

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Информатика

**Направление подготовки (специальность) Экономика**

**Профиль образовательной программы Экономика предприятий (организаций)**

**Форма обучения заочная**

## СОДЕРЖАНИЕ

<b>1. Конспект лекций</b>	<b>2</b>
1.1 Лекция № 1 Введение, основные понятия информатики	2
1.2. Лекция №2 . Системное программное обеспечение.....	6
1.3. Лекция №3 Алгоритмизация вычислительных процессов .....	11
1.4. Лекция №4 Программирование на алгоритмическом АЯ высокого уровня.....	20
1.5. Лекция №5 Компьютерные сети .....	23
<b>2. Методические указания по выполнению лабораторных работ .....</b>	<b>28</b>
2.1. Лабораторная работа № ЛР-1 Системное программное обеспечение.....	28
2.2 Лабораторная работа № ЛР-2 Прикладное программное обеспечение .....	33
2.3. Лабораторная работа №3 Программирование на алгоритмическом языке высокого уровня .....	47

## 1. КОНСПЕКТ ЛЕКЦИЙ

### 1. 1 Лекция №1 ( 2 часа).

Тема: «Введение, основные понятия информатики»

#### 1.1.1 Вопросы лекции:

1. Этапы информатизации общества
2. Цель и задачи информатики. Основные понятия
3. Структура информатики
4. Социальные, правовые и этические аспекты информатики

#### 1.1.2

##### 1. Этапы информатизации общества

Бурное развитие компьютерной техники и ИТ послужило толчком к развитию общества, построенного на использовании различной информации и получившего название информационного общества. В информационном обществе изменяется не только производство, но и весь уклад жизни, система ценностей, возрастает значимость культурного досуга по отношению к материальным ценностям. По сравнению с индустриальным обществом, где все направлено на производство и потребление товаров, в информационном обществе производятся интеллект и знания. Материальной и технологической базой информационного общества становятся компьютерная техника и компьютерные сети, ИТ, телекоммуникационные связи.

**Информационное общество** – общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей ее формы – знаний.

Ближе всех на пути к информационному обществу стоят страны с развитой информационной индустрией (США, Япония, Англия, страны Западной Европы).

Для перехода от индустриального общества к информационному должна была возникнуть ситуация информационного кризиса. И она возникла с тем, что в 20 веке лавинообразный поток информации, хлынувший на человека, сделал практически невозможной его ориентацию в этом объеме. Возникло большое количество избыточной, лишней информации. Началом перехода к информационному обществу стало внедрение

современных средств обработки и передачи информации в различных сферах деятельности человека. Этот процесс называется информатизация.

**Информатизация общества** – это процесс, при котором создаются условия, удовлетворяющие потребностям любого человека в получении необходимой информации (по закону РФ «Об информации, информатизации и защите информации» от 25 января 1995 года).

**Цель информатизации** – улучшение качества жизни людей за счет увеличения производительности и облегчения условий их труда.

## 2 Цель и задачи информатики. Основные понятия

Термин «информатика» прочно и надежно вошел в нашу жизнь. Он был заимствован из французского языка и обозначал название области, связанной с автоматизированной обработкой информации с помощью ЭВМ. Именно развитие компьютерной техники способствовало выделению информатики в самостоятельную область человеческой деятельности.

Следует отметить, что определений информатики в современной литературе множество. Это происходит оттого, что данная область знаний относительно новая и соответствующий понятийный аппарат не совсем устоялся. Анализ определений позволил выделить их существенную часть и сформулировать то определение, которое приведено ниже.

**Информатика** - область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и других средств вычислительной техники.

## 3 Структура информатики

С информатикой часто связывают одно из следующих понятий: это либо **отрасль производства**, либо **фундаментальная наука**, либо **прикладная дисциплина**, либо совокупность определенных средств, используемых для преобразования информации.

В состав **технических средств** входят компьютеры и связанные с ними периферийные устройства (мониторы, клавиатуры, принтеры и плоттеры, модемы и т.д.), линии связи, средства оргтехники и т.п., т.е. те материальные ресурсы, которые обеспечивают преобразование информации, причем главенствующую роль в этом списке играет компьютер.

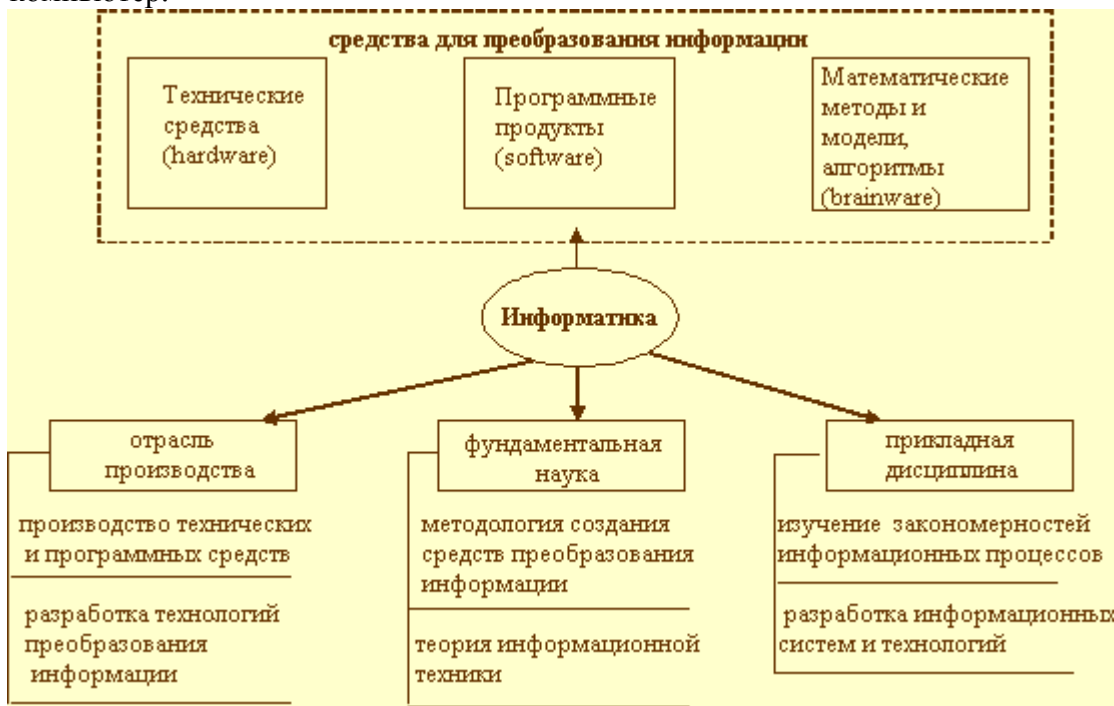


Рис 1.1 Структура информатики

К **программным средствам** (продуктам) относятся операционные системы, интегрированные оболочки, системы программирования и проектирования программных продуктов, различные прикладные пакеты, такие, как текстовые и графические редакторы, бухгалтерские и издательские системы и т.д. **Математические методы, модели и алгоритмы** являются тем базисом, который положен в основу проектирования и изготовления любого программного или технического средства в силу их исключительной сложности и, как следствие, невозможности умозрительного подхода к созданию.

Перечисленные выше три ресурсных компонента информатики играют разную роль в процессе информатизации общества. Так, совокупность программных и технических средств, имеющихся в том или ином обществе, и позволяет сделать его информационным, когда каждый член общества имеет возможность получить практически любую (исключая, естественно, секретную) интересующую его информацию (такие потребители информации называются конечными пользователями). В то же время, сложность технических и программных систем заставляет использовать имеющиеся технические и программные продукты, а также нужные методы, модели и алгоритмы для проектирования и производства новых и совершенствования старых технических и программных систем. В этом случае можно сказать, что средства преобразования информации используются для производства себе подобных. Тогда их пользователем является специалист в области информатики, а не конечный пользователь.

Разработкой абстрактных методов, моделей и алгоритмов, а также связанных с ними математических теорий занимается **фундаментальная наука**. Ее прерогативой является исследование процессов преобразования информации и на основе этих исследований разработка соответствующих теорий, моделей, методов и алгоритмов, которые затем применяются на практике.

Практическое использование результатов исследований информатики как фундаментальной науки воплощается в информатике - **отрасли производства**. В самом деле, широко известны западные фирмы по производству программных продуктов, такие как Microsoft, Lotus, Borland, и технических средств - IBM, Apple, Intel, Hewlett Packard и другие. Помимо производства самих технических и программных средств разрабатываются также и технологии преобразования информации.

Подготовкой специалистов в области преобразования информации занимается информатика как **прикладная дисциплина**. Она изучает закономерности протекания информационных процессов в конкретных областях и методологии разработки конкретных информационных систем и технологий.

Таким образом, главная функция информатики состоит в разработке методов и средств преобразования информации с использованием компьютера, а также в применении их при организации технологического процесса преобразования информации.

В рамках **прикладной дисциплины** информатики изучаются следующие вопросы:

- понятие информации, ее свойства, измерение информации, использование в управлении;
- способы кодирования информации;
- понятие и составные части информационных процессов;
- организация технических устройств преобразования информации, в частности компьютера;
- структура и методология проектирования программного обеспечения.

**Задачи информатики:**

- Разработка и производство современных средств вычислительной техники;
- Проектирование и внедрение прогрессивных технологий обработки информации, и как результат этого – возможность дальнейшей информатизации общества и повышения уровня его информационной культуры.

#### 4 Социальные, правовые и этические аспекты информатики

### **Социальные аспекты информатики**

Термин “социальные аспекты” применительно к большей части наук. Однако, информатика – не только наука.

Мало какие факторы так влияют на социальную сферу обществ как информатизация.

Информатизация общества – процесс проникновения информационных технологий во все сферы жизни и деятельности общества. Она сильнейшим образом влияет на структуру экономики ведущих в экономическом отношении стран. В числе их лидирующих отраслей промышленности традиционные добывающие и обрабатывающие отрасли оттеснены максимально наукоемкими производствами электроники, средств связи и вычислительной техники. В этих странах постоянно растут капиталовложения в научные исследования, включая фундаментальные науки. Темпы развития сферы высоких технологий и уровень прибылей в ней превышают в 5-10 раз темпы развития традиционных отраслей производства. Такая политика имеет и социальные последствия – увеличение потребности в высокообразованных специалистах и связанный с этим прогресс системы высшего образования. Информатизация меняет и облик традиционных отраслей промышленности и сельского хозяйства. Промышленные роботы, управляемые ЭВМ, станки с ЧПУ стали обычным оборудованием. Новейшие технологии в сельскохозяйственном производстве не только увеличивают производительность труда, но и облегчают его, вовлекают более образованных людей.

Казалось бы, компьютеризация и информационные технологии несут в мир одну лишь благодать, но социальная сфера столь сложна, что последствия любого, даже гораздо менее глобального процесса, редко бывают однозначными. Рассмотрим, например, такие социальные последствия информатизации как рост производительности труда, интенсификацию труда, изменение условий труда. Все это, с одной стороны, улучшает условия жизни многих людей, повышает степень материального и интеллектуального комфорта, стимулирует рост числа высокообразованных людей, а с другой – является источником повышенной социальной напряженности. Например, появление на производстве промышленных роботов ведет к полному изменению технологии, которая перестает быть ориентированной на человека. Тем самым меняется номенклатура профессий. Значительная часть людей вынуждена менять либо специальность, либо место работы – рост миграции населения характерен для большинства развитых стран. Государство и частные фирмы поддерживают систему повышения квалификации и переподготовки, но не все люди справляются с сопутствующим стрессом. Одним словом, жизнь в “информационном обществе” легче, по-видимому, не становится, а вот то, что она значительно меняется – несомненно

### **Правовые аспекты информатики**

Деятельность программистов и других специалистов, работающих в сфере информатики, все чаще выступает в качестве объекта правового регулирования. Некоторые действия при этом могут быть квалифицированы как правонарушения (преступления).

Правовое сознание в целом, а в области информатики особенно, в нашем обществе находится на низком уровне. Все ли знают ответы на следующие вопросы:

- можно ли, не копируя купленную программу, предоставить возможность пользоваться ею другому лицу;
- кому принадлежит авторское право на программу, созданную студентом в ходе выполнения дипломной работы;
- можно ли скопировать купленную программу для себя самого, чтобы иметь резервную копию;
- можно ли декомпилировать программу, чтобы разобраться в ее деталях или исправить ошибки;
- в чем состоит разница между авторским и имущественным правом.

Вопросов, подобных этим, возникает множество, но остановимся на правовом регулировании в области информатики в России. К 1992 году был принят Закон Российской Федерации “О ПРАВОВОЙ ОХРАНЕ ПРОГРАММ ДЛЯ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И БАЗ ДАННЫХ”, содержащий обширный план приведения российского законодательства в сфере информатики в соответствие с мировой практикой. Действие этого Закона распространяется на отношения, связанные с созданием и использованием программ для ЭВМ и баз данных. Также предусматривалось внести изменения и дополнения в Гражданский кодекс РФ, в Уголовный кодекс РФ, другие законодательные акты, связанные с вопросами правовой охраны программ для электронных вычислительных машин и баз данных, привести решения Правительства РФ в соответствие с Законом, обеспечить пересмотр и отмену государственными ведомствами и другими организациями РФ их нормативных актов, противоречащих указанному Закону, обеспечить принятие нормативных актов в соответствии с указанным Законом и т.д. Главное содержание данного Закона – юридическое определение понятий, связанных с авторством и распространением компьютерных программ и баз данных, таких как Авторство, Адаптация, База данных, Воспроизведение, Декомпилирование. Использование, Модификация и т.д., а также установление прав, возникающих при создании программ и баз данных – авторских, имущественных, на передачу, защиту, регистрацию, неприкосновенность и т.д.

### **Этические аспекты информатики**

#### **Этика – система норм нравственного поведения человека.**

Далеко не все правила, регламентирующие деятельность в сфере информатики, можно свести в правовым нормам. Очень многое определяется соблюдением неписаных правил поведения для тех, кто причастен к миру компьютеров. Впрочем, в этом отношении информатика ничуть не отличается от любой другой сферы деятельности человека в обществе.

Морально-этические нормы в среде информатиков отличаются от этики повседневной жизни несколько большей открытостью, альтруизмом. Большинство нынешних специалистов-информатиков сформировались и приобрели свои знания и квалификацию благодаря бескорыстным консультациям и содействию других специалистов. Очевидно, поэтому они готовы оказать бескорыстную помощь, дать совет или консультацию, предоставить компьютер для выполнения каких-либо манипуляций с дискетами и т.д. Ярким примером особой психологической атмосферы в среде информатиков является расширяющееся международное движение программистов, предоставляющих созданные ими программные средства для свободного распространения.

## **2.1. Лекция № 2 (2 часа)**

Тема: Системное программное обеспечение.

### **2.1.1 Вопросы лекции:**

1. Понятие программного обеспечения и классификация. Системное программное обеспечение
2. Понятие операционных систем, назначение, функции, классификация
3. Понятие файловой системы
4. Операционная система WINDOWS. Общие сведения о WINDOWS. Интерфейс пользователя WINDOWS. Основы работы в WINDOWS
- 4.1.2. Краткое содержание вопросов
  1. Понятие программного обеспечения и классификация. Системное программное обеспечение

Совокупность программ, предназначенная для решения задач на ПК, называется программным обеспечением. Состав программного обеспечения ПК называют программной конфигурацией.

Программное обеспечение, можно условно разделить на три категории:

- системное ПО (программы общего пользования), выполняющие различные вспомогательные функции, например создание копий используемой информации, выдачу справочной информации о компьютере, проверку работоспособности устройств компьютера и т.д.
- прикладное ПО, обеспечивающее выполнение необходимых работ на ПК: редактирование текстовых документов, создание рисунков или картинок, обработка информационных массивов и т.д.
- инструментальное ПО (системы программирования), обеспечивающее разработку новых программ для компьютера на языке программирования.



