

Министерство образования Российской Федерации
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет математики и информатики

ИНФОРМАТИКА

КОМПЬЮТЕРНЫЙ ПРАКТИКУМ
ДЛЯ СТУДЕНТОВ ЭНЕРГЕТИЧЕСКИХ
СПЕЦИАЛЬНОСТЕЙ
ЗАОЧНОЙ ФОРМЫ ОБУЧЕНИЯ

Учебное пособие

Благовещенск

2004

*Печатается по решению
редакционно-издательского совета
факультета математики и информатики
Амурского государственного
университета*

Киселева А.Н., Чалкина Н.А. (составители)

Информатика. Компьютерный практикум для энергетических специальностей заочной формы обучения. Благовещенск.: Амурский гос. ун – т, 2004.

Предназначено для студентов энергетических специальностей заочной формы обучения, изучающих дисциплину «Информатика» на первом курсе. Пособие составлено на основе учебных программ по информатике для энергетических специальностей, содержит теоретические сведения, практические задания, вопросы для самоконтроля.

Пособие подготовлено авторским коллективом: А.Н. Киселева – лабораторные работы №1 - №4; Н.А. Чалкина – лабораторные работы №5 - №8, архитектура компьютерного практикума.

Рецензент: А.М. Емельянов, зав. кафедрой высшей математики ДальГАУ, д-р техн. наук, профессор

© Амурский государственный университет, 2004

ВВЕДЕНИЕ

Компьютерный практикум разработан в соответствии с государственным стандартом для студентов энергетического факультета заочной формы обучения.

Цель, которую поставили перед собой авторы, - помочь студентам как можно быстрее и основательнее овладеть практическими навыками по информатике.

Компьютерный практикум включает темы: «Основные приемы работы в операционной среде Windows», «Текстовый редактор Word», «Табличный процессор Excel», «Язык программирования Turbo Pascal 7.0».

По каждой теме приводятся теоретические сведения, технология работы.

Изучение тем завершается промежуточным самоконтролем по предлагаемым вопросам.

Лабораторная работа №1

Тема: Основы работы в среде Windows.

Цель работы: овладение практическими навыками работы с операционной системой Windows-98.

Краткая теория

Microsoft Windows – это высокопроизводительная, многозадачная и многопоточная 32-разрядная операционная система с графическим интерфейсом и расширенными сетевыми возможностями, работающая в защищенном режиме, поддерживающая 16-разрядные приложения без всякой их модификации.

Напомним, что *операционная система* – это главная программа, которая управляет работой всех компонентов персонального компьютера.

Преимущество Microsoft Windows в том, что программы, написанные под нее, имеют сходные меню и управление.

Кроме того, Windows легко меняется. Ее можно настраивать для каждого пользователя, его нужд и вкусов. Она позволяет менять цвета, шрифты, фон, звуковое сопровождение и внешний вид Рабочего стола. Также можно создавать пользовательские меню и ярлыки, чтобы вам было удобно работать.

Начать знакомство с Windows необходимо с Рабочего стола, который вы видите сразу после загрузки компьютера (рис. 1.1).

Рабочий стол (DeskTop) - все пространство экрана в среде Windows с расположенными на нем графическими объектами.

Рабочий стол – это просто папка на вашем жестком диске, хотя и особенная, если взглянуть на ее содержимое. Любая информация, которую вы поместили в эту папку, будет отображаться на Рабочем столе. Однако лучше использовать его рационально и не загромождать.



Рис. 1.1. Общий вид рабочего стола Windows

Вдоль одной из границ (чаще в нижней части) Рабочего стола находится *панель задач*.



Рис. 1.2. Панель задач

Панель задач содержит:

- кнопку *ПУСК* - *главное системное меню*;
- доступ ко всем открытым приложениям (на рис. 1.2. – кнопки приложений Microsoft Word и Microsoft Excel). При открытии приложения на панели задач появляется соответствующая открытому окну кнопка. При установке указателя мыши на кнопке появляется флажок-подсказка с названием программы или документа. Нажатие этой кнопки позволяет быстро перейти в выбранное окно. Кнопки можно использовать для переключения между окнами.
- пиктограммы специальных "фоновых" приложений: Системные часы, Звук, Индикатор текущей раскладки клавиатуры и др.

Щелчком мышью кнопки *Пуск* приводит к появлению всплывающего *Главного системного меню*.

Главное системное меню предназначено для быстрого запуска программ, поиска файлов, обеспечения доступа к справке, вызов панели

управления для настройки компьютера и др. Меню содержит в себе несколько пунктов, при подведении указателя мыши на одном из пунктов автоматически открывается подменю для выбора нужной операции (рис. 1.3).

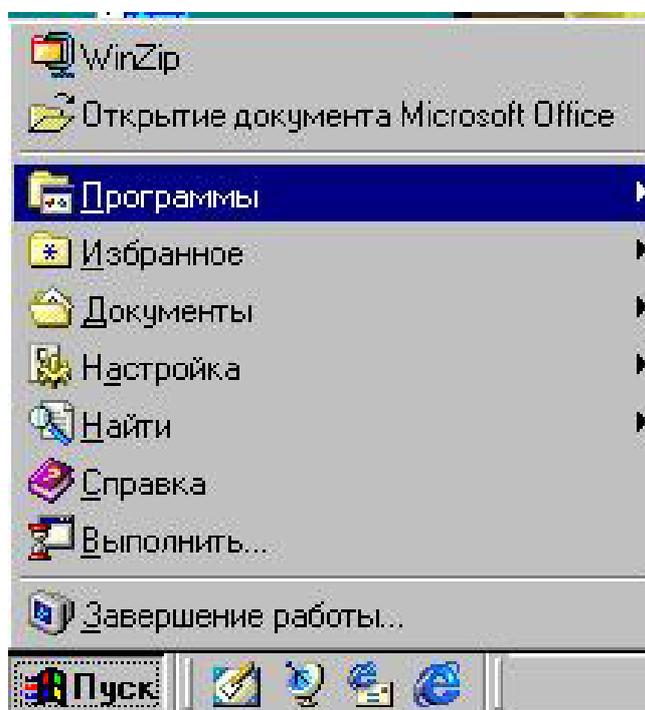


Рис. 1.3. Главное системное меню

1. Пункт *Программа* позволяет запускать программы. Содержит в себе подпункты:
 - *Автозагрузка*, с перечисленными программами и документами, которые будут автоматически открываться сразу после запуска Windows.
 - *Стандартные*, содержащий в себе стандартный набор прикладных программ Windows (служебные программы, игры, элементарные текстовый редактор Блокнот, графический редактор Paint, калькулятор).
2. Пункт *Документы* содержит ярлыки 15 последних открывавшихся документов.
3. Пункт *Избранное* характерен для ActiveDesktop и позволяют осуществлять доступ к средствам всемирной сети.
4. Пункт *Настройка* позволяет настраивать:

- панель задач, в том числе пункты главного меню;
 - панель управления, доступ к которой также можно получить через папку *Мой Компьютер*;
 - свойства папки *Принтеры и др.*
5. Пункт *Найти* позволяет осуществлять поиск папок и файлов на дисках компьютера, в локальной сети, в Интернет. Поиск можно осуществлять, используя маску файлов, дату создания файла, по типу файла, его размеру, поиск по тексту.
 6. Пункт *Справка* выводит справку по работе с Windows.
 7. Пункт *Выполнить* позволяет запускать программы на исполнение, при этом необходимо указать путь и имя запускаемого файла.
 8. Пункт *Завершение работы* необходим для корректного завершения работы с Windows, перезагрузки компьютера, перезагрузки с выходом в режим MS-DOS.

На рабочем столе размещено несколько графических объектов - ярлыков.

Ярлык – представляет собой высокоэффективное средство ускоренного доступа к объектам. *Ярлык* - это маленький файл-указатель для быстрого доступа к объекту. (файлу, каталогу, диску, программе). Он практически не занимает места на диске (всего 1 Кбайт), а результат щелчка по ярлыку тот же, что и по самому объекту.

Ярлык представлен в виде значка (**пиктограммы**) и названия. Двойной щелчок по ярлыку открывает объект, с которым он связан.

Ярлык обладает еще одним важным свойством: при удалении ярлыка исчезает только он, а сам объект остается без изменений.

Создавать ярлыки можно различными способами. Рассмотрим процесс создания ярлыка с помощью *Мастера создания ярлыков*.

Процесс создания ярлыка напоминает создание папки или файла.

1. На свободном месте рабочего стола щелкнуть правой кнопкой мыши.

2. В контекстном меню выбрать «Создать» → «Ярлык».

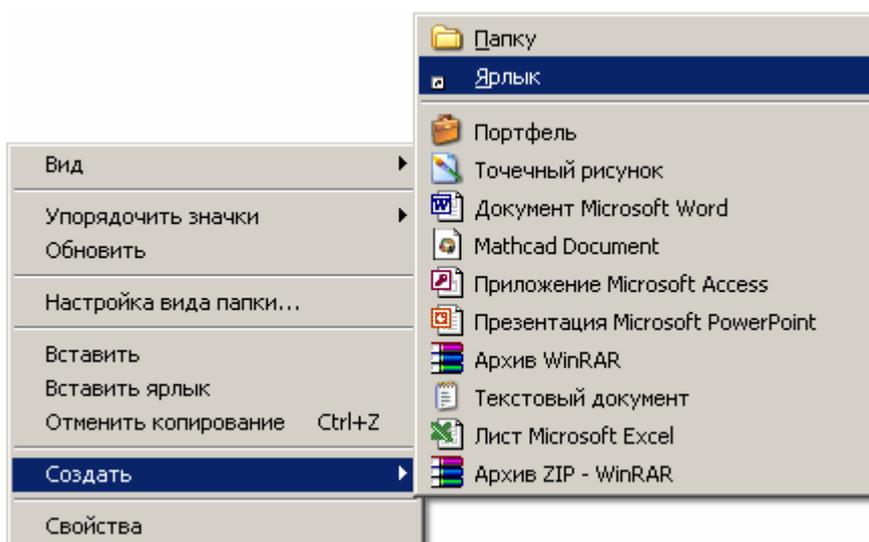


Рис. 1.4. Контекстное меню Рабочего стола «Создать» → «Ярлык»

3. После щелчка по строке «Ярлык» появится окно *Мастера создания ярлыков*, который предложит вам ввести путь к объекту и его имя.

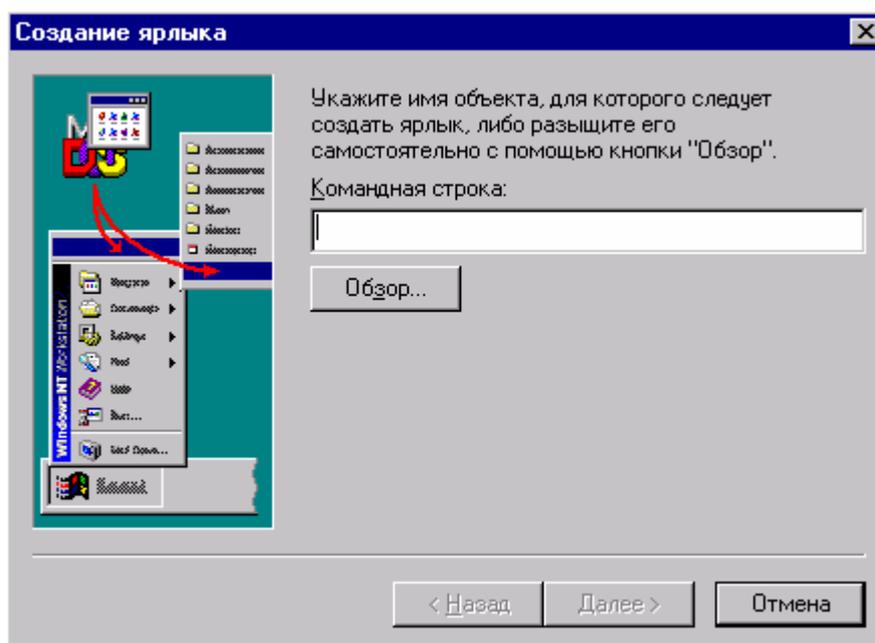


Рис. 1.5. Мастер создания ярлыков. Шаг 1

4. Чтобы не ошибиться в указании пути, необходимо воспользоваться кнопкой «Обзор» и найти нужный файл.

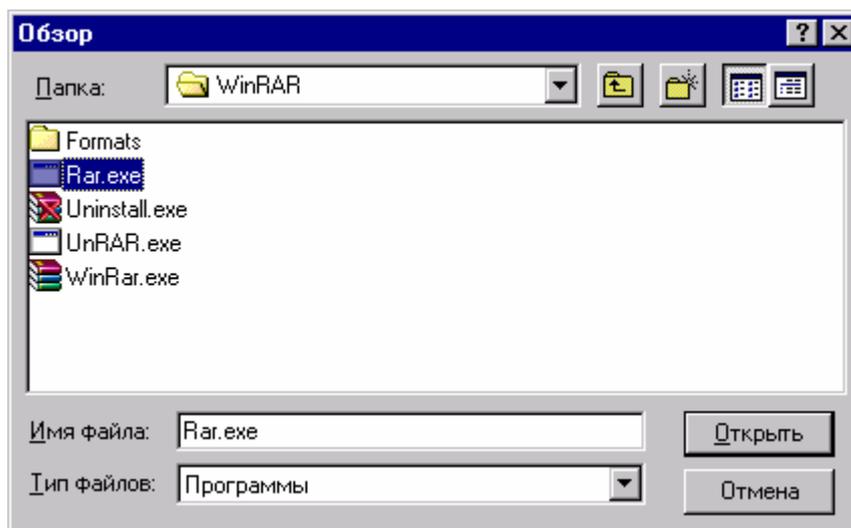


Рис. 1.6. Мастер создания ярлыков. Обзор

5. Как только нужный объект найден, выделяем его и щелкаем по кнопке «Открыть».
6. В окне *Мастера ярлыков* появится имя объекта с указанием полного пути к нему. Щелкаем по кнопке «Далее».

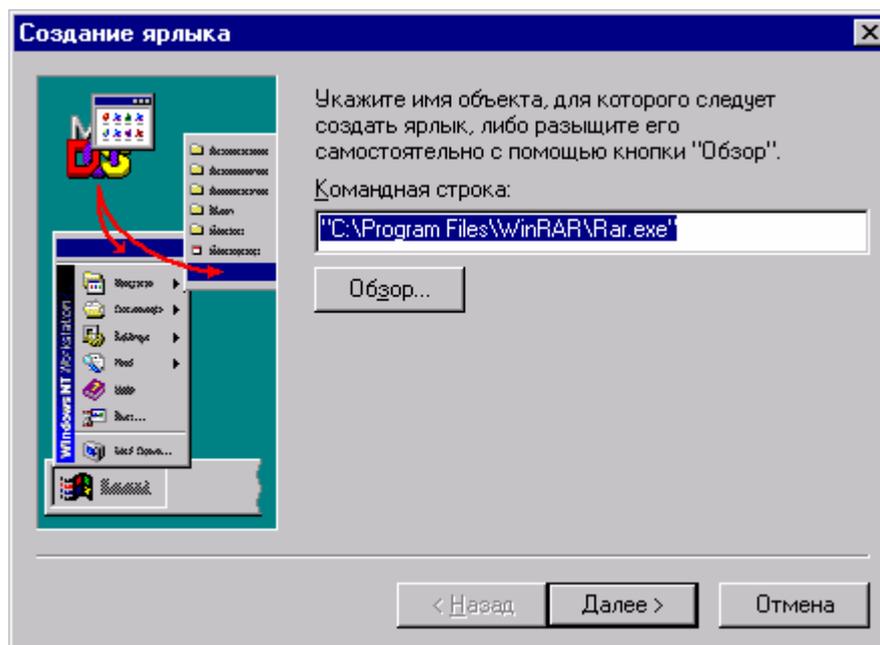


Рис. 1.7. Мастер создания ярлыков. Путь к объекту указан

7. В следующем окне *Мастер* предложит дать имя ярлыку. По умолчанию в соответствующем поле уже находится имя файла, но оно не всегда достаточно информативно, поэтому лучше с помощью клавиатуры ввести простое и понятное имя на русском языке.

