МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Информатика

Курс лекций для обучающихся специальности
09.02.07-«Информационные системы и программирование»

Дербент 2024 г

УДК 316

ББК 60.5

Информатика. Курс лекций для обучающихся специальности09.02.07- «Информационные системы и программирование»/-Дербент, ДГТУ, 2024. - 96с.

Составитель: Идрисова М.В., преподаватель кафедры естесвеннонаучных, гуманитарных и специальных дисциплин в филиале ДГТУ в г.Дербенте

Рецензенты:

1. Очаковская О.А., к. ф-м н., доцент кафедры ЕГОиСД филиала ДГТУ в г.Дербенте.

2. Гасанов В.М., к.т.н**.,** преподаватель ГБПОУ РД «Колледж экономики и права», г.Дербент.

Курс лекций по учебной дисциплине «Информатика», разработанный для специальности среднего профессионального образования09.02.07- «Информационные системы и программирование», предназначен для студентов I курса, очной и заочной формы обучения.В курсе лекций изложены теоретические сведения, необходимые для успешного освоения учебной дисциплины, изучения основных понятий в области информатики, информационных технологий и вычислительных сетей. Подробно изложены вопросы, связанные с применением сетевых средств ИКТ. Курс лекций содержит вопросы после каждой лекции для эффективной подготовки студентов к экзамену.

РЕГ№____

Печатается по решению Ученого совета ФГБОУ ВО «ДГТУ» в г.Махачкале. Протокол № ____ от ____ _____2024 г.

СОДЕРЖАНИЕ:

Введение……..4

Лекция 1.Информатика как наука. Предметная область информатики. Классификация информационных систем. Виды профессиональной информационной деятельности человека с использованием технических средств и информационных ресурсов…….5

Лекция 2. Кодирование и шифрование информации 10

Лекция 3.Программное обеспечение. Системы программирования……...27

Лекция 4..Оформление документов с помощью программы MicrosoftWord……32

Лекция 5. Понятие алгоритма и основные алгоритмические структуры. Языки программирования. Предпрограммная подготовка задачи……...44

Лекция 6.Базы данных. Программы управления базами данных……62

Лекция 7.Технология обработки информации в электронных таблицах……..69

Лекция 8.Вычислительные комплексы и сети…….75

Лекция 9.Аппаратное устройство компьютера……87

Список литературы ……96

Введение

Процессы информатизации современного общества и тесно связанные с ними процессы информатизации всех форм образовательной деятельности характеризуются процессами совершенствования и массового распространения современных информационных и коммуникационных технологий (ИКТ). Подобные технологии активно применяются для передачи информации и обеспечения взаимодействия преподавателя и обучаемого в современных системах открытого и дистанционного образования. Современный студент должен не только обладать знаниями в области ИКТ, но и быть специалистом по их применению в своей дальнейшей профессиональной деятельности.

Основным средством ИКТ для информационной среды любой системы образования является персональный компьютер, возможности которого определяются установленным на нем программным обеспечением. Основными категориями программных средств являются системные программы, прикладные программы и инструментальные средства для разработки программного обеспечения.

В современных системах образования широкое распространение получили универсальные офисные прикладные программы и средства ИКТ: текстовые процессоры, электронные таблицы, программы подготовки презентаций, системы управления базами данных, органайзеры, графические пакеты и т.п.

С появлением компьютерных сетей и других, аналогичных им средств ИКТ образование приобрело новое качество, связанное в первую очередь с возможностью оперативно получать информацию из любой точки земного шара. Через глобальную компьютерную сеть Интернет возможен мгновенный доступ к мировым информационным ресурсам (электронным библиотекам, базам данных, хранилищам файлов, и т.д.). В самом популярном ресурсе Интернет - всемирной паутине WWWопубликовано порядка двух миллиардов мультимедийных документов.

В сети доступны и другие распространенные средства ИКТ, к числу которых относятся электронная почта, списки рассылки, группы новостей, чат.

Для обеспечения эффективного поиска информации в телекоммуникационных сетях существуют автоматизированные поисковые средства, цель которых - собирать данные об информационных ресурсах глобальной компьютерной сети и предоставлять пользователям услугу быстрого поиска.

С помощью сетевых средств ИКТ становится возможным широкий доступ к учебно-методической и научной информации, организация оперативной консультационной помощи, моделирование научно-исследовательской деятельности, проведение виртуальных учебных занятий.

.

Лекция №1.

Тема: «Информатика как наука.Предметная область информатики.Классификация информационных систем. Виды профессиональной информационной деятельности человека с использованием технических средств и информационных ресурсов»

План:

1. Информатика как наука.
2. Предметная область информатики.
3. Классификация информационных систем.
4. Виды профессиональной информационной деятельности человека с использованием технических средств и информационных ресурсов.

1. Информатика как наука.

Изучая мир, человек накопил большое количество знаний. Необходимость обработки информации стала для человека одной из важнейших задач. Эффективному решению этой задачи способствовало развитие вычислительной (компьютерной) техники и информационных технологий.

Первое общепринятое определение информатики сложилось в 60-е годы прошлого столетья и позже широко распространилось в учебной и научно-популярной литературе.

Информатика — отрасль науки, изучающая структуру и общие свойства информации, а также вопросы, связанные с ее сбором, хранением, поиском, переработкой, преобразованием, распространением и использованием в различных сферах деятельности. Она имеет теоретическое и прикладное (практическое) направление.

Как фундаментальная естественная наука информатика занимается изучением свойств информации, а также процессами сбора, хранения, поиска, передачи, переработки, преобразования и использования информации.

Как прикладная дисциплина информатика занимается изучением информационных процессов, т. е. передачей информации; созданием информационных моделей в различных сферах деятельности человека; выработкой рекомендаций по технологии проектирования и разработки систем, их производства, функционирования и т. д.

Фундаментальный и прикладной характер информатики определяют ее как комплексную научно-техническую дисциплину.

В современном мире роль информатики, средств обработки, передачи, накопления информации неизмеримо возросла. Средства информатики и вычислительной техники сейчас во многом определяют научно-технический потенциал страны, уровень развития ее народного хозяйства, образ жизни и деятельности человека.

Для целенаправленного использования информации ее необходимо собирать, преобразовывать, передавать, накапливать и систематизировать. Процессы, связанные с определенными операциями над информацией, называются *информационными процессами.* Получение и преобразование информации является необходимым условием жизнедеятельности любого организма. Даже простейшие одноклеточные организмы постоянно воспринимают и используют информацию, например, температуре и химическом составе среды для выбора наиболее благоприятных условий существования. Живые существа способны не только воспринимать информацию из окружающей среды с помощью органов чувств, но и обмениваться ею между собой.

Человек также воспринимает информацию с помощью органов чувств, а для

обмена информацией между людьми используются языки. За время развития человеческого общества таких языков возникло много. Прежде всего, это родные языки (русский, татарский, английский и др.), на которых говорят многочисленные народы мира. Роль языка для человечества исключительно велика. Без него, без обмена информацией между людьми было бы невозможным возникновение и развитие общества.

2. Предметная область информатики.

Рассмотрим объект и предмет исследования в информатике. Под объектом исследования понимается часть окружающей нас реальной действительности, которая изучается различными науками.

Предмет исследования - исследуемый участок объекта конкретной наукой, объект и предмет исследования относятся друг с другом как общее и частное.

Объект исследования - автоматизированные информационные системы, основанные на электронно-вычислительных машинах и телекоммуникационной технике. Информатика изучает все стороны их проектирования, создания, анализа и использования их на практике.

Если говорить о системе вообще, то это любой объект, который одновременно рассматривается и как единое целое, и как совокупность разнородных элементов, объединенная для достижения поставленной цели (производство, услуги). Добавление к понятию "система" слова "информационная" отражает цель ее создания и функционирования. Системой может называться аппаратная часть компьютера, множество программ для решения конкретных задач, сам компьютер и т.д. Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации; помогают анализировать проблемы и создавать новые продукты.

Информационная система - это система, связанная с существованием информационных процессов.

Информационный процесс – это процесс передачи информации от одного объекта к другому, а также ее создание и преобразование.

Предмет исследования - информационный ресурс. Информатика изучает его сущность, законы функционирования, механизмы взаимодействия с другими ресурсами общества. Наука информатика подводит теоретический фундамент под использование компьютеров для создания, хранения и использования информационного ресурса.

Информационный ресурс – это совокупность документов в архивах, библиотеках, фондах, базах и банках данных и других информационных системах. Информационные ресурсы являются объектами отношений физических, юридических лиц и государства, поэтому подлежат обязательному учету.

Информационный ресурс может быть пассивный и активный. Пассивный информационный ресурс можно использовать только как источник информации (например, книга). Активный информационный ресурс предполагает определенное взаимодействие, в результате которого можно его изменить или получить новый ресурс (например, файл на диске). Резюмируя вышесказанное, можно вывести следующее утверждение (триаду): автоматизированная информационная система реализуется посредством информационных технологий, в результате на выходе мы получаем информационный ресурс.

3. Классификация информационных систем.

Существует множество классификаций информационных систем в зависимости от типа решаемых задач.

Рассмотрим классификацию информационных систем по назначению и по способу функционирования.

По назначению выделяют следующие информационные системы:

- информационно-управляющие системы (системы для сбора, анализа и обработки информации, необходимой для управления организацией, предприятием, отраслью);
- информационно-поисковые системы (системы, основное назначение которых поиск информации, содержащейся в различных базах данных, различных вычислительных системах, разнесенных, как правило, на значительные расстояния. Примером может служить информационно-поисковые системы Интернет: Rambler, Yandex и др.);
- информационно-справочные системы (автоматизированные системы, обеспечивающие пользователей справочной информацией.Например, автоматические справочные системы в аэропортах, вокзалах и др.);
- системы обработки данных. Банки данных (основная функция - обработка и архивация больших объемов данных.Например, системы управления базами данных, объединенные в один банк данных).

По способу функционирования выделяют разомкнутые и замкнутые информационные системы.

В разомкнутой информационной системе получаемая потребителем информация используется произвольно (рис. 1). От потребителя в информационную систему ничего не поступает.

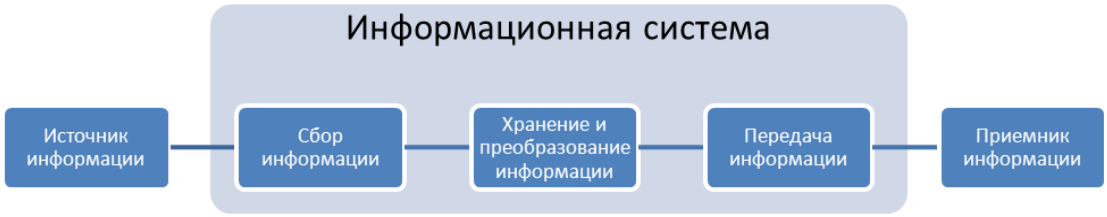


Рис.1. Разомкнутая информационная система

Примером разомкнутой информационной системы служит электронная справочная система каталогов библиотеки. Установленная в библиотеке система обеспечивает любого читателя информацией по интересующей его тематике. Получив по запросу перечень литературы, читатель прекращает взаимодействие с информационной системой, не повлияв ни на ее работу, ни на хранящуюся в ней информацию.

В замкнутой информационной системе (рис. 2) существует обратная связь между ее структурой и потребителем. По каналу обратной связи передается в аппаратно-программную часть информационной системы реакция потребителя на полученную им информацию. Происходит обработка вновь поступивших данных с уже имеющимися данными. Результирующая информация вновь отправляется потребителю.

Примером замкнутой информационной системы служит электронный каталог билетов железнодорожной кассы. При покупке билета вводятся данные об этом в компьютер, для того, чтобы повторно не продать этот же билет.

Рис. 2. Замкнутая информационная система

4. Виды профессиональной информационной деятельности человека с использованием технических средств и информационных ресурсов.

Деятельность человека, связанную с процессами получения, преобразования, накопления и передачи информации, называют *информационной деятельностью*.

В настоящее время компьютеры используются для обработки не только числовой, но и других видов информации. Благодаря этому информатика и вычислительная техника прочно вошли в жизнь современного человека, широко применяются в производстве, проектно-конструкторских работах, бизнесе и многих других отраслях. Разработка способов и методов представления информации, технологии решения задач с использованием компьютеров, стала важным аспектом деятельности людей многих профессий.

Информационная деятельность бывает массовой, специальной и личностной.

Ради построения компьютерных моделей для решения научно-технических задач были созданы первые компьютеры. В настоящее время компьютерное моделирование активно применяется во всех науках, но обеспечить его применение по-прежнему поручают *математикам.* Только фундаментальное математическое образование позволяет сформировать специалиста, владеющего теорией дифференциальных уравнений и численными методами, программированием, компьютерным моделированием, способного построить компьютерную модель реальной системы. При подготовке эти специалисты изучают широкий спектр курсов, связанных с вычислительной техникой и программированием

Профессия *«Информатик»* с квалификацией в некоторой прикладной области утверждена в России в 2000 г. Эта специальность применяется в экономике, юриспруденции, политологии и т. д., где используются информационные системы. Информатик в соответствующей области занимается созданием и сопровождением информационной системы. Например, Информатик-экономист является специалистом по информационным системам в административном управлении, банковском, страховом деле, бухгалтерском учете и т.д. Этот специалист, имеющий глубокую фундаментальную подготовку, может создавать информационно-логические модели объектов, разрабатывать новое программное и информационное обеспечение для решения задач науки, техники, экономики и управления, адаптировать систему на всех стадиях ее жизненного цикла.

Защита информационных ресурсов от несанкционированного доступа, обеспечение безопасности информационных и телекоммуникационных систем особенно актуальны для современного мира. Эти задачи привели к возникновению новой специальности — *специалист по защите информации.* Специалисты изучают защиту информации в автоматизированных системах, в персональных компьютерах, в компьютерных сетях с помощью программных и аппаратных средств. Для обеспечения информационной безопасности применяются технические, программные, организационные и правовые методы.

Вычислительная техника и телекоммуникации составляют аппаратную основу любой информационной технологии. Их разработкой и созданием занимаются *инженеры.* Они владеют принципами построения вычислительных и телекоммуникационных систем, электротехникой и микроэлектроникой, базисом является инженерное образование.

Правовое регулирование в информационной сфере является новой и сложной задачей для государства. В Российской Федерации существует ряд законов в этой области.

Закон «О правовой охране программ для ЭВМ и баз данных» регламентирует юридические вопросы, связанные с авторскими правами на программные продукты и базы данных.

Закон «Об информации, информатизации и защите информации» позволяет защищать информационные ресурсы (личные и общественные) от искажения, порчи, уничтожения. Статья 11 этого закона «информация о гражданах (персональные данные)» содержит гарантии недопущения сбора, хранения, использования и распространения информации о частной жизни граждан (это может делаться лишь на основании решения суда), недопустимости использования собранной любым путем информации для дискриминации граждан по любому признаку.

В Уголовном кодексе РФ имеется раздел «Преступления в сфере компьютерной информации». Он предусматривает наказания за:

- неправомерный доступ к компьютерной информации;
- создание, использование и распространение вредоносных программ для ЭВМ;
- умышленное нарушение правил эксплуатации ЭВМ и их сетей.

Информационная безопасность-совокупность мер по защите информационной среды общества и человека. Информационные угрозы безопасности информации можно разделить на *преднамеренные* (хищение информации, компьютерные вирусы, физическое воздействие на аппаратуру) и *случайные* (ошибки пользователя, ошибки профессионалов, отказы и сбои аппаратуры, форс-мажорные обстоятельства).

Политика безопасности - это совокупность технических, программных и организационных мер, направленных на защиту информации в компьютерной сети.

Техническое обеспечение (ТО) - комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

К современным **техническим средствам** работы с информацией относятся не только компьютеры, но и другие устройства, обеспечивающие ее передачу, обработку и хранение:

1. Сетевое оборудование: модемы, кабели, сетевые адаптеры.
2. Аналого-цифровые и цифро-аналоговые преобразователи.
3. цифровые фото- и видеокамеры, цифровые диктофоны.
4. Записывающие устройства (CD-R, CD-RW, DVD-RW и др.).
5. Полиграфическое оборудование.
6. Цифровые музыкальные студии.
7. Медицинское оборудование для УЗИ и томографии;
8. Сканеры в архивах, библиотеках, магазинах, на экзаменах и избирательных участках;
9. ТВ-тюнеры для подачи телевизионного сигнала в компьютер.
10. Плоттеры и различные принтеры.
11. Мультимедийные проекторы.
12. Флэш-память, используемая также в плеерах и фотоаппаратах.
13. Мобильные телефоны.

Кроме персональных компьютеров существуют мощные вычислительные системы для решения сложных научно-технических и оборонных задач, обработки огромных баз данных, работы телекоммуникационных сетей:

- Многопроцессорные системы параллельной обработки данных (управление сложными технологическими процессами).
- Серверы в глобальной компьютерной сети, управляющие работой и хранящие огромный объем информации.
- Специальные компьютеры для проектно-конструкторских работ.

Вопросы:

1. Что означает термин «информатика», какие направления имеет информатика?

2. Какими вопросами занимается информатика как фундаментальная естественная наука и как прикладная дисциплина?
3. Какова роль информатики в современном мире?
4. Какие информационные системы вы знаете?
5. Что такое информационная деятельность человека? С чем она связана?
6. Какие виды профессиональной информационной деятельности вы можете назвать?
7. Какие современные технические средства вы знаете ?

Лекция №2.

Тема: «Кодированиеи шифрование информации»

План:

1. Универсальность дискретного представления информации.
2. Кодирование и шифрование информации
3. Понятие о кодировании различных видов информации.
4. Выбор способа представления информации в соответствии с поставленной задачей.
5. Представление информации в различных системах счисления
6. Информационная безопасность ИС

1.Универсальность дискретного (цифрового) представления информации

Давайте подумаем об информации как о сигнале. Мы знаем, что сигнал рассматривается с позиции носителя информации по техническим средствам передачи.

Для передачи информации, или, правильнее сказать, данных, используется физический процесс, который может быть описан математической формулой и называется сигналом. Именно сигналы различают по способу их представления как аналоговые и дискретные (см. рис. 1 и 2).

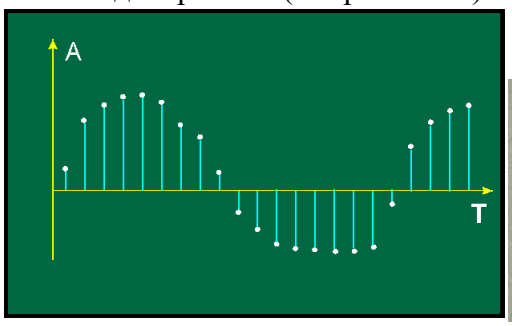


Рис. 3 Аналоговый сигнал

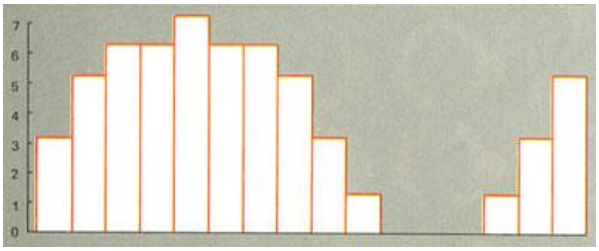


Рис. 4. Дискретный сигнал

Аналоговая информация характеризуется плавным изменением ее параметров. Основные параметры наиболее простых синусоидальных аналоговых сигналов могут непрерывно и плавно меняться.

Дискретная информация базируется на ряде фиксированных уровней представления заданных параметров, взятых в определенные промежутки времени. Если этих уровней много, можно говорить о цифровом представлении информации, то есть когда в определенные дискретные моменты они принимают конкретные дискретные значения. К счастью, аналоговую информацию легко преобразовать в цифровую. Это делают так называемые аналогоцифровые преобразователи (АЦП). Обратное преобразование обеспечивают цифроаналоговые преобразователи (ЦАП).

В качестве носителей аналоговой информации могут использоваться различные физические величины, принимающие различные значения на некотором интервале, например, электрический ток, радиоволна и т.д. При дискретизации, то есть при преобразовании непрерывных изображений и звука в набор дискретных значений в

форме кодов, за основу берется какое-либо конкретное значение, а любые другие, отличающиеся от нормы, просто игнорируются.

Аналоговыми устройствами являются:

- телевизор - луч кинескопа непрерывно перемещается по экрану, чем сильнее луч, тем ярче светится точка, в которую он попадает; изменение свечения точек происходит плавно и непрерывно;
- проигрыватель грампластинок – чем больше высота неровностей на звуковой дорожке, тем громче звучит звук;
- телефон – чем громче мы говорим в трубку, тем выше сила тока, проходящего по проводам, тем громче звук, который слышит собеседник.

К ***дискретным устройствам*** относятся:

- монитор – яркость луча изменяется не плавно, а скачкообразно (дискретно). Луч либо есть, либо его нет. Если луч есть, то мы видим яркую точку (белую или цветную). Если луча нет, мы видим черную точку. Поэтому изображение на экране монитора получается более четким, чем на экране телевизора;
- проигрыватель аудиокомпакт-дисков – звуковая дорожка представлена участками с разной отражающей способностью;
- струйный принтер – изображение состоит из отдельных точек разного цвета.

Человек благодаря своим органам чувств привык иметь дело с аналоговой информацией, а в компьютере информация представлена в цифровом виде. Преобразование графической и звуковой информации из аналоговой формы в дискретную производится путем дискретизации, то есть разбиения непрерывного графического изображения или звукового сигнала на отдельные элементы.

Компьютер работает исключительно с дискретной (цифровой) информацией. Память компьютера состоит из отдельных битов, а значит, дискретна. Датчики, посредством которых воспринимается информация, измеряют в основном непрерывные характеристики - температуру, нагрузку, напряжение и т.д. Встает проблема преобразования аналоговой информации в дискретную форму.

Идея дискретизации непрерывного сигнала заключается в следующем. Пусть имеется некоторый непрерывный сигнал. Можно допустить, что на маленьких промежутках времени значение характеристик этого сигнала постоянно и меняется мгновенно в конце каждого промежутка. "Нарезав" весь временной интервал на эти маленькие кусочки и взяв на каждом из них значение характеристик, получим сигнал с конечным числом значений. Таким образом, он станет дискретным. Непрерывная величина часто ассоциируется с графиком функции, а дискретная - с таблицей ее значений.

Такой процесс называется оцифровкой аналогового сигнала, а преобразование информации - аналого-цифровым преобразованием. Точность преобразования зависит от величины дискретности - частоты дискретизации: чем выше частота дискретизации, тем ближе цифровая информация к качеству аналоговой. Но и тем больше вычислений приходится делать компьютеру и тем больше информации хранить и обрабатывать.

Дискретизация – это преобразование непрерывных изображений и звука в набор дискретных значений в форме кодов.

При передаче дискретных данных по каналам связи применяются два основных типа физического кодирования – на основе синусоидального несущего сигнала и на основе последовательности прямоугольных импульсов. Первый способ часто называется также модуляцией или аналоговой модуляцией, подчеркивая тот факт, что кодирование осуществляется за счет изменения параметров аналогового сигнала. Второй способ обычно называют цифровым кодированием. Эти способы отличаются шириной спектра результатирующего сигнала и сложностью аппаратуры, необходимой для их реализации.

В настоящее время все чаще данные, изначально имеющие аналоговую форму

(речь, телевизионное изображение), передаются по каналам связи в дискретном виде, то есть в виде последовательности единиц и нулей. Процесс представления аналоговой информации в дискретной форме называется ***дискретной модуляцией***. ***Аналоговая модуляция*** применяется для передачи дискретных данных по каналам с узкой полосой частот, типичным представителем которых является канал тональной частоты (телефонная сеть).

В простых вычислительных машинах, в таких, как цифровые электромеханические или аналоговые, перенастройка на различные задачи осуществлялась с помощью изменения системы связей между элементами на специальной коммутационной панели. В современных универсальных компьютерах такие изменения производятся с помощью запоминания в специальном устройстве, накапливающем информацию, той или иной программы ее работы.

В отличие от аналоговых машин, оперирующих непрерывной информацией, современные компьютеры имеют дело с дискретной информацией, на входе и выходе которых в качестве такой информации могут выступать любые последовательности десятичных цифр, букв, знаков препинания и других символов. Внутри системы эта информация кодируется в виде последовательности сигналов, принимающих лишь два различных значения.

В то время как возможности аналоговых машин ограничены преобразованиями строго ограниченных типов сигналов, современные компьютеры обладают свойством универсальности, иными словами, компьютер может производить преобразования любых буквенно-цифровых данных благодаря программе, составленной для выполнения той или иной задачи. Эта способность компьютера достигается за счет универсальности его системы команд, то есть элементарных преобразований информации.

Свойство универсальности компьютера не ограничивается возможностью оперирования одной лишь буквенно-цифровой информацией. В данном виде может быть представлена (закодирована) любая дискретная информация, а также – с любой заданной степенью точности – произвольная непрерывная информация. Таким образом, компьютеры могут рассматриваться как универсальные преобразователи информации. Свойство универсальности современных компьютеров открывает возможность моделирования с их помощью любых других преобразователей информации, в том числе любых мыслительных процессов.

2.Кодирование и шифрование информации

В современном обществе успех любого вида деятельности сильно зависит от обладания определенными сведениями (информацией) и от отсутствия их (ее) у конкурентов. Одним словом, возникновение индустрии обработки информации привело к возникновению индустрии средств ее защитыикактуализациисамойпроблемызащитыинформации, проблемы информационной безопасности.

Проблема защиты информации от несанкционированного доступа заметно обострилась в связи с широким распространением компьютерных сетей (особенно глобальных). Защита информации необходима для уменьшения вероятности разглашения, утечки, умышленного искажения, утраты или уничтожения информации, представляющей определенную ценность для ее владельца.

Однаизнаиболееважныхзадач(всегообщества)– задача *кодирования* сообщений и *шифрования* информации.Вопросамизащитыискрытияинформациизанимается наукакриптология(*криптос – тайный, логос – наука*).Криптология имеет дваосновныхнаправления– криптографиюикриптоанализ.Целиэтих направлений противоположны. Криптография занимается построением и исследованием

математических методов преобразованияинформации, а криптоанализ– исследованиемвозможностирасшифровкиинформациибез *ключа*.

Криптография- наука о защите информации от несанкционированного получения ее посторонними лицами. Сфера ее интересов - разработка и исследование методов шифрования информации.

Шифрованиеэто такое преобразование информации, которое делает исходные данные нечитаемыми и труднораскрываемыми без знания ключа.

Сфераинтересовкриптоанализа-разработкаиисследование методов дешифрования шифрограммы даже без знания ключа.

Подключомпонимается секретная информация, определяющая, какое преобразование из множества возможных преобразований выполняется в данном случае над открытым текстом.

Дешифрование - обратный шифрованию процесс. На основе ключа зашифрованный текст преобразуется в исходный открытый. Процесс получения открытого сообщения из шифрованного без заранее известного ключа называется вскрытием или взломом шифра.

Кодирование- это процесс ввода последовательности символов в специальный формат для целей передачи или хранения.

Код–правилосоответствиянаборазнаковводного*множества*Хзнакам другого *множества* Y.Есликаждому символуХпри*кодировании*соответствуетотдельныйзнакY,то это*кодирование*.Еслидлякаждогосимволаиз Yотыщется*по*некоторомуправилуегопрообразв X,тоэтоправило называется декодированием.

Кодирование–процесспреобразованиябукв(слов) алфавитаХвбуквы (слова) алфавита Y (запись в другой системе символов, в другом алфавите). Приэтомобычнокодированиемназываютпереводинформациис

«человеческого»языканаформальный,например,вдвоичныйкод,адекодированием– обратный переход. Один символ исходного сообщения можетзаменятьсяодним символом новогокодаилинесколькимисимволами, а может быть и наоборот – несколько символов исходного сообщения заменяются одним символом в новом коде.

Шифрованиепредставляетсобой *сокрытиеинформации*от неавторизованных лиц с предоставлением в это же время авторизованным пользователям доступа к ней. Пользователи называются авторизованными, если у них есть соответствующий *ключ* для дешифрования информации.

Целью любой системы шифрования является максимальноеусложнение получения доступа к информации неавторизованными лицами, даже если у них есть зашифрованный текст и известен *алгоритм*, использованный для шифрования. Пока неавторизованный*пользовател*ьне обладает ключом, секретность и *целостность* информации не нарушается.

С помощью шифрования обеспечиваются три состояния безопасности информации.

- Конфиденциальность. Шифрование используется для *сокрытия информации*от неавторизованных пользователей при передаче или при хранении.
- Целостность. Шифрование используется для предотвращения изменения информации при передаче или хранении.
- Идентифицируемость. Шифрование используется для аутентификацииисточникаинформациииипредотвращенияотказаотправителяинформациио ттогофакта,чтоданныебылиотправленыименно им.

Целишифрования

Шифрование применяется для хранения важной информации в ненадёжных источниках и передачи её по незащищённым каналам связи.

Такаяпередачаданныхпредставляетизсебядвавзаимнообратныхпроцесса:

- Передотправлениемданныхполинииисвязиилипередпомещением на хранение они подвергаются *зашифрованию*.
- Для восстановления исходных данных из зашифрованных к ним применяется процедура *расшифрования*.

Шифрование изначально использовалось только для передачи конфиденциальной информации. Однако впоследствии шифровать информацию начали с целью её хранения в ненадёжных источниках. Шифрование информации с целью её хранения применяется и сейчас, это позволяетизбежатьнеобходимостивфизическизащищённомхранилище[4][5].

Шифромназывается пара алгоритмов, реализующих каждое из указанных преобразований. Эти алгоритмы применяются к данным с использованием ключа. Ключи для шифрования и для расшифрования могут различаться, а могут быть одинаковыми. Секретность второго (расшифровывающего) из них делает данные недоступными для несанкционированного ознакомления, а секретность первого (шифрующего) делает невозможным внесение ложных данных. В первых методах шифрования использовались одинаковые ключи, однако в 1976 году были открытыалгоритмыспримененим разныхключей.Сохранениеэтихключей в секретности и правильное их разделение между адресатами является очень важной задачей с точки зрения сохранения конфиденциальности передаваемой информации. Эта задача исследуется в теории управления ключами (в некоторых источниках она упоминается как разделение секрета).

В настоящий момент существует огромное количество методов шифрования. Главным образом эти методы делятся, в зависимости от структуры используемых ключей, на симметричные методы и асимметричные методы. Кроме того, методы шифрования могут обладать различной криптостойкостью и по-разному обрабатывать входные данные— блочные шифры и поточные шифры. Всеми этими методами, их созданием и анализом занимается наука криптография.

Различают два типа алгоритмов шифрования *симметричные (с секретным ключом)*и*асимметричные (с открытым ключом)*. В первом случае обычно ключ расшифрования совпадает с ключом зашифрования, т.е. $K_d = K_e = K$, либо знание ключа зашифрования позволяет легко вычислить ключ расшифрования. В асимметричных алгоритмах такая возможность отсутствует: для зашифрования и расшифрования используются разные ключи, причем знание одного из них не дает практической возможности определитьдругой.Поэтому,еслиполучательАинформациисохраняетвсекретеключрасшифрования $K_{dA} = SK_A$,ключ зашифрования $K_{eA} = PK_A$ может быть сделан общедоступным (*SK -secretkey, PK - publickey*).

