

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

**Кафедра «Организация работы с молодежью»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**Б1.В.ДВ.04.01 ИНФОРМАТИКА**

**Направление подготовки (специальность) «Управление персоналом»**

**Профиль образовательной программы «Управление персоналом организации»**

**Форма обучения очная**

Оренбург 201- г.

## СОДЕРЖАНИЕ

1. КОНСПЕКТ ЛЕКЦИЙ .....	3
1. 1 Лекция № 1 (2 часа).....	3
Тема: Основные понятия информации. Классификация информации .....	3
1. 2 Лекция № 2 (2 часа).....	14
Тема: «Представление информации в ПК» .....	14
1. 3 Лекция № 3 (2 часа).....	16
Тема: «Понятие и основные виды архитектуры ПК. Алгебра логики» .....	16
1. 4 Лекция № 4 (2 часа).....	23
Тема: «Программное обеспечение (ПО). Классификация ПО» .....	23
1. 5 Лекция № 5 (2 часа).....	24
Тема: «Офисное программное обеспечение».....	24
1. 6 Лекция №6 (2 часа).....	29
Тема: «Компьютерные вычислительные сети» .....	29
1. 7 Лекция №7 (2 часа).....	35
Тема: «Защита информации в локальных и глобальных компьютерных сетях».....	35
1. 8 Лекция №8 (2 часа).....	41
Тема: «Алгоритмизация и программирование».....	41
1. 9 Лекция №9 (2 часа).....	42
Тема: «Понятие о структурном программировании».....	42
2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ СЕМИНАРСКИХ ЗАНЯТИЙ .....	49
2.1 Семинарское занятие №1 (2 часа) .....	50
Тема: «Основные понятия информации. Классификация информации» .....	50
2.2 Семинарское занятие №2 ( 2 часа) .....	50
Тема: «Представление информации в ПК» .....	50
2.3 Семинарское занятие №3 ( 2 часа) .....	51
Тема: «Понятие и основные виды архитектуры ПК. Алгебра логики» .....	51
2.4 Семинарское занятие №4 ( 2 часа) .....	51
Тема: «Программное обеспечение (ПО). Классификация ПО» .....	51
2.5 Семинарское занятие №5 ( 2 часа) .....	52
Тема: «Офисное программное обеспечение».....	52
2.6 Семинарское занятие №6 ( 2 часа) .....	53
Тема: «Компьютерные вычислительные сети» .....	53
2.7 Семинарское занятие №7 ( 2 часа) .....	53
Тема: «Защита информации в локальных и глобальных компьютерных сетях».....	53
2.8 Семинарское занятие №8( 2 часа) .....	54
Тема: «Алгоритмизация и программирование».....	54
2.9 Семинарское занятие №9 ( 2 часа) .....	54
Тема: «Понятие о структурном программировании».....	54

# **1. КОНСПЕКТ ЛЕКЦИЙ**

## **1. 1 Лекция № 1 (2 часа).**

### **Тема: Основные понятия информации. Классификация информации**

#### **1.1.1 Вопросы лекции:**

- 1. Современные информационные системы и технологии.**
- 2. Информационная культура.**
- 3. Правовое регулирование рынка информационных продуктов и услуг.**

#### **1.1.2 Краткое содержание вопросов:**

##### **1. Современные информационные системы и технологии**

Деятельность отдельных людей, групп, коллективов и организаций зависит от их информированности и способности эффективно использовать имеющуюся информацию. Прежде чем предпринять какие-то действия необходимо провести большую работу по сбору и переработке информации, ее осмыслению и анализу. Отыскание рациональных решений в любой сфере требует обработки больших объемов информации, что невозможно без привлечения специальных технических средств. Современные сферы деятельности все больше нуждаются в соответствующем информационном обслуживании.

Происходит информационный взрыв и вместе с тем информационный кризис. Проявляются противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими мощными потоками и массивами хранящейся информации. Циркулирует большое количество избыточной информации, которая затрудняет восприятие информации, полезной для потребителя. Существует проблема отбора качественной и достоверной информации. Общество создает экономические, политические, социальные барьеры, препятствующие распространению информации. В мире накоплен громадный информационный потенциал, но люди не могут им воспользоваться в полном объеме в силу ограниченности своих возможностей.

Необходимо подготовить человека к быстрому восприятию и обработке больших объемов информации, овладению им современными средствами, методами и технологией работы. Новые условия порождают зависимость информированности одного человека от информации, приобретенной другими людьми. Поэтому уже недостаточно уметь самостоятельно осваивать и накапливать информацию, а надо научиться такой технологии работы с информацией, когда подготавливаются и принимаются решения на основе коллективного знания. Залог успеха - в умении извлекать информацию из разных источников (из периодической печати, электронных коммуникаций), представлять ее в понятном виде и уметь эффективно использовать.

Информатизация характеризуется возрастанием объема и роли информации и широким использованием технических средств для производства, переработки, хранения, распределения и использования информации.

Информационные ресурсы - это знания, подготовленные людьми для социального использования в обществе и зафиксированные на материальных носителях. Как и при использовании традиционных ресурсов и продуктов, важно знать: где находятся информационные ресурсы, сколько они стоят, кто ими владеет, кто в них нуждается, насколько они доступны. Совокупность средств, методов и условий, позволяющих использовать информационные ресурсы, составляет информационный потенциал общества.

В настоящее время в России быстрыми темпами идет формирование рынка информационных продуктов и услуг. Развитие рыночных отношений в информационной деятельности обострило проблему защиты информации как объекта интеллектуальной собственности и имущественных прав. Приняты Законы: "Об информации, информатизации и защите информации"; "Об авторском праве и смежных правах"; "О правовой охране программ для ЭВМ и баз данных". В Уголовный кодекс включена глава о компьютерных преступлениях.

Информатика - это область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и их взаимодействием со средой применения. Одна из ее главных задач как фундаментальной науки - выяснение, что такое информационные системы, какое место они занимают, какую должны иметь структуру, как функционируют, какие общие закономерности им свойственны.

Под системой понимают "множество элементов, находящихся в отношениях и связях друг с другом, которое образует определенную целостность, единство". Системный подход применим к любому объекту, рассматриваемому одновременно и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов.

Информационная система (ИС) - взаимосвязанная совокупность средств, методов и персонала, используемых для сбора, хранения, обработки и выдачи информации в интересах достижения поставленной цели. Компьютеры, оснащенные специализированными программными средствами, служат технической базой и инструментом для информационных систем. Т.е. современная информационная система - человеко-компьютерная.

Структура информационной системы - это взаимосвязанная совокупность ее частей, называемых обеспечивающими подсистемами. Среди них обычно выделяют

информационное, техническое, математическое, программное, организационное и правовое обеспечение.

Информационное обеспечение - совокупность единой классификации и кодирования информации, унифицированных систем документации, схем информационных потоков.

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

Математическое и программное обеспечение - совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

Организационное обеспечение - совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Правовое обеспечение - совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Различают три типа задач, для которых создаются информационные системы: структурированные (формализуемые), неструктурированные (неформализуемые) и частично структурированные.

Структурированная (формализуемая) задача - в которой известны все ее элементы и взаимосвязи между ними. В такой задаче удастся выразить ее содержание в форме математической модели, имеющей точный алгоритм решения. Подобные задачи обычно приходится решать многократно и они носят рутинный характер. Использование информационной системы для решения структурированных задач обеспечивает полную автоматизацию их решения.

Неструктурированная (неформализуемая) задача - в которой невозможно выделить элементы и установить связи между ними. Решение этих задач связано с большими трудностями из-за невозможности создания математического описания и разработки алгоритма. Возможности использования информационной системы невелики. Решение в таких случаях принимается человеком на основе своего опыта (из эвристических соображений) и косвенной информации из разных источников.

На практике сравнительно немного полностью структурированных или совершенно неструктурированных задач. В большинстве задач известна лишь часть их элементов и связей между ними. Такие задачи называются частично структурированными. В этих

условиях можно создать информационную систему. Получаемая в ней информация анализируется человеком, который играет определяющую роль. Такие информационные системы являются автоматизированными, т.к. в их функционировании принимает участие человек.

ИС классифицируются по различным основаниям.

Информационные системы, используемые для решения частично структурированных задач, подразделяются на два вида:

- создающие управленческие отчеты и ориентированные главным образом на обработку данных (поиск, сортировку, агрегатирование, фильтрацию);
- разрабатывающие возможные альтернативы решения.

Информационные системы, создающие управленческие отчеты, обеспечивают информационную поддержку пользователя, предоставляют доступ к информации в базе данных и ее частичную обработку. Информационные системы, разрабатывающие альтернативные решения, могут быть модельными или экспертными.

Тип информационной системы зависит от того, чьи интересы она обслуживает и на каком уровне управления. Информационные системы специалистов помогают в работе с данными, повышают продуктивность и производительность работы. Задача подобных ИС - интеграция новых сведений и помощь в обработке документов. По функциональному назначению для различных категорий специалистов выделяют: ИС офисной автоматизации; ИС менеджеров среднего звена; управленческие ИС; ИС поддержки принятия решений.

ИС офисной автоматизации вследствие своей простоты и многопрофильности активно используются работниками любого организационного уровня. Наиболее часто их применяют работники средней квалификации. Основная цель - обработка данных, повышение эффективности их работы и упрощение канцелярского труда. ИС офисной автоматизации в основном охватывают управление документацией, коммуникации и т.п.

ИС менеджеров среднего звена - для мониторинга (постоянного слежения), контроля, принятия решений и администрирования. Некоторые ИС обеспечивают принятие нетривиальных решений. В случае, когда требования к информационному обеспечению определены не строго, они способны отвечать на вопрос: "что будет, если ...?" На этом уровне выделяют два типа ИС: управленческие (для менеджмента) и системы поддержки принятия решений.

Управленческие ИС имеют небольшие аналитические возможности. Они полезны для ежедневной (еженедельной и т.д.) информации о состоянии дел и периодического составления сводных типовых отчетов.

Системы поддержки принятия решений обслуживают частично структурированные задачи, результаты которых трудно спрогнозировать заранее. Они имеют более мощный аналитический аппарат с несколькими моделями. Эти системы полезны всем, кто принимает решения.

По характеру использования информации различают информационно-поисковые и информационно-решающие системы.

Информационно-поисковые системы вводят, систематизируют, хранят, выдают информацию по запросам пользователя без сложных преобразований данных.

Информационно-решающие системы осуществляют все операции переработки информации по определенному алгоритму. Среди них проводят классификацию по степени воздействия выработанной информации на процесс принятия решений и выделяют два класса: управляющие и советующие.

Управляющие ИС вырабатывают информацию, на основании которой человек принимает решение. Для этих систем характерны задачи расчетного характера и обработка больших объемов данных.

Советующие ИС вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, для них характерна обработка знаний, а не данных.

Технология при переводе с греческого (techne) означает искусство, мастерство, умение, т.е. процессы. Под процессом понимают определенную последовательность действий, направленных на достижение поставленной цели. Процесс определяется выбранной стратегией и реализуется совокупностью различных средств и методов.

Информационная технология - процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества (информационного продукта). Современный этап развития информационной технологии характеризуется внедрением персональных компьютеров и применением телекоммуникационных средств связи.

Принципы компьютерной информационной технологии: интерактивный (диалоговый) режим работы с компьютером; интеграция (стыковка, взаимосвязь) программных средств; гибкость изменения задач и данных.

Технологический процесс переработки информации представляется в виде иерархической структуры. Его уровни (снизу вверх): элементарные манипуляции; действия; операции; этапы. Освоение информационной технологии и ее использование

сводятся к следующему. Нужно сначала овладеть набором элементарных манипуляций, число которых ограничено. Из их различных комбинаций составляются действия, а из действий, также в разных комбинациях, составляются операции, которые определяют технологический этап. Совокупность технологических этапов образует технологический процесс (технологию).

Информационные продукты, как никакие другие виды материальных товаров, имеют чрезвычайно высокую скорость сменяемости новыми видами или версиями. Периоды сменяемости колеблются от нескольких месяцев до одного года. Неудачи с внедрением информационной технологии обычно связывают с несовершенством технических средств, тогда как основной причиной неудач является отсутствие или слабая проработанность методологии использования информационной технологии.

Информационные технологии классифицируются по типам ИС.

Информационная технология обработки данных предназначена для решения хорошо структурированных задач, по которым имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки. Эта технология применяется на уровне исполнительской деятельности персонала невысокой квалификации целях автоматизации некоторых рутинных постоянно повторяющихся операций. Основные компоненты: сбор, обработка, хранение данных, создание отчетов (документов).

Обработка данных включает типовые операции: классификация или группировка; сортировка, с помощью которой упорядочивается последовательность записей; вычисления, включающие математические и логические операции; укрупнение (агрегатирование), служащее для уменьшения количества данных и реализуемое в форме расчетов итоговых или средних значений.

