Гранатовый браслет	1980	320
3762	Тургенев И. С.	Вешние воды	1986	560

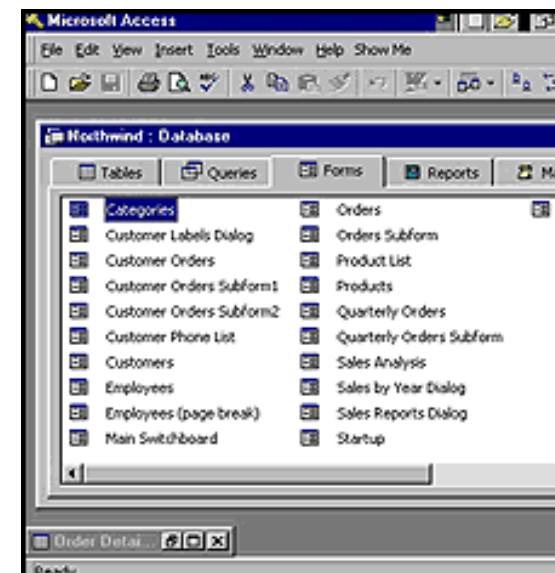


Рис. 34

Здесь объектами являются: художественная и техническая литература, подписки газет, журналов и т.д., а их признаками будут: жанр, название книги, фамилия автора, год издания, количество страниц, есть ли книга на месте или выдана кому-то из читателей.

Особенность базы данных в том, что у всех объектов, входящих в эту базу, количество признаков (полей) одно и то же. Другое дело, что у некоторых из этих признаков может не быть значения (например, при составлении базы данных телефонов учащихся класса, у некоторых в поле телефон будет прочерк, т.е. отсутствовать значение).

Иногда некоторые из признаков объявляют **ключевыми**. Это полезно, потому что по ним, в частности, в дальнейшем можно сделать сортировку.

Некоторые из признаков могут быть объявлены обязательно присутствующими. Например, у каждой книги, находящейся в библиотеке имеется инвентарный номер и код.

Каждая конкретная база предназначена для решения своего круга задач. В свою очередь, для каждого класса задач характерен свой набор объектов и их признаков. Поэтому и соответствующие базы будут различными.

Для того, чтобы заставить компьютер найти интересующие нас сведения, нужно составить **запрос**. Правила записи запросов для каждой ИПС свои. Эти правила устанавливаются теми, кто создает ИПС.

Обычно самые распространенные запросы к базе данных уже заранее составлены. Чтобы получить информацию по такому запросу, надо просто выбрать соответствующий пункт в меню, расположенном на экране.

Сейчас в мире созданы сотни тысяч баз данных. Они используются в библиотеках и больницах, на заводах, в магазинах и планирующих организациях, в банках и частных фирмах.

Тематические базы данных для узких специалистов называют **банками данных**. Это может быть банк данных по микропроцессорам, банк по лекарственным средствам, банк публикаций в области ядерной физики и т.п.

Из всего множества разнообразных баз данных можно выделить следующие направления (хотя деление на них условное):

- Пополняемые централизованно, за плату, например, базы данных правовой информации Консультант+, Кодекс и другие.
- Непополняемые, например энциклопедия "Кирилла и Мефодия". Обычно такие базы данных поставляются на CD-ROM, а при изменении данных выходит новый выпуск.

- Написанные под заказ, позволяющие вводить определенные данные и решать конкретную задачу, например ИПС "телефонный справочник".

- Программы, (СУБД), позволяющие создавать любые базы с любым количеством и типом полей, и выполнять над данными любые операции.

Наиболее часто используемые типы полей:

- текстовый
- дата
- формула
- число
- счетчик (специальное поле, подсчитывающее количество записей)

Каждое поле имеет определенную длину, измеряемую в символах и название.

Многие СУБД и ИПС используют *справочник* для некоторых полей, например справочник наименований товаров, справочник единиц измерения и т.д.

Для ввода данных используются специальные карточки, называемые *формами* (окно на экране компьютера с именами полей и местом для ввода информации).

Например, в базе данных библиотеки сортировка выполнена по полю "инвентарный номер" в убывающем порядке:



Инв. №	Фамилия, имя, отчество автора	Произведение	Год издания	Количество страниц
3762	Тургенев И.С.	Вешние воды	1986	560
1298	Куприн А. И.	Гранатовый браслет	1980	320
1054	Чехов А. П.	Дом с мезонином	1983	320

В этом примере сортировка выполнена по полю "Автор" в возрастающем порядке:

Инв. №	Фамилия, имя, отчество автора	Произведение	Год издания	Количество страниц
1298	Куприн А. И.	Гранатовый браслет	1980	320
3762	Тургенев И. С.	Вешние воды	1986	560
1054	Чехов А. П.	Дом с мезонином	1983	320

Все данные из полей хранятся в одном или нескольких файлах в виде записей. Любую базу данных можно представить как таблицу (или несколько таблиц), в заголовке столбцов которой будут имена полей, а запись будет представлена в виде строки таблицы, содержащей данные.

Операции, выполняемые над данными:

- ввод и корректировка информации
- сортировка по ключевым полям (по возрастанию, по убыванию)
- поиск информации (запрос)
- распечатка отчетов на принтере или в файл
- выполнение расчетов (программы бухучета)

1.14 Интегрированные пакеты программ

Интегрированные пакеты представляют собой набор нескольких программных продуктов, объединенных в один инструмент. Обычно они включают в себя текстовый редактор, органайзер, электронную таблицу, СУБД, средства поддержки электронной почты, программу создания презентационной графики.

Результаты, полученные отдельными программами, могут быть объединены в окончательный документ, содержащий табличный, графический и текстовый материал.

Пример: **интегрированный пакет для написания книг, содержащих иллюстрации.** Он содержит:

- текстовый редактор;
- орфографический корректор на 80000 слов (программу обнаружения орфографических ошибок);
- программу слияния текстов;
- программу формирования оглавлений и составления указателей;
- автоматический поиск и замену слов и фраз;
- средства телекоммуникации;
- электронную таблицу;
- систему управления базами данных;
- модули графического оформления;
- графический редактор;
- возможность печати сотнями разных шрифтов и т.д.

Наиболее известные интегрированные пакеты:

Microsoft Office. В этот мощный профессиональный пакет вошли такие необходимые программы, как текстовый редактор **WinWord**, электронная таблица **Excel**, программа создания презентаций **PowerPoint**, СУБД **Access**, средство поддержки электронной почты **Mail**. Мало того, все части этого пакета составляют единое целое, и даже внешне все

программы выглядят единообразно, что облегчает как их освоение, так и ежедневное использование.

Microsoft Works – это очень простой и удобный пакет, объединяющий в себе текстовый редактор, электронные таблицы и базы данных, а также телекоммуникационные средства для соединения с другими компьютерами по телефонным линиям. Пакет ориентирован на людей, не имеющих времени осваивать сложные продукты, на начинающих пользователей, а также на домашних пользователей.

Ч А С Т Ь 2

Алгоритмы. Алгоритмизация. Алгоритмические языки

2.1 Понятие алгоритма

Название "**алгоритм**" произошло от латинской формы имени величайшего среднеазиатского математика **Мухаммеда ибн Муса ал-Хорезми** (Alhorithmi), жившего в 783 – 850 гг. В своей книге "Об индийском счете" он изложил правила записи натуральных чисел с помощью арабских цифр и правила действий над ними.

Человек ежедневно встречается с необходимостью следовать тем или иным правилам, выполнять различные инструкции и указания. Например, переходя через дорогу на перекрестке без светофора надо сначала посмотреть направо. Если машин нет, то перейти полдороги, а если машины есть, ждать, пока они пройдут, затем перейти полдороги. После этого посмотреть налево и, если машин нет, то перейти дорогу до конца, а если машины есть, ждать, пока они пройдут, а затем перейти дорогу до конца.

В математике для решения типовых задач мы используем определенные правила, описывающие последовательности действий. Например, правила сложения дробных чисел, решения квадратных уравнений и т. д. Обычно любые инструкции и правила представляют собой последовательность действий, которые необходимо выполнить в определенном порядке. Для решения задачи надо знать, что дано, что следует получить и какие действия и в каком порядке следует для этого

выполнить. Порядок выполнения действий над данными с целью получения необходимых результатов, и есть алгоритм.

Алгоритм – заранее заданное понятное и точное указание возможному исполнителю совершить последовательность действий для получения решения задачи за конечное число шагов.