Методышифрования.

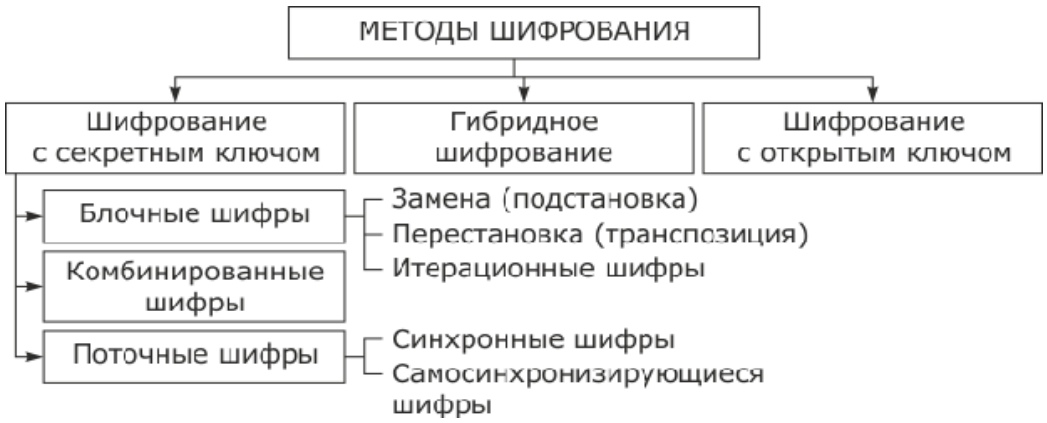


Рис.5.Классификацияметодовшифрованияинформации

3.Понятие о кодировании различных видов информации. Применение кодирования

Естественные языки обладают большой избыточностью для экономии памяти,объем которойограничен,имеетсмыслликвидироватьизбыточноститекста,существуютнесколькос пособов:

1. *Переход от естественных обозначений к более компактным.* Этот способ применяется для сжатия записи дат, номеров изделий, уличных адресов и т.д. Идея способа показана на примере сжатия записи даты.Обычно мы записываем дату в виде 10. 05. 01. , что требует 6 байтов памяти ЭВМ.Однакоясно,чтодляпредставленияднядостаточно5битов,месяца-4, года- не более 7, т.е. вся дата может быть записана в 16 битах или в 2-х байтах.

2. *Подавление повторяющихся символов.* В различных информационных текстах часто встречаются цепочки повторяющихся символов, например пробелы или нули в числовых полях. Если имеется группа повторяющихся символов длиной более 3, то ее длину можно сократить до трех символов. Сжатая таким образом группа повторяющихся символов представляет собой триграф S P N , в котором S – символ повторения; P – признак повторения; N- количество символов повторения, закодированных в триграфе. В других схемах подавления повторяющихся символов используют особенность кодов ДКОИ, КОИ- 7, КОИ-8 , заключающуюся в том , что большинство допустимых в них битовых комбинаций не используется для представления символьных данных.

3. *Кодирование часто используемых элементов данных.* Этот способ уплотнения данных также основан на употреблении неиспользуемых комбинаций кода ДКОИ. Для кодирования, например, имен людей можно использовать комбинации из двух байтов дикраф PN, где P – признак кодирования имени, N – номер имени. Таким образом может быть закодировано 256 имен людей, чего обычно бывает достаточно в информационных системах. Другой способ основан на отыскании в текстах наиболее часто встречающихся сочетании букв и даже слов и замене их на неиспользуемые байты кода ДКОИ.

*Посимвольноекодирование.*Семибитовыеивосьмибитовыекодыне обеспечиваютдостаточнокомпактногокодированиясимвольной информации. Более пригодными для этой цели являются 5 - битовые коды, например международный телеграфный код МГК-2. Перевод информации в код МГК-2 возможен с помощью программного перекодирования или с использованием специальных элементов на основе больших интегральных схем (БИС).Пропускнаяспособность каналовсвязи припередаче алфавитно- цифровой информации в коде МГК-2 повышается по сравнению с использованием восьмибитовых кодов почти на 40%.

Кодированиеданныхдвоичнымкодом

Для автоматизации работы с данными разных типов важно уметь представлять их в унифицированной форме. Для этого используется кодирование.

Кодирование– это представление данных одного типа через данные другого типа. Естественные языки – это не что иное, как системы кодирования понятий для выражения мыслей с помощью речи. В качестве другого примера можно привести азбуку Морзе для передачи телеграфных сигналов, морскую флажковую азбуку.

В вычислительной технике используется двоичное кодирование, основанное на представлении данных последовательностью из двух символов:0и1.Этизнакиназываютсядвоичнымицифрами,по- английски***digit***или сокращенно ***bit***(бит) .

Одним битом можно выразить два понятия: да или нет, черное или

белое,истинаилиложь,0или1.Есликоличествобитовувеличитьдодвух,то уже можно выразить четыре различных понятия:

Тремябитамиможнозакодировать8понятий:

001011100101110111.

Увеличивая на единицу количество разрядов, мы увеличиваем в два раза количество значений, которое может быть выражено в данной системе, то есть

N= 2^{m}гдеN– количество кодируемых значений;

m–количестводвоичныхразрядов.

Единицы измерения информации.

В компьютере бит является наименьшей возможной единицей информации. Объем информации, записанной двоичными знаками в памяти компьютера или на внешнем носителе информации, подсчитывается просто по количеству требуемых для такой записи двоичных символов.

Компьютер работает не с отдельными битами, а с восемью битами сразу. Восемь последовательных битов составляют байт.

Байт (byte) - является основной единицей информации. В одном байте можно закодировать значение одного символа из 256 возможных символов (256=2^8).

Более крупными единицами информации являются:

Килобайт (KB) = 1024 байт = 2^{10} байт;

Мегабайт (MB) = 1024 Кбайт = 2^{20} байт;

Гигабайт (GB) = 1024 Мбайт = 2^{30} байт;

Терабайт (TB) = 1024 Гбайт = 2^{40} байт.

Кодированиецелыхчисел

Для автоматизации работы с данными, относящимися к различным типам очень важно унифицировать их форму представления – для этого обычноиспользуетсяприёмкодирования,т.е.выражениеданныходноготипа через данные другого типа. Естественные человеческие языки – системы кодирования понятий для выражения мыслей посредством речи. К языкам близко примыкают азбуки – системы кодирования компонентов языка с помощью графических символов [5].

Своя системы существует и в вычислительной технике – она называется двоичным кодированием и основана на представлении данных последовательностью всего двух знаков: 0 и 1. Эти знаки называют двоичными цифрами, по-английски – binarydigit или сокращённо bit (бит). Одним битом могут быть выражены два понятия: 0 или 1 (да или нет, чёрное илибелое,истинаилиложьит.п.).Есликоличествобитовувеличитьдодвух, то уже можно выразить четыре различных понятия. Тремя битами можно закодировать восемь различных значений.

Для преобразования целого числа в двоичный код надо делить его пополам до тех пор, пока в остатке не образуется ноль или единица. Совокупность остатков от каждого деления, записанных справа налево, образует двоичный код десятичного числа.

Для представления целых чисел используется байт, имеющий восемь двоичных разрядов

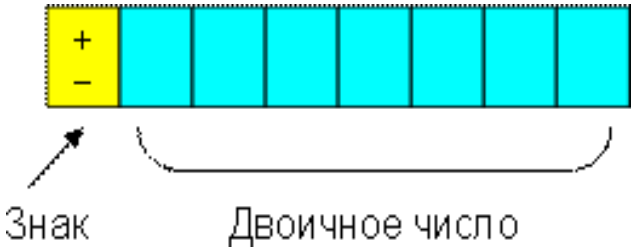


Рис.6.Представлениецелыхчисел.

Первый разряд используется для хранения знака числа. Обычно «+» кодируется нулём, а «–» – единицей. Диапазон представления целых чисел зависит от числа двоичных разрядов. С помощью одного байта могут быть представлены числа в диапазоне от –128 до +127. При использовании двух байтов могут быть представлены числа от –32 768 до +32 767

Кодированиевещественныхчисел

Существуют два способа представления вещественных чисел в памяти компьютера: с фиксированной точкой и с плавающей точкой.

При представлении вещественных чисел в форме с фиксированной точкой положение десятичной точки в машинном слове фиксировано.

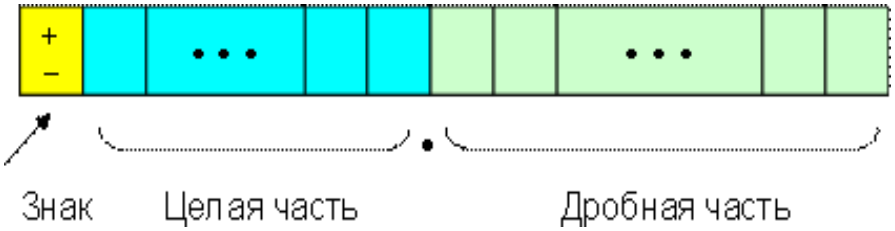


Рис.7.Вещественноечислосфиксированнойточкой.

Чащевсеготочкафиксируетсяпередпервымразрядомчисла.

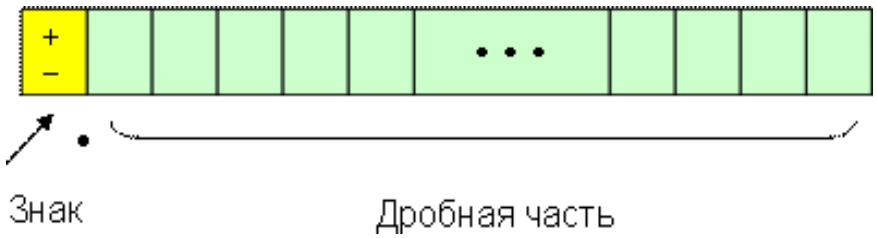


Рис.8.Вещественноечислосточкойпередпервымразрядом.

Целое число является частным случаем числа с фиксированной точкой, когда точка фиксирована после последнего разряда.

Вформесплавающейточкойвещественноечисло***x***представляетсяв виде

$x = Mx\ 2^p$

где$|\boldsymbol{M}|<\boldsymbol{1}$иназываетсямантиссой,***p***–целоечисло,называемое порядком.

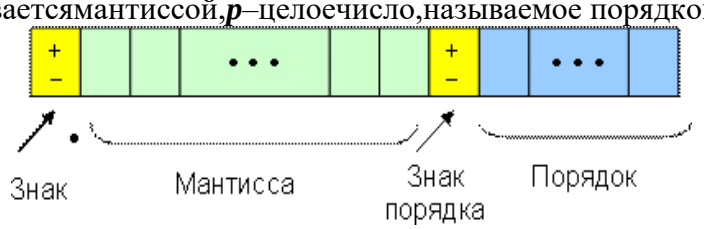


Рис. 9.Вещественноечислосплавающейточкой.

Количество позиций, отводимых для мантиссы, определяет точность представления чисел, а количество позиций, отводимых для порядка – диапазон представления чисел.

Обычно мантисса записывается в нормализованном виде, то есть так, чтобы отсутствовали незначащие нули в старших разрядах:

0.0011101ненормализованноепредставление,

0.1110100нормализованноепредставление.

При сложении чисел в форме с плавающей точкой в общем случае нельзя складывать их мантиссы. Если слагаемые имеют разные порядки, то одинаковые

разряды мантиссы будут на самом деле изображать разные разряды числа. Поэтому при сложении чисел необходимо предварительно выровнять их порядки, то есть числу с меньшим порядком приписатьпорядок второго числа.

Кодирование текстовых данных

Если каждому символу алфавита сопоставить определённое целое число, то с помощью двоичного кода можно кодировать текстовую информацию. Восьми двоичных разрядов достаточно для кодирования 256 различных символов. Это хватит, чтобы выразить различными комбинациями восьми битов все символы английского и русского языков, как строчные, так и прописные, а также знаки препинания, символы основных арифметических действий и некоторые общепринятые специальные символы Технически это выглядит очень просто, однако всегда существовали достаточно веские организационные сложности. В первые годы развития вычислительной техники они были связаны с отсутствием необходимых стандартов, а в настоящее время вызваны, наоборот, изобилием одновременно действующих и противоречивых стандартов. Для того чтобы весь мир одинаково кодировал текстовые данные, нужны единые таблицы кодирования, а это пока невозможно из-за противоречий между символами национальных алфавитов, а также противоречий корпоративного характера.

Для английского языка, захватившего де-факто нишу международного средства общения, противоречия уже сняты. Институт стандартизации США ввёл в действие систему кодирования ASCII (AmericanStandardCodeforInformationInterchange – стандартный код информационного обмена США). В системе ASCII закреплены две таблицы кодирования базовая и расширенная. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255 [8].

Первые 32 кода базовой таблицы, начиная с нулевого, отданы производителям аппаратных средств. В этой области размещаются управляющие коды, которым не соответствуют ни какие символы языков. Начиная с 32 по 127 код размещены коды символов английского алфавита, знаков препинания, арифметических действий и некоторых вспомогательных символов].

Кодировка символов русского языка, известная как кодировка Windows- 1251, была введена «извне» - компанией Microsoft, но, учитывая широкое распространение операционных систем и других продуктов этой компании в России, она глубоко закрепилась и нашла широкое распространение.

Другая распространённая кодировка носит название КОИ-8 (код обмена информацией, восьмизначный) – её происхождение относится к временам Действия Совета Экономической Взаимопомощи государств Восточной Европы. Сегодня кодировка КОИ – 8 имеет широкое распространение в компьютерных сетях на территории России и в российском секторе Интернета.Для хранения двоичного кода одного символа в этой кодировке выделен 1 байт = 8 бит.

Пусть K – количество символов в тексте, i – информационный «вес» одного символа. Тогда количество информации V, содержащейся в тексте, вычисляется по формуле:

$$V = K \times i$$

Международный стандарт, в котором предусмотрена кодировка символов русского языка, носит названия ISO (InternationalStandardOrganization – Международный институт стандартизации). На практике данная кодировка используется редко.

Кодирование графических данных

Если рассмотреть с помощью увеличительного стекла чёрно-белое графическое изображение, напечатанное в газете или книге, то можно увидеть, что оно состоит из мельчайших точек, образующих характерный узор, называемый растром. Поскольку линейные координаты и индивидуальные свойства каждой точки (яркость) можно

выразить с помощью целых чисел, то можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графическихданных.

Общепринятым на сегодняшний день считается представление чёрно-белых иллюстраций в виде комбинации точек с 256 градациями серого цвета, и, таким образом, для кодирования яркости любой точки обычно достаточно восьмиразрядного двоичного числа.

Для кодирования цветных графических изображений применяется принцип декомпозиции произвольного цвета на основные составляющие. В качестве таких составляющих используют три основные цвета: красный (Red), (Green) и синий (Blue). На практике считается, что любой цвет, видимый человеческим глазом, можно получить механического смешения этих трёх основных цветов. Такая система кодирования получила названия RGB по первым буквам основных цветов.

Режим представления цветной графики с использованием 24 двоичных разрядов называется полноцветным (TrueColor).

Каждому из основных цветов можно поставить в соответствие дополнительный цвет, т.е. цвет, дополняющий основной цвет до белого. Нетрудно заметить, что для любого из основных цветов дополнительным будет цвет, образованный суммой пары остальных основных цветов. Соответственно дополнительными цветами являются: голубой (Cyan), пурпурный (Magenta) и жёлтый (Yellow). Принцип декомпозиции произвольного цвета на составляющие компоненты можно применять не только для основных цветов, но и для дополнительных, т.е. любой цвет можно представить в виде суммы голубой, пурпурной и жёлтой составляющей. Такой метод кодирования цвета принят в полиграфии, но в полиграфии используется ещё и четвёртая краска – чёрная (Black). Поэтому данная система кодирования обозначается четырьмя буквами CMYK (чёрный цвет обозначается буквой К, потому, что буква В уже занята синим цветом), и для представления цветной графики в этой системе надо иметь 32 двоичных разряда. Такой режим также называется полноцветным.

Если уменьшить количество двоичных разрядов, используемых для кодирования цвета каждой точки, то можно сократить объём данных, но приэтом диапазон кодируемых цветов заметно сокращается. Кодирование цветной графики 16-разрядными двоичными числами называется режимом HighColor.

При кодировании информации о цвете с помощью восьми бит данных можно передать только 256 оттенков. Такой метод кодирования цвета называется индексным.

Любое изображение на мониторе компьютера представляет собой набор светящихся точек (пикселей). Под пикселем понимается минимальный графический объект экрана, внутри которого свойства не меняются. Качество картинки, которую мы видим на мониторе, напрямую зависит от разрешения. Каждому цвету пикселя ставится в соответствие свое число, в зависимости от глубины цвета. Количество байт для хранения информации о цвете одного пикселя может быть различным:

- Для черно-белой картинки – 1 пк = 1 бит;
- Для 16-цветной картинки – 1 пк = 4 бита;
- Для 256-цветной картинки – 1 пк = 1 байт.

Объем видеопамяти (V), необходимый для формирования графического изображения на экране:

$$V = l \times h \times k,$$

где l – число точек по горизонтали, h – число точек по вертикали, k – глубина цвета (бит).

Кодирование звуковой информации

Приёмы и методы работы со звуковой информацией пришли в вычислительную технику наиболее поздно. К тому же, в отличие от числовых, текстовых и графических данных, у звукозаписей не было столь же длительной и проверенной истории

кодирования. В итоге методы кодирования звуковой информации двоичным кодом далеки от стандартизации. Множество отдельных компаний разработали свои корпоративные стандарты, но среди них можно выделить два основных направления.

Метод FM (FrequencyModulation) основан та том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый из которых представляет собой правильную синусоиду, а, следовательно, может быть описан числовыми параметрами, т.е. кодом. В природе звуковые сигналы имеют непрерывный спектр, т.е. являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальный устройства – аналогово-цифровые преобразователи (АЦП). Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифро-аналоговые преобразователи (ЦАП). При таких преобразованиях неизбежны потери информации, связанные с методом кодирования, поэтому качество звукозаписи обычно получается не вполне удовлетворительным и соответствует качеству звучания простейших электромузыкальныхинструментов с окрасом характерным для электронной музыки. В то же время данный метод копирования обеспечивает весьма компактный код, поэтому он нашёл применение ещё в те годы, когда ресурсы средств вычислительной техники были явно недостаточны.

Метод таблично волнового (Wave-Table) синтеза лучше соответствует современному уровню развития техники. В заранее подготовленных таблицах хранятся образцы звуков для множества различных музыкальных инструментах. В технике такие образцы называют сэмплами. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звучания. Поскольку в качестве образцов исполняются реальные звуки, то его качество получается очень высоким и приближается к качеству звучания реальных музыкальных инструментов.

Аудиоадаптер – специальное устройство, предназначенное для преобразования электрических колебаний звуковой частоты в числовой двоичный код при вводе звука и для обратного преобразования при воспроизведении звука.

Качество компьютерного звука определяется характеристиками аудиоадаптера: частотой дискретизации и разрядностью.

Разрядность регистра (k) – количество бит в регистре аудиоадаптера. Разрядность определяет точность измерения звукового сигнала.

Частота дискретизации (f) – количество измерений входного сигнала за единицу времени. Частота измеряется в герцах (Гц).

Объём аудиофайла (V):

$$V=N \times f \times k,$$

где N – длительность звучания, k – разрядность регистра, f – частота дискретизации.

Скорость передачи информации

Скорость передачи информации - скорость, с которой передается или принимается информация в двоичной форме. Обычно скорость передачи данных измеряется количеством бит, переданных в одну секунду.

$$\upsilon = \frac{V}{t}$$

4.Выбор способа представления информации в соответствии с поставленной задачей.

Для обмена информацией с другими людьми человек использует естественные языки (русский, английский, китайский и др.), то есть информация представляется с помощью естественных языков. В основе языка лежит алфавит, то есть набор символов (знаков), которые человек различает по их начертанию. В основе русского языка лежит кириллица, содержащая 33 знака, английский язык использует латиницу (26 знаков), китайский язык использует алфавит из десятков тысяч знаков (иероглифов).

Последовательности символов алфавита в соответствии с правилами грамматики образуют основные объекты языка — слова. Правила, согласно которым образуются предложения из слов данного языка, называются синтаксисом. Необходимо отметить, что в естественных языках грамматика и синтаксис языка формулируются с помощью большого количества правил, из которых существуют исключения, так как такиеправила складывались исторически.

Наряду с естественными языками были разработаны формальные языки (системы счисления, язык алгебры, языки программирования и др.). Основное отличие формальных языков от естественных состоит в наличии строгих правил грамматики исинтаксиса.

Например, системы счисления можно рассматривать как формальные языки, имеющиеалфавит (цифры) и позволяющие не только именовать и записывать объекты (числа), но и выполнять над ними арифметические операции по строго определенным правилам.

Некоторые языки используют в качестве знаков не буквы и цифры, а другие символы, например, химические формулы, ноты, изображения элементов электрических или логических схем, дорожные знаки, точки и тире (код азбуки Морзе) и др.

Знаки могут иметь различную физическую природу. Например, для представления информации с использованием языка в письменной форме используются знаки, которые являются изображениями на бумаге или других носителях, в устной речи в качестве знаков языка используются различные звуки (фонемы), а при обработке текста накомпьютере знаки представляются в форме последовательностей электрическихимпульсов (компьютерных кодов).

5. Представление информации в различных системах счисления

Позиционные и непозиционные системы счисления.

Понятие числа является фундаментальным как для математики, так и для информатики. С числами связано еще одно важное понятие — система счисления.

Система счисления — это способ изображения чисел и соответствующие ему правила действий над числами.

Разнообразные системы счисления, которые существовали раньше и которые используются в наше время, можно разделить нанепозиционные и позиционные.

В древние времена, когда люди начали считать, появилась потребность в записи чисел. Первоначально количество предметов отображали равным количеством каких-нибудь значков: насечек, черточек, точек.

Изучение археологами «записок» времен палеолита на кости, камне, дереве показало, что люди стремились группировать отметки по 3, 5, 7, 10 штук. Такая группировка облегчала счет. Люди учились считать не только единицами, но и тройками, пятерками и пр. Поскольку первым вычислительным инструментом учеловека были пальцы, поэтому и счет чаще всего вели группами по 5 или по 10 предметов.

В дальнейшем свое название получили десяток десятков (сотня), десяток сотен (тысяча) и так далее. Такие узловые числа для удобства записи стали обозначать особыми значками — цифрами. Если при подсчете предметов их оказывалось 2 сотни, 5 десятков и еще 4 предмета, то при записи этой величины дважды повторяли знак сотни, пять раз — знак десятков и четыре раза знак единицы.

В таких системах счисления от положения знака в записи числа не зависит величина, которую он обозначает; поэтому они называются непозиционными системами счисления.

Непозиционными системами пользовались древние египтяне, греки, римляне и некоторые другие народы древности.

На Руси вплоть до XVIII века, использовалась непозиционная система славянских цифр. Буквы кириллицы (славянского алфавита) имели цифровое значение, если над ними ставился специальный знак ~ титло. Например Ã — 1, Д — 4, Р — 100.

Непозиционные системы счисления были более или менее пригодны для выполнения сложения и вычитания, но совсем не удобны при умножении и делении.
Идея позиционной системы счисления впервые возникла в древнем Вавилоне.

В позиционных системах счисления величина, обозначаемая цифрой в записи числа, зависит от ее позиции.

Количество используемых цифр называется основанием позиционной системы счисления.

Система счисления, применяемая в современной математике, является позиционной десятичной системой. Ее основание равно десяти, так как запись любых чисел производится с помощью десяти цифр:

0,1, 2, 3, 4, 5, 6, 7, 8, 9.

Хотя десятичную систему принято называть арабской, но зародилась она в Индии, в V веке. В Европе об этой системе узнали в XII веке из арабских научных трактатов, которые были переведены на латынь. Этим и объясняется название «арабские цифры». Однако широкое распространение в науке и в обиходе десятичная позиционная система получила только в XVIвеке. Эта система позволяет легко выполнять любые арифметические вычисления, записывать числа любой величины. Распространение арабской системы дало мощный толчок развитию математики.

С позиционной десятичной системой счисления вы знакомы с раннего детства, только, возможно, не знали, что она так называется.

Позиционный тип этой системы легко понять на примере любого многозначного числа. Например, в числе 333 первая тройка означает три сотни, вторая — три десятка, третья — три единицы. Одна и та же цифра в зависимости от позиции в записи числа обозначает разные величины.

$$333 = 3x100 + 3x10 + 3.$$

Еще пример:

$$32478 = 3 \times 10000 + 2 \times 1000 + 4 \times 100 + 7 \times 10 + 8 =$$
$$= 3 \times 10^4 + 2 \times 10^3 + 4 \times 10^2 + 7 \times 10^1 + 8 \times 10^0.$$

Отсюда видно, что всякое десятичное число можно представить как сумму произведений составляющих его цифр на соответствующие степени десятки. То же самое относится и к десятичным дробям.

$$26{,}387 = 2 \times 10^1 + 6 \times 10^0 + 3 \times 10^{-1} + 8 \times 10^{-2} + 7 \times 10^{-3}$$

Системы счисления, используемые в ЭВМ

Очевидно, число «десять» — не единственно возможное основание позиционной системы. Известный русский математик Н.Н.Лузин так выразился по этому поводу: «Преимущества десятичной системы не математические, а зоологические. Если бы у нас на руках было не десять пальцев, а восемь, то человечество пользовалось бы восьмеричной системой».

За основание позиционной системы счисления можно принять любое натуральное число большее 1. Упомянутая выше вавилонская система имела основание 60. Следы этой системы сохранились до наших дней в порядке счета единиц времени (1 час = 60 мин, 1 мин = 60 с).

Для записи чисел в позиционной системе с основанием n нужно иметь ***алфавит***изn цифр. Обычно для этого при n<10 используют n первых арабских цифр, а при n>10 к десяти арабским цифрам добавляют буквы.

Вот примеры алфавитов нескольких систем:

Табл.1.

Основание	**Система**	**Алфавит**
n=2	двоичная	01
n=3	троичная	01 2
n=8	восьмеричная	01234567
n=16	шестнадцатеричная	0123456789ABCDEF

Основание системы, к которой относится число, обозначается подстрочным индексом к этому числу.

101101_2, 3671_8, $3B8F_{16}$.

Принципы архитектуры ЭВМ были сформулированы Джоном фон Нейманом в 1946 году. Им долгие годы следовали конструкторы ЭВМ. Многие из этих принципов сохранились и в архитектуре современных компьютеров.

Один из этих принципов Неймана:

ЭВМ выполняет арифметические расчеты в двоичной системе счисления.

Компьютеры используют двоичную систему потому, что она имеет ряд преимуществ перед другими системами:

- для ее реализации нужны технические устройства с двумя устойчивыми состояниями (есть ток — нет тока, намагничен — не намагничен и т.п.), а не, например, с десятью, — как в десятичной;
- представление информации посредством только двух состояний надежно и помехоустойчиво;
- возможно применение аппарата булевой алгебры для выполнения логических преобразований информации;
- двоичная арифметика намного проще десятичной.

Недостаток двоичной системы — быстрый рост числа разрядов, необходимых для записи чисел.

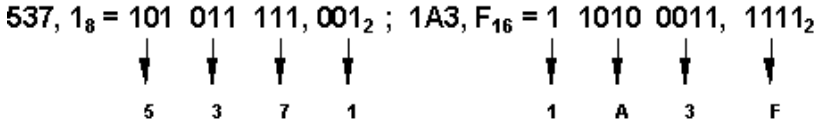
Гораздо проще сконструировать процессор, который работает в двоичной системе счисления, чем работающий в десятичной. Двоичная система, удобная для компьютеров, для человека неудобна из-за ее громоздкости и непривычной записи.

Перевод чисел из десятичной системы в двоичную и наоборот выполняет машина. Однако, чтобы профессионально использовать компьютер, следует научиться понимать слово машины. Для этого и разработаны восьмеричная и шестнадцатеричная системы.

Числа в этих системах читаются почти так же легко, как десятичные, требуют соответственно в три (восьмеричная) и в четыре (шестнадцатеричная) раза меньше разрядов, чем в двоичной системе (ведь числа 8 и 16 – соответственно, третья и четвертая степени числа 2).

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

Например:



Чтобы перевести число из двоичной системы в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на триады (для восьмеричной) или тетрады (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной (шестнадцатеричной) цифрой.

Например,

$$10101001{,}10111_2 = \underset{2}{10}\ \underset{5}{101}\ \underset{1}{001},\ \underset{5}{101}\ \underset{6}{110}_2 = 251{,}56_8$$

$$10101001{,}10111_2 = \underset{A}{1010}\ \underset{9}{1001},\ \underset{B}{1011}\ \underset{8}{1000}_2 = A9{,}B8_{16}.$$

Шестнадцатеричная система счисления используется для компактного представления (на бумаге или на экране) двоичной информации, хранимой в памяти ЭВМ.

Алгоритмы перевода чисел из одной системы счисления в другую.

Мы настолько привыкли к десятичному счету, что число в любой другой системе мало что нам говорит о соответствующем ему количестве. Например, что за величина 112_3? Чтобы понять «много это или мало», нужно перевести его в десятичную систему. Сделать это довольно просто.

Число 112_3 содержит в себе 2 единицы, 1 тройку и 1 девятку. Как и в десятичной системе, число можно представить в виде суммы произведений составляющих его цифр на соответствующие степени основания системы (в нашем примере — тройки).

$$112_3 = 1x3^2 + 1x3^1 + 2x3^\circ = 9 + 3 + 2 = 14_{10}$$

Следовательно, $112_3 = 14_{10}$

Переведем двоичное число 101101_2 в десятичную систему счисления. Принцип тот же. Теперь в сумму надо подставлять степени двойки:

$$101101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1x2^\circ = 32 + 8 + 4 + 1 = 45_{10}.$$

И еще один пример — с шестнадцатеричным числом:

$$15FC_{16} = 1 \times 16^3 + 5 \times 16^2 + 15 \times 16^1 + 12 = 4096 + 1280 + 240 + 12 = 5628$$

Аналогично переводятся дробные числа.

$$101{,}11_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^\circ + 1 \times 2^{-1} + 1 \times 2^{-2} =$$
$$= 4 + 1 + 1/2 + 1/4 = 5 + 0{,}5 + 0{,}25 = 5{,}75_{10}.$$

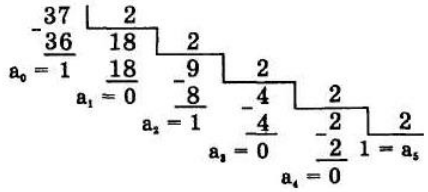
А как произвести обратный перевод из десятичной системы в недесятичную ($n \neq 10$)? Для этого нужно суметь разложить десятичное число на слагаемые, содержащие степени n. Например, при n = 2 (двоичная система):

$$15_{10} = 8 + 4 + 2 + 1 = 1x2^3 + 1x2^2 + 1x2^1 + 1 = 1111_2$$

Эта задача уже посложнее, чем перевод в десятичную систему. Попробуйте, например, таким образом перевести в двоичную систему число 157. Конечно можно, но трудно!