Информационная технология управления ориентирована на работу при худшей структурированности решаемых задач. Информация должна быть представлена в агрегированном виде так, чтобы просматривались тенденции изменения данных, причины возникших отклонений и возможные решения. Решаются следующие задачи: оценка планируемого состояния объекта управления; оценка отклонений от планируемого состояния; выявление причин отклонений; анализ возможных решений и действий.

Информационная технология автоматизированного офиса - для организации и поддержки коммуникационных процессов как внутри фирмы, так и с внешней средой на базе компьютерных сетей и других современных средств передачи и работы с



информацией. Основные компоненты: текстовый и табличный процессоры, электронная почта и т.п.

Информационная технология поддержки принятия решений организует взаимодействие человека и компьютера. Выработка решений происходит в результате циклического процесса, в котором участвуют: система поддержки принятия решений в роли вычислительного звена и объекта управления; человек как управляющее звено, задающее входные данные и оценивающее полученный результат вычислений на компьютере. Ее отличительные характеристики: ориентация на решение плохо структурированных (слабо формализованных) задач; - сочетание традиционных методов доступа и обработки компьютерных данных с возможностями математического моделирования; направленность на непрофессионального пользователя; высокая адаптивность - приспособляемость к особенностям используемого технического и программного обеспечения, требованиям пользователя.

В состав системы поддержки принятия решений входят три главных компонента: база данных, база моделей и программная подсистема, которая состоит из системы управления базой данных (СУБД), системы управления базой моделей (СУБМ) и системы управления интерфейсом (взаимодействием между пользователем и компьютером).

Существует множество типов моделей и способов их классификации: по цели использования, области возможных приложений, способу оценки переменных и т.п. По цели использования модели подразделяются на оптимизационные и описательные, описывающие поведение соответствующей системы.

По способу оценки модели классифицируются на детерминистские, использующие оценку переменных одним числом при конкретных значениях исходных данных, и стохастические, оценивающие переменные несколькими параметрами, когда исходные данные заданы вероятностными характеристиками. Детерминистские модели более популярны, чем стохастические, они менее дорогие, их легче строить и использовать, с их помощью получается достаточная информация для принятия решения.

По области возможных приложений модели разделяются на специализированные, используемые только одной системой, и универсальные - для использования несколькими системами. Специализированные модели более дорогие, они обычно применяются для описания уникальных систем и обладают большей точностью.

Система управления интерфейсом во многом определяет эффективность и гибкость информационной технологии. Интерфейс характеризует: язык пользователя; язык сообщений компьютера, организующий диалог на экране дисплея; знания пользователя.

Язык пользователя - это те действия, которые пользователь производит в отношении системы путем использования возможностей клавиатуры, "мыши" и т.п. Наиболее прост язык пользователя в форме входных и выходных документов. Выведя на экран входную форму (документ), пользователь заполняет его необходимыми данными и вводит в компьютер. Система поддержки принятия решений производит необходимый анализ и выдает результаты в виде выходного документа установленной формы. Значительно возрастает популярность визуального интерфейса. С помощью манипулятора "мышь" пользователь выбирает представленные ему в форме картинок на экране объекты и команды, реализуя таким образом свои действия. Ожидается появление систем поддержки принятия решений, использующих речевой ввод информации.

Язык сообщений - это то, что пользователь видит на экране дисплея (символы, графика, цвет), данные, полученные на принтере, звуковые выходные сигналы и т.п.

Важный показатель эффективности используемого интерфейса - форма диалога между пользователем и системой. Распространены следующие формы диалога: запросно-ответный режим, командный режим, режим меню, режим заполнения пропусков в выражениях, предлагаемых компьютером. Каждая форма в зависимости от типа задачи, особенностей пользователя и принимаемых решений имеет свои достоинства и недостатки.

Знания пользователя - это то, что пользователь должен знать, работая с системой. К ним относятся не только план действий, находящийся в голове пользователя, но и учебники, инструкции, справочные данные, выдаваемые компьютером.

Наибольший прогресс среди компьютерных ИС отмечен в области разработки экспертных систем, основанных на использовании искусственного интеллекта. Экспертные системы дают возможность менеджеру или специалисту получать консультации экспертов по любым вопросам, о которых этими системами накоплены знания. Основные компоненты информационной технологии, используемой в экспертной системе: интерфейс пользователя, база знаний, интерпретатор, модуль создания системы.

Сходство информационных технологий, используемых в экспертных системах и системах поддержки принятия решений, состоит в том, что обе они обеспечивают высокий уровень поддержки принятия решений. Однако имеются и существенные различия. Решение проблемы в рамках систем поддержки принятия

решений отражает уровень ее понимания пользователем и его возможности получить и осмыслить решение. Технология экспертных систем, наоборот, предлагает пользователю принять решение, превосходящее его возможности.

Другое отличие указанных технологий выражается в способности экспертных систем пояснять свои рассуждения в процессе получения решения. Часто эти пояснения оказываются более важными для пользователя, чем само решение. Третье отличие связано с использованием нового компонента информационной технологии - знаний.

## **2. Информационная культура**

В настоящее время информационную культуру все чаще трактуют как особый феномен информационного общества. В зависимости от объекта рассмотрения стали выделять информационную культуру общества, информационную культуру отдельных категорий потребителей информации (например, детей или юристов) и информационную культуру личности.

Информационная культура в широком смысле – это совокупность принципов и механизмов, обеспечивающих взаимодействие этнических и национальных культур, их соединение в общий опыт человечества; в узком смысле слова – оптимальные способы обращения с информацией и представление ее потребителю для решения теоретических и практических задач; механизмы совершенствования технических сред производства, хранения и передачи информации; развитие системы обучения, подготовки человека к эффективному использованию информационных средств и информации.

Информационная культура личности – одна из составляющих общей культуры человека, совокупность информационного мировоззрения и системы знаний и умений, обеспечивающих целенаправленную самостоятельную деятельность по оптимальному удовлетворению индивидуальных информационных потребностей с использованием как традиционных, так и новых информационных технологий.

Сопоставление понятий «информационная грамотность» и «информационная культура личности» свидетельствует об их значительном сходстве. Оба понятия характеризуют сложный, многоуровневый и многоаспектный феномен взаимодействия человека и информации. В составе объема обоих понятий выделяется много компонентов: от умения вести поиск информации, анализировать и критически оценивать найденные источники информации, до их творческого использования в целях решения многообразных задач, возникающих в учебной, профессиональной, досуговой или иной деятельности.

Вместе с тем, концепция информационной культуры личности шире, чем концепция информационной грамотности. В отличие от информационной грамотности, она включает

такой компонент, как информационное мировоззрение, предполагающее обязательную мотивацию личности на необходимость специальной информационной подготовки.

Концепция информационной культуры позволяет отнести информационную подготовку личности к сфере культуры, что дает возможность обеспечить синтез и целостность традиционной книжной (библиотечной) и новой (компьютерной) информационных культур, избежать в информационном обществе конфронтации двух полярных культур – технократической и гуманитарной.

В целом, различия между развиваемой в России концепцией формирования информационной культуры личности и международной концепцией информационной грамотности не носят принципиального характера; они лишь отражают стремление российских ученых и практиков сочетать достижения международной теории и практики с традициями национальной культуры и образования, имеющимся опытом российских библиотек и учреждений образования.

Концепция информационной культуры личности, разработанная в НИИ информационных технологий социальной сферы КемГУКИ, получила высокую оценку на III сессии Межправительственного совета Программы ЮНЕСКО «Информация для всех» и была предложена в качестве одного из приоритетных направлений деятельности Программы на 2006-2007 гг.

### **3. Правовое регулирование рынка информационных продуктов и услуг**

Производство услуг, информационных и программных продуктов приводит в движение огромные финансовые потоки, создает массу рабочих мест и способствует прогрессу во всех областях деятельности, не вызывая проблем в экологии, энергетике и других сферах. Компьютерная программа фактически идеальна – можно считать, что в ней нет ничего материального, только одна ее стоимость.

Информационные ресурсы представляют собой сосредоточенный в компактной форме накопленный потенциал фирмы и имеют соответствующую стоимость. Эти ценности обязательно кому-то принадлежат или, по крайней мере, должны принадлежать. В противном случае они будут использоваться произвольно или неправомерно, по существу расхищаться, как всякие другие ценности в подобных условиях.

Производство машинной информации имеет массовый характер, представляет собой настоящую индустрию и нуждается в серьезном правовом обеспечении. В связи с этим всем субъектам этого процесса необходимо иметь четко определенные права и обязательства по отношению как к самой информации, так и к другим субъектам и их правам.

В зарубежных странах уже достаточно давно возникла специальная область права – компьютерное право. Постепенно компьютерное право приобрело характер более широкой области – информационного права. Во всех передовых по этому профилю странах существуют правительственные программы развития норм права и определенная политика в сфере защиты национальных информационных ресурсов.

Законодательство Российской Федерации по вопросам информатики и информационных ресурсов также отражает постепенное формирование адекватной правовой базы регулирования отношений в сфере информатизации. Принят ряд указов, постановлений, законов, таких, как:

- "Об информации, информатизации и защите информации";
- "Об авторском праве и смежных правах";
- "О правовой охране программ для ЭВМ и баз данных";
- "О правовой охране топологий интегральных схем".

Базовым юридическим документом, открывающим путь к принятию дополнительных нормативных законодательных актов, является закон "Об информации, информатизации и защите информации". В законе определены цели и основные направления государственной политики в сфере информатизации. Закон создает условия для включения России в международный информационный обмен, предотвращает бесхозяйственное отношение к информационным ресурсам и информатизации, обеспечивает информационную безопасность и права юридических и физических лиц на информацию. В нем определяются комплексное решение проблемы организации информационных ресурсов, правовые положения по их использованию и предлагается рассматривать информационные ресурсы в двух аспектах:

- как материальный продукт, который можно покупать и продавать;
- как интеллектуальный продукт, на который распространяется право интеллектуальной собственности, авторское право.

Закон закладывает юридические основы гарантий прав граждан на информацию. Он направлен на урегулирование важнейшего вопроса экономической реформы – формы, права и механизма реализации собственности на накопленные информационные ресурсы и технологические достижения. Обеспечена защита собственности в сфере информационных систем и технологий, что способствует формированию цивилизованного рынка информационных ресурсов, услуг, систем, технологий, средств их обеспечения.

## 1. 2 Лекция № 2 (2 часа).

### Тема: «Представление информации в ПК»

#### 1.2.1 Вопросы лекции:

1. Измерение информации. Системы счисления.
2. Кодирование информации.
3. Представление информации в ЭВМ.

#### 1.2.2 Краткое содержание вопросов:

##### 1. Измерение информации. Системы счисления

Понятие **информации** является центральным для информатики. Точное выяснение понятия “информация” существенно необходимо для глубокого понимания систем обработки информации.

**Классификация** – система распределения объектов по классам в соответствии с определенными признаками.

Существуют три системы классификации:

1) Иерархическая – построение древовидной структуры только с вертикальными связями. На каждом уровне свой признак классификации. Количество уровней характеризуют **глубину** классификации.

2) Фасетная - признаком классификации является **фасет**

Единица измерения информации была определена в науке под названием «теория информации». Информацию измеряют в **битах**. Бит – это один разряд в ячейке компьютерной памяти. В один разряд можно вписать только одну цифру (0 или 1). Это очень маленькая величина, поэтому используют величину в 8 раз больше – **байт**. Т.о. байт - это 8 бит

##### 2. Кодирование информации

Кодирование информации – это процесс формирования определенного представления информации.

В более узком смысле под термином «кодирование» часто понимают переход от одной формы представления информации к другой, более удобной для хранения, передачи или обработки.

Компьютер может обрабатывать только информацию, представленную в числовой форме. Вся другая информация (звуки, изображения, показания приборов и т. д.) для обработки на компьютере должна быть преобразована в числовую форму. Например, чтобы перевести в числовую форму музыкальный звук, можно через небольшие

промежутки времени измерять интенсивность звука на определенных частотах, представляя результаты каждого измерения в числовой форме. С помощью компьютерных программ можно преобразовывать полученную информацию, например «наложить» друг на друга звуки от разных источников.

Аналогично на компьютере можно обрабатывать текстовую информацию. При вводе в компьютер каждая буква кодируется определенным числом, а при выводе на внешние устройства (экран или печать) для восприятия человеком по этим числам строятся изображения букв. Соответствие между набором букв и числами называется кодировкой символов.

Как правило, все числа в компьютере представляются с помощью нулей и единиц (а не десяти цифр, как это привычно для людей). Иными словами, компьютеры обычно работают в двоичной системе счисления, поскольку при этом устройства для их обработки получаются значительно более простыми.

