

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

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.Б.14 Информатика

**Направление подготовки 38.03.01 Экономика
Профиль подготовки Бухгалтерский учет, анализ и аudit
Форма обучения заочная**

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 Краткое содержание вопросов:

1. Этапы информатизации общества

Деятельность отдельных людей, групп, коллективов и организаций сейчас все в большей степени начинает зависеть от их информированности и способности эффективно использовать имеющуюся информацию. Прежде чем предпринять какие-то действия, необходимо провести большую работу по сбору и переработке информации, ее осмыслению и анализу. Отыскание рациональных решений в любой сфере требует обработки больших объемов информации, что подчас невозможно без привлечения специальных технических средств.

Возрастание объема информации особенно стало заметно в середине XX в. Лавинообразный поток информации хлынул на человека, не давая ему возможности воспринять эту информацию в полной мере. В ежедневно появляющемся новом потоке информации ориентироваться становилось все труднее. Подчас выгоднее стало создавать новый материальный или интеллектуальный продукт, нежели вести розыск аналога, сделанного ранее. Образование больших потоков информации обусловливается:

- чрезвычайно быстрым ростом числа документов, отчетов, диссертаций, докладов и т.п., в которых излагаются результаты научных исследований и опытно-конструкторских работ;
- постоянно увеличивающимся числом периодических изданий по разным областям человеческой деятельности;
- появляются противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими мощными потоками и массивами хранящейся информации;
- существует большое количество избыточной информации, которая затрудняет восприятие полезной для потребителя информации;
- возникают определенные экономические, политические и другие социальные барьеры, которые препятствуют распространению информации. Например, по причине соблюдения секретности часто необходимой информацией не могут воспользоваться работники разных ведомств.

Эти причины породили весьма парадоксальную ситуацию – в мире накоплен громадный информационный потенциал, но люди не могут им воспользоваться в полном объеме в силу ограниченности своих возможностей. Информационный кризис поставил общество перед необходимостью поиска путей выхода из создавшегося положения. Внедрение ЭВМ, современных средств переработки и передачи информации в различные сферы деятельности послужило началом нового эволюционного процесса, называемого информатизацией, в развитии человеческого общества, находящегося на этапе индустриального развития.

Информатизация общества – организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной

власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов.

Рассмотрим этот процесс более подробно.

История развития информатизации началась в США с 60-х гг., затем с 70-х гг. – в Японии и с конца 70-х – в Западной Европе.

Современное материальное производство и другие сферы деятельности все больше нуждаются в информационном обслуживании, переработке огромного количества информации. Универсальным техническим средством обработки любой информации является компьютер, который играет роль усилителя интеллектуальных возможностей человека я общества в целом, а коммуникационные средства, использующие компьютеры, служат для связи и передачи информации. Появление и развитие компьютеров – это необходимая составляющая процесса информатизации общества.

Информатизация общества является одной из закономерностей современного социального прогресса. Этот термин все настойчивее вытесняет широко используемый до недавнего времени термин "компьютеризация общества". При внешней похожести этих понятий они имеют существенное различие.

При компьютеризации общества основное внимание уделяется развитию и внедрению технической базы компьютеров, обеспечивающих оперативное получение результатов переработки информации и ее накопление.

При информатизации общества основное внимание уделяется комплексу мер, направленных на обеспечение полного использования достоверного, исчерпывающего и своевременного знания во всех видах человеческой деятельности.

Таким образом, "информатизация общества" является более широким понятием, чем "компьютеризация общества", и направлена на скорейшее овладение информацией для удовлетворения своих потребностей. В понятии "информатизация общества" акцент надо делать не столько на технических средствах, сколько на сущности и цели социально-технического прогресса. Компьютеры являются базовой технической составляющей процесса информатизации общества.

Информатизация на базе внедрения компьютерных и телекоммуникационных технологий является реакцией общества на потребность в существенном увеличении производительности труда в информационном секторе общественного производства, где сосредоточено более половины трудоспособного населения. Так, например, в информационной сфере США занято более 60% трудоспособного населения, в СНГ – около 40%.

На основе вышесказанного можно сделать вывод: во второй половине XX века человечество вступило в новый этап своего развития. В этот период начался переход от **индустриального общества к информационному**. Информационное общество имеет следующие основные признаки:

1) Большинство работающих в информационном обществе (около 80%) занято в информационной сфере, т.е. сфере производства информации и информационных услуг.

2) Обеспечены техническая, технологическая и правовая возможности доступа любому члену общества практически в любой точке территории и в приемлемое время к нужной ему информации.

3) Информация становится важнейшим стратегическим ресурсом общества и занимает ключевое место в экономике, образовании и культуре.

Следовательно, **информационное общество** – это общество, в котором большая часть населения занята получением, обработкой, передачей и хранением информации.

Процесс, обеспечивающий переход от индустриального общества к информационному, получил название **информатизации**.

Информатизация общества – повсеместное внедрение комплекса мер, направленных на обеспечение полного и своевременного использования достоверной информации, и зависит от степени освоения и развития новых информационных

технологий.

Неизбежность информатизации общества обусловлена резким возрастанием роли и значения информации. Информационное общество характеризуется высокоразвитой информационной сферой, которая включает деятельность человека по созданию, переработке, хранению, передаче и накоплению информации.

Процесс перехода от индустриального общества к информационному происходит неодновременно в различных странах, он характеризуется также разными темпами развития. Первыми на этот путь встали в конце 50-х – начале 60-х годов XX века США, Япония и страны Западной Европы. В этих государствах, начиная с 60 – 70-х годов, проводится политика повсеместной информатизации всех сфер деятельности человека. В СССР в 1989 г. Разработана Концепция информатизации общества. По мнению специалистов, любая страна, насколько бы индустриально развитой она ни была, перейдет в разряд стран третьего мира, если опоздает с информатизацией.

Поэтому умение правильно использовать компьютерные технологии в своей профессиональной работе становится неотъемлемым показателем квалификации современного специалиста в любой области.

2. Цель и задачи информатики. Основные понятия

Научным фундаментом процесса информатизации общества является научная дисциплина – **информатика**.

Термин «информатика» был заимствован из французского языка и обозначал название области, связанной с автоматизированной обработкой информации с помощью электронных вычислительных машин. Именно развитие компьютерной техники способствовало выделению информатики в самостоятельную область человеческой деятельности.

Информатика – это наука, занимающаяся исследованием форм и методов сбора, хранения, накопления и передачи информации с помощью компьютерных и других технических средств.

Это дисциплина комплексная, тесно связанная с другими науками, у которых есть общий объект исследования – информация.

Задачами информатики являются:

1. Разработка и производство современных средств вычислительной техники.
2. Проектирование и внедрение прогрессивных технологий
3. Дальнейшая информатизация общества и повышение его информационной культуры.

Целью курса информатики является:

4. Помочь всем желающим (непрофессиональным пользователям) овладеть основами компьютерной грамотности.
5. Способствовать развитию логического мышления.
6. Познакомиться с аппаратными и программными средствами компьютера.
7. Изучить основы алгоритмизации и программирования.

Понятие «информатика» неразрывно связано с такими понятиями, как «информационная система» и «информационная технология». Информатика изучает свойства, структуру и функции информационных систем, а также происходящие в них информационные процессы.

Под **информационной системой** понимают – взаимосвязанную совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Подавляющее большинство современных информационных систем являются автоматизированными. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом для информационных систем.

Информационная система немыслима без персонала, взаимодействующего с компьютерами и телекоммуникациями.

Информационная технология – это процесс, включающий совокупность способов сбора, хранения, обработки и передачи информации на основе применения средств вычислительной техники.

Каждая ИТ реализуется в рамках конкретной информационной системы. Таким образом ИС является средой обитания для соответствующей ИТ. В качестве инструментария ИТ выступают такие программные средства, как: текстовый редактор, электронные таблицы, системы управления базами данных, издательские системы и т.д. Информатика тесно связана с кибернетикой, но не тождественна ей. **Кибернетика** – это наука об общих принципах управления в различных системах: технических, биологических, социальных и др. независимо от наличия или отсутствия компьютеров.

Информатика как научная дисциплина занимается изучением информационных процессов. **Информационными процессами** – называются процессы, связанные с получением, хранением, преобразованием и передачей информации.

Количество информации в современном обществе стремительно нарастает, человек оказывается погруженным в море информации. Чтобы быть востребованным, он должен обладать **информационной культурой**, т.е. знаниями и умениями в области информационных технологий, а также знать юридические и этические нормы, действующие в этой сфере.

Информатика как наука стала развиваться с середины нашего столетия, что связано с появлением ЭВМ и начинающейся компьютерной революцией.

Появление вычислительных машин в 50-е годы создало для информатики необходимую ей аппаратную поддержку, или, иначе говоря, благоприятную среду для ее развития как науки. Всю историю информатики принято разбивать на два больших этапа: **предыстория и история**.

Предыстория информатики такая же древняя, как и история развития человеческого общества. В предыстории, в свою очередь, выделяют ряд этапов. Каждый из этих этапов характеризуется по сравнению с предыдущим резким возрастанием возможностей хранения, передачи и обработки информации.

Начальный этап предыстории – появление устной речи. Освоение природы человеком началось, прежде всего, с присвоения готовых продуктов (растений, животных, плодов, тепла солнца и т.д.). Затем человек научился не только брать готовое у природы, но и воздействовать на нее, создавая новые вещества, новые виды энергии, приобретая новые знания. Появление устной речи позволило первобытному человеку делиться накопленными знаниями со своими соплеменниками. Таким образом, членораздельная речь, язык стал специфическим социальным средством хранения и передачи накопленных знаний.

По мере накопления знаний людей об окружающем мире возникла необходимость использования искусственных способов их фиксации. Так появилась письменность – второй этап предыстории. При этом, прежде всего, резко возросли (по сравнению с предыдущим этапом) возможности по хранению информации. Человек получил искусственную внешнюю память. Организация почтовых служб позволила использовать письменность как средство передачи информации. Кроме того, возникновение письменности было необходимым условием для начала развития наук (вспомним, например, Древнюю Грецию).

Третий этап – книгопечатание. Книгопечатание можно смело назвать первой информационной технологией. Воспроизведение информации было поставлено на поток, на промышленную основу. По сравнению с предыдущим этот этап не столько увеличил возможности по хранению, сколько повысил доступность информации и точность ее воспроизведения.

Четвертый и последний этап предыстории связан с успехами точных наук (прежде

всего математики и физики) и начинающейся в то время научно-технической революцией. Этот этап характеризуется возникновением таких мощных средств связи, как радио, телефон и телеграф, к которым по завершению этапа добавилось и телевидение. Кроме средств связи появились новые возможности по получению и хранению информации – фотография и кино. К ним также следует добавить разработку методов записи информации на магнитные носители (магнитные ленты и диски).

В XVII веке появились первые счетные машины на зубчатых передачах, прообразы арифмометра, созданного в 70-х годах XIX века петербургским изобретателем В.Т.Однером и прослужившего человечеству почти столетие. В этот же период английский математик Чарльз Бэббидж попытался построить универсальное вычислительное устройство – аналитическую машину, которая должна была выполнять вычисления без участия человека. Попытка оказалась неудачной, но в этой машине были заложены основные идеи, используя которые в 1943 году Говард Эйкеп смог построить первую электромеханическую вычислительную машину "Марк-1". С этого же времени начинается и электронная эпоха развития информатики.

Возникновение информатики как науки, начало ее **истории** принято связывать с разработкой первых электронных вычислительных машин (ЭВМ) в середине 40-х годов. Для такой "привязки" имеется несколько причин. Во-первых, сам термин "информатика" появился на свет благодаря развитию вычислительной техники, и поначалу под ним понималась наука о вычислениях (первые ЭВМ большей частью использовались для проведения числовых расчетов). Во-вторых, выделению информатики в отдельную науку способствовало такое важное свойство современной вычислительной техники, как единая форма представления обрабатываемой и хранимой информации. Вся информация, вне зависимости от ее вида, хранится и обрабатывается на ЭВМ в двоичной форме. Так получилось, что компьютер в одной системе объединил хранение и обработку числовой, текстовой (символьной) и аудиовизуальной (звук, изображение) информации. В этом состояла инициирующая роль вычислительной техники при возникновении и оформлении новой науки.

Поколения ЭВМ

В соответствии с элементной базой и уровнем развития программных средств выделяют четыре реальных поколения ЭВМ, краткая характеристика которых приведена в таблице:

Параметры сравнения	Поколения ЭВМ			
	первое	второе	третье	четвертое
Период времени	1946 - 1959	1960 - 1969	1970 - 1979	с 1980 г.
Элементная база (для УУ, АЛУ)	Электронные (или электрические) лампы	Полупроводники (транзисторы)	Интегральные схемы	Большие интегральные схемы (БИС)
Основной тип ЭВМ	Большие		Малые (мини)	Микро
Основные устройства ввода	Пульт, перфокарточный, перфоленточный ввод	Добавился алфавитно-цифровой дисплей, клавиатура	Алфавитно-цифровой дисплей, клавиатура	Цветной графический дисплей, сканер, клавиатура
Основные устройства вывода	Алфавитно-цифровое печатающее устройство (АЦПУ), перфоленточный вывод		Графопостроитель, принтер	
Внешняя память	Магнитные ленты, барабаны, перфоленты, перфокарты	Добавился магнитный диск	Перфоленты, магнитный диск	Магнитные и оптические диски
Ключевые решения в ПО	Универсальные языки программирования, трансляторы	Пакетные операционные системы, оптимизирующие трансляторы	Интерактивные операционные системы, структурированные языки программирования	Дружественность ПО, сетевые операционные системы
Режим работы ЭВМ	Однопрограммный	Пакетный	Разделения времени	Персональная работа и сетевая обработка данных
Цель использования ЭВМ	Научно-технические расчеты	Технические и экономические расчеты	Управление и экономические расчеты	Телекоммуникации, информационное обслуживание

3. Структура информатики

По своей структуре информатика представляет собой единство трех взаимосвязанных частей – технические средства, программные средства, алгоритмические средства. В свою очередь, информатику как в целом, так и каждую часть обычно рассматривают с разных позиций: как отрасль народного хозяйства, как теоретическую науку, как прикладную дисциплину.

Информатика как отрасль народного хозяйства обеспечивает все другие отрасли необходимыми информационными ресурсами.

Информационные ресурсы – информация, используемая на производстве, в технике, управлении обществом, специально организованная и обрабатываемая на ЭВМ. Информационные ресурсы в объеме страны – национальные информационные ресурсы. Информационные ресурсы страны определяют ее научно-технический прогресс, научный потенциал, экономическую и стратегическую мощь.

Теоретическая информатика рассматривает все аспекты разработки автоматизированных информационных систем: их проектирования, создания и использования не только с формально-технической, но и с содержательной стороны, а также комплекс экономического, политического и культурного воздействия на социальную динамику. В орбиту анализа теоретической информатики попадают и традиционные системы преобразования информации и распространения знаний: средства и системы массовой информации, система лекционной пропаганды, кино, театры, справочные службы и т.д. При этом информатика рассматривает их с позиций получения и использования информационного ресурса, форм и способов воздействия указанных систем на общественный прогресс.

Теоретическая информатика изучает информационный ресурс, законы его функционирования и использования как движущей силы социального прогресса, а также общие, фундаментальные проблемы информационных технологий как исторического феномена, выводящего общество на новую ступень развития.

Решающее значение для рождения теоретической информатики имеет появление информационных технологий высшего уровня, основанных на искусственном интеллекте.

Прикладная информатика изучает конкретные разновидности информационных технологий, которые формируются с помощью специальных информационных систем (управленческих, медицинских, обучающих, военных и др.). Информационные технологии в различных отраслях, имея общие черты, в то же время существенно отличаются между собой. Разные операции и процедуры, различное оборудование, специализация критериев и показателей, даже разные носители информации, – все это становится объектом изучения конкретных функциональных и отраслевых информатик. Так рождаются ветви прикладной информатики, обслуживающие создание проектирующих систем, экспертных систем, управляющих и других функциональных систем. Поэтому наряду с теоретической информатикой развиваются ее конкретные ветви: экономическая информатика, военная информатика, медицинская информатика и многие

другие.

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 ход 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.

Авторское право распространяется на любые программы для ЭВМ и базы данных, представленные в объективной форме, независимо от их материального носителя, назначения и достоинства. Авторское право распространяется на программы для ЭВМ и базы данных, являющиеся результатом творческой деятельности автора. Творческий характер деятельности автора предполагается до тех пор, пока не доказано обратное.

Предоставляемая настоящим Законом правовая охрана распространяется на все виды программ для ЭВМ, которые могут быть выражены на любом языке и в любой форме, и на базы данных, представляющие собой результат творческого труда по подбору и организации данных. Предоставляемая правовая охрана не распространяется на идеи и принципы, лежащие в основе программы для ЭВМ и базы данных или какого-либо их элемента, в том числе идеи и принципы организации интерфейса и алгоритма, а также языки программирования.

Авторское право на программы для ЭВМ и базы данных вступают в силу их создания. Авторское право на базу данных признается при условии соблюдения авторского права на каждое из произведений, включенных в базу данных. Автором программы для ЭВМ и базы данных признается физическое лицо, в результате творческой деятельности которого они созданы.

Этические аспекты информатики

Этика – система норм нравственного поведения человека.

Далеко не все правила, регламентирующие деятельность в сфере информатики, можно свести в правовыми нормами. 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, представляющих созданные ими программные средства для свободного распространения.

1.2 Лекция № 2 (2 часа).

Тема: «Системное программное обеспечение»

1.2.1 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

1.2.2 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 связанны 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 автономными 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.

Инструментальное ПО

Инструментальное ПО или системы программирования - это системы для автоматизации разработки новых программ на языке программирования.

В самом общем случае для создания программы на выбранном языке программирования (языке системного программирования) нужно иметь следующие компоненты:

1. Текстовый редактор для создания файла с исходным текстом программы.
2. Компилятор или интерпретатор. Исходный текст с помощью программы-компилятора переводится в промежуточный объектный код. Исходный текст большой программы состоит из нескольких модулей (файлов с исходными текстами). Каждый модуль компилируется в отдельный файл с объектным кодом, которые затем надо объединить в одно целое.

3. Редактор связей или сборщик, который выполняет связывание объектных модулей и формирует на выходе работоспособное приложение – исполняемый код.

Исполняемый код – это законченная программа, которую можно запустить на любом компьютере, где установлена операционная система, для которой эта программа создавалась. Как правило, итоговый файл имеет расширение .EXE или .COM.

4. В последнее время получили распространение визуальный методы программирования (с помощью языков описания сценариев), ориентированные на создание Windows-приложений. Этот процесс автоматизирован в средах быстрого проектирования. При этом используются готовые визуальные компоненты, которые настраиваются с помощью специальных редакторов.

Наиболее популярные редакторы (системы программирования программ с использованием визуальных средств) визуального проектирования:

Borland Delphi - предназначен для решения практически любых задач прикладного программирования

Borland C++ Builder – это отличное средство для разработки DOS и Windows приложений

Microsoft Visual Basic – это популярный инструмент для создания Windows-программ Microsoft Visual C++ - это средство позволяет разрабатывать любые приложения, выполняющиеся в среде ОС типа Microsoft Windows

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 EC)
- 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, Widows 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.

В связи с этим группы секторов условно объединяются в кластеры. Кластер является наименьшей единицей адресации к данным. Размер кластера, в отличие от размера сектора, не фиксирован и зависит от емкости диска.

Сначала для дисков и небольших жестких дисков (менее 16 Мбайт) использовалась 12-разрядная версия FAT (так называемая FAT12). Затем в MS-DOS была введена 16-разрядная версия FAT для более крупных дисков.

Операционные системы MS DOS, Win 95, Win NT реализуют 16 – разрядные поля в таблицах размещения файлов. Файловая система FAT32 была введена в Windows 95 OSR2 и поддерживается в Windows 98 и Windows 2000.

FAT32 представляет собой усовершенствованную версию FAT, предназначенную для использования на томах, объем которых превышает 2 Гбайт.

FAT32 обеспечивает поддержку дисков размером до 2 Тбайт и более эффективное расходование дискового пространства. FAT32 использует более мелкие кластеры, что позволяет повысить эффективность использования дискового пространства.

В Windows XP применяется FAT32 и NTFS. Более перспективным направлением в развитии файловых систем стал переход к NTFS (New Technology File System – файловая система новой технологии) с длинными именами файлов и надежной системой безопасности.

Объем раздела NTFS не ограничен. В NTFS минимизируется объем дискового пространства, теряемый вследствие записи небольших файлов в крупные кластеры. Кроме того, NTFS позволяет экономить место на диске, сжимая сам диск, отдельные папки и файлы.

По способам именования файлов различают “короткое” и “длинное” имя.

Согласно соглашению, принятому в MS-DOS, способом именования файлов на компьютерах IBM PC было соглашение 8.3., т.е. имя файла состоит из двух частей: собственно имени и расширения имени. На имя файла отводится 8 символов, а на его расширение – 3 символа.