Однако существует процедура, позволяющая легко выполнить такой перевод. Она состоит в том, что данное десятичное число делится с остатком на основание системы. Полученный остаток — это младший разряд искомого числа, а полученное частное снова делится с остатком, который равен второй справа цифре и т.д. Так продолжается до тех пор, пока частное не станет меньше делителя (основания системы). Это частное — старшая цифра искомого числа.

Продемонстрируем этот метод на примере перевода числа 37_{10} в двоичную систему. Здесь для обозначения цифр в записи числа используется символика: $a_5a_4a_3a_2a_1a_0$.



Отсюда: 37_{10} - 100101_2

Перевод п другую систему счисления производится путем последов ание системы с выделением целой части произведений. Однако мы остановимся лишь на целых числах.

Двоичная арифметика.

Вам хорошо знакомы правила выполнения арифметических операций с многозначными десятичными числами. В младших классах школы вы учились складывать, вычитать, умножать «столбиком» и делить «уголком». В конечном счете для выполнения вычислений нужно уметь складывать и умножать однозначные числа. Таблицу умножения десятичных чисел многие первоклассники заучивают долго и с большим трудом. Но вот если бы в школе изучали не десятичную, а двоичную арифметику, проблем бы не было ни у кого и все ученики были бы отличниками! Сейчас вы убедитесь в том, что двоичная арифметика, действительно, очень проста.

С двоичной системой счисления вы уже знакомы. В ней всего две цифры: 0 и 1. Вот все варианты их сложения:

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 1 = 10.$$

Вам уже должно быть понятно, что $10_2 = 2_{10}$ (напомним, что нижний индекс обозначает основание системы счисления и всегда записывается в десятичной системе). Ряд двоичных натуральных чисел легко записать, получая каждое следующее число путем прибавления единицы к предыдущему.

Десятичные числа от 1 до 16 и равные им двоичные числа

Табл.2.

«10»	«2»	«10»	«2»	«10»	«2»	«10»	«2»
1	1	5	101	9	1001	13	1101
2	10	6	110	10	1010	14	1110
3	11	7	111	11	1011	15	1111
4	100	8	1000	12	1100	16	10000

Из таблицы 1 видно, как быстро нарастает количество цифр в двоичных числах. Но этот недостаток двоичной системы компенсируется простотой арифметики. Вот пример сложения столбиком двух многозначных двоичных чисел:

```
  1 0 1 1 0 1 1 1 0  1
+ 1 1 1 0 1 0 1 1 0
-------------------
1 0 0 1 0 1 1 0 0 1 1
```

Двоичная таблица умножения:

$$0 \times 0 = 0, \quad 1 \times 0 = 0, \quad 1 \times 1 = 1.$$

Пример:

```
   111
 x 11
-----
   111
+ 111
-----
 10101
```

6.Информационная безопасность ИС

Информационная безопасность информационной системы – защищенность информации, обрабатываемой компьютерной системой, от внутренних (внутрисистемных) или внешних угроз, то есть состояние защищенности информационных ресурсов системы, обеспечивающее устойчивое функционирование, *целостность* и эволюцию системы. К защищаемой информации (информационным ресурсам системы) относятся электронные документы и спецификации, *программное обеспечение*, структуры и *базы данных* и др.

Оценка безопасности компьютерных систем базируется на различных *классах защиты* систем:

- *класс* систем минимальной защищенности (*класс* D);
- *класс* систем с защитой по усмотрению пользователя (*класс* C);
- *класс* систем с обязательной защитой (*класс* B);
- *класс* систем с гарантированной защитой (*класс* A).

Эти *классы* имеют и подклассы, но мы их не будем здесь детализировать.

Основными типами средств воздействия на *компьютерные сети* и системы являются компьютерные *вирусы*, *логические бомбы* и мины (закладки, жучки), внедрение в информационный обмен.

Пример. Многократно разославшая свой *код* в 2000 году вирусная *программа* в Интернете могла при открытии приложения к тексту письма с интригующим заголовком (*ILoveYou – ЯТебяЛюблю*) рассылать свой *код по* всем адресам, зафиксированным в адресной книге данного получателя *вируса*, что приводило к веерному размножению *вируса по* Интернету, ибо *адресная книга* каждого пользователя может содержать десятки и сотни адресов.

Компьютерный вирус – специальная *программа*, которая составлена кем-то со злым умыслом или для демонстрации честолюбивых, в плохом смысле, интересов, способная к воспроизводству своего *кода* и к переходу от программы к программе (инфицирование). *Вирус* подобен инфекции, проникающей в кровяные тельца и путешествующей *по* всему организму человека. Перехватывая управление (прерывания), *вирус* подключается к работающей программе или к другим программам и затем дает команду компьютеру для записи зараженной версии программы, а затем возвращает управление программе как ни в чем не бывало. Далее или сразу же этот *вирус* может заработать (перехватив управление от программы).

По мере появления новых компьютерных *вирусов* разработчики антивирусных программ пишут вакцину против нее – так называемую антивирусную программу, которая, анализируя файлы, может распознать в них скрытый *код вируса* и либо удалить этот *код* (вылечить), либо удалить зараженный *файл*. Базы антивирусных программ обновляются часто.

Пример. Одну из самых популярных антивирусных программ *AIDSTEST автор* (Д. Лозинский) обновляет иногда дважды в неделю. Известная антивирусная *программа AVP* лаборатории Касперского содержит в своей базе данные о нескольких десятках тысяч *вирусах*, вылечиваемых программой.

Вирусы бывают следующих основных видов:

- **загрузочные** – заражающие стартовые секторы дисков, где находится самая важная информация о структуре и файлах диска (служебные области диска, так называемые boot–сектора);
- **аппаратно-вредные** – приводящие к нарушению работы, а то и вовсе к разрушению аппаратуры, например, к резонансному воздействию на винчестер, к "пробою" точки на экране дисплея;
- **программные** – заражающие исполняемые файлы (например, exe-файлы с непосредственно запускаемыми программами);
- **полиморфные** – которые претерпевают изменения (мутации) от заражения к заражению, от носителя к носителю;

- **стелс-вирусы** – маскирующиеся, незаметные (не определяющие себя ни размером, ни прямым действием);
- **макровирусы** – заражающие документы и шаблоны текстовых редакторов, используемые при их создании;
- **многоцелевые вирусы**.

Особенно опасны *вирусы* в компьютерных сетях, так как они могут парализовать работу всей сети.

Вирусы могут проникать в *сеть*, например:

- с внешних носителей информации (из копируемых файлов, с дискет);
- через электронную почту (из присоединенных к письму файлов);
- через Интернет (из загружаемых файлов).

Существуют различные методы и пакеты программ для борьбы с *вирусами* (антивирусные пакеты).

При выборе антивирусных средств необходимо придерживаться следующих простых принципов (аналогичных противогриппозной профилактике):

- если используются в системе различные платформы, операционные среды, то антивирусный пакет должен поддерживать все эти платформы;
- антивирусный пакет должен быть простым и понятным, дружественным в использовании, позволяющим выбирать опции однозначно и определенно на каждом шаге работы, иметь развитую систему понятных и информативных подсказок;
- антивирусный пакет должен обнаруживать – скажем, с помощью различных эвристических процедур – новые неизвестные *вирусы* и иметь пополняемую и обновляемую регулярно базу данных о *вирусах* ;
- антивирусный пакет должен быть лицензионным, от надежного известного поставщика и производителя, который регулярно обновляет базу данных, а сам поставщик должен иметь свой антивирусный центр – сервер, откуда можно получить необходимую срочную помощь, информацию.

Вопросы:

1. Чем отличается непрерывный сигнал от дискретного?
2. Что такое частота дискретизации и на что она влияет?
3. Объясните понятие информации?
4. Перечислите основные формы представления информации?
5. В чем отличие позиционной и непозиционной систем счисления?
6. Что такое система счисления?
7. Почему в вычислительной технике взята за основу двоичная система счисления?
8. Перечислить единицы измерения информации, которые используются при записи данных в память компьютера?
9. Охарактеризуйте универсальность дискретного представления информации?
10. В чем заключается информационная безопасность ИС?

Лекция №3.
Тема: «Программное обеспечение. Системы программирования»

План:

1. Программное обеспечение вычислительной техники
2. Классификация программного обеспечения.
3. Системы программирования: основные понятия

1.Программное обеспечение вычислительной техники

В основу работы компьютеров положен программный принцип управления, состоящий в том, что компьютер выполняет действия по заранее заданной программе. Этот принцип обеспечивает универсальность использования компьютера: в определенный момент времени решается задача соответственно выбранной программе. После ее завершения в память загружается другая программа и т. д.

Для нормального решения задач на компьютере нужно, чтобы программа была отлажена, не требовала доработок и имела соответствующую документацию. Поэтому относительно работы на компьютере часто используют термин **программное обеспечение (ПО, software)**, под которым понимают совокупность программ, процедур, правил и, касающихся функционирования программной системы для решения поставленной задачи.

Поскольку безПО функционирование ПК невозможно в принципе, оно является неотъемлемой составной частью любого ПК и поставляется вместе с его аппаратной частью (hardware).

Программа– полное и точное описание последовательности действий (инструкций) компьютера по обработке информации, написанное на языке, понятном компьютеру.

Программное обеспечение (ПО) – совокупность специальных программ, облегчающих процесс подготовки задач к выполнению на ЭВМ и организующих прохождение их через машину, а также процедур, описаний, инструкций и правил вместе со всей связанной с этими компонентами документацией, используемых при эксплуатации вычислительной системы.

Назначение ПО:

- обеспечение работоспособности компьютера;
- облегчение взаимодействия пользователя с компьютером;
- сокращение цикла от постановки задачи до получения результата;
- повышение эффективности использования ресурсов компьютера.

Программное обеспечение позволяет:

- усовершенствовать организацию работы вычислительной системы с целью максимального использования ее возможностей;
- повысить производительность и качество труда пользователя;
- адаптировать программы пользователя к ресурсам конкретной вычислительной системы;расширить ПО вычислительной системы.

Повышение производительности и качества труда пользователей при использовании программного обеспечения происходит за счёт автоматизации процедур расчётного и оформительского характера, реализуемых с помощью разнообразных средств программирования (алгоритмических языков, пакетов прикладных программ) и удобных средств ввода и вывода информации.

Программное обеспечение в настоящее время составляет сотни тысяч программ, которые предназначены для обработки самой разнообразной информации с самыми различными целями. В состав программного обеспечения включают программы и необходимые для их функционирования данные. Все программы состоят из совокупности операторов и данных, описанных на некотором языке программирования и создаются с помощью инструментальных программ. Программы хранятся в файлах либо в виде текста программы на определенном языке программирования, либо в виде исполняемой программы. В первом случае для выполнения программы ее необходимо наличие транслятора или соответствующей системы программирования, во втором случае для выполнения программы достаточно просто запустить ее.

2.Классификация программного обеспечения

Программное обеспечение (ПО) - это совокупность всех программ и соответствующей документации, обеспечивающая использование ЭВМ в интересах каждого ее пользователя.

Различают системное ПО, прикладное ПО и системы программирования. Схематически программное обеспечение можно представить так:

Рис. 10.

Системное ПО – это совокупность программ для обеспечения работы компьютера. Системное ПО подразделяется на **базовое** и **сервисное**. Системные программы предназначены для управления работой вычислительной системы, выполняют различные вспомогательные функции (копирования, выдачи справок, тестирования, форматирования и т. д.).

Базовое ПО включает в себя:

- операционные системы;
- оболочки;
- сетевые операционные системы.

Сервисное ПО включает в себя программы (утилиты):

- диагностики;
- антивирусные;
- обслуживания носителей;
- архивирования;
- обслуживания сети.

Прикладное ПО – это комплекс программ для решения задач определённого класса конкретной предметной области.Прикладное ПО работает только при наличии системного ПО.

Прикладные программы называют приложениями. Они включает в себя:

- текстовые процессоры;
- табличные процессоры;
- базы данных;
- интегрированные пакеты;
- системы иллюстративной и деловой графики (графические процессоры);
- экспертные системы;
- обучающие программы;
- программы математических расчетов, моделирования и анализа;
- игры;
- коммуникационные программы.

Особую группу составляют системы программирования (инструментальные системы), которые являются частью системного ПО, но носят прикладной характер.

Системы программирования – это совокупность программ для разработки, отладки и внедрения новых программных продуктов. Системы программирования обычно содержат:

- трансляторы;
- среду разработки программ;
- библиотеки справочных программ (функций, процедур);
- отладчики;
- редакторы связей и др.

3.Системы программирования: основные понятия

Система программирования — система, образуемая языком программирования, компилятором или интерпретатором программ, представленных на этом языке, соответствующей документацией, а также вспомогательными средствами для подготовки программ к форме, пригодной для выполнения.

Этапы подготовки программы

При разработке программ, а тем более — сложных, используется принцип модульности, разбиения сложной программы на составные части, каждая из которых может подготавливаться отдельно. Модульность является основным инструментом структурирования программного изделия, облегчающим его разработку, отладку и сопровождение.

Программный модуль — программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

При выборе модульной структуры должны учитываться следующие основные соображения:

- **Функциональность** — модуль должен выполнять законченную функцию
- **Несвязность** — модуль должен иметь минимум связей с другими модулями, связь через глобальные переменные и области памяти нежелательна
- **Специфицируемость** — входные и выходные параметры модуля должны четко формулироваться

Программа пишется в виде исходного модуля.

Исходный модуль — программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

Первым (не для всех языков программирования обязательным) этапом подготовки программы является обработка ее Макропроцессором (или Препроцессором). Макропроцессор обрабатывает текст программы и на выходе его получается новая редакция текста. В большинстве систем программирования Макропроцессор совмещен с транслятором, и для программиста его работа и промежуточный ИМ «не видны».

Следует иметь в виду, что Макропроцессор выполняет обработку текста, это означает, с одной стороны, что он «не понимает» операторов языка программирования и «не знает» переменных программы, с другой, что все операторы и переменные Макроязыка (тех выражений в программе, которые адресованы Макропроцессору) в промежуточном ИМ уже отсутствуют и для дальнейших этапов обработки «не видны».

Так, если Макропроцессор заменил в программе некоторый текст А на текст В, то транслятор уже видит только текст В, и не знает, был этот текст написан программистом «своей рукой» или подставлен Макропроцессором.

Следующим этапом является трансляция.

Трансляция — преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

Как правило, выходным языком транслятора является машинный язык целевой вычислительной системы. (Целевая ВС — та ВС, на которой программа будет выполняться.)

Машинный язык — язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Трансляторы — общее название для программ, осуществляющих трансляцию. Они подразделяются на Ассемблеры и Компиляторы — в зависимости от исходного языка программы, которую они обрабатывают. Ассемблеры работают с Автокодами или языками Ассемблера, Компиляторы — с языками высокого уровня.

Автокод — символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.

Язык Ассемблера — язык программирования, который представляет собой символьную форму машинного языка с рядом возможностей, характерных для языка высокого уровня (обычно включает в себя макросредства).

Язык высокого уровня — язык программирования, понятия и структура которого удобны для восприятия человеком.

Объектный модуль — программный модуль, получаемый в результате трансляции исходного модуля.

Поскольку результатом трансляции является модуль на языке, близком к машинному, в нем уже не остается признаков того, на каком исходном языке был написан программный модуль. Это создает принципиальную возможность создавать программы из модулей, написанных на разных языках. Специфика исходного языка, однако, может сказываться на физическом представлении базовых типов данных, способах обращения к процедурам/функциям и т.п. Для совместимости разноязыковых модулей должны выдерживаться общие соглашения. Большая часть объектного модуля — команды и данные машинного языка именно в той форме, в какой они будут существовать во время выполнения программы. Однако, программа в общем случае состоит из многих модулей. Поскольку транслятор обрабатывает только один конкретный модуль, он не может должным образом обработать те части этого модуля, в которых запрограммированы обращения к данным или процедурам, определенным в другом модуле. Такие обращения называются внешними ссылками. Те места в объектном модуле, где содержатся внешние ссылки, транслируются в некоторую промежуточную форму, подлежащую дальнейшей обработке. Говорят, что объектный модуль представляет собой программу на машинном языке с неразрешенными внешними ссылками. Разрешение внешних ссылок выполняется на следующем этапе подготовки, который обеспечивается Редактором Связей (Компоновщиком). Редактор Связей соединяет вместе все объектные модули, входящие в программу. Поскольку Редактор Связей «видит» уже все компоненты программы, он имеет возможность обработать те места в объектных модулях, которые содержат внешние ссылки. Результатом работы Редактора Связей является загрузочный модуль.

Загрузочный модуль — программный модуль, представленный в форме, пригодной для загрузки в оперативную память для выполнения.

Загрузочный модуль сохраняется в виде файла на внешней памяти. Для выполнения программа должна быть перенесена (загружена) в оперативную память. Иногда при этом требуется некоторая дополнительная обработка (например, настройка адресов в программе на ту область оперативной памяти, в которую программа загрузилась). Эта функция выполняется Загрузчиком, который обычно входит в состав операционной системы. Возможен также вариант, в котором редактирование связей выполняется при каждом запуске программы на выполнение и совмещается с загрузкой.

Это делает Связывающий Загрузчик. Вариант связывания при запуске более расходный, т.к. затраты на связывание тиражируются при каждом запуске. Но он обеспечивает:

- большую гибкость в сопровождении, так как позволяет менять отдельные объектные модули программы, не меняя остальных модулей;
- экономию внешней памяти, т.к. объектные модули, используемые во многих программах не копируются в каждый загрузочный модуль, а хранятся в одном экземпляре.

Вариант интерпретации подразумевает прямое исполнение исходного модуля.

Интерпретация — реализация смысла некоторого синтаксически законченного текста, представленного на конкретном языке.

Интерпретатор читает из исходного модуля очередное предложение программы, переводит его в машинный язык и выполняет. Все затраты на подготовку тиражируются при каждом выполнении, следовательно, интерпретируемая программа принципиально менее эффективна, чем транслируемая. Однако, интерпретация обеспечивает удобство разработки, гибкость в сопровождении и переносимость.

Не обязательно подготовка программы должна вестись на той же вычислительной системе и в той же операционной среде, в которых программа будет выполняться. Системы, обеспечивающие подготовку программ в среде, отличной от целевой называются кросс-системами. В кросс-системе может выполняться вся подготовка или ее отдельные этапы:

- Макрообработка и трансляция
- Редактирование связей
- Отладка

Типовое применение кросс-систем — для тех случаев, когда целевая вычислительная среда просто не имеет ресурсов, необходимых для подготовки программ, например, встроенные системы. Программные средства, обеспечивающие отладку программы на целевой системе можно также рассматривать как частный случай кросс-системы.

Вопросы:

1. Что вы понимаете под программным обеспечением?
2. На какие части делится программное обеспечение?
3. В чем отличие базового ПО от сервисного ПО?
4. Дайте определение системного ПО?
5. Какие этапы подготовки программы вы знаете?
6. В чем заключается основное назначение интерпретатора?

Лекция №4.

Тема: «Оформление документов с помощью программы MicrosoftWord»

План:

1. Автоматизированные средства и технологии организации текста.
2. Функции MicrosoftWord.
3. Применение MicrosoftWord.

1.Автоматизированные средства и технологии организации текста.

Редакторы текстов программ рассчитаны на редактирование программ на том или ином языке программирования. Редакторы текста, и рассчитаны на тексты программ, и выполняют следующие функции:

- диалоговый просмотр текста;
- редактирование строк программы;
- копирование и перенос блоков текста из одного места в другое;
- копирование одной программы или её части в указанное место другой программы;
- контекстный поиск и замену подстрок текста;
- автоматический поиск строки, содержащей ошибку;
- распечатку программы или её необходимой её части.

Редакторы документов – программы для обработки документов, ориентированные на работу с текстами, имеющие структуру документа, т. Е. состоящими из разделов, страниц, абзацев, предложений, слов и т.д. Следовательно, редакторы для обработки документов обеспечивают такие функции, как:

- возможность использования различных шрифтов символов;
- задание произвольных межстрочных промежутков;
- автоматический перенос слов на следующую строку;
- автоматическую нумерацию страниц;
- обработку и нумерацию строк;
- печать верхних и нижних заголовков страниц (колонтитулов);
- выравнивание краев абзаца;
- набор текста в несколько столбцов;
- создание таблиц и построение диаграмм;
- проверку правописания и подбор символов.

Редакторы текстов программ, можно использовать для создания и корректировки небольших документов. Однако при необходимости серьезной работы с документами лучше использовать редакторы, ориентированные на работу с документами.

Современные текстовые процессоры предоставляют пользователю широкие возможности по подготовке документов. Это и функции редактирования, допускающие возможность любого изменения, вставки, замены, копирования и перемещения фрагментов в рамках одного документа и между различными документами, контекстного поиска, функции форматирования символов, абзацев, страниц, разделов документа, верстки, проверки грамматики и орфографии, использования наряду с простыми текстовыми элементами списков, таблиц, рисунков, графиков и диаграмм.

Значительное сокращение времени подготовки документов обеспечивают такие средства автоматизации набора текста, как автотекст и автозамена, использование форм, шаблонов и мастеров типовых документов.

Наличие внешней памяти компьютера обеспечивает удобное длительное хранение подготовленных ранее документов, быстрый доступ к ним в любое время.

Существенно упрощают процедуру ввода данных сканеры и голосовые устройства. Существующие системы распознавания текстов, принимаемых со сканера, включают функцию экспорта документа в текстовые редакторы.

Широкий спектр печатающих устройств в сочетании с функциями подготовки документа к печати, предварительного просмотра, обеспечивает получение высококачественных черно-белых и цветных копий на бумаге и прозрачной пленке.

Таким образом, современные программы предусматривают множество функций, позволяющих готовить текстовую часть документа на типографическом уровне. Кроме того, современные программы позволяют включать в текст графические объекты: рисунки, диаграммы, фотографии. Благодаря этим возможностям файл, представляющий собой текстовый документ, может содержать, помимо алфавитно-цифровых символов,

обширную двоичную информацию о форматировании текста, а также графические объекты.

Текстовый процессор — вид прикладной компьютерной программы, предназначенной для производства (включая набор, редактирование, форматирование, иногда печать) любого вида печатной информации. Иногда текстовый процессор называют текстовым редактором второго рода.

Текстовыми процессорами в 1970-е — 1980-е годы называли предназначенные для набора и печати текстов машины индивидуального и офисного использования, состоящие из клавиатуры, встроенного компьютера для простейшего редактирования текста, а также электрического печатного устройства. Позднее наименование «текстовый процессор» стало использоваться для компьютерных программ, предназначенных для аналогичного использования.

Текстовые процессоры, в отличие от текстовых редакторов, имеют больше возможностей для форматирования текста, внедрения в него графики, формул, таблиц и других объектов. Поэтому они могут быть использованы не только для набора текстов, но и для создания различного рода документов, в том числе официальных. Классическим примером текстового процессора является MicrosoftWord.

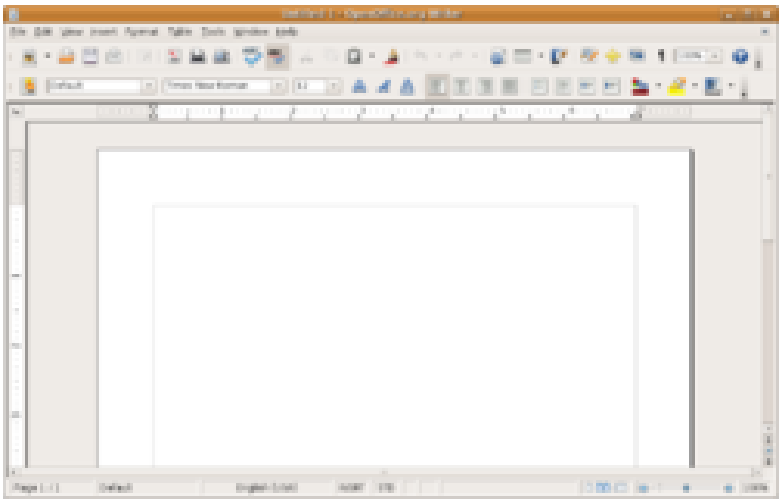


Рис. 11.

OpenOffice.org Writer

Программы для работы с текстами можно разделить на простые текстовые процессоры, мощные текстовые процессоры и издательские системы.

Известные текстовые процессоры

- AbiWord
- AdobeInCopy
- JWPce — текстовый процессор для японского языка.
- LotusWordPro
- MicrosoftWord
- MicrosoftWorks
- OpenOffice.org Writer
- PolyEdit
- WordPad — входит в дистрибутив MS Windows
- WordPerfect

2.Функции MicrosoftWord.

Текстовый процессор MSWord предназначен для создания, редактирования и распечаткидокументов.

Документ-текст,структурнымиэлементамикоторогоявляютсяслова,предложения,

абзацы,разделы,страницыит.д.

По своим функциям Word вплотную приближается к издательским системам и программамверстки. Этозначит,чтовэтомредактореможнополностьюподготовитькпечати документ практически любого уровня сложности.

Функциональные возможности текстового процессора Word :

– создание нового документа с помощью специальных шаблонов (стандартные письма, поздравительные записки, отчеты, факсы и др.);
– одновременное открытие и работа с большим количеством документов;
– автоматическая проверка орфографии, грамматики и даже стилистики при вводе документа;
– автоматическая коррекция наиболее часто повторяющихся ошибок;
– расширенные возможности форматирования документа, Word допуска
– использование стилей для быстрого форматирования документа;
– возможность автоматизации ввода повторяющихся и стандартных элементов текста;
– автоматизация формирования и работы со списками;
– удобные механизмы работы с ссылками, сносками, колонтитулами;
– включение в текст элементов, созданных в других программах MicrosoftOffice , -

графических изображений, электронных таблиц и т.д.;

– возможность подготовки простых электронных таблиц и гипертекстовых документов в интернете;
– возможность работы с математическими формулами,
– возможность автоматического создания указателей и оглавления документа;
– возможность отправки готового документа непосредственно из MS Word на факс и по электронной почте;
– расширенные возможности индексации готового документа;
– встроенный мастер подсказок и объемная система помощи.

В MicrosoftWord реализованы возможности технологии OLE– связывания и внедрения объектов. Эти технологии позволяют включать в документ текстовые фрагменты, иллюстрации, таблицы, созданные другими приложениями Windows. Встроенные объекты можно редактировать средствами этих приложений. В MicrosoftWord применены новые технологические решения: система готовых шаблонов и стилей оформления, приемы создания и модификации таблиц, функции автотекста и автокоррекции и т.д.

Все эти, а также другие достоинства программы позволяют просто, удобно и легко готовить как простые тексты, так и сложные документы.

3.Применение MicrosoftWord.

ВводтекставWord осуществляетсяпострочно, переходнаследующуюстроку в пределаходногоабзацавыполняетсяавтоматически.Посленажатияклавиши<Enter>завершае тсяпредыдущийабзациначинаетсяновый.

Команды Правка**,** Отменить и Правка**,** Повторить или кнопки и позволяют последовательноотменитьилиповторитьпредшествующиедействия.

Элементамиинтерфейсатекстовогопроцессораявляются(см.рисунок 3):

Строка иерархического меню, содержащая имена групп команд, объединенных по функциональному признаку. Строка меню находится в верхней части экрана Система вложенных (ниспадающих) меню составляет основу интерфейса текстового процессора. Команды меню выбираются с помощью мыши, клавиш управления курсором или комбинаций нажатия определенных клавиш

Строка состояния (статуса) содержит имя редактируемого документа и

определяет положение курсора в этом документе. В строке выводится справочная информация

Строка подсказки содержит информацию о возможных действиях пользователя в текущий момент.

Панели инструментов, ускоряющие доступ к функциям текстового процессораРабочее поле – это пространство на экране дисплея для создания документа и работы с ним. Максимальный размер рабочего поля определяется стандартными параметрами монитора.

Координатные линейки определяют границы документа и позиции табуляции (горизонтальная линейка). По умолчанию координатная линейка градуирована в сантиметрах.

Полосы прокрутки служат для перемещения текста документа в рабочем поле окна.

Клавиатурный курсор, показывает позицию рабочего поля, в которую будет помещен вводимый символ.

Управление интерфейсом осуществляют при помощи клавиатуры и мыши.

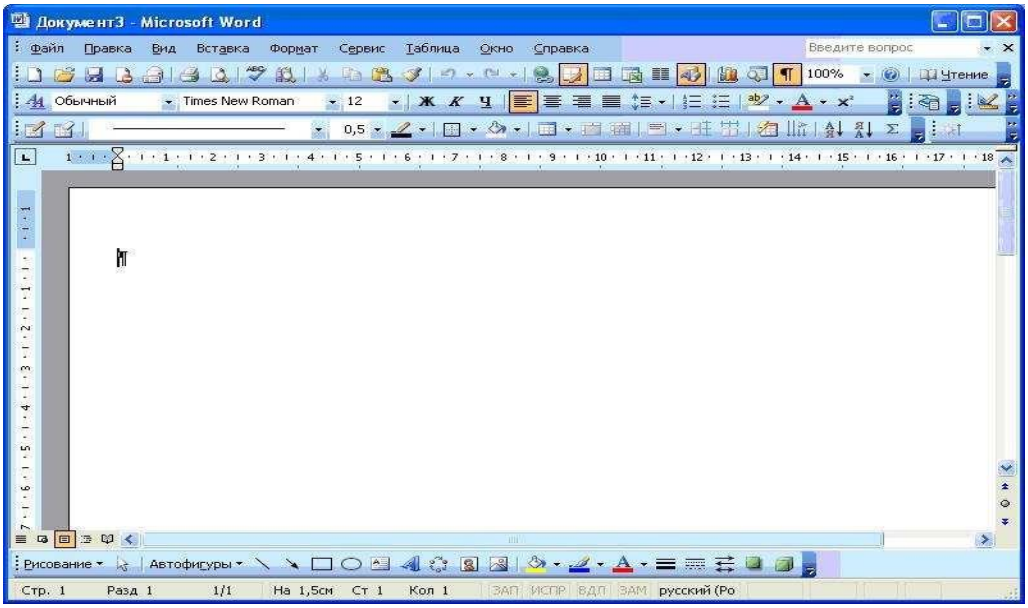


Рис. 12.. ИнтерфейстекстовогопроцессораMicrosoftWord.

