

Учреждение образования
«Белорусский государственный университет культуры и искусств»
Факультет культурологии и социокультурной деятельности
Кафедра информационных технологий в культуре

СОГЛАСОВАНО

Заведующий кафедрой

СОГЛАСОВАНО

Декан факультета

П.В. Гляков
« ____ » _____ 2017 г.

И.Н. Воронович
« ____ » _____ 2017 г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

для специальностей второй ступени высшего образования университета

Семестр – 1

Лекции – 40

Лабораторные занятия – 32

Аудиторных часов

По учебной дисциплине – 72

Всего часов

По учебной дисциплине – 108

Кандидатский зачет – 1 семестр

Форма получения высшего образования – дневная, заочная

Составители:

П.В. Гляков, заведующий кафедрой информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств»;

Т.С. Жилинская, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств»;

Т.И. Песецкая, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств»

Рассмотрен и утвержден

на заседании Совета университета 19.09.2017 г.

протокол № 1

Составители:

Гляков Петр Владимирович, заведующий кафедрой информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат физико-математических наук, доцент;

Жилинская Татьяна Степановна, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат педагогических наук;

Песецкая Татьяна Ивановна, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат физико-математических наук

Рецензенты:

В.С. Романчик, заведующий кафедрой Web-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент;

А.М. Стельмах, доцент кафедры менеджмента социокультурной деятельности учреждения образования «Белорусский государственный университет культуры и искусств», кандидат искусствоведения

Рассмотрен и рекомендован к утверждению:

Кафедрой информационных технологий в культуре

(протокол от 19.06.2017 г., № 10);

Советом факультета культурологии и социокультурной деятельности

(протокол от 30.08.2017 г., №)

Оглавление

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	4
2 ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ	6
2.1 Учебные издания	6
2.2 Конспект лекций	6
3 ПРАКТИЧЕСКИЙ РАЗДЕЛ	139
3.1 Описание лабораторных работ	139
4 РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ	201
4.1 Задания для контролируемой самостоятельной работы студентов	201
4.2 Контрольные вопросы	201
4.3 Перечень вопросов к кандидатскому зачету	201
4.4 Критерии оценки результатов учебной деятельности студентов	204
5 ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ	204
5.1 Учебная программа	204
5.2 Учебно-методическая карта учебной дисциплины для дневной формы получения высшего образования	205
5.3 Учебно-методическая карта учебной дисциплины для заочной формы получения высшего образования	205
5.4 Список основной литературы	206
5.5 Список дополнительной литературы	207

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Информационные технологии являются одной из самых динамично развивающихся областей. Совершенствуется элементная база и архитектура компьютеров, развиваются языки программирования, создаются новые пакеты прикладных программ на основе современных математических методов моделирования и оптимизации. Поэтому необходимым элементом подготовки специалистов является как систематизация базовых понятий, так и знакомство с современными достижениями в области информационных технологий.

Учебная дисциплина "Основы информационных технологий" призвана стать основой для понимания процессов информатизации в сфере образования, культуры и искусств.

Подготовка обучающихся к использованию современных информационных технологий как инструмента для решения на высоком уровне научных и практических задач в своей предметной области является целью изучения дисциплины "Основы информационных технологий".

Учебно-методический комплекс по учебной дисциплине «Основы информационных технологий» предназначен для решения поставленных в учебной программе задач:

- ознакомить с современным состоянием информационных технологий;
- систематизировать базовые понятия информационных технологий;
- сформировать умения и навыки работы с текстовыми, графическими и табличными процессорами, системами управления базами данных, средствами подготовки презентаций, средствами поддержки математических вычислений, сервисами Интернета, инструментальными средствами веб-сайтов.

Учебно-методический комплекс включает следующие основные разделы: пояснительную записку, в которой определена цель и задачи дисциплины, теоретический раздел, практический раздел, раздел контроля знаний и вспомогательный раздел.

В теоретическом разделе приведен конспект лекций, основанный на учебно-методических публикациях авторов.

Практический раздел содержит тематику, задания и подробные методические рекомендации по выполнению лабораторных работ, которые имеют творческий характер и культурологическую направленность.

Раздел контроля знаний представлен в соответствии с учебной программой. Он содержит вопросы по каждой теме. Тематика и вопросы способствуют усвоению и закреплению пройденного материала. Здесь так же представлены вопросы к кандидатскому зачету.

Во вспомогательном разделе приведена учебно-методическая карта учебной дисциплины для дневной и заочной форм получения образования, учебная программа, список литературы (основной и дополнительной).

Приведенные в учебно-методическом комплексе лабораторные занятия позволят студентам второй ступени высшего образования университета приобрести устойчивые умения и навыки работать в современных операционных системах; находить необходимую информацию в глобальной сети Интернет; использовать словари, переводчики и архиваторы; создавать информационные ресурсы помощью инструментальных средств и скриптовых языков программирования.

РЕПОЗИТОРИЙ БГУКИ

2 ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

2.1 Учебные издания

1. Бураўкін, А.Г. Інфармацыйныя тэхналогіі ў мастацтве / А.Г. Бураўкін ; пад рэд. М.А. Ярмаша. – Мінск : БУК, 1999. – 246 с.
2. Гляков, П.В. Система управления базами данных Access 2.0 : учеб. пособие / П.В. Гляков, С.Н. Карачун. – Минск : РИПО, 1998. – 100 с.
3. Гляков, П.В. Импорт, экспорт и связывание данных в Microsoft Access : метод. рекомендации / П.В. Гляков. – Минск : РИПО, 2005. – 34 с.
4. Гляков, П.В. Базы данных: компьютерный практикум: учебное пособие / П.В. Гляков. – Минск : БГУКИ, 2008. – 130 с.
5. Жилинская, Т.С. Медиакультура специалиста : учебно-методическое пособие для студентов высших учебных заведений по специальности 1-210401 Культурология (по направлениям), специализации 1-210401-0204 Информационные системы в культуре / Т. С. Жилинская. – Минск : [б. и.], 2011. – с. 62.

2.2 Конспект лекций

Тема 1. Современные информационные технологии

Лекция 1. (2 часа)

Современное состояние информационных технологий.

Основные вопросы:

История, современное состояние и перспективы развития вычислительной техники.

Элементная база, архитектура, сетевая компоновка, производительность.

Понятие информации.

Классификация и виды информационных технологий.

Цель: изучить состояние информационных технологий на современном этапе развития

Первый этап в развитии электронной вычислительной техники (ЭВМ) базировался на электронных лампах. Но электронные лампы обладали существенными недостатками: большие размеры, низкая надежность и др. Поэтому начала развиваться твердотельная электроника, а в качестве элементной базы стали применять диоды и транзисторы. Компьютеры, основанные на транзисторах, не устранили полностью эти недостатки. Для решения проблем начали применяться микросборки, а затем и микросхемы.

Число элементов микросхем постепенно увеличивалось, стали появляться микропроцессоры. В настоящее время развитию электроники способствует появление сотовой связи, а также различных беспроводных устройств, навигаторов, коммуникаторов, планшетов и т. п.

Архитектура ЭВМ — концептуальная структура вычислительной машины, определяющая проведение обработки информации и включающая методы информации в данные и принципы взаимодействия технических средств и программного обеспечения.

Архитектура ЭВМ определяется совокупностью общих принципов организации аппаратно-программных средств и их основных характеристик, обеспечивающих функциональные возможности вычислительной машины при решении соответствующих типов задач. Архитектура ЭВМ включает в себя как структуру, отражающую состав компьютера, так и программно – математическое обеспечение.

Структура ЭВМ - совокупность элементов и связей между ними. Основным принципом построения всех современных ЭВМ является программное управление.

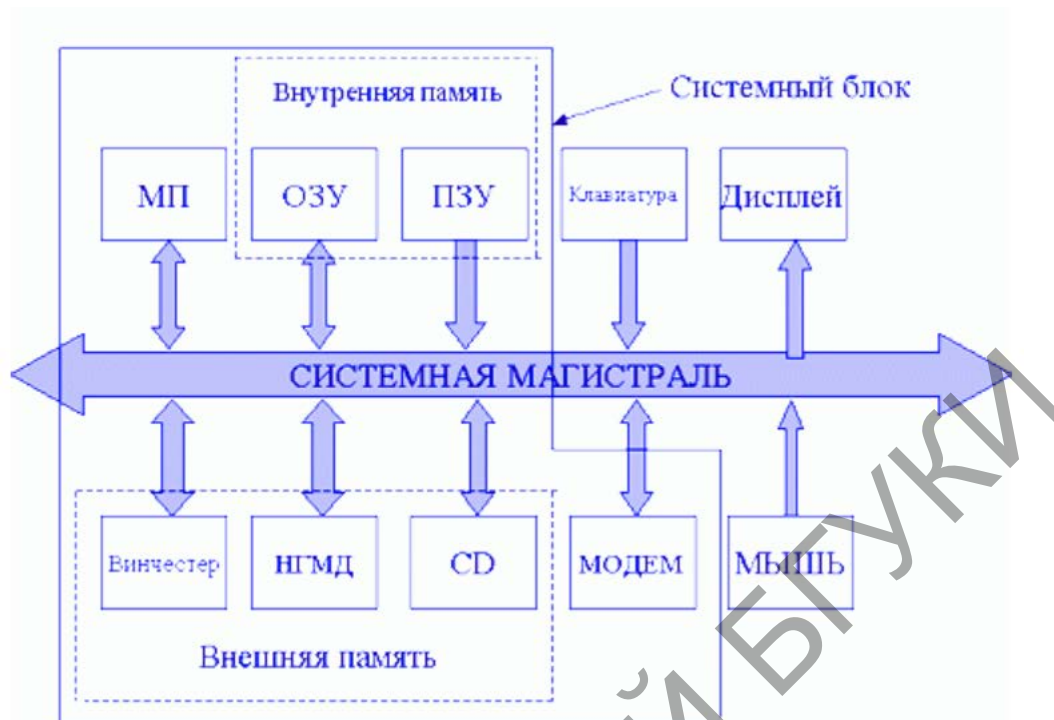
Основы учения об архитектуре вычислительных машин были заложены Джон фон Нейманом. Совокупность этих принципов породила классическую архитектуру ЭВМ.

		Внешнее запоминающее устройство			
Устройство ввода	Процессор Арифметико-логическое устройство	Устройство управления	Устройство вывода		
		Оперативное запоминающее устройство			

Сущность этих принципов заключалась в следующем:

- компьютер состоит из нескольких основных устройств (арифметико-логическое устройство, устройство управления, память, внешняя память, устройства ввода и вывода);
- арифметико-логическое устройство (АЛУ) – выполняет арифметические и логические операции, необходимые для переработки информации, хранящейся в памяти;
- устройство управления (УУ) – обеспечивает управление и контроль всех устройств компьютера (управляющие сигналы на рисунке указаны пунктирными стрелками);
- данные, которые хранятся в запоминающем устройстве (ЗУ), представляются в двоичной форме;
- программа, которая задает работу компьютера, и данные хранятся в одном и том же запоминающем устройстве;

- для ввода и вывода информации используются устройства ввода и вывода.



Принцип работы и структуру ЭВМ рассмотрим на примере персонального компьютера (ПК). На рисунке представлена схема структуры ПК. Основу ПК составляет системный блок, в котором размещены: микропроцессор (МП), блок оперативного запоминающего устройства (ОЗУ), постоянного запоминающего устройства (ПЗУ), долговременной памяти на жёстком магнитном диске (Винчестер), устройства для запуска компакт-дисков (CD) и дискет (НГМД). Там же находятся платы: сетевая, видеопамяти, обработки звука, модем (модулятор-демодулятор), интерфейсные платы, обслуживающие устройства ввода-вывода: клавиатуры, дисплея, "мыши", принтера и др.

Все функциональные узлы ПК связаны между собой через системную магистраль (шину), представляющую из себя более трёх десятков упорядоченных микропроводников, сформированных на печатной плате.

Микропроцессор служит для обработки информации: он выбирает команды из внутренней памяти (ОЗУ или ПЗУ), расшифровывает и затем исполняет их, производя арифметические и логические операции. Получает данные из устройства ввода и посылает результаты на устройства вывода. Он вырабатывает также сигналы управления и синхронизации для согласованной работы его внутренних узлов, контролирует работу системной магистрали и всех периферийных устройств.

С развитием вычислительной техники появились и приобрели широкое использование системы физического соединения двух или более компьютеров - компьютерные сети. По территориально-организационным признакам (количеству машин и расстоянию между ними) компьютерные сети принято разделять на локальные и глобальные. Глобальные сети (например, Internet)

распространяют свое действие по всему миру и используют все каналы связи, включая спутниковые.

Архитектурный принцип построения большинства сетей называется «клиент-сервер». Сервер - компьютер сети, предоставляющий свои программные и аппаратные ресурсы пользователям сети для хранения данных, выполнения программ и других услуг. Клиент - компьютер сети, пользующийся услугами сервера; в его роли часто выступают программы, имеющие доступ к информационным ресурсам или устройствам сервера. Термины «клиент» и «сервер» используются для обозначения как программных, так и аппаратных средств.