### **3. Представление информации в ЭВМ**

Компьютерная техника уже давно стала частью обыденной жизни человека. Бесконечно число электроники окутывает наше пространство, а радиоволны пронзают все оставшееся. Не смотря на то, что интеллектуальная техника сильно эволюционировала за последние пятьдесят лет, основные принципы ее функционирования остались теми же. Первым делом необходимо рассмотреть, чем же оперирует ЭВМ. Правильно, информация – важнейший элемент, циркулирующий в любом интеллектуальном устройстве. Информация представляется в виде сигналов дискретных или аналоговых. Аналоговые сигналы используются не часто. Высокоэффективные современные ЭВМ работают с дискретными сигналами, что гарантирует надежность, высокую степень интегрированности и совместимости. Дискретные сигналы в большинстве случаев интерпретируются как цифровые. Информация в ЭВМ представляется в виде цифр, каждой букве, символу, команде сопоставлен определенный набор цифр. На физическом уровне цифры представлены только 0 (отсутствие напряжения) и 1 (наличие напряжения). Таким образом основной системой счисления ЭВМ является двоичная, т.е. нули и единицы. При выводе информации она переводится не редко в десятичную наиболее привычную людям. Цифровой сигнал (информация) последовательно (параллельно) подается на вход устройства, кодируется определенным методом и воспринимается оборудованием. Для чтения такой информации существуют триггеры – элементы памяти, которые могут принимать значение один и ноль, а так же высокоимпедансное состояние, которое является отличным от нуля и единицы. Выстраивание таких триггеров в цепочки,

обеспечивает формирование памяти, которая структурируется в зависимости от необходимого объема и ее типа. Вопреки распространенному мнению, что компьютер читает только нули и единицы, это не совсем так. Большинство компьютеров имеет свою систему интерпретации, которая переводит шестнадцатеричную систему в двоичную, но на этапе ввода команды, текст остается в исходной системе исчисления. Все команды шестнадцатеричного кода отправляются параллельным методом, ширина канала которого, ограничивается физической шириной шины данных. На этом этапе информация предоставляется «словами», каждое из которых может быть 8ми, 16ти, 32х, 64х разрядное и более. Соответственно, чем более разрядность физической шины данных, тем более эффективна работа системы. Центральный процессор или контроллер обрабатывает информацию не сразу, а дает возможность отправиться ей в предварительный кэш памяти, так данные укладываются в стек по системе *fifo* или *lifo*, а уже после этого, в зависимости от приоритета команды, данные поступают в процессор, где обрабатываются. После этого интерпретатор переводит информацию в шестнадцатеричный код и передает обратно в отдельные узлы системы. Благодаря этим манипуляциям, обеспечивается быстрый обмен информацией, легкое трансформирование данных любого стандарта и взаимодействие центральных узлов и периферии.

### **1. 3 Лекция № 3 (2 часа).**

**Тема: «Понятие и основные виды архитектуры ПК. Алгебра логики»**

#### **1.3.1 Вопросы лекции:**

- 1. Логические выражения и функции.**
- 2. Упрощение логических выражений**

#### **1.3.2 Краткое содержание вопросов:**

##### **1. Логические выражения и функции.**

**Выражение** это формальное правило для вычисления некоторого значения. Выражение строится как совокупность операндов, объединенных знаками операций, выполнение которых приводит к вычислению значения выражения. Главное свойство выражений - возможность иметь значение.

В зависимости от типа значения выражения можно разделить на три вида:

- численные (арифметические);
- логические;
- символьные.

Рассмотри правила построения выражений каждого из этих видов.



### **Численные (арифметические) выражения**

Представляют собой совокупность имен переменных, констант, функций, а также других выражений числовых типов, объединенных знаками арифметических операций. В Паскале предусмотрены шесть арифметических операций:

+	сложение (унарный плюс)
-	вычитание (унарный минус)
*	умножение
/	деление
<b>div</b>	целочисленное деление
<b>mod</b>	остаток от целочисленного деления

Операция возведения в степень в языке не предусмотрена.

В сложных выражениях порядок выполнения операций определяется правилами приоритета:

I-й (высший) приоритет	унарные минус и плюс
II-й приоритет	* и /
III-й приоритет	+ и -

Для изменения порядка выполнения операций используют круглые скобки ( ).

Тип значения выражения зависит типов операндов и от выполняемой операции.

- Если все операнды целочисленные, то операции сложения, вычитания, умножения, целочисленного деления дают значение выражения также целочисленного типа. Операция деления "/" дает вещественный результат! Если в выражение присутствуют целочисленные операнды, имеющие различные типы (например, Byte и Integer), то значение выражения будет иметь тип с наибольшим диапазоном (в данном случае Integer).

- Если среди операндов хоть один имеет вещественный тип, то значение выражения будет также вещественным.

При построении арифметических выражений можно использовать различные встроенные математические функции. Ниже в таблице приведены их имена, назначение и характеристики.

Вызов	Тип	Тип	Назначение
-------	-----	-----	------------

функции	аргумента	результата	
<b>Abs(x)</b>	целый вещественный	целый вещественный	Абсолютное значение <b>x</b>
<b>Pi</b>	-	вещественный	Значение числа <b>Π</b>
<b>Sin(x)</b>	целый вещественный	вещественный	Синус <b>x</b> радиан
<b>Cos(x)</b>	целый вещественный	вещественный	Косинус <b>x</b> радиан
<b>ArcTan(x)</b>	целый вещественный	вещественный	Арктангенс <b>x</b> радиан
<b>Sqrt(x)</b>	целый вещественный	вещественный	Квадратный корень из <b>x ≥ 0</b>
<b>Sqr(x)</b>	целый вещественный	целый вещественный	Значение квадрата <b>x</b>
<b>Exp(x)</b>	целый вещественный	вещественный	Значение <b>e</b> в степени <b>x</b>
<b>Ln(x)</b>	целый вещественный	вещественный	Натуральный логарифм <b>x</b> , <b>x &gt; 0</b>
<b>Trunc(x)</b>	целый вещественный	LongInt	Целую часть значения <b>x</b>
<b>Frac(x)</b>	целый вещественный	вещественный	Дробную часть значения <b>x</b>
<b>Int(x)</b>	целый вещественный	вещественный	Целую часть значения <b>x</b>

<b>Round(x)</b>	целый вещественный	LongInt	Округляет x до ближайшего целого
<b>Random</b>	-	вещественный	Случайное число от 0 до 1
<b>Random(X)</b>	Word	Word	Случайное число от 0 до X
<b>Odd(x)</b>	целый	логический	Проверяет нечетность значения x

### Примеры записи арифметических выражений

$x+5$      $(a+b)/n$      $a+b/n$      $\sin(x)*\text{sqr}(x)$

$1/(1+\text{abs}(\sin(x)))$      $\text{Trunc}((\text{max}-\text{min})/dx)$

$23.05*(x-0.01)-\text{sqr}(x+1)$

### Логические выражения

**Условное выражение** - совокупность переменных и констант простых типов, объединенных знаками операций сравнения. Результат вычисления условного выражения - одно из двух логических значений типа [Boolean](#): **True** (истина) или **False** (ложь).

В языке Паскаль определены шесть операций сравнения (все они относятся к четвертому (низшему) приоритету):

<	меньше
>	больше
<=	меньше или равно
>=	больше или равно
=	равно
<>	не равно

Типы операндов должны быть совместимыми по присваиванию.

### Примеры условных выражений (условий):

$x<2$      $ch='Y'$      $a+b<>0$      $\sin(x)>=0.5$

**Важно:** При сравнении вещественных чисел следует помнить, что в следствие неточности их представления в памяти компьютера (в виду неизбежности округления) стоит избегать попыток определения строгого равенства двух вещественных значений.

Есть шанс, что равенство окажется ложным, даже если на самом деле это не так. Например, выражение  $x = (2.23 * x / 2.23)$  формально является истинным, но из-за ошибок округления в вещественном типе может оказаться ложным.

**Логическое выражение** - это совокупность переменных и констант логического типа, а также условных выражений, объединенных знаками логических операций. Операндами логических операций могут выступать другие логические выражения. Результат вычисления логического выражения, как и в случае условных выражений, - одно из двух логических значений типа [Boolean](#): **True** (истина) или **False** (ложь).

В Паскале предусмотрены четыре логические операции:

Приоритет	Операция	Название
I-й	<b>NOT</b>	логическое отрицание
II-й	<b>AND</b>	логическое умножение
III-й	<b>OR</b>	логическое сложение
III-й	<b>XOR</b>	логическое исключающее сложение

Результаты логических операций в зависимости от значения операндов приведены ниже в виде "таблицы истинности":

L1	L2	not L1	L1 and L2	L1 or L2	L1 xor L2
False	False	True	False	False	False
False	True	True	False	True	True
True	False	False	False	True	True
True	True	False	True	True	False

**Примеры логических выражений:**

$(x > 0) \text{ and } (x <= 10) \quad (ch = 'y') \text{ or } (ch = 'Y') \quad (x = 5) \text{ or } (y = 1)$

$(x + 1 > 0) \text{ and } (x + 1 < 10) \text{ or } (y > 0) \text{ and } (y < 2)$

Обратите внимание на необходимость заключения в скобки условных выражений, выступающих операндами логических операций. Это необходимо сделать для того, чтобы сначала выполнялись операции сравнения (они имеют самый низший приоритет), а лишь затем логические операции.

## 2. Упрощение логических выражений.

Равносильные преобразования логических формул имеют то же назначение, что и преобразования формул в обычной алгебре. Они служат для упрощения формул или приведения их к определённому виду путем использования основных законов алгебры логики.

Под упрощением формулы, не содержащей операций импликации и эквиваленции, понимают равносильное преобразование, приводящее к формуле, которая либо содержит по сравнению с исходной меньшее число операций конъюнкции и дизъюнкции и не содержит отрицаний неэлементарных формул, либо содержит меньшее число вхождений переменных.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т.п.), тогда как другие преобразования основаны на свойствах, которыми не обладают операции обычной алгебры (использование распределительного закона для конъюнкции, законов поглощения, склеивания, де Моргана и др.).

Покажем на примерах некоторые приемы и способы, применяемые при упрощении логических формул:

$$1) \quad \overline{x \vee y} \cdot (x \cdot \bar{y}) = \bar{x} \cdot \bar{y} \cdot (x \cdot \bar{y}) = \bar{x} \cdot x \cdot \bar{y} \cdot \bar{y} = 0 \cdot \bar{y} \cdot \bar{y} = 0 \cdot \bar{y} = 0$$

(законы алгебры логики применяются в следующей последовательности: правило де Моргана, сочетательный закон, правило операций переменной с её инверсией и правило операций с константами);

$$2) \quad \bar{x} \cdot y \vee \overline{\bar{x} \vee \bar{y}} \vee x = \bar{x} \cdot y \vee \bar{x} \cdot \bar{y} \vee x = \bar{x} \cdot (y \vee \bar{y}) \vee x = \bar{x} \vee x = 1$$

(применяется правило де Моргана, выносится за скобки общий множитель, используется правило операций переменной с её инверсией);

$$3) \quad (x \vee y) \cdot (\bar{x} \vee y) \cdot (\bar{x} \vee \bar{y}) = (x \vee y) \cdot (\bar{x} \vee y) \cdot (\bar{x} \vee \bar{y}) = (x \vee y) \cdot (\bar{x} \vee \bar{y}) = y \cdot \bar{x}$$

(повторяется второй сомножитель, что разрешено законом идемпотенции; затем комбинируются два первых и два последних сомножителя и используется закон склеивания);

$$\begin{aligned}
& x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot z = x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot z \cdot (y \vee \bar{y}) = \\
& = x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot y \cdot z \vee x \cdot \bar{y} \cdot z = (x \cdot \bar{y} \vee x \cdot \bar{y} \cdot z) \vee (\bar{x} \cdot y \cdot z \vee x \cdot y \cdot z) = \\
4) & = x \cdot \bar{y} \vee y \cdot z
\end{aligned}$$

(вводится вспомогательный логический сомножитель  $(y \vee \bar{y})$ ; затем комбинируются два крайних и два средних логических слагаемых и используется закон поглощения);

$$5) \quad \overline{x \cdot y \vee z} = \overline{x \cdot y} \cdot \bar{z} = (\bar{x} \vee \bar{y}) \cdot \bar{z}$$

(сначала добиваемся, чтобы знак отрицания стоял только перед отдельными переменными, а не перед их комбинациями, для этого дважды применяем правило де Моргана; затем используем закон двойного отрицания);

$$6) \quad x \cdot y \vee x \cdot y \cdot z \vee x \cdot z \cdot p = x \cdot (y \cdot (1 \vee z) \vee z \cdot p) = x \cdot (y \vee z \cdot p)$$