Приизучениитемынеобходимообратитьвниманиенаследующиевопросы:

1. Порядоксозданияновогодокумента:

а)выбратьтипсоздаваемогодокументаилишаблона(меню«Файл»,«Создать»);

б)установитьполяиориентациюстраниц(меню«Файл»,«Параметрыстраницы», закладки«Поля»,«Размербумаги»).Еслипредполагаетсясозданиеброшюры,включить опцию«2страницыналисте»;

в)установитьнеобходимыепараметрыиспользуемогошрифта(меню«Формат», «Шрифт»,закладки«Шрифт»,«Интервал»);

г)установитьнеобходимыепараметрыабзаца(меню«Формат»,«Абзац», закладки «Отступыи интервалы»,«Положениена странице»);

д)принеобходимостиизменитьпараметрыредактора(меню«Сервис», «Параметры»,закладки«Вид»,«Общие»,«Сохранение»,«Совместимость»идр.);

е)ввестиипринеобходимостиотформатироватьтекст;

ж)сохранитьдокумент(меню«Файл»,«Сохранить»),присвоивфайлуимя.

2. Копирование,перемещениеиудалениетекста,работасфрагментами текста.

Фрагментомназываетсянепрерывнаячастьтекста. Выделениефрагментаделает

егообъектомпоследующейкоманды.Выделениеобъектаосуществляетсяспомощью мыши или клавиатуры. Выделенный фрагмент может быть строчным, блочным (несколько строк) или линейным (часть строки). При копировании, перемещении, удалении объект помещается в буфер обмена (область оперативной памяти).Технология выполнениявышеуказанных операцийвключает в себя несколько этапов:

- Выделениечаститекста(фрагмента),
- Переносвыделенногофрагментавбуферпромежуточногохранения;
- Перемещениекурсоравнужноеместодокумента;копирование(перенос)выделенногофрагментаизбуферавместодокумента, указанноекурсором.

3. Форматированиедокумента.

Процесс оформления внешнего вида документа в целом или его фрагментов в любой программной среде называют *форматированием.* Само слово «форматирование» происходит от слова «форма», т.е. чему-либо надо придать определенную форму. Различные способы и инструменты форматирования, которые предоставляет текстовый процессор Word, позволяют получить профессионально оформленный текст.

Форматирование документов осуществляется в результате следующих действий:

- установки параметров страницы документа;
- применения шрифтового оформления символов текста;
- задания положения абзацев на странице и установка для них отступов и интервалов (слева и справа, межстрочный и межабзацный интервалы);
- и выбора вариантов обрамления и заполнения абзацев;
- расположения текста в колонках;
- задания стиля оформления символа, абзаца, страницы и т.п.

Большая часть этих действий может быть реализована с помощью инструментов меню Формат**.** Форматирование документа основано на задании новых форматов элементам текста, которые должны быть предварительно выделены.

*Внимание**!*** Прежде чем форматировать текст, надо его выделить!

3.1. Шрифтовое выделение текста (форматирование символов)

Текст документа набирается установленным по умолчанию шрифтом, настройка которого выполняется в диалоговом окне <Шрифт>, вызываемого командой Формат**,** Шрифт**.** Установки формата шрифта могут быть сделаны для любого фрагмента текста. Установленныепараметрьг шрифта действуют применительно ко вновь вводимому тексту или к выделенному фрагменту текста.

*Внимание!*Дляшрифтовоговыделенияфрагментатекстанеобходимоего предварительно выделить. Диалоговое окно команды Шрифт содержит вкладки (рис.13.).

Накаждой вкладкевокне <Образец>отображаетсярезультатнастройки шрифта.

Вкладка*Шрифт*спараметрами:

- *типшрифта.*Длявводарусскихбуквобычноприменяютсяшрифты:TimesNewRoman, Arial, Courier и др.;
- *начертание*шрифта:обычный,курсив,полужирный,полужирныйкурсив;
- *размер*шрифтавпунктах(пт)илидругихединицах;
- *подчеркивание*выделенногофрагменталиниямиразноготипа;
- *цвет*шрифта;
- *однойилидвумязачеркнутыми*линиямисимволы выделенного фрагмента;
- отображениевыделенныхсимволовнауровне*верхних*или*нижних*индексов;
- сделатьвыделенныйфрагмент*скрытым;*
- выделенный фрагмент отображать обычными или малыми прописными буквами, стеньюилипоконтуру(двойнойобводкой), приподнятымилиутопленным.

Вкладка*Интервал* спараметрами:

- *интервал,*которыйпозволяетуказатьрасстояниевпунктах(пт):нормальное, разреженное,уплотненное.Расстояниеможноизменитьрядомвокне;

– *смещение* устанавливаетсмещениевыделенногофрагментавптотносительно базовойлинииввверхиливниз;

– *кернинг*служитдляавтоматическогоподбораинтерваламеждусимволами.

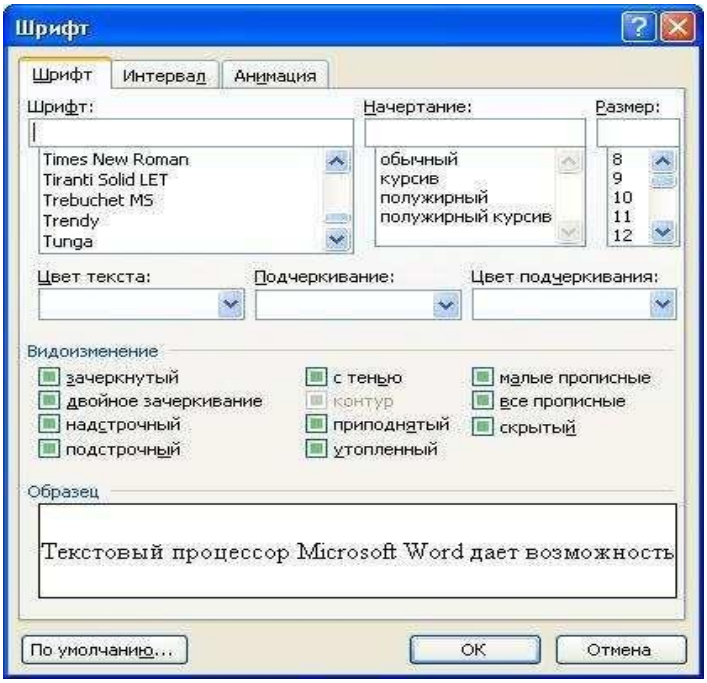


Рис. 13.ДиалоговоеокнокомандыШрифт

3.2. Форматированиеабзацатекста

Текстдокументасостоитиз*абзацев,*абзацзаканчиваетсянажатиемклавиши <Enter>.Приэтомвтекстевставляетсяспецсимвол К.Удалениеданногосимвола обеспечиваетслияниеабзацев,причемобъединенныйабзацполучаетформатныеустановки нижнего присоединенного абзаца.При наборе текста переход на новую строкувыполняетсяавтоматически.

Формат абзацев устанавливается командой Формат**,** Абзац**,** которая вызывает диалоговое окно «Абзац» (рис.14), содержащее вкладки *Отступы и интервалы. Положение на странице*На вкладке *Отступы и интервалы* задаются:

- выравнивание—по ширине, по центру, по левому или правому краям;
- выбор уровня структуры документа, которому приписываются сделанные установки;
- границы абзацев(отступов) слева и справа от края печатного листа;
- интервалы—междустрочный и межабзацный (перед и после);
- вид первой строки абзаца—с отступом вправо (красная) или влево (висячая).

Вкладка *Положение на странице* определяет правила разбиения строк строку абзаца на другой странице (можно не менее 2 строк);

– Неразрыватьабзац,т.е.располагатьцеликомнаоднойстранице;

– Неотрыватьотследующегоабзаца—текущийиследующийабзацыпечатаются наоднойстранице;

– сновой страницы —выделенныйабзац начинать сновойстраницы,вставляя разделительстраниц;

– запретитьнумерациюстрокабзаца;

– запретитьавтоматическийпереносслов.

Быстрое форматирование абзацев может выполняться с помощью панели*Форматирование,*котораясодержитнеобходимыекнопки выравнивания абзаца.

Всепрочиеустановкиформатаабзацавыполняютсявдиалоговомокне«Абзац».

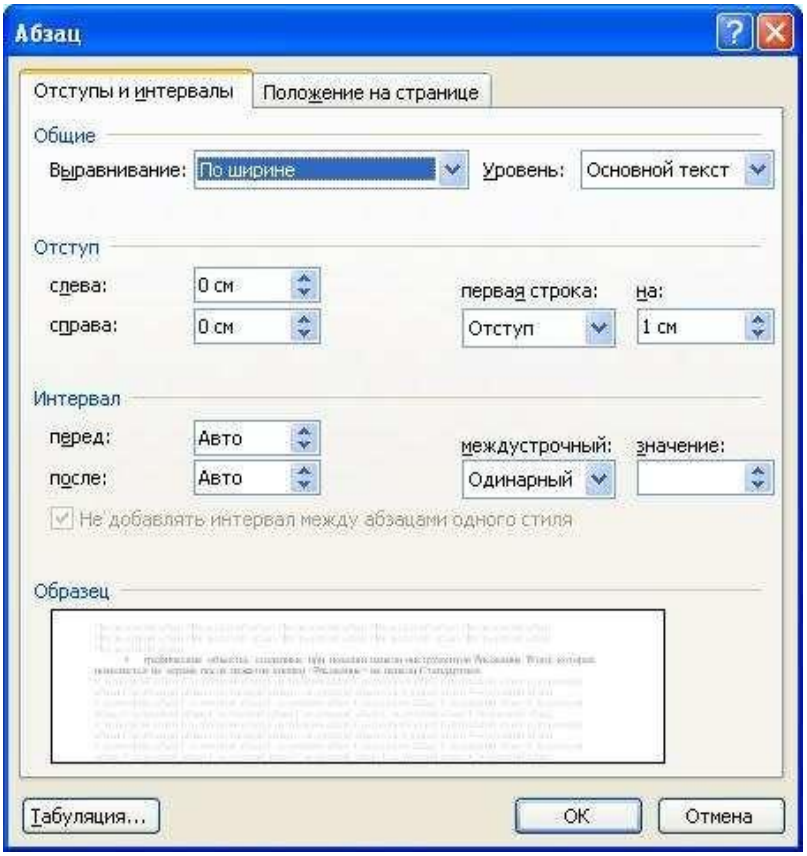


Рис.14.ДиалоговоеокнокомандыАбзац**.**

3.3. Измененирегистрадляизображения букв

Регистровое форматирование обеспечивает преобразование выделенного фрагментатекстаспомощьюкомандыФормат**,**Регистривыборомсоответствующего кнопочногопереключателя:

- *Каквпредложениях*первыйбуквапервогословапредставляетсякак прописная;
- *Всестрочные*—всебуквывыделенноготекстастрочные;
- *ВСЕПРОПИСНЫЕ*—всебуквывыделенноготекстапрописные;
- *НачинатьсПрописной*—первыйбуквакаждоговыделенногословапрописная;
- *ИЗМЕНИТЬРЕГИСТР*—заменавыделенномтекстепрописныхбуквна строчные, астрочных—напрописные.

4. Поискизаменафрагментовтекста

ВдокументахWordможноосуществлятьпоискизаменутекста,форматов(шрифта, абзаца,языка,стиля),различныхспециальныхсимволов(маркераабзаца,сноскиили примечания, концараздела, колонкиит.п.). Дляэтогоиспользуюткомандыправка**,** Найти(толькопоиск)илиПравка**,**Заменить(поискизамена).

В окне «Найти» указывается образец поиска, в окне «Заменить на» — образец замены.Объектомпоискаизаменяявляетсягруппасимволовсучетомилибезучета формата— шрифт, абзац, язык, стиль. Далееприводятсяданныедляиспользования расширенногонаборакритериевпоиска.

Направлениепоискапоотношениюктекущейустановкекурсоравтекстедокумента указываетсявокне«Направление»(Везде,Вперед,Назад).

Кнопка<Найтидалее>«ищет»следующеевхождениеобразца,указанноговполе«Найти» .

Кнопка<Заменить>»заменяет«найденныйобразецнаобразец,помещенныйвполе«Замен

ительна»,и«ищет»следующеевхождение.Кнопка<Заменитьвсе>«ищет»всевхожденияобразцаиз поля«Найти»и«заменяет»

Егонаобразецизполя«Заменитьна»безпредварительныхзапросов.

5. Автозаменатекста

Автотекст—фрагментдокумента, включающийтекстилиграфику,которыйможет использоваться для вставки в документ под управлением пользователя. Примерами элементов автотекста являются:

– подписи должностных лиц с указанием названия занимаемойдолжности,званияи т.п.;
– полные названия организаций;
– «шапки»стандартныхформдокументов;
– Типовыебланкидокументовидр.

Хранениеэлементовавтотекстаосуществляетсявшаблонедокумента,поэтомуони доступнывсемдокументам,которыесвязанысэтимшаблоном.

Работа с элементами автотекста происходит по команде Сервис**,** Автозамена и установкойзначенийпараметроввдиалоговомокне«Автотекст».Элементыавтотекстаможнодо бавлятьиудалять,нонередактировать.

Кнопка<Добавить>добавляетвыделенныйфрагментказановыйэлементавтотекстасзадан нымименем.Кнопка<Вставить>вставляетвыбранныйэлементвтекстдокумента.

*Автозамена*работаетвдинамическомрежиме(вотличиеотавтотекста,который вставляется под управлением пользователя). Элементы автозамены создаются и удаляются с помощью команды Сервис**,** Автозамена. Элементы автозамены можно добавлять, заменять и удалять.

Указываютсяпереключателирежимаавтокоррекции:

– Исправлениедвойныхначальныхзаглавныхбукв;
– Перваябуквапредложения—прописная;
– Устранениепоследствийслучайногонажатияклавиши<CapsLock>;
– Заменатекставпроцессенабора.Впервомполеуказываетсяисходныйтекст.Во втором полеуказываетсяновыйтекстсучетомилибезучетаформатирования.

6. Проверкаорфографии

Команда Сервис**,** Правописание**,** вкладка *Правописание* позволяет задать параметрыпроверкитекста.

Проверка орфографии выполняется по основному словарю выбранного языка и по дополнительнымсловарямпользователя,которыемогутсодержатьпроизвольныеслова (например,условныеобозначения,ключевыесловаязыкапрограммированияидр.).

Можносоздатьновыйсловарьпользователялибоотредактироватьсуществующий словарь.

Проверкаорфографииохватываетосновнойтекст, текстколонтитулов, сносок, концевыхсносокипримечаний.

7. Спискидля оформленияперечисленийвтексте

Перечисления в текстовых документах часто оформляются в виде списков. Различаюттритипасписков:*маркированный,нумерованный,многоуровневый.*На рис.6приведеныпримерытрехтиповсписков.Списокформатируетсякакдонводаэлементов, такидляуженабранныхввидеотдельныхабзацевэлементов. Длясозданныхсписков допускаетсяизменениеихтипа.

Существуетнесколькоразличныхспособовформатированиясписков:

– СпомощьюкомандыФормат,Список;
– СпомощьюкомандыСписокизконтекстногоменю;
– Быстроеформатированиеспомощьюкнопок<Нумерация>и<Маркеры>на панели *Форматирование.*

КомандаФормат**,** Списоквыводитдиалоговоеокно«Список» длявыборавкладки, соответствующейтипусписка.Выбранныйтипспискаможнонастроить,нажавкнопку<Изменить>иустановиввдиалоговомокне«Изменениесписка»необходимые параметры:

– Длямаркированногоспискавыбираетсясимвол(маркер)изнаборашрифтов Word; задается размер и цвет маркера; указывается положение маркера и положениетекста(отступы);

– Длянумерованногоспискауказываетсяформатномераизнаборашрифтов Word; положениесписканастранице(полевомуилиправомукраю,поцентру)иего отступ;отступтекстаотномера;начальныйномерсписка;для многоуровневого списка указывается номер иерархического уровня,а далее для выбранного уровня осуществляется настройка параметров по тем же правилам,тоидлянумерованногосписка.

Иллюстрация типов списков

Табл.3.

Маркированныйсписок	Нумерованныйсписок	Многоуровневыйсписок
Аппаратноеобеспечение: • Системныйблок • Монитор • Клавиатура • Принтер Программноеобеспечение: • Системное • Прикладное • Инструментарийпрограммирования	Аппаратноеобеспечение: I. Системныйблок II. Монитор III. Клавиатура IV. Принтер Программноеобеспечение: I. Системное II. Прикладное III. Инструментарийпрограммирования	1. Аппаратноеобеспечение: 1.1. Системныйблок 1.2. Монитор 1.3. Клавиатура 1.4. Принтер 2. Программноеобеспечение: 2.1. Системное 2.2. Прикладное 2.3. Инструментарийпрограммирования

Соответствующиеотступыдляэлементовспискаможноизменитькакспомощью команды Формат**,** Список**,** так и с помощью кнопок панели инструментов *Форматирование*—<Уменьшитьотступ>,^Увеличитьотступ>.Крометого,спомощью мышинагоризонтальнойлинейкедлявыделенныхэлементовспискаможновыполнить перемещениеуказателейотступов.

Дляизмененияуровняиерархиисследуетустановитькурсорнаэлементинажать клавиши:

<Shift><Alt><->>дляпонижения уровняиерархии;

<Shift><Alt><<+>дляповышения уровняиерархии.

УдалитьсписокможнообычнымиспособамилибоспомощьюкомандыФормат**,** Списоккнопкой<Удалить>.

8. Многоколоночнаяверстка

Для текстов газетного типа выполняется набор в несколько колонок, после заполнения левой колонки (по высоте страницы или до установленного ограничения) курсор автоматически переходит в следующую колонку.

Любые вставки или удаления текста и графики внутри колонок автоматически обеспечивают "перетекание" текста из колонки в колонку. Текст колонок форматируется по общим правилам.

Формат газетного текста задается с помощью команды Формат**,** Колонки**:**

– количество колонок (одна, две, три и т.д. колонки);

– ширина каждой колонки (или одинаковая ширина всех колонок);
– наличие разделительной линейки между колонками.

Если документ новый, после выполнения данной команды текст вводится в заданное число колонок на странице.

Существующий текст также можно расположить в колонках, предварительно выделив либо его фрагмент, либо весь документ.

9. Нумерация страниц

Для нумерации страниц используется команда Вставка**,** Номера страниц**,** с помощью которой можно указать:

– положение — вверху или внизу страницы;
– выравнивание — справа, в центре, слева, снаружи или внутри страницы;
– номер первой страницы;

10. формат номеров страниц.

Использование стилей для быстрого форматирования документа

Все документы Word создаются с помощью команды Файл**,** Создать**.** Важным компонентом создаваемого документа являются *стили.* Стили позволяют быстро оформлять разнообразные по внешнему виду и характеру тексты. Стартовый набор стилей выбирается из присоединенного к документу шаблона. *Стиль* — поименованная совокупность форматов элементов текста.

Различают стандартные и пользовательские (специальные) стили. Стандартные стили создаются текстовым процессором Word автоматически. Пользовательские стили создаются пользователем модификацией стандартных или в результате отбора из имеющихся характеристик требуемых. Стиль пользователя может быть доступным либо только для отдельного документа, либо для шаблона.

Для сокращения трудоемкости форматирования документа используются:

– приписывание стандартных стилей к выделенным фрагментам текста;
– создание новых стилей;
– переопределение (изменение) стилей;
– заимствование стилей других шаблонов.

С помощью команды Формат**,** Библиотека стилей вызывается соответствующее диалоговое окно, содержащее список шаблонов документов.

Основные положения**,** которые следует помнить при форматировании документа**:**

– Документ Word состоит из разделов (секций), каждый из которых разделен на абзацы.
– Каждый раздел имеет свои параметры страницы (поля, расположение и вид колонтитулов, количество колонок).
– Каждый абзац характеризуется своим расположением (интервалы сверху и снизу, межстрочный интервал, шрифт, вид первой строки, тип списка и т.п.).
– К каждому абзацу приписан стиль, параметры которого можно модифицировать.
– Каждый стиль может основываться на другом стиле. Большинство стилей, как правило, основано на стиле "Обычный", изменение которого отразится на всем документе.
– Параметры абзаца ассоциируются с маркером его конца, удаление которого приведет к удалению стилевого оформления абзаца.
– Параметры раздела ассоциируются с разделителем (маркером) его конца, удаление которого приведет к удалению стилевого оформления раздела.

11. Работа с таблицами

Документы Word часто содержат данные, оформленные в виде *таблицы.* Обычно таблицы используются для более удобного расположения информации документа.

Таблица состоит из *столбцов* и *строк,* на пересечении которых находятся *ячейки.* Таблица Word может содержать максимум 63 столбца и произвольное число строк. Ячейки таблицы имеют адреса , образованные именем столбца (A B C...) и номером строки (1, 2, 3...)В ячейках таблиц размещается информация произвольного типа: текст, числа, графика,рисунки,формулы.

Высота строк таблицы — произвольная, может различаться для разных строк таблицы,ноячейкиоднойстрокиимеютодинаковуювысоту.Ширинаячеекоднойстроки и даже одного столбца — произвольная, в том числе и одинаковая. Первоначально указанноеприсозданиитаблицычислострокистолбцовможноизменять, добавляя новыеилиудаляясуществующиестрокиистолбцы.

Word предоставляет возможность использования таблиц произвольной конфигурации, сразличнымчисломстрок, столбцовдаженауровнеотдельнойстроки таблицы.

Способы создания таблиц

Существует несколько способов создания таблиц в документе,которые вставляются поместуустановкикурсора.

1- й способ. Новую таблицу можно создать с помощью команды главного меню Таблица**,** Добавитьтаблицу**,** котораявызываетдиалоговоеокно«Вставкатаблицы» для указанияразмерноститаблицы—числастрокистолбцов.Послеуказанияразмерности создаваемойтаблицыможноосуществитьееавтоформатирование,нажавкнопку<Автоформат>. В окне «Автоформат таблицы» можно выбрать формат, который отображаетсяврасположенномрядомокне«Образец».Переключателивнижнейчасти окна позволят вам сделать дополнительное оформление для выбранного стандартного типа таблицы.

2- й способ. С помощью кнопки<Вставить таблицу> на *Стандартной* панели инструментовопределяетсяструктурановойтаблицы—принажатойлевойкнопкемыши закрашиваетсятребуемоечислостолбцовистроктаблицы.

3- й способ. С помощью команды Вставка**,** Объект**,** вкладка *Создание объекта* можновдокументWordвставитьэлектроннуютаблицуExcel,указатипобъекта— *ЛистMicrosoftExcel.*

4- й способ. Ранее набранный текст может быть преобразован в табличное представлениеспомощьюкомандыТаблица**,**Преобразоватьвтаблицуприусловии, что текст подготовлен с использованием специальных символов — разделителей колонок(табулятор,абзац,пробелидр.).

Измененииеструктурытаблицы

Изменение структуры таблицы означает изменение числа строк и столбцов первоначально созданной таблицы. Выполнению команд по изменению структуры таблицы должно предшествовать выделение соответствующего блока ячеек.

Использованиевтаблицеформул

Текстовый процессор Word позволяет выполнять вычисления, записывая в отдельные ячейки таблицы формулы с помощью команды «Таблица», «Формулы». Формулазадаетсякаквыражение,вкотороммогутбытьиспользованы:

а)абсолютныессылкинаячейкитаблицыввидесписка(A1;B5;E10)илиблока (A1:F10);

б)ключевыесловадлярссылкинаблокячеек:

- LEFT—ячейки,расположенныевстрокелевееячейкисформулой ;
- RIGHT—ячейки,расположенныевстрокеправееячейкисформулой;
- ABOVE—ячейки, расположенныевстолбценижеячейкисформулой;

в)константы-числа,текствдвойныхкавычках;

г)встроенныефункцииWord;
д)знакиопераций(+;—;*;/;%;=;<;<=;>;>=).

12. Графика

Всоставныхдокументах Wordчастоиспользуетсяразличноговидаграфика:

– клипы— рисункиизколлекции, созданнойпроизводителямипрограммного обеспечения;

– графические объекты,хранящиеся в файлах и созданные специализированными средствамимашиннойграфики;

– графическиеобъекты, созданныеприпомощипанелииинструментов*Рисование* Word, которая появляется на экране после нажатия кнопки <Рисование> на панели *Стандартная.*

Вопросы:

1. Какие вы знаете автоматизированные средства и технологии организации текста?
2. Назначение и основные возможности текстового процессора Word?
3. Как подготовить страницы для размещения на ней текстового документа?
4. Каковы правила ввода и редактирования текста?
5. Технология форматирования документа?
6. Как расставить и настроить колонтитулы?
7. Технология работы с таблицами?
8. Работа с иллюстрациями и внедренными объектами?

Лекция №5.

На тему: «Понятие алгоритма и основные алгоритмические структуры. Языки программирования. Предпрограммная подготовка задачи.»

План:

1. Понятие алгоритма. Свойства алгоритма. Основные алгоритмические структуры. Константы, переменные и ячейки памяти. Массивы. Вспомогательные алгоритмы. Декомпозиция алгоритма.
2. Языки программирования.
3.Предпрограммная подготовка задачи.

1. Понятие алгоритма и основные алгоритмические структуры.

Алгоритм – это инструкция о том, в какой последовательности нужно выполнить действия при переработке исходного материала в требуемый результат. (последовательность точных предписаний, понятных исполнителю (компьютеру, роботу и пр.), совершить последовательность действий, направленных на достижение конкретного результата)

Наряду с понятием алгоритма используют термин *алгоритмизация* , под которой понимают совокупность приемов и способов составления алгоритмов для решения алгоритмических задач.

Часто алгоритм используется не как инструкция для автомата, а как схема алгоритмического решения задачи. Это позволяет оценить эффективность предлагаемого способа решения, его результативность, исправить возможные ошибки, сравнить его еще до применения на компьютере с другими алгоритмами решения этой же задачи.

Наконец, алгоритм является основой для составления программы, которую пишет программист на каком-либо языке программирования с тем, чтобы реализовать процесс обработки данных на компьютере.

Всякий алгоритм должен обладать следующими свойствами:

1. Результативность. Получение требуемого результата за конечное число шагов; это означает, что неправильный алгоритм, который не достигает цели, вообще не нужно считать алгоритмом. Алгоритмическая инструкция лишь тогда может быть названа алгоритмом, когда при любом сочетании исходных данных она гарантирует, что через конечное число шагов будет обязательно получен результат.

*2. Дискретность (пошаговость).*Под дискретностью понимают, что алгоритм состоит из последовательности действий, шагов. Выполнение каждого следующего шага невозможно без выполнения предыдущих. Последний шаг, как правило, выдаёт результат действия алгоритма.

*3. Детерминированность (определённость).*Означает, что действия, выполняемые на каждом шаге, однозначно и точно определены.

*4. Понятность.*Алгоритм должен быть понятен не только автору, но и исполнителю.

*5. Выполнимость.*Алгоритм должен содержать команды, записанные на понятном языке и выполнимые исполнителем.

*6. Массовость.*Один тот же алгоритм может применяться для решения большого количества однотипных задач с различающимися условиями.

На практике получили известность два способа изображения алгоритмов:

1) в виде пошагового словесного описания;
2) в виде блок-схем.

Первый из этих способов получил значительно меньшее распространение из-за его многословности и отсутствия наглядности. Второй, напротив, оказался очень удобным средством изображения алгоритмов и получил широкое распространение в научной и учебной литературе. Именно этот способ будет использован ниже при составлении и описании алгоритмов.

Блок-схема – это последовательность блоков, предписывающих выполнение определенных операций, и связей между этими блоками. Внутри блоков указывается информация об операциях, подлежащих выполнению. Конфигурация и размеры блоков, а также порядок графического оформления блок-схем регламентированы ГОСТ 19002-80 и ГОСТ 19003-80 "Схемы алгоритмов и программ".

В табл. 4. приведены наиболее часто используемые блоки, изображены элементы связей между ними и дано краткое пояснение к ним. Блоки и элементы связей называют элементами блок-схем.

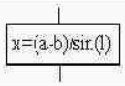
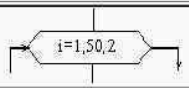
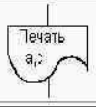
При соединении блоков следует использовать только вертикальные и горизонтальные линии потоков.

Горизонтальные потоки, имеющие направление справа налево, и вертикальные потоки, имеющие направление снизу вверх, должны быть обязательно помечены стрелками.

Прочие потоки могут быть помечены или оставлены непомеченными.

По характеру связей между блоками различают алгоритмы линейной, разветвляющейся и циклической структуры.

Табл.4.

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Константы, переменные и ячейки памяти

Для того чтобы ясно представить как "работает" алгоритм, опишем простейший автомат, который предназначен для выполнения операций, предписанных этим алгоритмом.

В состав такого автомата входят:

- память, состоящая из отдельных ячеек;
- считывающая/записывающая головка;

процессор, т. е. устройство, способное выполнять операции, в том числе математические, и отдавать головке указания читать данные из ячеек или записывать данные в ячейки памяти автомата.

Головка, получив указание от процессора, может записывать в ячейку или считывать из нее одну константу .

В простейшем случае константой является любое арифметическое число. Например, 12, 0.78, 0, –45.33 и т. д. (Константами могут быть такие строки символов, структуры данных и др.).

Под простой переменной , или просто переменной будем понимать некоторую ячейку памяти, т. е. отдельное место для хранение одной константы. В отдельной ячейке за время работы алгоритма может побивать множество различных констант (отсюда название – переменная). Такими ячейками (электронными, магнитными, оптическими) снабжен реальный компьютер.

Переменные имеют буквенно-символьное обозначение. Например, 1, n, a, a1, b , H2 – переменные. Одновременно обозначение переменной является индексом ячейки, в которую будут записываться константы. Любая из таких констант называется значением переменной . Например, Z является переменной и адресом ячейки Z одновременно. С алгоритмической точки зрения понятия “переменная” и “адрес ячейки” памяти являются идентичными.

Запись вида Y = 5.5 следует понимать так: записать константу 5.5 в ячейку с адресом Y (если до этой операции в ячейку была записана константа, то она будет затерта, а на ее место будет помещена константа 5.5). Произносить эту запись следует так: “переменной Y присвоить значение 5.5” .

Запись вида L = M следует понимать так: прочитать константу, расположенную по адресу M и скопировать эту константу в ячейку с адресом L (при этом константа из

ячейки M не удаляется, а остается такой, какой она была до чтения). Произносить эту запись нужно так: "переменной L присвоить значение переменной M (или просто: L присвоить M)".

Массивы

Другой разновидностью переменных являются так называемые индексированные переменные или массивы . Массив – это некоторая совокупность ячеек, объединенная одним обозначением (массивом может быть одна ячейка). Всякий массив обязательно имеет размерность. Массивы бывают одномерными, двумерными, трехмерными и т. д.

Одномерный массив – это последовательность ячеек, расположенных в одну линию. Пример такого массива:

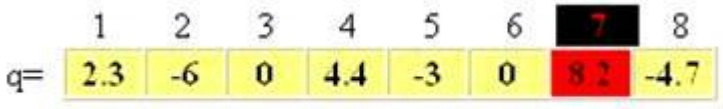


Рис. 1. Одномерный массив *q*

Рис.15.