## Системное ПО

Это программы общего пользования не связаны с конкретным применением ПК и выполняют традиционные функции: планирование и управление задачами, управления вводом-выводом и т.д.

Другими словами, системные программы выполняют различные вспомогательные функции, например, создание копий используемой информации, выдачу справочной информации о компьютере, проверку работоспособности устройств компьютера и т.п.

К системному ПО относятся:

- операционные системы (эта программа загружается в ОЗУ при включении компьютера)
- программы – оболочки (обеспечивают более удобный и наглядный способ общения с компьютером, чем с помощью командной строки DOS, например, Norton Commander)
- операционные оболочки – интерфейсные системы, которые используются для создания графических интерфейсов, мультипрограммирования и т.
- драйверы (программы, предназначенные для управления портами периферийных устройств, обычно загружаются в оперативную память при запуске компьютера)
- утилиты (вспомогательные или служебные программы, которые представляют пользователю ряд дополнительных услуг)

К утилитам относятся:

- диспетчеры файлов или файловые менеджеры

- средства динамического сжатия данных (позволяют увеличить количество информации на диске за счет ее динамического сжатия)
- средства просмотра и воспроизведения
- средства диагностики; средства контроля позволяют проверить конфигурацию компьютера и проверить работоспособность устройств компьютера, прежде всего жестких дисков
- средства коммуникаций (коммуникационные программы) предназначены для организации обмена информацией между компьютерами
- средства обеспечения компьютерной безопасности (резервное копирование, антивирусное ПО).

Необходимо отметить, что часть утилит входит в состав операционной системы, а другая часть функционирует автономно. Большая часть общего (системного) ПО входит в состав ОС. Часть общего ПО входит в состав самого компьютера (часть программ ОС и контролирующих тестов записана в ПЗУ или ППЗУ, установленных на системной плате). Часть общего ПО относится к автономным программам и поставляется отдельно.

### **Прикладное ПО**

Прикладные программы могут использоваться автономно или в составе программных комплексов или пакетов. Прикладное ПО – программы, непосредственно обеспечивающие выполнение необходимых работ на ПК: редактирование текстовых документов, создание рисунков или картинок, создание электронных таблиц и т.д.

Пакеты прикладных программ – это система программ, которые по сфере применения делятся на проблемно – ориентированные, пакеты общего назначения и интегрированные пакеты. Современные интегрированные пакеты содержат до пяти функциональных компонентов: тестовый и табличный процессор, СУБД, графический редактор, телекоммуникационные средства.

К прикладному ПО, например, относятся:

- Комплект офисных приложений MS OFFICE
- Бухгалтерские системы
- Финансовые аналитические системы
- Интегрированные пакеты делопроизводства
- CAD – системы (системы автоматизированного проектирования)
- Редакторы HTML или Web – редакторы
- Браузеры – средства просмотра Web - страниц
- Графические редакторы
- Экспертные системы
- И так далее.

### **2. Понятие операционных систем, назначение, функции, классификация**

Операционная система составляет основу программного обеспечения ПК. Операционная система представляет комплекс системных и служебных программных средств, который обеспечивает взаимодействие пользователя с компьютером и выполнение всех других программ.

С одной стороны, она опирается на базовое программное обеспечение ПК, входящее в его систему BIOS, с другой стороны, она сама является опорой для программного обеспечения более высоких уровней – прикладных и большинства служебных приложений.

Для того чтобы компьютер мог работать, на его жестком диске должна быть установлена (записана) операционная система. При включении компьютера она считывается с дисковой памяти и размещается в ОЗУ. Этот процесс называется загрузкой операционной системы.

Операционные системы различаются особенностями реализации алгоритмов управления ресурсами компьютера, областями использования.



Так, в зависимости от алгоритма управления процессором, операционные системы делятся на:

- Однозадачные и многозадачные
- Однопользовательские и многопользовательские
- Однопроцессорные и многопроцессорные системы
- Локальные и сетевые.

По числу одновременно выполняемых задач операционные системы делятся на два класса:

- Однозадачные (MS DOS)
- Многозадачные (OS/2, Unix, Windows)

В однозадачных системах используются средства управления периферийными устройствами, средства управления файлами, средства общения с пользователями. Многозадачные ОС используют все средства, которые характерны для однозадачных, и, кроме того, управляют разделением совместно используемых ресурсов: процессор, ОЗУ, файлы и внешние устройства.

В зависимости от областей использования многозадачные ОС подразделяются на три типа:

- Системы пакетной обработки (ОС ЕС)
- Системы с разделением времени (Unix, Linux, Windows)
- Системы реального времени (RT11)

Системы пакетной обработки предназначены для решения задач, которые не требуют быстрого получения результатов. Главной целью ОС пакетной обработки является максимальная пропускная способность или решение максимального числа задач в единицу времени.

Эти системы обеспечивают высокую производительность при обработке больших объемов информации, но снижают эффективность работы пользователя в интерактивном режиме.

В системах с разделением времени для выполнения каждой задачи выделяется небольшой промежуток времени, и ни одна задача не занимает процессор надолго. Если этот промежуток времени выбран минимальным, то создается видимость одновременного выполнения нескольких задач. Эти системы обладают меньшей пропускной способностью, но обеспечивают высокую эффективность работы пользователя в интерактивном режиме.

Системы реального времени применяются для управления технологическим процессом или техническим объектом, например, летательным объектом, станком и т.д.

По числу одновременно работающих пользователей на ЭВМ ОС разделяются на однопользовательские (MS DOS) и многопользовательские (Unix, Linux, Windows 95 - XP)

В многопользовательских ОС каждый пользователь настраивает для себя интерфейс пользователя, т.е. может создать собственные наборы ярлыков, группы программ, задать индивидуальную цветовую схему, переместить в удобное место панель задач и добавить в меню Пуск новые пункты.

В многопользовательских ОС существуют средства защиты информации каждого пользователя от несанкционированного доступа других пользователей.

Многопроцессорные и однопроцессорные операционные системы. Одним из важных свойств ОС является наличие в ней средств поддержки многопроцессорной обработки данных. Такие средства существуют в OS/2, Net Ware, Windows NT. По способу организации вычислительного процесса эти ОС могут быть разделены на асимметричные и симметричные.

Одним из важнейших признаков классификации ЭВМ является разделение их на локальные и сетевые. Локальные ОС применяются на автономных ПК или ПК, которые используются в компьютерных сетях в качестве клиента.

В состав локальных ОС входит клиентская часть ПО для доступа к удаленным ресурсам и услугам. Сетевые ОС предназначены для управления ресурсами ПК включенных в сеть с целью совместного использования ресурсов. Они представляют мощные средства разграничения доступа к информации, ее целостности и другие возможности использования сетевых ресурсов.

### 3. Понятие файловой системы

Все современные ОС обеспечивают создание файловой системы, которая предназначена для хранения данных на дисках и обеспечения доступа к ним.

Основные функции файловой системы можно разделить на две группы:

Функции для работы с файлами (создание, удаление, переименование файлов и т.д.)

Функции для работы с данными, которые хранятся в файлах (запись, чтение, поиск данных и т.д.)

Известно, что файлы используются для организации и хранения данных на машинных носителях. Файл – это последовательность произвольного числа байтов, обладающая уникальным собственным именем или поименованная область на машинных носителях.

Структурирование множества файлов на машинных носителях осуществляется с помощью каталогов, в которых хранятся атрибуты (параметры и реквизиты) файлов. Каталог может включать множество подкаталогов, в результате чего на дисках образуются разветвленные файловые структуры. Организация файлов в виде древовидной структуры называется файловой системой.

Принцип организации файловой системы – табличный. Данные о том, в каком месте на диске записан файл, хранятся в таблице размещения файлов (File Allocation Table, FAT).

Эта таблица размещается в начале тома. В целях защиты тома на нем хранятся две копии FAT. В случае повреждения первой копии FAT дисковые утилиты могут воспользоваться второй копией для восстановления тома.

По принципу построения FAT похожа на оглавление книги, так как операционная система использует ее для поиска файла и определения кластеров, которые этот файл занимает на жестком диске.

Наименьшей физической единицей хранения данных является сектор. Размер сектора 512 байт. Поскольку размер FAT – таблицы ограничен, то для дисков, размер которых превышает 32 Мбайт, обеспечить адресацию к каждому отдельному сектору не представляется возможным.

### 4. Операционная система WINDOWS. Общие сведения о WINDOWS. Интерфейс пользователя WINDOWS. Основы работы в WINDOWS

Операционная система Windows - это современная многозадачная многопользовательская 32 - разрядная ОС с графическим интерфейсом пользователя. Операционные системы семейства Windows являются наиболее распространенными ОС, которые установлены в домашних и офисных ПК.

Графическая оболочка ОС Windows обеспечивает взаимодействие пользователя с компьютером в форме диалога с использованием ввода и вывода на экран дисплея графической информации, управления программами с помощью пиктограмм, меню, окон, панелей (управления, задач, инструментов) и других элементов управления.

Основными элементами графического интерфейса Windows являются: Рабочий стол, Панель задач с кнопкой Пуск. Так как в Windows применен графический пользовательский интерфейс, то основным устройством управления программами является манипулятор мышь.

Основными элементами графического интерфейса Windows являются:

- Рабочий стол с пиктограммами
- Панель задач, на которой размещаются программные кнопки, индикаторы, Панель быстрого запуска
- Главное меню (кнопка Пуск)
- Контекстное меню (отображается при щелчке правой кнопкой мыши по выбранному объекту)

В ОС Windows применяются четыре типа меню (меню – это список команд, выводимых на экран и предлагаемых пользователю для выбора):

- Главное меню (открывается кнопкой Пуск)
- Строка меню в окнах приложения (все программы, входящие в стандартный пакет поставки Windows, имеют строку меню)
- Системное меню в окнах приложения (для изменения размеров окна и его положения)
- Контекстное меню

#### Работа с файлами

Все файлы, документы и программы в Windows хранятся в папках. В электронной папке, как правило, хранят файлы, сгруппированные по какому-либо признаку, типу и другие папки.

Папка – это контейнер для программ и файлов в графических интерфейсах пользователя, отображаемый на экране с помощью значка, имеющего вид канцелярской папки. Windows предоставляет средства для управления файлами и папками.

К таким средствам относятся программа Проводник и окно Мой компьютер. Приложение Проводник является главным инструментом Windows для просмотра файлов и папок, хранящихся на жестких и гибких дисках и других носителях информации.

Проводник отображает иерархическую структуру файлов, папок и дисков на ПК. В левой части проводника Windows использует иерархическое представление папок, файлов и других ресурсов, подключенных к компьютеру или сети.

Мой компьютер – программа, используемая для работы с файлами и папками, хранящимися на дисках компьютера. Мое сетевое окружение – программа, используемая для работы с сетевыми ресурсами в рабочей группе.

### 3.1. Лекция № 3 (2 часа)

Тема: Алгоритмизация вычислительных процессов

#### 3.1.1. Вопросы лекции:

1. Понятие алгоритмизации вычислительных процессов
2. Понятие алгоритма и его свойства, способы описания алгоритмов
3. Основные типы алгоритмов
  - 3.1. Алгоритм линейной структуры
  - 3.2. Алгоритм ветвящейся структуры
- 3.1.2. Краткое содержание вопросов
  1. Понятие алгоритмизации вычислительных процессов

Разработке алгоритма предшествуют такие этапы, как формализация и моделирование задачи. Формализация предполагает замену словесной формулировки решаемой задачи краткими символьными обозначениями, близкими к обозначениям в языках программирования или к математическим. Моделирование задачи является важнейшим этапом, целью которого является поиск общей концепции решения. Обычно моделирование выполняется путем выдвижения гипотез решения задачи и их проверке любым рациональным способом (прикидочные расчеты, физическое моделирование и т.д.). Результатом каждой проверки является либо принятие гипотезы, либо отказ от нее и разработка новой.

При разработке алгоритма используют следующие основные принципы.

Принцип поэтапной детализации алгоритма (другое название - "проектирование сверху-вниз"). Этот принцип предполагает первоначальную разработку алгоритма в виде укрупненных блоков (разбиение задачи на подзадачи) и их постепенную детализацию.

Принцип "от главного к второстепенному", предполагающий составление алгоритма, начиная с главной конструкции. При этом, часто, приходится "достраивать" алгоритм в обратную сторону, например, от середины к началу.

Принцип структурирования, т.е. использования только типовых алгоритмических структур при построении алгоритма. Нетиповой структурой считается, например, циклическая конструкция, содержащая в теле цикла дополнительные выходы из цикла. В программировании нетиповые структуры появляются в результате злоупотребления командой безусловного перехода (GoTo). При этом программа хуже читается и труднее отлаживается.

Говоря о блок-схемах, как о средстве записи алгоритма, можно дать еще один совет по их разработке. Рекомендуется после внесения исправлений в блок-схему аккуратно перерисовывать ее с учетом этих исправлений. Аккуратность записи есть аккуратность мысли программиста. Аккуратно записанный и детализованный алгоритм упрощает его программирование и отладку.

Этапы решения задачи на компьютере.

Решение задачи разбивается на этапы:

1. Постановка задачи
2. Формализация (математическая постановка)
3. Выбор (или разработка) метода решения
4. Разработка алгоритма
5. Составление программы
6. Отладка программы
7. Вычисление и обработка результатов

1. При постановке задачи выясняется конечная цель и вырабатывается общий подход к решению задачи. Выясняется сколько решений имеет задача и имеет ли их вообще. Изучаются общие свойства рассматриваемого физического явления или объекта, анализируются возможности данной системы программирования.

2. На этом этапе все объекты задачи описываются на языке математики, выбирается форма хранения данных, составляются все необходимые формулы.

3. Выбор существующего или разработка нового метода решения (очень важен и, в то же время личностный этап).

4. На этом этапе метод решения записывается применительно к данной задаче на одном из алгоритмических языков (чаще на графическом).

5. Переводим решение задачи на язык, понятный машине.

2. Понятие алгоритма и его свойства, способы описания алгоритмов

"Алгоритм" является фундаментальным понятием информатики. Представление о нем необходимо для эффективного применения вычислительной техники к решению практических задач. Алгоритм - это предписание исполнителю (человеку или автомату) выполнить точно определенную последовательность действий, направленных на достижение заданной цели. Алгоритм - это сформулированное на некотором языке правило, указывающее на действия, последовательное выполнение которых приводит от исходных данных к искомому результату. Значение слова алгоритм очень схоже со значением слов рецепт, процесс, метод, способ. Однако любой алгоритм, в отличие от рецепта или способа, обязательно обладает следующими свойствами.

Свойства алгоритма (отличающие его от любых других предписаний): понятность (для конкретного исполнителя); дискретность (команды последовательны, с точной фиксацией моментов начала и конца выполнения команды); точность (после выполнения каждой команды точно известно, завершено ли исполнение алгоритма или же какая команда должна выполняться следующей); результативность (после конечного числа шагов задача решается или же становится ясно, что процесс решения не может быть продолжен); массовость (алгоритм единым образом применяется к любой конкретной формулировке задачи, для которой он разработан).

1. Дискретность - разбиение алгоритма на ряд отдельных законченных действий - шагов. Выполнение алгоритма разбивается на последовательность законченных действий - шагов. Каждое действие должно быть закончено исполнителем алгоритма прежде, чем он приступит к исполнению следующего действия.

2. Точность - однозначные указания. На каждом шаге однозначно определено преобразование объектов среды исполнителя, полученной на предыдущих шагах алгоритма. Если алгоритм многократно применяется к одному и тому же набору исходных данных, то на выходе он получает каждый раз один и тот же результат. Запись алгоритма должна быть такой, чтобы на каждом шаге его выполнения было известно, какую команду надо выполнять следующей.

3. Понятность - однозначное понимание и исполнение каждого шага алгоритма его исполнителем. Алгоритм должен быть записан на понятном для исполнителя языке.