Имя от расширения отделяется точкой. Как имя, так и расширение могут включать

только алфавитно-цифровые символы латинского алфавита. Имена файлов, записанные в соответствии с соглашением 8.3, считаются “короткими”.

С появлением операционной системы Windows 95 было введено понятие “длинного” имени. Такое имя может содержать до 256 символов. Этого вполне достаточно для создания содергательных имен файлов. “Длинное” имя может содержать любые символы, кроме девяти специальных: \ : * ? < > |.

В имени разрешается использовать пробелы и несколько точек. Имя файла заканчивается расширением, состоящим из трех символов. Расширение используется для классификации файлов по типу.

Уникальность имени файла обеспечивается тем, что полным именем файла считается собственное имя файла вместе с путем доступа к нему. Путь доступа к файлу начинается с имени устройства и включает все имена каталогов (папок), через которые проходит. В качестве разделителя используется символ “\” (обратный слеш - обратная косая черта). Например: D:\Documents and Settings\TBA\Мои документы\lessons-tva\robots.txt

Несмотря на то, что данные о местоположении файлов хранятся в табличной структуре, пользователю они представляются в виде иерархической структуры – людям так удобнее, а все необходимые преобразования берет на себя операционная система.

К функции обслуживания файловой структуры относятся следующие операции, происходящие под управлением операционной системы: создание файлов и присвоение им имен;

- создание каталогов (папок) и присвоение им имен;
- переименование файлов и каталогов (папок);
- копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска;
- удаление файлов и каталогов (папок);
- навигация по файловой структуре с целью доступа к заданному файлу, каталогу(папке);
- управление атрибутами файлов.

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 интерфейса, 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)
- Стока 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

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.

1.3 Лекция № 3 (2 часов).

Тема: «Алгоритмизация вычислительных процессов»

1.3.1 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
 - 3.3. Алгоритм циклической структуры

1.3.2 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 символыми 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 выполниться 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)

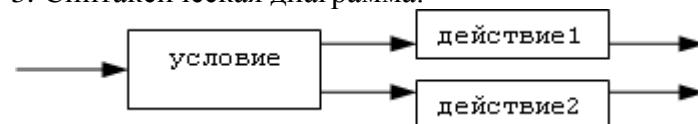
1. Естественный язык:

если условие то действие1 иначе действие2

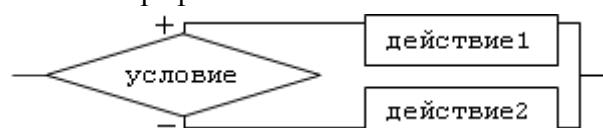
2. Структурограмма:



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 процесса (рис. 1).

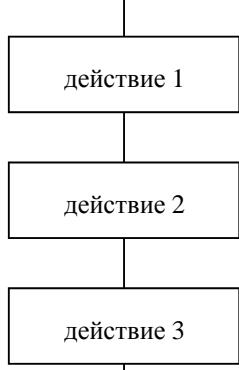


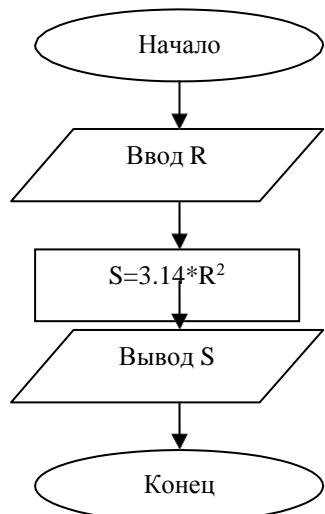
Рисунок 1- Блок-схема линейного алгоритма

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, составление алгоритма линейного вычислительного процесса не вызывает затруднений, однако в "чистом виде" такие вычислительные процессы встречаются весьма редко. Чаще всего они являются составной частью более сложных вычислительных процессов. Таким образом:

- линейный вычислительный процесс является наиболее простым при реализации его на ЭВМ.
- блок-схема алгоритма линейного вычислительного процесса представляет собой **композицию** (объединение) нескольких следующих друг за другом блоков процесс.

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**.

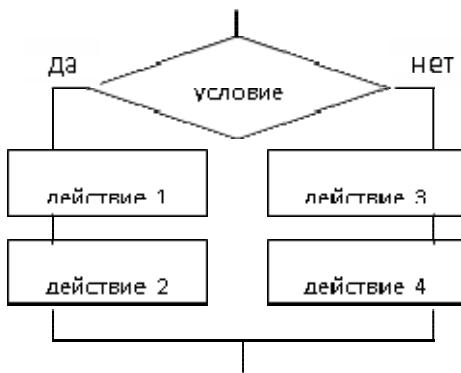


Рис. 2 Блок разветвления с полным выбором
выбором

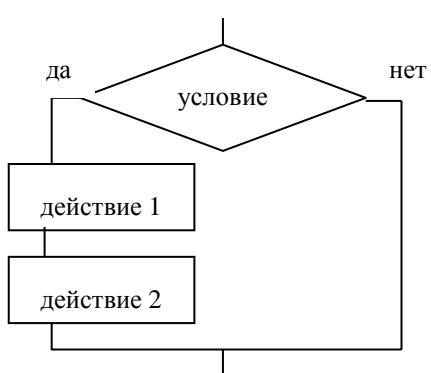


Рис. 3 Блок разветвления с неполным

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.

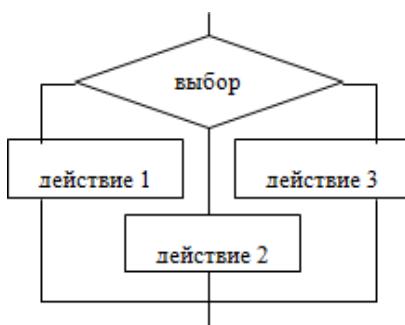


Рисунок 3– Блок множественного выбора

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.

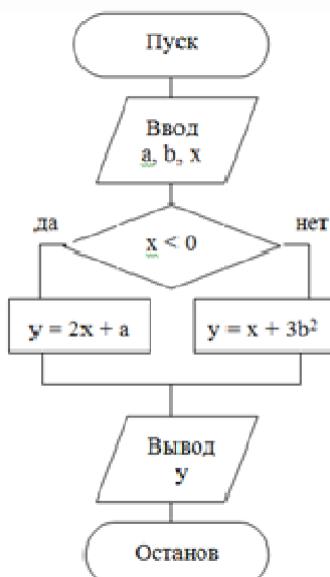


Рисунок 4 – Блок-схема алгоритма

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 в a , если $x = 0$,

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

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.

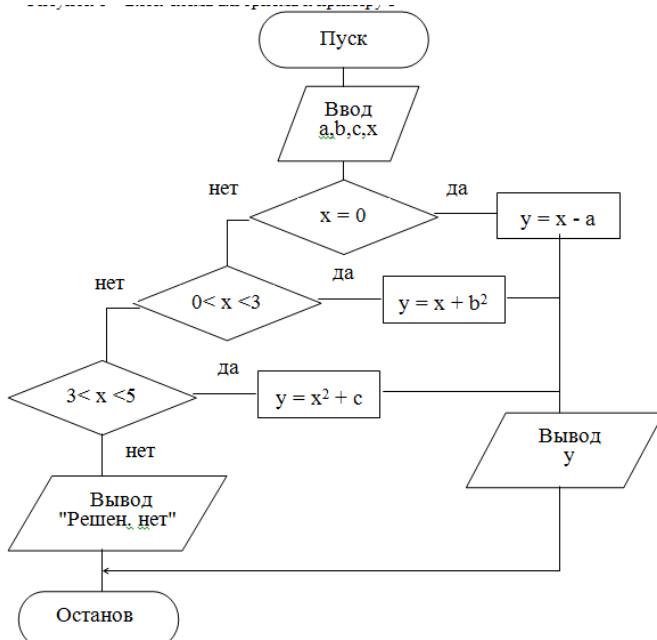


Рисунок 5 – Блок-схема алгоритма к примеру 3В заключение кратко подведем итог:

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.

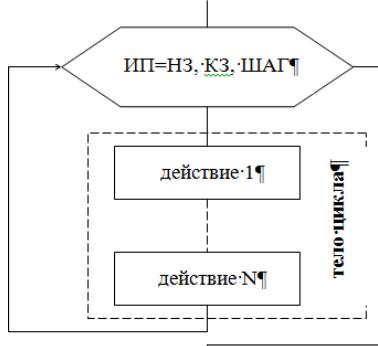


Рисунок 6 – Блок цикла с параметром

На рис. 6 приняты следующие сокращения:

ИП – имя ячейки памяти, в которую заносится значение параметра; Н3 – начальное значение параметра;

К3 – 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 Н3 до К3. 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.

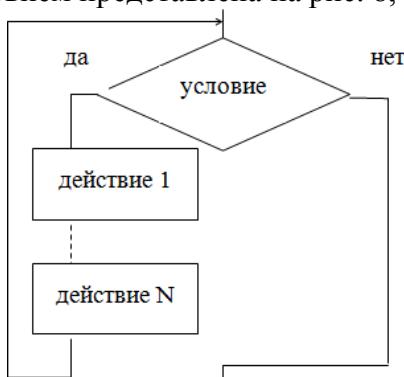


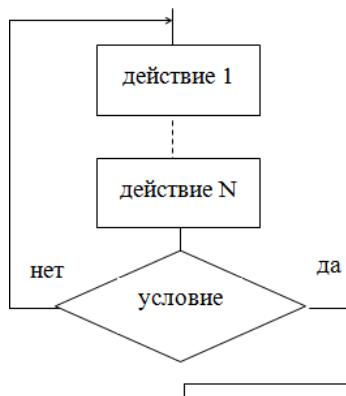
Рисунок 7 – Блок цикла с предварительным условием

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 –

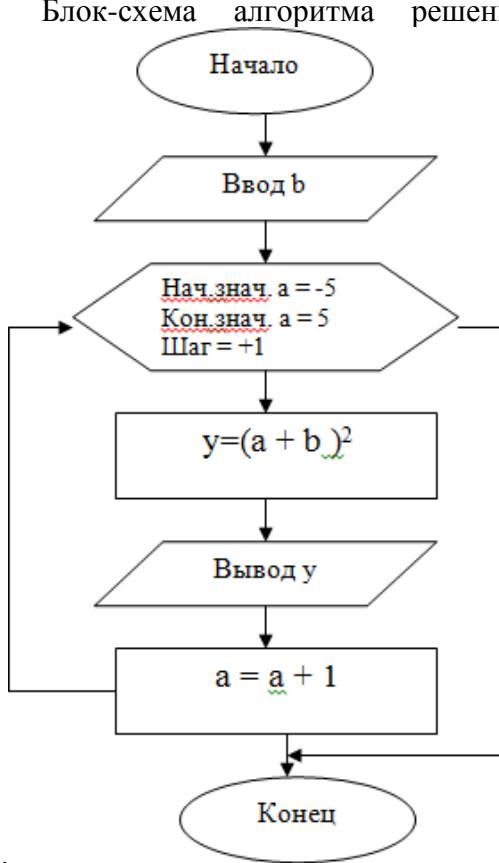


внутренним. Правила организации как внешнего, так и внутреннего циклов те же, что и для простых циклов. Параметры внешнего и внутреннего циклов должны быть разными.

Рисунок 8 - Блок цикла с последующим условием

Рассмотрим пример алгоритмизации циклического вычислительного процесса. Пример: Разработать алгоритм расчета значения функции по формуле $y=(a + b)^2$, при значениях a из интервала $[-5, 5]$ с шагом $+1$

Блок-схема алгоритма решения задачи на ЭВМ представлена на рис.



9.

Рисунок 9 – Блок-схема алгоритма

Подведем итог:
1. В циклическом вычислительном процессе получение результата обеспечивается путем многократного повторения некоторой последовательности

действий.

2. Для описания циклических вычислительных процессов в блок-схемах алгоритмов используют блоки циклов. Различают блок цикла с параметром, блок цикла с предварительным условием и блок цикла с последующим условием.

3. Циклические вычислительные процессы бывают простые и вложенные.

1.4 Лекция № 4 (4 часа).

Тема: «Программирование на языке высокого уровня»

1.4.1 Вопросы лекции:

1. Средства разработки программ на языке Pascal
2. Тип. Иерархия типов
3. Структура программы 4.Выражения. Оператор присваивания
5. Операторы повтора

1.4.2 Краткое содержание вопросов:

1. Средства разработки программ на языке Pascal

Язык программирования Pascal был разработан в конце 60-х годов XX века Николасом Виртом. Он разрабатывал этот язык как учебный язык для своих студентов. С тех пор Pascal, сохранив простоту и структуру языка, разработанного Н. Виртом, превратился в мощное средство программирования. С помощью современного языка Pascal можно производить простые расчеты, разрабатывать программы для проведения сложных инженерных и экономических вычислений.

Разработку программы можно разбить на следующие этапы:

1. Составление алгоритма решения задачи. Алгоритм – это описание последовательности действий, которые необходимо выполнить для решения поставленной задачи.

2. Написание текста программы. Текст программы пишут на каком-либо языке программирования (например на Pascal) и вводят его в компьютер с помощью текстового редактора.

3. Отладка программы. Отладка программы – это процесс устранения ошибок из текста программы. Все ошибки делятся на синтаксические и логические. При наличии синтаксических ошибок (ошибок в написании операторов) программа не запускается. Подобные ошибки исправляются проще всего. Логические ошибки – это ошибки, при которых программа работает, но неправильно, выдавая не те результаты, которые ожидает разработчик или пользователь. Логические ошибки исправить сложнее, чем синтаксические, иногда для этого приходится переписывать отдельные участки программы, а иногда перерабатывать весь алгоритм.

4. Тестирование программы. Тестирование программы – процесс выявления ошибок в ее работе.

Процессы отладки и тестирования сопровождаются неоднократным запуском программы на выполнение. Процесс запуска может быть осуществлен только после того, как введенная в компьютер программа на алгоритмическом языке Pascal будет переведена в двоичный машинный код и создан исполняемый файл.

Процесс перевода текста программы в машинный код называют трансляцией. Все трансляторы делятся на два класса:

- интерпретаторы – трансляторы, которые переводят каждый оператор программы в машинный код и по мере перевода операторы выполняются процессором;
- компиляторы переводят всю программу целиком, и если этот перевод прошел без ошибок, то полученный двоичный код можно запускать на выполнение.

Если в качестве транслятора выступает компилятор, то процесс перевода текста программы в машинный код называют компиляцией.

При переводе программы с языка Pascal в машинный код используются именно компиляторы.

Рассмотрим основные этапы обработки компилятором программы на языке Pascal.

1. Компилятор анализирует, какие внешние библиотеки нужно подключить, разбирает текст программы на составляющие элементы, проверяет синтаксические ошибки и в случае их отсутствия формирует объектный код. Получаемый на этом этапе двоичный файл (объектный код) не включает в себя объектные коды подключаемых библиотек.

2. На втором этапе компоновщик подключает к объектному коду программы объектные коды библиотек и генерирует исполняемый код программы. Этот этап называется компоновкой, или сборкой, программы. Полученный на этом этапе исполняемый код можно запускать на выполнение.

2. Тип. Иерархия типов

Тип – это множество значений переменной вместе с множеством операций, которые можно выполнять над элементами этого множества.

Приписывая переменной некоторый тип, мы тем самым явно определяем множество значений, которые можно присвоить этой переменной, а также операции, с помощью которых можно манипулировать ее значениями.

Число отдельных значений, принадлежащих некоторому типу, называется мощностью типа. Так, например, мощность типа Boolean есть 2.

В языке программирования Pascal принято все типы данных разделять на три группы:

1. простейшие, или скалярные (к скалярным типам данных относятся типы данных таких величин, значения которых не содержат составных частей, т.е. простейшие типы: целочисленный, вещественный, логический, символьный, перечисляемый, интервальный (тип диапазон) и ссылочный);

2. структурированные (структурный тип данных, или структуры данных, в своей основе имеет один или несколько скалярных типов данных, к ним относятся: строки, массивы, множества, записи, файлы);

3. процедурные и типа object. (этим типам данных трудно поставить в соответствие данные в обычном понимании этого слова. Их названия отражают их базовые признаки и используются, как правило, в объектно-ориентированном программировании, которое поддержано старшими версиями языка Pascal. Понимание работы с этими типами данных требует наличие определенного опыта и навыков программирования).

Рассмотрим некоторые скалярные (простейшие) типы данных.

1. Диапазон значений типа Byte: 0..255. Над данными типа Byte допустимы следующие операции:

сравнения ($=, <, <=, >, >=$);
+ (сложение); + (одноместный (унарный) плюс);
- (вычитание); - (одноместный (унарный) минус);
* (умножение); DIV (деление (получение частного));

MOD (получение остатка (второй операнд должен быть положительным)); ShL (логический сдвиг влево); ShR (логический сдвиг вправо).

При логическом сдвиге биты, оказавшиеся за пределами разрядной сетки кода, отбрасываются, а вместо недостающих устанавливаются нули.

2. Диапазон значений типа Integer: -32768..32767. Над данными типа Integer допустимы те же операции, что и над данными типа Byte.

3. Диапазон значений типа Word: 0..65535. Над типом Word допустимы те же операции, что и над данными типа Integer.

4. Диапазон значений типа Real: 2.9E-39..1.7E+38 (по абсолютной величине). Данные типа Real имеют точность до одиннадцати значащих цифр. Над данными типа Real допустимы следующие операции:

сравнения (=, <>, <, <=, >, >=);
+ (сложение); + (одноместный (унарный) плюс);
- (вычитание); - (одноместный (унарный) минус);
* (умножение); DIV (деление (получение частного));

5. Тип данных Char определяет множество символов с кодами ASCII. Над данными типа Char допустимы операции сравнения (=, <>, <, <=, >, >=).

6. Данные типа Boolean могут принимать два значения: TRUE и FALSE. Над данными типа Boolean допустимы следующие операции:

сравнения (=, <>, <, <=, >, >=);
AND (логическое И); OR (логическое ИЛИ);
XOR (логическое исключающее ИЛИ); NOT (логическое отрицание).

3. Структура программы

Программа на языке Pascal состоит из заголовка, раздела описаний и раздела операторов.

PROGRAM Имя (Input,Output); Заголовок программы label - раздел меток
const - раздел констант
type - раздел типов Раздел описаний var - раздел переменных
procedure - раздел процедур function - раздел функций BEGIN
Оператор 1;
Оператор 2; Раздел операторов... (тело программы) Оператор n END.

Заголовок содержит служебное слово PROGRAM, имя программы, задаваемое программистом, и в круглых скобках имена стандартных процедур Input, Output для связи программы с внешними устройствами ввода-вывода. Заканчивается заголовок символом ";" (точка с запятой).

Раздел описаний предназначен для объявления всех встречающихся в программе данных и их характеристик (имена данных, их тип, возможные значения). Этот раздел содержит следующие подразделы: объявление меток, констант, типов, переменных, объявление процедур и функций. Порядок расположения разделов не важен и описания могут повторяться.

Объявление процедур и функций является одним разделом. Следует заметить, что не все перечисленные разделы обязательны в программе. В простых программах могут потребоваться, например, только разделы описания констант и переменных.

После каждого описания ставится символ ";".

Раздел операторов ("тело" программы) заключается в операторные скобки вида: BEGIN ("начать") и END ("окончить"), при этом после служебного слова END обязательно ставится точка. В разделе операторов записывается последовательность исполняемых операторов и каждый выражает действие, которое необходимо выполнить. Исполняемые операторы отделяются друг от друга символом ";".

Хороший стиль программирования требует:

1. записывать слова PROGRAM, BEGIN, END с одинаковой позиции строки;
2. по отношению к ним описания и операторы принято сдвигать вправо на 3 символа. Желательно сдвиг делать на одинаковое количество позиций от края или по отношению к предыдущему сдвигу.

4. Выражения. Оператор присваивания

Выражения состоят из операций и операндов. Оператор присваивания-определяет процесс вычисления нового значения с помощью выражения и запоминание полученного значения переменной в ОП.

Большинство операций в языке Pascal являются бинарными, то есть содержат два операнда. Остальные операции являются унарными и содержат только один операнд. В бинарных операциях используется обычное алгебраическое представление, например: $a+b$. В унарных операциях операция всегда предшествует операнду, например: $-b$.

Операторы присваивания служат для вычисления значения выражения и присваивания его имени результата. Общий вид оператора присваивания:

$X := a$; где X – имя переменной; a – выражение.

Основные свойства присваивания:

1. Значение переменной не определено, если ей не присвоено никакого значения;
2. Новое значение, присваиваемое переменной, заменяет ее старое значение;
3. Присвоенное переменной значение сохраняется в ней вплоть до новогоприсваивания.

Для ввода используются следующие процедуры: Read (англ.) – читать
READ (a, b, ...); ReadLn (a, b, ...); ReadLn (a);

Для вывода используются следующие процедуры:

Write (англ.) – писать WRITE(a, b, ...);

WriteLn(a, b, ...);

WriteLn ('a=' , a);

Типы вводимых \ выводимых данных должны соответствовать типам переменных в списке процедуры ввода \ вывода.