Для передачи данных в сети используются специальные стандарты, обеспечивающие их совместимость - сетевые протоколы. Примером универсального протокола является семейство TCP/IP, широко применяющееся во всем мире для объединения компьютеров в сеть Internet которая состоит из множества сетей различной физической природы.

История его возникновения связана с задачей, поставленной после второй мировой войны правительством США. Требовалось создать единую сеть, которая могла бы своими средствами находить маршруты передачи данных, а также в случае повреждения некоторых каналов связи перенаправлять поток информации по другим каналам. При реализации этого проекта были созданы отдельные представители семейства протоколов TCP/IP.

Информация – это совокупность каких-либо сведений, данных, передаваемых устно (в форме речи), письменно (в виде текста, таблиц, рисунков, чертежей, схем, условных обозначений)либо другим способом (например, с помощью звуковых или световых сигналов, электрических и нервных импульсов, перепадов давления или температуры и т.д.).

В середине 20 века термин «информация» стал общенаучным понятием, включающим обмен сведениями между людьми, человеком и автоматом (ЭВМ), автоматом и автоматом, обмен сигналами в животном и растительном мире, передачу признаков от клетки к клетке, от организма к организму.

Классификация информационных технологий

1. По методам и средствам обработки данных:

- глобальные ИТ включают модели, методы и средства использования информационных ресурсов в обществе в целом;

- базовые ИТ ориентированны на определенную область применения: производство, научные исследования, проектирование, обучение и т.д.;

- конкретные ИТ задают обработку данных в реальных задачах пользователя.

2. По обслуживаемым предметным областям:

- ИТ в бухгалтерском учете;

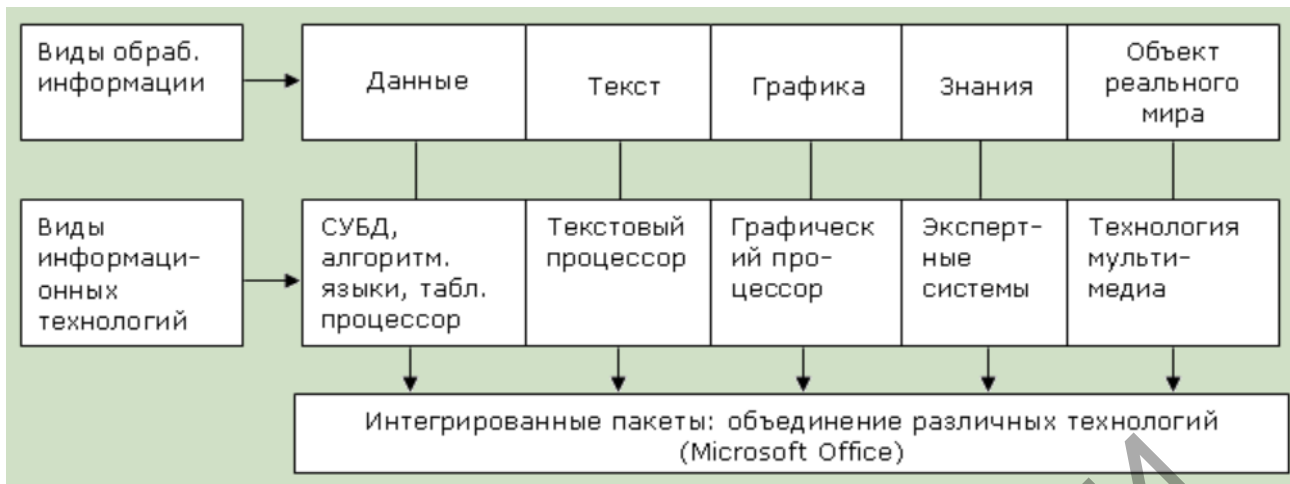
- ИТ в банковской деятельности;

- ИТ в налоговой деятельности;

- ИТ в страховой деятельности;

- ИТ в статистической деятельности и т.д.

3. По видам обрабатываемой информации



4. По типу пользовательского интерфейса

Пользовательский интерфейс - взаимодействие компьютера с пользователем.

Эта классификация позволяет говорить о системном и прикладном интерфейсе.

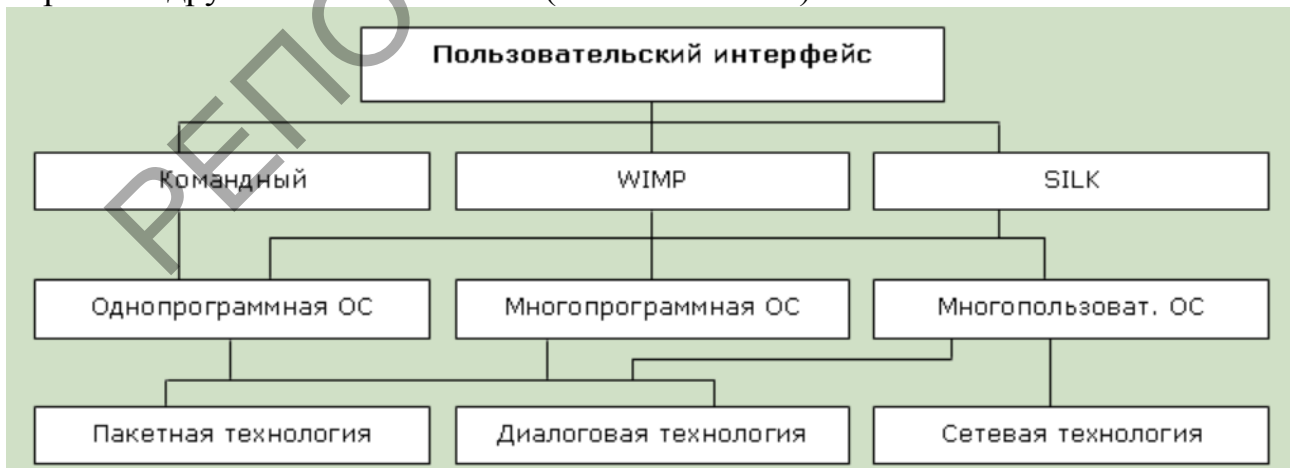
Прикладной интерфейс связан с реализацией некоторых функциональных информационных технологий.

Системный интерфейс - набор приемов взаимодействия с компьютерами, которое реализуется операционной системой или ее надстройкой.

Командный интерфейс - самый простой, обеспечивает выдачу на экран системного приглашения для ввода команды (в ОС MS DOS системное приглашение: C:\>, в ОС Unix - \$).

WIMP - интерфейс . При его использовании на экране высвечивается окно, содержащее образы программ и меню действий. Для выбора одного из них используется указатель мыши.

SILK - интерфейс. При использовании этой информационной технологии на экране по речевой команде происходит перемещение от одних поисковых образов к другим по смысловым (семантическим) связям.



Однопрограммная операционная система - MS DOS.

Многопрограммные операционные системы: Unix, Windows 3.1 и выше, Dos 7.0 позволяют одновременно выполнять несколько приложений на рабочем месте одного пользователя. Различаются они алгоритмом деления времени.

Если однопрограммные операционные системы работают или в диалоговом или в пакетном режимах, то многопрограммные совмещают указанные режимы.

Многопользовательские системы - реализуется сетевыми операционными системами. Они обеспечивают удаленные сетевые технологии, а также пакетную и диалоговую технологии на рабочем месте пользователя.

Большинство обеспечивающих информационных технологии и функциональных информационных технологии могут быть использованы управленческим работником без дополнительных посредников (программистов). При этом пользователь может влиять на последовательность применения тех или иных технологий.

Пакетные ИС работают в пакетном режиме: вначале данные накапливаются, и формируется пакет данных, а затем пакет последовательно обрабатывается рядом программ. Недостаток этого режима - низкая оперативность принятия решений и обособленность пользователя от системы.

Экономические задачи, решаемые в пакетном режиме, характеризуются следующими свойствами:

- алгоритм решения задач формализован, процесс ее решения не требует вмешательства человека;
- имеется большой объем входных и выходных данных, значительная часть которых хранится на магнитных носителях;
- расчет выполняется для большинства записей входных файлов;
- большее время решения задачи обусловлено большим объемом данных;
- регламентность, т.е. задачи решаются с заданной периодичностью.

Диалоговые ИС работают в режиме обмена сообщениями между пользователями и системой (например, система продажи авиабилетов). Этот режим особенно удобен, когда пользователь может выбирать перспективные варианты из числа предлагаемых системой.

Диалоговый режим (интерактивный) является развитием пакетного режима. Если применение пакетного режима позволяет уменьшать вмешательство пользователя в процесс задачи, то диалоговый режим предполагает отсутствие жестко закрепленной последовательности операций обработки данных.

Информационные системы. Сетевые технологии обеспечивают взаимодействие многих пользователей. Все объекты представляют собой так называемую систему. Их поведение, характеристики рассматриваются в системном объекте.

Система - это образующая единое целое совокупность материальных и нематериальных объектов, объединенных некоторыми общими признаками, назначениями, свойствами, условиями существования, жизнедеятельности, функционирования и т.д.

Функционирование системы - процесс переработки входной информации в выходную, носящий последовательный характер во времени.

Подсистема - часть любой системы.

Свойства системы (в т.ч. ИС):

•сложность - система зависит от множества входящих в нее компонентов, их структурного взаимодействия, а так же сложности внутренних и внешних связей;

•делимость - система состоит из ряда подсистем или элементов, выделенных по определенным признакам и отвечающих конкретным целям и задачам;

•целостность системы - означает то, что все элементы системы функционируют как единое целое;

•многообразие элементов системы и различие их природы - свойство связано с функционированием элементов, их спецификой и автономностью;

•структурность - определяет наличие установленных связей и отношений между элементами внутри системы, распределение элементов системы по уровням и иерархиям;

•адаптивность системы - означает приспособляемость системы к условиям конкретной предметной области;

•интегрируемость - означает возможность взаимодействия системы с вновь подключаемыми компонентами или подсистемами.

Системы значительно отличаются между собой как по составу, так и по главным целям. Приведем несколько систем, состоящих из разных элементов и направленных на реализацию разных целей.

Информационная система - это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи экономической информации и принятия управленческих решений.

Система	Элементы системы	Главная цель системы
Фирма	Люди, оборудование, материалы, здания и др.	Производство товаров
Компьютер	Электронные и электромеханические элементы, линии связи и др.	Обработка данных
Телекоммуникационная система	Компьютеры, модемы, кабели, сетевое программное обеспечение и др.	Передача информации
Информационная система	Компьютеры, компьютерные сети, люди, информационное и программное обеспечение	Производство профессиональной информации

Свойства информационных систем:

•любая ИС может быть подвергнута анализу, построена и управляема на основе общих принципов построения сложных систем;

•при построении ИС необходимо использовать системный подход;

•ИС является динамичной и развивающейся системой;

•ИС следует воспринимать как систему обработки информации, состоящую из компьютерных и телекоммуникационных устройств, реализованную на базе современных технологий;

•выходной продукцией ИС является информация, на основе которой принимаются решения или производятся автоматическое выполнение рутинных операций;

- участие человека зависит от сложности системы, типов и наборов данных, степени формализации решаемых задач.

Процессы в информационной системе:

- ввод информации из внешних и внутренних источников;
- обработка входящей информации;
- хранение информации для последующего ее использования;
- вывод информации в удобном для пользователя виде;
- обратная связь, т.е. представление информации, переработанной в данной организации, для корректировки входящей информации.

С учетом сферы применения выделяют: технические ИС, экономические ИС, ИС в гуманитарных областях и т.д.

Экономическая информационная система (ЭИС) представляет собой систему, функционирование которой во времени заключается в сборе, хранении, обработке и распространении информации о деятельности какого-то экономического объекта реального мира. ЭИС предназначены для решения задач обработки данных, автоматизации конторских работ, выполнения поиска информации и отдельных задач, основанных на методах искусственного интеллекта.

В зависимости от сферы применения ЭИС классифицируются:

- ИС фондового рынка;
- страховые ИС;
- статистические ИС;
- ИС в налоговой сфере;
- ИС в таможенной деятельности;
- финансовые ИС;
- банковские ИС (БИС);
- ИС промышленных предприятий и организаций (в этот контур входят бухгалтерские ИС - БуИС).

Этапы развития информационных систем.

Первые ИС появились в 50-х гг. В эти годы они были предназначены для обработки счетов и расчета зарплаты, а реализовывались на электромеханических бухгалтерских счетных машинах. Это приводило к некоторому сокращению затрат и времени на подготовку бумажных документов.

60-е гг. знаменуются изменением отношения к ИС. Информация, полученная из них, стала применяться для периодической отчетности по многим параметрам. Для этого организациям требовалось компьютерное оборудование широкого назначения, способное обслуживать множество функций, а не только обрабатывать счета и считать з/пл.

В 70-х - начале 80-х ИС начинают широко использоваться в качестве средства управленческого контроля, поддерживающего и ускоряющего процесс принятия решений.

К концу 80-х гг. концепция использования ИС вновь изменяется. Они становятся стратегическим источником информации и используются на всех уровнях организации любого профиля. ИС этого периода, предоставляя

вовремя нужную информацию, помогают организации достичь успеха в своей деятельности, создавать новые товары и услуги, находить новые рынки сбыта, обеспечивать себе достойных партнеров, организовывать выпуск продукции по низкой цене и многое другое.