(выносятся за скобки общие множители; применяется правило операций с константами);

$$\begin{aligned}
& x \vee \overline{y \cdot \bar{z} \vee \bar{x} \vee y \vee \bar{z}} = x \vee \bar{y} \vee \bar{z} \vee \bar{x} \cdot \bar{y} \cdot \bar{z} = x \vee \bar{y} \vee z \vee x \cdot \bar{y} \cdot z = \\
7) & = x \vee z \vee (\bar{y} \vee x \cdot \bar{y} \cdot z) = x \vee z \vee \bar{y}
\end{aligned}$$

(к отрицаниям неэлементарных формул применяется правило де Моргана; используются законы двойного отрицания и склеивания);

$$\begin{aligned}
& x \cdot \bar{y} \vee x \cdot y \cdot z \vee x \cdot \bar{y} \cdot z \vee x \cdot \bar{y} \cdot \bar{z} = x \cdot (\bar{y} \vee y \cdot z \vee \bar{y} \cdot z \vee \bar{y} \cdot \bar{z}) = \\
8) & = x \cdot ((\bar{y} \vee \bar{y} \cdot z) \vee (y \cdot z \vee \bar{y} \cdot z)) = x \cdot (\bar{y} \vee \bar{y} \cdot z \vee 1) = x \cdot 1 = x
\end{aligned}$$

(общий множитель  $x$  выносится за скобки, комбинируются слагаемые в скобках — первое с третьим и второе с четвертым, к дизъюнкции  $\bar{y} \cdot z \vee \bar{y} \cdot \bar{z}$  применяется правило операции переменной с её инверсией);

$$\begin{aligned}
& (x \cdot \bar{y} \vee z) \cdot (\bar{x} \vee y) \vee \bar{z} = x \cdot \bar{y} \cdot \bar{x} \vee x \cdot \bar{y} \cdot y \vee z \cdot \bar{x} \vee z \cdot y \vee \bar{z} = 0 \vee 0 \vee \\
& \vee z \cdot \bar{x} \vee z \cdot y \vee \bar{z} = z \cdot \bar{x} \vee (z \vee \bar{z}) \cdot (y \vee \bar{z}) = z \cdot \bar{x} \vee 1 \cdot (y \vee \bar{z}) = z \cdot \bar{x} \vee y \vee \\
& \vee \bar{z} = (z \cdot \bar{x} \vee \bar{z}) \vee y = (z \vee \bar{z}) \cdot (\bar{x} \vee \bar{z}) \vee y = 1 \cdot (\bar{x} \vee \bar{z}) \vee y = \bar{x} \vee \bar{z} \vee y
\end{aligned}$$

(используются распределительный закон для дизъюнкции, правило операции переменной с её инверсией, правило операций с константами, переместительный закон и распределительный закон для конъюнкции);

$$\begin{aligned}
& x \cdot y \cdot (\bar{x} \cdot z \vee \overline{\bar{x} \cdot \bar{y} \cdot z} \vee z \cdot t) = x \cdot y \cdot (\bar{x} \cdot z \vee \bar{x} \cdot \bar{y} \vee \bar{z} \vee z \cdot t) = \\
10) & = x \cdot y \cdot (\bar{x} \cdot z \vee x \cdot y \vee \bar{z} \vee z \cdot t) = x \cdot y \vee x \cdot y \cdot \bar{z} \vee x \cdot y \cdot z \cdot t = x \cdot y
\end{aligned}$$

(используются правило де Моргана, закон двойного отрицания и закон поглощения).

Из этих примеров видно, что при упрощении логических формул не всегда очевидно, какой из законов алгебры логики следует применить на том или ином шаге. Навыки приходят с опытом.

## **1. 4 Лекция № 4 (2 часа).**

### **Тема: «Программное обеспечение (ПО). Классификация ПО»**

#### **1.4.1 Вопросы лекции:**

- 1. Классификация ПО ЭВМ.**
- 2. Функции и назначение ОС.**
- 3. Концептуальные положения ОС Windows`XP.**

#### **1.4.2 Краткое содержание вопросов:**

##### **1. Классификация ПО ЭВМ..**

Рассмотрим определения: программа, программное обеспечение, задача, приложение. С позиций специфики разработки и вида программного обеспечения различают два вида задач:

- 1) *технологические задачи;*
- 2) *функциональные задачи.*

**Предметная область** – совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей.

**Сопровождение программы** – поддержка ее работоспособности, переход ее на новые версии, внесение изменений, исправление обнаруженных ошибок и т.п.

##### **2. Функции и назначение ОС.**

Операционная система (ОС) - это комплекс программного обеспечения, предназначенный для снижения стоимости программирования, упрощения доступа к системе, повышения эффективности работы.

Цель создания операционной системы - получить экономический выигрыш при использовании системы, путем увеличения производительности труда программистов и эффективности работы оборудования.

связь с пользователем в реальном времени для подготовки устройств к работе, переопределение конфигурации и изменения состояния системы.

выполнение операций ввода-вывода; в частности, в состав операционной системы входят программы обработки прерываний от устройств ввода-вывода, обработки запросов к устройствам ввода-вывода и распределения этих запросов между устройствами.

управление памятью, связанное с распределением оперативной памяти между прикладными программами.

управление файлами; основными задачами при этом являются обеспечение защиты, управление выборкой и сохранение секретности хранимой информации.

обработка исключительных условий во время выполнения задачи

появление арифметической или машинной ошибки, прерываний, связанных с неправильной адресацией или выполнением привилегированных команд.

вспомогательные, обеспечивающие организацию сетей, использование служебных программ и языков высокого уровня.

### **3. Концептуальные положения ОС Windows`XP.**

Операционная система Windows XP - это современная многозадачная многопользовательская 32 - разрядная ОС с графическим интерфейсом пользователя. История развития ОС Windows изложена в разделе дистанционное обучение основам работы в операционной системе Windows XP.

Операционные системы семейства Windows являются наиболее распространенными ОС, которые установлены в домашних и офисных ПК.

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

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

## **1. 5 Лекция № 5 (2 часа).**

**Тема: «Офисное программное обеспечение»**

### **1.5.1 Вопросы лекции:**

- 1. Программы обслуживания дисков.**
- 2. Антивирусные продукты.**
- 3. Программы-архиваторы.**

### **1.5.2 Краткое содержание вопросов:**

#### **1. Программы обслуживания дисков.**

Основные операции, которые необходимо иногда проводить с жесткими дисками:



*Разбиение на разделы.* На жесткий диск может быть установлено одновременно несколько операционных систем. Для этого жесткий диск должен быть разбит на разделы, т.е. независимые области на диске, в каждом из которых может быть создана своя файловая система. Наиболее простой и традиционно используемой программой для этих целей в Windows является программа *FDisk*. ОС Windows 2000/XP имеют встроенную программу разбиения жестких дисков на разделы.

*Форматирование.* Оно делится на низкоуровневое (физическое), которое выполняется производителями и делит поверхности магнитных пластин на дорожки и сектора и высокоуровневое (логическое), которое заключается в разбиении на кластеры и размещении на диске файловой системы. Логическое форматирование выполняется стандартной программой ОС Windows *Format*(Форматирование дисков).

*Проверка диска на наличие логических и физических ошибок.* Если каким-то образом соответствие между тем, что записано в загрузочной области диска, и тем, что на самом деле находится на диске, нарушено, последствия могут быть непредсказуемы. Это может возникнуть вследствие сбоев ОС, и другого ПО. В частности, велика вероятность возникновения ошибок при некорректном завершении работы компьютера, при зависании системы и т.д. Обнаружить возникшие проблемы и предотвратить неприятности поможет стандартная программа Windows *Проверка диска* или *ScanDisk*. Но эта программа недостаточно мощна и функциональна. Поэтому при серьезных проблемах необходимо использовать более мощные средства (например, *Norton Disk Doctor(NDD)* из пакета *Norton Utilities* фирмы Symantec).

*Дефрагментация.* Как известно, с точки зрения быстродействия винчестер одно из самых слабых мест системы. К счастью, помогает тот факт, что данные, которые расположены "подряд", считать можно намного быстрее. Что значит "подряд"? Каждый файл на диске занимает определенное пространство. Это пространство разбито на блоки - кластеры. Каждый кластер принадлежит определенному файлу. Хорошо, если кластеры одного файла следуют подряд, но так бывает не всегда. Файлы на диске постоянно создаются и уничтожаются. Операционная система не всегда может выделить файлу место таким образом, чтобы его кластеры шли друг за другом. То есть файл может занимать несколько кластеров, разбросанных по разным местам диска. В этом случае говорят, что файл фрагментирован. При этом скорость чтения и записи файла замедляется заметно. Если на диске образуется много таких файлов, то скорость работы системы заметно падает. Для решения этой проблемы помогает стандартная программа Windows *Дефрагментация диска* или *Defrag*. Опять же можно

порекомендовать использовать более мощное средство дефрагментации (например, *Norton Speed Disk* из *Norton Utilities*).

*Очистка диска.* При регулярной работе на компьютере иногда накапливается некоторый пользовательский и системный "мусор", который полезно периодически расчищать и ликвидировать. Для этого существует много различных программ, а в Windows существует утилита - *Очистка диска*.

## **2. Антивирусные продукты**

Антивирусные программы предназначены для защиты компьютеров от большинства вирусов, червей и «троянских коней», которые могут удалять файлы, получать доступ к личным данным или использовать зараженную систему как средство атаки на другие компьютеры.

*Антивирусная программа (антивирус)* – это программа для обнаружения компьютерных вирусов и других вредоносных программ, лечения и восстановления инфицированных файлов, а также для профилактики – предотвращения заражения файлов или операционной системы вредоносным кодом.

Антивирусные программы обычно используют два различных метода для выполнения своих задач:

- 1) сканирование (просмотр) файлов для поиска уже известных вирусов, для которых в вирусной базе (входящей в комплект антивирусной программы специальной БД) есть информация о характерных фрагментах вирусного программного кода (сигнатурах вирусов);
- 2) обнаружение подозрительного поведения любой программы, которое похоже на поведение зараженной программы («эвристическое сканирование»).

Антивирусное программное обеспечение состоит из пакета программ, которые обнаруживают, предотвращают размножение и удаляют компьютерные вирусы и другие вредоносные программы.

При выборе антивирусной программы необходимо учитывать следующие параметры, которым антивирус должен соответствовать:

1. *Постоянство и надежность работы.* Этот параметр является определяющим. При стабильной работе антивирусной программы нет ощущения, что какие-то зараженные файлы остались незамеченными.
2. *Большой объем и постоянное обновление вирусной базы.* Сюда же относится умение программы быстро опознавать виды вирусов работать с файлами различных типов (архивами, документами), и осуществлять автоматическую проверку всех новых файлов

по мере их копирования.

3. *Быстрота работы антивируса и дополнительные функции.* К дополнительным функциям можно отнести наличие эвристического сканирования и возможность лечения зараженных файлов (когда вирусы из них удаляются, а файлы приводятся в исходное состояние, бывшее до их заражения).

4. *Поддержка различных программой многих операционных систем – многоплатформность.* При работе в сетевом варианте немаловажным является также наличие у антивирусной программы серверных модулей, предназначенных для администрирования, и наличие возможности работы на разных серверах.

### 3. Программы-архиваторы

Назначение программ-архиваторов заключается в экономии места на диске за счет сжатия (упаковки) одного или нескольких файлов в архивный файл. Программы-архиваторы используют для хранения в упакованном виде больших объемов информации, которая понадобится в будущем; переноса информации между компьютерами с помощью дискет или электронной почты; создания в сжатом виде резервных копий файлов; для защиты от компьютерных вирусов. В результате работы программ-архиваторов создаются архивные файлы (архивы).

В основе работы программ-архиваторов лежит процедура поиска и перекодирования одинаковых фрагментов содержимого файлов. Сжатие информации в файлах производится за счет устранения избыточности различными способами (за счет упрощения кодов, исключения постоянных битов, замены их повторяющейся последовательности коэффициентом повторения и т. д.). Существует множество алгоритмов сжатия данных. Например, метод энтропийного кодирования. Сжиматься могут как один, так и несколько файлов, которые в сжатом виде помещаются в архив.

Архивный файл включает оглавление, содержащее следующую информацию о файлах, хранящихся в архиве: имя файла; сведения о каталоге, в котором он находился; дату и время последней модификации файла; размер файла на диске и в архиве; код циклического контроля для каждого файла, используемый для проверки целостности архива. В качестве расширений архивным файлам по умолчанию устанавливается имя соответствующей программы-архиватора.