В более сложных выражениях порядок, в котором выполняются операции, соответствует приоритету операций.

Старшинство операций

Операция	Приоритет	Вид операции
@, not	первый (высший)	унарная операция
*, /, div, mod, and, shl, shr	второй	операция умножения, деления, сдвига...
+, -, or, xor	третий	операция сложения
=, <>, <, >, <=, >=, in	четвертый (низший)	операция отношения

Для определении старшинства операций имеется три основных правила:

1. Во-первых, операнд, находящийся между двумя операциями с различными приоритетами, связывается с операцией, имеющей более высокий приоритет.
2. Во-вторых, операция, находящаяся между двумя операциями с равными приоритетами, связывается с той операцией, которая находится слева от него.
3. В-третьих, выражение, заключенное в скобки, перед выполнением вычисляется, как отдельный операнд.

5. Операторы повтора

Если в программе возникает необходимость неоднократно выполнить некоторые операции, то используются операторы повтора (цикла). В языке Pascal различают три вида операторов цикла: While, repeat, for.

Они используются для организации циклов различных типов. Выражение, управляющее повторениями, должно иметь булевский тип

(«Trune(Истина)/Flase(Ложь)», «Да(1)/Нет(0)»).

Если число повторений заранее неизвестно, а задано лишь условие его повторения (или окончания), используются операторы while, repeat. Оператор for используется, если

число повторений заранее известно.

Оператор повтора WHILE

Оператор while (пока) часто называют оператором цикла с предусловием за то, что проверка условия выполнения тела цикла производится в самом начале оператора. Формат записи:

While <условие продолжения повторений> do <тело цикла>;

Перед каждым выполнением тела цикла вычисляется значение выражения условия. Если результат равен True (Истина), тело цикла выполняется и снова вычисляется выражение условия. Если результат равен False (Ложь), происходит выход из цикла и переход к первому полю while оператору.

Оператор повтора REPEAT

Оператор повтора Repeat аналогичен оператору While, но отличается от него, во-первых, тем, что условие повторяется после очередного выполнения операторов тела цикла и таким образом гарантируется хотя бы однократное выполнение цикла, а во-вторых, что критерием прекращения цикла является равенство выражения константе True (Истина). За это цикл Repeat часто называют циклом с постусловием или циклом «До», так как он прекращает выполняться, как только значение выражения условия, записанного после слова Until равно True (Истина).

Формат записи:

Repeat

<оператор;>

... оператор >

Until <условие окончания цикла>;

Операторы, заключенные между словами Repeat и Until, являются телом цикла. Вначале выполняется тело цикла, затем проверяется условие выхода из цикла. Если результат булевского выражения равен False, то тело цикла активизируется еще раз; если результат True, происходит выход из цикла.

Оператор повтора FOR

В случаях, когда число повторений может быть заранее известно, то для организации циклической обработки информации применяется оператор повтора For. Часто этот оператор повтора называют оператором цикла с параметром, так как число повторений задается переменной, называемой параметром цикла, или управляющей переменной.

Оператор For состоит из заголовка и тела цикла. Формат записи:

For <параметр цикла> := <S1> to <S2> do <оператор>; или

For <параметр цикла> := <S1> downto <S2> do <оператор>;

Где S1 и S2 – выражения, определяющие соответственно начальное и конечное значения параметра цикла;

а) For ... do – заголовок цикла, который определяет диапазон изменения значений управляющей переменной и одновременно число повторений оператора, содержащегося в теле цикла, направление изменения значений параметра цикла (возрастание – to, убывание – downto).

б) <оператор> - тело цикла, которое может быть простым или составным оператором. Оператор For обеспечивает выполнение тела цикла до тех пор, пока не будут перебраны все значения параметра цикла от начального до конечного.

1.5 Лекция № 4 (4 часа).

Тема: «Компьютерные сети»

1.5.1 Вопросы лекции:

1. **Основные понятия.** Классификация компьютерных сетей

2. Локальные вычислительные сети

3. Глобальная вычислительная сеть

4. Требования, предъявляемые к современным вычислительным сетям

1.5.2 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.

Появление малых ЭВМ, микроЭВМ и, наконец, персональных компьютеров потребовало нового подхода к организации систем обработки данных, к созданию новых информационных технологий. Возникло логически обоснованное требование перехода от использования отдельных ЭВМ в системах централизованной обработки данных к распределенной (децентрализованной) обработке 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.

Несколько персональных ЭВМ, если они входят в состав вычислительной сети, могут выполнять значительно более широкий круг задач, чем каждая в отдельности. Использование сети позволяет ПЭВМ, подсоединенным к ней, обмениваться данными, организовывать коллективную обработку данных, а также совместно использовать программы, принтеры и прочие аппаратные средства.

В настоящее время вычислительные (компьютерные) сети являются высшей формой многомашинных ассоциаций.

Особенностью эксплуатации вычислительных сетей является не только приближение аппаратных средств непосредственно к местам возникновения и использования данных, но и разделение функций обработки и управления на отдельные составляющие с целью их эффективного распределения между несколькими ЭВМ, а также

обеспечение надежного и быстрого доступа пользователей к вычислительным и информационным ресурсам и организация коллективного использования этих ресурсов.

В условиях вычислительной сети обеспечивается возможность:

- организовать параллельную обработку данных многими ЭВМ;
- создавать распределенные базы данных, размещаемые в памяти различных ЭВМ;
- специализировать отдельные ЭВМ (группы ЭВМ) для эффективного решения определенных классов задач;
- автоматизировать обмен информацией и программами между отдельными ЭВМ и пользователями сети;
- резервировать вычислительные мощности и средства передачи данных на случай выхода из строя отдельных из них с целью быстрого восстановления нормальной работы сети;
- перераспределять вычислительные мощности между пользователями сети в зависимости от изменения их потребностей и сложности решаемых задач;
- стабилизировать и повышать уровень загрузки ЭВМ и дорогостоящего периферийного оборудования.

Как показывает практика, за счет расширения возможностей обработки данных, лучшей загрузки ресурсов и повышения надежности функционирования системы в целом стоимость обработки данных в вычислительных сетях не менее чем в полтора раза ниже по сравнению с обработкой аналогичных данных на автономных ЭВМ.

Прогресс в развитии микропроцессорной техники сделал ее доступной массовому потребителю, а высокая надежность, относительно низкая стоимость, простота общения с пользователем – непрофессионалом в области вычислительной техники послужили основой для организации систем распределенной обработки данных, базирующихся на ПЭВМ. Такие вычислительные сети могут объединять десятки, сотни, тысячи и более ПЭВМ.

В зависимости от территориального расположения абонентских систем вычислительные сети можно разделить на три основных класса:

- глобальные сети (WAN – Wide Area Network);
- региональные сети (MAN – Metropolitan Area Network);
- локальные сети (LAN – Local Area Network).

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.

Все рабочие станции в локальной сети могут использовать специализированные приложения, хранящиеся на сетевом сервере, и работать с общими устройствами: принтерами, факсами и другой периферией. ЛВС позволяют отдельным пользователям легко и быстро взаимодействовать друг с другом. Вот лишь некоторые задачи, которые позволяет решать ЛВС:

- совместная работа с документами;
- упрощение документооборота: руководитель получает возможность просматривать, корректировать и комментировать документы, не покидая своего рабочего места, не организовывая собраний и совещаний, отнимающих много времени;
- сохранение и архивирование созданных документов на сервере, чтобы не использовать ценнное пространство на жестком диске ПК;
- простой доступ к приложениям на сервере;
- облегчение совместного использования в организациях дорогостоящих ресурсов, таких как принтеры, накопители CD-ROM, жесткие диски и приложений (например, текстовые процессоры или программное обеспечение баз данных).

Для организации взаимодействия между сервером и рабочими станциями используются специальные аппаратные средства, основными из которых являются:

– адаптер (сетевой адаптер) – техническое устройство, выполняющее функции сопряжения ПЭВМ с каналами связи. Один адаптер обеспечивает сопряжение ПЭВМ с одним каналом связи. Адаптер часто называют сетевой картой. Он устанавливается непосредственно в один из свободных слотов (ISA, PCI, EISA или другому) материнской платы ПЭВМ и к нему на задней панели системного блока подстыковывается линия связи.

В большинстве из адаптеров предусмотрено использование микросхемы ПЗУ (ROM) удаленной загрузки. Это нужно в том случае, если компьютер рабочей станции не имеет НГМД или НЖМД. Загрузка операционной системы в память таких компьютеров осуществляется через сеть из сервера.

– концентратор (Hub) – это устройство множественного доступа, выполняющее роль центральной точки соединения в топологии "физическая звезда". Концентратор является основным узлом сети Ethernet на витой паре. Каждый компьютер должен быть подключен к нему с помощью сегмента кабеля. Длина каждого сегмента не должна превышать 100 м. На концах кабельных сегментов устанавливается разъем RJ-45.

Концентраторы выпускают на разное количество портов: как правило, 8, 12 или 16. К каждому порту может быть подключен либо персональный компьютер, либо другой концентратор.

– повторитель – устройство, обеспечивающее сохранение формы и амплитуды сигнала при передаче его на большее, чем предусмотрено данным типом физической

передающей среды, расстояние. Под передающей средой в данном случае понимается канал связи.

Существуют локальные и дистанционные повторители. Локальные повторители позволяют соединять фрагменты сетей, расположенные на расстоянии до 50 м, а дистанционные – до 2000 м;

– канал связи – это физическая среда, посредством которой происходит обмен данными между клиентами ЛВС. Канал связи может быть как проводным, так и беспроводным. Наиболее часто используются проводные линии связи. При этом обычно применяют четыре типа сетевых кабелей:

- незащищенная витая пара;
- защищенная витая пара;
- коаксиальный кабель;
- волоконно-оптический кабель.

Первые три типа кабелей передают электрический сигнал по медным проводникам. Волоконно-оптические кабели передают свет по стеклянному волокну. Большинство сетей допускает несколько вариантов кабельных соединений.

Витые пары представляют собой два провода, скрученных вместе шестью оборотами на дюйм для обеспечения защиты от электромагнитных помех и согласования импеданса или электрического сопротивления. Защищенные витые пары схожи с незащищенными, за исключением того, что они используют более толстые провода и защищены от внешнего воздействия слоем изолятора. Наиболее распространенный тип такого кабеля, применяемого в локальных сетях, представляет собой защищенный кабель с двумя витыми парами непрерывного провода.

Незащищенные витые пары чувствительны к электромагнитным помехам, например электрическим шумам, создаваемым люминесцентными светильниками и движущимися лифтами.

Задача и тщательное соблюдение числа повивов на дюйм делают защищенный кабель с витыми парами надежным альтернативным кабельным соединением. Однако эта надежность приводит к увеличению стоимости.

Коаксиальные кабели состоят из двух проводников, окруженных изолирующими слоями. Первый слой изоляции окружает центральный медный провод. Этот слой оплетен снаружи внешним экранирующим проводником. Наиболее распространенными коаксиальными кабелями являются толстый (около 10 мм в диаметре) и тонкий (около 4 мм) кабели "Ethernet". Такая конструкция обеспечивает хорошую помехозащищенность и малое затухание сигнала на расстоянии.

Обладая преимуществами по помехозащищенности, прочности, длине линий, толстый коаксиальный кабель дороже и сложнее в монтаже (его сложнее протягивать по кабельным каналам), чем тонкий.

Волоконно-оптические кабели передают данные в виде световых импульсов по стеклянным "проводам". Большинство систем локальных сетей в настоящее время поддерживает волоконно-оптическое кабельное соединение. Волоконно-оптический кабель обладает существенными преимуществами по сравнению с любыми вариантами медного кабеля. Волоконно-оптические кабели обеспечивают наивысшую скорость передачи; они более надежны, так как не подвержены потерям информационных пакетов из-за электромагнитных помех. Оптический кабель очень тонок и гибок, что делает его транспортировку более удобной по сравнению с более тяжелым медным кабелем. Однако наиболее важно то, что только оптический кабель имеет достаточную пропускную способность, которая в будущем потребуется для более быстрых сетей.

Пока еще цена волоконно-оптического кабеля значительно выше медного. По сравнению с медным кабелем монтаж оптического кабеля более трудоемок, поскольку концы его должны быть тщательно отполированы и выровнены для обеспечения надежного соединения. Однако ныне происходит переход на оптоволоконные линии,

абсолютно неподверженные помехам и находящиеся вне конкуренции по пропускной способности. Стоимость таких линий неуклонно снижается, технологические трудности стыковки оптических волокон успешно преодолеваются.

В настоящее время наибольшее распространение получили две разновидности ЛВС – одноранговая сеть и сеть типа клиент-сервер.

В одноранговой сети все компьютеры полностью равноправны. Такая сеть не имеет сервера. Некоторые аппаратные средства (например, винчестеры, приводы CD-ROM, сканеры, принтеры и т. д.), подключенные к отдельным ПЭВМ, используются совместно на всех рабочих местах.

Каждый пользователь одноранговой сети может определить права доступа другим пользователям к информации на своем компьютере. Механизмом ограничения прав пользователей является возможность блокировки доступа к дискам и другим периферийным устройствам, подключенным к его компьютеру.

Под сетью типа клиент-сервер понимают сеть, в "центре" которой находится мощный персональный компьютер (сетевой сервер), соединенный с отдельными рабочими станциями (клиентами). Клиенты используют ресурсы сервера, поэтому могут быть укомплектованы относительно недорогими внешними устройствами. Управление сетью, в смысле управления отдельными рабочими станциями, а также контроль за периферийными устройствами сети (модемы, факсы и т. д.) осуществляется специальным мощным сетевым программным обеспечением.

ЛВС в зависимости от назначения и технических решений могут иметь различную топологию (конфигурацию). Топология ЛВС отражает структуру связей между ее основными функциональными элементами – сервером и рабочими станциями (другими словами, топология ЛВС – это способ соединения компьютеров в сети). Наибольшее распространение получили четыре топологии ЛВС:

- кольцевая;
- звездообразная (радиальная);
- шинная;
- древовидная.

В кольцевой ЛВС (рисунок 2, а) информация передается по замкнутому каналу. Каждый абонент непосредственно связан с двумя ближайшими соседями, хотя в принципе способен связаться с любым абонентом сети. В сети с кольцевой топологией данные передаются от одного компьютера к другому как бы по эстафете. Если компьютер получит данные, предназначенные для другого компьютера, он передает их дальше по кольцу. Если данные предназначены для получившего их компьютера, они дальше не передаются.

Достоинством кольцевой топологии является то, что нет ограничений на длину всей сети, то есть имеет значение только расстояние между отдельными компьютерами.

Недостатки:

- время передачи данных увеличивается пропорционально числу соединенных в кольцо компьютеров;
- выход из строя одной рабочей станции может парализовать работу всей сети.

В звездообразной (радиальной) ЛВС (рисунок 2, б) в центре находится центральный управляющий компьютер (сервер), последовательно связывающийся с абонентами и связывающий их друг с другом.

Сеть с такой топологией имеет следующие достоинства:

- повреждение кабеля является проблемой для одного конкретного компьютера и в целом не сказывается на работе сети;
- надежный механизм защиты от несанкционированного доступа;
- высокая скорость передачи данных от рабочей станции к серверу.

Недостатками сети являются:

- мощность всей сети зависит, в первую очередь, от возможностей сервера;

– невозможна коммутация между отдельными рабочими станциями без сервера.

В шинной конфигурации (рисунок 2, в) компьютеры подключены к общему для них каналу (шине), через который могут обмениваться сообщениями. Аналогичным образом к шине подключаются и другие сетевые устройства.

В процессе работы сети информация от компьютера, передающего данные, поступает на все рабочие станции, однако воспринимается только той рабочей станцией, которой она адресована.

Достоинства сети с шинной топологией:

- высокое быстродействие
- относительно небольшие затраты на кабели;
- рабочие станции могут коммутироваться друг с другом без помощи сервера;
- повышенная устойчивость к возможным неисправностям отдельных узлов.

В качестве недостатков можно выделить следующее:

- при обрыве кабеля выходит из строя весь участок сети от места разрыва;
- возможность несанкционированного подключения к сети.

В древовидной конфигурации (рисунок 2, г) существует "главный" компьютер, находящийся на вершине иерархической структуры. Ему подчинены компьютеры следующего уровня, и т. д.

Выбор той или иной топологии определяется областью применения ЛВС, географическим расположением ее узлов и размерностью сети в целом.

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 вебовских 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.

По имеющимся оценкам, в 1999 году число российских пользователей Сети превысило 5 миллионов. Сейчас в Интернете есть уже много интересных материалов на русском языке, но знание английского языка желательно - пионерская роль стран английского языка в развитии Интернета закрепила за английским роль языка межнационального общения.

Принципы и организация сети Интернет. Серверы, клиенты и протоколы IP-адресация

Интернет обладает некоторыми чертами почты, некоторыми чертами телеграфа и некоторыми чертами телефона. Так же как в телеграфе, в Интернете используется цифровая передача информации. Как в телефонной сети каждому телефону присваивается телефонный номер, так и каждому компьютеру в Интернете присваивается свой номер, который называется IP-адресом. Только в Интернете, в отличие от телефона, нет путаницы с локальными номерами и междугородними кодами: каждый IP-адрес имеет длину ровно 32 бита и записывается обычно как четыре десятичных числа (от 0 до 255), - например, 62.76.161.102. Это глобальная нумерация - каждый компьютер, подключенный к Интернету, имеет уникальный IP-адрес.

Наиболее глубокая аналогия существует между Интернетом и обычной почтой. В

данном случае речь идет о том, что информация по Интернету передается в виде отдельных пакетов. Если нужно передать длинное сообщение, оно разбивается на нужное число кусочков, и каждый из них снабжается адресом отправителя, адресом получателя и некоторой служебной информацией. Каждый пакет передается по Интернету независимо от всех остальных и, в принципе, они могут следовать разными маршрутами. По прибытии пакетов на место из них собирается исходное сообщение. Это называется коммутацией пакетов.

Одним из главных преимуществ режима коммутации пакетов это - эффективное использование общих коммуникационных ресурсов. В Интернете каждый компьютер может одновременно принимать пакеты от большого количества других компьютеров. При этом возможны перегрузки коммутационных узлов (серверов) в результате большого количества информации. Однако все пакеты, пусть с небольшой задержкой, все равно дойдут до адресата в порядке своей очереди. В то же время, если в данный момент вы никакой информации не посыпаете, то вы не потребляете никаких ресурсов сети, и тем самым можете находиться «на линии» сколь угодно долго, не создавая помех другим.

Для обеспечения жизнеспособности всей сети необходимо, чтобы обмен информацией между различными ее блоками или отдельными компьютерами велся на основе общепринятых стандартов. Набор формальных правил о том, как и в каком виде следует передавать данные между различными устройствами и программами, называется протоколом. Протокол позволяет корректно взаимодействовать программам, написанным разными авторами для разных типов компьютеров и операционных систем.

Интернет использует протокол TCP/IP. Этот протокол регламентирует, как следует разбивать длинное сообщение на пакеты, как должны быть устроены пакеты, как контролировать прибытие пакетов к месту назначения, что делать в случае ошибок передачи данных, и другие детали.

Собственно, Интернет представляет собой объединение десятков тысяч отдельных сетей, которые используют протокол TCP/IP и единое пространство IP-адресов. В остальном эти сети административно и финансово независимы.

Еще одно важное достоинство коммутации пакетов - это легкость объединения в единую сеть разных по скорости каналов связи. В связи с этим скоростные характеристики подключения к Интернету могут варьироваться в очень широких пределах, и разница будет заключаться лишь в скорости получения информации. Никаких других качественных отличий или ограничений не будет.

Доменная система имен и указатели ресурсов

Хотя люди уже привыкли пользоваться цифровыми номерами абонентов (например, звоня по телефону), все-таки имена, которые можно произнести, легче запоминаются и более удобны для использования. У большинства компьютеров в Интернете есть собственное имя, а не только IP-адрес. Служба, которая обеспечивает перевод имен компьютеров в их IP-адреса, называется Доменной Службой Имен (DNS). Это что-то вроде гигантского, распределенного по многим компьютерам телефонного справочника, с IP-адресами вместо телефонов.

Имя компьютера записывается как несколько слов, разделенных точками, например: `iomas.vsau.ru`. Это отражает иерархическую, или доменную, структуру службы DNS. В нашем примере “`iomas`” - это имя компьютера в домене (второго уровня) “`vsau`”, который принадлежит домену (первого уровня) “`ru`”. Администратор, который отвечает за домен первого уровня “`ru`” (Россия), зарегистрировал домен второго уровня “`vsau.ru`” (Воронежский агроуниверситет) и передал туда все полномочия на регистрацию новых имен в пределах этого домена. В свою очередь администратор домена “`vsau.ru`” зарегистрировал имя “`iomas.vsau.ru`” за определенным IP-адресом. Такая структура службы DNS обеспечивает, с одной стороны, уникальность имен компьютеров в пределах всего Интернета, а с другой стороны, четкое разделение административной ответственности.