Соотношение между ИС и ИТ.

Информационная технология - процесс различных операций и действий над данными. Все процессы преобразования информации в информационной системе осуществляются с помощью информационных технологий.

Информационная система - среда, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технологические и программные средства и т.д.

Таким образом, информационная технология является более емким понятием, чем информационная система. Реализация функций информационной системы невозможна без знаний ориентированной на нее информационной технологии. Информационная технология может существовать и вне сферы информационной системы.

Лекция 2 (2 часа)

Операционные системы и технологии программирования

Основные вопросы

Операционные системы.

Языки программирования.

Технологии программирования.

Процедурное, объектно-ориентированное, логическое программирование.

Цель: Изучить основные операционные системы и технологии программирования.

Операционная система (ОС) - это программа, которая обеспечивает возможность рационального использования оборудования компьютера удобным для пользователя образом.

Структура вычислительной системы

Из чего состоит любая вычислительная система? В первую очередь, это hardware, или техническое обеспечение: процессор, память, монитор, дисковые устройства и т.д., обычно объединенные магистральным соединением, которое называется шиной

Во вторую очередь это программное обеспечение. Все программное обеспечение принято делить на две части: прикладное и системное.

К прикладному программному обеспечению, как правило, относятся разнообразные банковские и прочие business программы, игры, текстовые процессоры, и т.п.

Под системным программным обеспечением обычно понимают программы, способствующие функционированию и разработке прикладных программ.

Деление на прикладное и системное программное обеспечение является отчасти условным и зависит от того, кто осуществляет такое деление.

Так, обычный пользователь, неискушённый в программировании, может считать Microsoft Word системной программой, а с точки зрения программиста это приложение. Компилятор языка Си для обычного программиста это системная программа, а для системного прикладная. Несмотря на эту нечеткую грань, эту ситуацию можно отобразить в виде последовательности слоев:

Операционная система как виртуальная машина

Архитектура большинства компьютеров на уровне машинных команд очень неудобна для ее использования прикладными программами. Например, работа с диском предполагает знакомство с внутренним устройством его электронного компонента - контроллера для ввода команд вращения диска, поиска и форматирования дорожек, чтения и записи секторов и т.д. Ясно, что средний программист не в состоянии учитывать все особенности работы оборудования (в современной терминологии заниматься разработкой драйверов устройств), а должен иметь простую высокоуровневую абстракцию, скажем, представляя информационное пространство диска как набор файлов

Файл можно открывать для чтения или записи, использовать для получения или сброса информации, а потом закрывать. Это концептуально проще, чем заботиться о деталях перемещения головок дисков или организации работы мотора. Аналогичным образом, с помощью простых и ясных абстракций, скрываются от программиста все ненужные ему подробности организации прерываний, работы таймера, управления памятью и т.д. Более того, на современных вычислительных комплексах может быть создана иллюзия неограниченного размера операционной памяти и числа процессоров. Всем этим занимается операционная система. Таким образом, операционная система представляется пользователю виртуальной машиной, с которой проще иметь дело, чем непосредственно с оборудованием компьютера.

Операционная система как менеджер ресурсов

Операционная система предназначена для управления всеми частями весьма сложной архитектуры компьютера. Представим для примера, что случится, если несколько программ, работающих на одном компьютере, будут пытаться одновременно осуществлять вывод на принтер. Мы получили бы неупорядоченную смесь строчек и страниц, выведенных различными программами. Операционная система предотвращает хаос такого рода за счет буферизации информации, предназначенной для печати, на диске и организации очереди на печать. Для многопользовательских компьютеров, необходимость управления ресурсами и их защиты еще более очевидна.

Следовательно, операционная система как менеджер ресурсов, осуществляет упорядоченное и контролируемое распределение процессоров, памяти и других ресурсов между различными программами, их использующими.

Операционная система как защитник пользователей и программ

Если вычислительная система допускает совместную работу нескольких пользователей, то возникает проблема организации их безопасной

деятельности. Необходимо обеспечить сохранность информации на диске, чтобы никто не мог удалить или повредить чужие файлы. Нельзя разрешить программам одних пользователей произвольно вмешиваться в работу программ других пользователей. Нужно пресекать попытки несанкционированного использования вычислительной системы. Всю эту деятельность осуществляет операционная система как организатор безопасной работы пользователей и их программ. С такой точки зрения операционная система выглядит системой безопасности в государстве, на которую возложены полицейские и контрразведывательные функции.

Операционная система как постоянно функционирующее ядро

Наконец, можно дать и такое определение: операционная система это программа, постоянно работающая на компьютере и взаимодействующая со всеми прикладными программами. Казалось бы, это абсолютно правильное определение, но во многих современных операционных системах постоянно работает на компьютере лишь часть операционной системы, которую принято называть ее ядром.

Краткая история эволюции вычислительных систем

- Первый период (1945 - 1955). Ламповые машины. Операционные систем отсутствовали.
- Второй период (1955 - Начало 60-х). Компьютеры на основе транзисторов. Пакетные операционные системы
- Третий период (Начало 60-х - 1980). Компьютеры на основе интегральных микросхем. Первые многозадачные ОС.
- Четвертый период (1980 – настоящее время). Персональные компьютеры. Классические, сетевые и распределенные системы.

Появление мультипрограммирования требует целой революции в строении вычислительной системы. Большую роль, здесь играет аппаратная поддержка, наиболее существенные особенности которой:

Реализация защитных механизмов. Программы не должны иметь самостоятельного доступа к распределению ресурсов, что приводит к появлению привилегированных и непривилегированных команд. Привилегированные команды, например команды ввода-вывода, могут исполняться только операционной системой. Говорят, что она работает в привилегированном режиме. Переход управления от прикладной программы к ОС сопровождается контролируемой сменой режима. Во-вторых, это защита памяти, позволяющая изолировать конкурирующие пользовательские программы друг от друга, а ОС от программ пользователей.

Наличие прерываний. Внешние прерывания оповещают ОС о том, что произошло асинхронное событие, например, завершилась операция ввода-вывода. Внутренние прерывания (сейчас их принято называть исключительными ситуациями) возникают, когда выполнение программы привело к ситуации, требующей вмешательства ОС, например, деление на ноль или попытка нарушения защиты.

Роль операционной системы в организации мультипрограммирования

Интерфейс между прикладной программой и ОС был организован при помощи набора системных вызовов.

Организация очереди из заданий в памяти и выделение процессора одному из заданий потребовали планирования заданий.

Для переключения процессора с одного задания на другое возникла потребность в сохранении содержимого регистров и структур данных, необходимых для выполнения задания, иначе говоря, контекста, для обеспечения правильного продолжения вычислений.

Поскольку память является ограниченным ресурсом, оказались нужны стратегии управления памятью, то есть потребовалось упорядочить процессы размещения, замещения и выборки информации из памяти.

Так как программы могут пожелать произвести санкционированный обмен данными, стало необходимо их обеспечить средствами коммуникации.

Для корректного обмена данными необходимо предусмотреть координацию программами своих действий, т.е. средства синхронизации.

Основные функции классических операционных:

- Планирование заданий и использования процессора.
- Обеспечение программ средствами коммуникации и синхронизации.
- Управление памятью.
- Управление файловой системой.
- Управление вводом-выводом.
- Обеспечение безопасности

Каждая из приведенных функций обычно реализована в виде подсистемы, являющейся структурным компонентом ОС. В каждой конкретной операционной системе эти функции, конечно, реализовывались по-своему, в различном объеме. Они не были придуманы как составные части деятельности операционных систем изначально, а появились в процессе развития, по мере того, как вычислительные системы становились удобнее, эффективнее и безопаснее.

Основные понятия ОС

- Системные вызовы
- Прерывания
- Исключительные ситуации
- Файлы
- Процессы, нити
- Системные вызовы

В любой операционной системе поддерживается некоторый механизм, который позволяет пользовательским программам обращаться за услугами ядра ОС. Эти средства общения с ядром часто назывались экстракодами или системными макрокомандами. В ОС UNIX такие средства называются системными вызовами.

Системные вызовы (system calls) интерфейс между операционной системой и пользовательской программой. Они создают, удаляют и используют различные объекты, главные из которых процессы и файлы. Пользовательская

программа запрашивает сервис у операционной системы, осуществляя системный вызов. Имеются библиотеки процедур, которые загружают машинные регистры определенными параметрами и осуществляют прерывание процессора, после чего управление передается обработчику данного вызова, входящему в ядро операционной системы. Цель таких библиотек сделать системный вызов похожим на обычный вызов подпрограммы.

Основное отличие состоит в том, что при системном вызове задача переходит в привилегированный режим или режим ядра (kernel mode). Поэтому системные вызовы иногда еще называют программными прерываниями в отличие от аппаратных прерываний, которые чаще называют просто прерываниями.

В этом режиме работает код ядра операционной системы, причем он исполняется в адресном пространстве и в контексте вызвавшей его задачи. Таким образом, ядро операционной системы имеет полный доступ к памяти пользовательской программы, и при системном вызове достаточно передать адреса одной или нескольких областей памяти с параметрами вызова и адреса одной или нескольких областей памяти для результатов вызова.

В большинстве операционных систем системный вызов осуществляется командой программного прерывания (INT). Таким образом, программное прерывание это синхронное событие.

Прерывания

Прерывание (hardware interrupt) событие, генерируемое внешним (по отношению к процессору) устройством. Посредством аппаратных прерываний аппаратура либо информирует центральный процессор о том, что возникло какое-либо событие, требующее немедленной реакции (например, пользователь нажал клавишу), либо сообщает о завершении асинхронной операции ввода-вывода (например, закончено чтение данных с диска в основную память).

Важный тип аппаратных прерываний - прерывания таймера, которые генерируются периодически через фиксированный промежуток времени. Прерывания таймера используются операционной системой при планировании процессов. Каждый тип аппаратных прерываний имеет собственный номер, однозначно определяющий источник прерывания.

Аппаратное прерывание это асинхронное событие, то есть оно возникает вне зависимости от того, какой код исполняется процессором в данный момент. Обработка аппаратного прерывания не должна учитывать, какой процесс является текущим.

Исключительные ситуации

Исключительная ситуация (exception) событие, возникающее в результате попытки выполнения программой недопустимой команды, доступа к ресурсу при отсутствии достаточных привилегий или обращения к отсутствующей странице памяти.

Исключительные ситуации так же, как и системные вызовы, являются синхронными событиями, возникающими в контексте текущей задачи. Исключительные ситуации можно разделить на исправимые и неисправимые. К исправимым относятся такие исключительные ситуации, как отсутствие

нужной информации в оперативной памяти. После устранения причины исправимой исключительной ситуации программа может продолжить выполнение.

Возникновение в процессе работы операционной системы исправимых исключительных ситуаций является нормальным явлением. Неисправимые исключительные ситуации обычно возникают в результате ошибок в программах. Обычно операционная система реагирует на такие ситуации завершением программы, вызвавшей исключительную ситуацию.

Файлы

Файлы предназначены для хранения информации на внешних носителях, то есть, принято, что информация, лежащая, например, на диске, должна находиться внутри файла. Обычно под файлом понимают часть пространства на носителе информации, имеющую имя.

Главная задача файловой системы (file system) скрыть особенности ввода-вывода и дать программисту простую абстрактную модель файлов, независимых от устройств. Для чтения, создания, удаления, записи, открытия и закрытия файлов также имеется обширная категория системных вызовов (create, delete, open, close, read, write).

Пользователям хорошо знакомы такие понятия, связанные с организацией файловой системы, как каталог, текущий каталог, корневой каталог, путь, для манипулирования которыми в операционной системе имеются системные вызовы.

Слоеные системы (Layered systems)

Продолжая структуризацию, можно разбить всю вычислительную систему на ряд более мелких уровней с хорошо определенными связями между ними, так чтобы объекты уровня N могли вызывать только объекты из уровня $N-1$. Нижним уровнем в таких системах обычно является hardware, верхним уровнем интерфейс пользователя. Чем ниже уровень, тем более привилегированные команды и действия может выполнять модуль, находящийся на этом уровне. Впервые такой подход был применен при создании системы TNE (Technische Hogeschool Eindhoven) Дейкстрой и его студентами в 1968 г.

Слоеные системы хорошо реализуются. При использовании операций нижнего слоя не нужно знать, как они реализованы, нужно знать лишь, что они делают.

Слоеные системы хорошо тестируются. Отладка начинается с нижнего слоя и проводится послойно. При возникновении ошибки мы можем быть уверены, что она находится в тестируемом слое.

Слоеные системы хорошо модифицируются. При необходимости можно заменить лишь один слой, не трогая остальные. Но слоеные системы сложны для разработки: тяжело правильно определить порядок слоев, и что, к какому слою относится.

Слоеные системы менее эффективны, чем монолитные. Так, например, для выполнения операций ввода-вывода программе пользователя придется последовательно проходить все слои - от верхнего до нижнего.

Виртуальные машины

Операционная система как виртуальная машина - это когда пользователю нет необходимости знать детали внутреннего устройства компьютера. Он работает с файлами, а не с магнитными головками и двигателем; он работает с огромной виртуальной, а не ограниченной реальной оперативной памятью; его мало волнует, единственный он на машине пользователь или нет.