Массив имеет имя q. Для того чтобы можно было отличить одну ячейку массива от другой ячейки этого же массива, их нумеруют. Нумерация ячеек обычно начинается с 1. Номер ячейки массива называется его индексом , а константа в ячейке – элементом массива. Теперь становится возможной работа с отдельной ячейкой такого массива. Например, команда q 7 = 8.2 означает, что в 7-ю ячейку массива q надлежит записать константу 8.2. Команда r 41 = q 2 + q 5 означает, что нужно сложить константы, записанные во 2-ю и 5-ю ячейки массива q, и результат записать в 41-ю ячейку одномерного массива r. Эту же операцию можно описать другими словами: 41-му элементу массива r присвоить значение суммы 2-го и 5-го элементов массива q.

Двумерный массив по расположению ячеек напоминает математическую матрицу. Элемент такого массива характеризуется двумя индексами: первый показывает строку, в которой расположена ячейка, второй – ее столбец. Например, команда d 2, 5 = 43 означает, что в ячейку, размещенную на пересечении 2-й строки и 5-го столбца двумерного массива d, нужно записать константу 43.

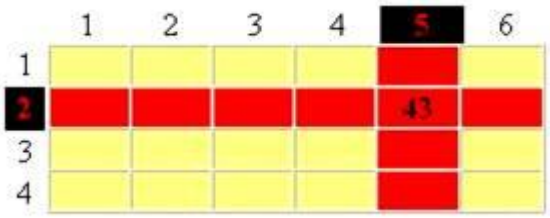


Рис.2. Двумерный массив d

Рис.16.

Аналогично устроена структура массивов трех- и большей размерности.

Линейные алгоритмы

Линейный алгоритм – это алгоритм, в котором блоки выполняются последовательно сверху вниз от начала до конца.

На рис. 17. приведен пример блок-схемы алгоритма вычисления периметра P и площади S квадрата со стороной длины A.

Блок-схема алгоритма состоит из шести блоков. Выполнение алгоритма начинается с блока 1 "Начало". Этот блок символизирует включение автомата, настройку его на выполнение алгоритма и выделение памяти под все переменные, которые задействованы в алгоритме. В алгоритме рис. 3 таких переменных три: A, P, S. Следовательно, под каждую из них алгоритмом будет выделено по одной ячейке памяти. На этом блок 1 будет отработан.

Как видно, блок 1 связан вертикальной линией потока с блоком 2. Эта линия не имеет стрелки, указывавшей направление потока. Следовательно, этот поток направлен вниз. Таким образом, после выполнения блока 1 управление будет передано на блок 2. Блок 2 показывает, что переменной A следует присвоить значение. Это означает, что в ячейку, отведенную автоматом под эту переменную, нужно поместить константу. На реальной компьютере эта константа может быть введена самыми разными способами. Способ зависит от того, как запрограммирован данный фрагмент. Можно, например, потребовать ввод константы с клавиатура или получить его из заранее подготовленного файла.Возможно эта константа будет получена через внешние источники данных, например, от физической установки, подключенной к компьютеру.

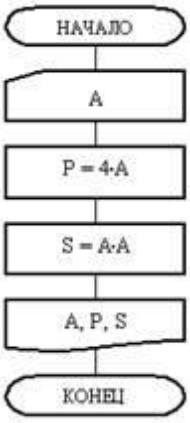


Рис. 17. Линейный алгоритм

Для данного примера способ передачи константы не имеет значения, важно лишь то, что при выполнении блока 2 в ячейку с адресом A будет занесена конкретная константа. Пусть такой константой является число 5.

Далее управление по линии потока передается к блоку 3 "Процесс". В этом блоке при выполнении размещенной в ней команды число 4 умножается на константу, помещенную в ячейку A (т. е. 5), и результат (т. е. 20) присваивается переменной P (т. е. константа 20 записывается в ячейку по адресу P). После выполнения этих операций управление передается к блоку 4.

В блоке 4 аналогичным образом производится умножение значений переменной A и результат (константа 25) присваивается переменной S (в ячейку по адресу S будет занесена константа 25). После этого выполняется переход к блоку 5.

При выполнении команд блока 5 выводятся (например, на экран, бумагу, во внешний файл и т. д.) значения переменных A, P, S, которые сохранились в соответствующих ячейках к этому моменту. Понятно, что для конкретного примера A = 5 будут выведена константы 5, 20, 25, т. е. длина сторона квадрата, его периметр и площадь. Далее управление передается последнему блоку 6.

В блоке б “Конец” производится освобождение ячеек памяти, которые были зарезервированы под переменные A, P, S, и алгоритм заканчивает работу.

Понятно, что при новом запуске этого же алгоритма можно получить совсем другие числа. Так, если в блоке 2 переменной A присвоить значение 20, то алгоритм выдаст в блоке 5 константы 20, 80, 400.

Разветвляющиеся алгоритмы

На практике алгоритмы линейной структуры встречается крайне редко. Чаще необходимо организовать процесс, который в зависимости от каких-либо условий проходит по той либо иной ветви алгоритма. Такой алгоритм называется разветвляющимся.

В блок-схемах ветвление начинается на выходах элемента "Решение", с помощью которого в алгоритме выполняется проверка какого-либо условия. Количество ветвей тем больше, чем больше проверяемых условий.

Для пояснения рассмотрим решение задачи нахождения значения функции z = y/x.

На первый взгляд представляется, что алгоритм решения этой задачи имеет линейную структуру. Однако, если учесть, что делить на нуль нельзя из-за переполнения ячейки, то, во-первых, нужно из вычислений исключить вариант x = 0 и, во-вторых, проинформировать пользователя алгоритма о возникшей ошибке. Если этого не сделать, то при вычислениях может возникнуть аварийный выход до получения результата. В профессиональной практике аварийные завершения крайне нежелательны.т. к. к этому моменту уже может быть накоплено определенное количество результатов, которые окажутся необработанными и попросту пропадут. Можно привести другие примеры, когда аварийный останов компьютера может повлечь куда более серьезные последствия.

Решение задачи представлено блок-схемой рис. 18.

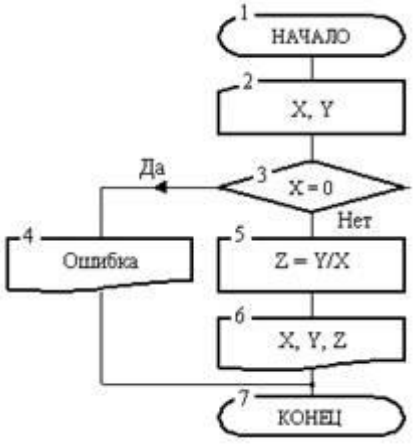


Рис. 18. Разветвляющийся алгоритм

Она состоит из 7 блоков. После начала работы алгоритм в блоке 2 требует ввода аргументов X и Y. Затем в блоке 3 производится проверка условия X = 0. Здесь автомат проверяет равна ли нули константа, введенная в ячейку с адресом X. Результатом такой проверки является ответ "Да" или "Нет". В зависимости от этого ответа выполнение алгоритма пойдет по одной или другой ветви. Если результат проверки окажется отрицательным, то на x можно делить и управление передается блоку 4.

В блоке 4 будет получен результат Z, затем в блоке б значения всех трех переменных будут отпечатаны и в блоке 7 алгоритм закончит работу. Если же ответ окажется положительным, то управление будет передано блоку 4. Выполняя команду блока 4, автомат выведет сообщение "Ошибка" и затем закончит работу в том же блоке 7.

Циклические алгоритмы

Часто при решении задач приходится повторять выполнение операций по одним и тем же зависимостям при различных значениях входящих в них переменных и производить многократный проход по одним и тем же участкам алгоритма. Такие участки называются циклами . Алгоритмы, содержащие циклы, называется циклическими . Использование циклов существенно сокращает объем алгоритма.

Различают циклы с наперед известным и наперед неизвестным количеством проходов.

Пример 1. Рассмотрим пример алгоритма с циклом, имеющим наперед неизвестное количество проходов. Для этого решим следующую задачу. Указать наименьшее количество членов ряда натуральных чисел 1, 2, 3, ..., сумма которых больше числа К.Блок-схема алгоритма решения этой задачи состоит из восьми блоков.

После начала работы в блоке 2 вводится значение числа К. Далее в блоке 3 переменная i получает значение 1, т. е. значение, с которого начнется отсчет натуральных чисел. Переменная S, предназначенная для накопления сумма этих чисел, перед началом суммирования получает значение 0. После этого управление передается блоку 5.

В нем при выполнении команды S = S + i производится сложение содержимого ячеек S и i, а результат записывается в ячейку S. Поскольку до операции сложения было S = 0, i = 1, то после операции будет S = 1. При записи нового значения старое содержимое ячейки S (нуль) стирается, а на его место записывается число 1.

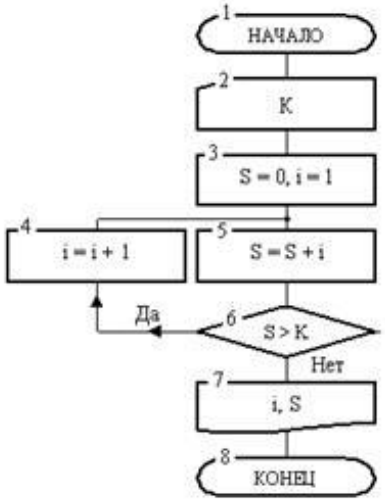


Рис. 19. Разветвляющийся алгоритм

Нужно обратить внимание на то, что если бы до этой операции в блоке 3 не была выполнена команда S = 0 (записать нуль в ячейку S), то при нахождении суммы S + 1 возникла бы ошибка, поскольку из ячейки S была бы извлечена константа, которая оказалась там после распределения памяти.

После суммирования первого члена последовательности в блоке 6 выполняется проверка условия о превышении суммы S заданного числа К.

Если условие 6 не выполнится, то производится переход к блоку 4, где при выполнении операции значение переменной увеличивается на 1 и становится равным 2. Теперь алгоритм вновь вернется к блоку 5 и к старому значении суммы добавит новый член 2. После этого сумма станет равной 3. В блоке б вновь проверяется условие получения требуемой суммы и т. д. Цепочка блоков 5-4 будет обрабатываться вновь и вновь до того момента, когда однажды при определенном значении переменной i, наконец, выполнится условие S> К, т. е. когда накапливаемая в таком цикле сумма впервые превысит заданное значение К. Переменная i, значение которой при очередном проходе цепочки этих блоков увеличивается на 1, играет роль счетчика этого цикла.

Далее производится переход к блоку 7, где отпечатается значение количества членов ряда (извлечено и отпечатано число из ячейки i, которое там хранится в момент выполнения условия), суммы S и в блоке 8 алгоритм закончит работу.

Пример 2. Теперь приведем пример алгоритма, содержащего цикл с наперед известным количеством проходов (повторений). Алгоритм решает задачу накопления

суммы положительных элементов одномерного массива Z длины N (под длиной массива понимается количество его элементов). Блок-схема алгоритма дана на рис. 20.

Вначале в блоке 2 производится ввод двух переменных N и Z. Первая из них представляет одну ячейку. В нее записывается одна константа – число, равное количеству элементов массива Z. Именно такое количество ячеек объединяет другая переменная – Z.

Следует подчеркнуть, что если бы ввод этих переменных в блоке 2 производился в противоположном порядке, то это привело бы к ошибке. Действительно, невозможно заполнить N ячеек массива Z, когда самое N еще не известно (оно будет введено позже Z). Далее в блоке 3 переменной S присвоено начальное значение 0. Это сделано для того, чтобы приготовить ячейку к дальнейшему накоплению необходимой суммы.

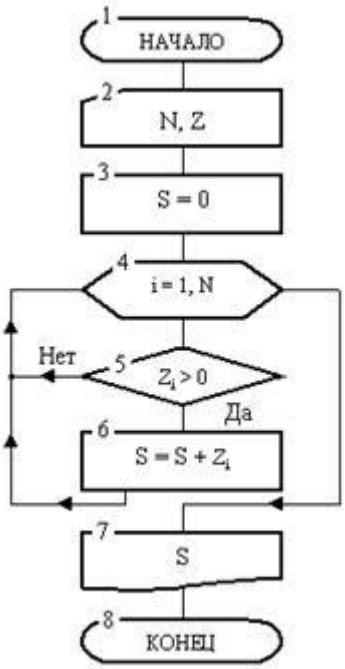


Рис. 20. Циклический алгоритм

Блоки 4-6 представляет собой сам цикл, в котором накапливается сумма.

Для того чтобы понять, как функционирует не только этот, а и любой другой цикл, обратимся к рис. 7, 8. На них показана общая структура цикла и его важнейшие параметры.

Как видно изрис. 21,цикл состоит из заголовка и тела. Всякий цикл обязательно имеет свой счетчик.

На рис. 22, где показана структура и параметры заголовка цикла, роль такого счетчика выполняет переменная i. Внутри заголовка после счетчика и символа "=" через запятую указывает начальное и конечное значения счетчика и шаг его изменения (на рис. 8 их роль выполняют переменные j, k, l соответственно). Если значение шага l = l, то его можно не указывать.

Сначала производится вход в цикл. После этого начинается его выполнение.

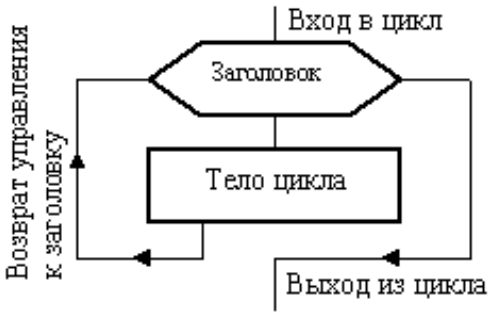


Рис. 21. Структура цикла

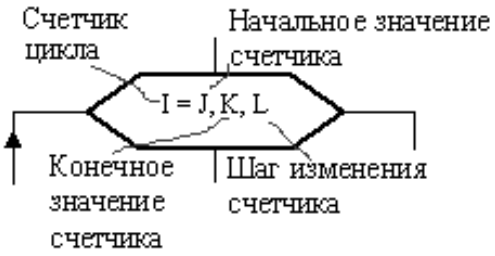


Рис. 22. Структура заголовка цикла

Внутри заголовка счетчику первоначально присваивается значение i = j. Затем выполняется блоки, образующие тело цикла. Обработка блоков внутри цикла производится по часовой стрелке. В результате после первого выполнения тела цикла управление вновь передается заголовку. Здесь к текущему значению счетчика добавится шаг. Теперь, если новое значение счетчика не вышло за свои пределы (т. е. не стало больше своего конечного значения при положительном шаге или меньше конечного значения – при отрицательном шаге), то снова выполняется тело цикла, вновь после возврата к заголовку к счетчику добавляется шаг. Так цикл будет выполняться до тех пор, пока значение счетчика однажды не выйдет за предписанный предел. Как только такой предел будет преодолен, произойдет выход из цикла и управление будет передано блоку, который следует сразу за циклом.

Вернемся к блок-схеме рис. 20. Заголовок ее цикла представлен блоком 4. Роль счетчика цикла играет переменная i, которая должна в цикле изменяться от 1 до N. Поскольку шаг явно не указан, то по умолчанию он подразумевается равным 1. Тело цикла образуют блоки 5 и 6.

Сразу после входа в цикл переменная i примет начальное значение i = 1. Далее в блоке 5 выполняется проверка положительности первого элемента массива Z (т. к. i = 1). Если этот элемент действительно положителен, то в блоке б он будет добавлен к переменной S, после чего выполняется возврат к заголовку цикла. Если этот элемент не положителен (т. е. нуль или отрицательный), то будет выполнен переход сразу к заголовку цикла, минуя блок суммирования 6.

На втором круге цикла счетчик i в заголовке увеличится на 1 и станет равным 2. Теперь, при новом выполнении тела цикла, в блоке 5 проверяется на положительность второй элемент массива Z и, если он положителен, то добавляется в сумму и т. д. Последний раз тело цикла выполнится при i = N. При этом значении счетчика проверяется последний элемент массива. Наконец, в заголовке цикла i примет значение N+1. Это значение выходит за предписанный предел, следовательно, произойдет выход из цикла и управление перейдет блоку 7. В этом блоке выводится накопленная сумма и алгоритм закончит работу.

Алгоритмы со структурами вложенных циклов

Нередко при алгоритмическом решении задачи возникает необходимость создания цикла, содержащего в своем теле другой цикл. Такие вложенные друг в друга циклы относятся к структурам вложенных циклов . Порядок вложенности циклов, когда в теле внутреннего цикла содержатся другие циклы, может быть достаточно большим. Этот порядок определяется методом, с помощью которого достигается решение поставленной задачи. Так, при обработке одномерных массивов, как правило, удается построить алгоритмическую схему без вложения циклов. Однако в ряде случаев при решении таких задач без вложенных циклов не обойтись.

Отметим, что все вложенные друг в друга циклы, включая наружный, должны иметь счетчики с различными именами. Вне этих циклов счетчики могут быть использованы как обычные переменные или как счетчики других циклов.

Пример 1. Рассмотрим задачу сортировки одномерного массива Z длины N. Отсортировать массив – значит расположить его элементы в порядке роста или убывания.

Опишем метод сортировки массива в порядке роста. Сначала выполняется проход по массиву с целью определения в нем наименьшего элемента. Затем производится перестановка этого элемента с первым. Далее совершается второй проход по массиву, начиная со второго элемента. Найденный наименьший элемент переставляется со вторым и т. д. После (N-1)-го прохода с выполнением названных операций массив окажется отсортированным.

Блок-схема этого алгоритма сортировки показана на рис. 23. Она включает 12 блоков. После начала работы в блоке 2 переменная N и массив Z заполняются константами. Затем в блоке 3 проверяется условие о том, нужно ли сортировать массив.

Это сводится к установлению факта наличия в массиве нескольких элементов, т. к. массив из одного элемента всегда отсортирован. Если этот факт установлен, то алгоритм приступает к сортировке. Процедура сортировки выполняется в цикле, объединяющем блоки 4-10. В теле этого цикла содержится другой цикл, который образован блоками 6-8. Его назначение станет ясно из дальнейшего разбора алгоритма.

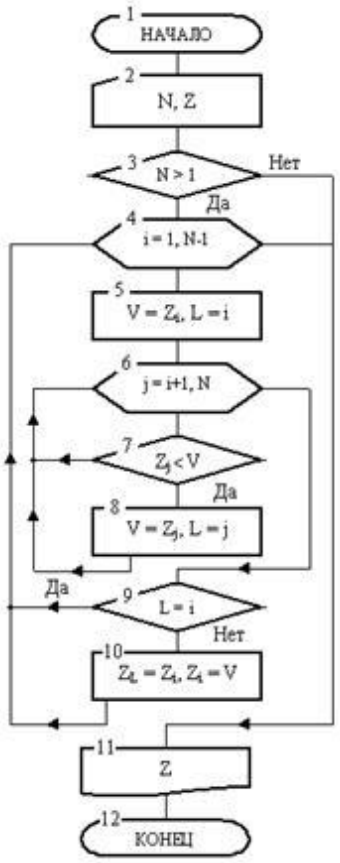


Рис. 23. Алгоритм сортировки массива

После входа в наружный цикл его счетчик i примет значение 1, что в рамках нашего метода подразумевает первый проход по массиву.

Далее будут выполнены блоки 5-10, составляющие тело наружного цикла. В блоке 5 размещены две вспомогательные переменные V и L. Первая из них

предназначена для фиксирования наименьшего элемента, а вторая – для запоминания его индекса. Так как i = 1, то при первом проходе в блоке 5 V примет значение первого элемента, а L значение 1. Затем во внутреннем цикле, образованном блоками 6-8, где его счетчик j будет изменяться от 2 до N, последовательно проводится сравнение соответствующих элементов массива Z со значением переменной V. При этом всякий раз, как будет найден меньший чем v элемент, значение V будет заменено на значение этого элемента, а в переменной L будет зафиксирован его индекс. Понятно, что после выполнения внутреннего цикла в переменной V будет содержаться значение, равное наименьшему элементу, а в L – индекс этого элемента. В блоке 9 далее проверяется, не является ли наименьший элемент первым элементом массива. Если это не так, то в блоке 10 на место наименьшего элемента (его номер L) запишется первый (т. к. при первом проходе L =1), а на место первого элемента – наименьший (он равен V). После этого произойдет возврат управления к заголовку наружного цикла блоку 4. В нем значение счетчика станет равным i = 2.

Затем вновь выполняется его тело, но уже для нового значения счетчика i. Теперь с помощью блоков 5-10 отыскивается наименьший элемент массива начиная с номера 2. Затем в блоках 9-10 он займет второе место в массиве и т. д. Когда тело наружного цикла выполнится (N-1), раз массив будет отсортирован.

В блоке 12 отсортированный массив будет выведен и в блоке 13 алгоритм окончит работу.

Алгоритмы со структурами вложенных циклов часто используют при решении задач обработки двумерных массивов. В таких алгоритмах счетчики циклов используются для манипуляции с индексами массивов.

Пример 2. Дан двумерный квадратный массив Z, состоящий из N строк и N столбцов. Необходимо найти среднее арифметическое S его отрицательных элементов и заменить положительные элементы побочной диагонали массива средним арифметическим S.

Блок-схема алгоритма показана на рис. 24. Она состоит из 13 блоков. В блоке 2 переменная N и весь массив Z заполняются константами. В блоке 3 рабочие переменные S и K получает значение нуль. Переменная S сначала будет играть роль сумматора отрицательных элементов массива, затем после накопления суммы она примет значение среднего арифметического. Переменная K нужна для подсчета количества отрицательных элементов массива.

В блоках 4-7 выполняется накопление суммы отрицательных элементов массива.

Эти блоки образует два вложенных цикла, причем внутренний цикл со счетчиком j является телом наружного цикла со счетчиком i. Проанализируем работу этой структуры.

После входа в наружный цикл в блоке 4 переменная i примет значение i = 1. Далее будет выполнено его тело (блоки 5-7), которое, в свою очередь, также является циклом. После входа во внутренний цикл в блоке 5 переменная j примет значение j = 1. Затем в блоке 6 проверяется на отрицательность элемент массива Z, расположенный в первой строке и первом столбце, т. к. i = 1 и j = 1.

Если он окажется отрицательным, то в блоке 7 переменная K увеличится на 1, а к S добавляется значение этого элемента. После этого выполняется возврат к блоку 5, т. е. к заголовку внутреннего цикла. Здесь j увеличится на 1, станет равной j = 2 и управление перейдет к блоку 6. В нем проверяется элемент, стоящий все в той же первой строке, но во втором столбце (i = 1, j = 2).

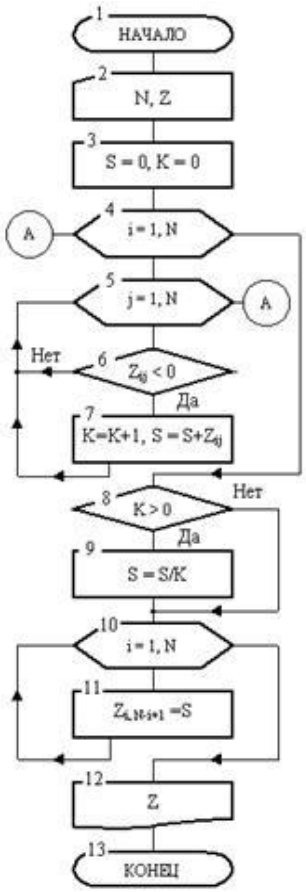


Рис. 24. Блок-схема алгоритма

Если он окажется отрицательным, то К снова увеличится на 1, а к накопленному к этому времени S добавляется значение этого элемента и т.д. Когда полностью выполнится внутренний цикл, т. е. переменная j "пробежит" от 1 до N, в переменную S накопится сумма всех отрицательных элементов первой строки массива, а в К – их количество. Теперь управление передается к блоку 4 заголовка наружного цикла, где i станет равной i = 2. Снова будет отработано его тело, т. е. цикл 5-7. При этом будет найдена уже сумма отрицательных элементов первых двух строк массива, а в К сохранится количество этих элементов. Когда выполнится весь наружный цикл, в S будет константа, равная сумме отрицательных элементов всего массива, а в К – их количество. Теперь управление перейдет к блоку 8. Если окажется, что в массиве есть отрицательные элементы (К>0), то в блоке 9 вычисляется среднее арифметическое как отношение суммы элементов к их количеству. Результат помещается а ту же переменную S. Отметим, что если бы блок 8 проверки отсутствовал, то при К = 0 (в массиве нет ни одного отрицательного элемента) в блоке 9 из-за деления на нуль возникла бы ошибка. Эта ошибка повлекла бы аварийное завершение вычислений до окончания работы алгоритма.

Далее выполняется блоки 10-11, которые также образует цикл. В нем производится замена элементов побочной диагонали на среднее арифметическое S (побочной диагональю является прямолинейная цепочка ячеек в диапазоне от нижнего левого угла до верхнего правого угла массива). Обратите внимание, на то что переменная i, которая использовалась ранее, в целях экономии памяти применяется вновь.

Проследим работу этого цикла. После входа в блок 10 счетчик примет значение i = 1. Затем в блоке 11 при этом значении будет вычислен индекс столбца элемента N – 1 + i = N. Таким образом, элемент с индексами (1, N) станет равным S. На втором круге

цикла i увеличится на 1 и станет i = 2. Нетрудно видеть, что теперь элемент (2, N-1) станет равным S и т. д. На последнем круге цикла элемент (N, 1) получит значение S, что завершит изменение значений всех элементов побочной диагонали на среднее арифметическое S.

Наконец, в блоке 12 измененный массив будет выведен и в блоке 13 алгоритм закончит работу.

Вспомогательные алгоритмы

Вспомогательный алгоритм является аналогом языковой подпрограммы. Он имеет имя и может иметь параметры, которые называются формальными параметрами . Имя служит для того. чтобы отличить его от других алгоритмов, а формальные параметры, которые напоминают переменные математических функций, выполняют роль входных и выходных параметров.

Формальные параметры должны быть выбраны таким образом, чтобы ими был исчерпан весь набор необходимых входных и выходных величин. Нередко один и тот же параметр может оказаться входным и выходным одновременно. Например, на вход такого алгоритма может быть подан массив для обработки, а на выходе процедуры он может предстать в измененном виде как выходной параметр.

Среди вспомогательных алгоритмов различают процедуры и функции .

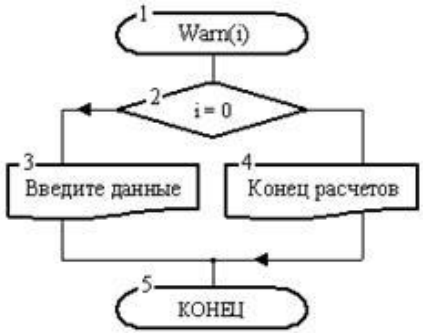


Рис. 25. Процедура Warn

Первый блок схемы в отличие от ранее рассмотренных примеров, где этот блок имел наименование “Начало”, включает имя процедуры Warn и один формальный параметр i. С помощью этого имени в алгоритме рис. 25 выполняется обращение именно к этой процедуре.

Из схемы видно, что если на вход процедуры Warn подать i = 0, то она в блоке 3 выдаст сообщение "Введите данные". При любом другом i будет выведено сообщение "Конец расчетов". Этим исчерпываются возможности процедура Warn.

На рис.26 дана схема головного алгоритма (первый блок имеет наименование "Начало"). Этот алгоритм в блоках 2 и 8 обращается к процедуре Warn.

Опишем последовательность и механизм обработки данных, которые предписаны алгоритмами.

Выполнение алгоритмических действий всегда начинаются с головного алгоритма. Поэтому сначала будет выполнен блок 1 схемы рис. 26. Далее в блоке 2 головной алгоритм выполняет обращение к процедуре Warn при конкретном значении ее аргумента (0). Это конкретное значение называется фактическим параметром процедуры.

Теперь управление временно переходит в алгоритм рис. 25 процедуры Warn. Здесь и далее по всей процедуре Warn формальный параметр i заменяется на фактический параметр 0 (нуль) всюду, где он встречается.

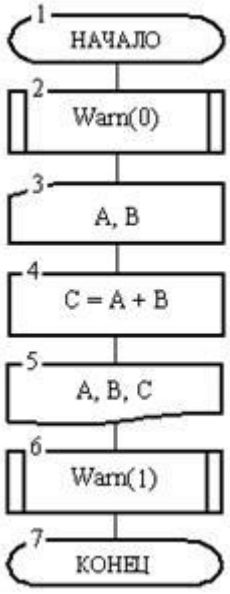


Рис. 26. Головной алгоритм

Далее обрабатывается блок 2 процедуры, где с учетом сказанного проверяется условие 0 = 0. Результатом проверки станет перевод управления к блоку 3, в котором выводится сообщение "Введите данные". На этом процедура заканчивается и управление вновь передается в головной алгоритм к блоку 3.

Далее в блоках 3-5 алгоритма рис. 26 выполняются уже понятные действия по вводу, суммированию и выводу переменных. Затем управление передается в блок б, который содержит новое обращение к процедуре Warn с фактическим параметром 1.

Снова управление переключается на схему рис. 25, где вместо формального параметра i всюду записывается фактический параметр – константа 1. Поскольку в блоке 2 условие 1 = 0 не выполнится, то будет выполнен блок 4 и алгоритм выведет сообщение "Конец расчетов". После этого управление возвращается в головной алгоритм к блоку 7, где и будет окончательно завершен алгоритмический процесс.

Внешне такой процесс может выглядеть примерно так. На экран выводится сообщение "Введите данные" и компьютер переходит в режим ожидания ввода двух констант с клавиатуры. Затем после их ввода на экране появляется три константы и надпись "Конец работы". На первый взгляд может показаться, что процедуры лишь усложняют решение задачи. Действительно, рассмотренную здесь задачу проще решить одним алгоритмом, не прибегая к составление процедуры. Однако при составлении алгоритма решения сложной задачи очень быстро становится ясно, что без использования процедур обойтись просто невозможно. На практике при решением серьезных алгоритмических задач часто одному программисту не под силу выполнить весь объем работ. Поэтому над ее решением работает обычно коллектив программистов под руководством координатора. Образно говоря, координатор здесь работает как головной алгоритм, а его программисты как процедуры. При этом каждый программист (часто независимо от других) получает от координатора задание по составление процедур определенного назначения. В результате такой организации работы задача получает разрешение.

Декомпозиция алгоритма

Под декомпозицией алгоритма понимают разложение его общей алгоритмической схемы на вспомогательные алгоритмы (процедуры и функции) и головной алгоритм. Напомним, такая задача ставится перед студентом при выполнении курсовой или контрольной работы. Одним из условий, которое должно быть обязательно выполнено,

является наличие в работе хотя бы одной процедуры или функции (кроме того, работа должна содержать текст описания всех процедур и головного алгоритма).

Метод, при помощи которого обычно выполняется декомпозиция, достаточно прост. Сначала вычленяют основные этапы предстоящей работы. Наиболее сложные этапы оформляет в процедуры или функции верхнего уровня. Затем, если необходимо, такие этапы делят на этапы более низкого уровня. Наиболее сложные из них также оформляют процедурами или функциями и т. д. Следуя методу "от сложного к простому", в конечном итоге достигают решения поставленной задачи.

2. Языки программирования.