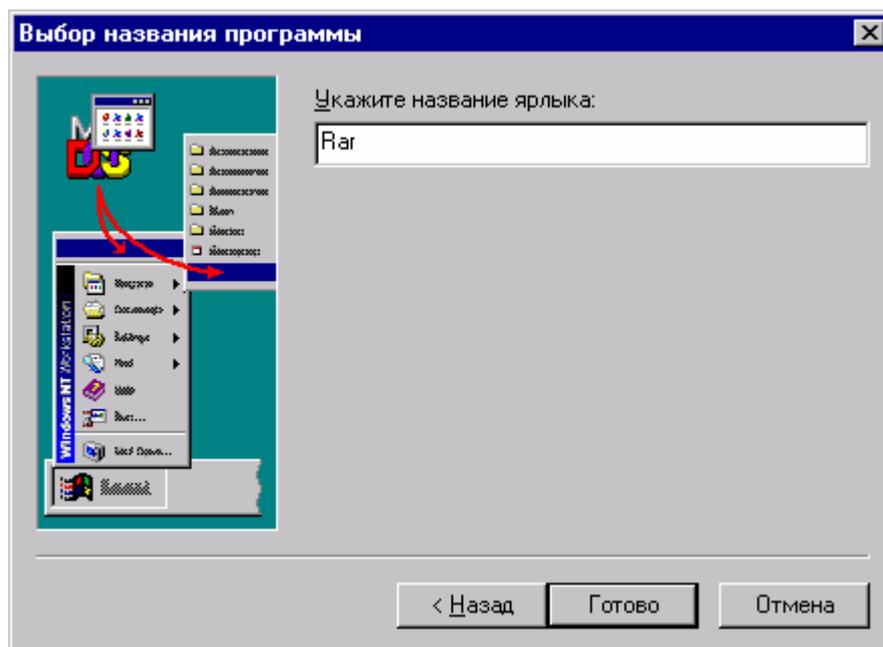


Рис. 1.8. Мастер создания ярлыков. Шаг 2

8. Последнее действие – щелчок по кнопке «Готово». На Рабочем столе появится ярлык объекта.

Ярлыки можно создавать еще одним способом.

1. Откройте папку, в которой находится нужный файл.
2. Выделите файл щелчком мыши.
3. Нажмите на правую кнопку мыши, не отпуская ее, переместите курсор на свободное место Рабочего стола.
4. В появившемся контекстном меню необходимо выбрать «Создать ярлык».

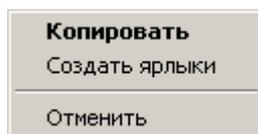


Рис. 1.9. Контекстное меню создания ярлыка

5. При необходимости можно переименовать созданный ярлык.

Рассмотрим часто расположенные ярлыки на Рабочем столе.

Папка Мой компьютер.

Папка *Мой компьютер* отражает содержание всего компьютера целиком (рис. 1.10).

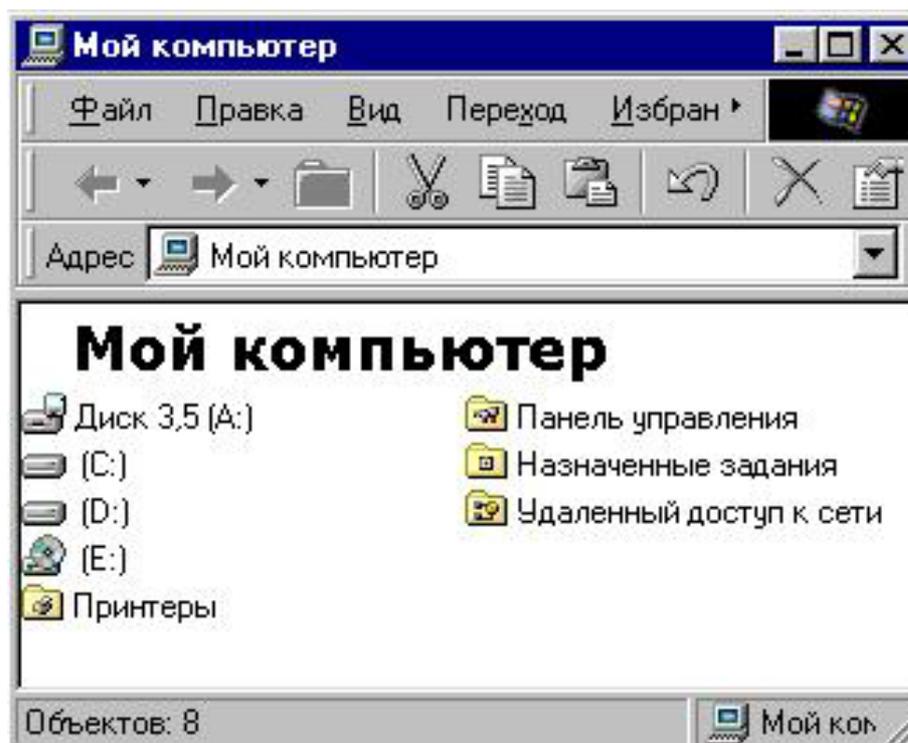


Рис. 1.10. Мой компьютер

Папка позволяет просмотреть содержимое находящихся на компьютере дисков, доступ к панели управления, сетевому окружению.

Утилита **Корзина**.

Корзина - специальная папка, предназначена для временного хранения удаленных файлов, папок, ярлыков (рис. 1.11).

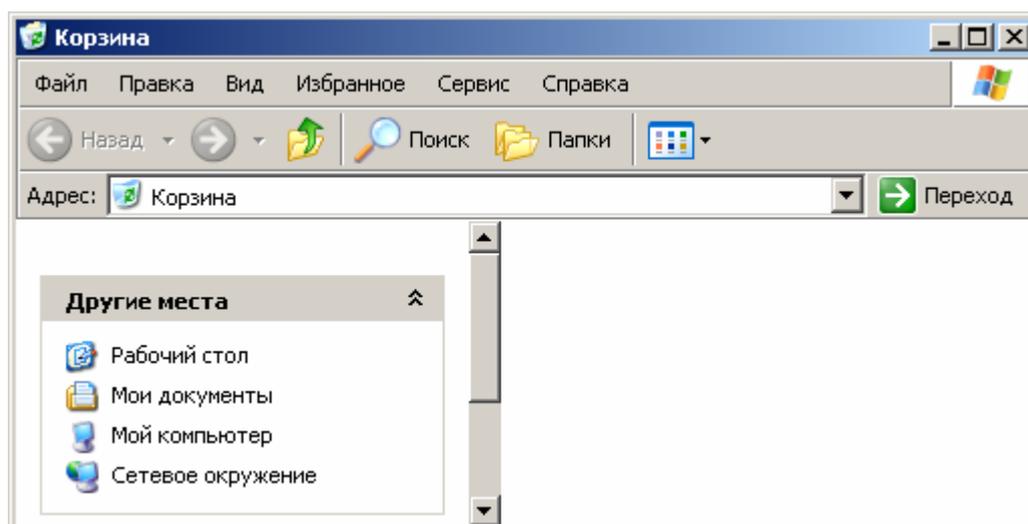


Рис. 1.11. Козина

Она позволяет восстановить объекты, удаленные по ошибке. Размер корзины устанавливает пользователь. По мере работы следует регулярно

очищать корзину, особенно когда возникают проблемы со свободным дисковым пространством.

Для работы с Windows существует **Панель управления**, которая позволяет настроить систему так, как нравится вам, и сделать ее работу более надежной.

Утилита *Панель управления* (рис. 1.12) предназначена для конфигурирования и установки параметров Windows: изменения режима работы ОС и пользовательского интерфейса, установки программного и аппаратного обеспечения, настройки рабочих параметров клавиатуры, мыши, экрана и т.п.

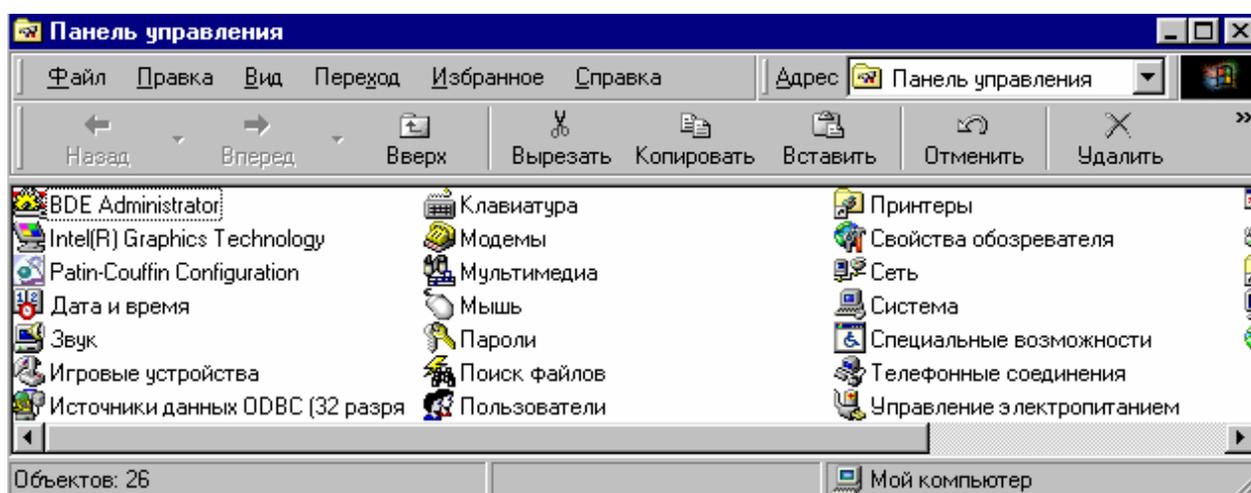


Рис. 1.12. Окно папки *Панель управления*

Окно папки *Панель управления* может содержать несколько десятков значков. Не вносите изменения в Панель управления без особой надобности. Но если вы четко представляете, чего хотите, вы сможете сделать свою работу более удобной и эффективной.

Для просмотра содержимого папок существует специальная программа, которая называется «**Проводник**» (рис. 1.13).

Проводник - программное средство Windows, предназначенное для управления файловой системой, обеспечивающее доступ к локальным и сетевым ресурсам.

Программа «Проводник» открывается с помощью Главного меню или через контекстное меню папок Мой компьютер, Сетевое окружение Корзина или другой любой папки.

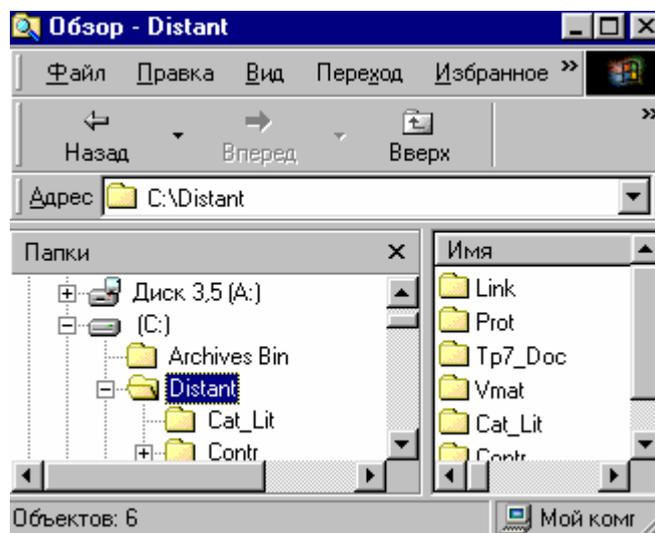


Рис 1.13. Окно Проводника

Окно Проводника делится на две части. Правая часть похожа на окно папки и функционирует аналогично. В левой части выводится структура пространства имен Windows. Здесь показано дерево каталогов локальных жестких дисков и все остальные, подключенные к компьютеру, ресурсы – дисководы, системные папки и сетевые серверы.

Левая область Проводника обладает очень важными свойствами: она существенно облегчает переход между папками, позволяет быстро просмотреть структуру папок и дает возможность перемещать и копировать файлы, перетаскивая их из левой области в правую.

Для перехода между папками следует щелкнуть по ее названию в левой части окна Проводника. Папка, которая была помечена таким образом, станет текущей, и ее содержимое появится в правой области.

Чтобы раскрыть (свернуть) элемент списка левой области, необходимо щелкнуть по значку «+» («-») слева от названия папки. При этом со-

держимое правой области останется прежним, изменится лишь степень детализации структуры в левой области.

Программа Проводник позволяет найти документ в папке, на диске или в сети. Для этого необходимо найти название программы и щелкнуть по нему дважды. После такого щелчка откроется указанная программа и в ней чистый лист нового документа. Закрывать программу Проводник можно как и любое другое окно Windows: с помощью строки меню или кнопками управления размерами окна.

Иногда поиск информации с помощью программы Проводник бывает очень долгим.

Более удобный поиск папок и файлов на компьютере можно осуществить с помощью программы **Поиск**, которая представляет собой средство для поиска любой информации на компьютере (рис. 1.14).

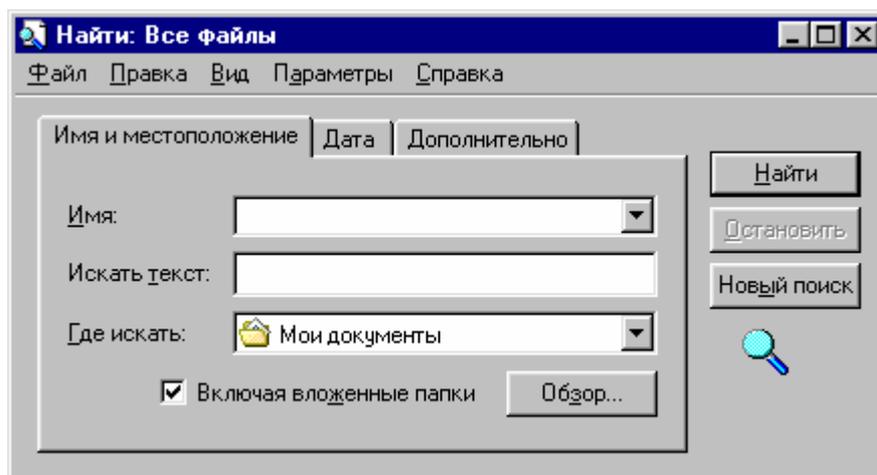


Рис. 1.14. Поиск

Программа запускается из Главного меню и имеет три вкладки:

- *Имя и размещение* – в строке Имя задается имя файла или папки, или же выбирается из списка имеющихся имен;
- *Дата* изменения – указываются временные данные файла;
- *Дополнительно* – указывается объект какой программы будет искаться, строку текста из документа, или размер файла.

Результаты работы программа выводит в специальное окно, из которого найденные программы можно запускать двойным щелчком.

Теперь рассмотрим структуру окна Windows.

Окно - прямоугольная область экрана, в которой может отображаться приложение, документ или сообщение.

Окно может быть представлено в виде: свернутое на панель задач (минимизировано), нормальное (окно с обрамление), полноэкранный экран (распахнутое на весь экран).

Любое окно Windows имеет ряд стандартных элементов (рис. 1.15).

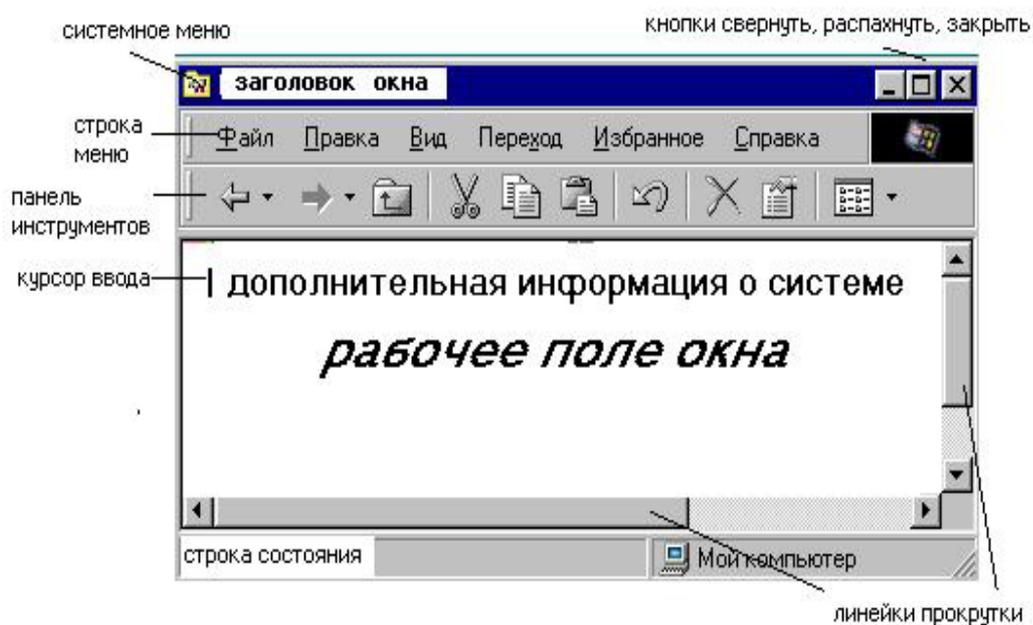


Рис. 1.15. Вид окна

Системное меню содержит команды для изменения размеров окна, его перемещения, минимизации, закрытия.