4. Результативность - обязательное получение результата за конечное число шагов. Каждый шаг (и алгоритм в целом) после своего завершения дает среду, в которой все объекты однозначно определены. Если это по каким-либо причинам невозможно, то алгоритм должен сообщать, что решение задачи не существует. Работа алгоритма должна быть завершена за конечное число шагов. Информатика оперирует только с конечными объектами и конечными процессами, поэтому вопрос о рассмотрении бесконечных алгоритмов остается за рамками теории алгоритмов.

5. Массовость - применение алгоритма к решению целого класса однотипных задач.

Порядок выполнения алгоритма:

- Действия в алгоритме выполняются в порядке их записи
- Нельзя менять местами никакие два действия алгоритма
- Нельзя не закончив одного действия переходить к следующему

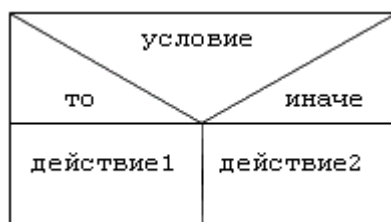
Для записи алгоритмов используются специальные языки:

1. Естественный язык (словесная запись)
2. Формулы
3. Псевдокод
4. Структурограммы
5. Синтаксические диаграммы
6. Графический (язык блок-схем)

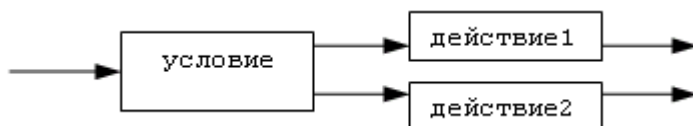
1. Естественный язык:

если условие то действие1 иначе действие2

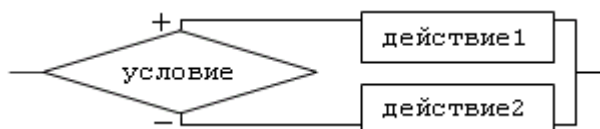
2. Структурограмма:



### 3. Синтаксическая диаграмма:



### 4. Графический язык:



Составление алгоритмов графическим способом подчиняется двум ГОСТам:

1. ГОСТ 19.002-80, соответствует международному стандарту ИСО 2636-73. Регламентирует правила составления блок-схем.
2. ГОСТ 19.003-80, соответствует международному стандарту ИСО 1028-73. Регламентирует использование графических примитивов.

Название	Символ (рисунок)	Выполняемая функция (пояснение)
1. Блок вычислений		Выполняет вычислительное действие или группу действий
2. Логический блок		Выбор направления выполнения алгоритма в зависимости от условия
3. Блоки ввода/вывода		Ввод или вывод данных вне зависимости от физического носителя
		Вывод данных на печатающее устройство
4. Начало/конец (вход/выход)		Начало или конец программы, вход или выход в подпрограмму
5. Предопределенный процесс		Вычисления по стандартной или пользовательской подпрограмме
6. Блок модификации		Выполнение действий, изменяющих пункты алгоритма
7. Соединитель		Указание связи между прерванными линиями в пределах одной страницы
8. Межстраничный соединитель		Указание связи между частями схемы, расположенной на разных страницах

### Правила построения блок-схем:

Блок-схема выстраивается в одном направлении либо сверху вниз, либо слева направо

Все повороты соединительных линий выполняются под углом 90 градусов

### 3. Основные типы алгоритмов

#### 3.1. Алгоритм линейной структуры

**Линейным** называется **вычислительный процесс**, в котором предусматривается получение результата путем однократного выполнения последовательности действий при любых значениях исходных данных. Характерной особенностью линейного вычислительного процесса является то, что направление вычислений не зависит от исходных данных и промежуточных результатов.

Алгоритм линейного вычислительного процесса графически может быть представлен **блоком следования – композицией** (объединением) нескольких следующих друг за другом блоков ПРОЦЕСС (рис. 1).

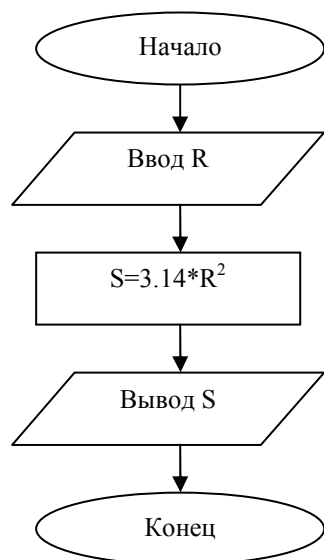
Рисунок 1- Блок-схема линейного алгоритма

Линии, соединяющие отдельные блоки на блок-схеме алгоритма и указывающие на последовательность действий, называются **линиями потока**. Направление линии потока сверху вниз принимается за основное и стрелкой не обозначается.

Рассмотрим пример задачи, решение которой представляет собой линейный вычислительный процесс, и составим блок-схему алгоритма.

Пример: Найдите площадь круга  $S$  при заданном значении радиуса  $R$ . Для вычисления площади использовать формулу  $S=3,14 \cdot R^2$ .

Блок-схема



Как правило, вычислительного "чистом виде" такие редко. Чаще всего вычислительных

- линейный
- блок-схема

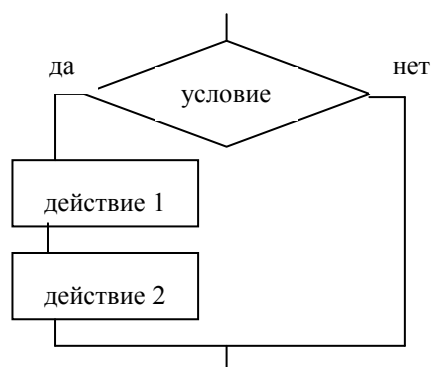
простым при  
процесса  
(объединение)  
ПРОЦЕСС.



составление алгоритма линейного процесса не вызывает затруднений, однако в вычислительные процессы встречаются весьма они являются составной частью более сложных процессов. Таким образом: вычислительный процесс является наиболее реализации его на ЭВМ. алгоритма линейного вычислительного представляет собой **композицию** нескольких следующих друг за другом блоков

### 3.2. Алгоритм ветвящейся структуры

Разветвляющимся вычислительным процессом называется процесс, направление вычислений в котором зависит от результата проверки некоторого условия (условий).



Алгоритм разветвляющегося вычислительного процесса предусматривает **выбор** одной из нескольких возможных **альтернатив** (последовательностей действий) в зависимости от значения исходных данных или промежуточных результатов. Каждую из этих последовательностей называют **ветвью алгоритма**. Блок-схема алгоритма разветвляющегося вычислительного процесса содержит, по крайней мере, один блок РЕШЕНИЕ. Направления линий потока сверху вниз и слева направо на блок-схеме принимаются за основные и стрелками не обозначаются.

Графической интерпретацией алгоритма разветвляющегося вычислительного процесса является **блок разветвления алгоритма**, в котором может быть предусмотрен **полный** (рис. 2) или **неполный** (рис. 3) **выбор**.

В блоке разветвления алгоритма с полным выбором в зависимости от результата проверки условия выполняются только действия ветви "да" (т. е., действия 1 и 2) или только действия ветви "нет" (действия 3 и 4). В блоке с неполным выбором в зависимости от результата проверки условия либо выполняются действия какой-либо ветви, либо они игнорируются.

Некоторой разновидностью блока разветвления алгоритма является **блок множественного выбора** (рис. 3). В нем, в зависимости от результатов выбора, выполняется одно из предусмотренных действий.

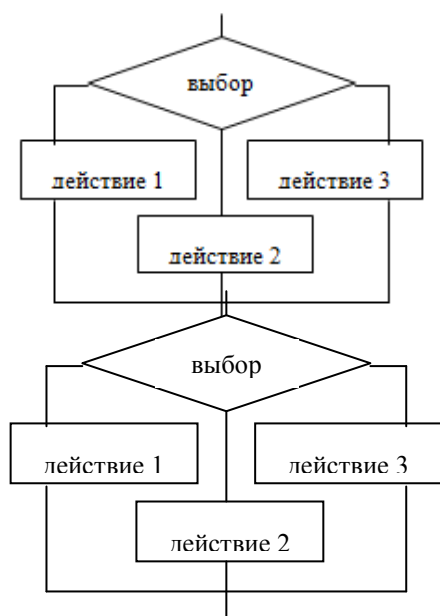


Рисунок 3— Блок множественного выбора

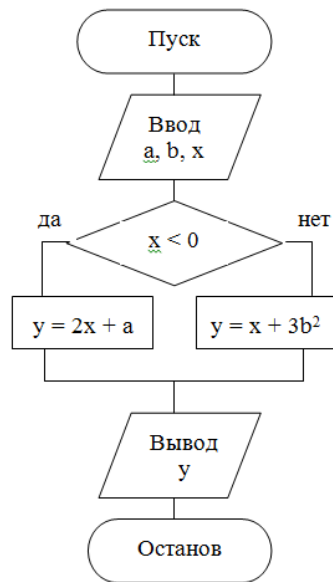
Рассмотрим пример задачи, алгоритм решения которой носит разветвляющийся характер.

Пример 2. Составить схему алгоритма для вычисления значений функции

$$y = \begin{cases} 2x + a, & \text{если } x < 0, \\ x + 3b^2, & \text{если } x \geq 0. \end{cases}$$



Поскольку значение функции вычисляется по разным формулам в зависимости от знака аргумента  $x$ , в алгоритме вычислительного процесса должна быть предусмотрена проверка знака названного аргумента. Блок-схема алгоритма решения задачи на ЭВМ представлена на рис. 4.



**Рисунок 4 – Блок-схема алгоритма**

В разветвляющихся вычислительных процессах могут иметь место так называемые "пустые" ветви. В направлении пустых ветвей задача, как правило, не имеет решения (вычислительный процесс не приводит к получению какого-либо промежуточного или конечного результата). Рассмотрим следующий пример.

Пример 3. Составить блок-схему алгоритма для вычисления значений функции

$$y = \begin{cases} x - a, & \text{если } x = 0, \\ x + b^2, & \text{если } 0 < x < 3, \\ x^2 + c, & \text{если } 3 < x < 5. \end{cases}$$

Как видно из условия задачи, переменная  $x$  может принимать только значения, находящиеся в диапазоне от 0 до 5. Следовательно, для остальных значений  $x$  ветви в алгоритме оказываются пустыми. В блок-схеме алгоритма должны быть предусмотрены действия ЭВМ на случай появления исходных данных (значений  $x$ ), не описанных логическими условиями. На рис. 6 показан один из возможных способов решения проблемы.

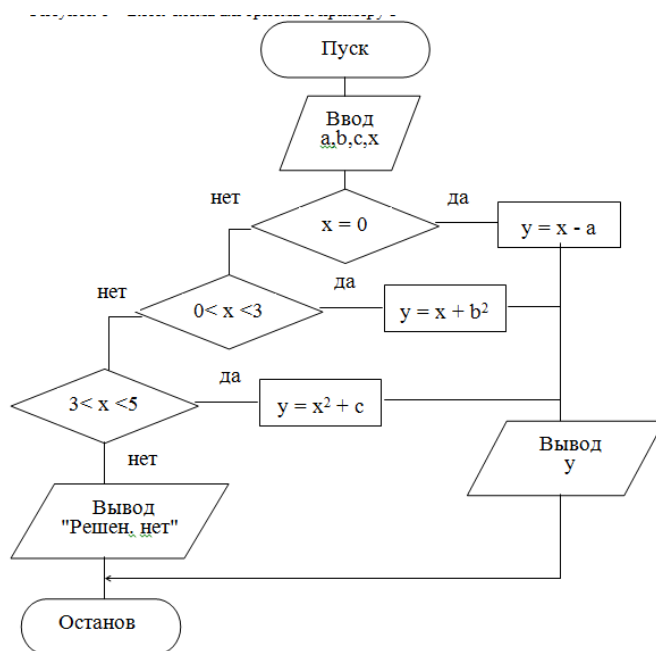


Рисунок 5 – Блок-схема алгоритма к примеру 3

В заключение кратко подведем итог:

1. Алгоритм разветвляющегося вычислительного процесса предусматривает выбор одной из нескольких возможных альтернатив в зависимости от значения исходных данных или промежуточных результатов.
2. В блоке разветвления алгоритма может быть предусмотрен полный или неполный выбор.
3. В разветвляющихся вычислительных процессах могут иметь место "пустые" ветви.

### 3.3. Алгоритм циклической структуры

**Циклическим** называется вычислительный процесс, в котором получение результата обеспечивается путем **многократного повторения некоторой последовательности действий**.

Графической интерпретацией алгоритма циклического вычислительного процесса является **блок цикла**. Различают несколько разновидностей блока цикла: **блок цикла с параметром**, **блок цикла с предварительным условием** и **блок цикла с последующим условием**.

Блок-схема **блока цикла с параметром** представлена на рис. 6.

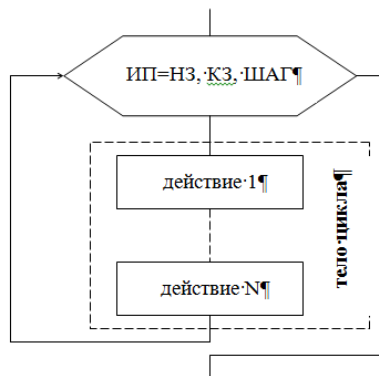


Рисунок 6 – Блок цикла с параметром

На рис. 6 приняты следующие сокращения:

ИП – имя ячейки памяти, в которую заносится значение параметра;

НЗ – начальное значение параметра;

КЗ – конечное значение параметра;

ШАГ – величина приращения параметра после каждого выполнения тела цикла.

Тело цикла представляет собой линейный вычислительный процесс и выполняется столько раз, сколько разных значений примет параметр в заданных пределах от НЗ до КЗ. Блок цикла с параметром относится к циклу с **явно выраженным числом повторений** (число повторений известно заранее). Для таких циклов характерным является то, что задаются:

- **начальное и конечное значения параметра** цикла;
- **закон изменения параметра цикла** при каждом повторном выполнении тела цикла;
- **количество повторных выполнений** тела цикла (вытекает из первых двух пунктов).

Блок цикла с предварительным условием и блок цикла с последующим условием относятся к так называемым **итерационным циклам**. В таких циклических вычислительных процессах число повторений тела цикла заранее не известно. Выход из цикла осуществляется не после того, как цикл повторится заданное число раз, а при выполнении определенного условия, связанного с проверкой значения монотонно изменяющейся в теле цикла величины. Блок-схема блока цикла с предварительным условием представлена на рис. 8, а блока цикла с последующим условием – на рис. 7.

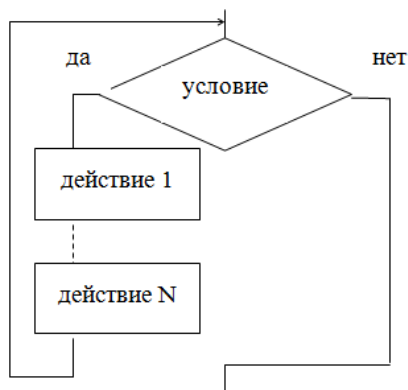


Рисунок 7 – Блок цикла с предварительным условием

Кратко суть алгоритма цикла с предварительным условием можно изложить следующим образом: **пока выполняется условие – повторять действия**. В таких циклах возможны ситуации, когда тело цикла не выполняется ни разу (например, если при первой же проверке не выполняется условие, то сразу происходит выход из цикла).

В цикле с последующим условием (рис. 8) тело цикла выполняется не менее одного раза. При этом **действия**, предусмотренные в теле цикла, **выполняются до тех пор, пока не выполнится заданное условие**.

Рассмотренные блоки циклов позволяют описать **простые** циклические вычислительные процессы. При решении сложных задач может возникнуть необходимость внутри одного цикла организовать дополнительно один или несколько циклов. Такие циклы называются **вложенными**. При этом цикл, внутри которого создается другой цикл, называется **внешним**, а цикл, создаваемый внутри другого – **внутренним**. Правила организации как внешнего, так и внутреннего циклов те же, что и для простых циклов. Параметры внешнего и внутреннего циклов должны быть разными.