В настоящее время в мире существует несколько сотен реально используемых языков программирования, для каждого из которых существует своя область применения. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования - чем меньше детализация, тем выше уровень языка. По этому критерию можно выделить следующие уровни языков программирования:

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

Машинные языки и машинно-ориентированные языки — это языки низкого уровня, требующие указания мелких деталей процесса обработки данных.

Языки же высокого уровня имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на:

- *алгоритмические* (Basic, Pascal, C и др.), которые предназначены для однозначного описания алгоритмов;
- *логические* (Prolog, Lisp и др.), которые ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи с тем, чтобы решение следовало из составленного описания.
- *объектно-ориентированные* (C++, Java и др.), в основе которых лежит понятие объекта, сочетающего в себе данные и действия над нами. Описание действительности в форме системы взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

3.Предпрограммная подготовка задачи.

На ЭВМ могут решаться задачи различного характера, например: научно-инженерные; разработки системного программного обеспечения; управления производственными процессами и т. д. В процессе подготовки и решения на ЭВМ научно-инженерных задач можно выделить следующие *этапы*:

1. постановка задачи;
2. формирование математической модели задачи;
3. выбор и обоснование метода решения;
4. алгоритмизация вычислительного процесса;
5. составление программы;
6. отладка и тестирование программы;
7. решение задачи на ЭВМ и анализ результатов.

В задачах другого класса некоторые этапы могут отсутствовать, например, в задачах разработки системного программного обеспечения отсутствует математическое описание. Перечисленные этапы связаны друг с другом. Например, анализ результатов может показать необходимость внесения изменений в программу; алгоритм или даже в

постановку задачи. Для уменьшения числа подобных изменений необходимо на каждом этапе по возможности учитывать требования, предъявляемые последующими этапами. В некоторых случаях связь между различными этапами, например, между постановкой задачи и выбором метода решения, между составлением алгоритма и программированием, может быть настолько тесной, что разделение их становится затруднительным.

Постановка задачи - этап словесной формулировки, определяющий цель решения, исходные данные, основные закономерности, условия и ограничения применения этих закономерностей. Анализируются характер и сущность всех величин, используемых в задаче, и определяются условия, при которых она решается. Корректность постановки задачи является важным моментом, так как от нее в значительной степени зависят другие этапы. Постановка задачи должна отвечать следующим требованиям:

- четкая формулировка цели с указанием вида и характеристик конечных результатов;
- представление значений и размерностей исходных данных;
- определение всех возможных вариантов решения, условий выбора каждого;
- обозначения границы применимости и действия в случае выхода за них.

Корректность постановки задачи является важным моментом, так как от нее в значительной степени зависят и другие этапы.

Формирование математической модели задачи - этап перевода словесной постановки задачи в совокупность математических зависимостей, описывающих исходные данные и вычисления промежуточных и конечных результатов.

Математическая модель формируется с определенной точностью, допущениями и ограничениями. При этом в зависимости от специфики решаемой задачи могут быть использованы различные разделы математики и других дисциплин.

Математическая модель должна удовлетворять по крайней мере двум требованиям: реалистичности и реализуемости. Под *реалистичностью* понимается правильное отражение моделью наиболее существенных *черт* исследуемого явления.

Реализуемость достигается *разумной* абстракцией, отвлечением от второстепенных деталей, чтобы свести задачу к проблеме с известным решением. Условием реализуемости является возможность практического выполнения необходимых вычислений за отведенное время при доступных затратах требуемых ресурсов.

Полученная математическая модель должна отвечать следующим требованиям:

- вначале составляется модель исходных данных, затем - расчетные зависимости;
- в модели исходных данных не изменяются размерности данных и не используются никакие математические операции;
- обозначение всех входящих в зависимости величин именами, определяющими их суть;
- указание размерностей всех используемых величин для контроля и дальнейшей модернизации решения;

Выбор и обоснование метода решения - этап разработки или выбора из уже имеющихся метода решения, в том числе выбор стандартных структур вычислительных процессов. Критерии выбора определяются математической моделью решения, требованиями к универсальности метода и точности результата, ограничениями технического и программного обеспечении. При обосновании выбора метода необходимо учитывать различные факторы и условия, в том числе точность вычислений, время решения задачи на ЭВМ, требуемый объем памяти и другие. Следует указать альтернативные методы и аргументы сделанного выбора.

Алгоритмизация вычислительного процесса - этап разработки совокупности предписаний, однозначно определяющих последовательность преобразования исходных данных в конечные результаты. На данном этапе составляется алгоритм решения задачи согласно действиям, задаваемым выбранным методом решения. Процесс обработки данных разбивается на отдельные относительно самостоятельные блоки, и устанавливается последовательность выполнения блоков. Разрабатывается блок-схема алгоритма.

Составление программы. При составлении программы алгоритм решения задачи переводится на конкретный язык программирования. Для программирования обычно используются языки высокого уровня, поэтому составленная программа требует перевода ее на машинный язык ЭВМ. После такого перевода выполняется уже соответствующая машинная программа.

1. Программа должна быть *универсальной*, то есть не зависящей от конкретного набора данных. Универсальная программа должна уметь обрабатывать ошибки, которые могут возникнуть в процессе обработки информации.
2. Вместо констант лучше использовать *переменные*. Если в программе используются константы, то при их изменении нужно изменять в исходной программе каждый оператор, содержащий прежнюю константу. В программе следует предусмотреть контроль вводимых данных.
3. Некоторые простые приемы позволяют повыситьэффективность программы. К таким приемам относится:

- использование операции умножения вместо возведения в степень если некоторое арифметическое выражение встречается в вычислениях несколько раз, то его следует вычислить заранее и хранить в памяти ЭВМ, а по мере необходимости использовать;
- при организации циклов в качестве границ индексов использовать переменные, а не выражения, которые вычислялись бы при каждом прохождении цикла;
- особое внимание обратить на организацию циклов, убрав из них все повторяющиеся с одинаковыми данными вычисления и выполняя их до входа в цикл.

4. Программа должна содержать *комментарии*, позволяющие легко проследить за логической взаимосвязью и функциями отдельных ее частей.

При написании программы следует структурировать ее текст так, чтобы она хорошо читалась. В программе должно быть хорошо видно, где начинается и где заканчивается цикл.

Отладка программы – процесс выявления и исправления синтаксических и логических ошибок в программе. Суть отладки заключается в том, что выбирается некоторый набор исходных данных, называемый тестовым набором, и задача с этим набором решается дважды: один раз – исполнением программы, второй раз – каким-либо иным способом, исходя из условия задачи, так сказать, «вручную». При совпадении результатов алгоритм считается верным. В качестве тестового набора можно выбрать любые данные, которые позволяют: обеспечить проверку выполнения всех операций алгоритма; свести количество вычислений к минимуму.

В ходе синтаксического контроля программы транслятором выявляются конструкции и сочетания символов, недопустимые с точки зрения правил их построения или написания, принятых в данном языке. Сообщения об ошибках ЭВМ выдает программисту, вид и форма выдачи подобных сообщений зависят от вида языка и версии используемого транслятора.

После устранения синтаксических ошибок проверяется логика работы программы в процессе ее выполнения с конкретными исходными данными. Для этого используются специальные методы, например, в программе выбираются контрольные точки, для которых вручную рассчитываются промежуточные результаты. Эти результаты сверяются со значениями, получаемыми ЭВМ в данных точках при выполнении

отлаживаемой программы. Для поиска ошибок могут быть использованы отладчики, выполняющие специальные действия на этапе отладки, например, удаление, замена или вставка отдельных операторов или целых фрагментов программы, вывод или изменение значений заданных переменных.

Тестирование - это испытание, проверка правильности работы программы в целом, либо её составных частей.

Отладка и тестирование (англ. test - испытание) - это два четко различимых и непохожих друг на друга этапа:

- при отладке происходит локализация и устранение синтаксических ошибок и явных ошибок кодирования;
- в процессе же тестирования проверяется работоспособность программы, не содержащей явных ошибок.

Тестирование устанавливает факт наличия ошибок, а отладка выясняет ее причину.

В современных программных системах отладка осуществляется часто с использованием специальных программных средств, называемых отладчиками. Эти средства позволяют исследовать внутреннее поведение программы. Программа-отладчик обычно обеспечивает следующие возможности:

- пошаговое исполнение программы с остановкой после каждой команды (оператора);
- просмотр текущего значения любой переменной или нахождение значения любого выражения, в том числе, с использованием стандартных функций; при необходимости можно установить новое значение переменной;
- установку в программе "контрольных точек" и др.

При отладке программ важно помнить следующее:

- в начале процесса отладки надо использовать простые тестовые данные;
- возникающие затруднения следует четко разделять и устранять строго поочередно;
- не нужно считать причиной ошибок машину, так как современные машины и трансляторы обладают чрезвычайно высокой надежностью.

Как бы ни была тщательно отлажена программа, решающим этапом, устанавливающим ее пригодность для работы, является контроль программы по результатам ее выполнения на системе тестов. Программу условно можно считать правильной, если её запуск для выбранной системы тестовых исходных данных во всех случаях дает правильные результаты.

Тестирование может показать лишь наличие ошибок, но не их отсутствие. Нередки случаи, когда новые входные данные вызывают "отказ" или получение неверных результатов работы программы, которая считалась полностью отлаженной.

Для реализации метода тестов должны быть изготовлены или заранее известны эталонные результаты. Вычислять эталонные результаты нужно обязательно до, а не после получения машинных результатов. В противном случае имеется опасность невольной подгонки вычисляемых значений под желаемые, полученные ранее на машине.

Тестовые данные должны обеспечить проверку всех возможных условий возникновения ошибок:

- должна быть испытана каждая ветвь алгоритма;
- очередной тестовый прогон должен контролировать нечто такое, что не было проверено на предыдущих прогонах;
- первый тест должен быть максимально прост, чтобы проверить, работает ли программа вообще;
- арифметические операции в тестах должны предельно упрощаться для уменьшения объема вычислений;

- количества элементов последовательностей, точность для итерационных вычислений, количество проходов цикла в тестовых примерах должны задаваться из соображений сокращения объема вычислений;
- тестирование должно быть целенаправленным, систематизированным;
- усложнение тестовых данных должно происходить постепенно.

Процесс тестирования можно разделить на три этапа.

1. *Проверка в нормальных условиях*. Предполагает тестирование на основе данных, которые характерны для реальных условий функционирования программы.

2. *Проверка в экстремальных условиях*. Тестовые данные включают граничные значения области изменения входных переменных, которые должны восприниматься программой как правильные данные.

3. *Проверка в исключительных ситуациях*. Проводится с использованием данных, значения которых лежат за пределами допустимой области изменений.

Решение задачи на ЭВМ и анализ результатов. После отладки программы ее можно использовать для решения прикладной задачи. При этом выполняется многократное решение задачи на ЭВМ для различных наборов исходных данных. Получаемые результаты интерпретируются и анализируются специалистом или пользователем, поставившим задачу.

Разработанная программа длительного использования устанавливается на ЭВМ. К программе прилагается документация, включая инструкцию для пользователя.

Вопросы:

1. Что называется алгоритмом?
2. Какие вы знаете свойства алглритма?
3. Какие основные алгоритмические структуры существуют?
4. Что такое константа?
5. Что такое переменная?
6. Дайте понятие массива данных?
7. В чем суть вспомогательных алгоритмов?
8. В чем заключается декомпозиция алгоритма?
9. Какие языки программирования существуют?
10. Перечислите этапы подготовки и решения задачи на ЭВМ?

Лекция №6.

Тема: «Базы данных. Программы управления базами данных»

План:

1. Основные понятия Базы данных, СУБД.
2. Классификация СУБД.
3. Реляционные базы данных
4. Проектирование БД.
5. Система управлениябазамиданныхMicrosoftOfficeAccess.

1.Основные понятия Базы данных, СУБД

База данных — это организованная структура, предназначенная для хранения информации.

Для взаимодействия пользователя с базами данных используют системы управления данными (СУБД).

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

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

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

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

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

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

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

Все СУБД поддерживают в той или иной форме четыре основных операции:

- добавить в базу данных одну или несколько записей;
- удалить из базы данных одну или несколько записей;
- найти в базе данных одну или несколько записей, удовлетворяющих заданному условию;
- обновить в базе данных значение некоторых полей.

Принципы построения систем управления баз данных следуют из требований, которым должна удовлетворять организация баз данных:

- *Производительность и готовность.* Запросы от пользователя базой данных удовлетворяются с такой скоростью, которая требуется для использования данных. Пользователь быстро получает данные всякий раз, когда они ему необходимы.
- *Минимальные затраты.* Низкая стоимость хранения и использования данных, минимизация затрат на внесение изменений.
- *Простота и легкость использования.* Пользователи могут легко узнать и понять, какие данные имеются в их распоряжении. Доступ к данным должен быть простым, исключающим возможные ошибки со стороны пользователя.
- *Простота внесения изменений.* База данных может увеличиваться и изменяться без нарушения имеющихся способов использования данных.
- *Возможность поиска.* Пользователь базы данных может обращаться с самыми различными запросами по поводу хранимых в ней данных. Для реализации этого служит так называемый язык запросов.
- *Целостность.* Современные базы данных могут содержать данные, используемые многими пользователями. Очень важно, чтобы в процессе работы элементы данных и связи между ними не нарушались. Кроме того, аппаратные ошибки и различного рода случайные сбои не должны приводить к необратимым потерям данных. Значит, система управления данными должна содержать механизм восстановления данных.
- *Безопасность и секретность.* Под безопасностью данных понимают защиту данных от случайного или преднамеренного доступа к ним лиц, не имеющих на это права, от неавторизированной модификации (изменения) данных или их разрушения. Секретность определяется как право отдельных лиц или организаций решать, когда, как какое количество информации может быть передано другим лицам или организациям.

Большинство СУБД поддерживают, кроме того, механизм *связей* между различными файлами, входящих в базу. Например,

Реляционная - модель данных строится по принципу взаимосвязанных таблиц;

Иерархическая - один тип объекта является главным, все нижележащие - подчиненными;

Сетевая - любой тип данных одновременно может быть главным и подчиненным.

2.Классификация СУБД

По модели данных

По типу управляемой базы данных СУБД разделяются на:

- Иерархические
- Сетевые
- Реляционные
- Объектно-реляционные
- Объектно-ориентированные

По архитектуре организации хранения данных

- локальные СУБД (все части локальной СУБД размещаются на одном компьютере)
- распределенные СУБД (части СУБД могут размещаться на двух и более компьютерах)

По способу доступа к БД

- Файл-серверные

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. Ядро СУБД располагается на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера, а недостатком — высокая загрузка локальной сети.

На данный момент файл-серверные СУБД считаются устаревшими.

Примеры: Microsoft Access, Paradox, dBase.

- Клиент-серверные

Такие СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера (см. *Клиент-сервер*). Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и по надобности его можно заменить другим. Недостаток клиент-серверных СУБД в самом факте существования сервера (что плохо для локальных программ — в них удобнее встраиваемые СУБД) и больших вычислительных ресурсах, потребляемых сервером.

Примеры: Firebird, Interbase, IBM DB2, MS SQL Server, Sybase, Oracle, PostgreSQL, MySQL, ЛИНТЕР.

- Встраиваемые

Встраиваемая СУБД — библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных (например, геоинформационные системы).

3.Реляционные базы данных

Базы данных с табличной формой организации называются реляционными БД.

В реляционных БД строка таблицы называется записью, а столбец — полем. В общем виде это выглядит так:

Каждое поле таблицы имеет имя.

Одна запись содержит информацию об одном объекте той реальной системы, модель которой представлена в таблице.

Поля — это различные характеристики (иногда говорят — атрибуты) объекта. Значения полей в одной строчке относятся к одному объекту. Разные поля отличаются именами.

Записи различаются значениями ключей.

Главным ключом в базах данных называют поле (или совокупность полей), значение которого не повторяется у разных записей.

С каждым полем связано еще одно очень важное свойство — тип поля.

Тип определяет множество значений, которые может принимать данное поле в различных записях.

В реляционных базах данных используются четыре основных типа полей: числовой, символьный, дата, логический.

Числовой тип имеют поля, значения которых могут быть только числами.

Символьный тип имеют поля, в которых будут храниться символьные последовательности (слова, тексты, коды и т.п.).

Тип «дата» имеют поля, содержащие календарные даты в форме «день/месяц/год».

Логический тип соответствует полю, которое может принимать всего два значения: «да» — «нет» или «истина» — «ложь».

От типа величины зависят те действия, которые можно с ней производить.

Например, с числовыми величинами можно выполнять арифметические операции, а с символьными и логическими — нельзя.

Существуют специальные приложения для создания баз данных такие, как:

Paradox

Oracle

dBase

FoxPro

OpenOffice.org Base

Microsoft Office Access ит.д.

Они постоянно развиваются, появляются новые версии этих приложений

4.Проектированиебазыданных

Режимыработысбазамиданных

Обычно сбазами данныхработают двекатегорииисполнителей.Перваякатегория—*проектировщики.*Ихзадачасостоитвразработкеструктурытаблицбазыданныхи согласованииеесзаказчиком.Крометаблицпроектировщикиразрабатываютидругие объектыбазыданных,предназначенные,соднойстороны,дляавтоматизацииработыс базой,асдругойстороны—дляограниченияфункциональныхвозможностейработыс базой (если это необходимо из соображений безопасности). Проектировщики не наполняют базу конкретными данными (заказчик может считать их конфиденциальными ине предоставлять посторонним лицам). Исключение составляет экспериментальное наполнениемодельнымиданныминаэтапеотладкиобъектовбазы.

Вторая категория исполнителей, работающих с базами данных, — *пользователи.* Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием. Вобщем случае пользователи неимеют средств доступа к управлению структурой базы — только к данным, да и то не ко всем, ак тем, работа с которыми предусмотрена на конкретном рабочем месте.

Соответственно, система управления базами данных имеет два режима работы: *проектировочный* и *пользовательский.* Первый режим предназначен для создания или изменения структуры базы и создания ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

При разработке Базы данных методически правильно начинать работу с карандашом и листом бумаги в руках, не используя компьютер. На данном этапе он просто не нужен. Неоптимальные решения и прямые ошибки, заложенные на этапе проектирования, впоследствии очень трудно устраняются, поэтому этот этап является основополагающим.

Разработка технического задания.

Техническое задание на проектирование базы данных должен предоставить заказчик. Однако для этого он должен владеть соответствующей терминологией и знать, хотя бы в общих чертах, технические возможности основных систем управления базами данных. К сожалению, на практике такое положение встречается не всегда. Поэтому обычно используют следующие подходы:

- Демонстрируют заказчику работу аналогичной базы данных, после чего согласовывают спецификацию отличий;

- Если аналога нет, выясняют круг задач и потребностей заказчика, после чего помогают ему подготовить техническое задание.

При подготовке технического задания составляют:

- список исходных данных, с которыми работает заказчик;
- список выходных данных, которые необходимы заказчику для управления структурой своего предприятия;
- список выходных данных, которые не являются необходимыми для заказчика, но которые он должен предоставлять в другие организации (вышестоящие структуры, в органы статистического учета, прочие административные и контролирующие организации).

При этом очень важно не ограничиваться взаимодействием с головным подразделением заказчика, а провести обсуждение со всеми службами и подразделениями, которые могут оказаться поставщиками данных в базу или их потребителями. Так, например, при подготовке базы данных для учета абитуриентов и студентов вуза, высшем учебном заведении, необходимо не только изучить документооборот ректората и всех деканатов, но и понять, что хотели бы получить от базы данных служб. Следует изучить работу подразделений, распределяющих учебную нагрузку преподавателей, отвечающих за распределение аудиторного фонда за проживание студентов в общежитии других. Врасчет должны приниматься и такие службы, как библиотека, отдел кадров и прочие. В любой момент может выясниться, например, что администрация библиотеки должна периодически поставлять кому-то отчеты, характеризующие читательскую активность студентов в зависимости от пола, возраста и социального положения. Кво зможным пожеланиям заказчика следует готовиться на этапе проектирования, до создания базы.

Разработка структуры базы данных.

Выяснив основную часть данных, которые

заказчикпотребляетилипоставляет,можноприступатьксозданиюструктурыбазы,то есть структурыееосновныхтаблиц.

1. Работа начинается с составления генерального списка полей — он может насчитыватьдесяткиидажесотнипозиций.

2. В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.

3. Далеераспределяютполягенеральногоспискапобазовымтаблицам.Наперв ом этапераспределениепроизводятпофункциональномупризнаку. Цель— обеспечить, чтобывводданныхводнутаблицупроизводился, повозможности, врамкаходного подразделения,аещелучше—наодномрабочемместе.

Наметив столько таблиц, сколько подразделений охватывает база данных, приступают к дальнейшему делению таблиц. Критерием необходимости деления являетсяфактмножественногоповтораданныхвсоседнихзаписях.

4. Вкаждойизтаблицнамечают*ключевое поле.*Вкачестветаковоговыбираютполе, данныевкоторомповторятьсянемогут.Например,длятаблицыданныхостудентах такимполемможетслужитьиндивидуальныйшифрстудента.Длятаблицы,вкоторой содержатсярасписаниязанятий,такогополяможноиненайти,ноегоможносоздать искусственнымкомбинированиемполей«Времязанятия» и«Номера аудитории». Эта комбинациянеповторима,таккакводнойаудиторииводноитожевремянепринято проводитьдваразличныхзанятия.Еслвтаблицевообщенетникакихполей,которыеможн обылобыиспользовать какключевые,всегдаможноввестидополнительноеполетипаСчетчик—ононеможет содержать повторяющихся данных по определению.

5. Спомощьюкарандашаибумагирасчерчиваютсвязимеждутаблицами.Такой чертеж называется *схемой данных.*

Существует несколько типов возможных связей между таблицами. Наиболее распространеннымиявляютсясвязи«одинкомногим»и«одинкодному».Связьмежду таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым, то есть на стороне «один» должно выступать ключевое поле, содержащее уникальные, неповторяющиеся значения. Значения на стороне«многие»могутповторяться.

Про подобные таблицы говорят, что они связаны *реляционными отношениями.* Соответственно, системыуправления, способныеработатьсосвязаннымитаблицами, называют *системами управления реляционными базами данных,* а схему данных в технической литературе могут называть *схемой реляционных отношений.*

6. Разработкой схемы данных заканчивается «бумажный» этап работы над техническим предложением. Эту схему можно согласоватьс заказчиком, послечего приступать к непосредственному созданию базы данных.

Следуетпомнить, чтопоходуразработкипроектазаказчикунепременнобудут приходитьвголовуновыеидеи.Навсехэтапахпроектированияонстремитсяохватить единойсистемойвсеновыеиновыеподразделенияислужбыпредприятия.Возможность гибкого исполнения его пожеланий во многом определяется квалификацией разработчика базы данных. Если схема данных составлена правильно, подключать к базе новые таблицынетрудно. Еслиструктурабазынерациональна, разработчикможетиспытать серьезныетрудностиивойтивпротиворечиясзаказчиком.

Противоречияисполнителясзаказчикомвсегдасвидетельствуютонедостаточной квалификацииисполнителя.Именнопоэтомуэтаппредварительногопроектирования базыданныхследуетсчитатьосновным. Отегоуспехазависит, насколькобазаданных станет удобной и будут ли с ней работать пользователи. Если отмечается, что пользователи базы «саботируют» ее эксплуатацию и предпочитают работать традиционнымиметодами,этоговоритнеонизкойквалификациипользователей,ао недостаточной квалификации разработчика базы.

Наэтомэтапезавершаетсяпредварительноепроектированиебазыданных,ина следующем этапе начинаетсяее непосредственнаяразработкаС этогомомента следует начатьработуссистемойуправлениябазамиданных.

6. Система управления базами данных MicrosoftOfficeAccess.

MSAccessявляется СУБД реляционного типа, в которой разумно сбалансированы все средства и возможности, типичных для современных СУБД. Реляционная база упрощает поиск, анализ, поддержку и защиту данных, поскольку они сохраняются в одном месте. Accessв переводе с английского означает «доступ».MSAccess— это функционально полная реляционная СУБД. Кроме того, MSAccessодна из самых мощных, гибких и простых в использовании СУБД. В ней можно создавать большинство приложений, не написав ни единой строки программы, но если нужно создать нечто очень сложное, то на этот случай MSAccessпредоставляет мощный язык программирования — VisualBasicApplication.

Популярность СУБД MicrosoftAccessобусловлена следующими причинами:

- доступность в изучении и понятность позволяют Accessявляться одной из лучших систем быстрого создания приложений управления базами данных;
- СУБД полностью русифицирована;
- интегрированность с пакетами MicrosoftOffice;
- визуальная технология позволяет постоянно видеть результаты своих действий и корректировать их; кроме того, работа с конструктором форм может существенно облегчить дальнейшее изучение таких систем программирования, как VisualBasicилиDelphi;
- широко и наглядно представлена справочная система;
- наличие большого набора «мастеров» по разработке объектов.

Характеристика некоторых основных объектов базы данных.

Окно базы данных выводится при открытии базы данных. Из него открывают таблицы, формы и другие объекты базы данных. Окно базы данных содержит следующие элементы:

Строка заголовка. Выводит имя открытой базы данных.

Кнопки. «Создать», «Открыть», «Конструктор» и т.д. Кнопки открывают объект в определенном окне или режиме.

Кнопки объектов. «Таблица», «Форма» и т.д. Кнопки объектов выводят список объектов, которые могут быть открыты или изменены.

Список объектов. Выводит список объектов, выбираемых пользователем. В нашем варианте список пока пуст.

Прежде чем начать непосредственную работу по разработке базы данных, остановимся на характеристиках некоторых основных объектов базы данных.

Таблица - это объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов). Обычно каждая таблица используется для хранения сведений по одному конкретному вопросу.

Форма - объект MicrosoftAccess, предназначенный, в основном, для ввода данных. В форме можно разместить элементы управления, применяемые для ввода, изображения и изменения данных в полях таблиц.

Запрос - объект, позволяющий получить нужные данные из одной или нескольких таблиц.

Отчет - объект базы данных MicrosoftAccess, предназначенный для печати данных.

Начинать работу следует с создания таблицы.

В таблице сохраняют записи, содержащие сведения определенного типа, например, список клиентов или опись товаров. Составной частью таблицы являются

поля.

Поле - это элемент таблицы, который содержит данные определенного рода, например, фамилию сотрудника. В режиме таблицы для представления поля используется столбец или ячейка, в этом случае имя поля является заголовком столбца таблицы.

Запись - полный набор данных об определенном объекте. В режиме таблицы запись изображается как строка.

Все перечисленные объекты можно создавать в режимах Мастер или Конструктор. В режиме Мастер предлагаются готовые образцы для использования, режим Конструктор позволяет разрабатывать собственные таблицы, формы, отчеты и т. д. Все изменения полей и типов данных возможны только в режиме Конструктор.

Вопросы:

1. Дайте определение базы данных?
2. Дайте определение СУБД?
3. Перечислите основные операции базы данных?
4. Какие требования нужно учитывать при организации базы данных?
5. На какие типы подразделяются базы данных по связям между файлами?
6. Дайте характеристику реляционной базы данных?
7. Последовательностьпроектированиябазы данных?
8. Основные возможности СУБДAccess?

Лекция №7.

Тема: «Технология обработки информации в электронных таблицах»

План:

1. Назначение и основные функции электронных таблиц.
2. Электронная таблица MicrosoftExcel. Основные понятия, применение

1. Назначение и основные функции электронных таблиц.

Сотни лет в деловой сфере при выполнении громоздких однотипных расчётов используются таблицы. С их помощью рассчитывается заработная плата, ведутся различные системы учёта материальных ценностей, просчитывается стоимость новых товаров и услуг, прогнозируется размер прибыли и т. д. Такие расчёты многие специалисты до конца прошлого века выполняли с помощью калькуляторов, вручную занося полученные результаты в соответствующие графы таблиц. Такая работа требовала больших временных затрат; на исправление незначительной ошибки, допущенной расчётчиком, уходили недели и даже месяцы.

Ситуация кардинально изменилась с появлением электронных таблиц, позволивших за счёт изменения исходных данных быстро решать большое количество типовых расчётных задач.

Электронные таблицы (табличный процессор) — это прикладная программа, предназначенная для организации табличных вычислений на компьютере.Электронная таблицаработает с большими массивами числовой информации. Она позволяет хранить в табличной форме не только большое количество исходных данных, результатов расчётов, но и математические соотношения между ними, значения которых автоматически пересчитываются по заданным формулам при изменении значений исходных данных.

Основные сферы применения ЭТ

- расчёт использования денежных средств в финансовых операциях
- статистическая обработка данных
- математическое моделирование процессов
- инженерные расчёты.

Основные функции электронных таблиц

- вычисления с участием данных, находящихся в таблице;
- поиск и сортировка информации;
- графическое отображение числовой информации из таблицы (построение графиков и диаграмм);
- статистический анализ данных.

ЗапускExcel: Пуск – ВсеПрограммы – MicrosoftOffice – MicrosoftExcel.

2.Электронная таблица MicrosoftExcel. Основные понятия, применение

СтруктураокнаMicrosoftExcel:

- строка заголовка
- строка меню;
- панель инструментов (содержит кнопки наиболее часто используемых команд);
- панель форматирования;
- поле имени (указывает имя выделенной ячейки);
- строка формул (служит для ввода и редактирования содержимого ячейки);
- рабочая область;
- вкладки листов;
- полосы прокрутки.

Документ в MicrosoftExcel называется ***рабочей книгой.*** Файлы рабочих книг имеют расширение ***.xls или .xlsx.***Рабочая книга состоит из листов (при запуске по умолчанию выводится три листа). Минимальный элемент ЭТ называется ***ячейкой.****Строки* в ЭТ обозначаются цифрами, *столбцы* – буквами латинского алфавита. ***Адрес ячейки*** указывается следующим образом:

- сначала указывают имя столбца (A, B, C, D, ….)
- затем указывают имя строки.

*Например:*A1, B2, C5, D11

Диапазон ячеек–это совокупность нескольких ячеек.

Например: C4:C9 – элементы столбца Cc 4-го по 9-й.

A3:D11 – элементы прямоугольного диапазона

Ввод данных – это запись в ячейки информации: текста, чисел, формул.

Ввод данных осуществляется двумя способами:

1. С помощью строки формул: для этогонеобходимо выделить ячейку**,** щёлкнуть в строке формул и ввести данные. При этом слева появятся кнопки:

X (Esc) – выход из режима редактирования без сохранения изменений;

V (Enter) – выход из режима редактирования с сохранением изменений.

2. Непосредственным способом: выделить ячейку и осуществить ввод данных (при этом вводимые в ячейку данные будут отображаться в строке формул).

Способы редактирования данных

- выделить ячейку и ввести новые данные (при этом ранее введённое содержимое ячейки будет утеряно);
- выполнить двойной щелчок по ячейке и внести необходимые изменения;
- выделить ячейку, нажать клавишу **F2** и внести нужные изменения;
- выделить ячейку, щёлкнуть мышью в строке формул и ввести изменения.