Заголовок окна является удобным управляющим элементом для перемещения окна. В заголовке окна отображается название приложения и документа (имя открытого файла).

Строка меню содержит пункты для открытия *ниспадающего меню*.

Контекстное меню (всплывающее меню) появляется при нажатии правой кнопки мыши и содержит активные команды для данного объекта.

Рабочее поле окна представляет собой область для размещения окна документа, и которая будет пустой, пока ни один документ не открыт.

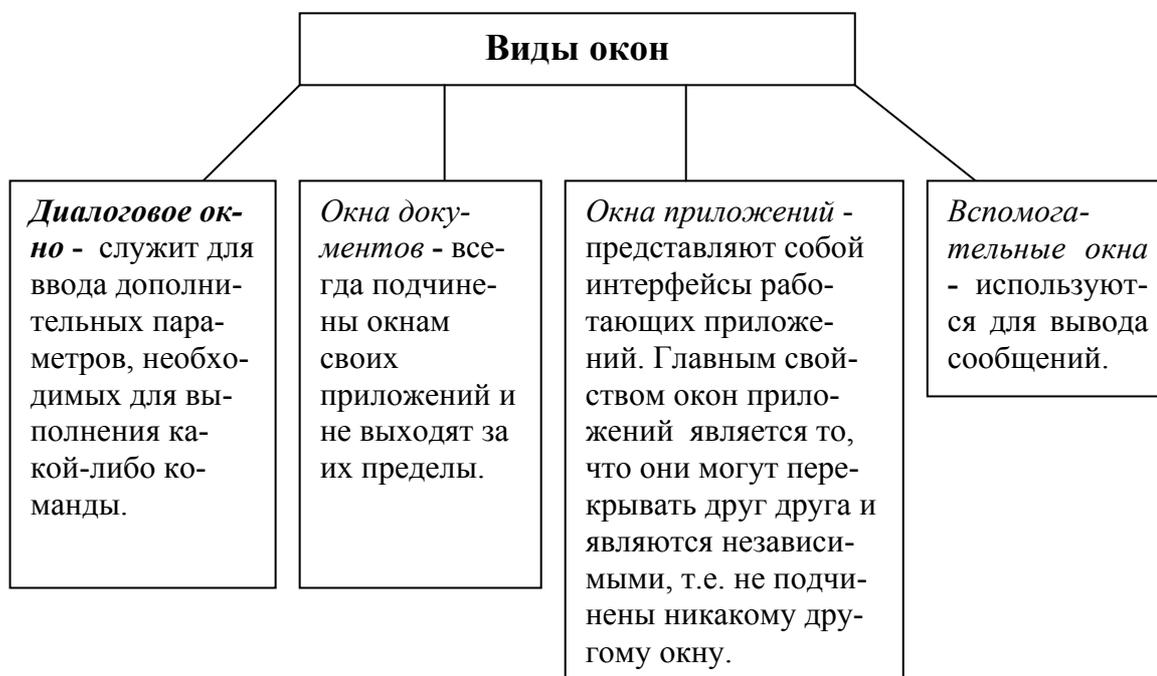
Рамка окна - двойная линия, обрамляющая нормальное окно. Она служит управляющим элементом для изменения размеров окна.

Линейки (полосы) прокрутки окаймляют левую и нижнюю стороны окна и служат для перемещения (прокрутки) документа по вертикали и горизонтали.

Панели инструментов представляют собой линейки командных кнопок, предназначенных для быстрого вызова той или иной команды мышью.

Строка состояния находится у нижнего края окна и содержит информацию о режимах работы приложения.

Активное окно - окно приложения, реагирующее в данный момент на действия пользователя. Заголовок активного окна отличается по цвету и яркости от неактивного.



Как уже говорилось выше, для Windows характерны такие понятия как файл и папка. Разберемся с этими понятиями и рассмотрим приемы работы с ними (создание, копирование, перемещение, удаление).

Файл – это поименованная область на магнитном диске или другом накопителе (магнитной ленте, оптическом или магнитооптическом диске). Полное имя файла обычно состоит из двух частей: имени и расширения. Имя от расширения отделяется точкой. Расширение – это необязательная характеристика, но по нему можно узнать, какая информация хранится в файле.

Тип	Назначение
ARJ	Архивный файл
BAT	Командный файл
COM	Командный системный файл, исполняемый файл
DAT	Файл данных
DOC	Файл документов (текстовый)
EXE	Исполняемый файл
PAS	Программа на языке Паскаль
SYS	Файлы, расширяющие возможности операционной системы
TXT	Текстовый файл

Файл можно создать в любой прикладной программе.

Создание папок.

Папка – это хранилище, в котором могут содержаться компьютеры, диски, очереди заданий на печать, файлы и другие папки. Папки облегчают поиск файлов и позволяют хранить в одном месте все файлы, относящиеся к определенному проекту или теме.

Для создания новой папки можно использовать команду из строки меню *Файл* → *Создать* → *Папка*.

Также для создания новой папки можно выполнить последовательность следующих действий:

1. Открыть ту папку, внутри которой вы создаете новую папку.

2. Щелкнуть правой кнопкой мыши в любой точке свободного места окна папки и выбрать в открывшемся контекстном меню *Создать* → *Папку*.

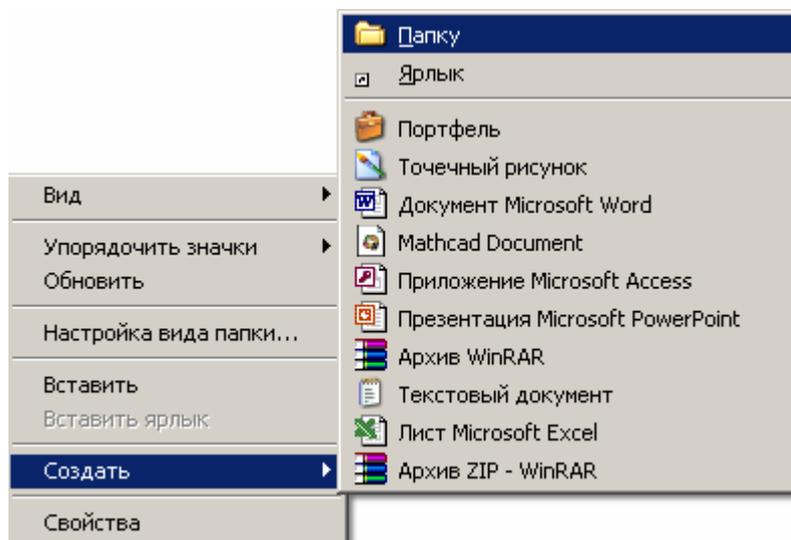


Рис. 1.16. Контекстное меню «Создать» → «Папку»

3. На экране появится папка с именем «Новая папка». Вводится новое имя папки и нажимается клавиша Enter. Если заданное имя совпадает с уже имеющимся именем папки, появится сообщение об ошибке и имя придется изменить.

Копирование и перемещение файлов и папок.

Копирование и перемещение – это основные операции, выполняемые в любой операционной системе. Существует несколько способов для выполнения этих операций.

Метод перетаскивания.

- Переместить файл, группу файлов или папку можно методом перетаскивания значка. Местом назначения может быть открытая папка.
- Папку не обязательно открывать. Перетаскивать значок файла или папки можно на значок закрытой папки.
- Если при перетаскивании файлов или папок использовать правую кнопку мыши. Тогда в конце операции появится контекст-

ное меню со списком возможных действий (в данном случае будет выбираться копирование или перемещение).

Использование буфера обмена.

Если открыть и исходную, и конечную папку невозможно или неудобно, для копирования и перемещения файлов и папок можно использовать буфер обмена.

- Выбрав файлы или папки в исходной папке, необходимо выбрать в строке меню Правка → Копировать для копирования или Правка → Вырезать для перемещения.
- Открыв конечную папку, необходимо выбрать в строке меню Правка → Вставить.

Если при копировании или перемещении выясняется, что файл или папка с таким именем в месте назначения уже существует, выдается запрос на подтверждение замены файла или папки.

Переименование папок, файлов, ярлыков.

1 способ

Щелкните по значку объекта правой кнопкой мыши.

Выберите в меню объекта команду Переименовать.

Введите новое имя.

2 способ

1. Выделите объект.

2. Выберите Файл → Переименовать.

3 способ

Выделите объект, щелкните по его названию - вокруг него появится прямоугольная рамка, в которой можно будет ввести новое имя или отредактировать текущее.

4 способ

Выделите объект.

Нажмите F2.

Введите новое имя или отредактируйте текущее.

Удаление файлов и папок.

- Для удаления файла или папки необходимо сначала выбрать удаляемый объект.
- Удалить выбранный файл или папку можно с помощью клавиатуры, нажав клавишу Delete.
- Файл или папку можно удалить перетащив значок файла или папки на значок корзины на Рабочем столе.

Восстановление объекта с помощью Корзины

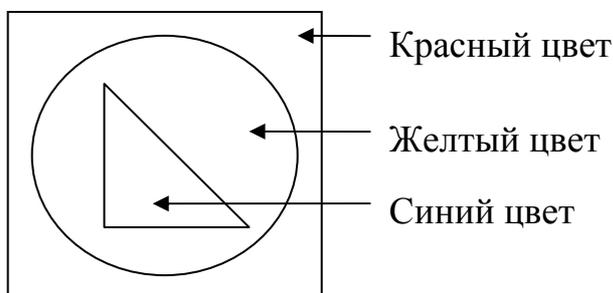
Для того, чтобы восстановить объект, попавший в Корзину, отметьте его и выберите команду *Файл → Восстановить* (или щелкните по объекту правой кнопкой мыши и выполните команду *Восстановить* из меню объекта). Эта команда перенесет выбранный объект в ту папку, из которой он был удален. Если такой папки уже нет, Windows попросит разрешения создать ее заново.

При восстановлении объекта вы можете поместить его и в другую папку. Для этого выделите нужный объект и выберите в меню объекта или в меню *Файл* команду *Вырезать*. После этого перейдите к той папке, в которую вы намерены поместить восстанавливаемый объект, и выполните *Вставить* из меню *Правка* окна папки-приемника.

Задания:

1. На диске C: создать папки Эксперимент1, Эксперимент2, Эксперимент3.
2. Открыть папку Эксперимент1 и создать с ней файл «Записки» (программа «Блокнот»), содержащий следующий текст:
«Информатика – это наука, изучающая все аспекты получения, хранения, переработки, передачи и использования информации.»

3. Открыть папку Эксперимент2 и создать с ней файл «Сочинение» (программа «Текстовый редактор WordPad »), содержащий следующий текст: «Основная задача информатики заключается в определении общих закономерностей, в соответствии, с которыми происходит создание научной информации, ее преобразование, передача и использование в различных сферах деятельности человека».
4. Открыть папку Эксперимент3 и создать с ней файл «Рисунок» (программа «Графический редактор «Paint»»), содержащий следующий рисунок:



5. Скопировать папку Эксперимент3 в папку Эксперимент1.
6. Перенести папку Эксперимент2 в Эксперимент1.
7. Открыть программу Проводник.
8. Открыть в левой области Проводника папки Эксперимент1 и Эксперимент2 и посмотреть их содержимое в правой области.
9. Поменять вид содержимого правой панели Проводника.
10. Создать на Рабочем столе ярлык для папки Эксперимент1 с помощью мастера создания ярлыков.
11. Создать в папке Эксперимент1 ярлык для папки Эксперимент2 методом перетаскивания.
12. Удалить папку Эксперимент3 в Корзину.
13. Восстановить папку Эксперимент3.
14. С помощью программы Поиск найти папку Эксперимент3 и убедиться, что она восстановлена в исходное место.

15. Найти файлы, содержащие следующий текст: «Информатика».
16. Включить заставку:
 - МЕТАМАРФОЗЫ, параметры: пружина, клетка, время – 1 минута.
 - ОБЪЕМНЫЙ ТЕКСТ, текст «ИНФОРМАТИКА», параметры: заливка рисунком «Волны», стиль движения «Волны», шрифт Courier.
17. Просмотрите предлагаемые рисунки и узоры для Рабочего стола. Установите узор «Пузырьки», «Колечки».
18. Показать выполненную работу преподавателю.
19. Удалить созданные папки и ярлыки.

Вопросы для самоконтроля:

1. Что такое ярлык, документ, папка?
2. Укажите назначение папок Мой компьютер, Корзина, программы Проводник.
3. Расскажите о Рабочем столе.
4. Что представляет собой панель задач?
5. Переселите пункты главного системного меню, их назначение.
6. Как вызвать на экран главное меню? Для чего оно предназначено?
7. Что такое окно? Перечислите основные элементы окна.
8. Какие виды окон вы знаете?
9. Перечислите все виды меню в Windows.
10. Как создать папку?
11. Как создать ярлык?
12. Как скопировать или переместить объект?
13. Как переименовать объект?
14. Как удалить объект и восстановить его?

Лабораторная работа №2

Тема: Текстовый редактор Word.

Цель работы: овладение практическими навыками работы с текстовым процессором Microsoft Word.

Краткая теория

Основные функции *текстовых редакторов* заключаются в автоматизации ввода и редактирования текстовых данных.

В отличие от текстовых редакторов, *текстовые процессоры* позволяют не только вводить и редактировать текст, но и форматировать его, т.е. оформлять. Соответственно, к основным средствам текстовых процессоров относятся средства обеспечения взаимодействия текста, графики, таблиц и других объектов.

Рассмотрим один из текстовых процессоров – **Ms Word**, который является приложением Windows.

Структура окна Word типична для приложений Windows. В нее входят элементы, необходимые для редактирования и форматирования текстов.

После установки программы в ней по умолчанию присутствуют две панели инструментов – Стандартная и Форматирование. Если в ходе работы возникает необходимость в других панелях инструментов, их можно открыть и расположить вдоль любой границы окна или отдельно.

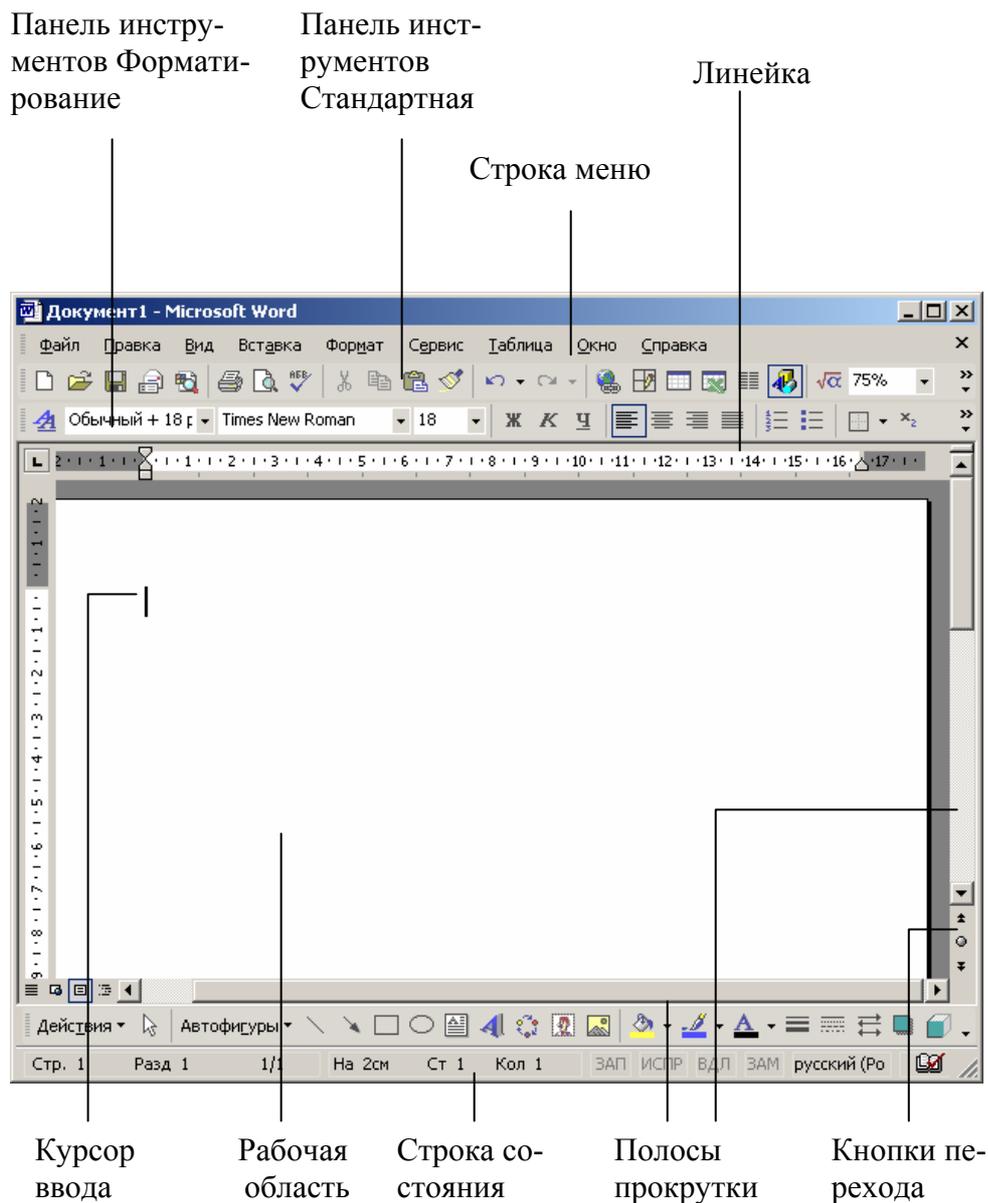


Рис. 2.1. Основные элементы окна

Основные приемы работы в Word:

1. Вставка формул.