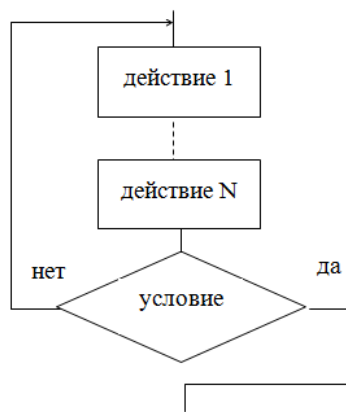


Рисунок 8 - Блок цикла с последующим условием

#### 4.1. Лекция № 4 (4 часа)

Тема: Программирование на алгоритмическом языке высокого уровня

##### 4.1.1. Вопросы лекции:

1. Основные понятия алгоритмического языка QBASIC. История развития QBASIC.

Операции над числами и данными

2. Встроенные функции

3. Арифметические выражения

4. Основные операторы алгоритмического языка QBASIC

4.1. Оператор присваивания LET

4.2. Операторы ввода данных READ, DATA, RESTORE, INPUT

4.3. Операторы вывода данных PRINT, LPRINT

4.4. Условные операторы IF ... THEN, GOTO, SELECT CASE

4.5. Операторы организации цикла FOR ... NEXT, WHILE ... WEND, DO ... LOOP

4.6. Операторы: REM, CLS, STOP, END

##### 4.1.2. Краткое содержание вопросов

1. Основные понятия алгоритмического языка QBASIC. История развития QBASIC.

Операции над числами и данными

Бейсик (от BASIC, сокращение от англ. Beginner's All-purpose Symbolic Instruction Code — универсальный код символических инструкций для начинающих; англ. basic — основной, базовый) — семейство высокоуровневых языков программирования.

Был разработан в 1963 году профессорами Дартмутского колледжа Томасом Курцем и Джоном Кемени.

Язык предназначался для обучения программированию и получил широкое распространение в виде различных диалектов, прежде всего, как язык для домашних компьютеров.

В 1975 году, Пол Аллен, молодой программист из Бостона, в содружестве со студентом Гарвардского университета Биллом Гейтсом написали программу, реализующую для микрокомпьютера Альтаир 8800 язык Бейсик, впервые использовав его для программного обеспечения персональных компьютеров. Впоследствии Гейтс и Аллен основали собственную фирму Microsoft.[1]

Алгоритмический язык Basic используется преимущественно в режиме диалога человека и ЭВМ. Этот язык ориентирован на решение различных задач вычислительного и невычислительного характера с небольшим объемом исходной информации. Название языка BASIC возникло от сокращения английских слов Beginner's All-purpose Symbolic Instruction Code (многоцелевой язык символических инструкций для начинающих). Следует отметить, что стандарта на язык Basic не существует и различные его модификации могут существенно отличаться друг от друга.

В языке Basic существуют как средства для описания действий алгоритма, которые используются при составлении программ, - операторы Basic, так и средства, которые служат для общения с ЭВМ. Последние имеют форму приказов для немедленного выполнения и называются командами.

Основным режимом в Basic является программный режим, когда заранее составленная программа полностью вводится в ЭВМ и затем выполняется.

Программа на Basic состоит из строк, которые могут иметь номер. В одной строке может содержаться один или несколько операторов, разделенных символом : (двоеточие). Обычно строки нумеруются, начиная с 10, с шагом 10. Номера строк используются в операторах передачи управления. При этом оператор, которому передается управление, должен быть первым в строке.

## 2. Встроенные функции

Для вычисления наиболее распространенных элементарных математических функций в языке БЕЙСИК применяют встроенные стандартные функции. Аргумент функции заключается в круглые скобки. Аргументом функции может быть произвольное арифметическое выражение. Для обращения к функции нужно набрать ее имя и указать аргумент. Стандартные математические функции, применяемые в языке Бейсик, приведены в ТАБЛИЦЕ 1.

Аргументы тригонометрических функций должны быть заданы в радианах. Если угол выражен в градусах, то его нужно перевести в радианы по формуле

**Значение в радианах = значение в градусах \* 3.14/180**

В некоторых версиях языка Бейсик имеется функция LOG(X) для вычисления натурального логарифма числа X. Однако от логарифма с любым основанием легко перейти к натуральному логарифму, используя формулу

**$\text{Log}_a N = \ln N / \ln a$ , где a – основание.**

Над числовыми константами, переменными и стандартными функциями можно производить обычные арифметические операции:

1. вычисляются значения функций
2. ^ возведение в степень,
3. \* умножение, / деление,
4. + сложение, - вычитание.

ТАБЛИЦА 1

Обозначение функции на языке Бейсик	Название	Пояснение
ABS(X)	Функция абсолютная величина  x  (модуль)	Функция вычисляет абсолютное значение (модуль) аргумента X
SQR(X)	Функция квадратный корень ( $\sqrt{x}$ )	Функция вычисляет квадратный корень положительного аргумента X
LOG(X)	Функция натуральный логарифм $\ln x$	Функция вычисляет натуральный логарифм положительного аргумента X
EXP(X)	Экспоненциальная функция $e^x$	Вычисляет $e^x$ , где $e=2,71828$ , X-любое число
COS(X)	Функция косинуса $\cos x$	Вычисляет косинус аргумента X
SIN(X)	Функция синуса $\sin x$	Вычисляет синус аргумента X

TAN(X)	Функция тангенса tg x	Вычисляет тангенс аргумента X, находящегося в диапазоне от $-\frac{\pi}{2}$ до $\frac{\pi}{2}$
ATN(X)	Функция арктангенса arctg x	Вычисляет арктангенс аргумента X
INT(X)	Целочисленная функция	Определяет наибольшее целое число, не превосходящее X
RND(X)	Функция случайных чисел	Выдает случайное число, лежащее в интервале от 0 до 1. Значение аргумента X игнорируется

### 3. Арифметические выражения

**Арифметическое выражение** – это символическая запись, указывающая правила вычисления числового выражения. Оно состоит из чисел (констант), имен переменных, функций, знаков арифметических операций и скобок.

**При записи выражений необходимо учитывать следующие рекомендации и ограничения:**

1. Формулу требуется записывать в строку без каких-либо подстрочных и надстрочных знаков.

2. Следует использовать круглые скобки для указания порядка действий, особенно в сомнительных случаях. Вычисления в скобках производятся в первую очередь. Если выражение, содержащее скобки само заключено в скобки, то вычисления производятся, начиная с внутренних скобок. Внутри скобок действия выполняются с лева на право в соответствии с приоритетом операций:

- Сначала вычисляются значения функций,
- Затем - выполняются все операции возведения в степень,
- Затем - умножения и деления,
- И наконец, сложения и вычитания.

Если математическое выражение содержит несколько операций одинакового старшинства, то такие операции выполняются последовательно слева на право.

3. Два знака арифметических операций нельзя ставить рядом, а также нельзя опускать знак умножения между сомножителями.

4. Выполнение арифметических операций над арифметическими выражениями одного типа дает результат того же типа. Операция над целой и вещественной величиной дает вещественный результат.

5. Возведение в целую степень выполняется многократным умножением или с помощью операции ^ возведения в степень.

6. Вычисление результата возведения в степень осуществляется с помощью функций EXP и LOG, если показатель степени - действительное число:

$$x^y = \text{EXP}(Y * \text{LOG}(X))$$

Так как логарифмы отрицательных чисел не существуют, то нельзя возводить отрицательные числа в действительную степень.

#### 3.4. Основные операторы алгоритмического языка QBASIC

В результате изучения студент должен знать:

- Понятие оператора;
- Оператор присваивания в Бейсике;
- Операторы задания значений переменным в Бейсике;

- Оператор вывода в Бейсике;
- Оператор ввода в Бейсике;
- Оператор комментариев в Бейсике;
- Оператор очищения экрана;
- Оператор конца текста программы;
- Оператор безусловного перехода;
- Оператор условного перехода в полной форме;
- Оператор условного перехода в неполной форме.

уметь:

- Приводить примеры использования операторов языка программирования Бейсик;
- Использовать операторы языка программирования Бейсик при составлении простейших программ.

Оператор – это предписание ЭВМ, написанное на языке БЕЙСИК. Операторы можно разделить на две группы:

1. Выполняемые операторы – определяют действие программы, указывая Бейсик-системе, какую операцию нужно выполнить (PRINT – печатать, INPUT - ввести);
2. Невыполняемые операторы – описывают характер и упорядочение данных, позволяют вводить в программу примечания и сообщения описательного характера (DATA, REM).

## 5.1. Лекция № 5 (2 часа)

Тема: Компьютерные сети

### 5.1.1. Вопросы лекции:

1. Основные понятия. Классификация компьютерных сетей
2. Локальные вычислительные сети
3. Глобальная вычислительная сеть
4. Требования, предъявляемые к современным вычислительным сетям

### 5.1.2. Краткое содержание вопросов

1. Основные понятия. Классификация компьютерных сетей

По мере того, как в информационные процессы вовлекалось все больше пользователей, как расширялся круг задач, решаемых с помощью ЭВМ, возникла необходимость в децентрализации процессов обработки данных. Принцип **централизованной** обработки данных (рисунок 1), когда к одной большой ЭВМ подключалось несколько терминалов (рабочих мест, состоящих из дисплея и клавиатуры), не отвечал высоким требованиям к надежности процесса обработки, затруднял развитие систем и не мог обеспечить необходимые временные параметры при диалоговой обработке данных в многопользовательском режиме. Кратковременный выход из строя центральной ЭВМ приводил к роковым последствиям для системы в целом. Поэтому приходилось дублировать функции центральной ЭВМ, значительно увеличивая затраты на создание и эксплуатацию систем обработки данных.

**Распределенная обработка данных – обработка данных, выполняемая на независимых, но связанных между собой ЭВМ, представляющих распределенную систему.**

Для реализации распределенной обработки данных были созданы **многомашинные ассоциации**, структура которых разрабатывается по одному из следующих направлений:

- многомашинные вычислительные комплексы (МВК);

– компьютерные (вычислительные) сети.

**Многомашинный вычислительный комплекс** – это группа установленных рядом вычислительных машин, объединенных с помощью специальных средств сопряжения и выполняющих совместно единый информационно-вычислительный процесс.

Многомашинные вычислительные комплексы могут быть:

- **локальными**, при условии установки ЭВМ в одном помещении;
- **дистанционными**, если некоторые ЭВМ комплекса установлены на значительном расстоянии от центральной ЭВМ и для передачи данных используются телефонные каналы связи.

С появлением персональных ЭВМ их также стали объединять в сети.

**Компьютерная (вычислительная) сеть** – это совокупность компьютеров, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

**Глобальная** вычислительная сеть объединяет абонентов, расположенных в различных странах, на различных континентах. Взаимодействие между абонентами такой сети может осуществляться на базе телефонных линий связи, радиосвязи и систем спутниковой связи. Глобальные вычислительные сети позволяют решить проблему объединения информационных ресурсов всего человечества и организации доступа к этим ресурсам. Примером глобальной вычислительной сети служит **всемирная сеть Internet**.

**Региональная** вычислительная сеть связывает абонентов, расположенных на значительном расстоянии друг от друга. Она может включать абонентов внутри большого города, экономического региона, отдельной страны. Обычно в этом случае расстояние между абонентами вычислительной сети составляет десятки – сотни километров.

**Локальная** вычислительная сеть объединяет абонентов, расположенных в пределах небольшой территории. В настоящее время не существует четких ограничений на территориальный разброс абонентов локальной вычислительной сети. Обычно такая сеть привязана к конкретному месту. К классу локальных вычислительных сетей относятся сети отдельных предприятий, фирм, банков, офисов и т. д. Протяженность такой сети можно ограничить пределами 2 ... 2,5 км.

## 2. Локальные вычислительные сети

Назначение всех видов компьютерных ЛВС определяется, в основном, двумя функциями:

**обеспечение совместного использования аппаратных и программных ресурсов сети;**

**обеспечение совместного доступа к ресурсам данных.**

Локальные вычислительные сети объединяют относительно небольшое число ПЭВМ (обычно от 10 до 100) в пределах небольшой территории (одно помещение, здание или учреждение). Традиционное название – локальная вычислительная сеть – скорее дань тем временам, когда сети использовались в основном для решения вычислительных задач. Сегодня же в 99% случаев речь идет исключительно об обмене информацией в виде текстов, графических и видео-образов, числовых массивов.

Основными техническими средствами ЛВС являются **рабочие станции** (обычные ПЭВМ), которые могут быть подключены к более мощному компьютеру – **сетевому серверу**.

**Сервер** – это мощный компьютер, способный одновременно обрабатывать сотни (тысячи) запросов на обслуживание, поступающих от рабочих станций.

**Рабочей станцией** (или **сетевым узлом**) называется ПЭВМ, используемая в качестве клиента, то есть потребителя сетевых услуг, предоставляемых сервером.

## 3. Глобальная вычислительная сеть



Зарождение Интернета принято считать с момента появления первой компьютерной сети, родиной которой в середине 60-х годов двадцатого века стала Америка.

В то время еще не существовало персональных компьютеров, и крупные американские университеты могли себе позволить 1-2 больших компьютера. Компьютерное время было драгоценным ресурсом, и на него заранее записывались. Люди работали ночами, чтобы ни минуты этого времени не пропало даром.

Наконец появилась идея соединить между собой компьютеры разных университетов, чтобы сделать возможным удаленное использование любого свободного в данный момент компьютера. Этот проект получил название ARPANET. К концу 1969 года были соединены компьютеры четырех университетов и появилась первая компьютерная сеть.

Очень скоро обнаружилось, что сеть в основном используется не для вычислений на удаленном компьютере, а для обмена сообщениями между пользователями. В 1972 году, когда ARPANET уже соединял 23 компьютера, была написана первая программа для обмена электронной почтой по сети. Электронную почту оценили по достоинству, что побудило целый ряд государственных организаций и корпораций к созданию собственных компьютерных сетей. Эти сети обладали тем же недостатком, что и ARPANET: они могли соединять только ограниченное число однотипных компьютеров. Кроме того, они были не совместимы друг с другом.

В середине 70-х годов для ARPANET были разработаны новые стандарты передачи данных, которые позволяли объединять между собой сети произвольной архитектуры, тогда же было придумано слово "Интернет". Именно эти стандарты, впоследствии получившие название протокола TCP/IP, заложили основу для роста глобальной компьютерной сети путем объединения уже существующих сетей. Их важным достоинством было то, что сеть считалась в принципе не стопроцентно надежной, и предусматривались средства борьбы с ошибками при передаче данных. В 1983 году сеть ARPANET перешла на новый протокол и разделилась на две независимые сети - военную и образовательную. К этому времени сеть объединяла более тысячи компьютеров, в том числе в Европе и на Гавайских островах. Последние использовали спутниковые каналы связи.

Развитие Интернета получило новый импульс благодаря инициативе Национального научного фонда США (NSF) по созданию глобальной сетевой инфраструктуры для системы высшего образования (1985-88). NSF создал сеть скоростных магистральных каналов связи и выделял средства на подключение к ней американских университетов, при условии, что университет обеспечивал доступ к сети для всех подготовленных пользователей. Интернет оставался преимущественно университетской сетью до начала 90-х годов, однако NSF сразу взял курс на то, чтобы сделать его в дальнейшем независимым от государственного финансирования. В частности, NSF поощрял университеты к поиску коммерческих клиентов. К 1988 году Интернет уже насчитывал около 56 тысяч соединенных компьютеров.

Настоящий расцвет Интернета начался в 1992 году, когда была изобретена новая служба, получившая странное название «Всемирная паутина» (World Wide Web, или WWW, или просто «веб»). WWW позволял любому пользователю Интернета публиковать свои текстовые и графические материалы в привлекательной форме, связывая их с публикациями других авторов и предоставляя удобную систему навигации. Постепенно Интернет начал выходить за рамки академических институтов и стал превращаться из средства переписки и обмена файлами в гигантское хранилище информации. К 1992 году Интернет насчитывал более миллиона соединенных компьютеров.