Понятие алгоритма является не только одним из главных понятий математики, но одним из главных понятий современной науки.

Исполнитель алгоритма – это техническая, биологическая или биотехническая система, способная выполнить действия, предписываемые алгоритмом.

Исполнителя характеризуют:

- среда;
- элементарные действия;
- система команд;
- отказы.

Среда (или обстановка) – это "место обитания" исполнителя.

Каждый исполнитель может выполнять команды только из некоторого строго заданного списка – **системы команд** исполнителя. Для каждой команды должны быть заданы **условия применимости** (когда может быть выполнена команда) и описаны **результаты выполнения команды**.

После вызова команды исполнитель совершает соответствующее **элементарное действие**.

Отказы исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды.

Обычно исполнитель ничего не знает о цели алгоритма. Он выполняет все полученные команды, не задавая вопросов "почему" и "зачем".

В информатике универсальным исполнителем алгоритмов является **компьютер**.

2.2 Основные свойства алгоритма

Основные свойства алгоритмов следующие:

1 **Понятность** для исполнителя – исполнитель алгоритма должен понимать, как его выполнять.

2 **Дискретность** (прерывность, раздельность) – алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов (этапов).

3 **Определенность** – каждое правило алгоритма должно быть четким и однозначным. Благодаря этому свойству выполнение алгоритма

носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

4 **Результативность** (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

5 **Массовость** означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он применяется для целого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

2.3 Представление алгоритмов

На практике наиболее распространены следующие формы представления алгоритмов:

- словесная – запись на естественном языке;
- графическая – изображения из графических символов;
- псевдокоды – описания алгоритмов на условном алгоритмическом языке, включающие в себя элементы языка программирования и фразы естественного языка, общепринятые математические обозначения и др.;
- программная – тексты на языках программирования.

Словесный способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.

Например: записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел (алгоритм Эвклида).

Алгоритм может быть следующим:

- задать два числа;
- если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
- определить большее из чисел;
- заменить большее из чисел разностью большего и меньшего из чисел;
- повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным числам и должен приводить к решению поставленной задачи. (Убедитесь в этом самостоятельно, определив с помощью этого алгоритма наибольший общий делитель чисел 125 и 75).

Словесный способ не имеет широкого распространения, так как такие описания:

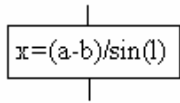
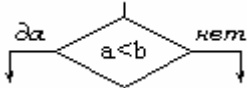
- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

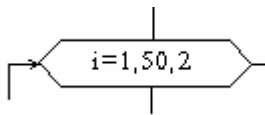


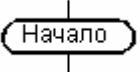

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

Такое графическое представление называется схемой алгоритма или **блок-схемой**. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий. В табл. 3 приведены наиболее часто употребляемые символы.

Таблица 3

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий

Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Блок **"процесс"** применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок.

Блок **"решение"** используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок **"модификация"** используется для организации циклических конструкций. Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок **"предопределенный процесс"** используется для указания обращений к вспомогательным алгоритмам, существующим автономно в

виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов.

Псевдокод занимает промежуточное место между естественным и формальным языками. С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

В псевдокоде не приняты строгие синтаксические правила для записи команд, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд, рассчитанный на абстрактного исполнителя.

Однако в псевдокоде обычно имеются некоторые конструкции, присущие формальным языкам, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке. В псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда.

Возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.

Примером псевдокода является школьный алгоритмический язык в русской нотации. Основные служебные слова этого псевдокода приведены в табл. 4.

Таблица 4

<i>алг</i> (алгоритм)	<i>сим</i> (символьный)	<i>дано</i>	<i>для</i>	<i>да</i>
<i>арг</i> (аргумент)	<i>лит</i> (литерный)	<i>надо</i>	<i>от</i>	<i>нет</i>
<i>рез</i> (результат)	<i>лог</i> (логический)	<i>если</i>	<i>до</i>	<i>при</i>
<i>нач</i> (начало)	<i>таб</i> (таблица)	<i>то</i>	<i>знач</i>	<i>выбор</i>
<i>кон</i> (конец)	<i>нц</i> (начало цикла)	<i>иначе</i>	<i>и</i>	<i>ввод</i>
<i>цел</i> (целый)	<i>кц</i> (конец цикла)	<i>все</i>	<i>или</i>	<i>вывод</i>

<i>вещ</i> (вещественный)	<i>длин</i> (длина)	<i>пока</i>	<i>не</i>	<i>утв</i>
---------------------------	---------------------	-------------	-----------	------------

Общий вид алгоритма:

алг название алгоритма (аргументы и результаты)
дано условия применимости алгоритма
надо цель выполнения алгоритма
нач описание промежуточных величин
| последовательность команд (тело алгоритма)
кон

Часть алгоритма от слова **алг** до слова **нач** называется **заголовком**, а часть, заключенная между словами **нач** и **кон** – **телом** алгоритма.

В предложении **алг** после названия алгоритма в круглых скобках указываются **характеристики** (**арг**, **рез**) и **тип значения** (**цел**, **вещ**, **сим**, **лит** или **лог**) всех **входных** (аргументы) и **выходных** (результаты) переменных. При описании массивов (таблиц) используется служебное слово **таб**, дополненное **границными парами** по каждому индексу элементов массива.

Примеры предложений алг:

алг Объем и площадь цилиндра (**арг вещ** R, H, **рез вещ** V, S)
алг Корни КвУр (**арг вещ** a, b, c, **рез вещ** x1, x2, **рез лит** t)
алг Исключить элемент (**арг цел** N, **арг рез вещ таб** A[1:N])
алг Диагональ (**арг цел** N, **арг цел таб** A[1:N, 1:N], **рез лит** Otvet)

Предложения **дано** и **надо** не обязательны. В них рекомендуется записывать утверждения, описывающие состояние среды исполнителя алгоритма, например:

- 1 **алг** Замена (**арг лит** Str1, Str2, **арг рез лит** Text)
дано | длины подстрок Str1 и Str2 совпадают
надо | всюду в строке Text подстрока Str1 заменена на Str2
- 2 **алг** Число максимумов (**арг цел** N, **арг вещ таб** A[1:N], **рез цел** K)
дано | N0
надо | K – число максимальных элементов в таблице A
- 3 **алг** Сопротивление (**арг вещ** R1, R2, **арг цел** N, **рез вещ** R)
дано | N5, R10, R20
надо | R – сопротивление схемы

Здесь в предложениях **дано** и **надо** после знака "|" записаны **комментарии**. Комментарии можно помещать в конце любой строки. Они не обрабатываются транслятором, но облегчают понимание алгоритма.

Существуют следующие команды псевдокода:

1 *Команда присваивания.* Служит для вычисления выражений и присваивания их значений переменным.

Общий вид: **A := B**, где знак " := " означает команду **заменить прежнее значение переменной, стоящей в левой части, на вычисленное значение выражения, стоящего в правой части.**

Например:

$a := (b+c) * \sin(\pi/4); i := i+1.$

2 *Команды ввода и вывода.*

- **ввод** имена переменных

- **вывод** имена переменных, выражения, тексты.

3 *Команды если и выбор.* Применяют для организации ветвлений.

4 *Команды для и пока.* Применяют для организации циклов.

Пример записи алгоритма с помощью псевдокода:

алг Сумма квадратов (**арг** цел n, **рез** цел S)

дано | n 0

надо | $S = 1*1 + 2*2 + 3*3 + \dots + n*n$

нач цел i

ввод n; S:=0

нц для i от 1 до n

S:=S+i*i

кц

вывод "S = ", S

кон

2.4 Базовые структуры алгоритмов

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных **базовых** (т.е. основных) **элементов**.

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: следование, ветвление, цикл.

Характерной особенностью базовых структур является наличие в них **одного входа и одного выхода**.

1 **Базовая структура "следование".** Образуется последовательностью действий, следующих одно за другим:

Псевдокод	Язык блок-схем
-----------	----------------

действие 1 действие 2 действие n	<pre> graph TD Start(()) --> A[действие 1] A --> B[действие 1] B --> C[...] C --> D[действие 1] D --> End(()) </pre>
---	---

2 Базовая структура "ветвление". Обеспечивает в зависимости от результата проверки условия (**да** или **нет**) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Структура **ветвление** существует в четырех основных вариантах:

- если – то;
- если – то – иначе;
- выбор;
- выбор – иначе.