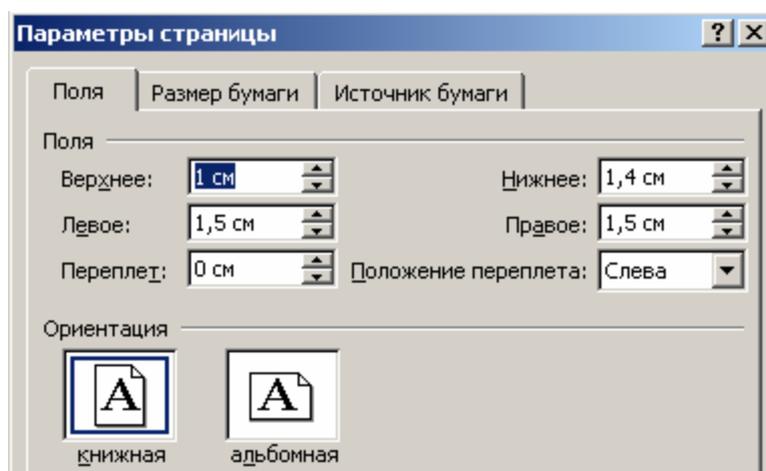
Для вставки различных формул необходимо вызвать редактор формул, нажав на панели инструментов кнопку . С помощью появившегося редактора можно вставлять различные формулы.



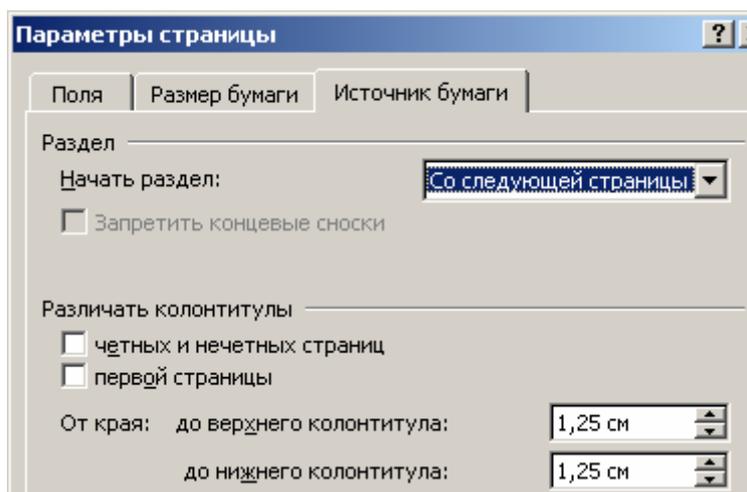
Рис2.2. Редактор формул

2. Форматирование страниц.

При форматировании страниц задают отступы и ориентацию бумаги. Для этого необходимо выбрать *Файл* → *Параметры страницы*. В появившемся окне *Параметры страницы* задаются поля (отступы) страницы и ориентация (книжная или альбомная).



Для введения нумерации страниц в создаваемом документе необходимо выбрать *Вставка* → *Номера страниц*. В появившемся окне *Номера страниц* задается формат номеров страниц. Номера страниц проставляются в колонтитулах.



Колонтитулы представляют собой одну или несколько строк, помещаемых в начале или конце каждой страницы документа. Для расстановки колонтитулов необходимо выбрать *Вид* → *Колонтитулы*.

Можно установить колонтитулы для первой страницы, а так же для четных и нечетных страниц. Для этого необходимо выбрать *Файл* → *Параметры страницы*. В появившемся окне выбрать вкладку *Источник бумаги* и задать расположение колонтитулов.

3. Форматирование шрифта.

Для форматирования шрифта необходимо выбрать *Формат* → *Шрифт*.

В появившемся окне *Шрифт* задаются: размер (14 пт – деловой документ, 12 пт – документ Internet, 8 пт - газе-

та); название; интервал между буквами (уплотненный, обычный, разреженный); цвет; параметры (жирный, курсив, подчеркнутый и др.); анимация для вводимого текста.

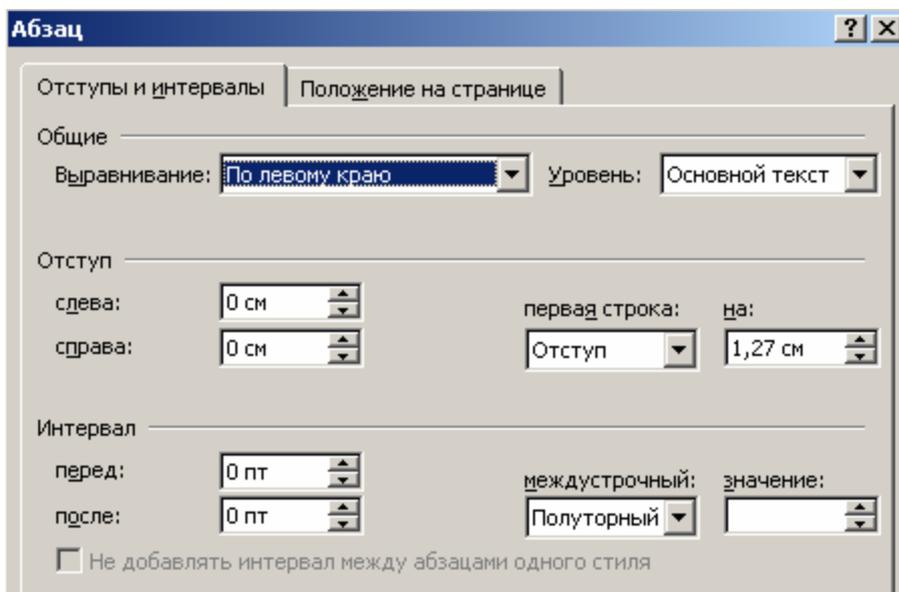
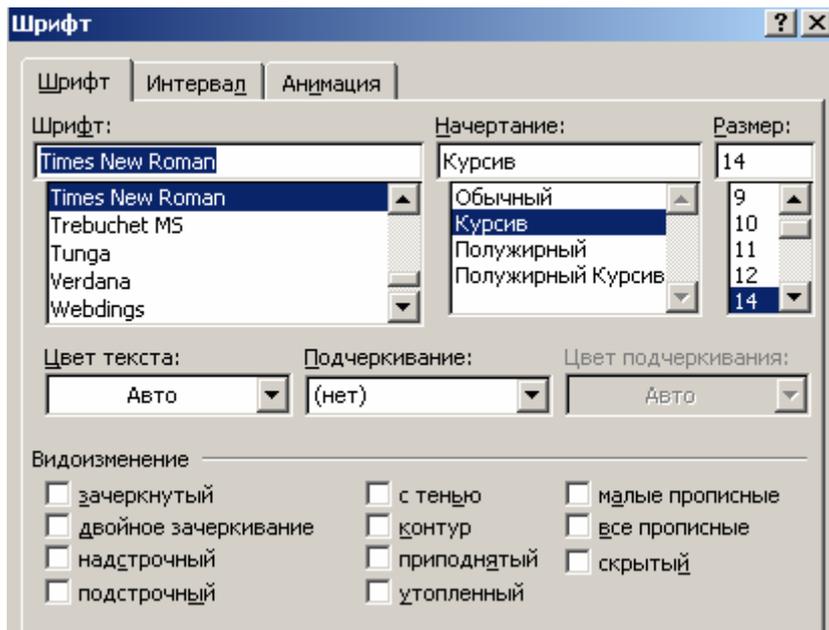
4. Форматирование абзаца.

Для форматирования шрифта необходимо выбрать *Формат* → *Абзац*.

В появившемся окне *Абзац* задаются отступы слева, справа; отступ

для первой строки; межстрочный интервал; выравнивание текста (по ширине, по центру, по краям).

5. Вставка объекта WordArt.



Для вставки объекта WordArt необходимо нажать кнопку . Откроется коллекция WordArt.

Задания:

1. Вставить номера страниц: внизу страницы, от центра.
2. Вставить колонтитулы: для первой страницы - «Лабораторная работа по теме «Microsoft Word», для нечетной страницы – «Информатика», для четной страницы – «Microsoft Word».

Каждое следующее задание выполнять на отдельной странице.

3. Напечатайте приведенный ниже текст, выполняя задание:

Математическое моделирование (численное исследование)

Существует два класса методов моделирования – *физическое и математическое*. В работе рассматривается математическое моделирование. Методы математического моделирования распадаются на **три**

большие группы:

- I) *аналитические;*
- II) *численные;*
- III) *комплексные;*

Методы разных групп различаются типами, способами и сложностью математического аппарата, используемого для решения дифференциальных уравнений в частных производных (**ДУЧП**), описывающих физический процесс.

С появлением электронных вычислительных машин широко доступным стало непосредственное применение разнообразных численных методов для решения ДУЧП с целью правильного и точного моделирования задач механики и гидравлики.

Задание:

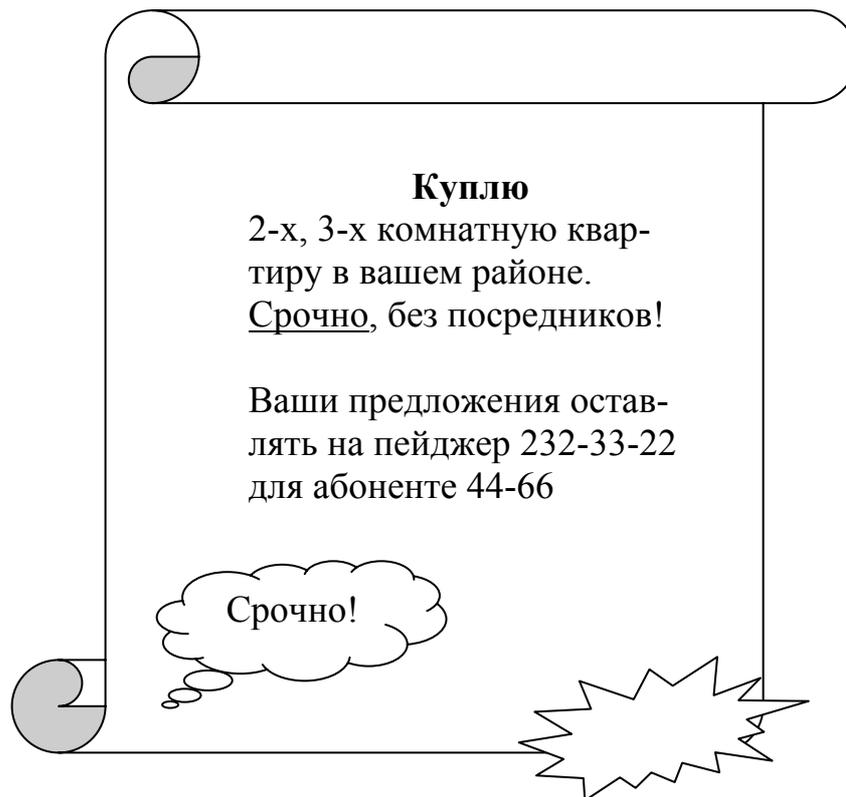
– Заголовок выровнен по центру, шрифт №16, полужирный, подчеркнутый.

- Основной текст – шрифт №14.
 - Первый абзац выровнен по правому краю, второй – по левому краю, третий – по ширине.
 - Нумерованный список – шрифт №12, курсив.
4. Создайте следующую таблицу, выполните задание.

Основная заработная плата					
№ п/п	Наименование этапа	Исполнитель	Зарплата за день, руб.	Трудоемкость	Сумма, руб.
1	Формулировка проблемы и постановка задачи	Руководитель	1450	2	2900
2	Внедрение	Программист	250		500
3	Разработка технического задания	Начальник отдела	830	3	2490
4	Подбор литературы	Программист	250	5	1250
5	Рабочее проектирование			27	6750
6	Отладка и тестирование		250	32	8000
7	Создание технической документации			4	1000
Итого:				75	22890

Задание:

- Текст таблицы – шрифт №14.
 - Заливка последней строки таблицы – 20%.
 - Заливка первого столбца – голубой.
 - Заливка ячейки с заголовком – желтый.
 - Обратить внимание на оформление таблицы.
5. Создать следующее объявление и оформить его соответствующим образом:



6. Напечатайте приведенный ниже текст. Расположите его сначала в две, а затем в четыре колонки.

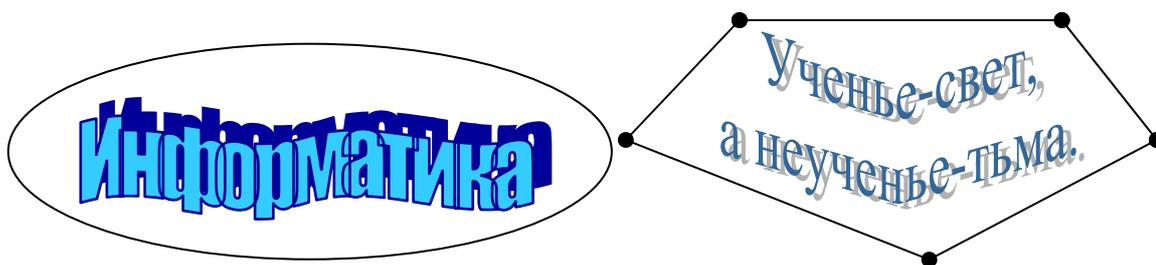
«Под сетью (net) принято понимать совокупность множества узлов (nodes), объединенных между собой нитями (линиями связи, lines). Узлами компьютерной сети являются мощные вычислительные машины, принадлежащие, как правило, государственным организациям, научным и исследовательским центрам, а также крупным коммерческим фирмам. Узлы сети соединятся нитями – оптоволоконными, спутниковыми, радиорелейными или телефонными линиями связи».

7. Создать следующие формулы:

$$\frac{\partial z}{\partial x} = \frac{3 \cdot \frac{1}{3\sqrt[3]{x^2}}}{(3\sqrt[3]{x} + 2y^2)\ln 2} = \frac{1}{(3x - 2y^2\sqrt[3]{x})\ln 2};$$

$$\lim_{x \rightarrow 0} \frac{abmx^{m-1}}{(1+ax^m)nx^{n-1}} = \frac{abm}{n} \lim_{x \rightarrow 0} x^{m-n} = \begin{cases} 0, & m > n \\ ab, & m = n \\ \infty, & m < n. \end{cases}$$

8. Создать следующие надписи, используя объекты рисования и WordArt.



9. Создайте следующий многоуровневый список, соблюдая все элементы форматирования.