Рассмотрим несколько другой подход. Пусть операционная система реализует виртуальную машину для каждого пользователя, но, не упрощая ему жизнь, а, наоборот, усложняя. Каждая такая виртуальная машина предстает перед пользователем как абсолютно голое железо копия всего hardware в вычислительной системе, включая процессор, привилегированные и непривилегированные команды, устройства ввода-вывода, прерывания и т.д. И он один на один с этим железом. При попытке обратиться к этому виртуальному железу на уровне привилегированных команд, в действительности происходит системный вызов реальной операционной системы, которая и производит все необходимые действия. Такой подход позволяет каждому пользователю загрузить свою собственную операционную систему на виртуальную машину и делать с ней все, что душа пожелает.

Микроядерная архитектура

Современная тенденция в разработке операционных систем это перенесение значительной части системного кода на уровень пользователя и одновременной минимизации ядра. Речь идет о подходе к построению ядра, называемом микроядерной архитектурой (microkernel architecture) операционной системы, когда большинство ее составляющих являются самостоятельными программами.

В этом случае взаимодействие между ними обеспечивает специальный модуль ядра, называемый микроядром. Микроядро работает в привилегированном режиме и обеспечивает взаимодействие между программами, планирование использования процессора, первичную обработку прерываний, операции ввода-вывода и базовое управление памятью.

Остальные компоненты системы взаимодействуют друг с другом путем передачи сообщений через микроядро.

Основное достоинство микроядерной архитектуры высокая степень модульности ядра операционной системы. Это существенно упрощает добавление в него новых компонент. В микроядерной операционной системе можно, не прерывая ее работы, загружать и выгружать новые драйверы, файловые системы и т. д.

Существенно упрощается процесс отладки компонент ядра, так как новая версия драйвера может загружаться без перезапуска всей операционной системы. Компоненты ядра операционной системы ничем принципиально не отличаются от пользовательских программ, поэтому для их отладки можно применять обычные средства. В то же время, микроядерная архитектура операционной системы вносит дополнительные накладные расходы, связанные с передачей сообщений, что существенно влияет на производительность.

Для того чтобы микроядерная операционная система по скорости не уступала операционным системам на базе монолитного ядра, требуется очень аккуратно проектировать разбиение системы на компоненты, стараясь минимизировать взаимодействие между ними. Таким образом, основная сложность при создании микроядерных операционных систем необходимость очень аккуратного проектирования.

Смешанные системы

Все рассмотренные подходы к построению операционных систем имеют свои преимущества и недостатки. В большинстве случаев современные операционные системы используют различные комбинации этих подходов.

Примером смешанного подхода может служить возможность запуска операционной системы с монолитным ядром под управлением микроядра. Так устроены 4.4BSD и MkLinux, основанные на микроядре Mach. Микроядро обеспечивает управление виртуальной памятью и работу низкоуровневых драйверов. Все остальные функции, в том числе взаимодействие с прикладными программами, осуществляется монолитным ядром. Данный подход возник в результате попыток использовать преимущества микроядерной архитектуры, сохраняя по возможности хорошо отлаженный код монолитного ядра.

Смешанные системы

Наиболее тесно элементы микроядерной архитектуры и элементы монолитного ядра переплетены в ядре Windows NT. Хотя Windows NT часто называют микроядерной операционной системой, это не совсем так. Микроядро NT слишком сложно и велико (более 1 Мб), чтобы носить приставку микро. Компоненты ядра Windows NT располагаются в вытесняемой памяти и взаимодействуют друг с другом путем передачи сообщений, как и положено в микроядерных операционных системах. В тоже время все компоненты ядра работают в одном адресном пространстве и активно используют общие структуры данных, что свойственно операционным системам с монолитным ядром. Кроме того, в Windows NT существует разделение между режимом ядра и режимом пользователя еще одна черта монолитного ядра. Причина всего этого проста. По мнению Microsoft, причина проста: чисто микроядерный дизайн коммерчески непрактичен, так как слишком неэффективен.

Таким образом, Windows NT можно с полным правом назвать гибридной операционной системой.

Классификация ОС

Существует несколько схем классификации операционных систем. Ниже приведена классификация по некоторым признакам с точки зрения пользователя:

- Реализация многозадачности
- Поддержка многопользовательского режима.
- Многопроцессорная обработка
- Системы реального времени
- Реализация многозадачности

По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

- многозадачные (Unix, OS/2, Windows).
- однозадачные (например, MS-DOS) и

Многозадачная ОС, решая проблемы распределения ресурсов и конкуренции, полностью реализует мультипрограммный режим(см.слайд 14)

Приблизительность классификации очевидна из приведенных примеров. Так в ОС MS-DOS можно организовать запуск дочерней задачи и одновременное сосуществование в памяти двух и более задач. Однако эта ОС традиционно считается однозадачной, главным образом из-за отсутствия защитных механизмов и коммуникационных возможностей

Поддержка многопользовательского режима

По числу одновременно работающих пользователей ОС можно разделить на:

- однопользовательские (MS-DOS, Windows 3.x);
- многопользовательские (Windows NT, Unix).

Наиболее существенно отличие заключается в наличии у многопользовательских систем механизмов защиты персональных данных каждого пользователя.

Многопроцессорная обработка

Многопроцессорные системы состоят из двух или более центральных процессоров, осуществляющих параллельное выполнение команд. Поддержка мультипроцессорирования является важным свойством ОС и приводит к усложнению всех алгоритмов управления ресурсами. Многопроцессорная обработка реализована в таких ОС, как Linux, Solaris, Windows NT и в ряде других.

Многопроцессорные ОС разделяют на симметричные и асимметричные. В симметричных ОС на каждом процессоре функционирует одно и то же ядро и задача может быть выполнена на любом процессоре, то есть обработка полностью децентрализована.

В асимметричных ОС процессоры неравноправны. Обычно существует главный процессор (master) и подчиненные (slave), загрузку и характер работы которых определяет главный процессор.

Системы реального времени

В разряд многозадачных ОС, наряду с пакетными системами и системами разделения времени, включаются также системы реального времени, не упоминавшиеся до сих пор.

Они используются для управления различными техническими объектами или технологическими процессами. Такие системы характеризуются предельно допустимым временем реакции на внешнее событие, в течение которого должна быть выполнена программа, управляющая объектом. Система должна обрабатывать поступающие данные быстрее, чем те могут поступать, причем от нескольких источников одновременно.

Столь жесткие ограничения сказываются на архитектуре систем реального времени, например, в них может отсутствовать виртуальная память, поддержка которой дает непредсказуемые задержки в выполнении программ.

Приведенная классификация ОС не является исчерпывающей.

Языки и технологии программирования

Язык программирования – формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программы и действия, которые выполнит исполнитель (компьютер) под её управлением. Со времени создания первых программируемых машин человечество придумало более 2500. Каждый год их число увеличивается. Наиболее известные языки программирования:

Ада, APL, ActionScript, АВАР/4, AutoIt, AWK, Бейсик, Си, Кобол, С++ , С#, Clarion, Clojure, ColdFusion, Common Lisp, dBase, Delphi, Eiffel, Erlang, Euphoria, F#, Форт, Фортран, Gambas, Go, Groovy, HAL/S, Haskell, Icon, Java, JavaScript, Limbo, Lua, Модула-3, Object Pascal, Objective-C, OCaml, Oz, Parser, Pascal, Object Pascal (Delphi)....

Программирование – сравнительно молодая и быстро развивающаяся отрасль науки и техники. Опыт ведения реальных разработок и совершенствования имеющихся программных и технических средств постоянно переосмысливается, в результате чего появляются новые методы, методологии и технологии, которые, в свою очередь, служат основой более современных средств разработки программного обеспечения. Исследовать процессы создания новых технологий и определять их основные тенденции целесообразно, сопоставляя эти технологии с уровнем развития программирования и особенностями имеющихся в распоряжении программистов программных и аппаратных средств.

Технологией программирования называют совокупность методов и средств, используемых в процессе разработки программного обеспечения. Как любая другая технология, технология программирования представляет собой набор технологических инструкций, включающих:

- указание последовательности выполнения технологических операций;
- перечисление условий, при которых выполняется та или иная операция;
- описания самих операций, где для каждой операции определены исходные данные, результаты, а также инструкции, нормативы, стандарты, критерии и методы оценки и т. п.

Кроме набора операций и их последовательности, технология также определяет способ описания проектируемой системы, точнее модели, используемой на конкретном этапе разработки.

Различают технологии, используемые на конкретных этапах разработки или для решения отдельных задач этих этапов, и технологии, охватывающие несколько этапов или весь процесс разработки. В основе первых, как правило, лежит ограниченно применимый метод, позволяющий решить конкретную задачу. В основе вторых обычно лежит базовый метод или подход (парадигма),

определяющий совокупность методов, используемых на разных этапах разработки, или методологию.

В настоящее время выдел несколько типов языков программирования. Одним из важн признаков их классиф - принадлежность их к одному из стилей, осн-ми из кот являются следующие стили:

- 1) процедурный,
- 2) логический,
- 3) объектно-ориентированный.

1. Процедурное программирование является отражением архитектуры традиционных ЭВМ, которая была предложена фон Нейманом в 40-х годах.

Программа на этом языке программирования состоит из последовательности операторов (инструкций), задающих процедуру решения задачи. Основным явл оператор присваивания, служащий для изменения содержания областей памяти. Концепция памяти как хранилища значений, содержимое которого может обновляться операторами программы, является фундаментальной в императивном программировании.

Процедурные языки характеризуются следующими особенностями:

- 1) необходимостью явного управления памятью, в частности, описанием переменных;
- 2) малой пригодностью для символьных вычислений;
- 3) отсутствием строгой математической основы;
- 4) высокой эффективностью реализации на традиционных ЭВМ.

Одним из важнейших классификационных признаков процедурного языка является его уровень. Чем более язык ориентирован на человека, тем выше его уровень.

К процедурным языкам относятся: язык Ассемблера, С, Basic, Pascal.

2. Логическое программирование

Новую область — логическое, или реляционное программирование, — открыло появление языка PROLOG. Этот язык был создан французским ученым А. Кольмероз в 1973 году. Языки логического программирования, в особенности Пролог, широко используются в системах искусственного интеллекта.

Центральным понятием в логическом программировании является отношение. Программа представляет собой совокупность определений отношений между объектами (в терминах условий или ограничений) и цели (запроса). Языки логического программирования характеризуются:

- 1) высоким уровнем;
- 2) строгой ориентацией на символьные вычисления;
- 3) возможностью инверсных вычислений, то есть переменные в процедурах не делятся на входные и выходные;
- 4) возможной логической неполнотой, поскольку зачастую невозможно выразить в программе определенные логические соотношения, а также невозможно получить из программы все выводы правильные.

3. Объектно-ориентированное программирование

Прототипом объектно-ориентированного программирования послужил ряд средств, входящих в состав языка SIMULA-67. Но в самостоятельный стиль оно оформилось с появлением языка SMALLTALK, разработанного А. Кеем в 1972 году и первоначально предназначенного для реализации функций машинной графики.

Основная цель ООП, как и большинства других подходов к программированию – повышение эффективности разработки программ. Идеи ООП оказались плодотворными и нашли применение не только в языках программирования, но и в других областях Computer Science, например, в области разработки операционных систем.

Концепция объектно-ориентированного программирования подразумевает, что основой управления процессом реализации программы является передача сообщений объектам. Поэтому объекты должны определяться совместно с сообщениями, на которые они должны реагировать при выполнении программы. В этом состоит главное отличие ООП от процедурного программирования. Таким образом, объектно-ориентированная программа состоит из объектов – отдельных фрагментов кода, обрабатывающего данные, которые взаимодействуют друг с другом через определённые интерфейсы.

Тема 2. Основные программные средства информационных технологий

Лекция 3 (2 часа)

Графические редакторы. Компьютерная графика и анимация: основные понятия и определения

Основные вопросы

1. Определение и основные задачи компьютерной графики. Области применения.
2. История развития компьютерной графики.
3. Виды компьютерной графики.

Цель. Дать понятие о видах компьютерной графики, познакомить с историей развития, областями применения и основными задачами, которые решаются с помощью компьютерной графики.

С философской и культурологической точки зрения компьютерная графика – это область деятельности, связанная с созданием и обработкой цифровых изображений.

Компьютерная графика – это область информатики, занимающаяся проблемами получения различных изображений (рисунков, чертежей, мультипликации) при помощи компьютера.

С практической точки зрения компьютерная графика - это процесс создания, обработки и вывода изображений разного рода с помощью компьютера.

Компьютерная графика - это вид искусства. По мнению художника Кудерского Максима Викторовича, члена Союза художников России, по творческим затратам, создание произведения искусства средствами компьютерной графики даже более трудоемкое дело, чем обычная работа живописца.

Основные задачи, которые решаются с помощью компьютерной графики: представление изображения в компьютерной графике (оцифровка некомпьютерной графики); визуализация компьютерных данных; осуществление действий с изображением (создание, редактирование и т.д.).

Области применения компьютерной графики: научная графика, деловая графика, конструкторская графика, иллюстративная графика, компьютерная анимация, графика для Интернета, компьютерные игры, системы виртуальной реальности и др.