Большинство программ-архиваторов позволяют создавать многотомные архивы. Многотомный архив — это последовательность архивных файлов, размер которых не превышает заданного размера тома. Обычно создается при необходимости переноса архива на гибких дисках, если размер архива превышает емкость дискеты. При создании

таких архивов архиватор делает паузу для смены дискеты. Многотомный архив имеет общее имя, в расширении же каждого файла многотомного архива указывается номер тома.

Самораспаковывающийся архив — это определенным образом обработанный архивный файл, являющийся выполняемым файлом. Выполнение такого файла приводит к распаковке содержащихся в нем файлов. Типовые функции программ-архиваторов следующие;

1. Помещение исходных файлов в архив.
2. Извлечение файлов из архива.
3. Удаление файлов из архива.
4. Просмотр оглавления архива.
5. Версификация (проверка) архива.

Для каждого файла из архива в оглавлении архивного файла запоминается код циклического контроля (CRC). При извлечении файла код циклического контроля для него вычисляется и сравнивается с тем, что записан в оглавлении архива. При их несовпадении выдается сообщение об ошибке.

В настоящее время наибольшее распространение получили программы-архиваторы, ориентированные на работу под управлением операционной системы Windows. Это WinRAR и WinZip, отличающиеся большой степенью сжатия, работой с длинными именами файлов и удобным интерфейсом.

Особенно удобным пользовательским интерфейсом обладает программа WinRAR. Этот архиватор поддерживает обработку многих архивных форматов и использует оригинальный алгоритм упаковки, особенно эффективный для исполняемых и текстовых файлов. К важным дополнительным возможностям программы относятся: защита архива при помощи пароля; восстановление поврежденных архивов; создание многотомных и самораспаковывающихся архивов; сохранение комментариев к архивам. Пользовательский интерфейс WinRAR содержит основное меню, панель инструментов и рабочую область, в которой показаны все файлы текущей папки. При работе с WinRAR архивы воспринимаются как папки, содержимое которых можно просмотреть традиционными способами.

Операционная система Windows 2000 включает программы обслуживания дисков. Для устранения фрагментации файлов пользователю требуется время от времени выполнять оптимизацию размещения файлов на дисках. Одной из программ, осуществляющих оптимизацию, является программа Disk Defragmenter (находится Пуск/

Программы/Стандартные/Служебные программы). Если в диалоговом окне команды нажать кнопку Анализ, приложение проанализирует размещенные на диске файлы и выдаст рекомендацию о необходимости выполнения дефрагментации. Использование кнопки Просмотр по-прежнему позволяет получить информацию об используемом и свободном пространстве на диске, количестве фрагментов в каждом фрагментированном файле и т. д. Если степень фрагментации невелика, программа предложит произвести обработку.

Запустив утилиту SysInfo, пользователь имеет возможность получить информацию о персональном компьютере. С помощью основного меню утилиты пользователь просматривает информацию об операционной системе, установленных устройствах, программных компонентах и конфигурации программного обеспечения.

## **1. 6 Лекция №6 (2 часа).**

**Тема: «Компьютерные вычислительные сети»**

### **1.6.1 Вопросы лекции:**

**1. Особенности графического интерфейса.**

**2. Базовые операции.**

**3. Работа с шаблонами документов.**

### **1.6.2 Краткое содержание вопросов:**

#### **1. Особенности графического интерфейса.**

##### ***Современные виды интерфейсов:***

1) *командный интерфейс* – пользователь дает команды компьютеру, который их выполняет и выдает результат пользователю. Командный интерфейс реализован в виде пакетной технологии и технологии командной строки;

2) *WIMP-интерфейс* (WIMP от: *Window* – окно; *Image* – образ; *Menu* – меню; *Pointer* – указатель) – диалог пользователя с компьютером ведется при помощи графических образов: меню, окон и других элементов. Интерфейс реализован на двух уровнях технологий: простой графический интерфейс и WIMP-интерфейс;

3) *SILK-интерфейс* (SILK от: *Speech* – речь; *Image* – образ; *Language* – язык; *Knowledge* – знание) – разговор пользователя с компьютером. Интерфейс наиболее приближен к обычной, человеческой форме общения. При этом компьютер определяет команды, анализируя человеческую речь и находя в ней ключевые фразы. Результат выполнения команд компьютер преобразует в понятную человеку форму. Этот вид

интерфейса наиболее требователен к аппаратным ресурсам компьютера, поэтому его применяют в основном для военных целей.

**Основными технологиями реализации интерфейсов** являются следующие (рис.1.3).

1. *Пакетная технология.* Исторически технология появилась первой и существовала уже на релейных машинах Зюса и Цюзе (Германия, 1937 г.). На вход компьютера подавалась последовательность символов, в которых по определенным правилам указывалась последовательность запущенных на выполнение программ. После выполнения очередной программы запускалась следующая программа и т. д. Машина по определенным правилам находила команды и данные. Например, в качестве такой последовательности выступали: перфолента, стопка перфокарт, последовательность нажатия клавиш электрической пишущей машинки (типа *CONSUL*). Машина выдавала свои сообщения на перфоратор, алфавитно-цифровое печатающее устройство (*АЦПУ*), ленту пишущей машинки. Такая машина представляла собой шкаф, в который постоянно подавалась информация, и который постоянно информировал о своем состоянии. Пользователь имел малое влияние на работу машины. Он мог лишь приостановить работу машины, сменить программу и вновь запустить ЭВМ.

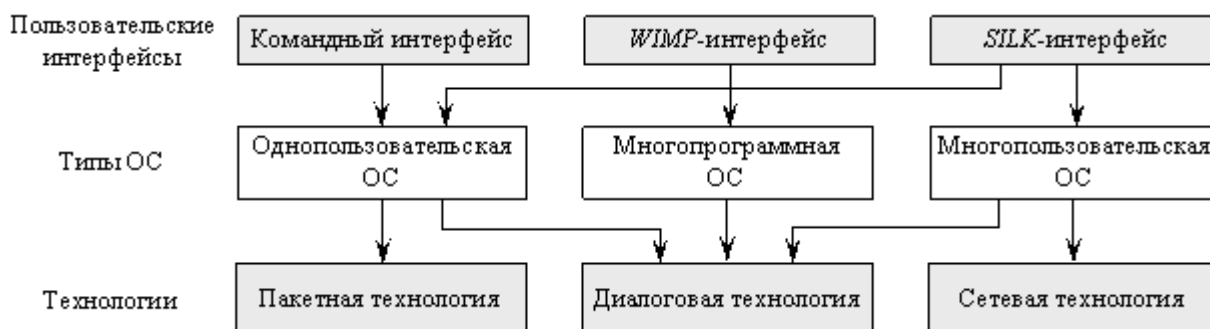


Рис. 1.3. Взаимодействие типов операционных систем, пользовательских интерфейсов и основных технологий их реализации

2. *Технология командной строки.* Информация пользователя для компьютера передается посредством клавиатуры. Компьютер выводит информацию на алфавитно-цифровой дисплей (монитор). Комбинацию «монитор + клавиатура» называли *терминалом* или *консолью*. Команды набираются в *командной строке*, которая представляет собой символ приглашения и мигающий прямоугольник – *курсор*. При нажатии клавиши на месте курсора появляются символы, и курсор смещается вправо, неправильно набранный символ стирается нажатием клавиши *Delete (del)*. Команда заканчивается нажатием клавиши *Enter (Return)*, после чего осуществляется переход в начало следующей строки, в позиции которой компьютер выдает на монитор результаты

своей работы. Затем процесс повторяется. Технология командной строки уже работала на монохромных алфавитно-цифровых дисплеях.

Поскольку вводить позволялось только буквы, цифры и знаки препинания, то технические характеристики дисплея были не существенны. В качестве монитора можно было использовать телевизионный приемник или трубку осциллографа. Преобладающим видом файлов при работе с командным интерфейсом были текстовые файлы, которые можно было создавать при помощи клавиатуры. На время наиболее широкого использования интерфейса командной строки приходится появление операционной системы *UNIX* и первых восьмиразрядных персональных компьютеров с многоплатформенной операционной системой *CP/M*.

3. *Технология графического интерфейса.* Идея графического интерфейса возникла в середине 70-х гг. XX в., когда в исследовательском центре *Xerox Palo Alto Research Center (PARC)* была разработана концепция визуального интерфейса. Предпосылкой графического интерфейса явилось уменьшение времени реакции компьютера на команду, увеличение объема оперативной памяти, развитие технической базы компьютеров. Аппаратным основанием концепции явилось появление алфавитно-цифровых дисплеев, которые стали поддерживать новые эффекты: мерцание символов, инверсию цвета (смена начертания белых символов на черном фоне черными символами на белом фоне), подчеркивание символов. Эффекты распространились не на весь экран, а только на один или более символов. Следующим шагом явилось создание цветного дисплея, позволяющего выводить вместе с этими эффектами символы в 16 цветах на фоне с палитрой (т. е. цветовым набором) из 8 цветов.

Первая система с графическим интерфейсом *8010 Star Information System* группы *PARC* появилась в начале 1981 г. Первоначально интерфейс использовался только в программах. Постепенно он стал переходить и на операционные системы, используемые сначала на компьютерах *Atari* и *Apple Macintosh*, затем и на *IBM*-совместимых компьютерах. Под влиянием новых концепций проходил процесс по унификации использования клавиатуры и мыши прикладными программами. Графический интерфейс пользователя за время своего развития с 1974 г. по настоящее время прошел две стадии.

*Простой графический интерфейс.* На первом этапе графический интерфейс очень походил на технологию командной строки, за исключением следующих отличий:

- при отображении символов допускалось выделение части символов цветом, инверсным изображением, подчеркиванием и мерцанием, благодаря чему повысилась выразительность изображения;

- в зависимости от конкретной реализации графического интерфейса курсор мог представляться мерцающим прямоугольником или некоторой областью, охватывающей несколько символов, которая отличалась от других невыделенных частей;
- нажатие клавиши *Enter* не всегда приводило к выполнению команды и переходу к следующей строке, так как реакция на нажатие любой клавиши во многом зависела от того, в какой части экрана находился курсор;
- кроме клавиши *Enter*, на клавиатуре стали использовать клавиши управления курсором и манипуляторы (мышь, трекбол и др., рис. 1.4), которые позволяли быстро выделять нужную часть экрана и перемещать курсор.

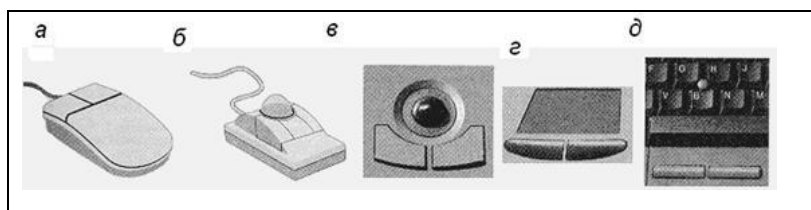


Рис. 1.4. Виды манипуляторов для работы с графическим интерфейсом: *а* – мышь; *б* – трекбол (для настольных компьютеров); *в* – трекбол (в портативном компьютере); *г* – сенсорная панель; *д* – трекпойнт (между клавишами с буквами G, H и B)

Отличительные особенности интерфейса: выделение областей экрана; переопределение клавиш клавиатуры в зависимости от контекста; использование манипуляторов и клавиш управления курсором; широкое использование цветных мониторов. Появление интерфейса совпадает с широким распространением операционной системы *MS-DOS*, которая внедрила этот интерфейс и улучшила характеристики отображения символов и другие параметры монитора. Примеры использования интерфейса: файловая оболочка *Nortron Commander*, текстовый редактор *Multi-Edit*, редакторы: *Лексикон* и *ChiWriter*, текстовый процессор *Microsoft Word for Dos*.

*WIMP-интерфейс* стал вторым этапом в развитии графического интерфейса, его характерные особенности:

- работа с программами, файлами и документами происходит в *окнах* – частях экрана, очерченных рамкой;
- программы, файлы, документы, устройства и другие объекты представляются в виде значков – *иконок*, которые при открытии превращаются в окна;
- действия с объектами осуществляются с помощью меню, которое стало основным элементом управления;



– одним из основных элементов управления стал манипулятор, которым указывают на область экрана, окна или иконки, выделяют ее и посредством *меню* или с использованием других технологий осуществляют управление ими.

Для реализации WIMP-интерфейсу требуются: цветной растровый дисплей с высоким разрешением, манипулятор и программы, ориентированные на данный вид интерфейса, которые предъявляют повышенные требования к производительности компьютера, объему его памяти, пропускной способности шины и т. п. В настоящее время WIMP-интерфейс является стандартом.