Интернет

Вместо введения

Глава 1. Подключение к Сети

- A. Сеть – первое знакомство
- B. Подключение к Internet
- C. Настройка аппаратного и программного обеспечения
 - C.a) Требования к персональному компьютеру
 - C.b) Выбор модема
 - C.c) Подключение модема к компьютеру

Глава 2. Работа в Сети

- A. Электронная почта
- B. Почтовые серверы – файлы почтой
- C. Работа с FTR
- D. World Wide Web – паутина, которая соединяет

Глава 3. Поиск информации в Сети

- A. Поисковые системы
 - A.a) ARCHIE – записная книжка FTR – архивов
 - A.b) VERONICA – подруга Archie
 - A.c) Интегрированные поисковые системы
- B. Тематический поиск информации

Словарь терминов Internet

Предметный указатель

Приложение I

Оглавление

10. Сохранить файл с именем word.doc.
11. Показать выполненную работу преподавателю.
12. Удалить файл word.doc.

Лабораторная работа №3

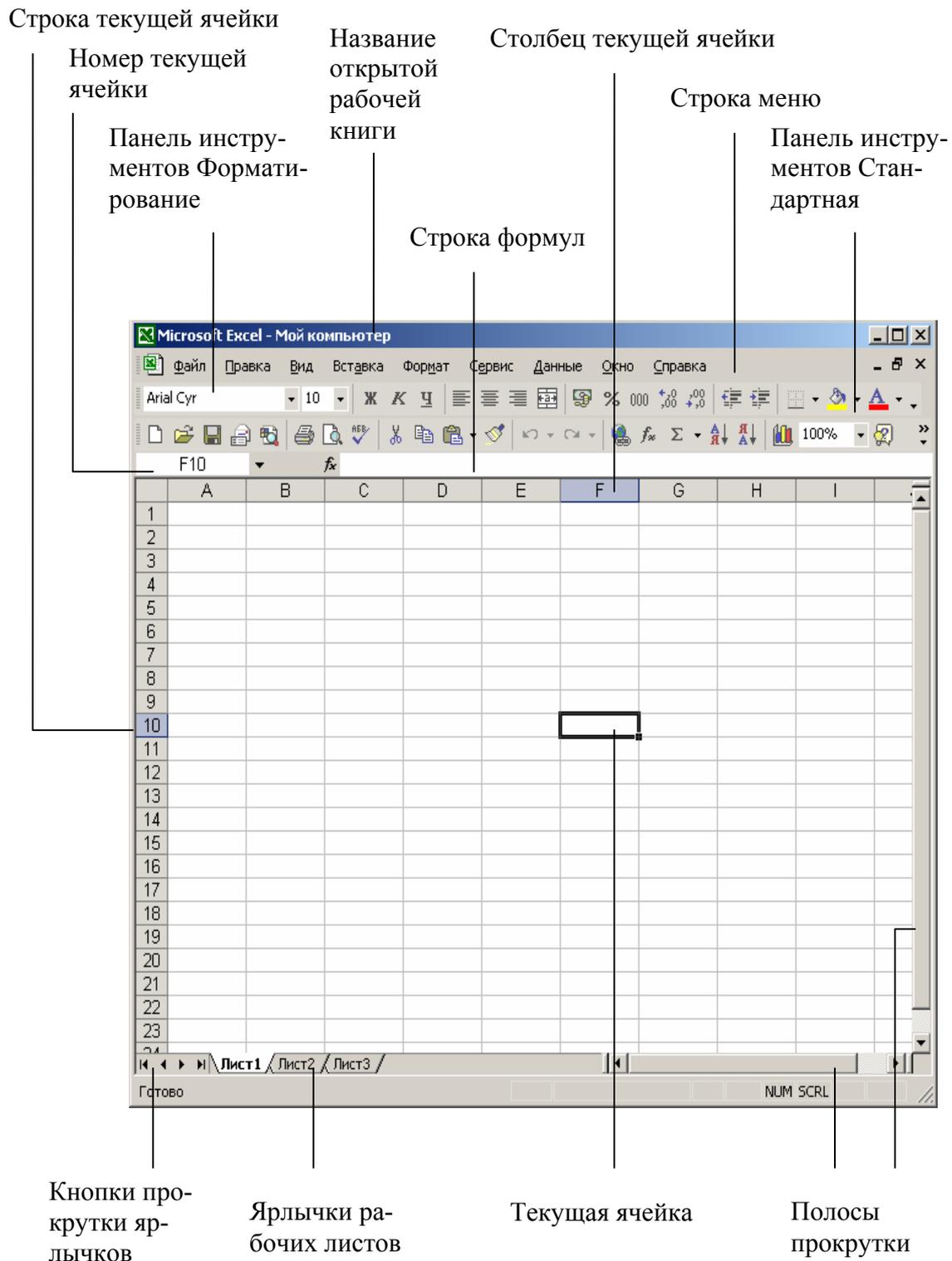
Тема: Табличный процессор Excel .

Цель работы: овладение практическими навыками работы с табличным процессором Microsoft Excel.

Краткая теория

Электронная таблица (ЭТ) – компьютерный эквивалент обычной таблицы, в клетках (ячейках) которой записаны данные различных типов: тексты, даты, формулы, числа.

Для управления ЭТ используется специальный комплекс программ – *табличный процессор*.



Рабочая область ЭТ состоит из *строк* и *столбцов*, имеющих свои имена. Имена строк – это их номера. Нумерация строк начинается с 1 и заканчивается максимальным числом, установленным для данной програм-

мы. Имена столбцов – это буквы латинского алфавита сначала от А до Z, затем от AA до AZ, BA до BZ и т. д.

Пересечение строки и столбца образует *ячейку таблицы*, имеющую свой уникальный адрес. Для указания адресов ячеек в формулах используются *ссылки* (например, A2 или C4).

В ЭТ существует понятие *блока* (диапазона) *ячеек*. *Блок ячеек* – группа последовательных ячеек, может состоять из одной ячейки, строки (или ее части), столбца (или его части), а также последовательности строк или столбцов (или их частей). Адрес блока ячеек задается указанием ссылок первой и последней его ячеек, между которыми ставится разделительный символ – двоеточие <:> или две точки подряд <..>.

Пример. G3, A1:H1, B1:B10, D4:F5.

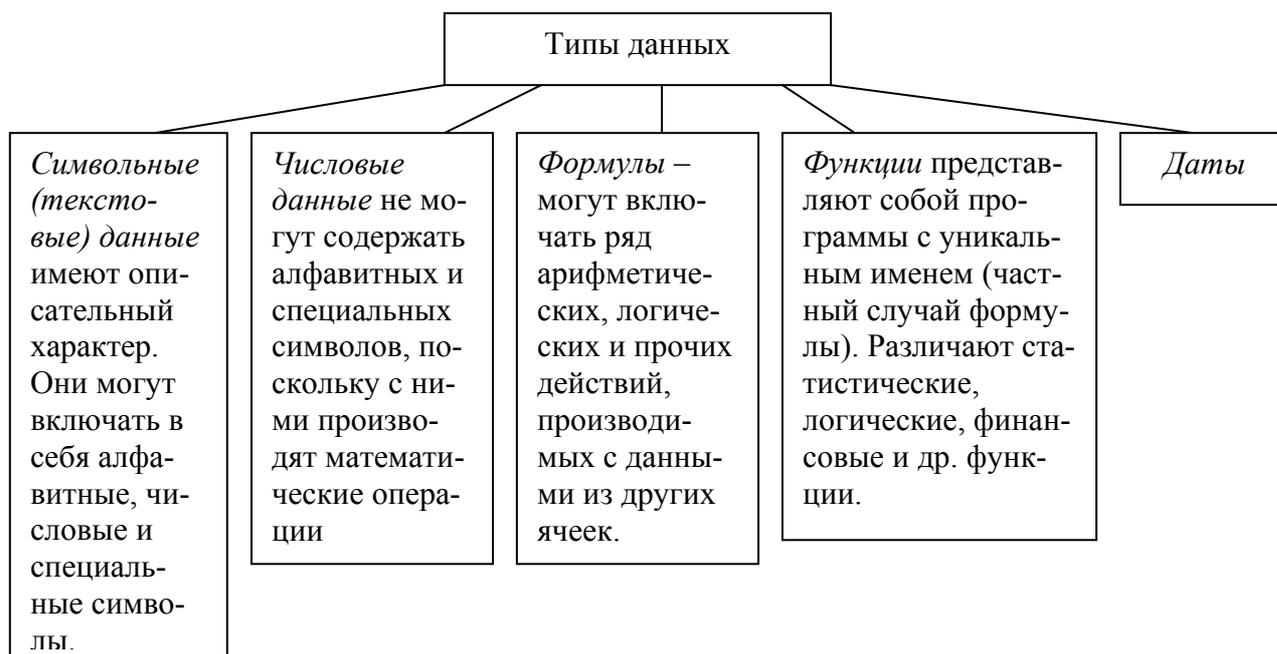
Каждая команда ЭТ требует указания блока (диапазона) ячеек, в отношении которых она должна быть выполнена.

Блок используемых ячеек может быть указан двумя путями: либо *непосредственным набором* с клавиатуры начального и конечного адресов ячеек, формирующих диапазон, либо *выделением* соответствующей части таблицы при помощи клавиш управления курсором. Удобнее задавать диапазон выделением ячеек.

Текущей (активной) называется *ячейка* ЭТ, в которой в данный момент находится курсор. Адрес и содержимое текущей ячейки выводятся в строке ввода ЭТ.

Каждый документ представляет собой набор таблиц – *рабочую книгу*, которая состоит из одного или многих *рабочих листов*.

В каждую ячейку ЭТ пользователь может ввести данные одного из следующих типов:



В рамках одной и той же ЭТ можно использовать различные форматы представления числовых данных.

Программа Excel рассматривает содержимое ячейки как формулу, если оно начинается со знака равенства (=).

В ЭТ могут быть представлены следующие виды функций: математические, статистические, текстовые, логические, финансовые и др.

При копировании или перемещении формулы в другое место таблицы необходимо организовать управление формированием адресов исходных данных. Поэтому в ЭТ при написании формул используются понятия относительной и абсолютной ссылок.

Абсолютная ссылка – это не изменяющийся при копировании и перемещении формулы адрес ячейки, содержащий исходное данные.

Для указания абсолютной адресации вводится символ \$. Различают два типа абсолютной ссылки: полная и частичная.

Полная абсолютная ссылка указывается, если при копировании или перемещении адрес клетки, содержащий исходное данные, не меняется.

Для этого символ \$ ставится перед наименованием столбца и номером строки.

Полная абсолютная ссылка указывается, если при копировании или перемещении адрес клетки, содержащий исходное данное, не меняется.

Для этого символ \$ ставится перед наименованием столбца и номером строки.

Пример. \$B\$5; \$D\$12

Частичная абсолютная ссылка указывается, если при копировании или перемещении не меняется номер строки или наименование столбца.

При этом символ \$ ставится перед номером строки, а во втором случае – перед наименованием столбца.

Пример. B\$5; D\$12; \$B5; \$D12

При абсолютной адресации копируемая формула не изменяется.

Относительная ссылка – это изменяющийся при копировании и перемещении формулы адрес ячейки, содержащий исходное данное. Изменение адреса происходит по *правилу относительной ориентации* клетки с исходной формулой и клеток с данными.

Excel располагает рядом команд для построения различных типов диаграмм: круговая диаграмма, линейный график и т. д.

Пример 7 Даны три числа. Если произведение первого и второго больше суммы всех чисел, то вывести натуральный логарифм первого числа, иначе экспоненту произведения этих чисел.

	A	B	C	D	E	F	G	H	I	J	K
1	1	2	3								
2											
3	Ответ:	403.43									
4											

Решение:

В ячейке В3 находится формула для вычисления:

=ЕСЛИ(ПРОИЗВЕД(А1;В1)>СУММ(А1:С1);LN(ABS(А1));EXP(ПРОИЗВЕД(А1:С1)))

Пример 8 Даны два числа. Если оба числа положительны, то вывести логарифм второго по основанию первого. Если оба отрицательны, то вывести десятичный логарифм их произведения, иначе (одно положительное, другое - отрицательное) вывести сумму всех чисел.

Решение:

Составим формулу для вычисления с использованием стандартных математических функций:

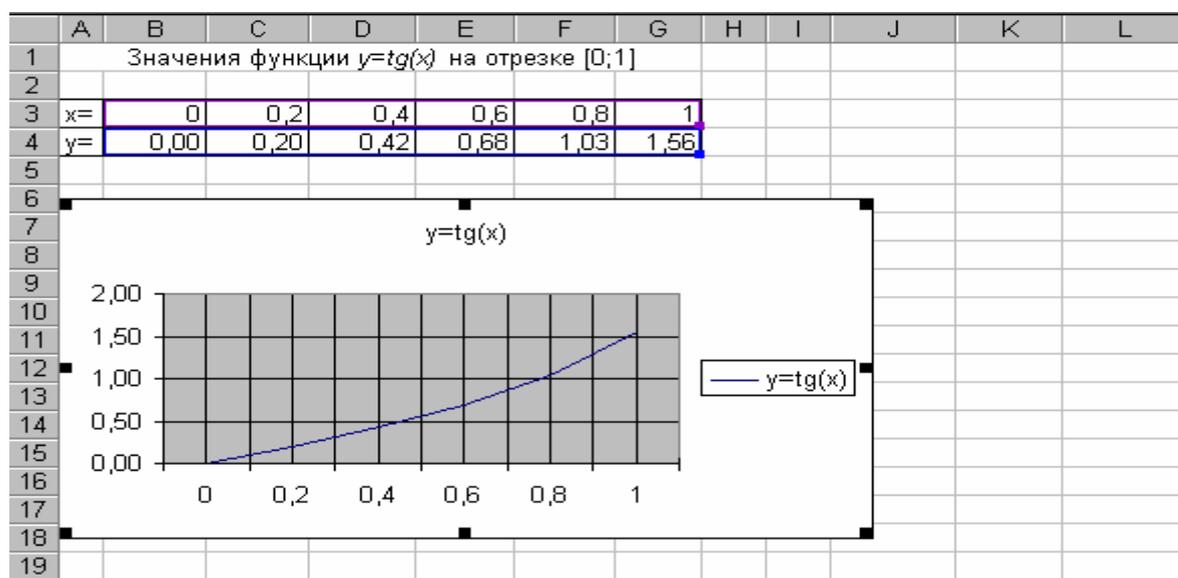
=ЕСЛИ(И(А1>0;В1>0);LOG(В1;А1);ЕСЛИ(И(А1<0;В1<0);LOG10(ПРОИЗВЕД(А1;В1));СУММ(А1;В1)))

Пример 6 Построить график функции $y=tg(x)$ на отрезке $[0;1]$ с шагом $h=0.2$

Решение: Рассчитаем значения аргумента:

- 1) В ячейку В3 вводим начальное значение;
- 2) В ячейку С3 вводим формулу: =В3+0,2;
- 3) Составляем таблицу значений.

Рассчитаем значения функции: в ячейку В4 вводим формулу: =TAN(В3).



По полученным данным строим график функции, используя Мастер функций.

Задания:

1. Подготовить лист, с помощью которого можно, задавая показания счетчика электроэнергии, определить её расход и сумму оплаты:

	Тариф		Коп./кВт.ч	
Месяц	Дата	Показание счетчика	Расход, кВт.ч	Сумма, руб.
Декабрь	27.12.01	2673		
Январь	30.01.02	1256		
Февраль	29.02.03	4528		
...				

Стоимость 1 кВт.ч электроэнергии задается в ячейке D1.

2. Два электрических сопротивления в схеме соединены последовательно. Определить общее сопротивление схемы для всех возможных сочетаний значений сопротивлений: первого – 1, 2, 3, ..., 10 ом, второго – 4, 4,5, 5, ..., 8 ом.
3. На листе записаны сведения о плотности различных материалов:

1	Материал	Плотность, кг/куб.дм
2	Алюминий	2,5
3	Бетон	2,4
4	Свинец	11,4
5	Сталь	7,85
6	Стекло	2,6
7	Чугун	7,13

Подготовить лист для определения названия материала, плотность которого является минимальной из всех плотностей, которые не меньше P кг/куб.дм (значение P будет указываться в отдельной ячейке).

4. На листе рассчитаны значения функции $y=x^2$ для нескольких значений x :

x	У
1	1
-2	4
1,5	2,25
-0,5	0,25
2	4
1	1
-1,5	2,25
0	0
0,5	0,25

Подготовьте приведенные данные так, чтобы по ним можно было построить график указанной функции.