DNS - это особая служба Интернета, потому что она используется всеми остальными службами, от telnet до www. Перевод имен DNS в IP-адреса происходит автоматически. Для этого надо только указать вашему компьютеру IP-адрес сервера DNS - того компьютера, которому будут направляться соответствующие запросы.

Хотя не существует особых правил, как следует называть домены, в применении к доменам первого, самого верхнего уровня сложилась определенная практика. Международные организации и США используют домены первого уровня com - для коммерческих, org и net - для некоммерческих организаций. В большинстве стран существует один домен первого уровня для страны: ru - для России, de - для Германии, uk - для Великобритании и т.д.

Так же, как каждый компьютер имеет свое уникальное имя, уникальное имя имеет и каждый документ в Интернете. Это уникальное имя называется URL - Универсальный Указатель Ресурса (Universal Resource Locator). URL имеет следующую форму:

служба://имя_компьютера/директория/поддиректория/.../имя_файла (например: <http://iomas.vsau.ru/people/peopl3.htm>).

Служба обозначается соответствующим протоколом, чаще всего вы встретите http:// для веб-страниц и ftp:// для файловых архивов. Обратите внимание, что используется "прямая" косая черта - "/", а не "обратная" - "\". Также нужно иметь ввиду, что в названиях директорий, поддиректорий и файлов большие и малые буквы различаются.

Серверы, клиенты и протоколы

Известно, что развитие общества связано с разделением труда. Разделение труда выгодно использовать и между людьми, и между компьютерами. Это очевидно: всегда лучше, когда каждый занимается своим делом.

В Интернете есть два «сорта» компьютеров - серверы и клиенты. Серверы - это серьезные, надежные машины. Они работают 24 часа в сутки, 7 дней в неделю. Они постоянно соединены с Интернетом и готовы предоставлять сервис - доставлять документы или пересыпать почту, отвечая при этом на десятки и сотни запросов одновременно. От их работоспособности зависит работа того множества компьютеров и локальных сетей, которые подключены к ним. Поэтому серверы и сопутствующая узловая коммуникационная аппаратура защищены от сбоев электропитания источниками бесперебойного питания (UPS). Чаще всего они работают под управлением того или иного варианта операционной системы UNIX. Серверы также называют хостами или сайтами, особенно применительно к веб-сайтам.

Клиенты - это те персональные компьютеры, за которыми сидят пользователи, то есть мы с вами. Сейчас это чаще всего компьютеры под управлением операционной системы Microsoft Windows различных версий. Здесь нет таких требований к надежности - в случае сбоя на вашем компьютере никто кроме вас от этого не пострадает. Зато взаимодействие клиентского компьютера с пользователем отлажено и стандартизовано. Часто такой компьютер не соединен с Интернетом постоянно, а подключается к нему по мере необходимости.

Та же самая терминология относится и к программам - существует клиентское программное обеспечение, которое взаимодействует с пользователем и формирует запрос, и серверное программное обеспечение, которое отвечает на такие запросы. Эти запросы формируются в соответствии с определенным протоколом.

Виды сервиса глобальной сети Интернет и программное обеспечение

Способы использования Интернета определяются предоставляемыми посетителям сервисами, в число которых входят традиционные, специальные и новые виды сервиса.

К традиционным видам сервиса относятся:

- электронная почта (e-mail);
- телеконференции (news);
- живое общение (chat);

- использование вычислительных и информационных ресурсов компьютера в режиме удаленного терминала (telnet);
- служба хранения, поиска и пересылки файлов - FTP (File Transfer Protocol);
- WWW (World Wide Web) - всемирная паутина (в ее составе в июле 1993г. было 130 хост-компьютеров, а в июне 1996г. 230000).

К специальным сервисным средствам относятся:

- информационная система Gopher;
- информационная система WAIS, ведущая поиск по ключевым словам,
- поисковые системы WWW;
- библиографические (диспетчерские) системы;
- Archie - поисковая система FTP;
- и др.

Новые виды сервиса включают в себя:

- системы для заказа товаров по сети (цветы, пицца, авиабилеты, номер в гостинице и т.д.);
- библиотечные системы,
- электронные издания газет и журналов, блоги;
- сервисные представительства фирм,
- коммерческие информационные системы по производителям товаров и услуг, котировкам акций на фондовых биржах;
- и др.

Всех работающих в Интернете можно разделить на две группы: тех, кто предоставляет различные интернет-услуги (провайдеров Интернета), и тех, кто эти услуги использует (пользователей Интернета).

Обучение пользованию Интернетом в основном сводится к общему ознакомлению с устройством Интернета, с предоставляемыми сервисами, адресацией ресурсов, с поиском информации в сервисе WWW и основами работы с электронной почтой.

Провайдеры Интернета по составу предоставляемых услуг делятся на три группы:

- провайдеры, предоставляющие доступ в Интернет - ISP (Internet Service Providers). Примером могут служить www.rol.ru, www.mtu.ru, и др.;
- провайдеры, предоставляющие услуги присутствия в Интернет - IPP (Internet Presence Providers), примером которых могут служить www.mail.ru, www.narod.ru, и др. К предоставляемым ими услугам относятся электронный почтовый ящик и услуги почты, размещение сайта клиента на ЭВМ провайдера, различные поисковые системы, новости, и т.д.;
- издатели собственных материалов - PCP (Private Content Publisher), которые содержат (т.е. разрабатывают и эксплуатируют) свои информационные системы (например, www.garant.ru, www.rbc.ru и т.п.), электронные магазины, рекламные агентства, туристические фирмы, и т.д.

ISP (Internet Service Provider)

ISP - это поставщик услуг Internet, т.е. организации или частные лица, предоставляющие доступ в Internet (hosting). Источником доходов ISP являются владельцы локальных ЭВМ, которым предоставляется доступ к данной глобальной вычислительной сети. Через имеющиеся шлюзы локальные пользователи могут получить доступ к другим глобальным сетям и таким образом получить возможность работать в Internet.

ISP подключены к Internet постоянно и имеют постоянный IP-адрес (IP-адрес является частью URL). Остальные пользователи (клиенты) могут подключаться к ISP лишь на время работы. IP-адрес присваивается им ISP каждый раз при подключении, а при отключении - отбирается и может быть отдан кому-нибудь другому. ISP часто предоставляет своим клиентам удаленный доступ по коммутируемым каналам телефонной связи (это называется "dual-up service"). Для этого ISP арендует у местной

телефонной компании телефонные номера, по которым с ним можно связаться.

ISP иногда может предоставлять (делегировать) функции хостинга локальным ЭВМ (которые при этом получают свой постоянный IP-адрес), например, стоящим дома или на работе, и превращать таким образом локальную ЭВМ в ISP более низкого уровня. В свою очередь, локальная хост-ЭВМ может делегировать такие права другой локальной ЭВМ, которая становится хост-провайдером (т.е. ISP) еще более низкого уровня (если она, конечно, имеет доступ к каналам связи - например, через учрежденческую АТС). Образуется цепочка провайдеров, различающихся своими IP-адресами: mesi.ru; ex.mesi.ru; stud.ex.mesi.ru и т.д.

Широко известными интернет-сервис-провайдерами являются, например, mtu.ru и rol.ru.

IPP (Internet Presence Provider)

IPP - это провайдер, обеспечивающий своим клиентам присутствие в Internet. Он так же подключен к Internet постоянно и имеет постоянный IP-адрес. В отличие от ISP, он не предоставляет услуг типа dual-up service. Он может только размещать на своих серверах публикации других лиц, рекламу, веб-сайты, организовывать работу электронной почты и т.д.

IPP после регистрации на их сайте предоставляют имя, которое будет являться наименованием вашего почтового ящика (e-mail), и кроме того - именем вашего сайта, который размещается на ЭВМ провайдера.

Например, после регистрации на mail.ru предоставляется почтовый ящик имя@mail.ru. Адрес вашего сайта будет: <http://www.mail.ru/~имя>.

Такое имя сайта не удовлетворяет многих пользователей. У провайдера www.narod.ru имя сайта выглядит по-другому: <http://www.имя.narod.ru>. Получается, что ваш сайт имеет имя в домене III уровня, а не где-то среди каталогов IPP. Это выглядит солиднее, похоже на то, что у вас есть свой компьютер с постоянным IP-адресом.

Широко известными провайдерами присутствия в Интернете являются:

- <http://www.mail.ru>;
- <http://www.yandex.ru>;
- <http://www.narod.ru>;
- <http://www.rambler.ru> и др.

PCP (Private Content Publisher)

PCP - это издатель собственных материалов. Он является участником межсетевого обмена (провайдером услуг), который готовит информацию для размещения в Internet, размещает ее, как правило, на своих компьютерах и постоянно обновляет ее. На определенных условиях он разрешает пользоваться своими материалами клиентам, приходящим из Internet.

Он может содержать базу телефонов города, справочники различного назначения. Такой провайдер является квалифицированным источником информации. Обычно его базы данных относятся к области, в которой он является квалифицированным специалистом.

Этот тип провайдера очень бережно относится к своей информации, к исправности своего компьютера, и у него всегда можно получить самые достоверные, постоянно обновляемые данные по выбранной им специальности.

Пример этого типа провайдера - <http://www.garant.ru> (юридические документы).

В отличие от ISP, он не предоставляет услуг dual-up service. Подготовка провайдеров Интернета предусматривает изучение таких дисциплин, как:

- общий курс по работе в сети Интернет;
- техническое обеспечение глобальных вычислительных сетей;
- программирование на специальных алгоритмических языках;
- операционные системы для глобальных вычислительных сетей;
- программное обеспечение глобальных вычислительных сетей;

- интернет-протоколы;
- инструментальные средства для исследования глобальных вычислительных сетей;
- WWW-программирование на стороне клиента;
- WWW-программирование на стороне сервера;
- разработка веб-сайтов;
- веб-дизайн;
- администрирование узлов глобальных вычислительных сетей.

Характеристики хостинг-провайдеров

Размещение сайтов на чужой технической базе называется хостингом. Хостинг бывает платный и бесплатный.

При платном хостинге оговаривается состав услуг, предоставляемых провайдером:

- провайдер предоставляет все: канал, ЭВМ, URL-адрес, обслуживание сайта и т.д.;
- провайдер предоставляет площадь для размещения вашей ЭВМ, канал, URL, обслуживание, и т.д.;
- и т.д.

При бесплатном хостинге провайдер предоставляет URL, место на своих магнитных носителях, сервисные программы для создания и обслуживания сайта. Но взамен размещает свою баннерную рекламу на вашем сайте.

Хостинг-провайдеры характеризуются:

- техническими ресурсами, к которым относятся ЭВМ, каналы связи, маршрутизаторы, которые определяют такую характеристику, как скорость отклика.

Скорость отклика сайта - это характеристика провайдера, которая связана с загрузкой его каналов. Например, провайдер имеет выход в Интернет, соединяясь по радиоканалу с другим провайдером, имеющим спутниковую связь. Канал может быть перегружен, и для связи с Интернетом через такого провайдера может понадобиться большое время. Большое значение также имеет связь локальных ЭВМ с провайдером (телефонный канал, количество телефонов для подключения локальных ЭВМ и др.);

- предоставляемыми услугами, которые могут быть весьма специфичными. При их выборе необходимо понимать, что услуга вам дает и что от вас требуется для ее использования.

Например, предоставление вам ЭВМ провайдера - и предоставление вам площадки для установки вашей ЭВМ различаются тем, что если вы устанавливаете у провайдера свою ЭВМ, то ее никто из сотрудников провайдера не имеет права трогать (но в договоре можно оговорить, что ремонт отказавшей ЭВМ проводится сотрудниками провайдера) - это ваша ЭВМ, тогда как если вам предоставлена ЭВМ провайдера, то ее могут заменить на другую без согласования с вами;

- программными ресурсами провайдера: какая операционная система используется (80% провайдеров работает под Unix), какое программное обеспечение используется для сервиса WWW (чаще всего - сервер Apache), какие виды сервиса разрешены и обеспечиваются провайдером (например, часто запрещаются ftp и telnet).

Программные ресурсы провайдера характеризуются также программным обеспечением, предоставляемым клиентам (компиляторы, интерпретаторы, СУБД, предустановленные скрипты, управляющие интерфейсы (мастера и шаблоны), и т.д.). Этот состав очень важен для размещения созданного сайта у хостинг-провайдера, так как провайдер может плохо относиться к продукции фирмы и не поддерживать конструкции FrontPage и другие расширения операционной системы Windows, а для клиента это программное обеспечение является основным;

- системой безопасности провайдера: для пользователя имеет значение наличие зеркальных серверов, резервных мощностей.

Зеркальный сервер - это дублирующий сервер, содержащий ту же самую

информацию, что и основной. Такие зеркала нужны для увеличения надежности системы и ее пропускной способности. Зеркальные серверы могут устанавливаться в различных частях света для того, чтобы не загружать глобальные вычислительные сети. Они имеют одно и то же имя, но службы DNS отправляют посетителей на ближайший из них.

Резервная мощность - это запасные технические средства, которые находятся в резерве (холодном, теплом, горячем). При отказе какой-либо ЭВМ провайдер переключает работу на резервную ЭВМ;

- службой технической поддержки, обеспечивающей помощь клиентам и снабжающей клиентов технической документацией по предоставляемым сервисам.

Эта служба ведет разработку техдокументации для клиентов, проводит мониторинг сайта и оценку его эффективности, резервное копирование сайта (backup) и др.;

- сопутствующими услугами, предоставляемыми хостинг-провайдером, такими как регистрация доменных имен, бонусами (т.е. скидками или дополнительными услугами, предоставляемыми некоторым клиентам), возможным освобождением вашего сайта от размещения чужих баннеров, возможностью подключения до-полнительных услуг, таких как сервис Spylog, и др.; условиями и ограничениями: ограничениями трафика (установлением разной платы за работу в Рунете и в Интернете); установкой лимита на скачивание информации с локальной ЭВМ (например, 1 Гбайт в месяц) - и взиманием дополнительной платы за превышение трафика; ограничением объема предоставляемой памяти; запретом на хранение информации, не связанной гиперссылками с основным содержанием сайта; с условиями типа: "мы не несем ответственности за потерю информации клиента", - и т.д.

Программное обеспечение Интернета

Каждый сервис требует своего программного обеспечения (в общем виде структура программного обеспечения Internet приведена на рис. 11).

Интернет построен на основе архитектуры "клиент-сервер". В сетях этого типа выделяется мощный хост-компьютер (или даже несколько хост-компьютеров), на который ставится серверное программное обеспечение.

На клиентских ЭВМ устанавливается клиентское программное обеспечение.

Хост-ЭВМ постоянно включены, имеют постоянные IP-адреса. Клиентские ЭВМ включаются по мере необходимости, связываются с серверным программным обеспечением хост-ЭВМ, получают от него временный IP-адрес, действующий только в пределах данного сеанса связи.

Глобальные вычислительные сети имеют узлы (хосты), на которых устанавливается серверная часть программного обеспечения сервисов Интернета. Серверное и клиентское программное обеспечение взаимодействуют между собой.

Из рис.1 видно, что программное обеспечение Интернета состоит из трех видов программ: серверное ПО, клиентское ПО и ПО систем безопасности. Серверное ПО устанавливается на хост-компьютерах, клиентское - на локальных ЭВМ (т. е. на ЭВМ клиентов). ПО систем безопасности может не соответствовать архитектуре "клиент-сервер".

Системы безопасности либо устанавливаются на клиентской ЭВМ или только на хост-ЭВМ, либо для них выделяется отдельная ЭВМ, на которой устанавливаются специализированные программы, обеспечивающие безопасность, - такие программы называются "брандмауэрами", или

Рис. 11. Структура программного обеспечения Internet (общий вид)

"firewall" (эти названия заимствованы у пожарных, которые требуют, чтобы при строительстве длинных домов производилось их разделение на части и одна часть от другой отделялась каменной стеной, основное назначение которой - не допустить распространения огня на всю постройку при возгорании одной из ее частей). Программы

безопасности являются аналогами такой стены между ЭВМ пользователей и Интернетом. Они могут быть настроены так, чтобы полностью разрывать связь между ЭВМ и Интернетом, могут допускать одностороннюю связь (например, разрешена только отправка почты с локальной ЭВМ) или разрешать двустороннюю связь только для определенных видов работ. Кроме того, может быть запрещено выполнение каких-либо действий.

Входная и выходная информация в таких программах проходит через фильтры, которые, например, могут быть настроены на выявление вирусов, на пропуск файлов, не превышающих заданных размеров, или файлов определенного типа, на запрет связи с определенными IP-адресами и т.д.

В клиентском программном обеспечении необходимо обратить внимание на VRML-браузеры, позволяющие просматривать виртуальные миры, работать в трех измерениях (в двух измерениях по экрану можно перемещаться вверх-вниз и вправо-влево; в трех измерениях, кроме этого, добавляется возможность приближаться к экрану или удаляться от него). VRML-браузеры - это клиентское программное обеспечение. В Интернете есть несколько сайтов с серверами VRML и своеобразными виртуальными мирами.

Как на клиентских ЭВМ, так и на хостах могут быть размещены программы, расширяющие возможности серверов и клиентов. Для написания таких программ применяются специальные алгоритмические языки: HTML, PHP, система программирования CGI, Java, Java-script, Perl, SSI и др.

При использовании таких программ на сервере должно быть установлено соответствующее программное обеспечение.

Программирование глобальных вычислительных сетей - сложное направление, в котором применяются приведенные выше алгоритмические языки на основе правил работы, определяемых протоколами TCP/IP и соответствующих сервисов Интернет

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.

Прозрачность сети предполагает скрытие (невидимость) особенностей сети от конечного пользователя. Пользователь обращается к ресурсам сети как к обычным локальным ресурсам компьютера, на котором он работает.

Компьютерная сеть объединяет компьютеры разных типов с разными операционными системами. Пользователю, у которого установлена, например, Windows, прозрачная сеть должна обеспечивать доступ к необходимым ему при работе ресурсам компьютеров, на которых установлена, например, UNIX. Другой важной стороной прозрачности сети является возможность распараллеливания работы между разными элементами сети. Вопросы назначения отдельных параллельных заданий отдельным устройствам сети также должны быть скрытыми от пользователя и решаться в автоматическом режиме.

Интегрируемость

Интегрируемость означает возможность подключения к вычислительной сети разнообразного и разнотипного оборудования, программного обеспечения от разных производителей. Если такая неоднородная вычислительная сеть успешно выполняет свои функции, то можно говорить о том, что она обладает хорошей интегрируемостью.

Современная компьютерная сеть имеет дело с разнообразной информацией, процесс передачи которой сильно зависит от типа информации. Передача традиционных компьютерных данных характеризуется неравномерной интенсивностью. При этом нет жестких требований к синхронности передачи. При передаче мультимедийных данных качество передаваемой информации в существенной степени зависит от синхронизации передачи. Существование двух типов данных с противоположными требованиями к

процессу передачи является сложной задачей, решение которой является необходимым условием вычислительной сети с хорошей интегрируемостью.

Основным направлением развития интегрируемости вычислительных сетей является стандартизация сетей, их элементов и компонентов. Среди стандартов различных видов можно выделить стандарты отдельных фирм, стандарты специальных комитетов, создаваемых несколькими фирмами, стандарты национальных организаций по стандартизации, международные стандарты.

3 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

2.1 Лабораторная работа № 1 (2 часа).

Тема: «Системное программное обеспечение»

2.1.1 Цель работы: освоение работы в операционной системе Windows.

2.1.2 Задачи работы:

1. Изучить возможности операционной системы WINDOWS.
2. Освоить основные операции с объектами WINDOWS.
3. Освоить основные приемы работы со стандартными приложениями WINDOWS.
4. Для закрепления и проверки полученных навыков, необходимо ответить на контрольные вопросы. Результаты работы продемонстрировать преподавателю в электронном виде.

2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер

2.1.4 Описание (ход) работы:

Операционная система WINDOWS. Операции с объектами WINDOWS.

1. Теоретическая часть

Обзор возможностей операционной системы WINDOWS

ОС Windows является 32-разрядной операционной системой. С ее появлением многие сложности и проблемы, связанные с 16-разрядными операционными системами остались позади. При этом ОС Windows обеспечивает совместимость с Windows 3.1 и DOS. т.е. в среде ОС Windows можно использовать множество уже существующих приложений, созданных для персональных компьютеров класса IBM PC.

ОС Windows имеет усовершенствованный интерфейс пользователя. Главная цель здесь предоставление любому пользователю, имеющему минимальные знания о компьютере, возможности сесть и запустить практически любое приложение без обучения. Главная особенность оконного интерфейса состоит в том, что он обеспечивает на экране модель «рабочего стола». Окна на экране являются эквивалентами разбросанных на столе бумаг. Выбирая окно, можно сделать его активным, т.е. поместить его поверх остальных, как бумаги на столе окна можно перемещать и перекладывать окна на экране. В продолжении аналогий с рабочим столом Каталоги теперь называются Папками (Folder). Все файлы сосредоточены в Папках.