Псевдокод	Язык блок-схем
1 если – то	
если условие то действия все	<pre> graph TD Start(()) --> Cond{условие} Cond -- да --> Act[действия] Act --> Join(()) Cond -- нет --> Join Join --> End(()) </pre>
2 если – то – иначе	
если условие то действия 1 иначе действия 2 все	<pre> graph TD Start(()) --> Cond{условие} Cond -- да --> Act1[действия 1] Act1 --> Join(()) Cond -- нет --> Act2[действия 2] Act2 --> Join Join --> End(()) </pre>
Псевдокод	Язык блок-схем

3 выбор	
выбор при условие 1: действия 1 при условие 2: действия 2 при условие N: действия N все	<pre> graph TD Start(()) --> D1{условие 1} D1 -- да --> A1[действия 1] D1 -- нет --> D2{условие 2} D2 -- да --> A2[действия 2] D2 -- нет --> Dn{условие N} Dn -- да --> An[действия N] Dn -- нет --> Exit(()) A1 --> Exit A2 --> Exit An --> Exit Exit --> End(()) </pre>
4 выбор – иначе	
выбор при условие 1: действия 1 при условие 2: действия 2 при условие N: действия N иначе действия N+1 все	<pre> graph TD Start(()) --> D1{условие 1} D1 -- да --> A1[действия 1] D1 -- нет --> D2{условие 2} D2 -- да --> A2[действия 2] D2 -- нет --> Dn{условие N} Dn -- да --> An[действия N] Dn -- нет --> A_n1[действия N+1] A1 --> Exit(()) A2 --> Exit An --> Exit A_n1 --> Exit Exit --> End(()) </pre>

Примеры структуры ветвление

Школьный алгоритмический язык	Язык блок-схем
если $x > 0$ то $y := \sin(x)$ все	<pre> graph TD Start(()) --> D{x > 0} D -- да --> A[y := sin(x)] D -- нет --> Exit(()) A --> Exit Exit --> End(()) </pre>
Школьный алгоритмический язык	Язык блок-схем

если $a > b$ то $a := 2*a; b := 1$ иначе $b := 2*b$ все	
выбор при $n = 1$: $y := \sin(x)$ при $n = 2$: $y := \cos(x)$ при $n = 3$: $y := 0$ все	
выбор при $a \neq 5$: $i := i+1$ при $a = 0$: $j := j+1$ иначе $i := 10; j := 0$ все	

3 Базовая структура "цикл". Обеспечивает **многократное выполнение некоторой совокупности действий**, которая называется **телом цикла**. Основные разновидности циклов представлены в таблице:

Псевдокод	Язык блок-схем
Цикл типа пока. Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока.	
нц пока условие тело цикла (последовательность действий) кц	
Псевдокод	Язык блок-схем

Цикл типа для.	
Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.	
нц для i от $i1$ до $i2$ тело цикла (последовательность действий) кц	

Примеры структуры цикл	
Псевдокод	Язык блок-схем
нц пока $i \leq 5$ $S := S + A[i]$ $i := i + 1$ кц	
нц для i от 1 до 5 $X[i] := i * i * i$ $Y[i] := X[i] / 2$ кц	

2.5 Итерационные циклы

Особенностью итерационного цикла является то, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл типа *пока*. Выход из итерационного цикла осуществляется в случае выполнения заданного условия.

На каждом шаге вычислений происходит **последовательное приближение к искомому результату и проверка условия достижения последнего.**

Пример. Составить алгоритм вычисления бесконечной суммы

$$S = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots (-1)^{i-1} \frac{x^i}{i} + \dots$$

с заданной точностью ε (для данной знакопеременной бесконечной суммы требуемая точность будет достигнута, когда очередное слагаемое станет по абсолютной величине меньше ε).

Вычисление сумм – типичная циклическая задача. Особенностью же нашей конкретной задачи является то, что число слагаемых (а, следовательно, и число повторений тела цикла) заранее неизвестно. Поэтому выполнение цикла должно завершиться в момент достижения требуемой точности.

При составлении алгоритма нужно учесть, что знаки слагаемых чередуются и степень числа x в числителях слагаемых возрастает.

Решая эту задачу путем вычисления на каждом i -ом шаге частичной суммы

$$S := S + ((-1)^{(i-1)} * (x^i) / i),$$

мы получим очень неэффективный алгоритм, требующий выполнения большого числа операций. Гораздо лучше организовать вычисления следующим образом: если обозначить числитель какого-либо слагаемого буквой p , то у следующего слагаемого числитель будет равен $-p*x$ (знак минус обеспечивает чередование знаков слагаемых), а само слагаемое m будет равно p/i , где i – номер слагаемого.

Алгоритм, записанный на псевдокоде	Блок-схема алгоритма
<p>алг Сумма (арг вещ x, Eps, рез вещ S) дано $0 < x < 1$ надо $S = x - x**2/2 + x**3/3 - \dots$ нач цел i, вещ m, p ввод x, Eps $S := 0; i := 1$ начальные значения $m := 1; p := -1$ нц пока $abs(m) > Eps$ $p := -p*x$ p – числитель очередного слагаемого $m := p/i$ m – очередное слагаемое $S := S + m$ S – частичная сумма $i := i + 1$ i – номер очередного слагаемого кц вывод S кон</p>	<pre> graph TD Start([начало]) --> Input[/x, Eps/] Input --> Init[S:=0; i:=1; m:=1; p:=-1] Init --> Decision{abs(m) > Eps} Decision -- да --> Loop[p:=-p*x; m:=p/i; S:=S+m; i:=i+1] Loop --> Decision Decision -- нет --> Output[/S/] Output --> End([конец]) </pre>

Алгоритм, в состав которого входит итерационный цикл, называется **итерационным алгоритмом**. Итерационные алгоритмы используются при реализации итерационных численных методов.

В итерационных алгоритмах необходимо обеспечить обязательное достижение условия выхода из цикла (**сходимость итерационного процесса**).

В противном случае произойдет "**зацикливание**" алгоритма, т.е. не будет выполняться основное свойство алгоритма – результативность.

2.6 Вложенные циклы

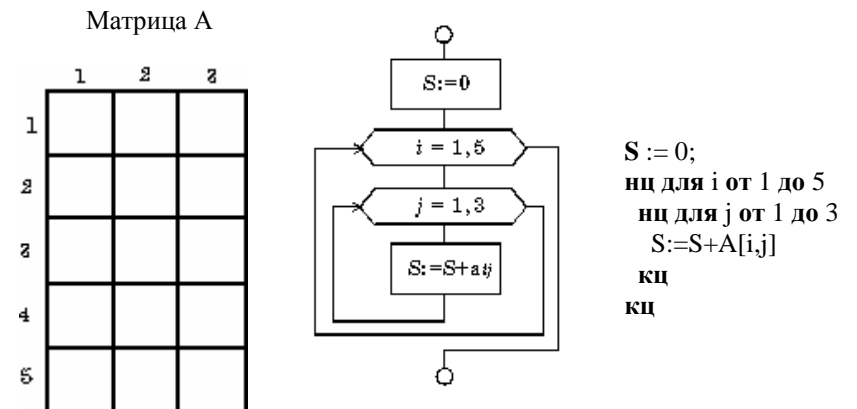
Возможны случаи, когда внутри тела цикла необходимо повторять некоторую последовательность операторов, т.е. организовать внутренний цикл.

Такая структура получила название цикла в цикле или **вложенных циклов**. Глубина вложения циклов (то есть количество вложенных друг в друга циклов) может быть различной.

При использовании такой структуры для экономии машинного времени необходимо выносить из внутреннего цикла во внешний все операторы, которые не зависят от параметра внутреннего цикла.

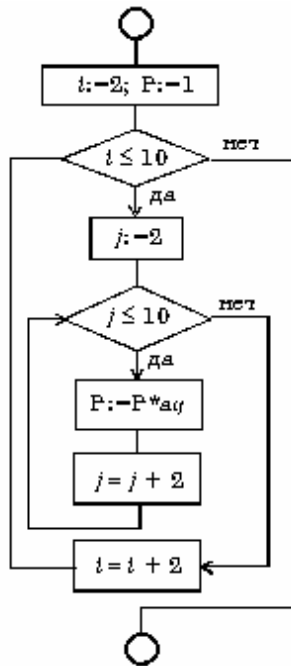
Пример вложенных циклов для

Вычислить сумму элементов заданной матрицы $A(5,3)$.