5. Известны данные о мощности двигателя 30 моделей легковых автомобилей. Подготовить лист для того, чтобы выяснить, есть ли среди них модель, мощность двигателя которой превышает 200л.с.
6. Запишите на листе электронной таблицы даты рождения каждого студента вашей группы (В формате *дата*):

№	Фамилия, Имя	Дата рождения
1	Арутюнов А.	12.04.87
2		

Для каждого ученика определить число прожитых дней и записать найденные значения в столбце D. Дату текущего дня указать в отдельной ячейке.

7. В ячейках A2:A15 записаны числа. В ячейках B2:B15 получить числа, найденные путем увеличения этих чисел на величину, значение которой вводится в ячейку B16:

Исходные числа	Полученные числа
13	
4	
...	
-30	
Величина увеличения:	

8. Получить таблицу перевода 1,2,3,...,20 долларов США в рубли по текущему курсу. (Значение курса указывается в отдельной ячейке).
9. Вычислить x^2, x^3, \dots, x^{15} для значения x , указываемого в отдельной ячейке.
10. Составить таблицу умножения на число n (значение n указывается в ячейке C1; $1 < n < 9$).
11. Назовите лист 1 «Задание №1». Заполните таблицу и вычислите максимальное, минимальное и среднее количество студентов по отдельным факультетам:

Факультет	1 курс	2 курс	3 курс	4 курс	5 курс	Максимальное количество	Минимальное количество	Средний балл
ФПИ	75	68	68	62	56			
ФМО	60	58	55	57	49			
ФМИ	100	91	92	86	69			
ЭкФ	150	142	140	141	135			
ЭФ	120	111	105	100	99			

Постройте следующие диаграммы, отображающие:

- Сравнение количества студентов на разных курсах по факультету ЭФ (круговая или кольцевая). Разместить на текущем листе. Диаграмма должна содержать заголовок, подписи данных.

- Сравнение среднего количества студентов на всех факультетах (гистограмма или линейчатая). Разместить на текущем листе.

Диаграмма должна содержать заголовок, подписи данных.

12. В приведенной ниже таблице сумма рассчитывается в зависимости от величины купленной партии товаров. Если куплено больше 20 ед., партия считается оптовой и покупателю предоставляется скидка 15%. Выведите данные о продаже со скидкой.

Наименование товара	Количество, ед.	Цена, руб	Сумма, руб (без скидки)	Сумма, руб (со скидкой)
Телевизор	56	6000		
Принтер	25	7000		
Сканер	15	3000		
Монитор	8	7800		
Минимальная партия для оптовиков, ед			20	
Скидка для оптовиков			15%	

13. Даны три произвольных числа. Если модуль разности этих чисел превышает 20, то вывести среднее арифметическое этих чисел, иначе – их произведение.
14. Даны три числа. Если сумма этих чисел превышает их произведение, то вывести абсолютное значение разности этих чисел, в противном случае – корень квадратный из их произведения.
15. Даны три числа. Если модуль их суммы меньше либо равен их среднему арифметическому, то вывести среднее геометрическое этих чисел, иначе – сумму квадратов этих чисел.

Вопросы для самоконтроля:

1. Дать определение ЭТ.

2. Перечислить и пояснить основные типы данных, которые могут быть введены в ячейки ЭТ.
3. Как указывается блок (диапазон) ячеек при выполнении какой-либо команды?
4. Пояснить, для чего используются абсолютные и относительные адреса ячеек.

Лабораторная работа №4

Тема: Структура программы. Линейное программирование.

Цель работы: овладение практическими навыками разработки и программирования вычислительного процесса линейной структуры и навыками по отладке и тестированию программы.

Краткая теория

Программирование - это процесс создания программы, который может быть представлен как последовательность следующих шагов:

- Определение требований к программе;
- Разработка или выбор алгоритма решения поставленной задачи;
- Написание команд;
- Отладка;
- Тестирование.

Язык программирования – это программа, имеющая специальные команды для перевода алгоритмического языка в машинный код.

Классификация языков программирования:

1. *Языки низкого уровня* – записи команд на языке машинного кода.
Пример. Ассемблер.
2. *Языки высокого уровня* – запись команд на обычном языке.

Пример. Бейсик, Паскаль, Си.

Программа, подготовленная на языке программирования, проходит этап *трансляции*, когда происходит перевод программы в машинные коды.

Трансляция может выполняться с использованием средств компиляторов или интерпретаторов. *Компиляторы* транслируют всю программу, но без ее выполнения. *Интерпретаторы* выполняют пооператорную обработку и выполнение программы.

Система программирования – это комплекс программ, облегчающих работу с языком программирования:

- Язык программирования;
- Компилятор;
- Отладчик;
- Справочная система;
- Визуальные компоненты для наглядного представления окна рабочей программы.

Моделирование (этапы решения функциональных вычислительных задач):

1. Определение исходных данных.
2. Выбор метода решения (словесное пояснение решений).
3. Алгоритмизация.
4. Программирование.
5. Тестирование – проверка на исправность и обнаружение ошибок.
6. Отладка.
7. Подготовка готовой программы (ее окончательное оформление).

Для изучения языков программирования рассмотрим язык программирования *Turbo Pascal 7.0*. Язык Паскаль, названный в честь французского математика и философа Блеза Паскаля (1623-1662), был создан как учебный язык программирования в 1968-1971 годах Никлаусом Виртом в

Высшей технической школе в Цюрихе. К настоящему времени этот язык несколько раз усовершенствовался в соответствии с научным прогрессом.

Перед тем как начать изучать возможности *Turbo Pascal 7.0*, необходимо изучить основные понятия этого языка, а так же изучить структуру данной оболочки и методы работы в ней.

Сначала рассмотрим основные понятия данного языка программирования.

Алгоритм – точное предписание, определяющее процесс перехода от исходных данных к результату.

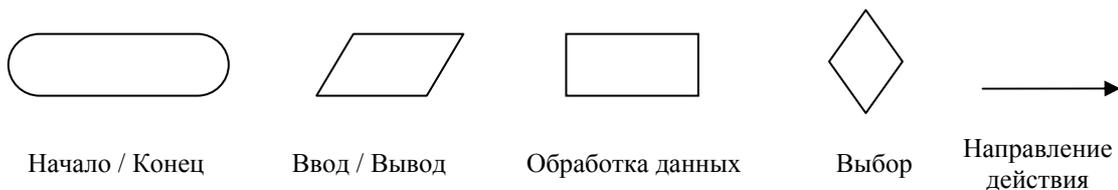
Свойства алгоритма:

1. *Дискретность* – алгоритм должен быть разбит на отдельные действия.
2. *Понятность* – точное понятие команды.
3. *Определенность* – исполнитель алгоритма не должен сомневаться в следующем шаге.
4. *Результативность* – по завершению выполнения алгоритма должен быть получен результат.
5. *Массовость* – алгоритм должен быть написан для решения ряда подобных задач.

Способы представления алгоритма:

- Словесный;
- Математическая формула;
- Табличный;
- Графический (блок - схема)

Основные элементы блок – схемы:



Переменная – это область памяти компьютера.

Каждая переменная, используемая в программе, должна иметь имя. В качестве имени переменной можно использовать последовательность букв латинского алфавита и цифр, причем первым символом в имени должна быть буква. Пробел в имени переменной использовать нельзя.

Компьютер обрабатывает данные различных типов. В *Pascal* существуют 5 основных типа данных:

- *Integer* – целый тип (может быть положительное и отрицательное число, а также 0; диапазон значений – от –32768 до 32767);
- *Real* – вещественный тип (используется для хранения чисел с дробной частью; диапазон – до 11 цифр десятичной части вещественного числа);
- *Char* – символьный тип (значениями являются символы языка *Pascal*);
- *Boolean* – логический тип (данные этого типа принимают два значения: *True* - истина, *False* - ложь);
- *String* – строковый тип.

В программах на языке *Pascal* каждая переменная перед использованием должна быть описана. С помощью описания устанавливается не только факт существования переменной, но и задается ее тип, тем самым определяется диапазон допустимых значений.

В тексте программы описание каждой переменной, как правило, помещают на отдельной строке. После имени переменной через двоеточие указывается тип переменной, затем ставится символ «точка с запятой». Если программе несколько переменных одного типа, то можно через запятую перечислить имена переменных, относящихся к одному типу, и после имени последней переменной через двоеточие указать тип.

Пример. A:integer; b,c,d:real;

Чтобы задать переменной какое – либо значение, используется **оператор присваивания**.

Синтаксис: Имя переменной:=Выражение;

Пример. D:=b*b-4*a*c;

В левой части оператора присваивания стоит идентификатор переменной, а в правой – выражение, результат вычисления которого должен стать значением этой переменной.

Выражения, стоящие в правой части оператора присваивания, могут содержать **операнды**, т. е. Функции, переменные или константы и знаки операций.

Операторы языка Pascal:

- + - сложение
- * - умножение
- - - вычитание
- / - деление
- *div* – деление нацело
- *mod* - вычисление остатка от деления

Тип выражения определяется типом операндов, входящих в выражение и зависит от операций, выполняемых над ними.

- Операторы +, -, * - если хотя бы один операнд имеет тип *Real*, то результат имеет тип *Real*; если все операнды типа *Integer*, то результат имеет тип *Integer*.
- Оператор / - результатом всегда является выражение типа *Real*.
- Операторы *div*, *mod* – операнды этих операторов всегда должны быть типа *Integer*.

В *Pascal* для вывода информации на экран служат операторы **Write** и **Writeln**.

Пример. Write(d); Writeln('Результат');

Writeln отличается от *Write* только тем, что после вывода сообщения или значения переменных курсор переводится в начало следующей строки.

Для ввода информации служат операторы **Read** и **Readln**.

Пример. Read(a); Readln(b,c);

Readln отличается от *Read* тем, что после выделения очередного числа из введенной с клавиатуры строки и присваивания его последней переменной из списка инструкции *Readln* оставшаяся часть строки теряется и следующая инструкция *Read* или *Readln* будет требовать нового ввода.

Pascal программа состоит из заголовка и тела программы.

Синтаксис: **Program** Имя;

тело программы: раздел описания и раздел операторов

Раздел описания определяет все используемые в программе переменные. Раздел описания операторов определяет действия, которые нужно выполнить над переменными.

Раздел описания состоит из 5 разделов:

1. Раздел меток.

Перед любым оператором можно поставить метку, что позволит выполнять переход при выполнении команды *GOTO*.

2. Раздел констант.

В нем идентификаторам присваиваются определенные значения, которые будут действовать в программе

Синтаксис: **CONST**

Идентификатор1=const1;

Идентификатор2=const2;

3. Раздел типов.

Создаются свои типы данных и используются при написании программы.

Синтаксис: **TYPE**

Идентификатор1=описание типа1;

Идетнификатор2=описание типа2;

4. Раздел переменных.

Каждая переменная в программе должна быть предварительно описана в разделе описания переменных.

Синтаксис: VAR

Список1 переменных:=тип1 переменных;

Список2 переменных:=тип2 переменных;

5. Раздел функций и процедур.

Раздел операторов последний в теле программы.

Синтаксис: Begin

Оператор1;

Оператор2;

.....

End.

При написании программы используются стандартные математические функции и логические операции.

Математические функции:

- *Abs*(выражение) – абсолютное значение аргумента (целый или вещественный тип)
- *Sqr*(выражение) – квадрат аргумента (целый или вещественный тип)
- *Sqrt*(выражение) – квадратный корень из аргумента (вещественный тип)
- *Sin*(выражение) – синус аргумента (вещественный тип)
- *Cos*(выражение) – косинус аргумента (вещественный тип)
- *Arctan*(выражение) – арктангенс аргумента (вещественный тип)
- *Exp*(выражение) – экспонента аргумента (вещественный тип)
- *Ln*(выражение) – натуральный логарифм аргумента (вещественный тип)
- *Pi* – число ПИ

Логические операции:

- *Not* – «НЕ», или логическое отрицание
- *Or* – «ИЛИ», или логическое сложение
- *And* – «И», или логическое умножение
- *Xor* – исключающее или.

Теперь рассмотрим структуру оболочки *Turbo Pascal 7.0*.

Вызов оболочки *Turbo Pascal 7.0* осуществляется посредством активизации файла *turbo.exe*. После запуска файла *turbo.exe* на экране появится основной экран оболочки *Turbo Pascal 7.0*.

Основной экран состоит из трех различных по функциональному значению частей: строки меню, рабочей зоны и строки состояния.

Строка меню оболочки активизируется нажатием функциональной клавиши *F10* или щелчком мыши. Строка меню содержит имена следующих меню:

- *File* – позволяет выполнять все основные операции с файлами (создавать новые, загружать имеющиеся, сохранять созданные и отредактированные файлы, выводить на печать содержимое файлов);
- *Edit* - позволяет выполнять все основные операции редактирования текста (копировать, вставлять, удалять фрагменты текста, а также восстанавливать первоначальный вариант редактируемого текста);
- *Search* – позволяет осуществлять поиск фрагментов текста и при необходимости производить замену найденного фрагмента новым;
- *Run* – позволяет запускать программу находящуюся на рабочей зоне, а также при необходимости пошагово выполнять данную программу или ее часть;
- *Compile* – позволяет осуществить компиляцию программы, которая находится в рабочей зоне;
- *Debug* – содержит программы, облегчающие процесс поиска ошибок в программе;

- *Tools* – позволяет выполнить некоторые программы, не выходя из оболочки Turbo Pascal 7.0;
- *Options* – позволяет установить необходимые для работы параметры компилятора оболочки Turbo Pascal 7.0;
- *Windows* – позволяет выполнять все необходимые операции с окнами (открывать, закрывать, перемещать, изменять размер);
- *Help* – позволяет получить имеющуюся в системе справочную информацию.

В рабочей зоне в момент открытия активизировано одно окно с номером 1 в правом верхнем углу и с заголовком Noname00.pas. В данном окне вводится текст программы. Мигающий курсор указывает то место на экране, в котором будет появляться текст.

Строка состояния, находящаяся в нижней части экрана, демонстрирует некоторые из доступных операций оболочки и комбинации клавиш для их быстрого вызова, которые позволяют выполнить соответствующие операции, минуя стандартную процедуру их вызова через меню.

Перед использованием созданную программу необходимо сохранить. Для этого необходимо воспользоваться клавишей *F12*, при нажатии которой выполняется операция *Save* (Сохранение). В открывшемся диалоговом окне *Save File As* задается имя программы и нажимается клавиша *Enter*.

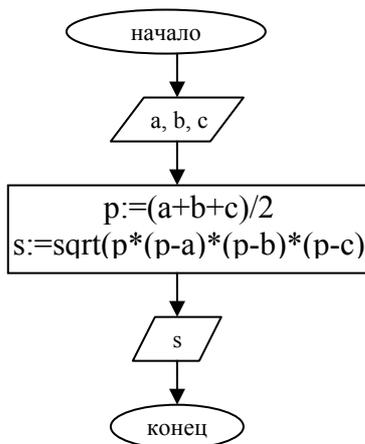
Компиляция и запуск программы в Turbo Pascal 7.0 осуществляется комбинацией клавиш *Ctrl+F9* (или вызовом команды *Run* из меню *Run*). Если транслятор обнаружит в программе синтаксическую ошибку, он выдаст соответствующее сообщение (*Error ...*) на экран, указав место ошибки.

Просмотреть результат работы программы, т. е. выводимую на экран информацию, а не сам текст программы, можно благодаря режиму «разумного переключения». По завершении процесса вывода результатов работы

программы на экране монитора снова появится текст программы. Необходимо нажать комбинацию клавиш *Alt+F5*. На экране появится результат работы программы. Чтобы завершить просмотр результат работы программы, необходимо нажать любую клавишу.

Завершить работу с оболочкой *Turbo Pascal 7.0* можно с помощью комбинации клавиш *Alt+X* или командой *Quit* меню *File*.

Пример. Вычислить площадь прямоугольника по формуле Герона.



```

Program Geron;
Var a,b,c,p,s:real;
Begin
  Writeln('Введите стороны треугольника');
  Readln(a,b,c);
  p:=(a+b+c)/2;
  s:=sqrt(p*(p-a)*(p-b)*(p-c));
  Writeln('Площадь треугольника =', s);
End.
  