Вызов списка форматов выполняется с помощью команды контекстного меню *Формат ячеек.* Затем в диалоговом окне *Формат ячеек* необходимо открыть вкладку *Число* и выбрать необходимый числовой формат данных. Существуют также логические значения Истина и Ложь, используемые при операциях сравнения.

Примеры числовых форматов данных: *общий, числовой, денежный, финансовый, текстовый, процентный, дробный, дата, время* и др.

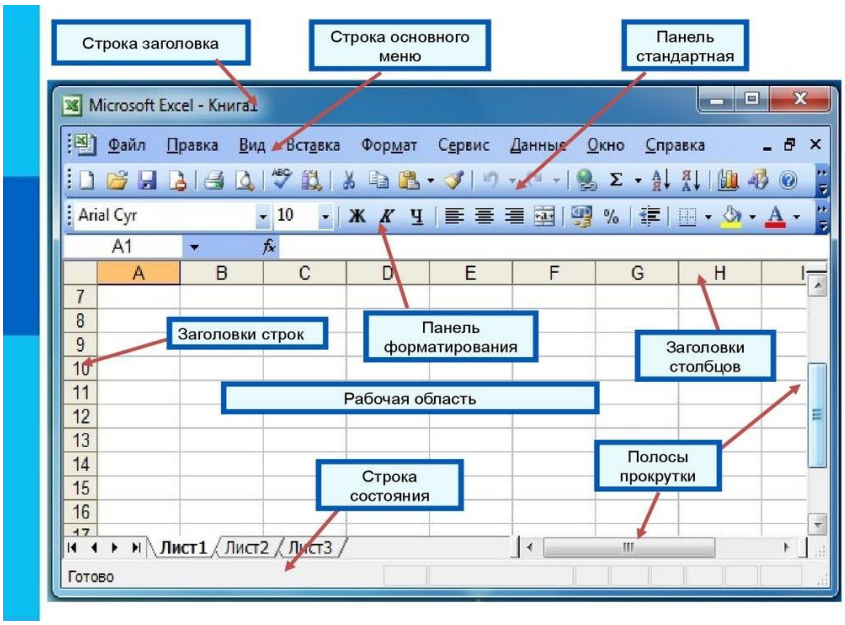


Рис.27

Использование формул в ЭТ

Формулы в MicrosoftExcel – это выражения, описывающие вычисления в ячейках.

Формулы вписываются в строку формул.

Написание формулы начинается с символа «=».

Формулы могут включать следующие компоненты:

- символ «=» (с него начинается написание формулы)
- операторы, т.е. инструкции для выполнения действий (+, - , *, / и т.д.)
- данные (числа или текст)
- функции
- ссылки на ячейки и диапазоны.

Примеры: = A1+B1;
= 0.2*D6;
= СУММ (A3:A5).

Вычисление формул

1. в строке формул написать знак «=»

2. ввести формулу, используя знаки арифметических операций (+, -,*,/,%), числа и имена ячеек

Пример.=2.5*A1

= 3.75/C2

= A3+C4

Операторы в MicrosoftExcel

1. ***текстовый оператор (& - объединение)***
2. ***адресные операторы (используются для указания ссылок на ячейки)***
3. ***арифметические операторы***
4. ***операторы сравнения***

Табл.3.

Символ оператора	Название оператора	Пример формулы	Результат
+	Сложение	=1,5+2,6	4,1
-	Вычитание	= 4,8-3,2	1,6
*	Умножение	= 0,5*8	4
/	Деление	= 6/5	1,2
%	Процент	= 40%	0,4
^	Возведение в степень	= 3^2	9

Табл. 4.

Символ оператора	Название оператора	Пример формулы	Результат
>	Больше	= 5>7	Ложь
<	Меньше	= 4<11	Истина
=	Равно	= 5=9	Ложь
<=	Меньше или равно	= 3<=2	Ложь
>=	Больше или равно	= 4>=4	Истина
<>	Не равно	= 5<>6	Истина

Приоритет операторов

1. Адресные операторы.
2. Оператор отрицания.
3. Процент.
4. Возведение в степень.
5. Умножение и деление.
6. Сложение и вычитание.
7. Текстовый оператор.

8. Операторы сравнения.

Использование встроенных функций в среде табличного процессора

Функции – это инструкции, которые вычисляют результат, обрабатывая аргументы.

Аргументы функции записывают после её имени в круглых скобках через точку с запятой.

Для вызова мастера функций необходимо выполнить команду *Вставка – Функция* либо щёлкнуть мышью по значку **f(x)** в панели инструментов.

Некоторые стандартные функции

Табл.5.

Название	Обозначение	Описание действия
Математические	SIN(x)	вычисление синуса угла
	COS(x)	вычисление косинуса угла
	ABS(x)	вычисление модуля числа
	СТЕПЕНЬ (число; степень)	вычисляет степень заданного числа
	СУММ (список аргументов)	вычисление суммы значений аргументов
	КОРЕНЬ (число)	вычисление квадратного корня из числа
Статистические	СРЗНАЧ (список аргументов)	Вычисление среднего арифметического значений аргументов
	МАКС (список аргументов)	Вычисление максимального значения среди аргументов
	МИН (список аргументов)	Вычисление минимального значения среди аргументов
Логическая функция	ЕСЛИ (логическое выражение; выражение 1; выражение 2)	Если логическое выражение истинно, то функция принимает значение выражения 1, иначе – значение выражения 2

Примеры

Табл.6.

Пример формулы	Результат
= СУММ(2, 4.5)	6,5
= СТЕПЕНЬ (2; 5)	32
= КОРЕНЬ (64)	8
=МАКС (8,12,24,16)	24

=МИН (2,5,11,7)	2
= СРЗНАЧ (6,8,10,12)	9

Функция СУММЕСЛИ:

СУММЕСЛИ (диапазон; критерий; диапазон суммирования) – суммирует ячейки, заданные указанным условием

Функция СЧЁТЕСЛИ:

СЧЁТЕСЛИ (диапазон; критерий) – подсчитывает количество непустых ячеек в диапазоне, удовлетворяющих заданному условию

Построение диаграмм и графиков

Диаграммы и графики, как известно, предназначены для наглядного представления данных и облегчение восприятия больших массивов данных. Эту возможность представляют и ЭТ MicrosoftExcel. Диаграммы обычно располагаются на рабочем листе и позволяют проводить сравнение данных, находить закономерности. MicrosoftExcel предоставляет чрезвычайно широкие возможности в построении всевозможных видов диаграмм.

Виды диаграмм в MicrosoftExcel: круговые, гистограмма, линейчатые, график, лепестковые, кольцевые и др.

Создание диаграмм с помощью Мастера диаграмм

- выделить области данных, по которым будет строится диаграмма;
- вызвать Мастер диаграмм (выполнить команду *Вставка – Диаграмма* либо нажать на соответствующую кнопку в панели инструментов);
- выбрать тип диаграммы и щёлкнуть по кнопке Далее;
- изменить диапазон данных (если необходимо) и щёлкнуть по кнопке Далее;
- установить необходимые параметры диаграммы: название, подписи осей, подписи значений и щёлкнуть по кнопке Далее;
- установить размещение диаграммы и щёлкнуть по кнопке Готово

Изменение отдельных параметров диаграмм

- выделить диаграмму (выполнить щелчок мышью по диаграмме);
- выбрать пункт меню Диаграмма (либо вызвать контекстное меню);
- из появившегося меню выбрать необходимую команду;
- в появившемся окне установить необходимые параметры;
- щёлкнуть по кнопке Ok.

Быстрый способ создания диаграмм

- выделить области данных, по которым будет строится диаграмма;
- нажать клавишу F11.
- При этом MicrosoftExcel на основе выделенного диапазона построит стандартный тип диаграммы на отдельном листе.

Печать диаграмм

- выделить диаграмму;
- выполнить команду Файл – Печать (или нажать комбинацию клавиш Ctrl + P);
- установить переключатель Выделенную диаграмму;

- перед печатью просмотреть диаграмму (щёлкнуть мышью по кнопке Просмотр);
- установить количество копий;
- нажать кнопку Ok.

Вопросы:

1. Что такое электронная таблица?
2. Перечислите основные функции программы MSExcel?
3. Какие данные можно вводить в ячейку?
4. Перечислите правила записи формул?
5. Какие форматы ячеек существуют?
6. Как можно создать диаграмму в программе MSExcel7
7. Назначение стандартных функций?
8. Какие операции можно выполнять с выделенными объектами?

Лекция № 8.

Тема: «Вычислительные комплексы и сети»

План:

1. Понятие вычислительных комплексов и вычислительных сетей.
2. Классификация компьютерных сетей.Линии связи, их основные компоненты и характеристики.
3. Топологии сетей.
4. Модель взаимосвязи открытых систем.
5. Сетевое оборудование.
6. Основные услуги компьютерных сетей: электронная почта, телеконференции, файловые архивы.

1.Понятие вычислительных комплексов и вычислительных сетей.

Обработкаинформацииприпомощи ЭВМразвиваетсяподвумнаправлениям:

- с использованиемвычислительныхкомплексов;
- сиспользованиемвычислительныхсетей.

Вычислительныекомплексыслужатдляповышенияпроизводительностиинадежности обработки информации. Они объединяют несколько ЭВМ, территориальнорасположенных водномместе, иделятся надватипа:

- многомашинные комплексы (несколько самостоятельных ЭВМ, в том числеирезервных, объединенныхобщим управлением);
- многопроцессорныекомплексы(несколькопроцессоров,работающихсодной общейпамятьюсразличными возможнымитипамидоступакней).

Использование вычислительных комплексов позволяет разделить поставленнуюзадачунанесколькоподзадач(если этопозволяетсамазадача) ирешать ихпараллельно.

Вычислительная или компьютерная сеть – это два или более компьютера, обменивающихся информацией по линиям связи.

Компьютерная сеть позволяет передавать информацию с одного компьютера на другой, а значит, совместно использовать ресурсы, например, принтеры, модемы и устройства хранения информации.

2.Классификация компьютерных сетей.Линии связи, их основные компоненты и характеристики.

По территориальной распространенности сети могут быть локальными, глобальными, и региональными.

Локальнаясеть (LAN - LocalAreaNetwork) - сеть в пределах предприятия, учреждения, одной организации.

Региональнаясеть (MAN - MetropolitanAreaNetwork) - сеть в пределах города или области.

Глобальнаясеть (WAN - WideAreaNetwork) – сеть на территории государства или группы государств.

Рис.28.

По скорости передачи информации компьютерные сети делятся на низко-, средне- и высокоскоростные:

- **низкоскоростные** сети - до 10 Мбит/с;
- **среднескоростные** сети- до 100 Мбит/с;
- **высокоскоростные** сети - свыше 100 Мбит/с.

По типу среды передачи сети разделяются на:

- **проводные** (на коаксиальном кабеле, на витой паре, оптоволоконные);
- **беспроводные** с передачей информации по радиоканалам или в инфракрасном диапазоне.

По способу организации взаимодействия компьютеров сети делят на **одноранговые** и **свыделеннымсервером** (**иерархические** сети).

Все компьютеры одноранговой сети равноправны. Любой пользователь сети может получить доступ к данным, хранящимся на любом компьютере.

Главное достоинство одноранговых сетей – это простота установки и эксплуатации. Главный недостаток состоит в том, что в условиях одноранговых сетей затруднено решение вопросов защиты информации. Поэтому такой способ организации сети используется для сетей с небольшим количеством компьютеров и там, где вопрос защиты данных не является принципиальным.

В иерархической сети при установке сети заранее выделяются один или несколько **серверов** - компьютеров, управляющих обменом данных по сети и распределением ресурсов. Любой компьютер, имеющий доступ к услугам сервера называют **клиентомсети** или **рабочейстанцией**.

Сервер в иерархических сетях - это постоянное хранилище разделяемых ресурсов. Сам сервер может быть клиентом только сервера более высокого уровня иерархии. Серверы обычно представляют собой высокопроизводительные компьютеры, возможно, с несколькими параллельно работающими процессорами, винчестерами большой емкости и высокоскоростной сетевой картой.

Иерархическая модель сети является наиболее предпочтительной, так как позволяет создать наиболее устойчивую структуру сети и более рационально распределить ресурсы. Также достоинством иерархической сети является более высокий уровень защиты данных. К недостаткам иерархической сети, по сравнению с одноранговыми сетями, относятся:

1. Необходимость дополнительной ОС для сервера.
2. Более высокая сложность установки и модернизации сети.
3. Необходимость выделения отдельного компьютера в качестве сервера.

По технологии использования сервера различают сети с архитектурой **файл-сервер** и сети с архитектурой **клиент-сервер**. В первой модели используется файловый сервер, на котором хранится большинство программ и данных. По требованию пользователя ему пересылаются необходимая программа и данные. Обработка информации выполняется на рабочей станции.

В системах с архитектурой клиент-сервер обмен данными осуществляется между приложением-клиентом и приложением-сервером. Хранение данных и их обработка производится на мощном сервере, который выполняет также контроль за доступом к ресурсам и данным. Рабочая станция получает только результаты запроса.

К основным характеристикам сетей относятся:

Пропускная способность – максимальный объем данных, передаваемых сетью в единицу времени. Пропускная способность измеряется в Мбит/с.

Время реакции сети - время, затрачиваемое программным обеспечением и устройствами сети на подготовку к передаче информации по данному каналу. Время реакции сети измеряется миллисекундах.

Internet

Internet – это глобальная компьютерная система, которая:

- логически взаимосвязана пространством глобальных уникальных адресов (каждый компьютер, подключаемый к сети имеет свой уникальный адрес);
- способна поддерживать **коммуникации** (обмен информацией);
- обеспечивает работу высокоуровневых сервисов (служб), например, WWW, электронная почта, телеконференции, разговоры в сети и другие.

Internetявляется **одноранговой** сетью, т.е. все компьютеры в сети равноправны, и любой компьютер можно подключить к любому другому компьютеру. Т.о., любой компьютер, подключенный к сети, может предлагать свои услуги любому другому.

Internet предоставляет пользователям всевозможные **информационные** и **коммуникационные** услуги.

Информационные услуги - услуги доступа к информации:

- доступ к информационным ресурсам сети, то есть можно получить необходимую информацию, имеющуюся на серверах сети, например, документы, файлы, информацию из различных баз данных и т.п.;
- размещение собственной информации в сети. Существует множество серверов, предоставляющих возможность бесплатно разместить на них информацию. Если информация размещается в целях публикации, то любые пользователи Internet могут получить доступ к этой информации и получать и просматривать ее в любое время.

Коммуникационные услуги - услуги обмена информацией, общения:

- обмен информацией в отсроченном режиме. Так работает, например, электронная почта. Отправитель направляет письмо в почтовый ящик получателя, который просмотрит это письмо в удобное для него время.
- обмен в режиме реального времени. Например, разговоры в сети. Люди набирают свои реплики с клавиатуры и посылают их на разговорный сервер, и эти реплики видят все участники разговора одновременно.

История возникновения Internet

1969 – 4 компьютера объединены в APRANET.

1971 – работа над проектом InternettingProject (Проект объединения сетей). В ходе выполнения проекта был разработан протокол TCP/IP (TransmissionControlProtocol/InternetProtocol – Протокол управления передачей/Межсетевой протокол).

1 января 1983 года был осуществлен одновременный переход всех компьютеров в составе ARPANET на протокол TCP/IP.

1990 – первая российская сеть **RELCOM**, созданная на базе Курчатовского института атомной энергии в Москве.

- При подключении кИнтернет сеть не должна подвергаться внутренним переделкам;
- Если пакет с информацией не прибыл в пункт назначения, источник должен вскоре повторно передать его;
- Для объединения сетей должны использоваться черные ящики (шлюзы и маршрутизаторы), которые должны оставаться простыми;
- Не должно существовать общей системы управления глобальной сетью.

Протоколы передачи информации в Internet

Протокол — это набор правил и соглашений, используемых при передаче данных.

Основополагающим протоколом сети Internet является **протокол TCP/IP**.

TCP (TransmissionControlProtocol) — протокол управления передачей. Он определяет, каким образом информация должна быть разбита на пакеты и отправлена по каналам связи. TCP располагает пакеты в нужном порядке, а также проверяет каждый пакет на наличие ошибок при передаче.

Каждый информационный пакет содержит **IP-адреса** (**IP** – InternetProtocol) компьютера-отправителя и компьютера-получателя. Специальные компьютеры, называемые маршрутизаторами, используя IP-адреса, направляют информационные пакеты в нужную сторону, то есть к указанному в них получателю.

Протоколтелеконференций
NewsNetTransferProtocol
NNTP
Протокол получения электронных писем
PostOfficeProtocol 3
POP3
Простой протокол отправки электронных писем
Simple Mail Transfer Protocol
SMTP
Протоколпередачифайлов
File Transfer Protocol
FTP
Протоколпередачигипертекста
Hyper Text Transfer Protocol
HTTP

Адресакомпьютеровв Internet

Каждый компьютер, подключенный к сети Internet, имеет свой уникальный **IP-адрес**.

IP-адрес — это уникальный номер, однозначно идентифицирующий компьютер в Internet. IP-адрес представляет собой четыре числа (октета), разделенные точками, например, 1**94.67.67.97** (после последнего числа точка не ставится).

Каждое число может быть в интервале от 0 до 255, что соответствует информационному объему в 1 байт или 8 бит. Таким образом, IP-адрес – это 4 байта или 32 бита. Если с помощью одного байта можно передать 2^8=256 вариантов, то с помощью 4-х байтов можно передать 2^{32}»4 млрд. вариантов, то к сети Internet может быть максимально подключено 4 млрд. пользователей.

Система доменных имен

Числовые адреса хороши для связи машин, но при работе в сети удобнее использовать имена. Поэтому всем компьютерам в Internet были присвоены собственные (**доменные**) имена (например, www.sch130.nsc.ru).

Раньше соответствие между адресом и именем определялось из специального текстового файла - hostfile (файл рабочих ЭВМ).

Система запросов в сети Интернет, позволяющая получать информацию о соответствии адресов и имен по сети, которая сейчас называется **доменной системой имен** - DNS (DomainNameSystem).

Домены в именах отделяются друг от друга точками:

nsc.ru - Сеть ННЦ
ict.nsc.ru - Сеть ИВТ СО РАН
nsu.ru - Университетская сеть в Новосибирске
www.sch130.nsc.ru – Лицей №130
http://www.ripn.net/ Российский НИИ Развития Общественных Сетей
ru– Россия,
us– США,
fr– Франция,
cn– Китай,
cl– Чили,
jp– Япония
edu- сеть университетов
com - сеть коммерческих организаций
gov- сеть государственных организаций
mil- сеть Министерства обороны США
org- сеть общественных организаций
net- сеть сетевых организаций

URL

IP-адрес или соответствующее ему доменное имя позволяют однозначно идентифицировать компьютер в сети Internet.

Но дело в том, что на компьютере может присутствовать множество различной информации в различных форматах, например, в виде файлов, электронных сообщений, страниц и т.п. Для того, чтобы можно было безошибочно получать нужную информацию и в нужном формате используется строка символов, которую называют универсальный указатель ресурса или **URL** (UniversalResourceLocator). Эта строка однозначно идентифицирует любой ресурс в сети Internet.

3.Топологии сетей

Топологией сети называется физическую или электрическую конфигурацию кабельной системы и соединений сети. В топологии сетей применяют несколько специализированных терминов:

- узел сети - компьютер, либо коммутирующее устройство сети;
- ветвь сети - путь, соединяющий два смежных узла;
- оконечный узел - узел, расположенный в конце только одной ветви;
- промежуточный узел - узел, расположенный на концах более чем одной ветви;
- смежные узлы - узлы, соединенные, по крайней мере, одним путём, не содержащим никаких других узлов.

Существует всего 5 основных типов топологии сетей:

1. Топология “Общая Шина”. В этом случае подключение и обмен данными производится через общий канал связи, называемый общей шиной:

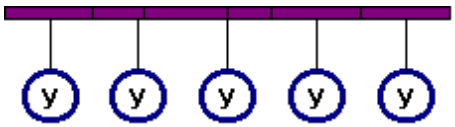


Рис.29.

Общая шина является очень распространенной топологией для локальных сетей. Передаваемая информация может распространяться в обе стороны. Применение общей шины снижает стоимость проводки и унифицирует подключение различных модулей. Основными преимуществами такой схемы являются дешевизна и простота разводки кабеля по помещениям. Самый серьезный недостаток общей шины заключается в ее низкой надежности: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть. Поэтому пропускная способность канала связи всегда делится здесь между всеми узлами сети.

2. Топология “Звезда”. В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому **концентратором**, который находится в центре сети:

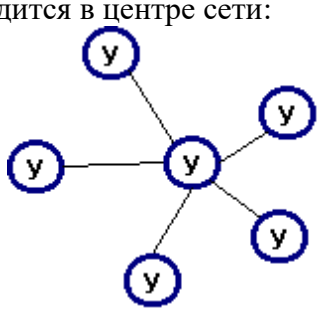


Рис.30

В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. Главное преимущество этой топологии перед общей шиной - существенно большая надежность. Любые неприятности с кабелем касаются лишь того компьютера, к которому этот кабель присоединен, и только неисправность концентратора может вывести из строя всю сеть. Кроме того, концентратор может играть роль интеллектуального фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения концентратора. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. В настоящее время иерархическая звезда является самым распространенным типом топологии связей как в локальных, так и глобальных сетях.

3. Топология “Кольцо”. В сетях с кольцевой топологией данные в сети передаются последовательно от одной станции к другой по кольцу, как правило, в одном направлении:

Если компьютер распознает данные как предназначенные ему, то он копирует их себе во внутренний буфер. В сети с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями. Преимущество данной топологии - простота управления, недостаток - возможность отказа всей сети при сбое в канале между двумя узлами.

Рис.31.

4.Ячеистая топология. Для ячеистой топологии характерна схема соединения компьютеров, при которой физические линии связи установлены со всеми рядом стоящими компьютерами:

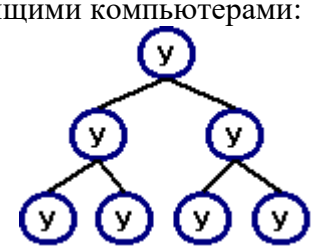


Рис.32.

В сети с ячеистой топологией непосредственно связываются только те компьютеры, между которыми происходит интенсивный обмен данными, а для обмена данными между компьютерами, не соединенными прямыми связями, используются транзитные передачи через промежуточные узлы. Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для глобальных сетей. Достоинства данной топологии в ее устойчивости к отказам и перегрузкам, т.к. имеется несколько способов обойти отдельные узлы.

5. **Смешанная топология.** В то время как небольшие сети, как правило, имеют типовую топологию - звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно подсети, имеющие типовую топологию, поэтому их называют сетями со смешанной топологией:

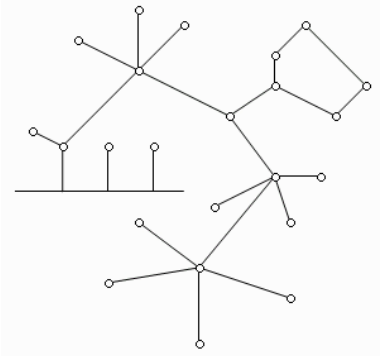


Рис.33.

4.Модель взаимосвязи открытых систем

Основной задачей, решаемой при создании компьютерных сетей, является обеспечение совместимости оборудования по электрическим и механическим характеристикам и обеспечение совместимости информационного обеспечения (программ и данных) по системе кодирования и формату данных. Решение этой задачи относится к области стандартизации и основано на так называемой **модели OSI** (модель

взаимодействия открытых систем –ModelofOpenSystemInterconnections). Модель OSI была создана на основе технических предложений Международного института стандартов ISO (InternationalStandardsOrganization).

Согласно модели OSI архитектуру компьютерных сетей следует рассматривать на разных уровнях (общее число уровней - до семи). Самый верхний уровень - прикладной. На этом уровне пользователь взаимодействует с вычислительной системой. Самый нижний уровень - физический. Он обеспечивает обмен сигналами между устройствами. Обмен данными в системах связи происходит путем их перемещения с верхнего уровня на нижний, затем транспортировки и, наконец, обратным воспроизведением на компьютере клиента в результате перемещения с нижнего уровня на верхний.

Рис.34.

Для обеспечения необходимой совместимости на каждом из семи возможных уровней архитектуры компьютерной сети действуют специальные стандарты, называемые **протоколами**. Они определяют характер аппаратного взаимодействия компонентов сети (**аппаратныепротоколы**) и характер взаимодействия программ и данных (**программныепротоколы**). Физически функции поддержки протоколов исполняют аппаратные устройства (**интерфейсы**) и программные средства (программы поддержки протоколов). Программы, выполняющие поддержку протоколов, также называют протоколами.

Каждый уровень архитектуры подразделяется на две части:

- спецификацию услуг;
- спецификацию протокола.

Спецификация услуг определяет, что делает уровень, а спецификация протокола - как он это делает, причем каждый конкретный уровень может иметь более одного протокола.

Рассмотрим функции, выполняемые каждым уровнем программного обеспечения:

1. *Физическийуровень* осуществляет соединения с физическим каналом, так, отсоединения от канала, управление каналом. Определяется скорость передачи данных и топология сети.

2. *Канальныйуровень* добавляет в передаваемые массивы информации вспомогательные символы и контролирует правильность передаваемых данных. Здесь

передаваемая информация разбивается на несколько пакетов или кадров. Каждый пакет содержит адреса источника и места назначения, а также средства обнаружения ошибок.

3. *Сетевой уровень* определяет маршрут передачи информации между сетями, обеспечивает обработку ошибок, а так же управление потоками данных. Основная задача сетевого уровня - маршрутизация данных (передача данных между сетями).

4. *Транспортный уровень* связывает нижние уровни (физический, канальный, сетевой) с верхними уровнями, которые реализуются программными средствами. Этот уровень разделяет средства формирования данных в сети от средств их передачи. Здесь осуществляется разделение информации по определенной длине и уточняется адрес назначения.

5. *Сеансовый уровень*осуществляет управление сеансами связи между двумя взаимодействующими пользователями, определяет начало и окончание сеанса связи, время, длительность и режим сеанса связи, точки синхронизации для промежуточного контроля и восстановления при передаче данных; восстанавливает соединение после ошибок во время сеанса связи без потери данных.

6. *Представительский*- управляет представлением данных в необходимой для программы пользователя форме, производит компрессию и декомпрессию данных. Задачей данного уровня является преобразование данных при передаче информации в формат, который используется в информационной системе. При приеме данных данный уровень представления данных выполняет обратное преобразование.

7. *Прикладной* уровень взаимодействует с прикладными сетевые программами, обслуживающими файлы, а также выполняет вычислительные, информационно-поисковые работы, логические преобразования информации, передачу почтовых сообщений и т.п. Главная задача этого уровня - обеспечить удобный интерфейс для пользователя.

На разных уровнях обмен происходит различными единицами информации: биты, кадры, пакеты, сеансовые сообщения, пользовательские сообщения.

5.Сетевое оборудование

Основными компонентами сети являются рабочиестанции, серверы, передающиесреды (кабели) и сетевоеоборудование.

Рабочимистанциями называются компьютеры сети, на которых пользователями сети реализуются прикладные задачи.

Серверысети - это аппаратно-программные системы, выполняющие функции управления распределением сетевых ресурсов общего доступа. Сервером может быть это любой подключенный к сети компьютер, на котором находятся ресурсы, используемые другими устройствами локальной сети. В качестве аппаратной части сервера используется достаточно мощные компьютеры.

Сети можно создавать с любым из типов кабеля.

1.*Витая пара (TP - TwistedPair)*– это кабель, выполненный в виде скрученной пары проводов. Он может быть экранированным и неэкранированным. Экранированный кабель более устойчив к электромагнитным помехам. Витая пара наилучшим образом подходит для малых учреждений. Недостатками данного кабеля является высокий коэффициент затухания сигнала и высокая чувствительность к электромагнитным помехам, поэтому максимальное расстояние между активными устройствами в ЛВС при использовании витой пары должно быть не более 100 метров.

2. *Коаксиальный кабель* состоит из одного цельного или витого центрального проводника, который окружен слоем диэлектрика. Проводящий слой алюминиевой фольги, металлической оплетки или их комбинации окружает диэлектрик и служит одновременно как экран против наводок. Общий изолирующий слой образует внешнюю оболочку кабеля.

Коаксиальный кабель может использоваться в двух различных системах передачи данных: без модуляции сигнала и с модуляцией. В первом случае цифровой сигнал используется в таком виде, в каком он поступает из ПК и сразу же передается по кабелю на приемную станцию. Он имеет один канал передачи со скоростью до 10 Мбит/сек и максимальный радиус действия 4000 м. Во втором случае цифровой сигнал превращают в аналоговый и направляют его на приемную станцию, где он снова превращается в цифровой. Операция превращения сигнала выполняется модемом; каждая станция должна иметь свой модем. Этот способ передачи является многоканальным (обеспечивает передачу по десяткам каналов, используя для этого всего лишь один кабель). Таким способом можно передавать звуки, видео сигналы и другие данные. Длина кабеля может достигать до 50 км.

3. *Оптоволоконный кабель* является более новой технологией, используемой в сетях. Носителем информации является световой луч, который модулируется сетью и принимает форму сигнала. Такая система устойчива к внешним электрическим помехам и таким образом возможна очень быстрая, секретная и безошибочная передача данных со скоростью до 2 Гбит/с. Количество каналов в таких кабелях огромно. Передача данных выполняется только в симплексном режиме, поэтому для организации обмена данными устройства необходимо соединять двумя оптическими волокнами (на практике оптоволоконный кабель всегда имеет четное, парное кол-во волокон). К недостаткам оптоволоконного кабеля можно отнести большую стоимость, а также сложность подсоединения.

4. *Радиоволны*в микроволновом диапазоне используются в качестве передающей среды в беспроводных локальных сетях, либо между мостами или шлюзами для связи между локальными сетями. В первом случае максимальное расстояние между станциями составляет 200 - 300 м, во втором - это расстояние прямой видимости. Скорость передачи данных - до 2 Мбит/с.

Беспроводные локальные сети считаются перспективным направлением развития ЛС. Их преимущество - простота и мобильность. Также исчезают проблемы, связанные с прокладкой и монтажом кабельных соединений - достаточно установить интерфейсные платы на рабочие станции, и сеть готова к работе.

Выделяют следующие виды сетевого оборудования:

1. *Сетевыекарты* – это контроллеры, подключаемые в слоты расширения материнской платы компьютера, предназначенные для передачи сигналов в сеть и приема сигналов из сети.

2. *Терминаторы* - это резисторы номиналом 50 Ом, которые производят затухание сигнала на концах сегмента сети.

3. *Концентраторы (Hub)* – это центральные устройства кабельной системы или сети физической топологии "звезда", которые при получении пакета на один из своих портов пересылает его на все остальные. В результате получается сеть с логической структурой общей шины. Различают концентраторы активные и пассивные. Активные концентраторы усиливают полученные сигналы и передают их. Пассивные концентраторы пропускают через себя сигнал, не усиливая и не восстанавливая его.