Для быстрого доступа к наиболее часто используемым файлам или устройствам в ОС Windows используются объекты, получившие названия Ярлыков (Shortcut). Ярлык к программе, документу, устройству может быть создан в любой папке или на Рабочем столе.

Рабочий стол ОС Windows снабжен кнопкой Пуск (Start) и имеет специальную область Панель задач (Taskbar). Кнопка Пуск предназначена для быстрого запуска программ. Панель задач позволяет очень просто переключаться между приложениями запущенными в среде ОС Windows.

ОС Windows позволяет присваивать длинные имена любому файлу. В именах файлов можно использовать пробелы, предельная длина имени файла составляет 255 символов. Имена файлов не должны содержать следующие символы : « \ », « / », « : », « *

», « ? », « " », « < », « > ».

В ОС Windows реализован механизм вытесняющей многозадачности, который позволяет одновременно запускать несколько приложений. Эти приложения теперь выполняются быстрее.

ОС Windows снабжена встроенной системой Plug-and-Play. Технология Plug-and-Play предоставляет автоматизировать процесс установки дополнительного оборудования. Пользователь физически устанавливает оборудование, операционная система сама отыскивает необходимые драйвера и настраивает работу устройства в среде ОС Windows. Специализированные средства Мастера (Wizards) помогают выполнять операции по настройке оборудования.

Для удобства управления программами и файлами в ОС Windows появились системные программы Мой компьютер (My Computer), Сетевое окружение (Network Neighborhood), Корзина (Recycle Bin). Мой компьютер объект, который визуализирует структуру устройств, папок и файлов компьютера и позволяет проводить операции над ними. Для реализаций этих операций используется прием Drag-and-Drop (потащить и бросить). Перетаскивать можно любые объекты (окна, значки элементов, папки, файлы). Для этого мышкой выбирается объект, нажимается левая кнопка, удерживая кнопку, перемещаем указатель мыши вместе с выбранным объектом в нужное место и отпускаем клавишу мыши.

Программа Сетевое окружение обеспечивает доступ к сетевым ресурсам компьютера. Корзина - это место для временного помещения удаленных файлов. Через обращение к Корзине можно восстанавливать ошибочно удаленные файлы.

Рабочий стол Windows

Легко представить рабочий стол как чистый лист, находящийся за всеми остальными изображениями на дисплее, однако рабочий стол является гораздо большим, нежели просто фон. Помимо изменяемого фона рабочий стол Windows 95 содержит папки, значки и другие объекты (см. рис.1). Для удобства управления программами и файлами в ОС Windows появились системные программы Мой компьютер (My Computer), Сетевое окружение (Network Neighborhood), Корзина (Recycle Bin). Мой компьютер объект, который визуализирует структуру устройств, папок и файлов компьютера и позволяет проводить операции над ними. Для реализаций этих операций используется прием Drag-and-Drop (потащить и бросить). Перетаскивать можно любые объекты (окна, значки элементов, папки, файлы). Для этого мышкой выбирается объект, нажимается левая кнопка, удерживая кнопку, перемещаем указатель мыши вместе с выбранным объектом в нужное место и отпускаем клавишу мыши.

Программа Сетевое окружение обеспечивает доступ к сетевым ресурсам компьютера, если данный ПК подсоединен к сети. Эти сетевые ресурсы включают диски и принтеры, общие для всех компьютеров в сети. Корзина - это место для временного помещения удаленных файлов. Через обращение к Корзине можно восстанавливать ошибочно удаленные файлы.

Панель задач также является частью рабочего стола.

В следующих разделах содержится обзор объектов, являющихся составными частями рабочего стола.

Папки

Папки являются средством организации и представления системных ресурсов ПК (каталогов, файлов, программ и т.д.). Папка может содержать другие папки (вложенные папки), программы, а также такие объекты, как, например, принтеры и диски. Объекты в папке представляются значками и каждый значок имеет название, расположеннное ниже него. Для того, чтобы открыть папку, запустить программу, открыть документ или активизировать и открыть объект любого другого типа в папке, достаточно дважды щелкнуть на соответствующем значке. Можно создавать свои собственные папки на рабочем столе для хранения объектов (программ и документов), которые вы используете

наиболее часто. Важно понимать, что стандартная папка — это в действительности графическое представление каталога. Если документ копируется в папку, то на самом деле он копируется в каталог, представленный этой папкой. Если вы создаете новую папку в папке, то вы тем самым создаете подкаталог в каталоге родительской папки. Некоторые папки, такие как Панель управления и Принтеры в действительности папками не являются. Они являются прикладными программами и специальными управляющими программами, которые Windows98/2000 представляет как папки для

того, чтобы обеспечить единообразный интерфейс со всеми ресурсами ПК.

Рабочий стол сам является в действительности скрытым каталогом под названием «\WINDOWS\Рабочий стол». Если вы копируете программный файл или файл документа в каталог «\WINDOWS\Рабочий стол», то его значок появляется в качестве объекта на рабочем столе. Аналогично, если вы создаете подкаталог в каталоге «\WINDOWS\Рабочий стол», то папка этого подкаталога появляется на рабочем столе. Это важно понимать, когда вы начинаете копировать или перемещать объекты на рабочем столе.

Панель задач

Панель задач обеспечивает другой способ открытия программ и документов, а также доступ к объектам, которые уже были открыты. Меню Пуск на панели задач содержит пункты, которые позволяют получить доступ к программам, документам и другим объектам. Оно содержит также команды для запуска программ из командной строки, установки системных параметров и завершения работы Windows 98/2000.

В нижеследующем списке приведены стандартные пункты меню Пуск:

Программы. Это каскадное меню дает быстрый доступ ко всем стандартным (в том числе игровым) программам, которые поставляются в составе Windows 98/2000. Оно включает также пункты для открытия Проводника и запуска сеанса MS-DOS. Документы. Это меню содержит 15 документов и папок, с которыми вы работали последними, включая те, с которыми вы работали в текущем сеансе.

Настройка. Это каскадное меню дает возможность доступа к Панели управления (используется для конфигурирования системы), к папке Принтеры (которая позволяет управлять всеми имеющимися принтерами) и к листам свойств панели задач.

Поиск. Это меню позволяет найти файл или папку на компьютере или в сети, а также найти в сети определенный компьютер.

Справка. Этот пункт меню открывает справочную систему Windows 98/2000.

Выполнить. Эта команда обеспечивает быстрый запуск программ. Вы можете ввести команду непосредственно или просмотреть каталоги, чтобы найти нужную программу.

Остановка. Эта команда появляется в меню Пуск, если данный ПК поддерживает режим экономии питания. Она используется для перевода ПК в этот режим.

Завершение работы. Выберите этот пункт меню для завершения с работы ПК, перезагрузки Windows 98/2000 или выхода из сети (но с сохранением работы Windows 98/2000).

Помимо меню Пуск на панели задач располагаются кнопки всех папок и программ, открытых на данный момент. Для выбора объекта необходимо щелкнуть на его кнопке на панели задач. При этом объект перемещается на передний план, после чего с ним можно работать.

Создание новых папок

Вы можете создать папку на рабочем столе или внутри другой папки. Главное, что нужно помнить при создании папки, это то, что папка является ничем иным, как каталогом — когда вы создаете папку, вы на самом деле создаете каталог.

Для того, чтобы создать папку на рабочем столе, щелкните на нем правой кнопкой мыши. Выберите из появившегося меню пункт Создать и затем Папка. Windows 98/2000 создаст на экране новую папку и присвоит ей имя из восьми знаков «минус». Вначале название будет выделено. Наберите желаемое название и нажмите клавишу Enter. Для

изменения названия существующей папки щелкните на папке, чтобы выбрать ее, затем щелкните на имени папки и отредактируйте его.

После того, как вы создали новую папку, все готово для копирования и перемещения объектов на рабочий стол или в папку.

Копирование и перемещение объектов

Вы можете копировать или перемещать программы, документы или другие объекты на рабочий стол или в папки. В обоих случаях ваши действия очень похожи. Чтобы скопировать объект из папки на рабочий стол или в другую папку, нажмите клавишу **Ctrl** и, удерживая ее, перетащите объект из папки на экран или из одной папки в другую. Для того, чтобы переместить объект, его надо просто тащить, не нажимая при этом на клавиатуре никаких клавиш. Таким образом, можно переместить объект только в пределах диска. При перетаскивании объекта на другой диск он будет скопирован. Чтобы переместить объект на другой диск, при перетаскивании удерживайте нажатой клавишу **Shift**.

При копировании и перемещении объектов помните, что файлы, входящие в копируемый объект, также копируются или перемещаются в новое место.

Если вы предпочитаете не пользоваться клавиатурой в сочетании с мышью или испытываете трудности в запоминании того, какую клавишу следует нажимать в какой ситуации, попробуйте тащить объект правой кнопкой вместо левой. Когда вы тащите объект правой кнопкой и освобождаете его, Windows выводит на экран контекстное меню. При этом Windows предлагает вам переместить или скопировать объект, создать его ярлык или прервать операцию.

Поиск стандартными средствами Windows

Стандартный поиск в Windows предполагает поиск по некоторой части имени файла и/или части текста файла. Например, в поле “Папка” указано “C:”, в поле “Имя” указано “a”, в поле “Искать текст” указано “ля-ля”. Будут найдены все файлы и все папки на диске C:, в имени (или расширении) которых присутствует буква a, а в самом файле где-либо встречается последовательность “ля-ля”. Для более конкретного поиска существуют так называемые маски.

Маска

Для поиска файлов с заданными условиями на имя файла используется маска. Маска - это схематичная запись условий на имя файла. Для этого используются следующие символы: “*” и “?”. “*” означает произвольную (даже пустую) последовательность символов. “?” означает ровно один символ. Кроме того, в маске можно использовать все те же символы, что и в именах файлов (включая “.”), они будут означать сами себя.

Например, если мы хотим найти все файлы, начинающиеся на букву “a”, надо использовать маску “a*”. (Расшифровка: сначала буква “a”, потом произвольная последовательность символов (м.б. пустая)).

Если мы хотим найти все файлы, в имени которых есть буква “a”, надо использовать маску “*a*”. (Расшифровка: сначала произвольная последовательность символов, затем буква “a”, затем снова произвольная последовательность символов).

Если мы хотим найти файлы, вторая буква имени которых – “a”, надо использовать маску “?a*”. (Расшифровка: сначала одна произвольная буква, затем буква “a”, потом произвольная последовательность символов).

Если мы хотим найти файл с заданным расширением “abc”, мы должны использовать маску “*.abc”

Ярлыки

В Windows существуют специальные файлы-ярлыки (линки, *.lnk). Программу (или др. файл) не всегда можно поместить в удобном для доступа месте. Поэтому в этом самом удобном месте (например, на рабочем столе или в главном меню) помещают ярлык для этой программы (файла).

2. Порядок выполнения работы

Задание 1

1. На рабочем столе создать папку с именем «Первая».
2. Открыть папку «Первая» и в ней создать папки «Вторая», «Третья» и «Четвертая»

3. С помощью команды Вид проверить включение следующих панелей (панель считается включенной, если она помечена галочкой):

- Обычные кнопки;
- Адресная строка;
- Подписи к кнопкам.

4. С помощью команды Вид проверить содержание папки «Первая», последовательно устанавливая режимы: крупные значки, Мелкие значки, Таблица, Список.

5. В папке «Третья» создать текстовый документ с именем a.txt. следующего содержания:

«Организаторы стремятся нанимать победителей. Преподносите себя как подарок судьбы их фирме. Упаси Вас Бог выглядеть жалкой попрошайкой».

6. Выполнить следующие действия;

- скопировать второе предложение в конец текста;
- поменять местами первое и последнее предложения текста;
- вставить в конец текста текущую дату.

7. Открыть окно приложения Калькулятор и выполнить следующие действия:

$$(344: 17 + 12.5) + 14 + 222 =$$

8. Полученный результат, используя Буфер обмена, вставить в конец документа a.txt.

9. Свернуть окно «Калькулятора» до ярлыка на Панели задач, а затем закрыть окно из Панели задач с помощью контекстно-зависимого меню.

10. Закрыть окно текстового документа a.txt, сохранив его содержимое.

11. Удалить папку «Первая» с Рабочего стола вместе с ее содержимым в Корзину.

12. Открыть Корзину и убедиться, что удаленная папка «Первая» переместилась в Корзину.

13. Удалить папку «Первая» из Корзины.

Задание 2

1. На диске С: создать папку “Find”.
2. В ней создать подпапки “a”, “b”, “c”, “d”.
3. Найти на диске С: все файлы, в имени которых встречается буква “а” (латинская).
4. Скопировать все эти файлы в папку C:\Find\a\.
5. Найти на диске С: все файлы, у которых “b” □ последняя буква имени (без учета расширения).
6. Скопировать все эти файлы в папку C:\Find\b\ (можно с клавиатуры: выделить несколько объектов – Ctrl; копировать объекты в буфер обмена – Ctrl+C).
7. В папке C:\Windows\ найти все файлы, название которых состоит из трех букв, а расширение содержит букву “с” (латинская). Вложенные папки не рассматривать.
8. Скопировать все эти файлы в папку C:\Find\c\.
9. Перенесите содержимое папки “a” в “d”(можно с клавиатуры: поместить объекты в буфер обмена – Ctrl+X; вставить объекты из буфера обмена - Ctrl+V).
10. В папке C:\Find\a\ создайте документ Microsoft Word и лист Microsoft Excel. В каждом файле впишите свою фамилию и дату.
11. При помощи стандартного средства поиска Windows посчитать, сколько файлов

на диске С:.

12. Создать в папке C:\Find\ ярлыки для программ mspaint.exe, explorer.exe, которые находятся на одном из локальных дисков.
13. Измените значок в свойствах ярлыка программы Проводник, подобрав подходящий из стандартных.
14. Сверните, разверните окно папки Find.
15. Восстановите нормальный размер окна папки Find и измените размеры его до максимума (на весь рабочий стол) и до минимума (чтобы были видны только созданные ярлыки).
16. Переместите окно папки Find сначала в верхний левый, затем в правый нижний угол рабочего стола.
17. Откройте в меню Пуск группу стандартных программ и отправьте на рабочий стол ярлык программы Калькулятор (используя контекстное меню).
18. Переместите ярлык программы Калькулятор с рабочего стола в папку Find.
19. В свойствах ярлыка Калькулятор установите для его быстрого вызова комбинацию клавиш, нажав ее (например: Ctrl+Alt+C).
20. Откройте программу Калькулятор установленной комбинацией клавиш.
21. Произведите любые арифметические и тригонометрические вычисления и закройте Калькулятор.
22. Загрузите графический редактор Paint через созданный ярлык.
23. Загрузите текстовый редактор WordPad через меню Пуск.
24. Переключитесь в окно Paint.
25. Вызовите диспетчер задач (Alt+Ctrl+Delete) и снимите задачу WordPad без сохранения данных (завершение выполнения "зависших" программ).
26. Закройте окно Paint, используя системное меню окна.
27. Сделайте копию ярлыков программы Проводник и Paint в папке Find.
28. Удалите копию ярлыка программы Проводник из папки Find перетаскиванием в Корзину.
29. Удалите копию ярлыка программы Paint из папки Find, минуя Корзину (Delete+Shift). В чем отличие этого способа удаления файлов от предыдущего?
30. Откройте свойства рабочего стола (экрана). Используя контекстную подсказку, изучите параметры настройки экрана.
31. Выберите любой фоновый рисунок из списка и нажмите кнопку Применить.
32. Измените оформление, выбрав любую цветовую схему из списка и нажмите OK.
33. Верните прежнюю цветовую схему, выбрав Стандартная из списка и нажмите OK.
34. Откройте свойства панели задач и изучите параметры (используя контекстную подсказку).
35. Выключите отображение часов в панели и включите "автоматически убирать панель с экрана". Нажмите кнопку OK.
36. Откройте свойства панели задач, верните старые параметры, установите классическое меню "Пуск" и нажмите OK.
37. Подготовьте компьютер к выключению и после появления надписи на экране, разрешающей выключение, выключите питание.

Проводник

1. Откройте программу Проводник, используя контекстное меню кнопки Пуск.
2. Создайте на диске С: папку с номером вашей группы.
3. Откройте ее и в ней создайте папку RAB.
4. Выполните поиск файла eula.txt на диске С:.
5. Скопируйте найденный файл в папку RAB, используя кнопки на панели

инструментов.

6. Отключите в свойствах вида папки опцию скрытия расширений для зарегистрированных файлов (Сервис-Свойства папки).
7. Переименуйте файл eula.txt в новый.doc.
8. Откройте папку WINDOWS и измените поочередно вид на: список, эскизы, таблица.
9. Упорядочьте содержимое папки по времени изменения, а затем по размеру, отслеживая изменения в таблице ярлыков. Найдите самый старый и самый большой файл.
10. Выделите и скопируйте с помощью меню Правка 5 произвольных файлов, суммарный объем которых не более 35 Кбайт из папки C:\WINDOWS в папку RAB.
11. Используя комбинации клавиш, переместите (вырезать и вставить) файл новый.doc в корневую папку диска C:.
12. Упорядочьте ярлыки по типу.
13. Скопируйте перетаскиванием все текстовые документы (файлы с расширением txt) из папки C:\WINDOWS в папку RAB.
14. Установите в свойствах файла новый.doc параметр «скрытый».
15. Выключите режим отображения «скрытых» файлов (спрятать имеющие статус «скрытый» файлы).
16. Создайте текстовый документ в директории RAB.
17. Откройте созданный файл, наберите в нем несколько строк и сохраните.
18. Удалите созданный файл перетаскиванием мышью.
19. Включите режим отображения «скрытых» файлов.
20. Удалите тренировочные файлы, используя клавишу на клавиатуре.
21. Удалите папку RAB перетаскиванием в Корзину панели папок.
22. Выдите из Проводника.

Стандартные приложения Windows

1. Запустите графический редактор Paint.
2. Сверните все окна.
3. Скопируйте в буфер обмена изображение рабочего стола Windows (клавиша Print/PrtScr).
4. Разверните Paint и вставьте содержимое буфера обмена в рабочую область программы.
5. Используя инструменты: линия и надпись, сделайте подписи с прозрачным фоном к основным элементам рабочего стола.
6. Сохраните рисунок в папку с номером вашей группы, введя имя "Рабочий стол" и установив тип файла JPG.
7. Очистите рабочую область.
8. Отмените очистку (последнюю операцию).
9. Создайте новый файл без сохранения последних изменений в файле Рабочий стол.jpg.
10. Растворите рабочую область на все окно.
11. Сохраните заготовку рисунка в папку Практика по информатике, введя имя файла Пейзаж и установив тип JPG.
12. Используя все инструменты рисования, создайте рисунок, например: домик на фоне голубого неба, белых облаков, желтого солнца, зеленой травки с цветочками и с серой дорожкой к порогу (можно нарисовать автомобиль, корабль, море или ночное небо). Периодически в процессе рисования сохраняйте внесенные изменения в рисунок.
13. Закройте графический редактор Paint с сохранением изменений в файле.
14. Сделайте рисунок Пейзаж.jpg фоном рабочего стола.
15. В соответствие с вариантом задания средствами графического редактора Paint подготовить рисунок. Высота рисунка - 6 см (Рисунок-Атрибуты).

16. Средствами текстового процессора WordPad подготовить отчет о работе, содержащий рисунок из Paint. При описании порядка создания рисунка следует выделить обычным курсивом названия инструментов, а полужирным курсивом – выбранные пункты меню.

17. Для защиты работы продемонстрировать отчет на компьютере в среде WordPad.

3. Контрольные вопросы

1. Что такое программное обеспечение компьютера?
2. Чем отличается системное программное обеспечение от прикладного? Приведите примеры системного и прикладного программного обеспечения.
3. Что такое операционная система? Какие операционные системы Вам известны?
4. Опишите основные элементы ОС Windows (рабочий стол, окно, значок, ярлык, кнопка, панель задач, меню, папка, меню «Пуск»).
5. Какие разновидности окон существуют в ОС Windows?
6. Опишите основные приемы работы с окнами в ОС Windows.
7. Что значит понятие «контекстное меню»? Как можно вызвать данное меню?
8. Опишите основные возможности программы «Проводник».
9. Как можно создать новую папку в ОС Windows?
10. Как можно переименовывать, копировать и удалять объекты в ОС Windows?

2.2 Лабораторная работа № 2 (4 часов)

Тема: «Прикладное программное обеспечение»

Лабораторная работа № 2 Создание комплексных документов в MS Word

Цель работы: изучение основных возможностей текстового процессора Microsoft WORD и получение практических навыков по созданию и редактированию текстовых документов.

2.7.2 Задачи работы:

1. Освоить основные приемы работы с документами и редактирование, форматирования текста в MS Word
2. Освоить основные приемы создания, редактирования и форматирования графических объектов в документах текстового процессора MS Word
3. Освоить основные приемы оформления структурированного документа в MS Word
4. Освоить основные приемы создания, редактирования и форматирования таблиц в документах текстового процессора MS Word
5. Освоить основные приемы создания и форматирования математических формул в текстовых документах MS Word
6. Для закрепления и проверки полученных навыков, необходимо ответить на контрольные вопросы и выполнить дополнительные задания. Результаты работы продемонстрировать преподавателю в электронном виде.