Основные этапы истории развития компьютерной графики: 1940-1970гг. – время больших компьютеров (до персональных компьютеров). Графикой занимались только при выводе на принтер. В этот период заложены математические основы. 1971-1985гг. – появились персональные компьютеры, т.е. появился доступ пользователя к дисплею. Роль графики резко возросла, но наблюдалось очень низкое быстродействие компьютера. Появилось цветное изображение. 1986-1990гг. – появление технологии мультимедиа. К графике добавились обработка звука и видеоизображения, общение пользователя с компьютером расширилось. 1991 г. – появление графики нашего дня Virtual Reality. Появились датчики перемещения, благодаря которым компьютер меняет изображения при помощи сигналов посылаемых на него. Появление стереочков (монитор на каждый глаз), благодаря высокому быстродействию которых, производится имитация реального мира.

Классификация компьютерной графики:

- по динамике: статичная и динамичная;
- по размерности: двумерная и трехмерная;
- по типу представления графической информации, и следующими из него алгоритмами обработки изображений: растровая и векторная.

Достоинства растровой графики: простота и, как следствие, техническая реализуемость автоматизации ввода (оцифровки) изобразительной информации (сканеры, видеокамеры, цифровые фотокамеры, графические планшеты); фотореалистичность – хорошее растровое изображение выглядит реально и естественно; форматы файлов, предназначенные для сохранения пиксельных изображений, являются стандартными (файл, сохраняющий пиксельное

изображение, легко открывается и импортируется в редакторах пиксельной и векторной графики, а также в программах верстки и браузерах); растровое изображение наиболее адаптировано для распространенных растровых устройств вывода – лазерных принтеров и др.

Недостатки растровой графики: занимает большой объем памяти; редактирование больших растровых изображений, занимающих большие массивы памяти, требуют большие ресурсы компьютера; любая трансформация (поворот, масштабирование, наклон и т. д.) приводит к искажениям; при увеличении размеров изображения сильно ухудшается качество.

Применение растровой графики: обработка фотоизображений, художественная графика, реставрационные работы, работа со сканером.

Достоинства векторной графики: при изменении размеров изображения не изменяется размер файла (формулы, описывающие изображение, остаются те же, меняется только коэффициент пропорциональности; при изменении масштаба изображения оно не теряет своего качества; максимально использует возможности разрешающей способности любого устройства вывода: изображение всегда будет выглядеть настолько качественно, насколько его способно вывести данное устройство; может включать в себя и изображения пиксельной графики, которые становятся таким же объектом, как и все остальные (с особым статусом и со значительными ограничениями в обработке); развитые средства интеграции изображений и текста, единый подход к ним и, как следствие, возможность создания конечного продукта, который может быть без дополнительных операций отправлен на цветоделение и растривание.

Недостатки векторной графики: векторные изображения выглядят искусственно; ограничена в живописных средствах и не предназначена для создания фотореалистических изображений; сложность векторного принципа описания изображения не позволяет автоматизировать ввод графической информации (сконструировать устройство, подобное сканеру для пиксельной графики); программная зависимость: каждая программа сохраняет данные в своем собственном формате; меньше оттенков и полутонов чем в растровой графике.

Применение векторной графики: компьютерная полиграфия, системы компьютерного проектирования, компьютерный дизайн и реклама.

Выводы. Компьютерная графика используется практически во всех видах деятельности человека для представления изображений в цифровой форме; визуализации компьютерных данных, создания и редактирования изображений. По типу представления графической информации и алгоритмам обработки изображений компьютерная графика делится на растровую и векторную. Выбор

графического редактора и соответственно вида компьютерной графики зависит от дальнейшего использования создаваемого или редактируемого изображения.

Ключевые понятия: компьютерная графика, статичная компьютерная графика, динамичная компьютерная графика, двумерная компьютерная графика, трехмерная компьютерная графика, растровая компьютерная графика, векторная компьютерная графика.

Лекция 3 (4 часа)

Способы хранения и обработки графической информации

Основные вопросы

1. Программное обеспечение для работы с цифровыми изображениями: обоснование выбора.
2. Разрешение и размер изображения.
3. Цветовые модели, их применение.
4. Основные типы графических форматов.

Цель. Провести анализ программного обеспечения для работы с цифровыми изображениями, дать понятие о разрешении и размере изображения, познакомить с цветовыми модели и графическими форматами, которые используются в компьютерной графике.

Программные средства для создания и обработки графических изображений условно можно разделить на программы рисования, программы черчения, программы верстки страниц, программы редактирования изображений, программы спецэффектов, программы трехмерного моделирования и визуализации, программы мультимедиа.

Разрешение - это степень детализации изображения, число пикселей, отводимых на единицу площади. Имеет смысл говорить о разрешении изображения только применительно к какому-либо устройству ввода или вывода изображения (разрешение изображения, монитора, печатающего устройства).

Разрешение изображения - это свойство самого изображения. Задается программой, в которой создается изображение или с помощью сканера. Измеряется в точках на дюйм – (dpi)- число пикселей на единицу длины. Уровень качества изображения закладывается в процессе сканирования в зависимости от устанавливаемого разрешения. Более высокое разрешение позволяет передавать больше деталей и более точно репродуцировать оригинал. Таким образом, величина разрешения значительной степени определяет качество цифрового изображения.

Увеличение разрешения цифрового изображения в любом графическом редакторе не может улучшить качество изображения. Программа не может

добавить новую изобразительную информацию (добавить новые более мелкие детали), а может только перераспределить уже имеющиеся данные на большее число пикселей.

Разрешение экрана – это свойство компьютерной системы (зависит от монитора, видеокарты) и операционной системы (например, зависит от настроек Windows) (число точек (пикселей) на единицу длины, обычно дюйм (pixels per inch))

Разрешение принтера - это свойство принтера, выражающее количество отдельных точек, которые могут быть напечатаны на участке единичной длины (dpi (сокр. dpi - dots per inch - точки на дюйм)). Определяет размер изображения при заданном качестве.

Значение разрешения изображения хранится в файле изображения и неразрывно связано с его физическим размером. Физический размер изображения измеряется как в пикселях, так и в единицах длины (миллиметрах, сантиметрах, дюймах). Он задается при создании изображения и хранится вместе с файлом.

Пересчитать размер изображения из пикселей в единицы длины и наоборот при известном разрешении изображения можно по следующей формуле: размер иллюстрации в дюймах равен размеру в пикселях разделенному на разрешение изображения.

Цвет является визуальным ощущением, которое появляется у наблюдателя в результате взаимодействия света и объекта.

Свет - это видимая часть электромагнитного спектра. Область электромагнитного спектра, видимая человеческим глазом, занимает диапазон примерно от 380 до 770 нанометров.

Цветовая модель – это способ описания цвета с помощью количественных характеристик. Основные цветовые модели: RGB, CMYK, HSB, Lab и другие.

Цифровые цветовые модели по принципу действия делятся на аддитивные, основанные на сложении цветов; субтрактивные, основу которых составляет операция вычитания цветов (субтрактивный синтез); перцепционные, базирующиеся на восприятии.

Базовыми цветами называют цвета, с помощью которых можно получить практически весь спектр видимых цветов.

Цветовой режим – это способ реализации определенной цветовой модели в рамках конкретной графической программы.

Цветовая модель RGB является аппаратно–зависимой, так как значения базовых цветов (а также точка белого) определяются качеством примененного в мониторе люминофора. В результате на разных мониторах одно и то же изображение выглядит неодинаково. Достоинством цветовой модели RGB

является возможность работы со всеми 16 миллионами цветами. Недостаток – это то, что при выводе изображения на печать часть из этих цветов теряется, в основном самые яркие и насыщенные.

В графических редакторах режим CMYK предназначается для подготовки изображения к печати в типографии, и в нем за качество цвета отвечают реальные красители, поэтому цветовой охват в режиме CMYK несколько меньше, чем в режиме RGB. Цвета модели CMYK являются полиграфической триадой и могут быть легко воспроизведены полиграфическими машинами. При цветоделении (переходе в субтрактивную модель CMYK) некоторая часть "серого" цвета, образованного тремя компонентами красок Cyan, Magenta, Yellow, частично заменяется черной краской. Количество черной краски, генерируемой при цветоделении, может быть различно, и зависит от пользовательских настроек среды цветоделения.

Все цветовые модели являются математическими и легко конвертируются одна в другую по простым формулам. Конверторы встроены во все графические программы.

Главная трудность при переходе из системы RGB в CMYK заключается в том, что на бумаге (в системе CMYK) не могут быть представлены некоторые цвета, которые с легкостью можно представить на экране. Если на экране легко можно сделать любой оттенок цвета, то в смешивании красителей такой точности добиться достаточно сложно. Поэтому часто изображение на экране сильно отличается от изображения на бумаге, где те же краски выглядят блекло и некрасиво.

Условно все графические форматы могут быть классифицированы по следующим категориям:

- по виду хранимых данных (растровая, векторная и смешанная формы)
- по допустимому объему данных, параметрам изображения (прозрачность, слой, альфа-каналы)
- хранению палитры или цветовые модели
- методике сжатия данных
- по способам организации файла (текстовый, двоичный)
- структуре файла (с последовательной или ссылочной (индексно-последовательной) структурой).

Примеры растровых форматов: bmp, psd, gif, jpeg, png, tiff и другие. Форматы для интернет-графики - gif, jpeg, png.

Примеры векторных форматов: cdr, ai и другие.

Преимущества форматов растровых файлов. Растровые файлы могут быть легко созданы из существующих пиксельных данных, записанных в памяти в виде массива. Воспроизведение пиксельных данных, сохраненных в растровом файле, может осуществляться в некой системе координат, позволяющей

представить эти данные в виде сетки. Пиксельные значения могут изменяться индивидуально либо большими группами с помощью палитры. Растровые файлы легко преобразовываются для передачи на точечные устройства вывода, такие как дисплеи на базе электронно-лучевой трубки и принтеры.

Недостатки форматов растровых файлов. Имеют очень большой размер, особенно если изображение многоцветное и большое. Применение различных схем сжатия уменьшает размер пиксельных данных, но необходимость распаковки перед использованием значительно замедляет процесс чтения и визуализации изображения. При этом, чем сложнее растровое изображение (большое количество цветов и мелких деталей), тем меньше эффективность процесса сжатия. Растровые форматы плохо поддаются масштабированию. Сжатие изображения децимацией (отбрасыванием каждого десятого пикселя) может изменить его неприемлемым образом, равно как и увеличение изображения копированием пикселей. Поэтому растровые файлы при печати обычно не масштабируются.

Преимущества форматов векторных файлов. Векторные файлы удобны для хранения изображений, состоящих из элементов, которые представлены линиями (например, из окружностей и многоугольников) или могут быть разложены на простейшие геометрические объекты. Более сложные форматы позволяют хранить трехмерные объекты, такие как многогранники и каркасные модели. Векторные данные легко масштабируются и поддаются другим манипуляциям, позволяющим адаптировать их к разрешающей способности различных устройств вывода. Многие векторные файлы, содержащие данные только в формате ASCII, могут быть модифицированы с помощью простых средств редактирования текстов. Отдельные элементы могут быть добавлены, удалены или изменены без ущерба для других объектов изображения. Векторные данные можно визуализировать, а затем сохранить их в этом же или преобразовать в другой векторный формат.

Недостатки форматов векторных файлов. Векторные файлы трудно применять для хранения сложных изображений, в частности некоторых фотографий, цветовая информация которых является очень важной и может изменяться буквально на пиксельном уровне. Внешнее представление векторных изображений может изменяться в зависимости от отображающей их программы (совместимость программы визуализации с программой, создавшей изображение, сложность набора геометрических примитивов и операций рисования). Векторные данные лучше отображаются на векторных устройствах вывода, таких как плоттеры и дисплеи с произвольным сканированием. Векторную графику можно эффективно отобразить только на растровых дисплеях с высокой разрешающей способностью. Визуализация векторных данных может потребовать значительно больше времени, чем визуализация

растрового файла равной сложности, поскольку каждый элемент изображения должен быть воспроизведен отдельно и в определенной последовательности.

Выводы. Программные средства для создания и обработки графических изображений используются в зависимости от решения поставленных перед пользователем задач. Качество конечного цифрового изображения зависит от его разрешения, которое задается в графическом редакторе, где оно создается или в сканере. Увеличение разрешения цифрового изображения в любом графическом редакторе не может улучшить качество изображения. Выбор способа описания цвета с помощью количественных характеристик зависит от области применения конечного изображения. При использовании графического формата для сохранения данных необходимо учитывать базовые характеристики каждого из них.

Ключевые понятия: разрешение изображения, разрешение монитора, разрешение принтера, цветовая модель, базовые цвета, цветовой режим, растровые графические форматы, векторные графические форматы.

Лекция 4 (2 часа)

Способы представления и обработки звуковой и видеоинформации

Основные вопросы

1. Оцифровка звуковой информации. Определения и характеристики.
2. Основные звуковые форматы.
3. Создание видео и анимации. Основные характеристики.

Цель. Познакомить с основными способами представления и обработки звуковой и видеоинформации на компьютере.

Звук – это звуковая волна с непрерывно меняющейся амплитудой и частотой. Звук характеризуется интенсивностью, которую человек воспринимает как громкость, и частотой, воспринимаемой как тон. Чем больше частота сигнала, тем выше тон звука. Чем больше амплитуда, тем громче звук.