В настоящее время Интернет продолжает расти с прежней головокружительной скоростью. По оценке специалистов, количество передаваемой информации (трафик) в Интернете увеличивается на 30% ежемесячно. В 1999 году Интернет объединял около 60

миллионов компьютеров и более 275 миллионов пользователей, и каждый день в нем появлялось полтора миллиона новых веб-документов. Эти оценки довольно приблизительны, потому что в Интернете нет центрального административного органа, который регистрировал бы новых пользователей и новые компьютеры.

В Россию Интернет впервые проник в начале 90-х годов. Ряд университетов и исследовательских институтов приступили в это время к построению своих компьютерных сетей и обзавелись зарубежными каналами связи. Особенно следует отметить Институт Атомной Энергии им. Курчатова. На базе ИАЭ сложились две крупнейшие коммерческие компании, предоставляющие услуги по подключению к Интернету – «Релком» и «Демос», а также Российский Институт Развития Общественных Сетей (РОСНИИРОС). Последний стал в дальнейшем головной организацией, координирующей развитие российской части Интернета.

#### 4. Требования, предъявляемые к современным вычислительным сетям

##### *Производительность*

Определяется такими показателями: время реакции системы - время между моментом возникновения запроса и моментом получения ответа. Пропускная способность сети определяется количеством информации, переданной через сеть или ее сегмент в единицу времени. Определяется в битах в секунду.

##### *Надежность*

Определяется надежностью работы всех ее компонентов. Для повышения надежности работы аппаратных компонентов обычно используют дублирование, когда при отказе одного из элементов функционирование сети обеспечат другие.

При работе компьютерной сети должна обеспечиваться *сохранность* информации и защита ее от искажений. Как правило, важная информация в сети хранится в нескольких экземплярах. В этом случае необходимо обеспечить согласованность данных (например, идентичность копий при изменении информации).

Одной из функций компьютерной сети является передача информации, во время которой возможны ее потери и искажения. Для оценки надежности исполнения этой функции используются показатели вероятности потери пакета при его передаче, либо вероятности доставки пакета (передача осуществляется порциями, которые называются пакетами).

В современных компьютерных сетях важное значение имеет другая сторона надежности - *безопасность*. Это способность сети обеспечить защиту информации от несанкционированного доступа. Задачи обеспечения безопасности решаются применением как специального программного обеспечения, так и соответствующих аппаратных средств.

##### *Управляемость*

При работе компьютерной сети, которая объединяет отдельные компьютеры в единое целое, необходимы средства не только для наблюдения за работой сети, сбора разнообразной информации о функционировании сети, но и средства управления сетью. В общем случае система управления сетью должна предоставлять возможность воздействовать на работу любого элемента сети. Должна быть обеспечена возможность осуществлять мероприятия по управлению с любого элемента сети. Управлением сетью занимается администратор сети или пользователь, которому поручены эти функции. Обычный пользователь, как правило, не имеет административных прав.

Другими характеристиками управляемости являются возможность определения проблем в работе компьютерной сети или отдельных ее сегментов, выработка управленческих действий для решения выявленных проблем и возможность автоматизации этих процессов при решении похожих проблем в будущем.

##### *Расширяемость и масштабируемость*

Любая компьютерная сеть является развивающимся объектом, и не только в плане модернизации ее элементов, но и в плане ее физического расширения, добавления новых

элементов сети (пользователей, компьютеров, служб). Существование таких возможностей, трудоемкость их осуществления входят в понятие расширяемости. Другой похожей характеристикой является масштабируемость сети, которая определяет возможность расширения сети без существенного снижения ее производительности. Обычно одноранговые сети обладают хорошей расширяемостью, но плохой масштабируемостью. В таких сетях легко добавить новый компьютер, используя дополнительный кабель и сетевой адаптер, но существуют ограничения на количество подключаемых компьютеров в связи с существенным падением производительности сети. В многосегментных сетях используются специальные коммуникационные устройства, которые позволяют подключать к сети значительное количество дополнительных компьютеров без снижения общей производительности сети.

#### *Прозрачность*

Прозрачность компьютерной сети является ее характеристикой с точки зрения пользователя. Эта важная характеристика должна оцениваться с разных сторон.

## 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

### 2.1 Лабораторная работа № 1 ( 2 часа).

**Тема:** Системное ПО

**2.1.1 Цель работы:** Познакомиться с видами , назначением, составом и этапами загрузки ОС.

#### 2.1.2 Задачи работы:

1. Различать ОС.
2. Проследить этапы загрузки ОС.

#### 2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. ПК
2. ОС MS DOS
3. Оболочка Norton Commander

#### 2.1.4 Описание (ход) работы:

Среди всех системных программ, с которыми пользователю приходится работать, особое место занимает ОС.

ОС управляет работой компьютера, запускает программы, обеспечивает защиту данных, выполняет различные сервисные функции по запросам пользователей и программ.

*Операционная система (ОС) – это комплекс системных программ, обеспечивающий совместное функционирование всех устройств компьютера и поддерживающий работу всех его программ.*

Каждая программа пользуется услугами ОС, а потому может работать только под управлением той ОС, которая обеспечивает для нее эту услугу. Таким образом, выбор ОС очень важен, так как он определяет с какими программами вы сможете работать. Так же от выбора ОС зависит производительность ваших работ, степень защиты данных, а также возможности прикладных программ.

Человек в форме команд дает указание ОС, она в свою очередь переводит их на язык, понятной машине и передает ей на выполнение. Компьютер обрабатывает полученную информацию, и результаты передает ОС. ОС переводит их на язык понятный человеку и выводит их на экран.



ОС – это дисковая ОС фирмы Дискос, потому что загружается в работу с диска (винчестера). Была разработана в 1981 году. Недостатками является то, что эта ОС является однозадачной и однопользовательской. Версия MS DOC 6.22 выпущенная в 1994 году работает как самостоятельная ОС. Версия 7.0 1995 года выпуска работает только в системе Windows., т.е. ее правильнее рассматривать, как подсистему Windows.

Диалог пользователя с DOS осуществляется в форме команд, которые пользователь вводит с клавиатуры в ответ на приглашение DOS. Ввод каждой команды заканчивается нажатием клавиши Enter..Когда ОС готова к приему команд, она выводит приглашение C:\>.

**Приглашение, как правило, содержит информацию о текущем дисководе и текущем каталоге.**

**Например:**

A:>\ - текущим является корневой каталог диска A:.

C:>\ - текущим является корневой каталог диска C:.

C:\WINDOW> - текущим является каталог WINDOWS диска C:.

D:\GAME\DOOM> - текущим является подкаталог DOOM каталога GAME

**диска D:.**

Логический диск – это **физический диск, реальный диск или часть физического диска, которому присвоено имя.**

**Способ хранения файлов на дисках компьютера называется файловой системой. Иерархическая структура, в виде которой ОС отображает файлы, и папки диска называют файловой структурой.**

**Файл** – это поименованная область на диске или другом носителе информации. В файлах может храниться различная информация: тексты, рисунки, чертежи, таблицы. Имя файла имеет обозначение, которое состоит из двух частей: собственно имени и расширения. Имя файла всегда уникально, оно образуется не более из 8 символов латинского алфавита. Тип файла (расширение) служит для характеристик, хранящейся в файле информации и образуется не более чем из 3 символов. Например:

com – командный системный файл,

exe – исполняемый файл,

txt – текстовый файл,

hlp – файл справки,

doc – расширение текстового процессора MS Word,

avi – файл содержит видеоинформацию,

html – файл содержит Web-страницу

Расширение от основного имени отделяется точкой, например, text.txt.

**Имя файла** – последовательность символов, позволяющая пользователю ориентироваться в файловой системе и идентифицировать файлы.

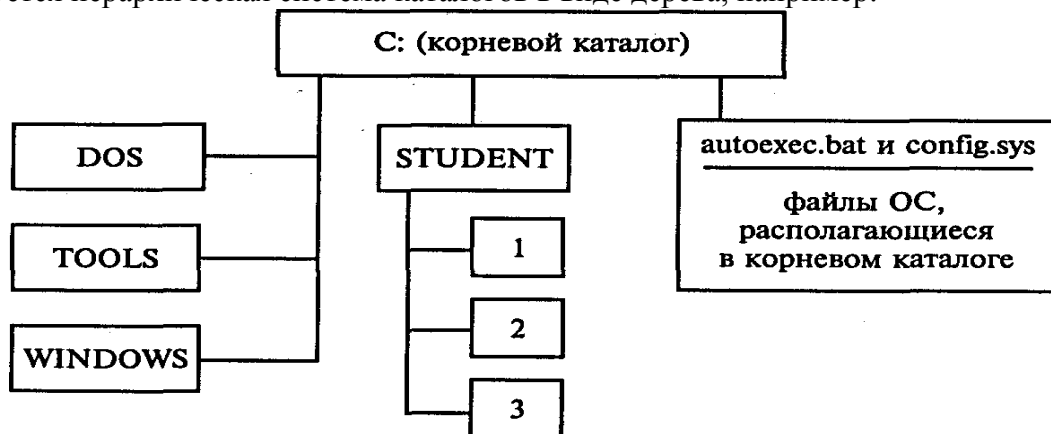
**Тип (расширение) файла** – последовательность символов, позволяющая компьютеру сопоставить программное обеспечение содержимому файла.

**Каталог** – это специальное место на диске, в котором регистрируются имена файлов, их месторасположение на диске, а также дополнительные сведения о файлах и их атрибутах. Каталоги имеют свои собственные имена и могут храниться в других каталогах вместе с обычными файлами. Имена каталогов строятся по тем же правилам, что и имена файлов. Например: Student, группа. Расширение для каталогов задавать не принято. Каталог, с которым в настоящее время работает пользователь, называется *текущим*. Каталог, с которым нет связи в данный момент времени, называется *пассивным*.

Чтобы получить доступ к файлу данных, необходимо указать к нему путь. **Путь к файлу** – это последовательность из имен дисков и каталогов, разделенных символом «\».

**Полное имя файла** – имя логического диска+путь к файлу+имя файла.

Кроме файлов, в каталогах могут содержаться подкаталоги. В результате на диске образуется иерархическая система каталогов в виде дерева, например:



Самым первым является корневой каталог. В данном примере это каталог диска C:

**Доступ к файлу** организуется с помощью задания пути (цепочка соподчиненных каталогов, которую необходимо пройти по иерархической структуре к каталогу, где зарегистрирован искомый файл). Взаимодействие пользователя с операционной системой осуществляется с помощью командной строки,

индицируемой на экране дисплея.

Групповые спецификации

При операциях копирования, перемещения, удаления файлов и т.п. для облегчения работы файлы можно объединять в группы. Для объединения файлов используют следующие значки:

\* — заменяет любое число символов в имени файла или его расширении;

? — заменяет один произвольный символ.

Обозна	Пояснения к команде
*.doc	Группа файлов, имеющих расширение dos :
n*g.*	Группа всех файлов, имеющих первую букву в имени файла n и последнюю букву g. Расширение может быть любым .
m?h.*	Группа всех файлов, имя файла которых состоит из трех букв, первая из которых m, последняя h. Расширение может быть любым

Список внутренних команд

Внутренние команды MS DOS встроены в командный процессор command.com и работают под его управлением. Команды вводятся с клавиатуры, их ввод завершается нажатием клавиши <ВВОД> (<ENTER>). Набирать команды DOS можно как строчными буквами, так и прописными. В формате записи команд в квадратных скобках помечены необязательные параметры.

#### 1. Команда смены текущего логического диска.

Для смены текущего логического диска надо выбрать имя логического диска, который должен стать текущим, и затем двоеточие, например:

A:

C:.

#### 2. Изменение текущего каталога (переход в другой каталог)

**CD <логический диск:> путь**

C:\>CD WINDOWS - переход из текущего каталога в каталог WINDOWS. Левая косая черта в начале пути (перед WINDOWS) заставляет идти ОС через корневой каталог

CD.. - переход на уровень вверх

CD\ - переход в корневой каталог текущего диска

#### 3. Просмотр оглавления каталога.

**DIR <логический диск:> <путь> <параметры>**

**DIR - просмотр содержимого в текущем каталоге**

A:\ DIR GRUPPA - просмотр содержимого, находящегося на диске A: в каталоге GRUPPA

**DIR K\*.\* - просмотр списка файлов текущего каталога, начинающихся на букву K**

DIR \*.txt - просмотр списка всех файлов с расширением txt

**DIR A?.\* - просмотр списка файлов с именами из двух знаков, первый из которых буква A, и произвольными расширениями**

DIR /P - просмотр списка файлов «порциями» (постранично)

DIR /W - выводит информацию в сокращенном виде — только имена файлов и директориев (в столбцы).

#### 4. Создание каталога

**MD <имя каталога>**

C:\ MD LETO - создание каталога под именем LETO в текущем каталоге на текущем диске

C:\ MD STUDENT\NATALI - создание подкаталога с именем NATALI в каталоге STUDENT диска C:

#### 5. Удаление каталога

**RD <имя каталога>**- если он пустой

**DELTREE <имя каталога>** - со всеми файлами

RD MY - удаление каталога MY

RD GRAFIK\SPIRAL - удаление каталога SPIRAL. Каталог SPIRAL является подкаталогом каталога GRAFIK.

#### 6. Переименование каталога.

**MOVE <старое имя каталога> <новое имя каталога>**

MOVE GRUPPA KLASS – переименование каталога GRUPPA в KLASS.

#### 7. Создание текстовых файлов.

**Сору con <имя файла>**

Сору con pismo.txt - создание текстового файла с именем pismo.txt. Ввести необходимый текст. Для сохранения файла и выхода из режима редактирования текста нажать F6 или Ctrl+Z, затем Enter. Появится сообщение о записи файла на диск в текущей директории и на диске появится файл с указанным именем.

#### 8. Переименование файла.

**REN <имя файла 1> <имя файла 2>**

REN pismo.doc letter.doc - переименование файла pismo.doc в текущем каталоге в файл letter.doc

#### 9. Вывод содержимого файла на экран.

**TYPE <имя файла>**

TYPE letter.doc –просмотр файла letter.doc из текущего каталога.

TYPEA:\TEXT\pismo.txt - просмотр содержимого файла pismo.txt, находящегося на диске A: в директории TEXT.

#### 10. Удаление файлов.

**DEL <имя файла>**

A:\ DEL \NC\\*.txt - удаление всех файлов с расширением txt из каталога NC диска A:

***DEL pismo.txt - удаляет файл pismo.txt***

DEL \*.psx - удаление всех файлов с расширением psx из текущего каталога

#### 11. Копирование файлов (каталогов).

**COPY <имя файла1> <имя файла2>**

COPY <имя файла1> <имя каталога2>

COPY <имя каталога1> <имя каталога2>

COPY abc.txt A: - копирование файла abc.txt на диск A:.

C:\ COPY STUDENT\pismo.doc A:\ТЕКСТ - копирование файла с именем pismo.doc из каталога STUDENT диска C: в директорию ТЕКСТ, находящуюся на диске A:

COPY \*.doc A:\ - копирование всех файлов из текущего каталога в корневой каталог диска A:

## 12. Объединение файлов

**COPY <имя файла1> +<имя файла2> <имя файла3>**

COPY celeron.txt+athol.txt pentium.txt – объединение файлов celeron.txt и athol.txt в pentium.txt

## 13. Просмотр дерева каталогов.

C:\TREE – дерево каталогов на диске C:.

TREE LETO дерево каталога LETO.

### **Команды общесистемного назначения**

#### **CLS – очистка экрана**

DATE – вывод информации о дате и установка даты в компьютере

TIME – вывод информации о времени и установка времени в компьютер

VER – вывод информации о версии ОС

Внешние команды

Помимо команд, распознаваемых и выполняемых командным процессором в операционной системе имеется большое число утилит — команд, реализованных в виде отдельных программ.