4. *Речевая технология* возникла в середине 90-х гг. XX в. после появления недорогих звуковых карт. По этой технологии команды подаются голосом путем произнесения специальных зарезервированных слов – команд. Основными командами являются:

«Проснись» – включение голосового интерфейса;

«Отдыхай» – выключение речевого интерфейса;

«Открыть» – переход в режим вызова той или иной программы, имя программы называется в следующем слове;

«Буду диктовать» – переход из режима команд в режим набора текста голосом;

«Режим команд» – возврат в режим подачи команд голосом и др.

Слова должны выговариваться четко, в одном темпе. Между словами обязательна пауза. Из-за незрелости алгоритма распознавания речи такие системы требуют индивидуальной предварительной настройки на каждого конкретного пользователя. Речевая технология является простейшей реализацией SILK-интерфейса.

5. *Биометрическая технология* (мимический интерфейс). Технология возникла в конце 90-х гг. XX в. Для управления компьютером используется выражение лица человека, направление его взгляда, размер зрачка и другие признаки. Для идентификации пользователя применяется рисунок радужной оболочки его глаз, отпечатки пальцев и другая уникальная информация. Изображения считываются с цифровой видеокамеры, а затем с помощью специальных программ распознавания образов из этого изображения выделяются команды. Эта технология используется в программных продуктах и приложениях для идентификации пользователя компьютера.

6. *Технология семантического интерфейса* (общественного интерфейса). Технология возникла в конце 70-х гг. XX в. с развитием искусственного интеллекта и основана на семантических сетях. Данный вид интерфейса включает в себя: интерфейс командной строки, графический, речевой и мимический интерфейсы. Основная его отличительная черта – отсутствие команд при общении с компьютером. Запрос формируется на естественном языке

в виде связанного текста и образов. По своей сути интерфейс является моделированием общения человека с компьютером.

## **2. Базовые операции.**

В основные функции рассматриваемой программы входит:

Редактирование, создание текста. Сохранение документа в виде файла с необходимым расширением (в 2003 расширение по умолчанию - \*.doc, в 2007, 2010 - \*.docx). Поиск необходимого файла на информационном носителе (жесткий диск, флешка, диск, дискета и др), а также считывание его с диска.

Поиск орфографических ошибок в существующем тексте и проверка лексики.

Возможность текст разбивать на страницы.

Пользователь может форматировать тексты по своему усмотрению.

Возможность создавать оглавление документа (причем в автоматическом режиме).

Встроенная возможность многооконного режима (работа с окнами).

Распечатка файлов различных форматов. Причем данный текстовый редактор отличается следующим: что видит пользователь — то и будет распечатано, так называемый режим WYSIWYG (What You See Is What You Get).

Удаление объектов из файла, а также их внедрение туда.

Вставка и создание рисунков в файле (причем можно вставить и уже готовые фотографии). Можно использовать библиотеку под названием CLIPART, где хранятся готовые рисунки формата \*.wmf, а также вставлять их в файл.

Вставка в файл научных формул (химических, математических и др.) и диаграмм.

Изменение размера и вида используемого в тексте шрифта (причем не для всего текста в целом, а для разных частей может быть использован свой собственный формат печати).

Выделение необходимых участков текста или блока, а также их перенос на новое место. Если потребуется, то их можно удалить. Также включена возможность обрамления требуемых участков текста.

Создание и вставка в файл электронных таблиц. Причем в них можно по-своему усмотрению изменить число строк и столбцов.

Создание баз данных в электронных таблицах, а также выполнение сложных или простых математических вычислений.

Возможность программировать на языке под названием Word Basic, а также создавать макрокоманды. Макрос или макрокоманда — это предложение языка, которое идентифицирует набор самых простых команд. В макросе обычно сохраняется

комбинация клавиш, которая в дальнейшем может использоваться далеко не один раз. С помощью макросов можно автоматизировать самые используемые операции. Следует отметить, что кроме клавишных макросов существуют языковые макросы, которые создаются на языке программирования Word Basic.

Создание конвертов писем, этикетов и эмблем.

Вставка в файл видеоклипов, текстовых спец эффектов, мультимедийных и звуковых файлов.

Просмотр перед печатью текста, с возможностью его увеличения для лучшего просмотра.

В рассматриваемый текстовый редактор входит обширная справочная система, благодаря которой пользователь может довольно быстро получить помощь.

### **3. Работа с шаблонами документов.**

**Шаблон** - это документ, используемый в качестве образца для создания новых документов. Шаблоны используются для унификации структуры и внешнего вида документов. Шаблон определяет основную структуру документа и содержит настройки документа, такие как элементы списков автотекста и автозамены, макросы, панели инструментов, пользовательские меню и сочетания клавиш, форматирование и стили.

Все новые документы в **Microsoft Word** создаются на основе шаблонов. По умолчанию при создании нового документа выбирается **шаблон Обычный** на вкладке *Общие*. В комплект поставки Word входят десятки шаблонов, которые могут быть использованы для создания документов различных типов.

После запуска **Microsoft Word** в окне приложения открывается пустой документ, который называется Документ 1. Этот документ основан на шаблоне Обычный (Normal.dot), а текст вводится в стиле Обычный, в котором установлены следующие параметры форматирования: гарнитура шрифта - Times New Roman, кегль (размер шрифта) - 12 пунктов, выравнивание по левому краю, межстрочный интервал - одинарный.

**Шаблон Normal.dot** является общим шаблоном. Созданные пользователем элементы, хранящиеся в этом шаблоне, доступны во всех документах.

## **1. 7 Лекция №7 (2 часа).**

**Тема: «Защита информации в локальных и глобальных компьютерных сетях»**

### **1.7.1 Вопросы лекции:**

#### **1.Классификация компьютерных сетей.**

## **2.Компоненты аппаратного и программного обеспечения сетей.**

## **3.Цифровая и доменная система имен.**

### **1.7.2 Краткое содержание вопросов:**

#### **1. Классификация компьютерных сетей.**

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

В зависимости от территориального расположения абонентов компьютерные сети делятся на:

- *глобальные* — вычислительная сеть объединяет абонентов, расположенных в различных странах, на различных континентах. Глобальные вычислительные сети позволяют решить проблему объединения информационных ресурсов человечества и организации доступа к этим ресурсам;
- *региональные* — вычислительная сеть связывает абонентов, расположенных на значительном расстоянии друг от друга. Она может включать абонентов большого города, экономического региона, отдельной страны;
- *локальные* — вычислительная сеть объединяет абонентов, расположенных в пределах небольшой территории. К классу локальных сетей относятся сети отдельных предприятий, фирм, офисов и т. д.

Базовые принципы организации компьютерной сети определяют ее основные характеристики:

- *операционные возможности* — перечень основных действий по обработке данных. Абоненты сети имеют возможность использовать память и процессоры многих компьютеров для хранения и обработки данных. Предоставляемая компьютерной сетью возможность параллельной обработки данных многими компьютерами и дублирования необходимых ресурсов позволяет сократить время решения задач, повысить надежность системы и достоверность результатов;
- *производительность* — представляет собой суммарную производительность компьютеров, участвующих в решении задачи пользователя;
- *время доставки сообщений* — определяется как статистическое среднее время от момента передачи сообщения в сеть до момента получения сообщения адресатом;
- *стоимость предоставляемых услуг.*

## **2. Компоненты аппаратного и программного обеспечения сетей.**

Даже в результате достаточно поверхностного рассмотрения работы в сети становится ясно, что вычислительная сеть - это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов. Изучение сети в целом предполагает знание принципов работы ее отдельных элементов:

- компьютеров;
- коммуникационного оборудования;
- операционных систем;
- сетевых приложений.

Весь комплекс программно-аппаратных средств сети может быть описан многослойной моделью. В основе любой сети лежит аппаратный слой стандартизованных компьютерных платформ. В настоящее время в сетях широко и успешно применяются компьютеры различных классов - от персональных компьютеров до мэйнфреймов и суперЭВМ. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью.

Второй слой - это коммуникационное оборудование. Хотя компьютеры и являются центральными элементами обработки данных в сетях, в последнее время не менее важную роль стали играть коммуникационные устройства. Кабельные системы, повторители, мосты, коммутаторы, маршрутизаторы и модульные концентраторы из вспомогательных компонентов сети превратились в основные наряду с компьютерами и системным программным обеспечением как по влиянию на характеристики сети, так и по стоимости. Сегодня коммуникационное устройство может представлять собой сложный специализированный мультипроцессор, который нужно конфигурировать, оптимизировать и администрировать. Изучение принципов работы коммуникационного оборудования требует знакомства с большим количеством протоколов, используемых как в локальных, так и глобальных сетях.

Третьим слоем, образующим программную платформу сети, являются операционные системы (ОС). От того, какие концепции управления локальными и распределенными ресурсами положены в основу сетевой ОС, зависит эффективность работы всей сети. При проектировании сети важно учитывать, насколько просто данная операционная система может взаимодействовать с другими ОС сети, насколько она обеспечивает безопасность и защищенность данных, до какой степени она позволяет наращивать число пользователей, можно ли перенести ее на компьютер другого типа и многие другие соображения.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных, системы автоматизации коллективной работы и др. Очень важно представлять диапазон возможностей, предоставляемых приложениями для различных областей применения, а также знать, насколько они совместимы с другими сетевыми приложениями и операционными сис

### **3. Цифровая и доменная система имен**

В сети Интернет существует единая система доменных имен, с помощью которой пользователи имеют возможность обращаться к существующим информационным ресурсам и компьютерным системам, используя в качестве адресов последних соответствующие доменные имена.

На настоящий момент практически каждая уважающая себя организация имеет или стремится иметь свой сайт в сети Интернет с собственным доменным именем, которое представляет из себя обозначение, состоящее из нескольких частей, называемых доменами и разделенных точками, вида domain3.domain2.domain1, где значения domain1, domain2, domain3 — домены первого, второго, третьего уровней соответственно, которые состоят из букв латинского алфавита, цифр и символа «-» (дефис). Старшим, наиболее приоритетным доменом считается домен первого уровня, находящийся в доменном имени справа, далее справа налево приоритет доменов падает, и, на самом деле, количество доменов в доменном имени (его «длина») зависит от волеизъявления и технических возможностей владельца информационного ресурса или системы.

Важно отметить, что система адресации с помощью доменных имен является вторичной по отношению к так называемой системе цифровых IP-адресов (вида a.d.c.d, где a, b, c, d — числа от 0 до 255), которая служит для индивидуализации функционирующих в сети Интернет компьютеров. Система доменных имен организована для того, чтобы пользователям в сети Интернет было удобнее использовать в качестве адреса не последовательность цифр (также разделенных точками), а имеющее определенное смысловое значение и формирующееся по определенным правилам иерархии имя.

Технически система доменных имен поддерживается системой так называемых DNS-серверов (DNS — Domain Name System), которые посредством специальных баз данных доменных имен автоматически преобразуют доменный адрес в цифровой IP-адрес системы, которая поддерживает соответствующий информационный ресурс1.

Существующая в сети Интернет доменная система организована таким образом, чтобы обеспечить уникальность каждого доменного имени, т. е. не могут существовать два идентичных доменных имени с одинаковым количеством совпадающих доменов. Это позволяет с помощью одного доменного имени однозначно адресоваться к конкретному информационному ресурсу, за которым это имя закреплено.

Кроме того, доменная система иерархична и организационно упорядочена не только относительно функционирующих в информационном пространстве компьютерных систем, но и относительно мирового геополитического деления.

В ее рамках существуют два вида доменов первого уровня: родовые (gTLD, generic Top-Level Domain) и национальные (ccTLD, country code Top-Level Domain). Первых сейчас около десяти, и изначально (с середины 80-х годов прошлого века) они были выбраны Internet Corporation for Assigned Names and Numbers (ICANN)<sup>2</sup> — одной из ведущих организаций, определяющих политику развития сети Интернет, и должны были служить определенному смысловому разграничению: ресурсы с доменом первого уровня COM должны были принадлежать коммерческим организациям, с доменом ORG — некоммерческим, с доменом NET — провайдерам и владельцам сетей. Однако со временем данная рекомендация перестала соблюдаться, и сейчас можно очень часто встретить адрес вида domain2.org у коммерческой организации, а домен вида domain2.com или domain2.net может служить адресом домашнего сайта физического лица.

1 Для определения «вручную» взаимного соответствия доменного имени и IP-адреса можно воспользоваться сервисом «who is» либо в рамках установленной на компьютере операционной системы, либо непосредственно с помощью многочисленных сервисов в сети Интернет (см., например, [www.samspade.org](http://www.samspade.org) или [www.leader.ru/secure](http://www.leader.ru/secure)).

2 Адрес в сети Интернет: <http://www.icann.org/>.

Национальные домены первого уровня являются двухбуквенными и именуются в соответствии с международным стандартом

ISO 3166-1, по которому каждой стране мира приписан определенный код из двух букв латинского алфавита. Например, для России зарезервирован код RU, для СССР использовался код SU (активных доменных имен в данной зоне в Российской Федерации сейчас довольно много), за Украиной закреплен код UA, у Германии код DE, у Франции — FR и т. д.