2.7.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер

2.7.4 Описание (ход) работы:

Для того, чтобы создавать, а также редактировать и форматировать (исправлять, изменять) тексты, существуют специальные программы, называемые текстовыми редакторами или текстовыми процессорами. Текстовые процессоры сложнее и обладают большими

функциональными возможностями для оформления текстов. В последнее время все большую популярность среди широкого круга пользователей завоевывает текстовый процессор Word для Windows.

Word – многофункциональная программа обработки текстов, настольная издательская система. Ее предназначение:

- 1) набор, редактирование, верстка текста и таблиц;
- 2) управление всеми пунктами меню, опциями и командами с помощью мыши;
- 3) просмотр на дисплее готового к печати документа без затраты бумаги на дополнительные распечатки;
- 4) вставка рисунков и слайдов;
- 5) заготовка бланков, писем и других документов;
- 6) обмен информацией с другими программами;
- 7) проверка орфографии и поиск синонимов.

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 приступать к вводу текста. Создание нового документа в других случаях (уже в самом текстовом редакторе) осуществляется командами **Создать** в меню **«Office»**.

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 Microsoft Office Word**. Окно содержит следующие элементы (Рис.1):

Строка названия документа (1) – слева в строке заголовка находится значок Word (кнопка **«Office»**), рядом имя загруженного файла документа, с которым вы работаете в текущий момент (в данном случае этот файл называется **Документ Microsoft Office Word**), затем – имя программы - **Microsoft Word** и справа 3 кнопки управления окном (свернуть, развернуть (свернуть в окно), закрыть документ).

Кнопка «Office» (2) – находится в верхнем левом углу документа. Некоторые пункты меню **Office** раскрываются и имеют соответствующее подменю, если навести на них указатель мыши.

Вкладка или панель инструментов (4) – находится ниже строки названия документа. Внутри каждой группы инструментов (5) 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.

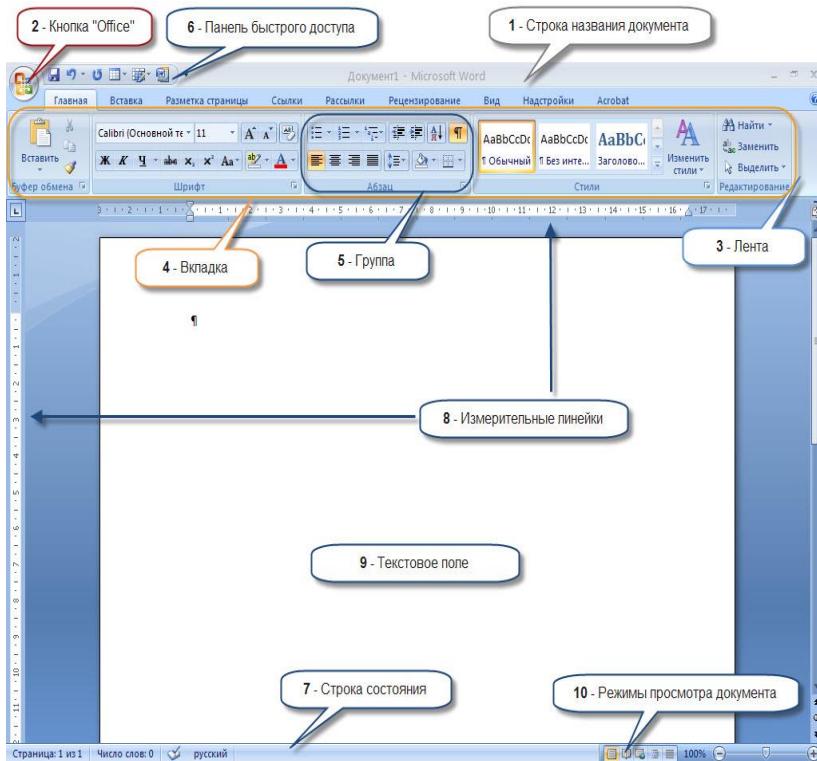


Рис.1 Окно программы Word

Ниже находится Рабочая область или Текстовое поле (9) – это область экрана, где отображается создаваемый документ и производится его редактирование. Сверху и слева документа находятся горизонтальная и вертикальная измерительные линейки (8).

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 или лента (3) 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 **Вид, 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(7). В ней отображается различная информация о самом документе и о текущем состоянии текстового процессора. Первые три числа означают номер страницы в документе, общее количество страниц в документе, число слов в документе.

Завершение работы с MS WORD

Завершение работы Word можно осуществлять с помощью кнопки **Закрыть** .

Если измененный документ не записан в файл, Word потребует записать его, либо подтвердить выход без записи.

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. Также задать подходящий режим можно при помощи кнопок Режим просмотра документа (10) в рабочем окне Word.

2. Используя команды **Вид**, **Масштаб** можно изменять масштаб видимости документа.

3. Начинать работу с текстовым документом необходимо с настройки параметров страницы. Это осуществляется с помощью команды меню **Разметка страницы** на вкладке (4). В открывшейся ленте(4) **Разметка страницы** в группе(5) **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 **Вставка**, **Верхний колонтитул** или **Нижний колонтитул** (все зависит от того в какое место документа – в верх или вниз – нужно вставить номер).

5. Для автоматического переноса слов в документе необходимо выполнить команды **Разметка страницы**, **Расстановка переносов** и в появившемся диалоговом окне включить переключатель **Авто** (переносы будут расставляться автоматически)

6. Для проверки правописания, необходимо щелкнуть кнопку **Рецензирование** на панели инструментов и выбрать команду **Правописание**.

Основные приемы редактирования документа

Под редактированием понимается изменение уже существующего документа. Начинают редактирование с загрузки (открытия) документа. Для этого используется команда **Открыть** в меню Office (2), 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 **Копировать**, находящийся в меню Главная в группе Буфер обмена
- установить курсор в то место текста, куда нужно вставить фрагмент;
- выполнить команду находящийся в меню Главная в группе Буфер обмена.

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.
Строчу 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 I , 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 **Вырезать**, находящийся в меню Главная в группе Буфер обмена

4. **Отмена** последнего действия осуществляется с помощью кнопок  на панели быстрого доступа (6). Вернуть отмененное действие можно кнопкой **Вернуть**  на той же панели (6).

Форматирование документа в MS Word

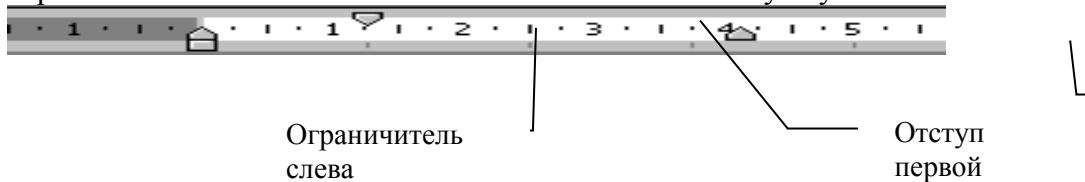
Целью обработки любого текста является приданье ему некоторого вида облегчающего восприятие информации и подчиненного общепринятым правилам и стандартам. **Форматирование** – приданье документу определенной формы. Форматирование текста осуществляется средствами меню **Главная**. Процедура форматирования 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 **Главная**, группа **Абзац** кнопки .

Установка межстрочного интервала. Это расстояние между строками в абзаце. Интервалы устанавливаются с помощью команд **Главная**, группа **Абзац**, в открывшемся диалоговом окне **Отступы и интервалы** выбрать нужные параметры.

Установка абзацных отступов. Это отступы абзаца от левого и правого ограничителя и отступ первой строки абзаца (красная строка). Отступы устанавливаются с помощью команд **Главная**, группа **Абзац**, в открывшемся диалоговом окне **Отступы и интервалы** пункт **Отступ** или с помощью *Линейки*, на которой необходимо перетаскивать ограничители в нужную позицию.



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 воспользоваться кнопками **Обычный + П** Times New Roman 12 **Ж К Ч** на панели инструментов **Главная** в группе **Шрифт** 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** в меню Office. Команда **Сохранить** или кнопка на панели быстрого доступа сохраняет документ с текущим именем, а команда **Сохранить как** позволяет сохранить документ под новым именем. Для этого в диалоговом 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: **Предварительный просмотр**, кнопка на панели быстрого доступа. Выход из режима просмотра документа осуществляется щелчком по кнопке «Закрыть».

Печать документа производится командами: **Печать** в меню Office. 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 «OK».

1. Установка параметров окна пользователя

1. Загрузим программу MS Word:
 - выполним команды **Пуск, Программы, Microsoft Office, MS Word**.
2. Выберем команду **Вид**, щелкнув по ней левой кнопкой мыши.
3. Установим следующие режимы просмотра документов:

Выбор соответствующего режима осуществляется щелчком по нему левой кнопки мыши:

- режим разметки страницы;

- режим чтения;
- режим электронного документа;
- режим структуры.

4. Вернемся к режиму **Разметка страницы**.

5. Выведем на экран линейки, выбрав команды: **Вид, Линейка**.

Пункт меню Линейка должен быть помечен символом ✓.

7. Установим масштаб – 150%, выполнив команду: **Вид, Масштаб**, а затем вернемся к исходному масштабу.

8. Завершим работу с редактором MS Word:

- выполним команды **Закрыть** в меню Office.

2. Сохранение, загрузка и печать документов

1. Загрузим редактор **MS Word**.

Если после запуска редактора на экран не будет выведено автоматически окно документа, то создадим новый документ, выполнив команды: Создать в меню Office и подтвердив тип устанавливаемого по умолчанию документа – новый документ.

В строке заголовка появилось предлагаемое по умолчанию имя – Документ 1.

2. Наберем текст в установленных по умолчанию границах и заданным по умолчанию шрифтом текста для набора.

Текст документа:

Находите время!

Находите время для размышлений – это источник силы!

Находите время для работы – это условие успеха!

Находите время для мечты – это путь к звездам!

Находите время для любви – это источник радости жизни!

Находите время для веселья – это музыка души!

Находите время для дружбы – это условие счастья!

Находите время для игры – это секрет молодости!

3. Сохраним документ под именем «**ww.doc**»:

- выполним команды: **Сохранить как** в меню Office или нажмем кнопку  на панели быстрого доступа;
- в окне диалога установим следующие параметры:
 - в поле **Папка** выберем папку «Мои документы»;
 - в поле **Имя файла** введем имя файла **ww.doc**;
 - в поле **Тип файла** выберем тип – **документ Word**;
 - щелкнем кнопку «Сохранить».

4. Просмотрим содержимое документа перед печатью:

- Нажмем кнопку  на панели быстрого доступа или же в меню Office, группа **Печать, Предварительный просмотр**
- для возврата в режим редактирования документа щелкнем левой кнопкой мыши кнопку «Закрыть».

5. Сохраним документ под другим именем «**ww1.doc**»:

- выполним команды: **Сохранить как** в меню Office;
- в поле **Имя файла** введем имя файла - **ww1.doc**.
- щелкнем кнопку «Сохранить».

6. Распечатаем документ:

- выполним команды: **Печать** в меню Office;
- в диалоговом окне **Печать** установим параметры печати;

- щелкнем кнопку «OK».

7. Закроем файл:

- выполним команды: **Закрыть** в меню Office.

8. Завершим работу в программе **MS Word**:

- выполним команды: **Выход** 

3. Редактирование документа

1. Загрузим редактор **MS Word**.
2. Откроем документ с именем «**ww.doc**»:
 - выполним команды: **Открыть** в меню Office ;
 - выберем папку «Мои документы» и файл **ww.doc**;
 - щелкнем кнопку «Открыть».

Открытый документ будет иметь вид:

Находите время!

Находите время для размышлений – это источник силы!

Находите время для работы – это условие успеха!

Находите время для мечты – это путь к звездам!

Находите время для любви – это источник радости жизни!

Находите время для веселья – это музыка души!

Находите время для дружбы – это условие счастья!

Находите время для игры – это секрет молодости!

3. Скопируем заголовок «Находите время!» в конец текста:
 - выделим заголовок текста;
 - выполним команды: **Копировать** в меню **Главная** в группе **Буфер обмена**;
 - переместим курсор в конец текста и нажмем клавишу **Enter**;
 - выполним команды: **Вставить** в меню **Главная** в группе **Буфер обмена**.

4. Поменяем местами слова «Находите» и «Время» в скопированном тексте:
 - выделим слово «Находите»;
 - выполним команды: **Вырезать** в меню **Главная** в группе **Буфер обмена**;
 - установим курсор после слова «Время!»;
 - выполним команды: **Вставить** в меню **Главная** в группе **Буфер обмена**.

5. Выполним замену слова «время» на слово «времечко»:
 - переместим курсор в начало текста;
 - выполним команды: **Заменить** в меню **Главная** в группе **Редактирование**;
 - в окне диалога установим следующие параметры:
 - в поле **поиск**: время
 - в поле **замена**: времечко;
 - щелкнем кнопку **Заменить все**.

На экран будет выдано сообщение: «Произведено 9 замен», щелкнем кнопку «OK».

После выполнения операций документ примет вид:

Находите времечко!

Находите времечко для размышлений – это источник силы!

Находите времечко для работы – это условие успеха!

Находите времечко для мечты – это путь к звездам!

Находите времечко для любви – это источник радости жизни!

Находите времечко для веселья – это музыка души!

Находите времечко для дружбы – это условие счастья!

Находите времечко для игры – это секрет молодости!

времечко! Находите

4. Форматирование документа

Предполагается, что в окне редактора выведен текст документа ww.doc.

1. Выделим заголовок и выполним команды: **Главная**, в группе **Шрифт**:
 - в окне диалога установим следующие параметры:
 - шрифт – Comic Sans MS;
 - начертание – полужирный;
 - размер – 18;
 - цвет – коричневый;
 - щелкнем кнопку «OK».
2. Выделим первое предложение текста и выполним операции:
 - выполним команды: щелкнув правой кнопкой мыши выбрать пункт **Абзац**;
 - в окне диалога установим следующие параметры:
 - выравнивание – по центру;
 - отступ слева – 2;
 - отступ справа – 4;
 - первая строка – отступ, на 2 см;
 - межстрочный интервал – двойной;
 - щелкнем кнопку «OK».
3. Выделим второе предложение текста и выполним операции:
 - выполним команды: щелкнув правой кнопкой мыши выбрать пункт **Шрифт**;
 - установим в окне диалога следующие параметры:
 - шрифт – Arial;
 - начертание на ваш выбор (желательно использовать все)
 - размер – 16;
 - цвет – синий;
4. Выделим третье предложение текста и выполним операции:
 - выполним команды: **Главная**, **Стили** (вызов контекстного меню осуществляется путём нажатия левой кнопки мыши на значек, находящийся в нижнем правом углу **Группы**);
 - в окне диалога установим следующие параметры: выделенная цитата
5. Выделим четвертое предложение текста и выполним операции:
 - выполним команды: **Главная**, **Стили** (вызов контекстного меню осуществляется путём нажатия левой кнопки мыши на значек, находящийся в нижнем правом углу **Группы**, изменение стиля происходит путем нажатия левой кнопкой мыши на стрелку находящуюся в строке названия стиля справа);
 - выберите стиль на свое усмотрение
6. Выделим пятое предложение текста и выполним операции:
 - выполним команды: **Главная**, **Стили** (вызов контекстного меню осуществляется путём нажатия левой кнопки мыши на значек, находящийся в нижнем правом углу **Группы**) **Заголовок 3**;
7. Выделим шестое предложение текста и выполним операции:
 - выполним команды: **Главная**, **Буквица**, **Положение – в тексте**;
8. Выделим седьмое предложение текста и выполним операции:
 - выполним команды: **Главная**, **Стили** (вызов контекстного меню осуществляется путём нажатия левой кнопки мыши на значек, находящийся в нижнем правом углу **Группы**, изменение стиля происходит путем нажатия левой кнопкой мыши на стрелку находящуюся в строке названия стиля справа);
 - в диалогом окне стиль шапка;

После выполнения операций документ примет вид:

Найдите времечко!

Найдите времечко для размышлений – это источник силы!

Найдите времечко для работы – это условие успеха!

Найдите время для мечты – это путь к звездам!

Найдите времечко для любви – это источник радости жизни!

Найдите времечко для веселья – это музыка души!

Н

аходите времечко для дружбы – это условие счастья!

Найдите времечко для игры – это секрет молодости!

времечко! Найдите

9. Создайте выражение **Маша Таня Маня Ваня Ляля**, содержащее верхний и нижний индексы, для этого:

- напишите выражение полностью одним шрифтом;
- выделяйте поочередно части выражения, которые должны быть индексами (сначала Таня, затем Ваня);
- выберите команду **Главная**, в группе **Шрифт**, кнопка **Надстрочный знак** или **Подстрочный знак**;

10. Сохраните документ под именем «**ww2.doc**».

11. Завершите работу в программе MS Word.

5. Вставка рисунков в текст документа

1. Загрузим редактор **MS Word**.

2. Откроем документ с именем «**ww1.doc**»:

– выполним команды: **Открыть** в меню Office;

– выберем папку «Мои документы» и файл «**ww1.doc**»;

– Щелкнем кнопку «**Открыть**»

Открытый документ будет иметь вид:

Найдите время!

Найдите время для размышлений – это источник силы!

Найдите время для работы – это условие успеха!

Найдите время для мечты – это путь к звездам!

Найдите время для любви – это источник радости жизни!

Найдите время для веселья – это музыка души!

Найдите время для дружбы – это условие счастья!

Найдите время для игры – это секрет молодости!

3. Вставим рисунок:

– переместим курсор в начало первой строки;

– выполним команды: **Вставка, Рисунок**;

– выберем картинку;

4. Уменьшим рисунок в размерах:

– щелкнем левой кнопки мыши по рисуноку для его выделения;

Вокруг рисунка появится маркерная рамка.

– установим указатель мыши на одном из маркеров;

Указатель примет вид двунаправленной стрелки ↔.

– нажмем левую кнопку мыши и, не отпуская ее, будем перетаскивать маркер до тех пор, пока рисунок не станет нужных размеров;

- отпустим кнопку мыши.
- 5. Аналогично вставим еще два рисунка.

После выполнения операций документ примет вид:



Находите время!

Находите время для размышлений – это источник силы!

Находите время для работы – это условие успеха!



Находите время для мечты – это путь к звездам!

Находите время для любви – это источник радости жизни!



Находите время для веселья – это музыка души!

Находите время для дружбы – это условие счастья!

Находите время для игры – это секрет молодости!

6. Работа со списками

1. Выполним команды: **Создать** в меню Office

2. Создадим нумерованный список:

- введем в заголовок списка текст «Список студентов», нажмем клавишу Enter;
- выберем команды: **Главная, Нумерация** в группе Абзац;
- выберем стиль списка, щелкнув соответствующий образец (1., 2., 3.);
- введем элементы списка, нажимая клавишу Enter в конце абзаца:
- Иванов И.И.;
- Петров П.П.;
- Сидоров С.С.;

На экране автоматически появляются номера списка.

После выполнения операций документ примет вид:

Список студентов

1. Иванов И.И.
2. Петров П.П.
3. Сидоров С.С.

3. Преобразуем нумерованный список в маркированный:

- выделим нумерованный список (без заголовка);
- выполним команды: **Главная, Маркеры**, выберем маркер на свой выбор;

После выполнения операций документ примет вид:

Список студентов

- ◆ Иванов И.И.

- ◆ Петров П.П.
- ◆ Сидоров С.С.

4. Создадим многоуровневый список:

- введем текст заголовка «Структура Университета»;
- выполним команды: **Главная, Многоуровневый список**;
- выберем образец многоуровневого списка (1., 1.1., 1.1.1.);
- наберем текст пункта 1 «Факультет» и нажмем клавишу Enter;
Автоматически продолжится нумерация списка с пункта 2.
- нажмем правую кнопку мыши для вызова контекстно-зависимого меню;
- выполним команду **Увеличить отступ**;

Нумерация списка изменится с «2.» на «1.1.».

- наберем следующий пункт и нажмем клавишу Enter;
- поступая аналогично и используя пункты контекстно-зависимого меню **Увеличить отступ** и **Уменьшить отступ**, получим многоуровневый список.

После выполнения операций документ примет вид:

Структура Университета

1. Факультет А
 - 1.1. Специальность 1
 - 1.2. Специальность 2
2. Факультет Б
 - 2.1. Специальность 1
 - 2.2. Специальность 2

5. Сохраним документ под именем «Списки» в папке «Мои документы»:

- выполним команды: **Сохранить как** в меню Office.

7. Создание оглавления

Для того чтобы сформировать оглавление документа, необходимо все заголовки, которые должны входить в оглавление, отформатировать стилем **Заголовок** (Заголовок 1, Заголовок 2 и т.д.). Для этого:

1. Перед набором заголовков выполним команды: **Главная, Стиль, Заголовок 1**.
– Наберем основной текст документа шрифтом Times New Roman размером 12.