**FORMAT [диск:] /f:V /q /s V-** задание объема диска. Максимальный объем форматирования для дискет с двумя отверстиями — 1,44 Мбайт, для дискет с одним отверстием — 720 кбайт

FORMAT A: /q - быстрое форматирование дискеты, объем по умолчанию 1,44Мбайт

**SYS.com** - переносит скрытые системные файлы io.sys, msdos.sys и command.com на требуемый диск

SYS A: - перенос системных файлов из текущей директории (где есть системные файлы) на диск A:

**MEM [/c] [/p]** - получение информации о распределении памяти компьютера

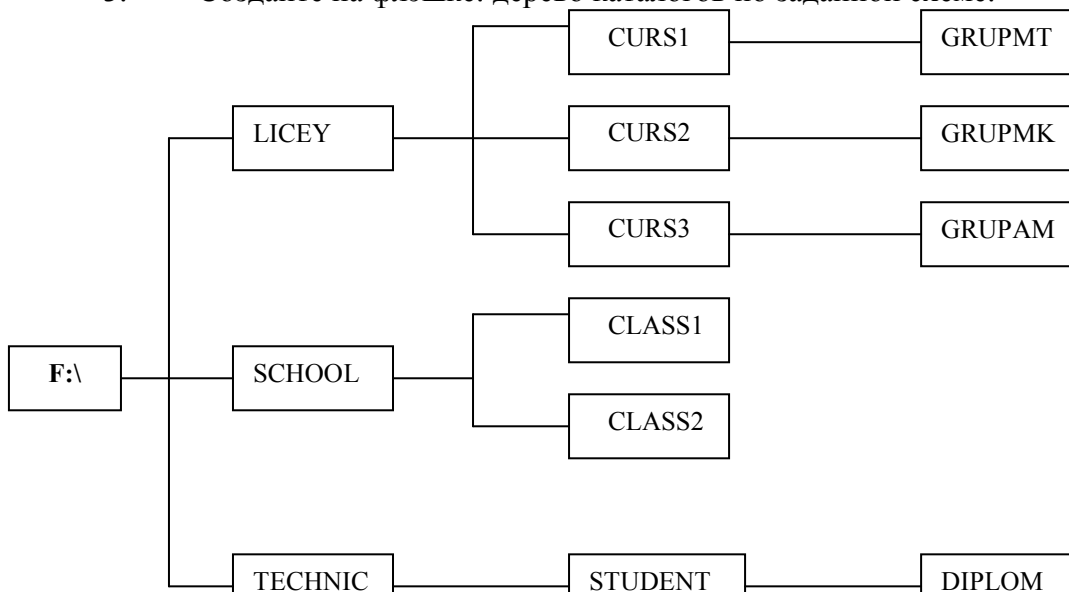
MEM /с /р - вывод постранично на экран списка программ, расположенных в оперативной памяти компьютера, указывая их местоположение и размер (значение параметра в десятичном коде)

*MEM - без параметров команды выводит лишь резюме с информацией об имеющемся в распоряжении объеме памяти*

**PRINT TEXT.txt** - вывод на принтер текстового файла с именем TEXT.txt

### **Выполните следующие задания:**

1. Включите компьютер.
2. Запустите MS DOS (Пуск→Программы→Стандартные→Командная строка).
3. Создайте на флэшке: дерево каталогов по заданной схеме.





4. Просмотрите оглавление корневого каталога диска F:.
5. Просмотрите оглавление подкаталога LICEY.
6. Удалите подкаталог GRUPAM.
7. Просмотрите оглавление подкаталога SCHOOL.
8. Удалите подкаталог CLASS1.
9. Создайте текстовые файлы zadania1.txt и zadania2.txt в подкаталоге CLASS2.
10. Просмотрите оглавление подкаталога CLASS2.
11. Переименуйте подкаталог DIPLOM в каталог ПРОЕКТ.
12. Создайте подкаталог ZACHET в каталоге STUDENT.
13. Выведите на экран названия файлов и каталогов, начинающихся на букву L.
14. Переименуйте текстовый файл zadania1.txt в prakt.txt.
15. Просмотрите содержимое файла prakt.txt.
16. Просмотрите содержимое каталога CLASS2.
17. Удалите файл prakt.txt в подкаталоге CLASS2.
18. Скопируйте файл zadania2.txt в подкаталог GRUPMT.
19. Просмотрите содержимое подкаталога GRUPMT.
20. Удалите подкаталог CLASS2.
21. Выведите Дерево каталогов диска F:.
22. Начертите полученную схему Древа каталогов в тетради для практических работ.
23. Продемонстрируйте результаты работы преподавателю.
24. Завершите работу MS DOS.

## 2.2 Лабораторная работа №2 (4 часа).

**Тема:** Прикладное программное обеспечение

«Создание, редактирование и форматирование документов с использованием текстового процессора MS Word.»

**2.2.1.Цель работы:** получить начальные навыки работы с текстовым документом.

Изучение информационной технологии создания документов MS Word с использованием Шаблонов. Использование списков.

**2.2.2 Задачи работы:**

1. Создание документов в MS WORD
2. Редактирование документов в MS WORD

**2.2.3 Перечень приборов, материалов, используемых в лабораторной работе:**


1. Персональный компьютер.
2. Текстовый процессор Microsoft Word

**2.2.4 Описание (ход) работы:**

**Запуск MS Word и создание нового документа**

Запуск программы Word может осуществляться любым из стандартных способов запуска приложений:


- Щелкните кнопку **Пуск** и выберите в меню пункт **Создать документ Microsoft Office**. В открывшемся окне дважды щелкните по значку **Новый документ**.
- Щелкните кнопку **Пуск**, выберите пункт **Программы, Microsoft Word**.
- Дважды щелкните левой клавишей мыши по значку **Microsoft Word** на Рабочем столе.

Как правило, при запуске программы MS Word можно сразу приступить к вводу текста. Создание нового документа в других случаях (уже в самом текстовом редакторе) осуществляется командами **Файл, Создать** или с помощью кнопки **Создать**  на панели инструментов Стандартная.

### **Экранный интерфейс редактора Microsoft Word (Окно программы MS Word)**

После запуска программы на экране появляется окно с открытым в нем пустым документом, которому по умолчанию присваивается имя **Документ 1**. Окно содержит следующие элементы (Рис.1):

Заголовок окна (первая строка) – слева в строке заголовка находится значок Word (системное меню), рядом имя загруженного файла документа, с которым вы работаете в текущий момент (в данном случае этот файл называется **Документ 1**), затем – имя программы - **Microsoft Word** и справа 3 кнопки управления окном (свернуть, развернуть (свернуть в окно), закрыть документ).

Строка меню – находится под строкой заголовка. Каждый пункт меню раскрывается и имеет соответствующее подменю, если по нему щелкнуть левой клавишей мыши. Пункты строки меню открываются в два приема. На первом этапе открываются сокращенные меню, а затем (при необходимости, щелкнув по кнопке ) - расширенное меню.

Панель инструментов – находится ниже строки меню. Обычно на экране, по умолчанию, отображаются две панели инструментов *Стандартная* и *Форматирование*. Остальные панели при необходимости могут быть открыты командами: **Вид, Панели инструментов**. Внутри панелей инструментов собраны различные элементы управления, которые представляют наиболее часто выполняемые операции при работе с документами. Панели инструментов обеспечивают быстрый и более простой доступ к важнейшим функциям редактора.

Ниже находится Рабочая область – это область экрана, где отображается создаваемый документ и производится его редактирование. Сверху и слева документа находятся горизонтальная и вертикальная линейки.

Линейка – используется при формировании абзацев и таблиц. Для вывода на экран линейки, нужно выполнить команды: **Вид, Линейка**.

Справа и внизу документа – полосы вертикальной и горизонтальной прокрутки. Они применяются в тех случаях, когда весь текст не помещается на экране и требует сдвига (прокрутки) вверх-вниз или влево-вправо, используя при этом *кнопки прокрутки* или *ползунок прокрутки*.

Окно документа Word можно разделить на две части (с помощью команды меню **Окно, Разделить**) и работать одновременно с двумя частями одного и того же документа. При этом каждая часть документа имеет свою собственную *полосу прокрутки*. Полосу разделения можно перемещать с помощью мыши.

Разделить экран на два подокна можно перетаскиванием вниз маленькой кнопки (вешки полосы разделения), которая находится над верхней стрелкой вертикальной полосы прокрутки.

Закрыть второе подокно можно командой меню **Окно, Снять разделение** или перемещением разделителя.

Мигающая вертикальная черта (штрих) - текстовый курсор – указывает, куда в следующий момент времени будет вводиться знак текста, вставляться таблица или рисунок. Передвигать текстовый курсор можно с помощью клавиш управления курсором или с помощью мыши, произведя щелчок в нужном месте документа.

Нижняя строка окна программы называется информационной строкой или строкой состояния. В ней отображается различная информация о самом документе и о текущем состоянии текстового процессора. Первые четыре числа означают номер страницы в

разделе, номер раздела, номер страницы в документе и сколько всего страниц.

### Первоначальные сведения и правила работы с Word.

1. При работе с документом пользователь задает подходящий режим просмотра и редактирования текста. Режим устанавливается с помощью пункта меню **Вид**. В режиме **Обычный** достигается наибольшая скорость ввода и редактирования текста. Режим **Разметка страницы** более точно отображает окончательный внешний вид страниц документа при печати, кроме того, он обязателен при работе с графическими объектами, рамками и многоколоночным текстом. Режим **Структура документа** и **Режим электронного документа** полезны опытным пользователям при работе с большими документами.

2. Используя команды **Вид, Масштаб** можно изменять масштаб видимости документа. Масштабирование текста можно также осуществлять с помощью комбинированного поля *Масштаб*  на панели инструментов *Стандартная*.

3. Начинать работу с текстовым документом необходимо с настройки параметров страницы. Это осуществляется с помощью команды меню **Файл, Параметры страницы**. В диалоговом окне **Параметры страницы** выбрав вкладку **Поля**, можно установить отступы от всех краев листа бумаги. Вкладка **Размер бумаги** позволяет установить размер самого листа бумаги и его ориентацию при печати: **Книжная** или **Альбомная** и т.д.

4. Так же можно пронумеровать страницы. Для этого нужно:


- выполнить команду меню **Вставка, Номера страниц**;
- указать положение и выравнивание номера.

Номера страниц можно установить, используя колонтитулы. Колонтитулы представляют собой специальное поле, размещаемое, как правило, в верхнем и нижнем полях страницы, содержащее различные объекты, повторяющиеся на каждой странице, например, время, имя файла, имя автора и т.п. Устанавливаются колонтитулы через команду меню **Вид, Колонтитулы**.

5. Для автоматического переноса слов в документе необходимо выполнить команды **Сервис, Язык, Расстановка переносов** и в появившемся диалоговом окне включить переключатель *Автоматическая расстановка переносов*.

6. Для проверки правописания, необходимо щелкнуть кнопку **Правописание** на панели инструментов *Стандартная* или в меню **Сервис** выбрать команду **Правописание**.

### Основные приемы редактирования текста



Под редактированием понимается изменение уже существующего документа. Начинают редактирование с загрузки (открытия) документа. Для этого используется команда **Файл, Открыть** или кнопка **Открыть**  на панели инструментов *Стандартная*, по выполнению которой на экране появляется стандартное диалоговое окно **Открытие документа**. Если нужный документ находится в другой папке, его надо разыскать и открыть.

Процедура редактирования текста включает следующий комплекс операций:

1. **Ввод текста** – производится обычным способом при помощи клавиатуры. Следует отметить, что клавиша Enter служит для обозначения конца абзаца. Перенос текста на новую строку производится автоматически при достижении текстом правой границы. Если при вводе текста была допущена ошибка, она устраняется удалением символов при помощи клавиш Backspace (удаляет символ слева от курсора) или Delete (удаляет символ справа).

2. **Копирование**. Чтобы скопировать фрагмент текста, надо:

- выделить нужный фрагмент:



- скопировать его в буфер обмена с помощью команды меню **Правка, Копировать** или использовать кнопку  на панели инструментов *Стандартная*;
- установить курсор в то место текста, куда нужно вставить фрагмент;
- выполнить команду **Правка, Вставить** или использовать кнопку  на панели инструментов *Стандартная*. Вставить фрагмент из буфера обмена можно столько раз, сколько нужно.

Прежде чем начинать редактировать или форматировать фрагмент текста, необходимо его выделить. Это можно сделать с помощью мыши или используя клавиатуру.

Чтобы выделить	Действие
Слово	Дважды щелкнуть слово.
Строку текста или несколько строк текста	Переместить указатель мыши в зону выделения (Зона выделения находится между левой границей текста и листа. В этой зоне указатель мыши принимает вид стрелки, направленной в правый верхний угол ↗) и щелкнуть левой кнопкой мыши. Для выделения более чем одной строки, необходимо удерживая левую кнопку мыши, осуществлять ее перемещение вверх или вниз.
Абзац	Переместить указатель мыши в зону выделения и сделать двойной щелчок левой кнопкой мыши.
Весь текст	Переместить указатель мыши в зону выделения и сделать тройной щелчок левой кнопкой мыши.
Часть строки	Переместите указатель мыши к первому выделяемому символу строки так, чтобы он принял вид <b>I</b> , удерживая нажатой левую кнопку мыши, протяните мышь вправо (влево) до нужного символа.
Блок текста	Щелкните начало фрагмента и удерживая клавишу Shift нажатой, щелкните конец фрагмента.

Чтобы снять выделение, щелкните по любому месту левой кнопкой мыши.

3. **Удаление текста.** Для удаления фрагмента текста необходимо его выделить, а затем выполнить команду **Правка, Вырезать**.


4. **Отмена** последнего действия осуществляется с помощью команд **Правка, Отменить** или с помощью кнопки  на панели инструментов *Стандартная*. Вернуть отмененное действие можно кнопкой **Вернуть** .

### Форматирование документа в MS Word

Целью обработки любого текста является придание ему некоторого вида облегчающего восприятие информации и подчиненного общепринятым правилам и стандартам. **Форматирование** – придание документу определенной формы. Форматирование текста осуществляется средствами меню **Формат** или с помощью панели инструментов *Форматирование*. Процедура форматирования текста включает следующие операции:

**Форматирование абзацев.** При вводе текста новый абзац начинается после очередного нажатия клавиши Enter. Форматирование абзаца подразумевает: выравнивание абзаца по заданным границам, установку межстрочных интервалов и отступов. Для форматирования одного абзаца его не обязательно выделять. Достаточно установить курсор в любое место внутри абзаца. Чтобы применить одинаковое форматирование к нескольким абзацам, их необходимо предварительно выделить.

**Выравнивание.** Это способ расположения строк между правой и левой границами текста. Существует четыре вида выравнивания абзаца – по центру, влево, вправо и по ширине. Выравнивание абзаца можно выполнить, используя команды **Формат, Абзац,**

вкладка **Отступы и интервалы**, в поле **Выравнивание** или кнопки  на панели инструментов *Форматирование*.

**Установка межстрочного интервала.** Это расстояние между строками в абзаце. Интервалы устанавливаются с помощью команд **Формат, Абзац**, вкладка **Отступы и интервалы**, в комбинированном поле **Межстрочный**.


**Форматирование символов** – подразумевает изменение типа, стиля, размера, начертания шрифта.

Установка шрифтов. Текст документа может быть оформлен разными шрифтами. Шрифтовое оформление текста производится либо по мере того, как он вводится, либо уже на введенном тексте. Прежде чем выполнять шрифтовое оформление существующего текста, необходимо произвести его выделение. Затем выполнить команды **Формат,**


**Шрифт** или воспользоваться кнопками  на панели инструментов *Форматирование* и выбрать нужные параметры.

Имеется ряд дополнительных возможностей при выборе шрифта: Подчеркивание, Цвет, Эффекты.