Таким образом, в рамках существующей иерархии доменных имен любое лицо может выбрать (зарегистрировать) для себя незанятое доменное имя или в какой-либо из родовых зон (в рамках иерархии родового домена), или в той или иной национальной зоне<sup>1</sup>.

Для этого достаточно придумать домен второго уровня (возможно, и третьего уровня — например, в географических зонах SPB, MSK), который еще не занят в соответствующей зоне, обратиться к уполномоченным организациям, так называемым регистраторам доменов, и на основе договорных возмездных отношений зарегистрировать доменное имя. После этого его можно использовать в течение определенного срока<sup>2</sup>.

1 Следует отметить, что в ряде доменов первого уровня регистрация не является открытой (например, в родовых зонах GOV и MIL), и решение о получении домена зависит, в частности, от правосубъектности и юрисдикции заявителя.

2 Обычно регистрация осуществляется на срок от 1 года до 10 лет, после чего владеющее доменом лицо имеет возможность до истечения срока регистрации пролонгировать существующие договорные отношения с регистратором.

3 Адрес в сети Интернет: <http://www.ripn.net>.

4 Адрес в сети Интернет: <http://www.ripn.net:8080/nic/dns/contract.html>.

Например, в зоне RU до середины 2001 года регистрацией занималась одна единственная организация, которая поддерживала функционирование системы доменных имен вида domain2.RU, — НО Российский научно-исследовательский институт развития общественных сетей (РосНИИРОС)<sup>3</sup>. РосНИИРОС занимается данной деятельностью с 1993 года в рамках разработанных и принятых им же процедур<sup>4</sup>. В настоящий момент объявлено о внедрении системы, которая, возможно, позволит уйти от доминирующего положения единственной организации путем создания института регистрации. Формально еще в середине 2001 года монополия РосНИИРОС была ликвидирована, и на рынок в первую очередь вышел являющийся его представителем АНО «Региональный Сетевой Информационный Центр»<sup>2</sup>.

При регистрации очень важно выбрать такое доменное имя, которое бы удачно характеризовало информационный ресурс и/или деятельность его владельца и легко запоминалось пользователями. Например, адрес сайта организации считается идеальным, если он совпадает с ее названием или принадлежащим ей товарным знаком или если выбирается какое-либо общеупотребительное слово или сочетание слов, имеющее отношение к сфере деятельности организации.

Однако задача выбора имени не является тривиальной, поскольку в настоящий момент в сети Интернет зарегистрировано огромное количество доменов: например, по данным NetNames International Ltd.<sup>2</sup> на февраль 2001 года всего в мире было зарегистрировано 36 149 297 доменных имен, из них в зоне COM — 22 263 512, в зоне NET — 3 857 319, в зоне ORG — 2 430 243, в российской адресной зоне RU сейчас зарегистрировано свыше восьмидесяти тысяч доменов.



Для большинства тех, кто только начинает формировать политику своего присутствия в сети Интернет, задача выбора доменного имени может стать довольно трудной. Например, зарегистрировать общеупотребительное слово в зонах COM или NET уже к концу 1999 года было практически невозможно — оказалось, что в зарегистрированных именах было использовано свыше 97 процентов слов из словаря Webster<sup>3</sup>.

1 Адрес в сети Интернет: <http://www.nic.ru>.

2 Адрес в сети Интернет: <http://www.domainstats.com>.

3 См. C-NET News.com: <http://news.cnet.com/news/0-1005-200-1507943.html?tag=st>.

Данные затруднения одних, как это всегда происходит, были использованы для собственного блага другими — возник рынок доменных имен. Появились две категории лиц, занимающихся регистрацией и продажей доменов, — торговцы таковыми, преимущественно регистрирующие общеупотребительные слова или их сочетания (domain name traders), и киберсквоттеры (cybersquatters), которые регистрируют имена, совпадающие или похожие на известные товарные знаки, названия компаний, имена знаменитых физических лиц.

Продажа доменных имен в настоящий момент процветает, и продают не только легко запоминающиеся и удачные с позиций маркетинга домены второго уровня (например, домен business.com был продан за 7,5 миллионов долларов США), но и национальные домены первого уровня — государство Тувалу за 50 миллионов уступило на 12 лет право регистрации доменов в зоне TV1.

Нужно отметить, что система доменных имен постоянно развивается. Так, в 2001 году ICANN было принято решение о введении новых родовых доменов, в конце осени 2001 года начали функционировать домены INFO и MUSEUM. Вводятся доменные имена с использованием национальных алфавитов<sup>2</sup>. Уделяется большое внимание разрешению конфликтов в связи с регистрацией и использованием доменных имен (о чем будет сказано ниже).

## **1. 8 Лекция №8 (2 часа).**

**Тема: «Алгоритмизация и программирование»**

### **1.8.1 Вопросы лекции:**

- 1. Понятие алгоритма. Формы представления алгоритмов.**
- 2. Свойства алгоритмов. Типы алгоритмов.**
- 3. Программа как средство записи алгоритма для реализации на ВМ.**

### **1.8.2 Краткое содержание вопросов:**

- 1. Понятие алгоритма. Формы представления алгоритмов.**

Алгоритм — заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения

решения задачи за конечное число шагов.

На практике наиболее распространены следующие формы представления алгоритмов: словесная (запись на естественном языке);

графическая (изображения из графических символов);

псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);

программная (тексты на языках программирования).

## 2. Свойства алгоритмов. Типы алгоритмов.

Основные свойства алгоритмов следующие:

1. Понятность для исполнителя — исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.

2. Дискретность (прерывность, раздельность) — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).

3. Определенность — каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

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

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

## 3. Программа как средство записи алгоритма для реализации на ВМ.

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

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы — компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.

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

## 1. 9 Лекция №9 (2 часа).

Тема: «Понятие о структурном программировании»

### 1.9.1 Вопросы лекции:

1. Теговая модель построения Web-страниц.
2. Форматирование HTML-документа и основные приемы.
3. Использование списков-перечислений и гиперссылок в Web-документах.

## 1.9.2 Краткое содержание вопросов:

### 1. Теговая модель построения Web-страниц

Все теги HTML по их назначению и области действия можно разделить на следующие основные группы:

- определяющие структуру документа;
- оформление блоков гипертекста (параграфы, списки, таблицы, картинки);
- гипертекстовые ссылки и закладки;
- формы для организации диалога;
- вызов программ.

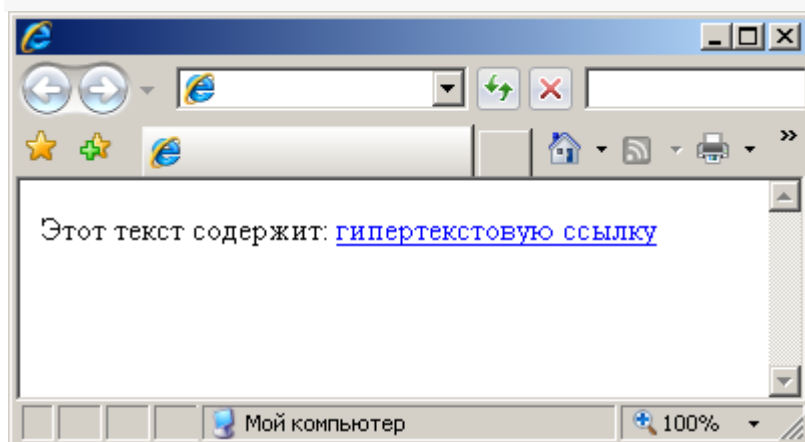
Структура гипертекстовой сети задается гипертекстовыми ссылками. Гипертекстовая ссылка — это адрес другого HTML-документа или информационного ресурса Internet, который тематически, логически или каким-либо другим способом связан с документом, в котором ссылка определена.

Естественно, при таких условиях очень важна схема адресации всех имеющихся информационных ресурсов.

Реальный механизм интерпретации идентификатора ресурса, опирающийся на URI (Uniform Resource Identifier, универсальный идентификатор ресурса), называется URL, и пользователи WWW имеют дело именно с ним.

Типичным примером использования такой записи можно считать следующий пример:

Этот текст содержит: `<A HREF="http://www.mail.ru/">гипертекстовую ссылку</A>`



В приведенном выше примере тег «A», который в HTML называют якорем (anchor), использует атрибут HREF, обозначающий гипертекстовую ссылку (Hypertext Reference), для записи этой ссылки в форме URL. Данная ссылка указывает на корневой документ на сервере «www.mail.ru», доступ к которому осуществляется по протоколу HTTP.

Гипертекстовые ссылки в HTML делятся на два класса: контекстные гипертекстовые ссылки и общие. Контекстные ссылки вмонтированы в тело документа, как это было

продемонстрировано в предыдущем примере, в то время как общие ссылки связаны со всем документом в целом и могут использоваться при просмотре любого фрагмента документа. Оба класса ссылок изначально присутствуют в стандарте языка, однако первое время наибольшей популярностью пользовались контекстные ссылки. Эта популярность привела к тому, что механизм использования общих ссылок практически полностью «атрофировался». В данном примере мы заключили URL в двойные кавычки. На самом деле, это необязательно. Кавычки (двойные или одинарные) применяются только тогда, когда внутри значения URL появляются символы-разделители (пробел, табуляция, неотображаемые символы). Но такого сорта URL следует всячески избегать.

Структура HTML-документа позволяет задействовать вложенные друг в друга контейнеры. Собственно, сам документ — это один большой контейнер, который начинается с тега `<HTML>` и заканчивается тегом `</HTML>`.

## 2. Форматирование HTML-документа и основные приемы

Чтобы наша Web-страница выглядела более привлекательно, разделим текст на абзацы и выделим заголовок. HTML имеет шесть уровней заголовков разделов документа, пронумерованных цифрами от 1 до 6. Заголовки объявляются парой тэгов с номерами, соответствующими уровню, например, `<h1></h1>` - заголовок раздела первого уровня, а `<h6></h6>` - заголовок раздела шестого уровня. От нормального текста заголовки отличаются размером и толщиной букв. Заголовок первого уровня **h1** отображается обычно очень крупным шрифтом, в то время как заголовок шестого уровня **h6** - очень мелким.

Не следует путать заголовки разделов документа с рассмотренным ранее заголовком документа, определяемым элементом `<head></head>`.

В качестве заголовка текста используем первое предложение - **Добро пожаловать на страничку лица 1548!** Для этого достаточно ограничить его тэгами `<h1>` и `</h1>`.

Вставьте в текст файла **licey1548.html** тэги `<h1>` и `</h1>` так, чтобы они ограничивали первое предложение текста, и этот фрагмент кода принял следующий вид:

```
<h1>Добро пожаловать на страничку лица 1548!</h1>
```

Сохраните файл, выбрав команду **Сохранить** меню **Файл** программы **Блокнот**.

Операцию сохранения необходимо всегда выполнять перед просмотром документа, так как браузер открывает файл для просмотра, загружая его в оперативную память компьютера с диска. Если после редактирования кода HTML вы не сохраните файл, то никаких изменений в браузере не увидите.

Нажмите клавишу **F5** или кнопку **Обновить** на панели инструментов рабочего окна программы Internet Explorer или аналогичную кнопку **Обновить** в Netscape Communicator. Файл **licey1548.html** будет перезагружен, и вы увидите в окне браузера, как выглядит заголовок первого уровня.

Используя шесть уровней заголовков, которые предоставляет в ваше распоряжение язык HTML, можно создать легко читаемый документ с интуитивно ясной структурой. Помните, что ваш документ всегда будет читаться значительно лучше, если в нем будет четкое разделение на разделы и подразделы.

По умолчанию заголовок выравнивается по левому краю страницы. Но его можно также выровнять по правому краю или центрировать. Для правостороннего выравнивания в тэге **<h1>** используется атрибут **align=right**, а для центрирования — **align=center**. Допускается также явное указание левостороннего выравнивания - **align=left**.

Добавьте в тэг **<h1>** атрибут **align=center**, чтобы центрировать заголовок. Этот элемент должен принять следующий вид:

```
<h1 align=center>Добро пожаловать на страничку лица 1548!</h1>
```

Теперь займемся остальным текстом. Увеличим его размер и оформим текст полужирным курсивным начертанием. Для установки полужирного начертания используются парные тэги **<b></b>**.

Вставьте в файле **licey1548.html** открывающий **<b>** и закрывающий **</b>** тэги так, чтобы они ограничили текст **Здесь вы узнаете...**. Этот элемент должен принять следующий вид:

```
<b>Здесь Вы узнаете о нашей деятельности, о наших интересах и увлечениях, о наших успехах и достижениях</b>
```

Курсивное начертание устанавливается с помощью тэгов **<i></i>**.