Пример вложенных циклов пока

Вычислить произведение тех элементов заданной матрицы $A(10,10)$, которые расположены на пересечении четных строк и четных столбцов.



```

i:=2; P:=1
нц пока i <= 10
  j:=2
  нц пока j <= 10
    P:=P*A[i,j]
    j:=j+2
  кц
  i:=i+2
кц
  
```

2.7 Языки программирования

2.7.1 Классификация языков программирования

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенная свобода при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы – компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на

понятном ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.

Следовательно, **язык для записи алгоритмов должен быть формализован.** Такой язык принято называть **языком программирования**, а запись алгоритма на этом языке – **программой для компьютера.**

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения.

Любой алгоритм есть последовательность предписаний, выполнив которые можно за конечное число шагов перейти от исходных данных к результату. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования – чем меньше детализация, тем выше уровень языка.

По этому критерию можно выделить следующие уровни языков программирования:

- машинные;
- машинно-ориентированные (ассемблеры);
- машинно-независимые (языки высокого уровня).

Машинные языки и машинно-ориентированные языки – это языки **низкого уровня**, требующие указания мелких деталей процесса обработки данных. Языки же **высокого уровня** имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на:

- **процедурные (алгоритмические)** (Basic, Pascal, C и др.), которые предназначены для однозначного описания алгоритмов; для решения задачи процедурные языки требуют в той или иной форме явно записать процедуру ее решения;
- **логические** (Prolog, Lisp и др.), которые ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи с тем, чтобы решение следовало из составленного описания;
- **объектно-ориентированные** (Object Pascal, C++, Java и др.), в основе которых лежит **понятие объекта, сочетающего в себе данные и действия над ними.** Программа на объектно-ориентированном языке, решая некоторую задачу, по сути описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы

взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

2.7.2 Машинные языки

Каждый компьютер имеет свой машинный язык, то есть свою совокупность машинных команд, которая отличается количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина и др.

При программировании на машинном языке программист может держать под своим контролем каждую команду и каждую ячейку памяти, использовать все возможности имеющихся машинных операций.

Но процесс написания программы на машинном языке очень **трудоемкий и утомительный**. Программа получается **громоздкой, труднообозримой, ее трудно отлаживать, изменять и развивать**.

Поэтому в случае, когда нужно иметь эффективную программу, в максимальной степени учитывающую специфику конкретного компьютера, вместо машинных языков используют близкие к ним машинно-ориентированные языки (ассемблеры).

Язык ассемблера – это машинно-зависимый язык низкого уровня, в котором короткие мнемонические имена соответствуют отдельным машинным командам. Используется для представления в удобочитаемой форме программ, записанных в машинном коде.

Язык ассемблера позволяет программисту пользоваться **текстовыми мнемоническими** (то есть легко запоминаемыми человеком) **кодами**, по своему усмотрению **присваивать символические имена регистрам компьютера и памяти, а также задавать удобные для себя способы адресации**. Кроме того, он позволяет использовать различные системы счисления (например, десятичную или шестнадцатеричную) для представления числовых констант, использовать в программе комментарии и др.

Программы, написанные на языке ассемблера, требуют значительно меньшего объема памяти и времени выполнения. Знание программистом языка ассемблера и машинного кода дает ему понимание архитектуры машины. Несмотря на то, что большинство специалистов в области программного обеспечения разрабатывают программы на языках высокого уровня, таких, как Object Pascal или C, наиболее мощное и эффективное

программное обеспечение полностью или частично написано на языке ассемблера.

Языки высокого уровня были разработаны для того, чтобы освободить программиста от учета технических особенностей конкретных компьютеров, их архитектуры. В противоположность этому, язык ассемблера разработан с целью учесть конкретную специфику процессора. Следовательно, для того, чтобы написать программу на языке ассемблера для конкретного компьютера, важно знать его архитектуру.

В качестве примера рассмотрим программу на языке ассемблера для IBM PC. Программа вычисляет значение $a = b + c$ для целых a , b и c :

	.MODEL	Директива .MODEL задает механизм
	.SMALL	распределения памяти под данные и
	.DATA	команды.
b	DW 5	Директива .DATA определяет начало
c	DW 3	участка программы с данными.
a	DW ?	Директивы DW задают типы переменных
	.CODE	и их значения.
begin	MOV AX,@DATA	Директива .CODE определяет
	MOV DS,AX	начало участка программы с
	MOV AX,B	командами.
	ADD AX,C	Команды MOV AX,@DATA и MOV
	MOV A,AX	DS,AX записывают адрес сегмента данных
	MOV AH,4CH	в регистр DS (Data Segment).
	INT 21H	Для вычисления a
	END begin	используются команды MOV AX,
		B, ADD AX,C и MOV A,AX.
		В директиве END задана метка первой
		выполняемой программы программы begin.

Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется **ассемблером** и является простейшим транслятором.

2.7.3 Алгоритмические языки

Основные преимущества алгоритмических языков перед машинными таковы:

- алфавит **алгоритмического языка** **значительно шире алфавита машинного языка**, что существенно повышает наглядность текста программы;
- **набор операций**, допустимых для использования, **не зависит от набора машинных операций**, а выбирается из соображений удобства формулирования алгоритмов решения задач определенного класса;
- **формат предложений** достаточно **гибок и удобен** для использования, что позволяет с помощью одного предложения задать достаточно содержательный этап обработки данных;
- требуемые операции задаются с помощью **общепринятых математических обозначений**;
- **данным в алгоритмических языках присваиваются индивидуальные имена**, выбираемые программистом;
- в языке может быть предусмотрен значительно **более широкий набор типов данных** по сравнению с набором машинных типов данных.

Таким образом, алгоритмические языки в значительной мере являются **машинно-независимыми**. Они облегчают **работу программиста и повышают надежность создаваемых программ**.

2.8 Составляющие алгоритмического языка

2.8.1 Основные понятия

Алгоритмический язык (как и любой другой язык) образуют три его составляющие: **алфавит, синтаксис и семантика**.

Алфавит – это фиксированный для данного языка набор основных символов, т.е. "букв", из которых должен состоять любой текст на этом языке – никакие другие символы в тексте не допускаются.

Синтаксис – это правила построения фраз, позволяющие определить, правильно или неправильно написана та или иная фраза.

Семантика определяет смысловое значение предложений языка.

Основными понятиями в алгоритмических языках обычно являются следующие.

1 **Имена** (идентификаторы) – употребляются для обозначения объектов программы (переменных, массивов, функций и др.).

2 **Операции**. Типы операций:

- арифметические операции + , - , * , / и др. ;
- логические операции и , или , не ;
- операции отношения < , , <= , = , > , >= ;

- операция сцепки (иначе "присоединения") символьных значений друг с другом с образованием одной длинной строки; изображается знаком "+".

3 **Данные** – величины, обрабатываемые программой. Имеется три основных вида данных: константы, переменные и массивы.

- Константы – это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения.

Примеры констант:

- числовые: 7.5 , 12 ;
- логические: да (истина), нет (ложь);
- символьные (содержат ровно один символ): "А" , "+" ;
- литерные (содержат произвольное количество символов) "a0", "Мир", "" (пустая строка).

- Переменные обозначаются именами и могут изменять свои значения в ходе выполнения программы. Переменные бывают целые, вещественные, логические, символьные и литерные.

- Массивы – последовательности однотипных элементов, число которых фиксировано и которым присвоено одно имя. Положение элемента в массиве однозначно определяется его индексами (одним, в случае одномерного массива, или несколькими, если массив многомерный). Иногда массивы называют таблицами.

4 **Выражения** – предназначены для выполнения необходимых вычислений, состоят из констант, переменных, указателей функций (например, $\exp(x)$), объединенных знаками операций.

Выражения записываются в виде линейных последовательностей символов (без подстрочных и надстрочных символов, "многоэтажных" дробей и т.д.), что позволяет вводить их в компьютер, последовательно нажимая на соответствующие клавиши клавиатуры.

Различают выражения арифметические, логические и строковые.

- Арифметические выражения служат для определения одного числового значения. Например, $(1 + \sin(x))/2$. Значение этого выражения при $x = 0$ равно 0.5, а при $x = \pi/2$ – единице.

- Логические выражения описывают некоторые условия, которые могут выполняться или не выполняться. Логическое выражение может принимать только два значения – "истина" или "ложь" (да или нет). Например, логическое выражение $x*x + y*y < r*r$ определяет принадлежность точки с координатами (x, y) внутренней области круга радиусом r с центром в начале координат. При $x = 1, y = 1, r = 2$ значение этого выражения – "истина", а при $x = 2, y = 2, r = 1$ – "ложь".