#### **Сохранение документа**

Для сохранения документа используются команды: **Файл, Сохранить** или **Сохранить как**. Команда **Сохранить** или кнопка  на панели инструментов *Стандартная* сохраняет документ с текущим именем, а команда **Сохранить как** позволяет сохранить документ под новым именем. Для этого в диалогом окне **Сохранение документа** необходимо выбрать папку, в которой нужно сохранить документ, ввести имя файла с клавиатуры и щелкнуть по кнопке «Сохранить»

#### **Печать документа**

Перед тем как печатать документ, его необходимо просмотреть. Для переключения в режим предварительного просмотра необходимо выбрать команды: **Файл, Предварительный просмотр** или использовать на панели инструментов *Стандартная* кнопку . Выход из режима просмотра документа осуществляется щелчком по кнопке «Заккрыть».

Печать документа производится командами: **Файл, Печать**. В окне диалога **Печать** выбирают принтер (щелкнув по кнопке **Свойства**, можно изменить свойства печати, например, задать качество печати), указывают номера распечатываемых страниц и количество копий документа. Затем нажимают на кнопку «ОК».

**ЗАДАНИЕ 1.** Наберите текст, приведенный ниже. В процессе набора текста меняйте начертание, размер шрифта и интервал шрифта.

**У В А Ж А Е М Ы Е   Г О С П О Д А !**

**Приглашаем вас на юбилейную презентацию**

компьютерной фирмы «МАКУЛШИРС».

Мы работаем на российском рынке много лет. Программные продукты нашей фирмы знают и любят многие пользователи страны. Наши компьютеры работают без рекламаций!

**Б у д е м   р а д   в и д е т ь   в а с .**

**Запомните наш адрес и время нашей презентации**

**Компьютерная улица, д.5, 18-00.**

**Справки по телефону 123-45-67**

**Выполните следующие действия:**

1. Выделите ПРИГЛАШЕНИЕ и заключите надпись в рамку. Для этого заключите текст в рамку и произведите цветовую заливку.

Для этого:

- выделить весь текст одним из возможных способов;
- в меню **ФОРМАТ** выбрать **ГРАНИЦЫ И ЗАЛИВКА**;
- во вкладке **ГРАНИЦА** установить тип **РАМКА** и тип линии, **цвет авто**;
- во вкладке **ЗАЛИВКА** установить тип узора (например, 30%) и любой цвет;
- нажмите кнопку ОК.

*Замечание:* если после рамки предполагается какой-либо текст, то прежде чем выполнять действия по установке рамки необходимо несколько раз нажать <Enter>.

2. Прodelайте ряд самостоятельных упражнений по копированию перестановки и удалению абзацев, предложений в абзацах, слов в предложениях и букв в словах. Для этого в меню ПРАВКА, используйте команды: **ВЫРЕЗАТЬ**, **КОПИРОВАТЬ**, **ВСТАВИТЬ**.

3. Проверьте орфографию, используя встроенный словарь: установив курсор в начале текста, выполните команду **СЕРВИС, ПРАВОПИСАНИЕ** (при отсутствии ошибок будет выдано сообщение об окончании проверки правописания; при подозрении на ошибку будет выделено соответствующее слово или предложение, которое можно отредактировать, используя кнопки диалогового окна).

4. Выделяя части текста, используйте **ФОРМАТ, ШРИФТ**, чтобы поменять цвет слов.

5. В меню **ФОРМАТ, ШРИФТ** освойте вкладки **ИНТЕРВАЛ** и **АНИМАЦИЯ**.

6. Освойте функцию поиска и замены. Для этого в меню **ПРАВКА, НАЙТИ** и **ЗАМЕНИТЬ**, найти слово Компьютер и заменить на ЭВМ.

7. Используйте в написании документа сноску, используя меню **ВСТАВКА, ССЫЛКА, СНОСКА**.

8. Примените в тексте буквицу. Для этого в меню **ФОРМАТ, выбрать вкладку БУКВИЦА**.


9. Сохраните документ, скопируйте его на свою дискету.


Лабораторная работа 13 (ЛР-13) Создание и редактирование таблиц, формул и графических объектов в MS Word. Создание текстовых документов на основе шаблонов. Создание шаблонов и форм.

### **СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦ В MS WORD**

Чтобы создать таблицу в том месте, где находится курсор, достаточно:

- Выбрать в меню **ТАБЛИЦА** опцию **ДОБАВИТЬ ТАБЛИЦУ (ВСТАВИТЬ, ТАБЛИЦА)**. В открывшемся диалоговом окне установить размерность таблицы (число столбцов и строк).

- Быстро создать таблицу можно с помощью кнопки  **Добавить таблицу** на панели инструментов *Таблицы и границы*. Для определения конфигурации новой таблицы нужно закрасить при нажатой левой кнопке мыши требуемое число столбцов и строк таблицы.

- Можно создавать таблицы, рисуя их «карандашом» с помощью мыши при активизированной кнопке  **Нарисовать таблицу**.

Новая таблица состоит из строк и столбцов с пустыми ячейками. Чтобы ввести данные (текст, числа, графика, рисунки, формулы) в ячейку таблицы, щелкните на ячейке (в ней появится текстовый курсор) и наберите нужные данные.

Совет: Если в момент создания таблицы вы еще не знаете, сколько строк вам понадобится, просто создайте таблицу из одной строки. Можно очень легко добавить к концу таблицы новые строки во время ввода текста.



Меню **ТАБЛИЦА** открывает также альтернативные возможности для вставки и изменения таблиц, кроме того, меню позволяет выполнять некоторые дополнительные операции, которые невозможны при использовании интерактивных приемов.

**Команды меню «Таблица» и их действия**

<b>Команды</b>	<b>Описание</b>
<i>Нарисовать таблицу</i>	<i>Создание новой таблицы</i>
<i>Добавить (Вставить)</i>	<i>Добавление таблицы, строк, столбцов или ячеек</i>
<i>Удалить</i>	<i>Удаление таблицы, выделенных строк, столбцов или ячеек</i>
<i>Выделить</i>	<i>Выделение строки, столбца, ячеек или таблицы целиком</i>
<i>Объединить ячейки</i>	<i>Объединение нескольких выделенных ячеек в одну</i>
<i>Разбить ячейки</i>	<i>Разбиение выделенных ячеек на заданное число строк или столбцов</i>
<i>Разбить таблицу</i>	<i>Разбиение таблицы на две</i>
<i>Автоформат</i>	<i>Автоматическое форматирование таблицы. В списке представлены различные стили оформления таблицы</i>
<i>Автоподбор</i>	<i>Автоматическая настройка ширины и высоты столбцов</i>
<i>Заголовки</i>	<i>Использование выделенной строки в качестве заголовка таблицы, который при печати будет выводиться на каждой странице, если таблица располагается на нескольких листах</i>
<i>Преобразовать</i>	<i>Преобразование таблицы в текст или наоборот</i>
<i>Сортировка</i>	<i>Упорядочивание данных в выделенных строках или списке по алфавиту, величине или дате</i>
<i>Формула</i>	<i>Выполнение математических операций в таблице</i>
<i>Отображать сетку</i>	<i>Команда позволяет включать и отключать отображение сетки из точечных линий</i>
<i>Свойства таблицы</i>	<i>Задание различных параметров таблицы</i>

**Вводить данные в таблицу**, форматировать и редактировать их по своему усмотрению можно с помощью мыши. Перед тем, как вставлять, удалять или регулировать размеры таблицы, строки, столбца, ячейки необходимо предварительно их выделить с помощью мыши или команд меню **Таблица**, **Выделить строку** (столбец, ячейку, всю таблицу).

**Способы выделения элементов таблицы с помощью мыши**

<b>Выделяемый элемент</b>	<b>Способ выделения</b>
Ячейка	Подвести указатель мыши в левый угол ячейки, чтобы он принял вид направленной вправо стрелки ↗, и щелкнуть левой клавишей мыши
Строка	Щелкнуть левой клавишей мыши на полосе выделения – слева от выделяемой строки
Столбец	Установить указатель мыши в верхней части столбца так, чтобы он принял вид направленной вниз черной стрелки ↓, и щелкнуть левой клавишей мыши
Вся	Нажать левую кнопку мыши в начале таблицы и протащить


таблица	указатель мыши до конца таблицы
---------	---------------------------------

Чтобы поменять **ширину колонки**, надо взяться мышью за правую ее границу или за движок координатной линейки (указатель мыши примет вид ↔) и отодвинуть на нужное расстояние. Чтобы поменять **высоту строки**, надо взяться мышью за горизонтальную линию, отмечающую нижнюю границу строки и подвинуть ее вверх или вниз. Чтобы поменять **ширину выделенной ячейки**, надо взяться мышью за правую или левую ее границу и подвинуть в нужную сторону. Она подвинет соседей и увеличится (уменьшится).

Для **выравнивания** текста в ячейке таблицы **по вертикали** следует установить курсор в нужную ячейку, нажать правую кнопку мыши и выбрать в контекстном меню ячейки команду **Выравнивание** и выбрать нужный шаблон.

Для изменения **направления текста** используется команда контекстного меню **Направление текста**.

Для получения **рамки** внутри и вокруг таблицы необходимо выделить таблицу и использовать:

- команды меню **Формат, Границы и заливка**;
- кнопки панели **Форматирование Внешние границы** ;
- кнопки панели инструментов **Таблицы и границы** и границы



. Использование панели инструментов **Таблицы и границы** упрощает работу с таблицами и позволяет реализовывать некоторые дополнительные возможности их форматирования.

**ЗАДАНИЕ 1.** Создать следующую таблицу (тип шрифта - Times New Roman, 14; межстрочный интервал - одинарный):

Наименование продукции	Посев, площадь, га	Урожайность, ц/га
Ячмень	760	15,4
Просо	300	12,6
Гречиха	210	10,4
Подсолнечник	340	20,6
Пшеница	560	12,5
Рожь	410	16,3

**ЗАДАНИЕ №2.** Скопируйте таблицу. Преобразуйте скопированную таблицу в ниже следующую, используя всевозможные способы редактирования:

Наименование продукции	Посев, площадь, га		Урожайность, ц/га		Недобор продукции
	200	200	200	200	
	5г.	6г.	5г.	6г.	
Ячмень	760	750	15,4	15,5	0,01
Просо	300	250	12,6	12,4	0,2
Гречиха	210	240	10,4	11,2	0,009
Подсолнечник	340	500	20,6	19,0	1,5
Пшеница	560	490	12,5	12,3	0,12
Рожь	410	350	16,3	15,4	1,01



Добавьте итоговую строку сводных результатов. В итоговой строке рассчитать общую посевную площадь за 2005 и 2006 годы. Для этого установить курсор в ячейку и выполнить команду ТАБЛИЦА, ФОРМУЛА. в диалоговом окне «Формула» набрать следующее выражение: =SUM(RIGHT).

Итого	2580	2580			
-------	------	------	--	--	--

#### Дополнительные задания:

1. Удалите любую строку из таблицы.
2. Добавьте произвольную по содержанию строку в таблицу (после первой строки ниже шапки таблицы).
3. Исклучите 2 столбец (посевная площадь за 2005 год).
4. Отработайте команду ВЫСОТА И ШИРИНА ЯЧЕЙКИ.
5. Измените шрифт и фон шапки таблицы.
6. Добавьте строку в конце таблицы с произвольным содержанием.
7. Измените цветовое оформление таблицы.
8. Сохраните документ на свою дискету.
9. Для закрепления материала создайте таблицу в Приложении (задание №2).

.ТЕМА №3.

#### ЗАПИСЬ МАТЕМАТИЧЕСКИХ ФОРМУЛ

**Цель:** научиться создавать и редактировать формулы и математические выражения с использованием программного блока Microsoft Equation 3.0 и библиотеки символов.

Для создания и редактирования формул используется программный блок Microsoft Equation 3.0. выполните следующие действия

Краткая справка. На верхней панели (математических символов) расположены кнопки для вставки в формулу более 150 математических символов, большая часть которых недоступна в стандартном шрифте Symbol. Для вставки символов в формулу нажмите кнопку в верхнем ряду панели инструментов, а затем выберите определенный символ из палитры, появляющейся над кнопкой.

На нижней панели (шаблонов) расположены кнопки, предназначенные для вставки шаблонов или структур, включающих символы типа дробей, радикалов, сумм, интегралов, произведений, матриц и различных скобок или соответствующих пар символов типа круглых и квадратных скобок. Во многих шаблонах содержатся специальные места, в которые можно вводить текст и вставлять символы. В редакторе формул содержится около 120 шаблонов, сгруппированных в палитры. Шаблоны можно вкладывать один в другой для построения многоступенчатых формул.

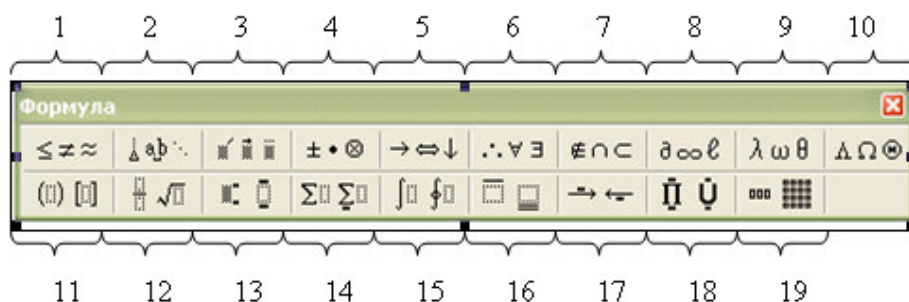


Рис. 3.1. Панель Equation Editor

#### Назначение нижних и верхних кнопок панели «Редактора формул»

(согласно нумерации кнопок панели на рис. 5.1)

1 – вставка символов отношений;

- 2 – вставка пробелов и многоточий;  
 3 – надсимвольные элементы, позволяющие добавлять к математическим переменным примы, крышки, черту или точку;  
 4 – вставка операторов;  
 5 – вставка стрелок;  
 6 – вставка логических символов;  
 7 – вставка символов теории множеств;  
 8 – вставка разных символов (символы дифференциального исчисления, символы градуса, угла, перпендикуляра и др.);  
 9 – вставка строчных букв греческого алфавита;  
 10 – вставка прописных букв греческого алфавита;  
 11 – вставка шаблонов разделителей:

$$\left(\frac{g}{r}\right) = \frac{g!}{r!(g-r)!};$$

- 12 – вставка шаблонов дробей и радикалов:

$$\sqrt{\frac{1}{z}} + \sqrt{\frac{1}{z} + \sqrt{z}};$$

- 13 – создание верхних и нижних индексов:

$$\lim_{h \rightarrow 0} h(x);$$

- 14 – создание сумм:

$$\sum_{\substack{1 \leq x \leq m \\ 1 \leq y \leq n}} a_{xy} \quad \sum_{i=1}^m \sum_{s=1}^n a_{is} \quad \sum_i \sum_j \sum_k a_{ij} b_{jk} c_{ki};$$

- 15 – вставка интегралов:

$$\Phi(\alpha, \beta) = \int_0^\alpha \int_0^\beta e^{-(x^2+y^2)} dx dy;$$

- 16 – создание математических выражений с чертой сверху и снизу:

$$\overline{\lim} h(n) \quad \underline{\lim} h(n);$$

- 17 – создание стрелок с текстом:

$$f_n(x) \xrightarrow{\text{равномерно}} f : X \xrightarrow{\text{на}} Y;$$

- 18 – вставка произведений и шаблонов теории множеств;

- 19 – вставка шаблонов матриц. Шаблоны этой палитры позволяют создавать векторные столбцы, определители, матрицы и другие макеты типа таблиц:

Функция	Производная
$x^n$	$nx^{n-1}$
$\lg x$	$x^{-1}$
$e^x$	$e^x$

**ЗАДАНИЕ №1.** Создайте следующий фрагмент формулы  $\sum_{I=1}^K Z$ . Для этого:

- поместите курсор в точку вставки формулы и выполните **Вставка, Объект, вкладка Создание**;
- в поле *Тип объекта* выберите **Microsoft Equation 3.0, ОК** (открыли редактор формул);

- в появившейся *Панели инструментов* выберите **Шаблоны сумм**;
- выберите **Шаблон суммы с верхним и нижним пределами** (крайний правый шаблон в верхнем ряду);
- введите нужный символ, число или выражение в каждый из слотов, предварительно помещая туда курсор;
- Завершите запись формулы (щелчок за пределами рамки или **Esc**).