Диапазон частот звука, воспринимаемых человеком, составляет от 20 Гц до 20000 Гц.

При аналоговом представлении физическая величина принимает бесконечное множество значений, причём её значения изменяются непрерывно. При дискретном представлении физическая величина принимает конечное множество значений, причём её значения изменяются скачкообразно.

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

Цифровой сигнал — это всегда некоторое приближенное и упрощенное представление аналогового. Звук разбивается на составляющие, каждой из которых присваивается числовой код - происходит оцифровка звука.

Чем чаще измеряется уровень на входе (то есть чем чаще идут вертикальные линии на рисунке), тем точнее цифровой сигнал воспроизводит форму аналогового. Этот параметр и есть частота дискретизации или частота семплирования.

Дискретизацией звукового сигнала занимаются звуковые карты наших компьютеров - их аналого-цифровые преобразователи. Звуковая карта (чаще ее называют Sound Blaster) представляет собой небольшую плату с набором микросхем со специальными разъемами для подключения микрофона, динамиков, клавиатуры и других подобных устройств. Карты Sound Blaster бывают различных типов и предоставляют широчайший спектр возможностей работы со звуком, от записи с микрофона до сложнейшим конструированием современных мелодий для большого оркестра. Аналого-цифровые преобразователи через определенные интервалы времени измеряет уровень сигнала на входе и записывает полученное число на диск. Последовательность этих чисел и составляет звуковой файл.

Формат WAVE (WAV). Стандартный формат файлов для хранения звука в системе Windows. Является специальным типом формата RIFF (Resource Interchange File Format). Файл RIFF составлен из блоков, некоторые из которых могут, в свою очередь, содержать другие вложенные блоки; перед каждым блоком данных помещается четырехсимвольный идентификатор и длина. Звуковые файлы WAV, как правило, более просты и имеют только один блок формата и один блок данных. В первом содержится общая информация об оцифрованном звуке (число каналов, частота дискретизации, характер зависимости громкости и т.д.), а во втором — сами числовые данные. Такая структурированность позволяет программному обеспечению оптимизировать процесс передачи данных при воспроизведении, но выигрыш во времени приводит к существенному увеличению размера файла. Таким образом, файлы в этом формате имеют большой размер, который зависит от: дискретизации (частоты семплирования); разрядности звука; моно или стереозвук; длительности.

Формат MP3 (MPEG Layer3). Это одним из самых распространённых и популярных форматов цифрового кодирования звуковой информации с потерями. Принцип сжатия заключается в снижении точности некоторых частей звукового потока, что практически неразлично для слуха на повсеместно распространённой аппаратуре низкой точности воспроизведения звука, однако в большинстве случаев чётко различимы на специальной аппаратуре высокой точности воспроизведения. Данный метод называют перцепционным кодированием. При этом на первом этапе строится диаграмма звука в виде последовательности коротких промежутков времени, затем на ней удаляется информация, не различимая человеческим ухом, а оставшаяся

информация сохраняется в компактном виде. Данный подход похож на метод сжатия, используемый при сжатии картинок в формат JPEG. Успехи технологии MP3 привели к тому, что ее применяют сейчас и во многих бытовых звуковых устройствах, например, плеерах и сотовых телефонах.

Формат MIDI. Название MIDI есть сокращение от Musical Instrument Digital Interface, т.е. цифровой интерфейс для музыкальных инструментов. MIDI базируется на пакетах данных, каждый из которых соответствует некоторому событию, в частности, нажатию клавиши или установке режима звучания. Интерфейс позволяет единообразно кодировать в цифровой форме такие данные как нажатие клавиш, настройку громкости и других акустических параметров, выбор тембра, темпа, тональности и др., с точной привязкой во времени. Воспроизводящее устройство или программа называется синтезатором MIDI и фактически является автоматическим музыкальным инструментом. Главным преимуществом файлов MIDI является их очень небольшой размер, поскольку это не детальная запись звука, а фактически некоторый расширенный электронный эквивалент традиционной нотной записи.

Для создания и редактирования цифровых звукозаписей применяются звуковые редакторы, например Adobe Audition, Sound Forge и др.

Под обработкой видеоматериалов понимается процесс оцифровки, то есть кодирования видеoinформации. Любую видеoinформацию можно дифференцировать, то есть разложить на две ключевые составляющие: звуковую и графическую. Следовательно, операция кодирования видеoinформации будет заключаться в сочетании операций кодирования звуковой информации и кодирования графической информации.

После проведения операции цифрования звука и изображений на выходе получается бинарный, двоичный код, который будет понятен процессору персонального компьютера. Именно в формате двоичного кода наша видеoinформация и будет храниться на электронных носителях. Если мы захотим проиграть видеоконтент на нашем персональном компьютере или другом устройстве, то нам придется провести операцию восстановления информации, то есть осуществить преобразование информации, записанной в двоичном коде в формат понятный человеку.

Видео – технология записи, обработки, передачи, хранения и воспроизведения визуального или аудиовизуального материала.

Основные характеристики видео:

– Битрейт или ширина видеопотока (для цифрового видео) – это количество обрабатываемых бит видеoinформации за секунду времени (измеряется «бит/с» – бит в секунду или чаще «Мбит/с» – мегабит в секунду). Чем шире видеопоток, тем лучше качество видео.

– Глубина цвета – количество бит, приходящихся на кодирование цвета в одном пикселе. Для цветовой модели RGB обычно характерны следующие режимы глубины цвета: 24 бит/пиксель, 48 бит/пиксель.

– Разрешение – ширина и высота кадра в пикселях. Стандартное разрешение (SD, Standard Definition) – формат DVD с разрешением 720x576 (PAL), 720x480 (NTSC). Высокое разрешение (HD, High Definition) – HD720 (1280x720 точек) и HD1080 (1920x1080 точек).

– Частота кадра – это число неподвижных изображений, сменяющих друг друга за 1 секунду показа видео или анимационного материала и создающих эффект движения объектов на экране. Чем больше частота кадра, тем более плавным и естественным будет казаться движение (24 кадра в секунду – скорость записи и воспроизведения кинофильмов, 25 и 30 кадров в секунду в телевизионных стандартах PAL/SECAM; и в NTSC; 4–15 кадров в секунду – для анимационного GIF или SWF, например, баннера на web-странице).

Компрессия - процесс удаления избыточной информации для уменьшения занимаемого файлами места. Компрессия без потери качества позволяет восстановить оригинальные (первоначальные) данные, компрессия с потерей качества удаляет часть полезной информации для наилучшего сжатия размеров файлов и не позволяет восстановить оригинальные данные.

MPEG - это аббревиатура от Moving Picture Experts Group. Эта экспертная группа работает под совместным руководством двух организаций - ISO (Организация по международным стандартам) и IEC (Международная электротехническая комиссия). Она разрабатывает единые нормы кодирования аудио- и видеосигналов. Стандарты MPEG используются в технологиях CD-i и CD-Video, являются частью стандарта DVD, активно применяются в цифровом радиовещании, в кабельном и спутниковом ТВ, Интернет-радио, мультимедийных компьютерных продуктах, в коммуникациях по каналам ISDN и многих других электронных информационных системах.

Компрессия MJPEG [Motion JPEG] основывается на независимом кодировании каждого кадра и объединении полученной последовательности в файл. Сжатие видео осуществляется по JPEG-алгоритму: каждое изображение разбивается на квадраты 8x8 точек и представляется в векторной форме путем дискретного преобразования и высокочастотной фильтрации полученного спектра. По сути, сжатое видео представляет собой последовательность независимых JPEG-изображений. Поскольку каждый кадр кодируется отдельно от других, возможно последующее покадровое редактирование изображения. Существенным преимуществом этого алгоритма сжатия видео является его симметричность, то есть для кодирования и декодирования необходимы одни и те же вычислительные затраты.

Формат AVI изначально был предназначен для обмена мультимедийными данными, был Microsoft совместно с IBM. Данный формат является наиболее распространенной формой представления видео на персональных компьютерах. В зависимости от формы представления видеоданных файлы AVI бывают различных стандартов.

Real Video - самый распространенный сетевой потоковый аудио/видео стандарт, предложенный фирмой RealOne, широко используемый для представления в Интернете и кабельных сетях музыки и видео. Стандарт завоевал популярность как средство размещения в Интернете аудио/видео контента с малым размером файлов и вполне приличным качеством. Небольшой размер выходного файла позволяет существенно экономить трафик при трансляции по каналам связи, в том числе и в GSM-сетях. Файлы RealVideo обычно имеют расширение ".rm", ".ram" или ".rmvb". Для воспроизведения на телефоне файл должен иметь геометрию изображения не больше 208x176 точек и содержать общий поток данных не выше 340 кБит/с.

Программы, необходимые для воспроизведения видео: проигрыватели мультимедиа Windows Media (стандартное средство Microsoft; достоинство - наличие большого количества кодеков; недостатки: требует значительных системных ресурсов и долго загружается), Winamp (универсальный проигрыватель распознает практически все аудио- и видеоформаты, поддерживает списки воспроизведения, имеет встроенный эквалайзер, с помощью которого можно точно отрегулировать качество звучания, и даже минибраузер; в программе предусмотрена возможность без прерывания процесса проигрывания пополнять список воспроизведения новыми файлами, удалять и менять их местами) и другие.

Кодек - сокращение от "compression/decompression" (компрессия/декомпрессия). Кодеком называют алгоритм или специальную компьютерную программу позволяющую обработать (применить компрессию) и уменьшить размеры большого файла. Файл закодированный (компрессия) каким-либо кодеком, требует тот же кодек для декомпрессии.

Создание и обработка видео и анимации

GIF-анимация - растровая компьютерная анимация, сохраненная в формате GIF, который поддерживает возможность использования режима индексированных цветов (не более 256), режим постепенного проявления изображения (interleaved), прозрачность и использует алгоритм сжатия без потерь качества LZW. Средства: Adobe Photoshop, GIF Animator и др.

FLASH-анимация - векторная компьютерная анимация, способная объединить в одном формате текст, графику, звук, анимацию, интерактивные компоненты. Средства: Adobe Flash.

Создание и обработка видео. Средства: Adobe Premiere, Adobe After Effects и др.

Ключевые понятия: звук, дискретизация, битрейт, кодек, компрессия.

Тема 3. Сетевые технологии и Интернет

Лекция 5 (6 часов)

Семиуровневая модель структуры протоколов связи

Основные вопросы

1. Структура базовой модели сети.
2. Прикладной уровень.
3. Представительский уровень.
4. Сеансовый уровень.
5. Транспортный уровень.
6. Сетевой уровень.
7. Канальный уровень.
8. Физический уровень.

Цель. Изучение семиуровневой модели структуры протоколов связи.

Для единого представления данных в сетях с неоднородными устройствами и программным обеспечением международная организация по стандартам ISO (International Standardization Organization) разработала базовую модель связи открытых систем OSI (Open System Interconnection). Эта модель описывает правила и процедуры передачи данных в различных сетевых средах при организации сеанса связи. Основными элементами модели являются уровни, прикладные процессы и физические средства соединения.

Каждый уровень модели OSI выполняет определенную задачу в процессе передачи данных по сети. Базовая модель является основой для разработки сетевых протоколов. OSI разделяет коммуникационные функции в сети на семь уровней, каждый из которых обслуживает различные части процесса области взаимодействия открытых систем.

Модель OSI описывает только системные средства взаимодействия, не касаясь приложений конечных пользователей. Приложения реализуют свои собственные протоколы взаимодействия, обращаясь к системным средствам.



Модель OSI

Если приложение может взять на себя функции некоторых верхних уровней модели OSI, то для обмена данными оно обращается напрямую к системным средствам, выполняющим функции оставшихся нижних уровней модели OSI.

Взаимодействие уровней модели OSI

Модель OSI можно разделить на две различных модели:

- горизонтальную модель на базе протоколов, обеспечивающую механизм взаимодействия программ и процессов на различных машинах;
- вертикальную модель на основе услуг, обеспечиваемых соседними уровнями друг другу на одной машине.

Каждый уровень компьютера-отправителя взаимодействует с таким же уровнем компьютера-получателя, как будто он связан напрямую. Такая связь называется логической или виртуальной связью. В действительности взаимодействие осуществляется между смежными уровнями одного компьютера.

Итак, информация на компьютере-отправителе должна пройти через все уровни. Затем она передается по физической среде до компьютера-получателя и опять проходит сквозь все слои, пока не доходит до того же уровня, с которого она была послана на компьютере-отправителе.

В горизонтальной модели двум программам требуется общий протокол для обмена данными. В вертикальной модели соседние уровни обмениваются данными с использованием интерфейсов прикладных программ API (Application Programming Interface).