- Строковые (литерные) выражения, значениями которых являются тексты. В строковые выражения могут входить литерные и строковые константы, литерные и строковые переменные, литерные функции, разделенные знаками операции сцепки. Например, $A + B$ означает присоединение строки B к концу строки A . Если $A = \text{"куст"}$, а $B = \text{"зеленый"}$, то значение выражения $A + B$ есть "куст зеленый" .

5 Операторы (команды). Оператор – это наиболее крупное и содержательное понятие языка: каждый оператор представляет собой законченную фразу языка и определяет некоторый вполне законченный этап обработки данных. В состав операторов входят:

- ключевые слова;
- данные;
- выражения и т.д.

Операторы подразделяются на исполняемые и неисполняемые. Неисполняемые операторы предназначены для описания данных и структуры программы, а исполняемые – для выполнения различных действий (например, оператор присваивания, операторы ввода и вывода, условный оператор, оператор цикла, оператор процедуры и др.).

2.8.2 Стандартная функция

При решении различных задач с помощью компьютера бывает необходимо вычислить логарифм или модуль числа, синус угла и т.д.

Вычисления часто употребляемых функций осуществляются посредством подпрограмм, называемых *стандартными функциями*, которые заранее запрограммированы и встроены в транслятор языка (табл. 5).

Таблица 5

Название и математическое обозначение функции		Указатель функции
Абсолютная величина (модуль)	$ x $	abs(x)
Корень квадратный	\sqrt{x}	sqrt(x)
Натуральный логарифм	$\ln x$	ln(x)
Десятичный логарифм	$\lg x$	lg(x)
Экспонента (степень числа $e \sim 2.72$)	e^x	exp(x)
Знак числа x (– 1, если $x < 0$; 0, если $x = 0$; 1, если $x > 0$)	sign x	sign(x)

Целая часть x (т.е. максимальное целое число, не превосходящее x)		int(x)
Минимум из чисел x и y		min(x,y)
Максимум из чисел x и y		max(x,y)
Частное от деления целого x на целое y		div(x,y)
Остаток от деления целого x на целое y		mod(x,y)
Случайное число в диапазоне от 0 до x – 1		rnd(x)
Синус (угол в радианах)	sin x	sin(x)
Косинус (угол в радианах)	cos x	cos(x)
Тангенс (угол в радианах)	tg x	tg(x)
Котангенс (угол в радианах)	ctg x	ctg(x)
Арксинус (главное значение в радианах)	arcsin x	arcsin(x)
Арккосинус (главное значение в радианах)	arccos x	arccos(x)
Арктангенс (главное значение в радианах)	arctg x	arctg(x)
Арккотангенс (главное значение в радианах)	arcctg x	arcctg(x)

, . :

sin (3.05) sin (x) sin (2 * y + t / 2) sin((exp(x) + 1) ** 2)
min (a, 5) min (a, b) min (a + b , a * b) min(min(a, b), min(c, d))

2.8.3 Арифметические выражения

Арифметические выражения (табл. 6) записываются по следующим правилам:

- Нельзя опускать знак умножения между сомножителями и ставить рядом два знака операций.
- Индексы элементов массивов записываются в квадратных (Pascal) или круглых (Basic) скобках.
- Для обозначения переменных используются буквы латинского алфавита.
- Операции выполняются в порядке старшинства: сначала вычисление функций, затем возведение в степень, потом умножение и деление и в последнюю очередь – сложение и вычитание.
- Операции одного старшинства выполняются слева направо.

Математическая запись	Запись на псевдокоде
$\frac{xy}{z}$	<code>x * y / z</code>
$\frac{x}{yz}$	<code>x / (y * z)</code> или <code>x / y / z</code>
$\frac{a^3 + b^3}{bc}$	<code>(a**3 + b**3) / (b*c)</code>
$\frac{a_{i+1} + b_{i-1}}{2xy}$	<code>(a[i+1] + b[i-1]) / (2*x*y)</code>
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	<code>(-b + sqrt(b*b - 4*a*c)) / (2*a)</code>
$\sqrt[5]{x} \quad (x < 0)$	<code>sign(x) * abs(x) ** (1/5)</code>
$0,49 e^{a^2 - b^2} + \ln^3 \cos a^2$	<code>0.49 * exp(a*a - b*b) + ln(cos(a*a)) ** 3</code>
$\frac{x}{1 + \frac{x^2}{3 + (2x)^3}}$	<code>x / (1 + x*x / (3 + (2*x)**3))</code>

Типичные ошибки в записи выражений:

$5x + 1$ Пропущен знак умножения между 5 и x
 $a + \sin x$ Аргумент x функции $\sin x$ не заключен в скобки
 $((a + b)/c)**3$ Не хватает закрывающей скобки

2.8.4 Логические выражения

В записи логических выражений (табл. 7) кроме арифметических операций сложения, вычитания, умножения, деления и возведения в степень используются операции отношения $<$ (меньше), $<=$ (меньше или равно), $>$ (больше), $=$ (больше или равно), $=$ (равно), $<$ (не равно), а также логические операции и, или, не.

Примеры записи логических выражений, истинных при выполнении указанных условий.

Таблица 7

Условие	Запись на псевдокоде
Дробная часть вещественного числа a равна нулю	<code>int(a) = 0</code>

Целое число a – четное	$\text{mod}(a, 2) = 0$
Целое число a – нечетное	$\text{mod}(a, 2) = 1$
Целое число k кратно семи	$\text{mod}(a, 7) = 0$
Каждое из чисел a, b положительно	$(a0)$ и $(b0)$
Только одно из чисел a, b положительно	$((a0) \text{ и } (b<=0))$ или $((a<=0) \text{ и } (b0))$
Хотя бы одно из чисел a, b, c является отрицательным	$(a < 0)$ или $(b < 0)$ или $(c < 0)$
Число x удовлетворяет условию $a < x < b$	(xa) и $(x<b)$
Число x имеет значение в промежутке $[1, 3]$	$(x=1)$ и $(x<=3)$
Целые числа a и b имеют одинаковую четность	$((\text{mod}(a, 2)=0) \text{ и } (\text{mod}(b, 2)=0))$ или $((\text{mod}(a, 2)=1) \text{ и } (\text{mod}(b, 2)=1))$
Точка с координатами (x, y) лежит в круге радиуса r с центром в точке (a, b)	$(x-a)**2 + (y-b)**2 < r*r$
Уравнение $ax^2 + bx + c = 0$ не имеет действительных корней	$b*b - 4*a*c < 0$
Точка (x, y) принадлежит первой или третьей четверти	$((x0) \text{ и } (y0))$ или $((x<0) \text{ и } (y0))$
Точка (x, y) принадлежит внешности единичного круга с центром в начале координат или его второй четверти	$(x*x + y*y > 1)$ или $((x*x + y*y <= 1) \text{ и } (x<0) \text{ и } (y0))$

Продолжение табл. 7

Условие	Запись на псевдокоде
Целые числа a и b являются взаимно противоположными	$a = -b$
Целые числа a и b являются взаимно обратными	$a*b = 1$
Число a больше среднего арифметического чисел b, c, d	$a > (b+c+d)/3$
Число a не меньше среднего геометрического чисел b, c, d	$a = (b+c+d) ** (1/3)$
Хотя бы одна из логических переменных F_1 и F_2 имеет значение да	$F1$ или $F2$
Обе логические переменные F_1 и F_2 имеют	$F1$ и $F2$

значение да	
Обе логические переменные F_1 и F_2 имеют значение нет	не F_1 и не F_2
Логическая переменная F_1 имеет значение да , а логическая переменная F_2 имеет значение нет	F_1 и не F_2
Только одна из логических переменных F_1 и F_2 имеет значение да	(F_1 и не F_2) или (F_2 и не F_1)