```

Задания:

1. Составить программу для вычисления высот треугольника со сторонами a, b, c , используя формулы $h_a = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)}$,

$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)}, \quad h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)}, \quad \text{где } p = \frac{a+b+c}{2}.$$

2. Найти значение функции: а) $F(x) = \frac{1 + \sin^2(x+y)}{2 + \left| x - \frac{2x}{1 + |\sin(x+y)|} \right|}$;

$$\text{б) } F(x) = \frac{\sin x}{2 \cos^2 x} - \cos x - \frac{3}{2} \operatorname{tg} x; \quad \text{в) } F(x) = x - \frac{x^2}{1 + \sin^2(x+y)}.$$

3. Вычислить значения переменных $\omega = \sqrt{x^2 + b} - \frac{b^2 \sin^3(x + a)}{x}$ и

$$y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}, \text{ где } a=1,5; b=15,5; x=-2,9.$$

Вопросы для самоконтроля:

1. Дать определение алгоритма.
2. Перечислить основные свойства алгоритма.
3. Перечислить способы задания алгоритма.
4. Основные элементы блок-схемы.
5. Дать определение переменной.
6. Основные типы переменных.
7. Перечислить основные разделы программы.
8. Перечислить основные математические функции и логические операции.

Лабораторная работа №5

Тема: Оператор ветвления.

Цель работы: овладение практическими навыками разработки и программирования вычислительного процесса разветвляющей структуры, получение дальнейших навыков по отладке и тестированию программы.

Краткая теория

В языке Pascal **условие** – это выражение логического типа (Boolean), которое может принимать одно из двух значений: «истина» (*True*) или «ложь» (*False*).

В Pascal имеется шесть операторов, позволяющих сравнивать между собой значения числовых переменных, а также значение переменной и константу (число).

<i>Оператор</i>	<i>Описание</i>	<i>Значение выражения</i>
>	Больше	True, если первый операнд больше второго, иначе False
<	Меньше	True, если первый операнд меньше второго, иначе False
=	Равно	True, если первый операнд равен второму, иначе False
<>	Не равно	True, если первый операнд не равен второму, иначе False
>=	Больше или равно	True, если первый операнд больше или равен второму, иначе False
<=	Меньше или равно	True, если первый операнд меньше или равен второму, иначе False

Использование операторов сравнения позволяет записывать простые условия. Из простых условий, которые являются выражениями логического типа, можно строить сложные условия с применением к ним, как к операндам, логических операторов: not, and, or.

<i>Op1</i>	<i>Op2</i>	<i>Op1 AND Op2</i>	<i>Op1 OR Op2</i>	<i>NOT Op1</i>
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

При записи сложных условий важно учитывать то, что логические операторы имеют более высокий приоритет, чем операторы сравнения, поэтому простые условия следует брать в скобки.

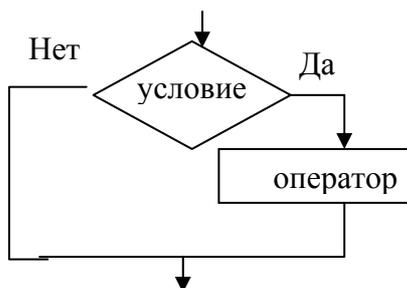
При решении задач нередко возникает ситуация, когда действие осуществляется в зависимости от некоторого условия. Для программирования проверки условия и выбора действия в зависимости от этого условия используются **условные операторы**.

Существует две формы условного оператора:

- *Неполное ветвление*

Синтаксис:

If условие
Then оператор;

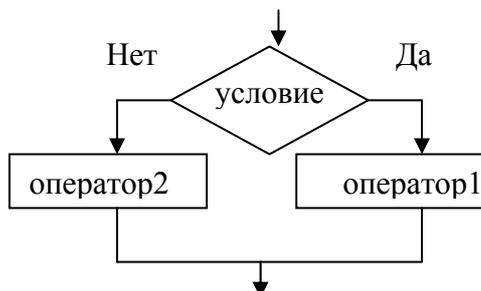


Действия: Если условие принимает значение *True*, то выполняется оператор, иначе никаких действий не происходит.

- *Полное ветвление (развилка)*

Синтаксис:

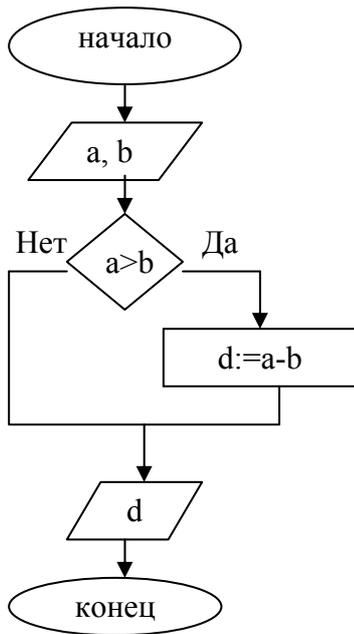
If условие
Then оператор1
Else оператор2;



После *оператор1* ";" не ставится.

Действия: Проверяется условие, если оно истинно, т. е. принимает значение *True*, то выполняется *оператор1*, иначе, если условие ложно, т. е. принимает значение *False*, то выполняется *оператор2*.

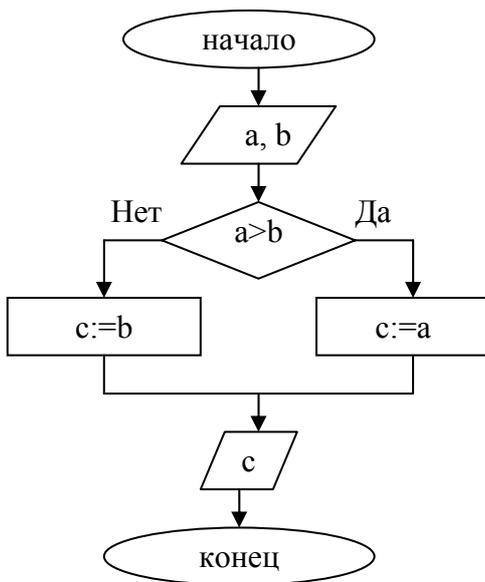
Пример. Даны два числа *a* и *b*. Если $a > b$, то вывести разность данных чисел.



```

Program chislo;
Var a,b,d:integer;
Begin
  Writeln('Введите числа');
  Readln(a,b);
  If a>b
  Then d:=a-b;
  Writeln('Разность чисел=',d);
End.
  
```

Пример. Найти максимум двух целых чисел.



```

Program max;
Var a,b,c:integer;
Begin
  Writeln('Введите числа');
  Readln(a,b);
  If a>b
  Then c:=a
  Else c:=b;
  Writeln(c);
End.
  
```

Задания:

1. Даны действительные числа x и y . Если $x < y$, то вывести сумму этих чисел, иначе – разность этих чисел.
2. Даны три действительных числа. Вывести на экран из них те, значения которых принадлежат промежутку $[-3; 3]$.

$$3. \text{ Вычислить } f = \begin{cases} a, & a < 0 \\ a^2 - 2a + 2, & 0 \leq a < 4 \\ a^2 - \sin a^2, & a > 4 \end{cases}.$$

4. Найти корни квадратного уравнения $ax^2 + bx + c = 0$, где

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}, \quad D = b^2 - 4ac.$$

Вопросы для самоконтроля:

1. Дать определение условия, условного оператора.
2. Неполное ветвление: синтаксис, блок-схема.
3. Полное ветвление: синтаксис, блок-схема.

Лабораторная работа №6

Тема: Составной оператор. Оператор выбора.

Цель работы: овладение практическими навыками разработки и программирования вычислительного процесса выборочной структуры, получение дальнейших навыков по отладке и тестированию программы.

Краткая теория

Оператор выбора позволяет выбрать одно из нескольких возможных продолжений программы. Параметром, по которому осуществляется выбор, служит **ключ выбора s** – выражение любого порядкового типа (кроме типов *Real* и *String*).

Синтаксис:

Case s of

c1:оператор1;

c2:оператор2;

.....
сn:оператор n
else оператор;

End;

где c_1, c_2, \dots, c_n – все простые типы *Integer*, *Char*, за исключением *Real*. Это конкретное значение управляющей переменной и выражения s , при котором необходимо выполнить соответствующий оператор, игнорируя все остальные.

Если в наборе несколько значений, то они разделяются между собой «,». Можно указывать диапазоны значений между которыми ставятся «...».

Между набором значений c и соответствующим ему оператором должно стоять «:». Значения $c_1 \dots c_n$ не должны повторяться.

Пример. По номеру дня недели вывести его название.

```
Program day;  
Var a:integer;  
Begin  
  Writeln('Введите номер дня недели');  
  Readln(a);  
  Case a of  
    1:writeln('понедельник');  
    2:writeln('вторник');  
    3:writeln('среда');  
    4:writeln('четверг');  
    5:writeln('пятница');  
    6:writeln('суббота');  
    7:writeln('воскресенье')  
    else writeln('дня недели с таким номером нет');  
  end;  
End.
```

Пример. Дана отметка в численной форме. В зависимости от балла вывести оценку.

```
Program ozenka;  
Var a:integer;  
Begin  
  Writeln('Введите вашу отметку');  
  Readln(a);  
  Case a of  
    5:writeln('отлично');  
    4:writeln('хорошо');  
    3:writeln('удовлетворительно');  
    1,2:writeln('неудовлетворительно')  
  Else writeln('такой оценки нет');  
End;  
End.
```

Составной оператор используется в том случае, если необходимо выполнить более одного оператора в тех случаях, когда синтаксис языка *Pascal* допускает присутствие только одного оператора. Составной оператор может состоять из произвольного количества операторов, разделенных между собой “;”.

Синтаксис:

```
Begin  
  Оператор1;  
  Оператор2;  
  .....  
  Оператор n;  
End;
```

Пример. Даны два целых числа. Вывести их квадраты, если они оба положительны и их модели в противном случае.

```

Program chisla;
Var a,b:integer;
Begin
  Writeln('Введите числа');
  Readln(a,b);
  If (a>0) and (b>0)
  Then
    Begin
      a:=sqr(a);
      b:=sqr(b);
    end
  Else
    Begin
      a:=sqr(a);
      b:=sqr(b);
    end;
  Writeln('a=',a,' ','b=',b);
End.

```

Задания:

1. Написать программу, выводящую на экран название времени года в зависимости от номера месяца.
2. Написать программу, которая определяет по номеру дня недели выходной это день или рабочий.
3. Дана цена в цифровом виде от 1 до 9. В зависимости от цифры вывести цену в правильном падеже (рубль, рублей, рубля).

4. Даны три числа. Вывести их квадраты, если первое число равно второму, иначе – кубы этих чисел.
5. Даны два целых числа. Если оба числа четные, то вывести произведение и сумму этих чисел, иначе вывести модули этих чисел.

Вопросы для самоконтроля:

1. Оператор выбора.
2. Составной оператор.

Лабораторная работа №7

Тема: Циклические структуры.

Цель работы: овладение практическими навыками разработки и программирования вычислительного процесса циклической структуры, получение дальнейших навыков по отладке и тестированию программы.

Краткая теория

При решении многих задач некоторая последовательность действий приходится выполнять несколько раз. Такие повторяющиеся действия называются **циклами**.

В языке Pascal циклические вычисления реализуются при помощи инструкций **For, While, Repeat**.

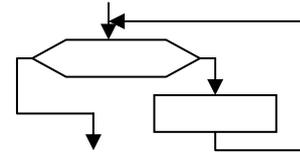
1. Оператор цикла с параметром.

Данный оператор используется, если надо выполнить некоторую последовательность действий несколько раз, причем число повторений заранее известно.

Синтаксис:

For параметр цикла: =мл знач. **to** ст знач. **do**

For параметр цикла: =ст..знач. **downto** мл.знач. **do**



Здесь *For*, *to*, *do* – зарезервированные слова (для, до, делать).

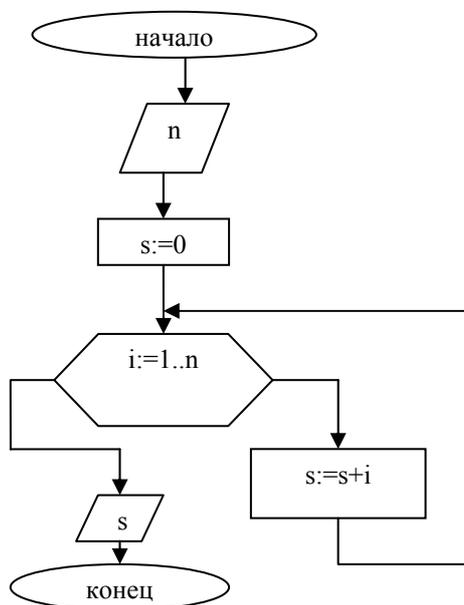
Если параметр возрастает, то между границами его значений ставится *to*, если убывает, то *downto*.

Действия: параметру цикла присваивается начальное значение и выполняется тело цикла, затем параметру цикла присваивается следующее значение и вновь выполняется тело цикла и так до тех пор, пока не будет перебраны все значения параметра цикла.

Замечание 1. Счетчик цикла увеличивается автоматически на единицу.

Замечание 2. Если в цикле используется более одного оператора, то необходимо использовать операторные скобки (составной оператор).

Пример. Найти сумму всех целых чисел от 1 до N.



```
Program summa;  
Var i,n,s:integer;  
Begin  
  Write('n=');  
  Readln(n);  
  s:=0;  
  For i:=1 to n do  
    s:=s+i;  
  Writeln('s=',s);  
End.
```

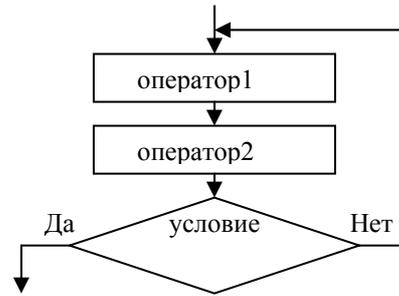
2. Оператор цикла с постусловием.

Данный оператор используется в программе, если надо провести некоторые повторяющиеся вычисления, однако число повторов заранее не известно и определяется самим ходом вычисления.

Синтаксис:

Repeat тело цикла

Until условие;



Здесь *Repeat, Until* – зарезервированные слова (повторять до тех пор, пока не будет выполнено условие).

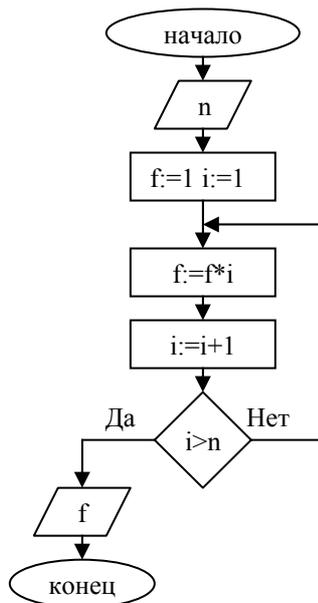
Действия: сначала выполняются операторы, потом проверяется условие. Если оно ложно, то снова выполняются операторы, и так до тех пор, при проверке условие не примет значение *True*.

Замечание 1. Тело цикла будет выполняться хотя бы один раз.

Замечание 2. Если в теле цикла используется несколько операторов, то составной оператор не нужен.

Замечание 3. Параметр цикла может быть вещественного типа и задается в теле цикла.

Пример. Вычислить факториал числа N ($N!$).



```
Program summa;
```

```
Var i,n:integer;
```

```
    f:real;
```

```
Begin
```

```
    Write('n=');
```

```
    Readln(n);
```

```
    f:=1;
```

```
    i:=1;
```

```
    Repeat
```

```
        f:=f*i;
```

```
        i:=i+1;
```

```
    Until i>n;
```

```

Writeln('f=',f);
End.

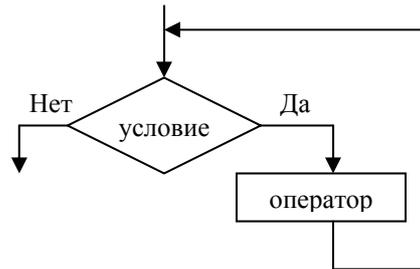
```

3. Оператор цикла с предусловием.

Данный оператор, как и оператор *Repeat*, используется в программе, если надо провести некоторые повторяющиеся вычисления, однако число повторов заранее не известно и определяется самим ходом вычисления.