Текст документа:

Microsoft Word

Microsoft Word – это текстовый редактор общего назначения. При работе с редактором пользователю предоставляются следующие возможности:

- набор и редактирование текста;
- форматирование текста;
- создание и редактирование таблиц;
- создание списков;
- работа в многооконном режиме.

Microsoft Excel

Microsoft Excel – это электронная таблица, которая используется для обработки табличных данных, построения диаграмм, проведения финансовых статистических расчетов, проведения математического моделирования.

Microsoft Access

Microsoft Access – это база данных. Она состоит из отдельных компонентов, которые используются для хранения и представления информации. Этими компонентами являются таблицы, формы, отчеты, запросы, макросы и модели.

2. Для автоматического формирования заголовка установим курсор в том месте документа, где будет находиться оглавление (в нашем примере перед набранным текстом):
 - выполним команды: **Ссылки, Оглавления**, вкладка **Оглавление**.
3. В окне диалога установим следующие параметры:
 - уровни – Уровень 1;
 - параметры, доступные стили, Заголовок 1, Уровень 1;
 - щелкнем кнопку «OK».

После выполнения операций документ примет вид:

Microsoft Word	1
Microsoft Excel	65
Microsoft Access	65

Microsoft Word

Microsoft Word – это текстовый редактор общего назначения. При работе с редактором пользователю предоставляются следующие возможности:

- набор и редактирование текста;
- форматирование текста;
- создание и редактирование таблиц;
- создание списков;
- работа в многооконном режиме.

Microsoft Excel

Microsoft Excel – это электронная таблица, которая используется для обработки табличных данных, построения диаграмм, проведения финансовых статистических расчетов, проведения математического моделирования.

Microsoft Access

Microsoft Access – это база данных. Она состоит из отдельных компонентов, которые используются для хранения и представления информации. Этими компонентами являются таблицы, формы, отчеты, запросы, макросы и модели.

8. Работа с таблицами

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

- Выбрать в меню **Вставка** опцию **Таблица**. В открывшемся диалоговом окне установить размерность таблицы (число столбцов и строк).
- Можно создавать таблицы, рисуя их «карандашом» с помощью мыши при выбранном меню  **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 понадобиться, 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

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

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

Выделяемый элемент	Способ выделения
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 таблица	Нажать левую кнопку мыши в начале таблицы и протащить 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:

- кнопки панели инструментов **Таблицы и границы** которая появляется при выделении таблицы упрощает работу с таблицами и позволяет реализовывать некоторые дополнительные возможности их форматирования.

ЗАДАНИЕ: Создать и отредактировать табличное представление информации в документе.

1. Загрузим редактор Microsoft Word.
2. Создадим таблицу. Для этого:
 - выполним команды: **Вставка, Таблица, Вставить таблицу**;
 - в окне диалога установить следующие параметры:
 - ◆ число столбцов – 4;
 - ◆ число строк – 4;
 - ◆ автоподбор ширины столбцов – авто;
 - ◆ щелкнем кнопку «OK».
 - введем в ячейки таблицы исходные данные.

Таблица примет вид:

1	2	3	5
6	7	8	10
11	12	13	15
16	17	18	20

3. Вставим в таблицу перед последним четвертым столбцом дополнительный столбец. Для этого:
 - выделим последний столбец;
 - выполним команды: **Макет, Вставить слева** в группе Строки и столбцы;
 - введем исходные данные в дополнительный столбец.

Таблица примет вид:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

4. Добавим в таблицу дополнительную строку. Для этого:
 - выделим последнюю строку таблицы;

- выполним команды: **Макет, Вставить снизу** в группе Строки и столбцы;
- введем исходные данные в строку.

Таблица примет вид:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

5. Изменим ширину первого столбца. Для этого:

- выделим первый столбец;
- выполним команды: **Макет, в группе Размер ячейки, в строке Ширина столбца таблицы ввести – 2 см;**

Таблица примет вид:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

6. Объединим ячейки с исходными данными – 1, 2, 3, 4, 5. Для этого:

- выделим ячейки 1, 2, 3, 4, 5;
- выполним команды: **Макет, Объединить ячейки** в группе Объединить.

Таблица примет вид:

1				
2				
3				
4				
5				
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

7. В объединенную ячейку введите новый текст «Объединенные ячейки». Для этого:

- выделим объединенную ячейку;
- введем текст «Объединенные ячейки»;
- выполним выравнивание текста по центру .

8. Аналогично объедините ячейки 16 и 21.

9. Удалите информацию из объединенных ячеек 16 и 21.

Таблица примет вид:

Объединенные ячейки				
6	7	8	9	10
11	12	13	14	15
	17	18	19	20
	22	23	24	25

10. Разобьем объединенную ячейку на 2 части по диагонали. Для этого:

- переместим курсор в объединенную ячейку 16, 21;

- выполним команды: **Конструктор, Нарисовать таблицу**;
- подвести указатель мыши к указанной клетке, при этом он примет форму карандаша  .
- нарисуем в ячейке диагональ.

Таблица примет вид:

Объединенные ячейки				
6	7	8	9	10
11	12	13	14	15
	17	18	19	20
	22	23	24	25

11. Разобьем ячейку 24 на 3 части. Для этого:

- выделим ячейку 24
- выполним команды: **Макет, Разбить ячейки**;
- в окне диалога установим **Число столбцов –3, Число строк 1**.

Таблица примет вид:

Объединенные ячейки				
6	7	8	9	10
11	12	13	14	15
	17	18	19	20
	22	23	24	25

12. Объединим ячейки 13 и 18. Для этого:

- выделим ячейки 13, 18;
- выполним команды: **Макет, Объединить ячейки**.

13. Разобьем объединенные ячейки 13 и 18 на 2 части. Для этого:

- выделим объединенные ячейки 13 и 18;
- выполним команды: **Макет, Разбить ячейки**;
- в окне диалога установим: **Число столбцов –2, Число строк 1**.

Таблица примет вид:

Объединенные ячейки				
6	7	8	9	10
11	12	13	14	15
	17	18	19	20
	22	23	24	25

14. Отформатируем таблицу. Для этого:

- данные первого столбца выровняем по центру;
- данные пятого столбца выровняем по правому краю;
- выполним внешнее и внутреннее обрамление таблицы с помощью команд меню: **Конструктор, Границы, Границы и заливка** (таблица должна быть выделена);
- в диалоговом окне задать тип линий, цвет линий, ширину линий, задать границы.
- Нажать «OK».

15. Оформим внешний вид таблицы. Для этого:

- выделим таблицу;
- выполним команды: **Конструктор, Стили таблицы, Создать стиль таблицы**, выберем в строке Основан на стиле – Объемная таблица 3;

- Щелкнем по кнопке **Ок**.

Таблица примет вид:

Объединенные ячейки

6	7	8	9	10
11	12	13	18	15
	17		19	20
	22	23	24	25

9. Создание и обработка графических объектов

1. Загрузите текстовый процессор Word.
2. Введите следующий текст:

Создание и обработка графических объектов

3. Создайте овал в документе, для этого:
 - Выполните команды **Вставка, Фигуры, Овал**;
 - расположите курсор (в форме крестика **+**) в месте размещения одного из краев овала;
 - нажмите левую кнопку мыши и протащите указатель до получения овала требуемой формы и размера.
4. Самостоятельно создайте прямоугольник и треугольник пользуясь теми же командами.
5. Создайте прямоугольник и овал правильной формы (при рисовании нажата клавиша **Shift**).
6. Отредактируйте созданные графические объекты, для этого:
 - подведите курсор к любому из объектов, и после того, как курсор примет форму стреловидного креста, щелкните левой клавишей мыши (объект выделен);
 - правой кнопкой мыши вызовите контекстное меню **Формат автофигуры**, измените **Цвет линии, Цвет заливки, Тип линии, Вид линии**;
 - снимите выделение с объекта – выполнив щелчок вне объекта.
7. Переместите графические объекты, поменяв их взаимное расположение, для этого:
 - подведите курсор к границе объекта так, чтобы курсор принял форму стреловидного крестика и перетащите объект.
8. Создайте схему, соединив между собой два созданных графических объекта стрелкой (с помощью команд **Вставка, Фигуры, Стрелка**).
9. Сделайте надпись «нет» над стрелкой, для этого:
 - выполните команды **Вставка, Надпись, Нарисовать надпись**;
 - изобразите прямоугольник на месте будущего слова;
 - напечатайте слово;
 - вызовите контекстное меню правой клавишей мыши, выберите команду **Формат надписи** и во вкладке **Цвета и линии** сделайте установки: **Заливка, цвет - Нет заливки, Линии, цвет - Нет линий**; **ОК**.
10. Сгруппируйте созданную графическую схему, для этого:
 - поочередно выделите эти объекты при нажатой клавише **Shift**;
 - правой кнопкой мыши вызвать контекстное меню;
 - в контекстном меню выберите команду **Группировка, Группировать**.
11. Переместите сгруппированный объект на текст.
12. Измените размещение объекта относительно текста, для этого:

- Вызовите контекстное меню;
- выберите **Порядок, Поместить за текстом.**

13. Разгруппируйте объекты, для этого:

- выделите сгруппированные объекты;
- вызовите контекстное меню;
- выберите команду **Группировка, Разгруппировать.**

14. Измените размеры графических объектов, для этого:

- выделите объект (или группу объектов);
- установите курсор на один из маркеров выделяющей рамки (форма курсора - двухсторонняя стрелка ↔) и переместите мышку;
- выполните предыдущую операцию при нажатой клавише **Shift** или **Ctrl** и просмотрите результат.

15. Вставьте в документ и настройте автофигуру, для этого:

- Выполните команды **Вставка, Фигуры;**
- выберите любую автофигуру;
- расположите курсор (в форме крестика +) в месте размещения одного из краев фигуры;
- нажмите левую кнопку мыши и протащите указатель до получения фигуры требуемой формы и размера;
- измените цвет, заливку, объем и другие параметры фигуры.

16. Установите надпись «Информатика» в автофигуру, для этого:

- выделите фигуру;
- вызовите контекстное меню (щелчок правой кнопкой мыши по фигуре) и выберите команду **Добавить текст.** Внутри фигуры появится курсор и можно вписывать текст или после выделения фигуры нажмите кнопку **Вставка, Надпись Нарисовать надпись,** изобразите прямоугольник на месте будущего слова, напечатайте слово.
- вызовите контекстное меню правой клавишей мыши, выберите команду **Формат надписи** и во вкладке **Цвета и линии** сделайте установки: **Заливка, цвет - Нет заливки, Линии, цвет - Нет линий;** **OK**
- после ввода надписи щелкните левой клавишей мыши вне фигуры.

*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.** Выберите вкладку **Надпись** и уберите внутренние поля (просветы) в фигуре, возможно текст поместится.*

17. Измените направление надписи «Информатика»:

- выделите фигуру;
- в контекстном меню выполните команду **Формат Автофигуры, Надпись, Ok;**
- в открывшейся на вкладке группе Работа с надписями, выбрать вкладку Формат, Направление текста (нажимать на кнопку направление текста следует до тех пор пока вы не получите нужного направления);
- щелкните вне фигуры.

18. Создайте графический объект произвольной формы, для этого:

- в меню **Вставка, Фигуры** выберите **Полилиния;**
- нажав левую кнопку мыши, нарисуйте линию произвольной формы (закончить рисование – двойной щелчок).

19. Измените созданный объект:

- выделите его и выберите в контекстном меню пункт - **Начать изменение**

узлов;

- с помощью курсора перетащите точки перегиба в нужном направлении.

20. Измените параметры графических объектов, используя пункт **Формат автофигуры** в контекстном меню (самостоятельно).

21. Создайте блок-схему изображенную на рисунке 2:

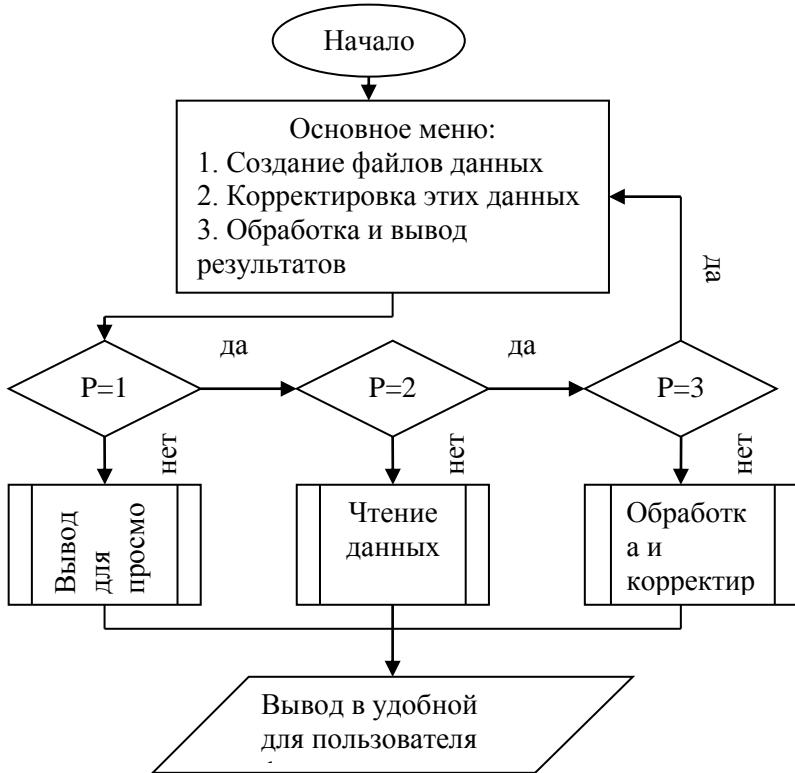


Рис. 2 Учебная блок-схема

22. Сохраните документ под именем **Автофигура**.

10. Запись математических формул

1. Загрузите текстовый процессор Word.

$$\sum_{I=1}^K Z$$

2. Создайте следующий фрагмент формулы $\sum_{I=1}^K Z$. Для этого:

- поместите курсор в точку вставки формулы и выполните **Вставка, Формула** (нажать на стрелку находящуюся в строке названия пункта), выбрать пункт **Вставить новую формулу**;
- в появившейся панели **Работа с формулами** выберите **Конструктор, Крупный оператор**;
- выберите **Шаблон суммы с верхним и нижним пределами** (второй шаблон в первом ряду);
- введите нужный символ, число или выражение в каждый из слотов, предварительно поместив туда курсор;
- Завершите запись формулы (щелчок за пределами рамки).

3. Создайте формулу остатка ряда Тейлора, для этого:

- поместите курсор в точку вставки $R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-a)^{(n+1)}$

формулы;

- выполните команды **Вставка, Формула** (нажать на стрелку находящуюся в строке названия пункта), выбрать пункт **Вставить новую формулу**;
- в появившейся панели **Работа с формулами** выберите **Конструктор, Индекс**, выберите нужный индекс
- в слоте ввода формулы в индексе введите соответствующие буквы;
- поместите курсор в конец введенной формулы и введите $=$;
- выберите во вкладке **Работа с формулами** выберите **Конструктор, шаблоны Дробей**, выберите соответствующую дробь;
- в слоте числителя вставьте шаблон нужного индекса и в каждом окошке наберите соответствующие символы;
- установите курсор справа от индекса и введите оставшуюся часть числителя $,$;
- в слот знаменателя введите $(n+1)!$;
- установите курсор в конец введенной формулы и введите оставшуюся часть формулы аналогичным способом используя соответствующий индекс;
- завершите запись формулы.

4. Создайте формулы с фигурными скобками по

образцу:

- Выполните команды **Вставка, Формула**, $y = \begin{cases} a + x, \text{при } x > 0 \\ a - x, \text{при } x \leq 0 \end{cases}$ **Вставить новую формулу**;
- в слоте ввода введите $y=$;
- выберите в группе **Конструктор** пункт **Скобки** и нажмите соответствующую скобку (первая в группе наборы условий и стопки);
- введите соответствующие выражения на места;
- завершите запись формул.

5. Создайте матричную формулу по образцу:

- выполните команды **Вставка, Формула**, $A = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_1 & b_1 & c_1 \\ a_1 & b_1 & c_1 \end{vmatrix}$ **Вставить новую формулу** ;
- введите левую часть формулы $A=;$
- в пункте **Скобка** выбрать соответствующую скобку (третья во втором ряду);
- затем щелкнув в появившийся квадрат
- в пункте **Матрица** выбрать соответствующую матрицу;
- в первый слот первой строки вставьте соответствующий шаблон **Индекса**;
- в каждый квадрат записать соответствующие символы;
- аналогичным образом заполните остальные слоты, завершите запись формулы.

Вставка символа

Для того чтобы вставить символ в текст, необходимо дать команду **Вставка, Символ** и выбирается нужный символ.

7. Составьте математическое выражение $\lim f(x)=A \Leftrightarrow \forall \varepsilon > 0 \exists \delta > 0 \dots$, используя символьный набор формул, для этого:

- установите курсор в то место текста, куда нужно вставить символ;
- выберите в меню команду **Вставка, Символ, Другие символы**;
- в поле **Шрифт** установите тип шрифта **Symbol**, выберите нужный символ и нажмите кнопку «Вставить» затем «Закрыть» и так с каждым символом.

8. Сохраните документ под именем **Формулы**.

Контрольные вопросы и задания

1. Как открыть уже существующий документ.
2. Как сохранить, созданный документ.
3. Каковы основные правила ввода текста.
4. Способы выделения текста. Для чего используется выделение текста.
5. Понятие абзаца.
6. Использование линейки для форматирования текста
7. Какие возможности предоставляет Word для преобразование внешнего вида текста.
8. Способы редактирования структуры таблицы:
 - объединение ячеек;
 - разбиение ячеек;
 - удаление строк и столбцов;
9. Понятие и назначение буфера обмена.
10. Как можно добавить столбцы в таблицу.
11. Как можно изменить ширину и высоту строк и столбцов.
12. Способы вставки рисунков.
13. Возможности WordArt для оформления документов
14. Какие Вы использовали специальные шрифты, и каким образом можно вставить в документ символ?
15. Нумерация страниц.
16. Сохранение документа.

Задание 1. Оформить докладную записку по образцу.

Краткая справка. Верхнюю часть докладной записи оформляйте в виде таблицы (2 столбца и 1 строка; тип линий – нет границ). Этот прием оформления позволит выполнить разное выравнивание в ячейках таблицы: в левой ячейке – по левому краю, в правой – по центру.

Образец задания

Сектор аналитики и экспертизы

Директору Центра ГАНЛ
Н.С.Петрову

Докладная записка

03.11.2002

Сектор не может завершить в установленные сроки экспертизу проекта маркетингового исследования фирмы «Астра-Н» в связи с отсутствием полных сведений о финансовом состоянии фирмы.

Прошу дать указания сектору технической документации предоставить полные сведения по данной фирме.

Приложение: протокол о некомплектности технической документации фирмы «Астра-Н».

Руководитель сектора

аналитики и экспертизы

(подпись)

М.П.Спелов

Задание 2. Оформить документ, содержащий таблицу, по образцу. Произвести расчет штатной численности по каждой группе оплаты труда.

Краткая справка. Верхнюю часть документа оформите с использованием таблицы (тип линий – нет границ). Произведите расчет суммы по столбцам. Дату вставьте командой

Вставка/Дата.

ОАО «Прогресс»
03.02.2006 № 38
Москва

Утверждаю
Генеральный директор
Б.Н.Добров
10.02.2006

Структура и штатная численность ОАО «Прогресс» на 2003 г.

Наименование должностей	Штатная численность и группы по оплате труда				
	Первая	Вторая	Третья	Четвертая	Пятая
Генеральный директор	1				
Главный бухгалтер	1				
Сотрудники бухгалтерии		2	2		
Старшие специалисты		3	7	1	
Специалисты			4	5	6
Итого	?	?	?	?	?

Задание 3. Оформить схему по образцу.



Задание 4. Набрать текст и формулу по образцу.

Образец задания

Точки $X_1 = -1$, $X_2 = 5/4$, $X_3 = 2$ для числовую ось на четыре промежутка.

Найдем знаки произведения на каждом интервале и отметим их на схеме. Решением неравенства $(4X - 5)(X - 2)(X + 1) > 0$ является объединение двух промежутков $[-1; 5/4]$ и $[2; \infty]$.

Решением неравенства является объединение промежутков $[-1; 5/4]$ и $[2; 3]$. Серединами этих промежутков являются числа 0,125 и 2,5.

Ответ: 0,125; 2,5.

Пример.

$$(2X + 1) : (X^2 - Y^2 + 1) > 2/(X - 2),$$

где $Y = (-X)^{1/2}$.