2.9 Упражнения

1 :

$$\begin{array}{ll}
 \text{а)} \quad \frac{x+y}{x-1/2} - \frac{x-z}{xy}; & \text{е)} \quad \frac{\sqrt{|\sin^2 x|}}{3,01x - e^{2x}}; \\
 \text{б)} \quad (1+z) \frac{x + \frac{y}{z}}{a - \frac{1}{1+x^2}}; & \text{ж)} \quad \frac{|\cos x^2 - \sin^2 x|}{\sqrt[4]{|\ln x|} + xy}; \\
 \text{в)} \quad (x^n)^{m+2} + x^{n^m}; & \text{з)} \quad \ln \left(y^{-\sqrt{|x+1|}} \right) \sin^2 \operatorname{arctg} z; \\
 \text{г)} \quad \frac{(a+b)^n}{1 + \frac{a}{a^m - b^{m-n}}}; & \text{и)} \quad r_{ij}^{|x-y|} - 0,15 \left| \sin e^{-z^8} \right|; \\
 \text{д)} \quad \frac{(a_i^{2l} + b_{j+1}^{2k})(3^n - x^2 y)}{z - \frac{d_{ij} + 1}{x + \frac{y}{\sqrt{t^2 + xyz}}}}; & \text{к)} \quad a^{(x+y)/2} - 3 \sqrt{\frac{x-1}{|y|+1}} e^{-(y+u/2)}.
 \end{array}$$

2 :

$$\begin{array}{ll}
 \text{а)} \quad a / b^{**2}; & \text{л)} \quad 5 * \operatorname{arctg}(x) - \operatorname{arctg}(y) / 4; \\
 \text{б)} \quad a + b / c + 1; & \text{м)} \quad \lg(u * (1/3) + \sqrt{v} + z); \\
 \text{в)} \quad 1 / a * b / c; & \text{н)} \quad \ln(y * (-\sqrt{\operatorname{abs}(x)})); \\
 \text{г)} \quad a^{**b^{**c/2}}; & \text{о)} \quad \operatorname{abs}(x^{**}(y/x) - (y/x)^{**}(1/3)); \\
 \text{д)} \quad (a^{**b})^{**c/2}; & \text{п)} \quad \sqrt{(x1 - x2)^{**2} + (y1 - y2)^{**2}};
 \end{array}$$

- е) $a/b/c/d * p * q$; п) $\exp(\text{abs}(x - y)) * (\text{tg}(z) ** 2 + 1) ** x$;
ж) $x ** y ** z / a / b$; с) $\lg(\sqrt{\exp(x - y)} + x ** \text{abs}(y) + z)$;
з) $4/3 * 3,14 * r ** 3$; т) $\sqrt{\exp(a * x) * \sin(x) ** n} / \cos(x) ** 2$;
и) $b / \sqrt{a * a + b}$; у) $\sqrt{\sin(\arctg(u)) ** 2 + \text{abs}(\cos(v))}$;
к) $d * c / 2 / R + a ** 3$; ф) $\text{abs}(\cos(x) + \cos(y)) ** (1 + \sin(y) ** 2)$.

3 Вычислите значения арифметических выражений при $x = 1$:

а) $\text{abs}(x - 3) / \ln(\exp(3)) * 2 / \lg(10000)$;

Решение: $\text{abs}(1 - 3) = 2$; $\ln(\exp(3)) = 3$; $\lg(10000) = 4$; $2/3 * 2/4 = 0,33$;

б) $\text{sign}(\sqrt{\sqrt{x+15}}) * 2 ** 2 ** 2$;

в) $\text{int}(-2, 1) * \text{int}(-2, 9) / \text{int}(2, 9) + x$;

г) $-\sqrt{x+3} ** 2 ** (\text{sign}(x+0,5) * 3) + \text{tg}(0)$;

д) $\lg(x) + \cos(x ** 2 - 1) * \sqrt{x+8} - \text{div}(2, 5)$;

е) $\text{sign}(x - 2) * \sqrt{\text{int}(4, 3)} / \text{abs}(\min(2, -1))$;

ж) $\text{div}(10, x+2) * \text{mod}(10, x+6) / \max(10, x) * \text{mod}(2, 5)$.

4 , :

а) площадь треугольника со сторонами a, b, c ($a, b, c > 0$) и полупериметром p ;

О т в е т : $\sqrt{p * (p - a) * (p - b) * (p - c)}$;

б) среднее арифметическое и среднее геометрическое чисел a, b, c, d ;

в) расстояние от точки с координатами (x, y) до точки $(0, 0)$;

г) синус от x градусов;

д) площадь поверхности куба (длина ребра равна a);

е) радиус описанной сферы куба (длина ребра равна a);

ж) координаты точки пересечения двух прямых, заданных уравнениями $a_1x + b_1y + c_1 = 0$ и $a_2x + b_2y + c_2 = 0$ (прямые не параллельны).

5 :

а) $x * x + y * y \leq 9$ при $x = 1, y = -2$

О т в е т : **да**;

б) $b * b - 4 * a * c < 0$ при $a = 2, b = 1, c = -2$;

в) $(a = 1)$ и $(a \leq 2)$ при $a = 1,5$;

г) $(a < 1)$ или $(a > 1,2)$ при $a = 1,5$;

д) $(\text{mod}(a, 7) = 1)$ и $(\text{div}(a, 7) = 1)$ при $a = 8$;

е) **не** $((a < b)$ и $(a < 9)$ или $(a * a = 4))$ при $a = 5, b = 4$.

6 , :

а) x принадлежит отрезку $[a, b]$

О т в е т : $(x = a)$ и $(x < = b)$;

б) x лежит вне отрезка $[a, b]$;

в) x принадлежит отрезку $[a, b]$ или отрезку $[c, d]$;

г) x лежит вне отрезков $[a, b]$ и $[c, d]$;) целое k является нечетным числом;

е) целое k является трехзначным числом, кратным пяти;

ж) элемент $a_{i,j}$ двумерного массива находится на пересечении нечетной строки и четного столбца;

з) прямые $a_1x + b_1y + c_1 = 0$ и $a_2x + b_2y + c_2 = 0$ параллельны;

и) из чисел a, b, c меньшим является c , а большим b ;

к) среди чисел a, b, c, d есть взаимно противоположные;

л) среди целых чисел a, b, c есть хотя бы два четных;

м) из отрезков с длинами a, b, c можно построить треугольник;

н) треугольники со сторонами a_1, b_1, c_1 и a_2, b_2, c_2 подобны;

о) точка с координатами (x, y) принадлежит внутренней области треугольника с вершинами $A(0, 5)$, $B(5, 0)$ и $C(1, 0)$;

п) точка с координатами (x, y) принадлежит области, внешней по отношению к треугольнику с вершинами $A(0,5)$, $B(1,0)$ и $C(5,0)$;

р) четырехугольник со сторонами a, b, c и d является ромбом.

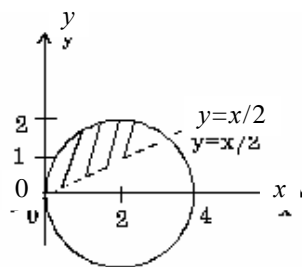
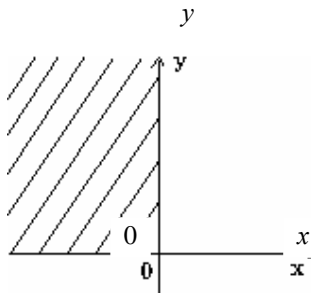
7 Начертите на плоскости (x, y) область, в которой и только в которой истинно указанное выражение. Границу, не принадлежащую этой области, изобразите пунктиром.

а) $(x < = 0)$ и $(y = 0)$

е) $((x - 2)**2 + y*y < = 4)$ и $(y = x/2)$

:

:



б) $(x = 0)$ или $(y < = 0)$

ж) $(x*x + y*y < 1)$ и $(y = x*x)$;

в) $x + y = 0$

з) $(y = x)$ и $(y + x = 0)$ и $(y < = 1)$;