Синтаксис:

While условие **Do** оператор;



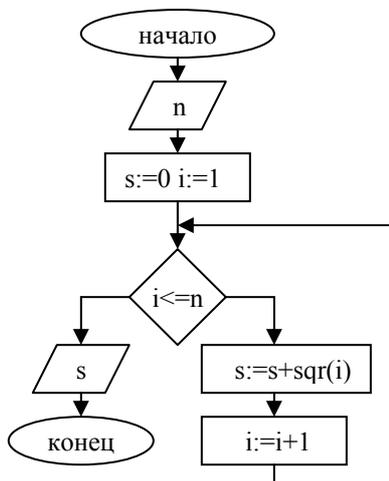
Здесь *While, Do* – зарезервированные слова (пока выполняется условие делать).

Действия: сначала проверяется условие, если оно истинно, то выполняется тело цикла. Затем снова проверяется условие и т. д. Если условие ложно, то цикл завершается и выполняется оператор, стоящий непосредственно после цикла.

Замечание 1. Если в теле цикла используется несколько операторов, то необходим составной оператор.

Замечание 2. Параметр цикла может быть вещественного типа, задается и увеличивается пользователем в теле цикла.

Пример. Вычислить сумму квадратов чисел от 1 до N.



```

Program summa;
Var i,n,s:integer;
Begin
  Write('n=');
  Readln(n);
  s:=0;
  i:=1;

```

```

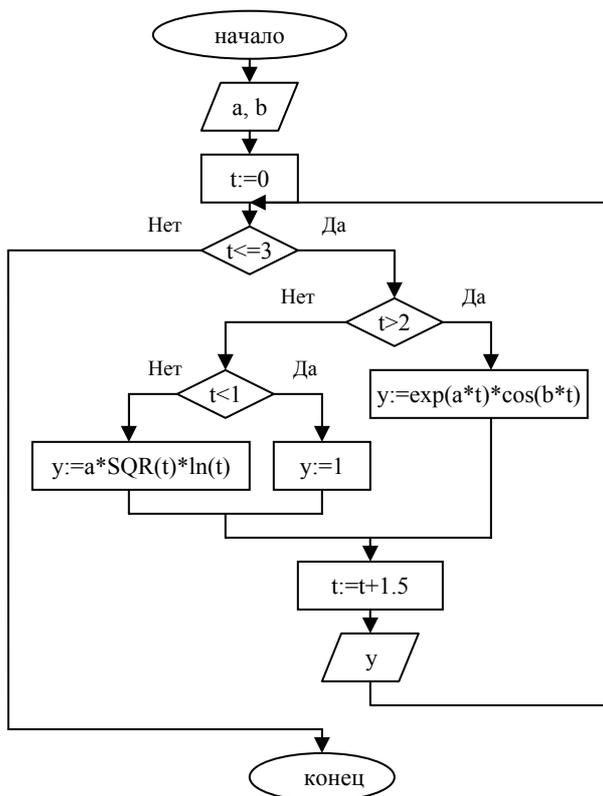
While i<=n do
  Begin
    s:=s+sqr(i);
    i:=i+1;
  End;
Writeln('s=',s);
End.

```

Пример. Вычислить значение функции $y = \begin{cases} at^2 \ln t, & 1 \leq t \leq 2 \\ 1, & t < 1 \\ e^{at} \cos bt, & t > 2 \end{cases}$ на интер-

вале [0; 3] с шагом 0,15.

Задачи такого типа называются задачами на табулирование, т. е. при решении таких задач результаты выводятся на каждом шаге.



```

Program taylir;
Var a,b:integer;
    t,y:real;
Begin
t:=0;
While t<=3 do
Begin
If t>2
Then y:=exp(a*t)*cos(b*t);
If t<1
Then y:=1
Else y:=a*SQR(t)*ln(t);
Writeln ('t=',t,' ','y=',y);
t:=t+0.15;
End;
End.

```

Задания:

1. Вычислить и вывести значения функции $y = \frac{x^2 - 3x + 2}{x^2 + 2x - 1}$ на интервале $[0; 2]$ с шагом $h=0,15$.

2. Вычислить значение функции $y = \begin{cases} \pi x^2 - \frac{7}{x^2}, & x < 1,3 \\ ax^3 + 7\sqrt{x}, & x = 1,3, a=1,5 \\ \lg(x + 7\sqrt{x}), & x > 1,3 \end{cases}$ на интервале $[0,8; 2]$ с шагом $h=0,1$.

3. Вычислить $z = 1 \cdot \frac{1}{2} \cdot \frac{1}{3} \cdot \Lambda \cdot \frac{1}{n}$.

4. Вычислить $y = \cos x + 2 \cos x + 3 \cos x + \Lambda + n \cos x$.

5. Вычислить $s = 1 + 3^2 + 5^3 + \Lambda + (2n - 1)^n$.

Вопросы для самоконтроля:

1. Дать определение цикла.
2. Цикл с параметром: синтаксис, блок-схема.
3. Цикл с предусловием: синтаксис, блок-схема.
4. Цикл с постусловием: синтаксис, блок-схема.
5. Сколько раз будут выполнены инструкции между Begin и End?

a) for i:=j to j+1 do b) k:=0;

Begin

for i:=2 downto j+1 do

...

Begin

End;

...

End;

6. Чему будет равен x после выполнения инструкций?

x:=0;

for i:=-5 to 0 do

n:=i*i;

```
x:=x+n;
```

7. Чему будет равно значение переменной *n* после выполнения инструкции?

```
n:=0;
```

```
while n<5 do
```

```
begin
```

```
  n:=n*n;
```

```
end;
```

8. Что делают следующие инструкции?

```
n:=0;
```

```
repeat
```

```
  write('*');
```

```
  n:=n+1;
```

```
until n<5;
```

Лабораторная работа №8

Тема: Одномерные массивы.

Цель работы: овладение практическими навыками работы с одномерными массивами, особенностями их ввода и вывода, приобретение дальнейших навыков по организации программ циклической структуры с использованием приемов программирования.

Краткая теория

Одномерный массив - это фиксированное количество элементов одного и того же типа, объединенные одним именем, где каждый элемент имеет свой номер.

Пример. Дано 30 целых чисел от 25 до 54, объединенных общим именем А.

N	A	A – общее имя для всех элементов
1	25	Элементы массива $A[i]$ – целые числа
2	25	$A[3]=27$
3	27	
...	...	
30	54	

Описание массива:

1. В разделе описания переменных массив описывается следующим образом:

Var имя массива: **array**[1.. t_1] **of** t_2 ;

Array – массив; *of* – из;

t_1 – количество элементов массива;

t_2 – тип элементов массива, допустимый в языке *Pascal*.

Пример. Var A:array[1..30] of integer;

2. В разделе описания типов.

В разделе *Type* указывается тип массива, затем в разделе описания переменных *Var* перечисляются массивы, относящиеся к описанному (указанному) типу.

TYPE имя типа=**ARRAY**[1.. t_1] **OF** t_2 ;

VAR имя массива: имя типа;

Пример. Type mas=array[1..30] of integer;

Var A: mas;

Если в программе несколько массивов, например, A, B, C имеют тип mas, то через запятую перечисляются в разделе описания переменных VAR.

Типичные действия с массивами:

1. Вывод массива на экран.

Ввод выводом массива понимается вывод на экран значений элементов массива. Если в программе необходимо вывести значения всех элементов массива, то для этого удобно использовать инструкцию For, переменная-счетчик которой может быть использована как индекс элемента массива.

2. Ввод элементов массива.

Под вводом массива понимается ввод значений элементов массива. Как и вывод массива, ввод удобно реализовать при помощи инструкции For. Чтобы пользователь программы знал, ввода какого элемента массива ожидает программа, следует организовать вывод подсказок перед вводом очередного элемента массива. В подсказке обычно указывают индекс элемента массива. Элементы могут вводиться пользователем, либо с помощью генераторов случайных чисел.

3. Сортировка массивов.

Под сортировкой массива подразумевается процесс перестановки элементов с целью упорядочивания их в соответствии с каким-либо критерием. Существует много методов (алгоритмов) сортировки массивов. В приведенных примерах рассмотрим сортировку массива методом прямого выбора.

4. Поиск в массиве заданного элемента.

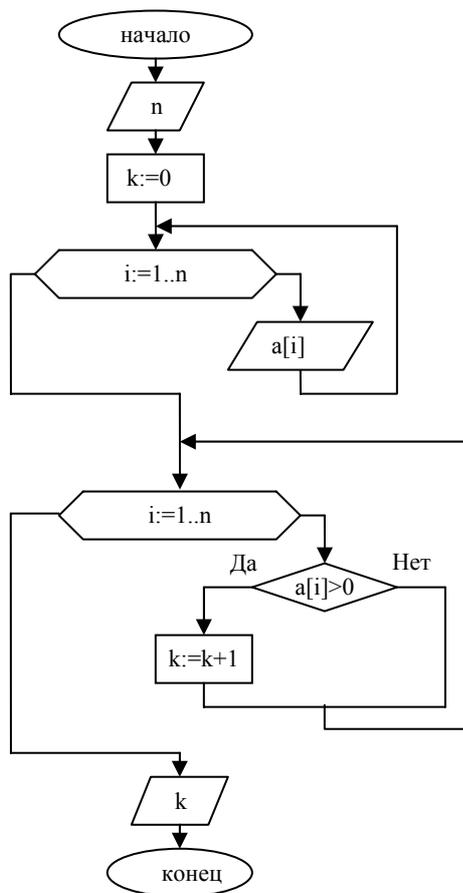
Для организации поиска в массиве могут быть использованы различные алгоритмы. Наиболее простой – это алгоритм простого перебора. Поиск осуществляется последовательным сравнением элементов с образцом, пока не будет найден искомый элемент или перебраны все элементы массива. Алгоритм простого перебора применяется, если элементы массива не упорядочены.

5. Поиск в массиве максимального и минимального элементов.

Алгоритм поиска максимального (минимального) элемента массива довольно очевиден. Делается предположение, что первый элемент массива

является максимальным (минимальным), затем остальные элементы массива сравниваются с этим элементом. Если обнаруживается, что проверяемый элемент больше (меньше) принятого за максимальный (минимальный), то этот элемент принимается за максимальный (минимальный) и продолжается проверка оставшихся элементов.

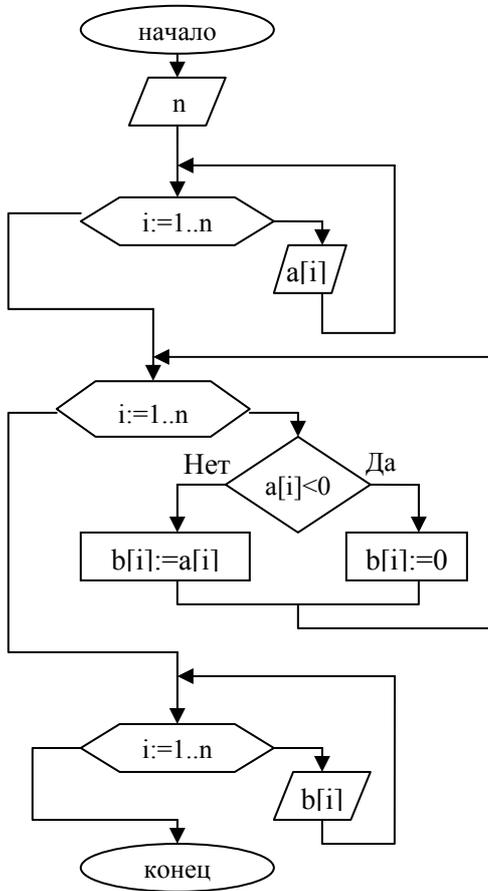
Пример. Дан массив А, состоящий из элементов целого типа. Найти количество положительных элементов этого массива.



```

Program massiv1;
Var n,i,k:integer;
    a:array[1..100] of integer;
Begin
  Write('Введите количество эле-
ментов массива: ');
  Readln(n);
  k:=0;
  Writeln('Введите элементы масси-
ва:');
  For i:=1 to n do
    Begin
      Write('a['i,']=');
      Readln(a[i]);
    End;
  For i:=1 to n do
    If a[i]>0
      Then k:=k+1;
  Writeln('k=',k);
End.
  
```

Пример. В одномерном массиве отрицательные элементы заменить на нуль. Вывести полученный массив на экран.



```

Program massiv2;
Var n,i:integer;
    a,b:array[1..100] of integer;
Begin
  Write('Введите количество элементов
массива: ');
  Readln(n);
  Writeln('Введите элементы массива:');
  For i:=1 to n do
    Begin
      Write('a[' ,i,']=');
      Readln(a[i]);
    End;
  For i:=1 to n do
    If a[i]<0
      Then b[i]:=0
      Else b[i]:=a[i];
  Writeln('Новый массив:');
  For i:=1 to n do
    Writeln('[b',i,']=' ,b[i], ' ');
  End.
  
```

В следующих примерах приводятся только программы, а блок-схемы предлагается изобразить самостоятельно.

Пример. В одномерном массиве упорядочить элементы по возрастанию. Вывести упорядоченный массив на экран.

```

Program porydoc;
Var a:array[1..100] of integer;
    buf,i,j,n:integer;
Begin
  Writeln('Введите количество элементов массива:');
  Readln(n);
  Writeln('Введите элементы массива:');
  For i:=1 to n do
  Begin
    Write('a[' ,i,']=');
    Readln(a[i]);
  End;
  For j:=1 to n-1 do
  Begin
    For i:=1 to n-1 do
      If a[i]>a[i+1]
      Then
      Begin
        buf:=a[i];
        a[i]:=a[i+1];
        a[i+1]:=buf;
      End;
    End;
  End;
  Writeln('Упорядоченный массив:');
  For i:=1 to n do
    Write(a[i], ' ');
  End.

```

Пример. В одномерном массиве найти минимальный элемент и его номер.

```

Program minimum;
Var a:array[1..100] of integer;
    min,number,i,n:integer;
Begin
  Writeln('Введите количество элементов массива:');
  Readln(n);
  Writeln('Введите элементы массива');
  For i:=1 to n do
  Begin
    Write('a[' ,i,']=');
    Readln(a[i]);
  End;
  min:=a[1];
  For i:=1 to n do
  If a[i]<min
  Then
  Begin
    min:=a[i];
    number:=i;
  End;
  Writeln('Минимальный элемент = ',min);
  Writeln('Номер минимального элемента = ',number);
End.

```

Задания:

1. Дан одномерный массив и целое число b . Найти произведение элементов массива, равных числу b .
2. Дан одномерный массив. Посчитать количество элементов, стоящих на четных местах и кратных 5.

3. Дан одномерный массив. Найти сумму квадратов отрицательных элементов массива и этой суммой заменить положительные элементы массива. Вывести новый массив на экран.
4. Дан одномерный массив. Упорядочить массив по убыванию.
5. Дан одномерный массив. Найти максимальный элемент этого массива и его порядковый номер.

Вопросы для самоконтроля:

1. Дать определение одномерного массива.
2. Перечислить разделы описания массивов. Привести примеры.
3. Перечислить типичные действия с массивами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексеев В.Е. Вычислительная техника и программирование. Практикум по программированию: Практик. пособие/ В.Е. Алексеев, А.С. Ваулин, Г.Б. Петров; Под ред. А.В. Петрова. – М.: Высш. шк., 1991. – 400 с.
2. Ильина М. М. Word 97. К вершинам мастерства. М.: Изд – во БИНОМ, 1998
3. Информатика для юристов и экономистов. /С.В. Симонович и др. – Учебник для вузов.- Санкт-Петербург: «Питер», 2001. – 688 с.
4. Информатика: Учебник/ Под ред. проф. Н. В. Макаровой. М.: Финансы и статистика, 2000
5. Информатика: Энциклопедический словарь для начинающих/Сост. Д. А. Поспелов - М.: Педагогика - Пресс, 1994. - 352 с. ил.
6. Лабораторный практикум. Текстовый редактор Word 97. По курсу информатика. О.В. Чугунова, О.П. Мельниченко. – Благовещенск: Амурский гос.ун-т –1999. - 51с.
7. Левин А. Самоучитель работы на компьютере. - М.: Международное агентство А.Д.&Т., 1997. - 480 с.

СОДЕРЖАНИЕ

Введение.....	3
Лабораторная работа №1.....	4
Лабораторная работа №2.....	23
Лабораторная работа №3.....	31
Лабораторная работа №4.....	41
Лабораторная работа №5.....	51
Лабораторная работа №6.....	55
Лабораторная работа №7.....	59
Лабораторная работа №8.....	65
Библиографический список.....	73

Алена Николаевна Киселева

старший преподаватель кафедры ОМиИ АмГУ,

Наталья Анатольевна Чалкина

старший преподаватель кафедры ОМиИ АмГУ