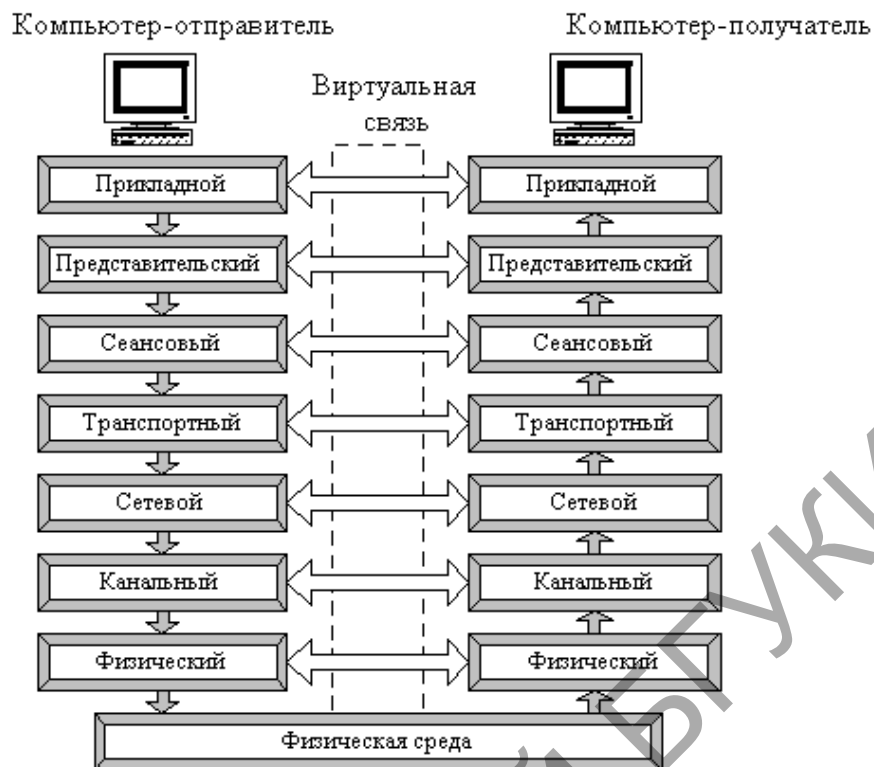
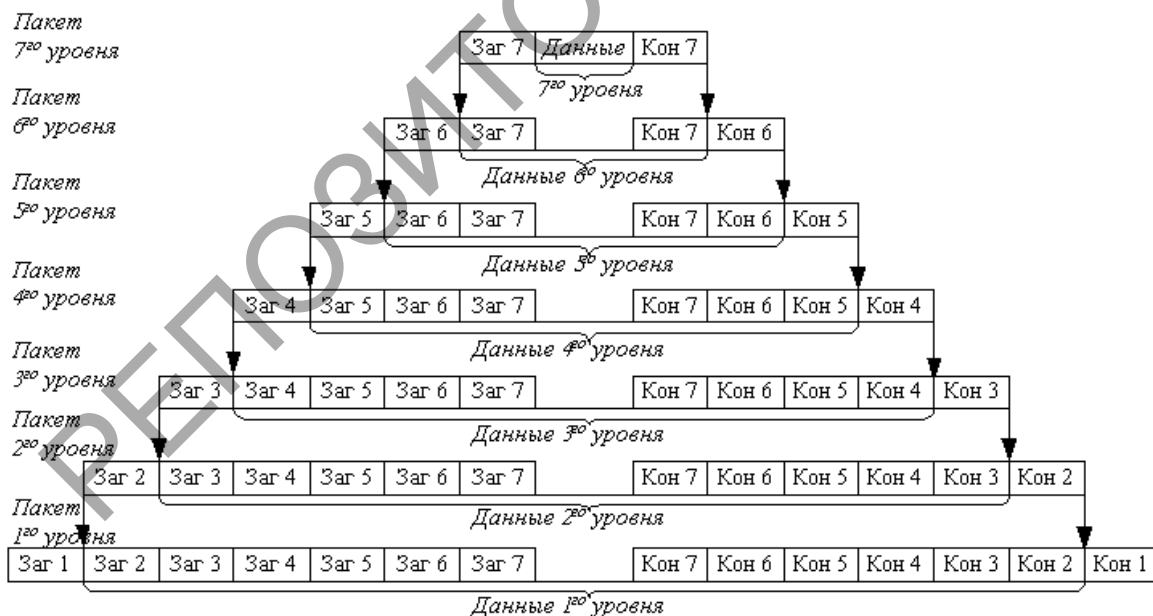


Схема взаимодействия компьютеров в базовой эталонной модели OSI

Перед подачей в сеть данные разбиваются на пакеты. Пакет (packet) – это единица информации, передаваемая между станциями сети.



Формирование пакета каждого уровня семиуровневой модели

При отправке данных пакет проходит последовательно через все уровни программного обеспечения. На каждом уровне к пакету добавляется управляющая информация данного уровня (заголовок), которая необходима для

успешной передачи данных по сети, как это показано на рисунке, где Заг – заголовок пакета, Кон – конец пакета.

На принимающей стороне пакет проходит через все уровни в обратном порядке. На каждом уровне протокол этого уровня читает информацию пакета, затем удаляет информацию, добавленную к пакету на этом же уровне отправляющей стороной, и передает пакет следующему уровню. Когда пакет дойдет до Прикладного уровня, вся управляющая информация будет удалена из пакета, и данные примут свой первоначальный вид.

Каждый уровень модели выполняет свою функцию. Чем выше уровень, тем более сложную задачу он решает.

Отдельные уровни модели OSI удобно рассматривать как группы программ, предназначенных для выполнения конкретных функций. Один уровень, к примеру, отвечает за обеспечение преобразования данных из ASCII в EBCDIC и содержит программы, необходимые для выполнения этой задачи.

Каждый уровень обеспечивает сервис для вышестоящего уровня, запрашивая в свою очередь сервис у нижестоящего уровня. Верхние уровни запрашивают сервис почти одинаково: как правило, это требование маршрутизации каких-то данных из одной сети в другую. Практическая реализация принципов адресации данных возложена на нижние уровни. На рисунке приведено краткое описание функций всех уровней.

7. Прикладной уровень представляет набор интерфейсов, позволяющий получить доступ к сетевым службам
6. Уровень представления преобразует данные в общий формат для передачи по сети
5. Сеансовый уровень поддерживает взаимодействие (сеанс) между удаленными процессами
4. Транспортный уровень управляет передачей данных по сети, обеспечивает подтверждение передачи
3. Сетевой уровень маршрутизация, управление потоками данных, адресация сообщений для доставки; преобразование логические сетевые адреса и имена в соответствующие им физические
2. Канальный уровень 2.1. Контроль логической связи (LLC): формирование кадров 2.2. Контроль доступа к среде (MAC): управление доступом к среде
1. Физический уровень обеспечивает битовые протоколы передачи информации

Функции уровней модели OSI

Рассматриваемая модель определяет взаимодействие открытых систем разных производителей в одной сети. Поэтому она выполняет для них координирующие действия по:

- взаимодействию прикладных процессов;
- формам представления данных;
- единообразному хранению данных;
- управлению сетевыми ресурсами;
- безопасности данных и защите информации;
- диагностике программ и технических средств.

Прикладной уровень

Прикладной уровень обеспечивает прикладным процессам средства доступа к области взаимодействия, является верхним (седьмым) уровнем и непосредственно примыкает к прикладным процессам.

В действительности прикладной уровень — это набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например с помощью протокола электронной почты. Специальные элементы прикладного сервиса обеспечивают сервис для конкретных прикладных программ, таких как программы пересылки файлов и эмуляции терминалов. Если, например программе необходимо переслать файлы, то обязательно будет использован протокол передачи, доступа и управления файлами FTAM (File Transfer, Access, and Management). В модели OSI прикладная программа, которой нужно выполнить конкретную задачу (например, обновить базу данных на компьютере), посылает конкретные данные в виде Дейтаграммы на прикладной уровень. Одна из основных задач этого уровня — определить, как следует обрабатывать запрос прикладной программы, другими словами, какой вид должен принять данный запрос.

Единица данных, которой оперирует прикладной уровень, обычно называется сообщением (message).

Прикладной уровень выполняет следующие функции:

1. Выполнение различных видов работ: передача файлов, управление заданиями, управление системой и т. д;
2. Идентификация пользователей по их паролям, адресам, электронным подписям;
3. Определение функционирующих абонентов и возможности доступа к новым прикладным процессам;

4. Определение достаточности имеющихся ресурсов;
5. Организация запросов на соединение с другими прикладными процессами;
6. Передача заявок представительскому уровню на необходимые методы описания информации;
7. Выбор процедур планируемого диалога процессов;
8. Управление данными, которыми обмениваются прикладные процессы и синхронизация взаимодействия прикладных процессов;
9. Определение качества обслуживания (время доставки блоков данных, допустимой частоты ошибок);
10. Соглашение об исправлении ошибок и определении достоверности данных;
11. Согласование ограничений, накладываемых на синтаксис (наборы символов, структура данных).

Указанные функции определяют виды сервиса, которые прикладной уровень предоставляет прикладным процессам. Кроме этого, прикладной уровень передает прикладным процессам сервис, предоставляемый физическим, канальным, сетевым, транспортным, сеансовым и представительским уровнями.

На прикладном уровне необходимо предоставить в распоряжение пользователей уже переработанную информацию. С этим может справиться системное и пользовательское программное обеспечение.

Прикладной уровень отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и управление сетью.

К числу наиболее распространенных протоколов верхних трех уровней относятся:

- FTP (File Transfer Protocol) протокол передачи файлов;
- TFTP (Trivial File Transfer Protocol) простейший протокол пересылки файлов;
- X.400 электронная почта;
- Telnet работа с удаленным терминалом;
- SMTP (Simple Mail Transfer Protocol) простой протокол почтового обмена;
- CMIP (Common Management Information Protocol) общий протокол управления информацией;
- SLIP (Serial Line IP) IP для последовательных линий. Протокол последовательной посимвольной передачи данных;
- SNMP (Simple Network Management Protocol) простой протокол сетевого управления;

- FTAM (File Transfer, Access, and Management) протокол передачи, доступа и управления файлами.

Уровень представления данных

Функции данного уровня – представление данных, передаваемых между прикладными процессами, в нужной форме.

Этот уровень обеспечивает то, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе. В случаях необходимости уровень представления в момент передачи информации выполняет преобразование форматов данных в некоторый общий формат представления, а в момент приема, соответственно, выполняет обратное преобразование. Таким образом, прикладные уровни могут преодолеть, например, синтаксические различия в представлении данных. Такая ситуация может возникнуть в ЛВС с неоднотипными компьютерами (IBM PC и Macintosh), которым необходимо обмениваться данными. Так, в полях баз данных информация должна быть представлена в виде букв и цифр, а зачастую и в виде графического изображения. Обрабатывать же эти данные нужно, например, как числа с плавающей запятой.

В основу общего представления данных положена единая для всех уровней модели система ASN.1. Эта система служит для описания структуры файлов, а также позволяет решить проблему шифрования данных. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных сервисов. Примером такого протокола является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP. Этот уровень обеспечивает преобразование данных (кодирование, компрессия и т.п.) прикладного уровня в поток информации для транспортного уровня.

Представительный уровень выполняет следующие основные функции:

1. Генерация запросов на установление сеансов взаимодействия прикладных процессов.
2. Согласование представления данных между прикладными процессами.
3. Реализация форм представления данных.
4. Представление графического материала (чертежей, рисунков, схем).
5. Засекречивание данных.
6. Передача запросов на прекращение сеансов.

Протоколы уровня представления данных обычно являются составной частью протоколов трех верхних уровней модели.

Сеансовый уровень

Сеансовый уровень – это уровень, определяющий процедуру проведения сеансов между пользователями или прикладными процессами.

Сеансовый уровень обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, вместо того чтобы начинать все сначала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется.

Сеансовый уровень управляет передачей информации между прикладными процессами, координирует прием, передачу и выдачу одного сеанса связи. Кроме того, сеансовый уровень содержит дополнительно функции управления паролями, управления диалогом, синхронизации и отмены связи в сеансе передачи после сбоя вследствие ошибок в нижерасположенных уровнях. Функции этого уровня состоят в координации связи между двумя прикладными программами, работающими на разных рабочих станциях. Это происходит в виде хорошо структурированного диалога. В число этих функций входит создание сеанса, управление передачей и приемом пакетов сообщений во время сеанса и завершение сеанса.

На сеансовом уровне определяется, какой будет передача между двумя прикладными процессами:

- полудуплексной (процессы будут передавать и принимать данные по очереди);
- дуплексной (процессы будут передавать данные, и принимать их одновременно).

В полудуплексном режиме сеансовый уровень выдает тому процессу, который начинает передачу, маркер данных. Когда второму процессу приходит время отвечать, маркер данных передается ему. Сеансовый уровень разрешает передачу только той стороне, которая обладает маркером данных.

Сеансовый уровень обеспечивает выполнение следующих функций:

1. Установление и завершение на сеансовом уровне соединения между взаимодействующими системами.
2. Выполнение нормального и срочного обмена данными между прикладными процессами.
3. Управление взаимодействием прикладных процессов.
4. Синхронизация сеансовых соединений.
5. Извещение прикладных процессов об исключительных ситуациях.
6. Установление в прикладном процессе меток, позволяющих после отказа либо ошибки восстановить его выполнение от ближайшей метки.

7. Прерывание в нужных случаях прикладного процесса и его корректное возобновление.

8. Прекращение сеанса без потери данных.

9. Передача особых сообщений о ходе проведения сеанса.