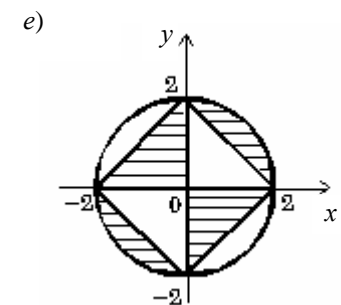
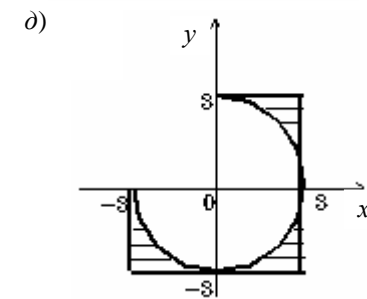
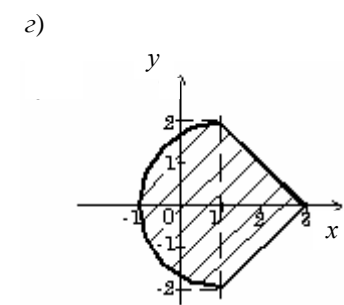
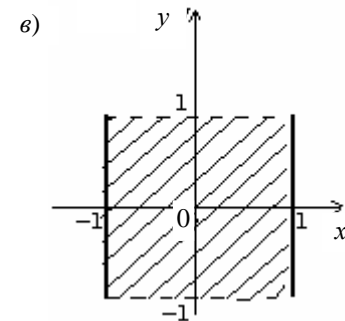
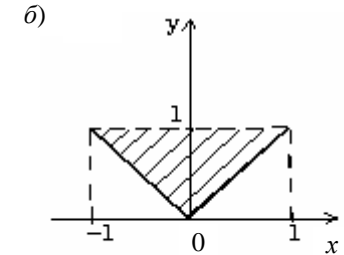
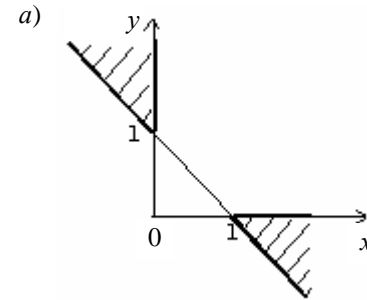
$$\text{г) } (x+y=0) \text{ и } (y<0)$$

$$\text{д) } \text{abs}(x) + \text{abs}(y) = 1$$

$$\text{и) } (\text{abs}(x) \leq 1) \text{ и } (y < 2);$$

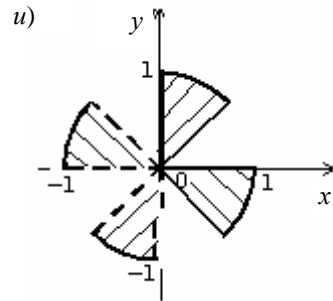
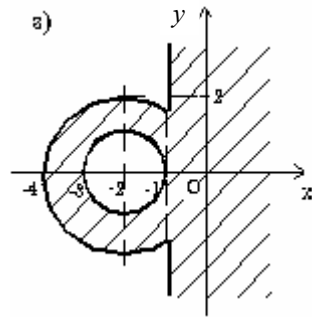
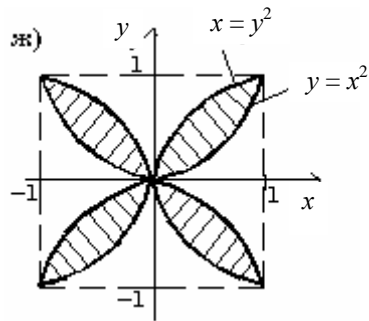
$$\text{к) } (x^2 + y^2 < 4) \text{ и } (x^2 + y^2 \geq 1);$$

8 Запишите логическое выражение, которое принимает значение "истина" тогда и только тогда, когда точка с координатами (x, y) принадлежит заштрихованной области.



ж)

з)



9 Пусть $a = 3$, $b = 5$, $c = 7$. Какие значения будут иметь эти переменные в результате выполнения последовательности операторов:

а) $a := a + 1$; $b := a + b$; $c := a + b$; $a := \text{sqrt}(a)$

Решение: $a = 3 + 1 = 4$, $b = 4 + 5 = 9$, $c = 4 + 9 = 13$, $a = \{\text{корень квадратный из } 4\} = 2$.

О т в е т: $a = 2$, $b = 9$, $c = 13$;

б) $c := a * b + 2$; $b := b + 1$; $a := c - b ** 2$; $b := b * a$;

в) $b := b + a$; $c := c + b$; $b := 1 / b * c$;

г) $p := c$; $c := b$; $b := a$; $a := p$; $c := a * b * c * p$;

д) $c := a ** (b - 3)$; $b := b - 3$; $a := (c + 1) / 2 * b$; $c := (a + b) * a$;

е) $x := a$; $a := b$; $b := c$; $c := x$; $a := \text{sqrt}(a + b + c + x - 2)$;

ж) $b := (a + c) ** 2$; $a := \lg(b ** 2) ** 2$; $c := c * a * b$.

10 Задайте с помощью операторов присваивания следующие действия:

а) массив $X = (x_1, x_2)$ преобразовать по правилу: в качестве x_1 взять сумму, а в качестве x_2 — произведение исходных компонент;

Решение: $c := x[1]$; $x[1] := x[1] + x[2]$;

$x[2] := c * x[2]$

б) поменять местами значения элементов массива $X = (x_1, x_2)$;

в) в массиве $A(N)$ компоненту с номером i ($1 < i < N$) заменить полусуммой исходных соседних с нею компонент, соседнюю справа компоненту заменить на нуль, а соседнюю слева компоненту увеличить на 0,5;

$$\text{г) } u = \max(x, y, z) + \min(x - z, y + z, y, z);$$

11 Задайте с помощью команд **если** или **выбор** вычисления по формулам:

$$\text{а) } y = \begin{cases} \sqrt[7]{x}, & \text{если } x \leq -100, \\ \sqrt[3]{x}, & \text{если } -100 < x < 100, \\ \sqrt{x}, & \text{если } x \geq 100; \end{cases}$$

$$\text{б) } z = \begin{cases} x^2 + y^2, & \text{если } x^2 + y^2 \leq 1, \\ x + y, & \text{если } x^2 + y^2 > 1 \text{ и } y \geq x, \\ 0,5, & \text{если } x^2 + y^2 > 1 \text{ и } y < x; \end{cases}$$

$$F(z) = \begin{cases} 2z + 1, & \text{если } z \geq 0, \\ \sin z, & \text{если } z < 0, \end{cases}$$

где

$$\text{в) } z = \begin{cases} \lg(-x), & \text{если } x < 0, \\ \sqrt{x-1}, & \text{если } x \geq 0, \end{cases}$$

$$\text{г) } z = \begin{cases} 1, & \text{если } c = 0, \\ x, & \text{если } c = 1, \\ 3x^2 - 1/2, & \text{если } c = 2, \\ x^3 - 3x/2, & \text{если } c = 3, \\ 2x^4 - 3x/2, & \text{в противном случае} \end{cases}$$

$$\text{д) } z = \begin{cases} \sqrt{x^2 + y^2}, & \text{если } |x| + |y| < r, \\ \max(|x|, |y|), & \text{если } |x| + |y| \geq r, \end{cases}$$

$$е) \quad v = \begin{cases} x + y, & \text{если } x > 1 \text{ и } y > 1, \\ x - y, & \text{если } x > 1 \text{ и } y \leq 1, \\ -x + y, & \text{если } x \leq 1 \text{ и } y > 0, \\ -x - y, & \text{если } x \leq 1 \text{ и } y \leq 0, \end{cases}$$

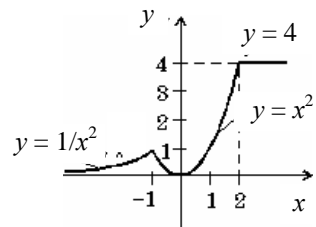
$$ж) \quad z = \begin{cases} |x| + |y| & \text{если точка лежит внутри круга радиусом } r \ (r > 0) \\ x + y & \text{с центром в точке } (a, b) \text{ в противном случае} \end{cases}$$

12 Постройте графики функций $y(x)$, заданных командами **если**:

а) **если** $x \leq -1$
то $y := 1/x^{**}2$
иначе
если $x \leq 2$
то $y := x * x$
иначе $y := 4$
все
все

в) **если** $x < -0.5$
то $y := 1/\text{abs}(x)$
иначе
если $x < 1$
то $y := 2$
иначе $y := 1/(x - 0.5)$
все
все

Решение



г) **если** $x < 0$
то $y := 1$
иначе
если $x < 3.14$
то $y := \cos(x)$
иначе $y := -1$
все
все

б) **если** $x < -5$
то $y := -5$
иначе
если $x < 0$
то $y := x$
иначе
если $x < 3$
то $y := 2 * x$
иначе $y := 6$
все
все
все

д) **если** $\text{abs}(x) > 2$
то $y := x * x$
иначе
если $x < 0$
то $y := -2 * x$
иначе
если $x = 1$
то $y := 4$
иначе $y := 4 * x * x$
все
все
все

а) S:=128
 нц для i
 от 1 до 4
 S:=div(S,2)
 кц

Решение	
i	S
	128
1	128/2=64
2	64/2=32
3	32/2=16
4	16/2=8
Ответ: S=8	

б) S:=1; a:=1
 нц для i от 1 до 3
 S:=S+i*(i+1)*a
 a:=a+2
 кц

в) S:=1; a:=1
 нц для i от 1 до 3
 S := S+i
 нц для j от 2 до 3
 S := S+j
 кц
 кц