Вставьте в исходный код HTML тэг **<i>** и **</i>** так, чтобы отредактированный элемент принял следующий вид:

```
<b><i>Здесь Вы узнаете о нашей деятельности, о наших интересах и увлечениях, о наших успехах и достижениях</i></b>
```

Элементы разметки могут быть вложенными, как в данной структуре, где элемент **<i>...</i>** вложен в элемент **<b>...</b>**. Современные браузеры способны правильно обрабатывать вложенные тэги. Поэтому вам необходимо следить за тем, чтобы не нарушался порядок вложения. Работа браузера окажется затрудненной, если вложенность будет нарушена. Например, такая запись будет неправильной: **<b><i>...</b></i>**. Соблюдение вложенности - очень важная часть общей культуры написания HTML-кода.

С помощью пары тэгов `<u></u>` можно установить подчеркнутое начертание текстового фрагмента, который ограничивают данные тэги, а с помощью пары тэгов `<tt></tt>` -отобразить текст телетайпным шрифтом.

После того, как вы просмотрите полученный результат, увеличим размер шрифта текста. Это можно сделать разными способами.

Тэги `<big></big>` увеличивают размер шрифта текста, заключенного между ними.

Добавьте в начало и конец вышеуказанного фрагмента кода соответственно тэги `<big>` и `</big>` так, чтобы элемент принял следующий вид:

`<big><b><i>` Здесь Вы узнаете о нашей деятельности, о наших интересах и увлечениях, о наших успехах и достижениях `</i></b></big>`

С помощью тэгов `<small></small>` вы можете уменьшить размер шрифта текста по сравнению с исходным.

Другой способ указания размера шрифта - с помощью тэгов `<font></font>` с атрибутом **size**. В качестве значений этого атрибута используются целые числа от 1 до 7. Причем значение 1 соответствует минимальному размеру шрифта, а значение 7 - максимальному.

В качестве значения атрибута **size** можно также использовать числа от 1 до 7 со знаком + (плюс) или - (минус). В этом случае размер шрифта соответственно увеличивается или уменьшается, по сравнению с исходным, например, тэги `<fontsize=+1></font>` увеличат размер шрифта, по сравнению с текущим, на один порядок.

В тэгах `<font></font>` может использоваться также атрибут **color** для указания цвета шрифта, ограниченного тэгами текста. Значения этого атрибута такие же, как и для рассмотренных ранее атрибутов, описывающих цвет фона и текста документа.

По умолчанию абзац с текстом **Здесь Вы узнаете...** выровнен влево. Центрируем его по горизонтали с помощью тэгов `<center></center>`. Вы можете также выравнивать абзац по правому краю страницы с помощью тэгов `<right></right>` или по левому - с помощью тэгов `<left></left>`.

Вставьте тэги `<center></center>`, ограничив ими указанный абзац так, чтобы код HTML принял следующий вид:

```
<html>
<head>
<title>Лицей № 1548</title>
</head>
<body bgcolor=blue text=yellow>
```

```
<h1 align=center>Добро пожаловать на страничку лица № 1548!</h1>  
<center><font size=5><b><i>Здесь Вы узнаете о нашей деятельности, о наших  
интересах и увлечениях, о наших успехах и достижениях</i></b></font><center>  
</body>  
</html>
```

Язык HTML допускает два подхода к шрифтовому выделению фрагментов текста. С одной стороны, можно прямо указать, что шрифт на некотором участке текста может быть полужирным или курсивным, т.е. явно указать начертание шрифта текста, как мы это делали в данном опыте. С другой стороны, можно пометить любой фрагмент текста, как имеющий некий, отличный от нормального, логический стиль, предоставив интерпретацию этого стиля браузеру. Таким образом, логический стиль указывает роль текстового фрагмента, например, большую значимость по сравнению с обычным текстом или то, что данный фрагмент является цитатой. В своей работе вы можете использовать следующие тэги, определяющие логические стили. Проверьте, как они работают в разных браузерах.

**<dfn></dfn>** - применяется для определения слова. Текст обычно выводится курсивом.

**<em></em>** - для выделения слов и усиления. Отображается курсивом.

**<cite></cite>** - для выделения названий книг, фильмов, спектаклей и т.д. Выводится курсивом.

**<code></code>** - для фрагментов кода программ. Показывается на экране шрифтом фиксированной ширины.

**<kbd></kbd>** - используется для текста, который пользователь вводит с клавиатуры. В разных браузерах может отображаться разными шрифтами.

**<samp></samp>** - служит для отображения сообщений программ. Выводится шрифтом фиксированной ширины.

**<strong></strong>** - для особо важных фрагментов. Обычно выделяется полужирным начертанием.

**<var></var>** - используется для указания, что часть текста или слово является символьной переменной, т.е. текстом, который может быть заменен различными выражениями. Отображается курсивом.

### 3. Использование списков-перечислений и гиперссылок в Web-документах

Для наглядного представления информации на Web-страницах часто используются списки. Списки могут быть нумерованными и ненумерованными. Создадим новую страницу нашего сайта, на которой вставим ненумерованный список.

Создайте структуру Web-документа, напечатав основные тэги:

```
<html>
<head>
<title></title>
</head>
<body>
</body>
</html>
```

В качестве заголовка документа в тэге **<title></title>** введите: Чем мы занимаемся?:

```
<title>Чем мы занимаемся?</title>
```

Такой же заголовок для списка введите в теле документа между тэгами **<body>** и **</body>**, используя для его отображения тэги **<h2></h2>** с атрибутом **align=center**, выравнивающим заголовок по центру страницы:

```
<h2 align=center>Чем мы занимаемся?</h2>
```

Самостоятельно подберите цвет фона страницы и цвет текста, указав соответствующие значения для атрибутов **bgcolor** и **text** в открывающемся тэге **<body>**, например, так:

```
<body bgcolor=aqua text=navy>
```

Помните, однако, что цвет текста должен быть таким, чтобы текст хорошо читался на выбранном фоне. Если атрибуты цвета не указывать, то по умолчанию текст будет отображаться черным цветом на белом фоне.

Теперь вставим на страницу ненумерованный список с информацией о сфере деятельности нашего лица. Ненумерованные списки создаются с помощью пары тэгов **<ul></ul>**, которые ограничивают список.

Между ними располагается столько элементов, начинающихся с тэга **<li>**, сколько элементов в списке.

Вставьте пустую строку под строкой с кодом **<h2 align=center>Чем мы занимаемся?</h2>** и введите в теле документа следующий код:

```
<ul>
<li>Изучением фундаментальных курсов физико-математических дисциплин.
<li>Изучением курсов дисциплин, составляющих основу современной теоретической и
прикладной экономики.
```



```
<li>Изучением цикла гуманитарных дисциплин.  
<li>Изучением сценического искусства.  
</ul>
```

Обратите внимание: тэг **<li>** может использоваться как одиночный, т.е. без парного, закрывающего тэга **</li>**.

Полный HTML-код документа должен иметь следующий вид:

```
<html>  
<head>  
<title>Чем мы занимаемся?</title>  
</head>  
<body bgcolor=aqua text=navy>  
<h2 align=center>Чем мы занимаемся?</h2>  
<ul>  
<li>Изучением фундаментальных курсов физико-математических дисциплин.  
<li>Изучение курсов дисциплин, составляющих основу современной теоретической и  
прикладной экономики.  
<li>Изучением цикла гуманитарных дисциплин.  
<li>Изучением сценического искусства.  
</ul>  
</body>  
</html>
```

Когда ввод кода будет закончен, сохраните документ в папке **Web** под именем **spisok.html**.

Воспользовавшись командой **Открыть** меню **Файл** в браузере Internet Explorer или командой **Открыть страницу** меню **Файл** в браузере Netscape Communicator, откройте файл **spisok.html** из папки **Web**.

На экране (рис. 4) Вы увидите, что каждый элемент нумерованного списка выделяется специальным маркером.

В нумерованном списке каждый элемент будет отмечаться его порядковым номером. Для создания нумерованных списков используются тэги **<ol></ol>**, между которыми помещаются элементы **<li>**.

## **2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ СЕМИНАРСКИХ ЗАНЯТИЙ**

## **2.1 Семинарское занятие №1 (2 часа).**

**Тема: «Основные понятия информации. Классификация информации»**

### **2.1.1 Вопросы к занятию:**

1. Рынок информационных продуктов
2. Правовое регулирование рынка информационных услуг.

### **2.1.2 Краткое описание проводимого занятия:**

Цель: ознакомление с информатикой как с наукой

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: познавательная

Форма: развернутая беседа на основании плана; устный опрос студентов по вопросам плана семинара

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя.

## **2.2 Семинарское занятие №2 ( 2 часа).**

**Тема: «Представление информации в ПК»**

### **2.2.1 Вопросы к занятию:**

- 1.Правила перевода чисел из одной системы счисления в другую.
- 2.Измерение количества информации.

### **2.2.2 Краткое описание проводимого занятия:**

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе освоения информации и ее свойств.

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: прослушивание и обсуждение докладов (рефератов) студентов; обсуждение письменных рефератов, заранее подготовленных отдельными студентами и затем до семинара прочитанных всей группой;

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя.

## **2.3 Семинарское занятие №3 ( 2 часа).**

**Тема: «Понятие и основные виды архитектуры ПК. Алгебра логики»**

### **2.3.1 Вопросы к занятию:**

1. Арифметические действия в разных системах счисления.
2. Упрощение логических выражений

### **2.3.2 Краткое описание проводимого занятия:**

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе изучения логических приемов.

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: семинар-коллоквиум;

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя

## **2.4 Семинарское занятие №4 ( 2 часа).**

**Тема: «Программное обеспечение (ПО). Классификация ПО»**

### **2.4.1 Вопросы к занятию:**

1. Операции с объектами в MS Windows'XP.
2. Стандартные приложения Windows.

## **2.4.2 Краткое описание проводимого занятия**

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе изучения программных продуктов.

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: комментированное чтение и анализ документов (литературы); решение задач на самостоятельность мышления;

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя

## **2.5 Семинарское занятие №5 ( 2 часа).**

**Тема: «Офисное программное обеспечение»**

### **2.5.1 Вопросы к занятию:**

1. Обслуживание магнитных дисков.
2. Работа с архивами документов, распаковка файлов.

### **2.5.2 Краткое описание проводимого занятия**

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе изучения сервисного ПО

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: семинар по материалам исследования, проведенного студентами под руководством преподавателя;

Структура:

- вступительное слово преподавателя,

- основная часть,
- заключительное слово преподавателя

## **2.6 Семинарское занятие №6 ( 2 часа).**

**Тема: «Компьютерные вычислительные сети»**

### **2.6.1 Вопросы к занятию:**

1. Ввод и редактирование данных MS Word

2. Простейшие формулы и функции.

### **2.6.2 Краткое описание проводимого занятия**

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе изучения издательских систем

Задачи: расширение знаний студентов информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: семинар;

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя.

## **2.7 Семинарское занятие №7 ( 2 часа).**

**Тема: «Защита информации в локальных и глобальных компьютерных сетях»**

### **2.7.1 Вопросы к занятию:**

Организация презентаций средствами MS PowerPoint'2003 (2007).

### **2.7.2 Краткое описание проводимого занятия**

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе изучения MS PowerPoint'2003 (2007)

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня

теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: семинар по материалам исследования, проведенного студентами под руководством преподавателя;

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя

## **2.8 Семинарское занятие №8( 2 часа).**

**Тема: «Алгоритмизация и программирование»**

### **2.8.1 Вопросы к занятию:**

- 1.Локальные вычислительные сети.
- 2.Создание сайта в Word

### **2.8.2 Краткое описание проводимого занятия**

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе изучения вычислительных сетей.

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: семинар по материалам исследования, проведенного студентами под руководством преподавателя;

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя

## **2.9 Семинарское занятие №9 ( 2 часа).**

**Тема: «Понятие о структурном программировании»**

### **2.9.1 Вопросы к занятию:**

- 1.Глобальная сеть интернет.

## 2. Язык HTML

### 2.9.2 Краткое описание проводимого занятия

Цель: развитие самостоятельности мышления и творческой активности студентов в ходе изучения HTML

Задачи: расширение знаний студентов по информатике; формирование умения постановки и решения интеллектуальных задач и проблем; совершенствование способностей по аргументации студентами своей точки зрения, а также по доказательству и опровержению других суждений; демонстрация студентами достигнутого уровня теоретической подготовки; формирование навыков самостоятельной работы с литературой.

Функция: развивающая; воспитательная

Форма: семинар по материалам исследования, проведенного студентами под руководством преподавателя;

Структура:

- вступительное слово преподавателя,
- основная часть,
- заключительное слово преподавателя