Решение: Область допустимых значений (ОДЗ)

$$-X \geq 0 \Leftrightarrow X \leq 0;$$

$$X - 2 \neq 0 \Leftrightarrow X \neq 2;$$

$$X \leq 0 \Rightarrow E = [-\infty; 0].$$

При $X \in E$ неравенство примет вид

$$\frac{2X + 1}{X^2 + X + 1} > \frac{2}{X - 2} \Leftrightarrow \frac{(2X + 1)(X - 2) - 2(X^2 + X + 1)}{(X^2 + X + 1)(X - 2)} > 0 \Leftrightarrow \frac{-5X - 4}{(X^2 + X + 1)(X - 2)} > 0.$$

Квадратный трехчлен $X^2 + X + 1$ положителен при всех X , так как его дискриминант отрицателен при $(X^2 + X + 1) > 0$, получим равносильное неравенство.

2.3 Лабораторная работа № 3 (6 часов)

Тема: «Программирование на алгоритмическом языке высокого уровня»

2.3.1 Цель работы: Знакомство со средой программирования Turbo Pascal. Изучение структуры программы, стандартных функций, оператора присваивания и процедур ввода-вывода.

2.3.2 Задачи работы:

- Научиться создавать программы на языке Turbo Pascal с использованием стандартных функций.
- Для закрепления и проверки полученных навыков, необходимо ответить на контрольные вопросы и выполнить дополнительные задания. Результаты работы продемонстрировать преподавателю.

Работа должна содержать:

- номер варианта для выполнения задания и условие своего варианта;
- блок-схемы решения задач;
- тексты программ;
- полученные при расчетах численные результаты;

2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:

- Персональный компьютер

2.3.4 Описание (ход) работы:

1 Порядок выполнения работы

- Загрузить среду программирования Turbo Pascal. Ознакомиться с назначением основных пунктов меню, содержанием строки состояния и работой в справочной системе.

2. Составить и отладить программу, включающую в себя:

- сообщение о вводе двух чисел;
- ввод значений двух вещественных чисел;
- вывод в виде таблицы суммы, разности, произведения и среднеарифметического значения 1-го и 2-го числа с разным числом знаков в дробной части.

3. Составить и отладить программу вычисления функции $y=f(x)$ с использованием стандартных функций языка Turbo Pascal в соответствии с заданным преподавателем вариантом (Приложение А).

4. Провести расчеты для $x=0.1$, $x=0.2$, $x=0.3$, $x=0.4$ и $x=0.5$.

5. Записать полученные значения в виде таблицы.

Теоретические положения

Элементы языка

Алфавит языка

При записи программ разрешены символы:

- буквы латинского алфавита A-Z (в любом регистре), а также знак подчеркивания _;
- буквы русского алфавита А-Я;
- цифры 0-9;
- специальные символы $>$ $<$ $=$ $+$ $-$ $/$ $*$ $[$ $]$ $($ $)$ $\{$ $\}$ $.$ $,$ $:$ $;$ $^$ $@$ $'$ $$$ $#$
- пары символов (их нельзя разделять пробелами) $<>$ $<=$ $>=$ $:=$ $(*$ $*$) $(.$ $.)$
- пробелы (рассматриваются как ограничители идентификаторов, констант, чисел, зарезервированных слов).

Идентификаторы

Неделимые последовательности символов алфавита образуют слова -идентификаторы, используемые для обозначения констант, переменных, процедур, функций и т.д.

Идентификатор должен начинаться с буквы или символа подчеркивания, не должен содержать пробелов и специальных символов. Примеры идентификаторов:

name

WorkPhone

_SUM1

Константы

В качестве констант могут использоваться числа, логические константы, символы и строки символов.

Целые числа записываются со знаком или без него по обычным правилам и могут иметь значение от -2147483648 до $+2147483647$.

Вещественные числа записываются со знаком или без него с использованием десятичной точки и/или экспоненциальной части. Экспоненциальная часть начинается символом e или E, за которым могут следовать знаки «+» или «-» и десятичный порядок. Символ e (E) означает десятичный порядок и смысл «умножить на 10 в степени». Например, запись $3.14E5$ означает $3,14 \times 10^5$, а запись $-17e-2$ – это -17×10^{-2} .

Логическая константа – это либо слово FALSE (ложь) либо слово TRUE (истина).

Символьная константа – это любой символ, заключенный в апострофы:

'z' – символ z;

'ф' – символ ф.

Строковая константа – последовательность символов, заключенная в апострофы. Если в строке нужно указать сам символ апострофа, он удваивается, например:

'Это - строка символов ';

'That's string'.

Строка символов может быть пустой, т.е. не иметь никаких символов в обрамляющих ее апострофах.

Выражения

Выражение задает порядок выполнения действий над элементами данных и состоит из операндов (констант, переменных, функций, круглых скобок и знаков операций).

Действия в выражении выполняются слева направо с соблюдением старшинства (в порядке убывания):

- 1) Not (логическое отрицание);
- 2) * (умножение), / (деление), div (целочисленное деление), mod (целочисленное деление с остатком по модулю), and (логическая операция “И”);
- 3) + (сложение), - (вычитание), or (логическая операция “ИЛИ”);
- 4) операции отношений: = (равно), \neq (не равно), < (меньше), > (больше), \leq (меньше или равно), \geq (больше или равно).

Для изменения порядка выполнения действий используются круглые скобки. Число открывающихся скобок равно числу закрывающихся. Любое выражение в скобках вычисляется раньше, чем выполняется операция, предшествующая скобкам.

Все составные части выражения записываются в одну строку, например, выражение запишется в виде $(a+b*x)/(c+d)$.

В выражение могут входить функции. Наиболее часто употребляемые функции называют стандартными. Для работы с ними не надо ни заказывать библиотеку, ни описывать их предварительно в программе. Примеры стандартных математических функций:

ABS(x) – модуль x ($|x|$);

SQR(x) – квадрат числа x (x^2);

SQRT(x) – квадратный корень из x ();

LN(x) – натуральный логарифм от x ($\ln x$);

EXP(x) – e в степени x (e^x);

SIN(x) – синус x ($\sin x$);

COS(x) – косинус x ($\cos x$);

ARCTAN(x) – арктангенс x ($\arctg x$).

Аргумент этих функций может быть как вещественным, так и целым. Результат – всегда вещественный.

Типы данных

Любые данные, т.е. константы, переменные, значения функций или выражения в Турбо Паскале характеризуются своими типами. Тип определяет множество допустимых значений, которые может иметь тот или иной объект, а также множество допустимых операций, которые применимы к нему. Все типы данных разделяются на две группы – простые и составные.

К простым (скалярным) типам относятся, например:

INTEGER - данные этого типа могут принимать только целые значения (положительные, отрицательные, 0) в диапазоне от -32768 до +32767;

REAL - величины этого типа могут принимать только вещественные значения (числа с дробной частью, целая часть от дробной отделяется точкой);

BOOLEAN - логический тип, принимает два значения TRUE (истина) и FALSE (ложь);

CHAR – символ.

Примером составного (структурированного типа) может служить тип STRING - строка символов. Этот тип широко используется для обработки текстов.

Структура программы

Структура программы должна быть такой:

<Заголовок программы>

{Блок описаний}

BEGIN

{Раздел исполняемых операторов}

END.

Заголовок программы

В заголовке указывается имя программы. Общий вид заголовка:

program n;

здесь n – имя программы.

Заголовок программы необязателен, его можно опускать без каких-либо последствий для программы.

Блок описаний

В блоке описаний объявляются идентификаторы типов, констант, переменных, а также метки, процедуры и функции. Блок описаний может состоять из пяти разделов, которые должны следовать в строго определенном порядке:

- 1) раздел меток (label);
- 2) раздел констант (const);
- 3) раздел типов (type);
- 4) раздел переменных (var);
- 5) раздел процедур и функций.

Раздел меток (label)

Любой выполняемый оператор может быть снабжен меткой – положительной константой, содержащей не более 4-х цифр. Метка отделяется от оператора двоеточием. Все метки, встречающиеся в программе, должны быть описаны в разделе label. Общий вид:
label 11, 12, 13...;
здесь 11, 12, 13... - метки.

Пример.

label 20;

Пусть оператор a:=b; имеет метку 20. Тогда этот оператор выглядит так:
20: a:=b ;

Раздел констант (const)

Если в программе используются константы, имеющие достаточно громоздкую запись (например, число π с 8-ю знаками), либо сменные константы (например, для задания варианта программы), то такие константы обычно обозначаются какими-либо именами и описываются в разделе const. Это делает программу более наглядной и удобной при отладке и внесении изменений.

Общий вид:

const a1 = c1; a2 = c2; ...

Здесь a1, a2, ... – имя константы, c1, c2, ... – значение константы.

Пример.

const pi=3.14; c=2.7531;

Раздел типов (type)

Если в программе вводится тип, отличный от стандартного, то этот тип описывается в разделе type:

type t1=<вид типа>;
t2=<вид типа>;
.

где t1 и t2 – идентификаторы вводимых типов.

Пример.

Type color=(red, yellow, green, blue);

Здесь описан тип color, задаваемый перечислением значений.

Раздел переменных (var)

В разделе var вводится имя каждой переменной и указывается, к какому типу эта переменная принадлежит:

```
var v11, v12, ...: type1;
      v21, v22, ...: type2; ...
```

Здесь v11, v12, ... - имена переменных; type1 – тип переменных v11, v12, ...; type2 - тип переменных v21, v22, ...

Пример.

```
var k,i,j:integer; a,b:real;
```

Раздел процедур и функций

Те алгоритмы, которые оформляются как подпрограммы (процедуры и функции) помещаются в главной программе после раздела var и перед begin программы.

Раздел действий (операторов).

Эта часть программы начинается с ключевого слова begin и заканчивается словом end, после которого должна стоять точка (end.). Раздел действий - это выполняемая часть программы, состоящая из операторов.

Комментарии

Комментарий – это произвольная последовательность любых символов, поясняющая текст программы. Комментарий разрешается вставлять в любое место программы, где по смыслу может стоять пробел. В качестве ограничителей комментария используются фигурные скобки « { » и « } », а также пары символов: « (* » - слева от комментария и « *) » - справа от него:

```
{ Это комментарий }
(* Это тоже комментарий *)
```

Оператор присваивания

Под операторами в языке Паскаль подразумевают описание действий. Операторы отделяются друг от друга точкой с запятой. Если оператор стоит перед end, until или else, то в этом случае точка с запятой не ставится.

Общий вид оператора присваивания:

```
v:=a;
```

здесь v – переменная, a – выражение, := - операция присваивания. Выражение a может содержать константы, переменные, названия функций, знаки операций и скобки. В операторе v:=a переменная v и выражение a должны иметь один и тот тип.

Примеры.

```
f:=3*c+2*sin(x);
```

```
x:=x+1;
```

Замечание. Разрешается присваивать переменной типа real выражение типа integer. Но нельзя присваивать переменной типа integer выражение типа real.

Процедура ввода информации

Общий вид:

```
Read (v1, v2, ...,vn);
```

или

```
Readln (v1, v2, ...,vn);
```

здесь v1, v2, ...,vn – идентификаторы переменных.

Значения переменных вводятся с клавиатуры и должны соответствовать типам переменных. В случае использования процедуры readln, после ввода происходит переход на следующую строку.

Процедура вывода информации на печать

Общий вид оператора:

```
write(p1, p2, ..., pn);
```

или

```
writeln(p1, p2, ..., pn);
```

Здесь p1, p2, ..., pn - список выражений, значения которых выводятся на печать.

Оператор write оставляет курсор в конце выведенной строки текста.

В случае использования процедуры writeln, после печати происходит переход на следующую строку.

Кроме значений выражений, на печать можно выводить и произвольный набор символов, заключенный в апострофы, например

```
writeln('p=',p);
```

Этот оператор выполняется так: сначала выводятся символы, заключенные в апострофы. Затем выводится значение переменной p, например 13.5. На экране в результате работы оператора появится:

p=13.5

Пример 1. Вычислить длину окружности радиуса 5,785.

```
program t10;
(* Программа вычисления длины окружности*)
const r=5.785;
var l:real;
begin
l:=2*3.1416*r;
writeln('l=',l);
end.
либо
program t11;
var r:real;
begin
readln(r);
writeln('l=',2*3.1416*r);
end.
```

Имеется возможность задать ширину поля (число позиций) M для выводимой величины P: Write (P1:M1, P2:M2, ...PN:MN);

Для вещественных чисел можно задавать поля M и N, где M – общее число позиций, отводимых под все число, N – число позиций под его дробную часть.

Например,

```
Write (P:10:2);
```

Здесь под P отводится 10 позиций, 2 из них под дробную часть.

Пример 2. Назначение следующей программы - ввести с клавиатуры два целых числа, найти результат деления первого числа на второе и вывести числа и полученный результат на экран в виде таблицы.

```
Program Input_Output;
```

{Программа вводит два целых числа

и выводит частное от деления 1-го на 2-е}

Var

```
n1,n2:Integer; {n1 и n2 - вводимые целые}
```

```
x:real; {x - результат}
```

Begin

```
Write('n1='); {Сообщение о вводе n1}
```

```
ReadLn(n1); {Ввод n1}
```

```
Write('n2='); {Сообщение о вводе n2}
ReadLn(n2); {Ввод n2}
X:=n1/n2; {Вычисление результата}
WriteLn('-----'); {Печать таблицы}
WriteLn('| n1 | n2 | Частное |');
WriteLn('-----');
WriteLn(n1:8,n2:8,x:8:4);{Вывод n1, n2 и x}
WriteLn('-----');
End.
```

Методические рекомендации

1 Запуск программы Turbo Pascal выполняется любым из стандартных способов запуска, предусмотренных в ОС Windows (например, с помощью ярлыка на рабочем столе).

Для перехода к выбору команд главного меню используется клавиша F10. Для возврата в режим редактирования нужно нажать клавишу ESC.

Для получения справки используются клавиши:

F1 – получение контекстно-зависимой справки;

SHIFT+F1 – выбор справки из списка доступных справочных сообщений;

CTRL+F1 – получение справки о нужной стандартной процедуре, функции, о стандартной константе или переменной.

2 При решении задач можно воспользоваться примерами программ из п.5.7. Текст программы набирается в текстовом редакторе среды Turbo Pascal.

После заполнения очередной строки следует нажать клавишу ENTER, чтобы перевести курсор на следующую строку.

Наиболее часто используемые команды редактора Turbo Pascal:

- смещение курсора

Page Up – на страницу вверх;

Page Down – на страницу вниз;

Home – в начало текущей строки;

End – в конец текущей строки;

Ctrl+ Page Up – в начало текста;

Ctrl+ Page Down – в конец текста.

- команды редактирования

Backspase – стирает символ слева от курсора;

Delete - стирает символ, на который указывает курсор;

Ctrl+Y – стирает строку, в которой расположен курсор;

Enter – вставляет новую строку, разрезает старую;

Ctrl+Q L – восстанавливает измененную строку (действует, если курсор не покидал строку после ее изменения).

- работа с блоками

Ctrl+K B – начинает выделение блока;

Ctrl+K K – заканчивает выделение блока (кроме того, блок можно выделить с помощью мыши);

Ctrl+K Y – уничтожает выделенный блок;

Ctrl+K C – копирует блок;

Ctrl+K V – перемещает блок на новое место;

Ctrl+K W – записывает блок в файл;

Ctrl+K R – читает блок из файла;

Ctrl+K P – печатает блок.

Набранный текст программы запишите в файл. Клавишей F2 вызывается окно диалога, в котором следует задать имя файла.

После подготовки текста программы нужно попытаться исполнить ее, т.е. откомпилировать программу, связать ее (если это необходимо) с библиотекой

стандартных программ и функций, загрузить в оперативную память и передать ей управление. Эта последовательность действий – прогон программы – осуществляется после нажатия клавиш **CTRL+F9**.

Если в программе нет синтаксических ошибок, то все действия выполняются последовательно одно за другим, при этом в небольшом окне сообщается о количестве откомпилированных строк и объеме доступной памяти. Перед передачей управления загруженной программе среда выводит на экран окно прогона программы, а после завершения работы восстанавливает на экране окно редактора.

Если на каком-то этапе среда обнаружит ошибку, она прекращает дальнейшие действия, восстанавливает окно редактора и помещает курсор на ту строку программы, при компиляции или исполнении которой обнаружена ошибка. При этом в верхней строке редактора появляется диагностическое сообщение о причине ошибки. Все это позволяет отладить программу, т.е. устранить в ней синтаксические ошибки и убедиться в правильности ее работы. Некоторые сообщения об ошибках, их перевод и пояснения приведены в приложении Б.

По оператору **read** (или **readln**) вызывается встроенная процедура ввода данных и программа останавливается в ожидании ввода. Необходимо набрать на клавиатуре нужные данные и нажать клавишу **ENTER**.

С помощью клавиш **ALT+F5** в любой момент можно просмотреть данные, выданные на экран в результате прогона программы.

2. Контрольные вопросы

1. Как в программе на языке Turbo Pascal описываются переменные?
2. Какие бывают типы переменных?
3. Какой вид имеет оператор присваивания?
4. Каким символом отделяются друг от друга операторы в программе?
5. В каких случаях после оператора не ставится точка с запятой?
6. Какая процедура служит для вывода информации на печать?
7. Какая процедура служит для ввода значений с клавиатуры?
8. Какие функции служат для вычисления квадрата, квадратного корня, модуля, экспоненты числа или числового выражения?
9. Какие стандартные тригонометрические функции существуют в языке Turbo Pascal?
10. Как в среде Turbo Pascal запустить программу на выполнение?

ПРИЛОЖЕНИЕ А

Номер варианта Функция

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

ПРИЛОЖЕНИЕ Б

Сообщения об ошибках.

2 Identifier expected (Не указан идентификатор).

В этом месте должен находится идентификатор.

3 Unknown identifier (Неизвестный идентификатор)

Этот идентификатор не был описан.

4 Duplicate identifier (Двойной идентификатор).

Попытка дважды описать один и тот же идентификатор.

5 Syntax error (Синтаксическая ошибка).

В исходном тексте найден недопустимый символ.

10 Unexpected end of file (Не найден конец файла).

Причины этого сообщения могут быть следующие:

- исходный файл закончился перед последним END основного раздела операторов; вероятно в программе неодинаковое количество операторов BEGIN и END;
- не закончен комментарий.

11 Line too long (Слишком длинная строка).

Максимальная длина строки, обрабатываемая компилятором, равна 126 символам.

12 Type identifier expected (Здесь нужен идентификатор типа).

Не указан тип идентификатора.

16 Disk full (Диск заполнен).

Нужно удалить некоторые файлы или воспользоваться новым диском..

20 Variable identifier expected (Отсутствует идентификатор переменной).

На этом месте должен быть идентификатор переменной.

21 Error in type (Ошибка в объявлении типа).

Объявление типа не может начинаться с этого символа.

26 Type mismatch (Несоответствие типа).

Это сообщение может быть вызвано следующими причинами:

- несовместимые типы переменной и выражения в операторе присваивания;
- тип выражения не совместим с типом индекса при объявлении массива;
- несовместимые типы операндов в выражении.

33 Label identifier expected (Нужен идентификатор метки)

Метка не обозначена с помощью идентификатора, как это требуется из контекста программы.

36 BEGIN expected (Нужен BEGIN)

37 END expected (Нужен END)

38 Integer expression expected (Нужно выражение типа Integer).

41 Operand types do not match operator (Типы операндов не соответствуют операции).

Данная операция не может быть применена к указанным operandам.

42 Error in expression (Ошибка в выражении)

Данный символ не может участвовать в выражении указанным образом. Возможно, не указана операция между двумя operandами.

50 DO expected (Нужен оператор DO)

57 THEN expected (Требуется THEN)

58 TO or DOWNTO expected (Требуется TO или DOWNTO)

62 Division by zero (Деление на ноль)

Предшествующая операция пытается выполнить деление на ноль.

64 Cannot Read or Write variables of this type (Нет возможности считать или записать переменные данного типа).

Нарушены следующие ограничения:

- процедуры READ и READLN могут считывать переменные символьного, целого, действительного и строкового типов;

- процедуры WRITE и WRITELN могут выводить переменные символьного, целого, действительного, логического и строкового типов.

76 Constant out of range (Константа нарушает границы).

Возможные причины сообщения:

- попытка указать индекс массива, выходящий за его границы;

- попытка присвоить переменной значение, выходящее за границы, допустимые для типа этой переменной.

79 Integer or real expression expected (Нужно выражение вещественного или целого типа).

81 Label already defined (Метка уже определена).

Данная метка уже помечает оператор.

85 «;» expected (Нужно указать «;»).

97 Invalid FOR control variable (Неправильный параметр цикла оператора FOR).

98 Integer variable expected (Нужна переменная целого типа).

Предшествующая переменная должна иметь целый тип.

103 Integer or real variable expected (Нужна переменная типа INTEGER or REAL).

113 Error in statement (Ошибка в операторе).

Данный символ не может быть первым символом в операторе.

207 Invalid floating point operation (Недопустимая операция с плавающей запятой) .

Возможные причины сообщения:

- отрицательный аргумент функции SQRT;

- аргумент функции LN равен нулю или имеет отрицательное значение.

Разработал

Н.В. Андреева