14 S :

а) i:=0; S:=0
 нц пока i<3
 i:=i+1;
 S:=S+i*i
 кц

Решение		
Условие i < 3	i	S
	0	0
0 < 3? Да	1	0+1 ² =1
1 < 3? Да	2	1+2 ² =5
2 < 3? Да	3	5+3 ² =14
3 < 3? Нет(кц)		
Ответ: S=14		

г) S:=0
 нц для i от 1 до 2
 нц для j от 2 до 3
 S:=S+i+j
 кц
 кц

Решение		
i	j	S
		0
1	2	0+1+2=3
	3	3+1+3=7
2	2	7+2+2=11
	3	11+2+3=16
Ответ: S=16		

д) нц для i от 1 до 3
 S:=0
 нц для j от 2 до 3
 S:=S+i+j
 кц
 кц

е) нц для i от 1 до 2
 S := 0
 нц для j от 2 до 3
 нц для k от 1 до 2
 S := S+i+j+k
 кц
 кц
 кц

г) S:=0; N:=125
 нц пока N≠0
 S:=S+mod(N,10) | S – сумма цифр
 N:=div(N,10) | числа N
 кц

Решение		
Условие N 0	S	N
	0	125
125 ≠ 0? Да	0+5=5	12
12 ≠ 0? Да	5+2=7	1
1 ≠ 0? Да	7+1=8	0
0 ≠ 0? нет (кц)		
Ответ: S=8		

б) S:= 0; i:=1

нц пока i l

S:=S+1/i

i:= i –1

кц

в) S:= 0; i:=1; j:= 5

нц пока i<j

S:= S+i*j

i:=i+1

j:=j–1

кц

д) a:= 1; b:= 1; S:= 0;

нц пока a<=5

a:= a + b; b:= b+a;

S:=S+a+b

кц

е) a:=1; b:= 1

нц пока a+b<10

a:=a+1

b:=b+a

кц

S:=a+b

15 Составьте алгоритмы решения задач линейной структуры:

а) в треугольнике известны три стороны a , b и c ; найти (в градусах) углы этого треугольника, используя формулы:

$$\cos \alpha = \frac{b^2 + c^2 - a^2}{2bc}, \quad \sin \beta = \frac{b \sin \alpha}{a}, \quad C = 180^\circ - (\alpha + \beta).$$

Пояснение. Обратите внимание на то, что стандартные тригонометрические функции \arccos и \arcsin возвращают вычисленное значение в радианной мере.

Решение:

алг Углы треугольника (**арг вещь** a, b, c , **рез вещь** $UgolA, UgolB, UgolC$)

нач вещь $RadGr, UgolARad$

| $RadGr$ – коэф. перевода угла из радианной меры в градусную

| $UgolARad$ – угол A (в радианах)

$RadGr:=180/3.14$

$UgolARad:=ArcCos((b*b+c*c-a*a)/(2*b*c))$

$UgolA:=UgolARad*RadGr$

$UgolB:=ArcSin(b*\sin(UgolARad)/a)*RadGr$

$UgolC:=180-(UgolA+UgolB)$

кон

б) в треугольнике известны две стороны a, b и угол C (в радианах) между ними; найти сторону c , углы A и B (в радианах) и площадь треугольника, используя формулы:

$$\sin \alpha = \frac{a \sin \gamma}{c}; \quad \sin \beta = \frac{b \sin \gamma}{c};$$

$$S = \frac{bc \sin \alpha}{2}; \quad c^2 = a^2 + b^2 - 2ab \cos C.$$

Пояснение. Сначала нужно найти сторону c , а затем остальные требуемые значения;

в) в треугольнике известны три стороны a , b и c ; найти радиус описанной окружности и угол A (в градусах), используя формулы:

$$S = \frac{bc \sin \alpha}{2}; \quad \operatorname{tg} \frac{\alpha}{2} = \sqrt{\frac{(p-b)(p-c)}{p(p-a)}},$$

где $p = \frac{a+b+c}{2}$.

г) в правильной треугольной пирамиде известны сторона основания a и угол A (в градусах) наклона боковой грани к плоскости основания; найти объем и площадь полной поверхности пирамиды, используя формулы:

$$V = S_{\text{осн}} H / 3; \quad S_n = S_{\text{осн}} \left(1 + \frac{1}{\cos \alpha} \right),$$

где $S_{\text{осн}} = \frac{a^2 \sqrt{3}}{4}$.

д) в усеченном конусе известны радиусы оснований R и r и угол A (в радианах) наклона образующей к поверхности большего основания; найти объем и площадь боковой поверхности конуса, используя формулы:

$$V = \frac{1}{3} \pi H (r^2 + R^2 + rR); \quad S_{\text{бок}} = \pi l (r + R),$$

где $H = (R - r) \operatorname{tg} \alpha$; $l = \frac{R - r}{\cos \alpha}$.

е) в правильной четырехугольной пирамиде сторона основания равна a , а боковое ребро наклонено к плоскости основания под углом A ; найти объем и площадь полной поверхности пирамиды и площадь сечения, проходящего через вершину пирамиды и диагональ основания d ; использовать формулы:

$$H = \frac{a\sqrt{2}}{2} \operatorname{tg} \alpha; \quad V = \frac{1}{3} S_{\text{осн}} H;$$

$$S_{\text{сеч}} = \frac{1}{2} H a \sqrt{2} = \frac{aH\sqrt{2}}{2}; \quad S_{\text{полн}} = a^2 + a^2 \sqrt{2 \operatorname{tg}^2 \alpha + 1}$$

16 :

а) определить, является ли треугольник с заданными сторонами a , b , c равнобедренным;
:

алг Треугольник(**арг** вещ a , b , c , **рез** лог **Ответ**)

дано | $a0$, $b0$, $c0$, $a + bc$, $a + cb$, $b + ca$

надо | **Ответ** = да, если треугольник равнобедренный

| Otvet = нет, если треугольник не равнобедренный

нач

если ($a = b$) **или** ($a = c$) **или** ($b = c$)

то Otvet:= да

иначе Otvet:= нет

все

кон

б) определить количество положительных чисел среди заданных чисел a , b и c ;

в) меньшее из двух заданных неравных чисел увеличить вдвое, а большее оставить без изменения;

г) числа a и b – катеты одного прямоугольного треугольника, а c и d – другого; определить, являются ли эти треугольники подобными;

д) даны три точки на плоскости; определить, какая из них ближе к началу координат;

е) определить, принадлежит ли заданная точка (x, y) плоской фигуре, являющейся кольцом с центром в начале координат, с внутренним радиусом r_1 и внешним радиусом r_2 ;

ж) упорядочить по возрастанию последовательность трех чисел a , b и c .

СПИСОК Литературы

1. Кушниренко А. Г. и др. Информатика. М.: Дрофа, 1998.
2. Кузнецов А. А. и др. Основы информатики. М.: Дрофа, 1998.
3. Гейн А. Г., Сенокосов А. И. Информатика. М.: Дрофа, 1998.
4. Сенокосов А. И., Гейн А. Г. Информатика для VIII – IX классов школ с углубленным изучением информатики. М.: Просвещение, 1995.
5. Ляхович В. Ф. Информатика для X–XI классов. М.: Просвещение, 1997.
6. Лебедев Г. В., Кушниренко А. Г. 12 лекций по преподаванию курса информатики. М.: Дрофа, 1998.
7. Ландо С. К. Алгоритмика. Методическое пособие. М.: Дрофа, 1997.
8. Каймин В. А. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1989.
9. Бордовский Г. А. Информатика в понятиях и терминах. М.: Просвещение, 1991.
10. Вычислительная техника и программирование. / Под ред. А. В. Ретрова. М.: Высш. шк., 1990.
11. Боглаев Ю. П. Вычислительная математика и программирование. М.: Высш. шк., 1990.
12. Каранчук В. П. Основы применения ЭВМ. М.: Радио и связь, 1988.
13. Власов В. К. и др. Элементы информатики. М.: Наука, 1988.
14. Абрамов С. А. и др. Задачи по программированию. М.: Наука, 1988.
15. Дудников Е. Е. Персональные компьютеры. М.: МНИИПУ, 1988.
16. Фигурнов В. Э. IBM PC для пользователя. М.: Наука, 1989.