**ЗАДАНИЕ №2.** Создайте формулу остатка ряда Тейлора, для этого:

- поместите курсор в точку вставки формулы; 
$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-a)^{(n+1)}$$
- откройте редактор формул;
- в слоте ввода формулы введите **R**;
- выберите **Шаблоны верхних и нижних индексов**;
- выберите соответствующий шаблон (второй в верхней строке) и введите **n**;
- поместите курсор в конец введенной формулы и введите **=**;
- выберите **Шаблоны дробей и радикалов** и выберите соответствующий шаблон (слева сверху);
- в слот числителя введите  $f(\xi)$  (греческую букву выберите на *панели инструментов*);
- установите курсор справа от **f**;
- выберите **Шаблоны верхних и нижних индексов** и соответствующий шаблон (сверху слева);
- введите **(n+1)**;
- в слот знаменателя введите **(n+1)!**;
- установите курсор в конец введенной формулы и введите оставшуюся часть формулы аналогичным способом;
- завершите запись формулы.

Для изменения формата формулы выполните следующие действия:

- выделите формулу (одинарный щелчок);
- выберите **Формат, Объект**, вкладка **Обтекание, Вокруг рамки, ОК**;
- увеличьте размер формулы и переместите формулу так, чтобы они не перекрывали друг друга.

**ЗАДАНИЕ №3.** Создайте формулы с фигурными скобками по образцу:

- откройте редактор формул;
- в слоте ввода введите **y=**;
- выберите **Шаблоны скобок** и нажмите соответствующую скобку; 
$$y = \begin{cases} a + x, \text{ при } x > 0 \\ a - x, \text{ при } x \leq 0 \end{cases}$$
- поместите курсор в слот и выберите **Шаблоны матриц**, выберите шаблон второй в верхней строке, в верхний и нижний слоты введите соответствующие выражения;
- завершите запись формул.

**ЗАДАНИЕ №4.** Создайте матричную формулу по образцу:

- откройте редактор формул;
- введите левую часть формулы (греческие буквы  $\Delta = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_1 & b_1 & c_1 \\ a_1 & b_1 & c_1 \end{vmatrix}$

прописные)=;

- выберите **Шаблоны матриц** и нажмите шаблон справа внизу;
- в открывшемся диалоговом окне задайте:
- число строк – **3**, число столбцов – **3**;
- выравнивание столбцов – **по центру**, выравнивание строк – **по опорной линии**;
- щелчком слева и справа от образца установите **тип линии** матрицы, подтвердите **ОК**.
- в первый слот первой строки введите **a**;
- выберите **Шаблоны верхних и нижних индексов**, нажмите шаблон, устанавливающий нижний индекс справа и введите **1**;
- аналогичным образом заполните остальные слоты, завершите запись формулы.

Краткая справка Для составления достаточно простых математических выражений можно использовать в меню **Вставка** вкладку **Символ**. Далее в окне диалога **Символ** выбрать шрифт и нужный символ. Выбранный символ вставляется в текст на место курсора за счет нажатия кнопки **Вставить**.

**ЗАДАНИЕ.№5.** Составьте математическое выражение, используя символьный набор формул.

$$\lim f(x)=A \Leftrightarrow \forall \varepsilon > 0 \exists \delta > 0$$

Для этого:

- установите курсор в то место текста, куда нужно вставить символ;
- выберите в меню команду **Вставка, Символ**, вкладка **Символы**;
- в поле *Шрифт* установите тип шрифта **Symbol**, выберите нужный символ и нажмите кнопку «Вставить».

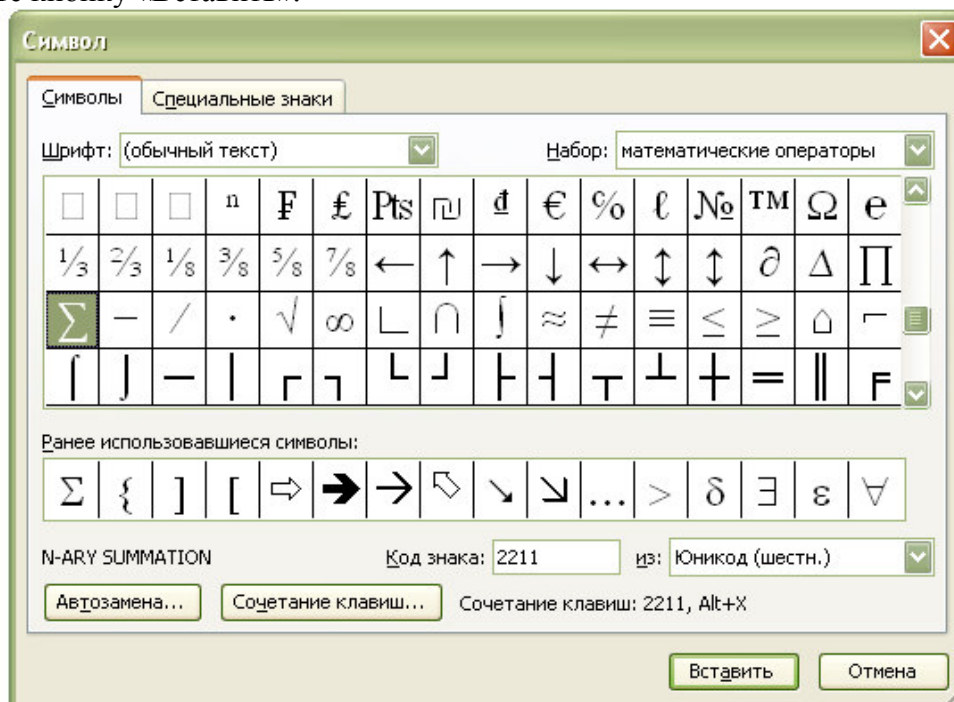


Рис. 3.2. Вставка символа суммы  $\Sigma$

Сохраните документ под именем **Формулы**.

**ЗАДАНИЕ № 6.** Для закрепления материала создайте формулу, предложенную в Приложении (задание №3)

#### **ТЕМА 4:**

#### **СОЗДАНИЕ БЛОК-СХЕМ В MICROSOFT WORD**

В папке «Мои документы», в своей собственной папке создайте Документ Word с названием «Блок-схема». Произведете установку всех необходимых параметров. Установите панель инструментов **РИСОВАНИЕ**.

*Рекомендация:* перед тем как приступить к созданию блок-схемы, прощелкайте весь лист клавишей <Enter> - это позволит устанавливать курсор в любую строку листа.

#### **Создание элементов блок-схемы**

Для того, чтобы создать блок-схему найдите на панели инструментов кнопку **АВТОФИГУРЫ** Щелкните по ней левой клавишей мыши. Здесь Вы можете выбрать необходимый элемент для Вашей блок-схемы и перенести его на лист. Любая автофигура является объектом редактирования (не забудьте предварительно ее выделить, щелкнув на ней один раз левой клавишей мыши) и на нее распространяются действия по копированию, удалению, всевозможные изменения, содержащиеся в меню **ФОРМАТ АВТОФИГУРЫ**. Все это подробно описано в предыдущей теме. Кроме того, выделенную автофигуру можно просто растягивать или сжимать используя для этого только мышь. Аналогично можно редактировать и стрелки, так как они также являются содержимым меню АВТОФИГУРЫ.

В меню **ДЕЙСТВИЯ** Вы можете группировать, разгруппировать, перегруппировать выбранный набор фигур (выбор набора осуществляется при помощи стрелки **ВЫБОР ОБЪЕКТОВ** путем установки и растяжки курсора), установить порядок, сделать сдвиг, выровнять/распределить, повернуть/отразить, изменить автофигуру. Кроме того, возможно осуществление свободного вращения автофигуры, используя соответствующую кнопку

#### **Установка надписи в автофигуру**

В автофигуру можно вставить надпись любого содержания. Для этого надо выделить автофигуру, а затем выбрать кнопку **НАДПИСЬ** и щелкнуть изменившимся курсором внутри автофигуры. Теперь вписывайте текст (можете выбрать свой шрифт, интервал и т.д.).

В рамках автофигуры можно изменить направление надписи (в меню **ФОРМАТ строка НАПРАВЛЕНИЕ ТЕКСТА**).

*Замечание:* если текст не умещается в фигуре, то уменьшите размер выбранного шрифта. Если это не принесло нужного результата, то щелкните по фигуре дважды левой клавишей мыши -появится меню **ФОРМАТ АВТОФИГУРЫ** (ранее это меню мы открывали правой клавишей мыши). Выберите вкладку **НАДПИСЬ** и уберите внутренние поля (просветы) в фигуре, возможно текст поместится.

#### **Дополнительные возможности**

Можно изменить цвет автофигуры, установить тень и объем. Все это можно сделать используя меню **ФОРМАТ АВТОФИГУРЫ** и меню **РИСОВАНИЕ** на панели инструментов. Кроме того, установить нужные глубину, освещение, направление, поверхность, цвет объема можно при помощи дополнительной панели (рис. 6, цифра 5), которая вызывается командой **ВИД, ПАНЕЛИ ИНСТРУМЕНТОВ, НАСТРОЙКА, НАСТРОЙКА ОБЪЕМА**.

**ЗАДАНИЕ.** Создайте блок-схему, изображенную на рис. 4.1.

Для этого выполните следующие действия:

1. В меню **АВТОФИГУРЫ** (или сразу непосредственно на панели инструментов **РИСОВАНИЕ**) выберите овал, начертите его (попробуйте изобразить **овал**, нажимая клавишу <Shift>). Установите глубину 72 пт, направление и освещение любое. Впишите в

овал надпись (процесс вписывания см. выше).

2. Начертите стрелку толщиной 0,75 пт.

3. В АВТОФИГУРАХ находите и чертите прямоугольник, устанавливайте глубину, направление, руководствуясь рисунком. Затем снова стрелку.

4. Изображая блок условия, нужно сделать надписи над стрелками. Это достигается следующим способом:

- выберите кнопку НАДПИСЬ;

- изобразите прямоугольник на месте будущего слова;

- напечатайте слово, если нужно измените направление текста, шрифт, размер надписи;

- правой клавишей мыши войдите в ФОРМАТ НАДПИСИ и во вкладке ЦВЕТА И ЛИНИИ сделайте установки: НЕТ ЗАЛИВКИ, НЕТ ЛИНИЙ; ОК.

5. Изображайте следующие фигуры блок-схемы, но при этом необходимо учесть следующие замечания:

**Замечания:**

- для тонкой настройки и подгонки положения фигур (особенно это относится к линиям и стрелкам) используйте стрелки управления курсором с одновременным нажатием клавиши <Ctrl>;

- не старайтесь определить конец линии точно от начала фигуры; удобнее, чтобы линия заходила на фигуру; так как, используя команду ДЕЙСТВИЯ. НА ЗАДНИЙ ПЛАН (предварительно выделив линию), Вы легко достигаете необходимого эффекта;

- при создании блок-схемы следите, чтобы ее части не выходили за выделенные параметры страницы.

**6. Завершающий этап**

Когда создание блок-схемы завершено, удобно хранить ее в сгруппированном виде. Для этого осуществляется выбор всех объектов блок-схемы при помощи соответствующей клавиши и выполняется команда ДЕЙСТВИЯ, ГРУППИРОВАТЬ. В таком виде удобно копировать созданный рисунок. Кроме того, его можно редактировать (сжимать, растягивать) как единую автофигуру.

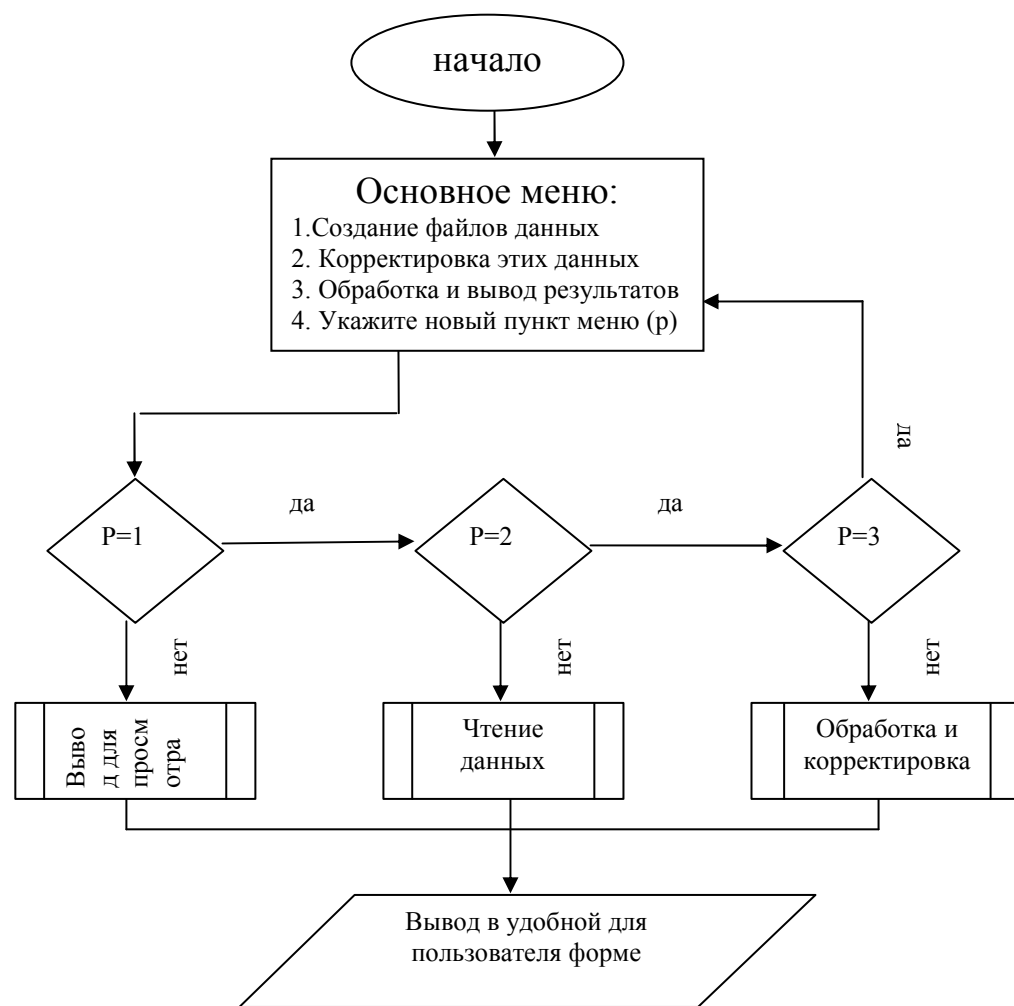


Рис. 4.1.

Если возникла необходимость отредактировать какую-либо внутреннюю составляющую сгруппированной блок-схемы, то рисунок необходимо разгруппировать (действия нелогичные группировке), произвести изменения и снова сгруппировать (или перегруппировать).

*Замечание:* вывод документа на печать производить только в разгруппированном виде.

### 2.3. Лабораторная работа № ЛР-3 (6 ЧАСОВ)

**Тема:** Программирование на алгоритмическом языке высокого уровня 2.3.1.Цель работы:

**Научится составлять простейшие программы на АЯ высокого уровня**

#### 2.3.1 Задачи работы:

1. Изучить основные операторы АЯ.

#### 2.3.2 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер.

#### 2.3.3 Описание (ход) работы:

**Задание 1.** Реализовать алгоритмы на АЯ высокого уровня, рассмотренные на ЛР-

1,2.