4. *Повторители (Repeater)*- устройства сети, усиливает и заново формирует форму входящего аналогового сигнала сети на расстояние другого сегмента. Повторитель действует на электрическом уровне для соединения двух сегментов. Повторители ничего распознают сетевые адреса и поэтому не могут использоваться для уменьшения трафика.

5. *Коммутаторы (Switch*) - управляемые программным обеспечением центральные устройства кабельной системы, сокращающие сетевой трафик за счет того, что пришедший пакет анализируется для выяснения адреса его получателя и соответственно передается только ему. Использование коммутаторов является более дорогим, но и более производительным решением. Коммутатор обычно значительно более сложное устройство и может обслуживать одновременно несколько запросов. Если по какой-то

причине нужный порт в данный момент времени занят, то пакет помещается в буферную память коммутатора, где и дожидается своей очереди. Построенные с помощью коммутаторов сети могут охватывать несколько сотен машин и иметь протяженность в несколько километров.

6. *Маршрутизаторы (Router)*- стандартные устройства сети, работающие на сетевом уровне и позволяющее переадресовывать и маршрутизировать пакеты из одной сети в другую, а также фильтровать широковещательные сообщения.

7. *Мосты (Bridge)*- устройства сети, которое соединяют два отдельных сегмента, ограниченных своей физической длиной, и передают трафик между ними. Мосты также усиливают и конвертируют сигналы для кабеля другого типа. Это позволяет расширить максимальный размер сети, одновременно не нарушая ограничений на максимальную длину кабеля, количество подключенных устройств или количество повторителей на сетевой сегмент.

8. *Шлюзы (Gateway)* - программно-аппаратные комплексы, соединяющие разнородные сети или сетевые устройства. Шлюзы позволяет решать проблемы различия протоколов или систем адресации. Они действует на сеансовом, представительском и прикладном уровнях модели OSI.

9. *Межсетевые экраны (firewall, брандмауэры)* - это сетевые устройства, реализующие контроль за поступающей в локальную сеть и выходящей из нее информацией и обеспечивающие защиту локальной сети посредством фильтрации информации. Большинство межсетевых экранов построено на классических моделях разграничения доступа, согласно которым субъекту (пользователю, программе, процессу или сетевому пакету) разрешается или запрещается доступ к какому-либо объекту (файлу или узлу сети) при предъявлении некоторого уникального, присущего только этому субъекту, элемента. В большинстве случаев этим элементом является пароль. В других случаях таким уникальным элементом является микропроцессорные карточки, биометрические характеристики пользователя и т. п. Для сетевого пакета таким элементом являются адреса или флаги, находящиеся в заголовке пакета, а также некоторые другие параметры. Таким образом, межсетевой экран - это программный и/или аппаратный барьер между двумя сетями, позволяющий устанавливать только авторизованные межсетевые соединения. Обычно межсетевые экраны защищают соединяемую с Internet корпоративную сеть от проникновения извне и исключает возможность доступа к конфиденциальной информации.

6.Информационные ресурсы сети Интернет: электронная почта, телеконференции, файловые архивы.

Глобальная сеть Интернет привлекает пользователей своими информационными ресурсами и сервисами (услугами). В настоящее время услугами Интернета пользуются несколько сотен миллионов человек.

Электронная почта — наиболее распространенный сервис Интернета, так как она является исторически первой информационной услугой компьютерных сетей и не требует обязательного наличия высокоскоростных и качественных линий связи.Любой пользователь Интернета может получить свой «почтовый ящик» на одном из почтовых серверов Интернета (обычно на почтовом сервере провайдера), в котором будут храниться передаваемые и получаемые электронные письма.Чтобы электронное письмо дошло до адресата, оно, кроме текста послания, обязательно должно содержать электронный адрес получателя письма.Адрес электронной почты записывается по определенной форме и состоит из двух частей: имя_пользователя@имя_сервера. Имя_пользователя имеет произвольный характер и задается самим пользователем; имя_сервера жестко связано с выбором пользователем сервера, на котором он разместил свой почтовый ящик. Например, имя почтового сервера компании МТУ-Интел — mtu-

net.ru. Соответственно имена почтовых ящиков пользователей будут иметь вид: username@mtu-net.ru Чтобы отправить электронное письмо, отправитель должен подключиться к Интернету и передать на свой почтовый сервер сообщение. Почтовый сервер сразу же отправит это письмо через систему почтовых серверов Интернет на почтовый сервер получателя, и оно попадет в его почтовый ящик. Однако получатель получит письмо только после того, как соединится с Интернетом и «скачает» почту из своего почтового ящика на собственный локальный компьютер.

Телеконференции. В Интернете существуют десятки тысяч конференций или групп новостей (news), каждая из которых посвящена обсуждению какой-либо проблемы. Любой конференции выделяется свой почтовый ящик на серверах Интернета, поддерживающих работу этой телеконференции.Пользователи могут посылать свои сообщения на любой из этих серверов. Серверы периодически синхронизируются, т. е. обмениваются содержимым почтовых ящиков телеконференций, поэтому материалы конференций в полном объеме доступны пользователю на каждом таком сервере.Принцип работы в телеконференциях мало чем отличается от принципа работы с электронной почтой. Пользователь имеет возможность посылать свои сообщения в любую телеконференцию и читать сообщения, посланные другими участниками.

Файловые архивы. Большое количество серверов Интернета содержат файловые архивы. Программное обеспечение, размещаемое на таких серверах, можно разделить на две большие группы: свободно распространяемое программное обеспечение freeware и условно бесплатное программное обеспечение shareware.Многие производители программного обеспечения и компьютерного оборудования заинтересованы в широком бесплатном распространении программного обеспечения. К таким программным средствам можно отнести новые недоработанные (бета) версии программных продуктов, драйверы к новым устройствам или улучшенные драйверы к уже существующим и т. д. В рекламных целях на файловых серверах фирмы часто размещают также условно бесплатное программное обеспечение (программы с ограниченным сроком действия или программы с ограниченными функциональными возможностями).Для работы с серверами файловых архивов можно использовать браузеры, однако удобнее пользоваться специальными программами, которые называются менеджерами загрузки файлов

Вопросы:

1. Что такое телекоммуникационные технологии?
2. Характеристика TCP/IP?
3. Какие среды каналов связи существуют?
4. Дайте характеристику оптоволокна?
5. Что такое Интернет -технологии?
6. Перечислите основные способы подключения к Интернет?
7. Что такое электронная почта?
8. Что такое поисковая система?
9. Какие популярные поисковые системы вы знаете?
10. Что такое компьютерная сеть?
11. Перечислите топологии сетей?
12. Что такое проводная связь?
13. Какие виды проводной связи различают?
14. Что используют при осуществлении проводной связи?
15. Для чего используется беспроводная технология?
16. Где используются беспроводные компьютерные сети и системы связи?
17. Какие преимущества имеет беспроводная технология?

Лекция №9.
Тема: «Аппаратное устройство компьютера»

План:

1. Принципы Джона фон Неймана
2. Архитектура ЭВМ
3.Основные характеристики модулей ЭВМ
4.Основные характеристики процессора компьютера.
5. Организация и основные характеристики памяти компьютера.
6.Магистраль.
7.Принцип открытой архитектуры
8.Устройства ввода– вывода.

1.Принципы Джона фон Неймана

В основу построения большинства ЭВМ положены принципы, сформулированные в 1945 г. Джоном фон Нейманом:

1. Принцип программного управления (программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в заданной последовательности).

2. Принцип однородности памяти (программы и данные хранятся в одной и той же памяти; над командами можно выполнять такие же действия, как и над данными).

3. Принцип адресности (основная память структурно состоит из пронумерованных ячеек).

ЭВМ, построенные на этих принципах, имеют ***классическую архитектуру*** (архитектуру фон Неймана).

2.Архитектура ЭВМ

Архитектура ЭВМ - это общее описание структуры и функций ЭВМ на уровне, достаточном для понимания принципов работы и системы команд ЭВМ, не включающее деталей технического и физического устройства компьютера.

К архитектуре относятся следующие принципы построения ЭВМ:

- структура памяти ЭВМ;
- способы доступа к памяти и внешним устройствам;
- возможность изменения конфигурации;
- система команд;
- форматы данных;
- организация интерфейса.

Архитектура определяет принцип действия, информационные связи и взаимное соединение основных ***логических узлов ЭВМ***:

- центрального процессора;
- периферийных процессоров;
- оперативного ЗУ (запоминающего устройства);
- внешних ЗУ;
- периферийных устройств.

Совокупность материальных компонент компьютера, в которых реализуются различные информационные процессы, называется ***аппаратным обеспечением,*** или ***оборудованием ПК***.

Рис.35.

3.Основные характеристики модулей ЭВМ

Персональные компьютеры обычно состоят из следующих основных модулей: системный блок, монитор, клавиатура, мышка - компьютер в настольном исполнении, компьютер в компактном исполнении (*notebook*)

В системном блоке находятся все основные узлы компьютера:

- материнская плата;
- электронные схемы (процессор, контроллеры устройств и т.д.);
- блок питания;
- дисководы (накопители).

Все эти компоненты жизненно важны для жизненно важны для компьютера, без них он не может работать. Поэтому данный блок и называется ***системным***.

Рис.36.

Оборудование, которое расположено вне системного блока, относится к внешним устройствам ввода-вывода. Это оборудование называют также *периферийными устройствами.* Однако к периферийным можно отнести и некоторые устройства внутри самого системного блока. В первую очередь, это все типы накопителей

Адаптер – электронная схема, обеспечивающая связь (сопряжение) периферийных устройств ПК с центральными (системными). Адаптер управляет периферийным устройством, контролирует правильность его работы (тогда он называется – ***контроллер***), обеспечивает интерфейс устройств ввода/вывода.

4.Основные характеристики процессора компьютера.

Процессор – основная микросхема компьютера, выполняющая обработку данных и управляющая работой всей системы.

Функции процессора:

- обработка данных по заданной программе (выполнение над ними арифметических и логических операций) – ***функция АЛУ*** (арифметико-логического устройства);
- программное управление работой устройств ЭВМ – ***функция УУ*** (устройства управления).

В общем случае под процессором понимают устройство, производящее набор операций над данными, представленными в цифровой форме (двоичным кодом). Применительно к вычислительной технике под процессором понимают центральное процессорное устройство (CPU), обладающее способностью выбирать, декодировать и выполнять команды, а также передавать и принимать информацию от других устройств. Проще говоря, ***процессор – этоэлектронная схема, выполняющая обработку информации***. Производство современных персональных компьютеров начались тогда, когда процессор был выполнен в виде отдельной микросхемы.

Количество фирм, разрабатывающих и производящих процессоры для IBM-совместимых компьютеров, невелико. Внастоящеевремяизвестны: Intel, Cyrix, AMD, NexGen, Texas Instrument... Кроме процессоров, которые составляют основу IBM-совместимых персональных компьютеров, существует целый класс процессоров, составляющих параллельную платформу (среди самых известных – персональные компьютеры американской фирмы Apple, для которых используются процессоры типа PowerPC, имеющие принципиально другую архитектуру, выпускаемые фирмой Motorola и др.). Производительность персональных компьютеров на основе процессоров PowerPC значительно выше, чем у IBM-совместимых, поэтому, несмотря на значительную разницу в цене, для серьезных профессиональных приложений им отдают предпочтение.

Производительность CPU характеризуется следующими основными параметрами:

- степенью интеграции;
- внутренней и внешней разрядностью данных;
- тактовой частотой;
- памятью, к которой может адресоваться CPU.

Рис.37.

Тактовая частотауказывает, сколько элементарных операций (тактов) микропроцессор выполняет за одну секунду (измеряется в МГц).***Степень интеграции микросхемы*** показывает, сколько транзисторов (самый простой элемент любой микросхемы) может поместиться на единице площади. Для процессора PentiumIntel эта величина составляет приблизительно 3 млн. на 3,5 кв.см, у PentiumPro – 5 млн.

Рис. 38.

Внутренняя разрядность процессора определяет, какое количество битов он может обрабатывать одновременно при выполнении арифметических операций (в зависимости от поколения процессоров – от 8 до 32 битов). ***Внешняя разрядность процессора*** определяет сколько битов одновременно он может принимать или передавать во внешние устройства (от 16 до 64 и более в современных процессорах). Тактовая частота определяет быстродействие процессора.

Для процессора различают внутреннюю (собственную) тактовую частоту процессора (с таким быстродействием могут выполняться внутренние простейшие операции) и внешнюю (определяет скорость передачи данных по внешней шине). Количество адресов ОЗУ, доступное процессору, определяется разрядностью адресной шины.С бурным развитием мультимедиа приложений перед разработчиками процессоров возникли проблемы увеличения скорости обработки огромных массивов данных, Рис. содержащих графическую, звуковую или видео информацию. В результате возникли дополнительно устанавливаемые специальные процессоры DSP а недавно появились разработанные на базе процессоров Pentium так называемые MMX-процессоры (первый из них – PentiumP55C). Желающие воспользоваться преимуществами новых MMX-процессоров должны позаботиться о приобретении нового программного обеспечения, ориентированного на работу с ними.

5.Организация и основные характеристики памяти компьютера.

Память компьютера предназначена для хранения информации (программ, данных и команд управления).

Выделяют три вида памяти компьютера: постоянное, оперативное и внешнее запоминающие устройства (ПЗУ, ОЗУ, ВЗУ).

Основные пользовательские характеристики:

- емкость (объем) – количество байтов памяти;
- быстродействие – время обращения к ячейкам памяти, определяемое временем считывания или временем записи информации. Измеряется в наносекундах (10^{-10}с);
- разрядность – количество линий ввода-вывода, которые имеют микросхемы оперативной и постоянной памяти или внешние накопители.

Во многих ПК ПЗУ(ROM) реализуется отдельной микросхемой, в которую при изготовлении ПК помещаются основные команды ввода/вывода, осуществляющие начальное взаимодействие аппаратного и программного обеспечения ПК.

Этот вид памяти доступен лишь для чтения хранящейся в ней информации.

После выключения питания компьютера информация в ПЗУ сохраняется, то есть ПЗУ – *энергоНЕзависимое* устройство.

ОЗУ (RAM) неотъемлемая часть любого ПК. Это быстродействующееЗУ сравнительно небольшого (по сравнению сВЗУ) объёма, реализованное в виде***электронной схемы.***

ОЗУ доступно как для чтения, так и для записи информации. Именно в ОЗУ хранится выполняемая ПК в текущий момент программа и необходимые для неё данные.

ОЗУ работает под непосредственным управлением микропроцессора, все данные для которого поступают только из ОЗУ.

ОЗУ обеспечивает хранение информации лишь в течение сеанса работы с ПК — после выключения компьютера из сети данные, хранимые в ОЗУ, теряются безвозвратно, то есть ОЗУ — *энергозависимое* устройство.

Ёмкость ОЗУ современных моделей ПК колеблется от 640 Кбайт (IBMPCXT) до 128 Мбайт.

Для ускорения вычислений информация из наиболеечасто используемых участковОЗУ помещается в сверхбыстродействующие микросхемы памяти — *кэш-память.* Отсутствие кэш-памяти может существенно (на 20-30%) снизить общую производительность компьютера.

В настоящее время используется кэш-память от64 до 512 Кбайт.

Размещение информации в памяти называется ***записью***, а получение информации из памяти – ***чтением*** или ***считыванием***.

Функции памяти:

- ***приём*** информации от других устройств;
- ***запоминание*** информации;
- ***передача*** информации по запросу в другие устройства машины.

Рис. 39.

Центральный процессор имеет доступ то к данным, находящимся в оперативной памяти (физическое устройство памяти называется ОЗУ – оперативное запоминающее устройство или RAM – RandomAccessMemory). Работа компьютера с пользовательскими программами начинается после того как данные будут считаны из внешней памяти в ОЗУ. ОЗУ работает синхронно с центральным процессором и имеет малое время доступа. *Оперативная память сохраняет данные только при включенном питании*. Отключение питания приводит к необратимой потере данных, поэтому пользователю, работающему с большими массивами данных в течение длительного времени, рекомендуют периодически сохранять промежуточные результаты на внешнем носителе. По способу реализации оперативная память делится на динамическую и статическую. Динамическая память напоминает дырявое ведро, в котором, если регулярно не доливать, скоро не останется воды. Регулярный долив применительно к динамической памяти, называется регенерацией и производится раз в несколько миллисекунд, что несколько снижает быстродействие системы. Однако эти недостатки искупаются простотой исполнения, а также большой емкостью микросхем динамической памяти. Статическая память при включенном питании надежно хранит записанные данные, имеет малое время доступа, потребляет мизерный ток, но емкость ее микросхем ограничена. *Основными характеристиками ОЗУ* являются: количество ячеек памяти (адреса) и время доступа к информации, определяемое интервалом времени, в течение которого информация записывается в память или считывается из нее.

Основой ОЗУ являются микросхемы памяти (chips), которые объединяются в блоки (банки) различной конфигурации. При комплектации банков различнымимикросхемами необходимо следить, чтобы время доступа у них не различалось больше, чем на 10 нс.

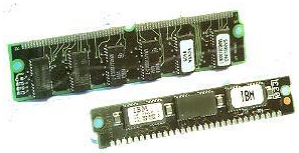


Рис.40.

Рис.41.

Для нормального функционирования системы большое значение имеет согласование быстродействия центрального процессора и ОЗУ.

Оперативнаяпамятьбывает: SIMM (Single In-Line Memory Module) и DIMM (Dual In-Line Memory Module).В системную плату модули SIMM необходимо было вставлять только попарно, а DIMM можно выбрать по одному, что связано с разрядностью внешней шины данных процессоров Pentium. Такой способ установки предоставляет больше возможностей для варьирования объема оперативной памяти.

Первоначально материнские платы поддерживали оба разъема, но уже довольно продолжительное время они комплектуются исключительно разъемами DIMM. Сейчас в

качестве оперативной памяти используются модули SIMM, DIMM, RIMM, SO-DIMM и SO-RIMM. Все они имеют разное количество контактов. Модули SIMM сейчас встречаются только в старых моделях материнских плат, а им на смену пришли 168-контактные DIMM. Модули SO-DIMM и SO-RIMM, имеющие меньшее количество контактов, чем стандартные DIMM и RIMM, широко используются в портативных устройствах. Модули RIMM можно встретить в платах на новом чипсете Intel 820.

Дисковая память – наиболее распространенный тип долговременной памяти, отличающийся высоким быстродействием и удобством использования. Для дисковой памяти характерно использование метода прямого доступа к памяти.

Кэш-память – используется для производительности современных ПК, имеет меньший объем чем ОЗУ (16-256 Кб), но обладает более высоким быстродействием. Кэш является отдельным устройством памяти, размещенным на материнской плате.

6.Магистраль.

Архитектура современных персональных компьютеров основана на магистрально-модульном принципе. Связь и обмен информацией всех узлов компьютера организуется с помощью ***системной шины***. Системная шина называется также***магистралью.***

Шиной в электротехнике называется толстый медный провод, предназначенный для передачи больших токов. В компьютерной технике словом «шина» обозначают устройство для связи между собой нескольких узлов компьютера. Системная шина связывает, в первую очередь, МП со всеми узлами компьютера. Кроме этого, через системную шину-магистраль узлы связываются между собой.

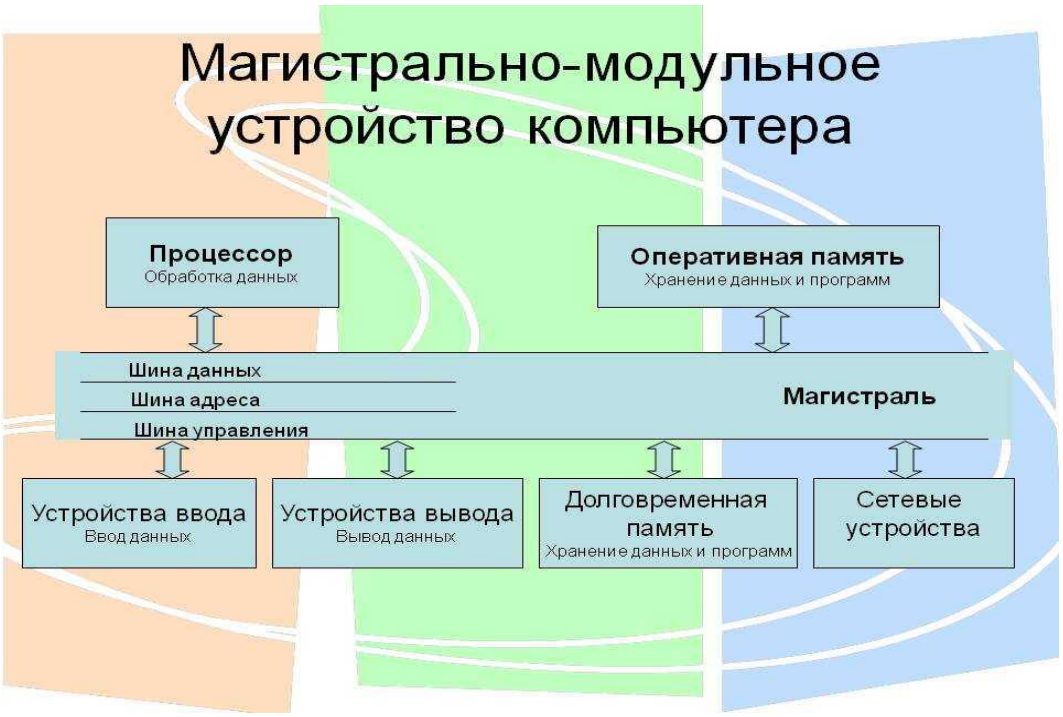


Рис.42.

Магистраль включает в себя следующие три шины:

- ✓ *Шина управления,* которая служит для управления со стороны МП всеми системами и процессами, происходящими в компьютере.
- ✓ *Шина адреса* (адресная шина), с помощью которой осуществляется выбор нужной ячейки памяти, а также портов ввода-вывода.

✓ *Шина данных,* по которой информация передается от МП к какому-либо устройству либо, наоборот, от устройства к МП.

Каждая шина – это набор электрических соединений-проводов. Адресная шина, например для МП Intel 8088 состоит из 20 параллельных проводов – по одному для каждого бита. То есть адресная шина для этого МП является 20-разрядной.

7.Принцип открытой архитектуры

В современных ЭВМ реализован принцип открытой архитектуры, позволяющий пользователю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости её модернизацию. Конфигурацией компьютера называют фактический набор компонентов ЭВМ, которые составляют компьютер. Принцип открытой архитектуры позволяет менять состав устройств ЭВМ. К информационной магистрали могут подключаться дополнительные периферийные устройства, одни модели устройств могут заменяться на другие.

Принцип открытой архитектуры заключается в следующем:

- Регламентируются и стандартизируются только описание принципа действия компьютера и его конфигурация. Компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.
- Компьютер легко расширяется и модернизируется за счёт наличия внутренних расширительных гнёзд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.

Для того, чтобы соединить друг с другом различные устройства компьютера, они должны иметь одинаковый *интерфейс* (англ. *interface* от *inter* — между, и *face* — лицо).

Каждый из функциональных элементов связан с шиной определённого типа - адресной, управляющей или шиной данных. Для согласования интерфейсов периферийные устройства подключаются к шине не напрямую, а через свои *контроллеры* (адаптеры) и *порты* примерно по такой схеме:

Контроллеры и адаптеры представляют собой наборы электронных цепей, которыми снабжаются устройства компьютера с целью совместимости их интерфейсов. Контроллеры, кроме этого, осуществляют непосредственное управление периферийными устройствами по запросам микропроцессора.

Порты устройств представляют собой некие электронные схемы, содержащие один или несколько *регистров ввода-вывода* и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора. Портами также называют *устройства стандартного интерфейса*: последовательный, параллельный и игровой порты.

Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. *Параллельный порт* получает и посылает данные побайтно.

К *последовательному* порту подсоединяют медленно действующие или достаточно удалённые устройства (мышь и модем). К *параллельному* порту подсоединяют более "быстрые" устройства (принтер и сканер). Через *игровой* порт подсоединяется джойстик.

Основные электронные компоненты, определяющие архитектуру процессора, размещаются на основной плате компьютера, которая называется *системной* или *материнской* (MotherBoard). Контроллеры и адаптеры дополнительных устройств, либо сами эти устройства, выполняются в виде *плат расширения* и подключаются к шине с помощью *разъёмов расширения*, называемых *слотами расширения* (англ. *slot* – щель, паз).

8.Устройства ввода – вывода

Компьютер обменивается информацией с внешним миром с помощью **периферийных устройств**. Только благодаря периферийным устройствам человек может взаимодействовать с компьютером, а также со всеми подключенными к нему устройствами. Любое подключенное периферийное устройство в каждый момент времени может быть или занято выполнением порученной ему работы или пребывать в ожидании нового задания. Влияние скорости работы периферийных устройств на эффективность работы с компьютером не меньше, чем скорость работы его центрального процессора. Скорость работы внешних устройств от быстродействия процессора не зависит. Наиболее распространенные периферийные устройства приведены на рисунке:

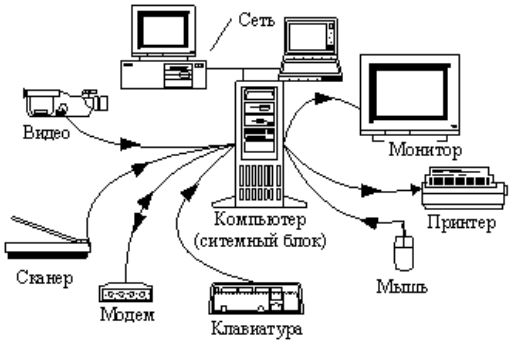


Рис.43.

Периферийные устройства делятся на устройства ввода и устройства вывода. **Устройства ввода** преобразуют информацию в форму понятную машине, после чего компьютер может ее обрабатывать и запоминать. **Устройства вывода** переводят информацию из машинного представления в образы, понятные человеку.

Самым известным устройством ввода информации является клавиатура (keyboard) – это стандартное устройство, предназначенное для ручного ввода информации. Работой клавиатуры управляет контроллер клавиатуры, расположенный на материнской плате и подключаемый к ней через разъем на задней панели компьютера. При нажатии пользователем клавиши на клавиатуре, контроллер клавиатуры преобразует код нажатой клавиши в соответствующую последовательность битов и передает их компьютеру. Отображение символов, набранных на клавиатуре, на экране компьютера называется эхом. Обычная современная клавиатура имеет, как правило, 101-104 клавиши, среди которых выделяют алфавитно-цифровые клавиши, необходимые для ввода текста, клавиши управления курсором и ряд специальных и управляющих клавиш. Существуют беспроводные модели клавиатуры, в них связь клавиатуры с компьютером осуществляется посредством инфракрасных лучей.

К манипуляторам относят устройства, преобразующие движения руки пользователя в управляющую информацию для компьютера. Среди манипуляторов выделяют мыши, трекболы, джойстики.

Дигитайзер – это устройство для ввода графических данных, таких как чертежи, схемы, планы и т. п. Он состоит из планшета, соединенного с ним визира или специального карандаша. Перемещая карандаш по планшету, пользователь рисует изображение, которое выводится на экран.

Сканер – устройство ввода графических изображений в компьютер. В сканер закладывается лист бумаги с изображением. Устройство считывает его и пересылает компьютеру в цифровом виде.

После ввода пользователем исходных данных компьютер должен их обработать в соответствии с заданной программой и вывести результаты в форме, удобной для восприятия пользователем или для использования другими автоматическими устройствам посредством **устройств вывода**.

Выводимая информация может отображаться в графическом виде, для этого используются **мониторы, принтеры** или **плоттеры.** Информация может также воспроизводиться в виде звуков с помощью **акустических колонок** или **головных телефонов**, регистрироваться в виде тактильных ощущений в технологии виртуальной реальности, распространяться в виде управляющих сигналов устройства автоматики, передаваться в виде электрических сигналов по сети.

Монитор (дисплей) является основным устройством вывода графической информации.

Для получения копий изображения на бумаге применяют принтеры, которые классифицируются:

- по способу получения изображения: литерные, матричные, струйные, лазерные и термические;
- по способу формирования изображения: последовательные, строчные, страничные;
- по способу печати: ударные, безударные;
- по цветности: чёрно-белые, цветные.

Плоттер (графопостроитель) – это устройство для отображения векторных изображений на бумаге, кальке, пленке и других подобных материалах.

Вопросы:

1. Техническая реализация информационных процессов?
2. Какова структура современной ЭВМ?
3. Назовите виды памяти?
4. В чем заключается магистрально-модульный принцип построения ПК?
5. Укажите преимущества открытой архитектуры?
6. Что такое конфигурация компьютера?
7. Какие внешние устройства можно подключить к компьютеру?
8. Что такое флэш- память?
9. Что такое устройства ввода-вывода?

Список литературы:

Печатные издания:

1. Трофимов В.В. Информатика в 2 т. Том 1: учебник для среднего профессионального образования/В.В.Трофимов;подредакциейВ.В.Трофимова.-3-еизд.,перераб.идоп.-М: Юрайт,2022.-553с.//URL:https://urait.ru/bcode/491211

2. ТрофимовВ.В.Информатикав2т.Том2:учебникдлясреднегопрофессионального образования/В.В.Трофимов;ответственный редакторВ.В.Трофимов.-3-еизд.,перераб.и доп.-М:Юрайт,2022.-406с.//URL:https://urait.ru/bcode/491213

3. Михеева Е.В. Информатика: учебник учебн. пособие для студ. учреждений сред. проф. образования / Е.В.Михеева, О.И.Титова. –2-е изд., стер. – М.: Издательский центр «Академия», 2018. –400 с.

4. Михеева Е.В. Информатика. Практикум: учебн. пособие для студ. учреждений сред. проф. образования / Е.В.Михеева, О.И.Титова. – М.: Издательский центр «Академия», 2017. – 224 с.

5. Цветкова М. С. Информатика и ИКТ: практикум для профессий и специальностейестественно-научногоигуманитарногопрофилей:учеб. пособие для студ. СПО 3-е изд., стер. М.: Академия, 2017.- 224с.

6. ЦветковаМ.С.Информатика:учебникдлястуд.СПО.-5-еизд.,стерМ.: Академия, 2018.-352с.

Электронныеиздания(электронныересурсы):

1. Единаяколлекцияцифровыхобразовательных ресурсов,http://www.school-collection.edu.ru

2. Единоеокнодоступак образовательным ресурсам,http://www.window.edu.ru

3. Мега энциклопедия Кирилла и Мефодия, разделы «Наука/Математика. Кибернетика»и«Техника/КомпьютерыиИнтернет»,http://www.megabook.ru

4. Открытаяэлектроннаябиблиотека«ИИТОЮНЕСКО» поИКТвобразовании, http://ru.iite.unesco.org/publications

5. Открытыеинтернет-курсы «Интуит»покурсу«Информатика», http://www.intuit.ru/studies/courses

6. ПорталСвободногопрограммногообеспечения,http://www.freeschool.altlinux.ru

7. Российский портал открытого образования, http://www.edu.ru