Сеансовый уровень отвечает за организацию сеансов обмена данными между оконечными машинами. Протоколы сеансового уровня обычно являются составной частью протоколов трех верхних уровней модели.

Транспортный уровень

Транспортный уровень предназначен для передачи пакетов через коммуникационную сеть. На транспортном уровне пакеты разбиваются на блоки.

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням модели (прикладному и сеансовому) передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Транспортный уровень определяет адресацию физических устройств (систем, их частей) в сети. Этот уровень гарантирует доставку блоков информации адресатам и управляет этой доставкой. Его главной задачей является обеспечение эффективных, удобных и надежных форм передачи информации между системами. Когда в процессе обработки находится более одного пакета, транспортный уровень контролирует очередность прохождения пакетов. Если проходит дубликат принятого ранее сообщения, то данный уровень опознает это и игнорирует сообщение.

В функции транспортного уровня входят:

1. Управление передачей по сети и обеспечение целостности блоков данных.

2. Обнаружение ошибок, частичная их ликвидация и сообщение о неисправленных ошибках.

3. Восстановление передачи после отказов и неисправностей.

4. Укрупнение или разделение блоков данных.

5. Предоставление приоритетов при передаче блоков (нормальная или срочная).

6. Подтверждение передачи.

7. Ликвидация блоков при тупиковых ситуациях в сети.

Начиная с транспортного уровня, все вышележащие протоколы реализуются программными средствами, обычно включаемыми в состав сетевой операционной системы.

Наиболее распространенные протоколы транспортного уровня включают в себя:

– TCP (Transmission Control Protocol) протокол управления передачей стека TCP/IP;

– UDP (User Datagram Protocol) пользовательский протокол дейтаграмм стека TCP/IP;

– NCP (NetWare Core Protocol) базовый протокол сетей NetWare;

– SPX (Sequenced Packet eXchange) упорядоченный обмен пакетами стека Novell;

– TP4 (Transmission Protocol) – протокол передачи класса 4.

Сетевой уровень

Сетевой уровень обеспечивает прокладку каналов, соединяющих абонентские и административные системы через коммуникационную сеть, выбор маршрута наиболее быстрого и надежного пути.

Сетевой уровень устанавливает связь в вычислительной сети между двумя системами и обеспечивает прокладку виртуальных каналов между ними. Виртуальный или логический канал – это такое функционирование компонентов сети, которое создает взаимодействующим компонентам иллюзию прокладки между ними нужного тракта. Кроме этого, сетевой уровень сообщает транспортному уровню о появляющихся ошибках. Сообщения сетевого уровня принято называть пакетами (packet). В них помещаются фрагменты данных. Сетевой уровень отвечает за их адресацию и доставку.

Прокладка наилучшего пути для передачи данных называется маршрутизацией, и ее решение является главной задачей сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних

показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например, надежности передачи.

Протокол канального уровня обеспечивает доставку данных между любыми узлами только в сети с соответствующей типовой топологией. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами.

Таким образом, внутри сети доставка данных регулируется канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень. При организации доставки пакетов на сетевом уровне используется понятие номер сети. В этом случае адрес получателя состоит из номера сети и номера компьютера в этой сети.

Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. Маршрутизатор – это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Для того чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач (hops) между сетями, каждый раз, выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, по которым проходит пакет.

Сетевой уровень отвечает за деление пользователей на группы и маршрутизацию пакетов на основе преобразования MAC-адресов в сетевые адреса. Сетевой уровень обеспечивает также прозрачную передачу пакетов на транспортный уровень.

Сетевой уровень выполняет функции:

1. Создание сетевых соединений и идентификация их портов.
2. Обнаружение и исправление ошибок, возникающих при передаче через коммуникационную сеть.
3. Управление потоками пакетов.
4. Организация (упорядочение) последовательностей пакетов.
5. Маршрутизация и коммутация.
6. Сегментирование и объединение пакетов.

На сетевом уровне определяется два вида протоколов. Первый вид относится к определению правил передачи пакетов с данными конечных узлов от узла к маршрутизатору и между маршрутизаторами. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых

протоколами обмена маршрутной информацией. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений.

Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

Наиболее часто на сетевом уровне используются протоколы:

– IP (Internet Protocol) протокол Internet, сетевой протокол стека TCP/IP, который предоставляет адресную и маршрутную информацию;

– IPX (Internetwork Packet Exchange) протокол меж сетевого обмена пакетами, предназначенный для адресации и маршрутизации пакетов в сетях Novell;

– X.25 международный стандарт для глобальных коммуникаций с коммутацией пакетов (частично этот протокол реализован на уровне 2);

– CLNP (Connection Less Network Protocol) сетевой протокол без организации соединений.

Канальный уровень

Единицей информации канального уровня являются кадры (frame). Кадры – это логически организованная структура, в которую можно помещать данные. Задача канального уровня – передавать кадры от сетевого уровня к физическому уровню.

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок.

Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит, в начало и конец каждого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка.

Задача канального уровня – брать пакеты, поступающие с сетевого уровня и готовить их к передаче, укладывая в кадр соответствующего размера. Этот уровень обязан определить, где начинается и где заканчивается блок, а также обнаруживать ошибки передачи.

На этом же уровне определяются правила использования физического уровня узлами сети. Электрическое представление данных в ЛВС (биты данных, методы кодирования данных и маркеры) распознаются на этом и только на этом уровне. Здесь обнаруживаются и исправляются (путем требований повторной передачи данных) ошибки.

Канальный уровень обеспечивает создание, передачу и прием кадров данных. Этот уровень обслуживает запросы сетевого уровня и использует сервис физического уровня для приема и передачи пакетов. Спецификации IEEE 802.X делят канальный уровень на два подуровня:

– LLC (Logical Link Control) управление логическим каналом осуществляет логический контроль связи. Подуровень LLC обеспечивает обслуживание сетевого уровня и связан с передачей и приемом пользовательских сообщений.

– MAC (Media Access Control) контроль доступа к среде. Подуровень MAC регулирует доступ к разделяемой физической среде (передача маркера или обнаружение коллизий или столкновений) и управляет доступом к каналу связи. Подуровень LLC находится выше подуровня MAC.

Канальный уровень определяет доступ к среде и управление передачей посредством процедуры передачи данных по каналу.

При больших размерах передаваемых блоков данных канальный уровень делит их на кадры и передает кадры в виде последовательностей.

При получении кадров уровень формирует из них переданные блоки данных. Размер блока данных зависит от способа передачи, качества канала, по которому он передается.

В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Канальный уровень может выполнять следующие виды функций:

1. Организация (установление, управление, расторжение) канальных соединений и идентификация их портов.
2. Организация и передача кадров.
3. Обнаружение и исправление ошибок.
4. Управление потоками данных.
5. Обеспечение прозрачности логических каналов (передачи по ним данных, закодированных любым способом).

Наиболее часто используемые протоколы на канальном уровне включают:

– HDLC (High Level Data Link Control) протокол управления каналом передачи данных высокого уровня, для последовательных соединений;

- IEEE 802.2 LLC (тип I и тип II) обеспечивают MAC для сред 802.x;
- Ethernet сетевая технология по стандарту IEEE 802.3 для сетей, использующая шинную топологию и коллективный доступ с прослушиванием несущей частоты и обнаружением конфликтов;
- Token ring сетевая технология по стандарту IEEE 802.5, использующая кольцевую топологию и метод доступа к кольцу с передачей маркера;
- FDDI (Fiber Distributed Date Interface Station) сетевая технология по стандарту IEEE 802.6, использующая оптоволоконный носитель;
- X.25 международный стандарт для глобальных коммуникаций с коммутацией пакетов;
- Frame relay сеть, организованная из технологий X25 и ISDN.

Физический уровень

Физический уровень предназначен для сопряжения с физическими средствами соединения. Физические средства соединения – это совокупность физической среды, аппаратных и программных средств, обеспечивающая передачу сигналов между системами.

Физическая среда – это материальная субстанция, через которую осуществляется передача сигналов. Физическая среда является основой, на которой строятся физические средства соединения. В качестве физической среды широко используются эфир, металлы, оптическое стекло и кварц.

Физический уровень состоит из Подуровня стыковки со средой и Подуровня преобразования передачи.

Первый из них обеспечивает сопряжение потока данных с используемым физическим каналом связи. Второй осуществляет преобразования, связанные с применяемыми протоколами. Физический уровень обеспечивает физический интерфейс с каналом передачи данных, а также описывает процедуры передачи сигналов в канал и получения их из канала. На этом уровне определяются электрические, механические, функциональные и процедурные параметры для физической связи в системах. Физический уровень получает пакеты данных от вышележащего канального уровня и преобразует их в оптические или электрические сигналы, соответствующие 0 и 1 бинарного потока. Эти сигналы посылаются через среду передачи на приемный узел.

Механические и электрические/оптические свойства среды передачи определяются на физическом уровне и включают:

- тип кабелей и разъемов;
- разводку контактов в разъемах;
- схему кодирования сигналов для значений 0 и 1.

Физический уровень выполняет следующие функции:

1. Установление и разъединение физических соединений.

2. Передача сигналов в последовательном коде и прием.
3. Прослушивание, в нужных случаях, каналов.
4. Идентификация каналов.
5. Оповещение о появлении неисправностей и отказов.

Оповещение о появлении неисправностей и отказов связано с тем, что на физическом уровне происходит обнаружение определенного класса событий, мешающих нормальной работе сети (столкновение кадров, посланных сразу несколькими системами, обрыв канала, отключение питания, потеря механического контакта и т.д.). Виды сервиса, предоставляемого каналному уровню, определяются протоколами физического уровня. Прослушивание канала необходимо в тех случаях, когда к одному каналу подключается группа систем, но одновременно передавать сигналы разрешается только одной из них. Поэтому прослушивание канала позволяет определить, свободен ли он для передачи. В ряде случаев для более четкого определения структуры физической уровень разбивается на несколько подуровней. Например, физический уровень беспроводной сети делится на три подуровня.

1c	Подуровень, не зависимый от физических средств соединения
1b	Переходный подуровень
1a	Подуровень, зависимый от физических средств соединения

Физический уровень беспроводной локальной сети

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером. Повторители являются единственным типом оборудования, которое работает только на физическом уровне.

Физический уровень может обеспечивать как асинхронную (последовательную) так и синхронную (параллельную) передачу, которая применяется для некоторых мэйнфреймов и мини-компьютеров. На Физическом уровне должна быть определена схема кодирования для представления двоичных значений с целью их передачи по каналу связи. Во многих локальных сетях используется манчестерское кодирование.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров,

манчестерский код для представления данных и другие характеристики среды и электрических сигналов.

К числу наиболее распространенных спецификаций физического уровня относятся:

- EIA-RS-232-C, CCITT V.24/V.28 – механические/электрические характеристики несбалансированного последовательного интерфейса;
- EIA-RS-422/449, CCITT V.10 – механические, электрические и оптические характеристики сбалансированного последовательного интерфейса;
- Ethernet – сетевая технология по стандарту IEEE 802.3 для сетей, использующая шинную топологию и коллективный доступ с прослушиванием несущей и обнаружением конфликтов;
- Token ring – сетевая технология по стандарту IEEE 802.5, использующая кольцевую топологию и метод доступа к кольцу с передачей маркера.

Лекция 6. Сетевые (online) сервисы Интернета(2 часа)

Основные вопросы

1. Сервисы для хранения информации в Сети.
2. Сервисы для организации совместной работы с документами в Сети.
3. Сервисы для организации видео и аудио трансляций и веб-конференций.
4. Сервисы для создание информационного ресурса в Сети: сайты и блоги.
5. Сервисы для составление маршрутов и виртуальных экскурсий.

Цель. Дать понятие об основных сетевых (online) сервисах, познакомить с основными направлениями их использования в управленческой и маркетинговой деятельности культуролога-менеджера.

Сетевое сообщество - группа людей, поддерживающих общение и ведущих совместную деятельность при помощи компьютерных сетевых средств.

Социальный сетевой сервис — виртуальная площадка, связывающая людей в сетевые сообщества с помощью программного обеспечения, компьютеров, объединенных в сеть (Интернет) и сети документов (Всемирной паутины).

Интернет открывает широкие возможности для модернизации управленческой и маркетинговой деятельности культуролога-менеджера.

Рассмотрим основные направления использования социальных сетевых сервисов при организации руководства и исследований в профессиональной деятельности культуролога.

Первое направление — хранение информации в «облачных» сервисах, таких как Dropbox, Google Drive, OneDrive и др. Доступ к размещенным данным можно получить в любое время, что позволяет не использовать дополнительные носители информации. Постоянный доступ к «облаку» позволяет более удобно и безопасно (по сравнению с локальными хранилищами) пользоваться всей промежуточной и итоговой учебной информацией в любом месте и в любое время.

Второе направление – совместная работа с документами в Сети (Google Docs, OneDrive и др.). Возможность совместно и одновременно из разных расположений создавать, просматривать и редактировать текстовые, табличные и мультимедийные документы с отображением всех изменений в реальном времени весьма полезна для организации самостоятельной коллективной проектной деятельности студентов.

Третье направление — видео и аудио трансляции (YouTube, Smotri.com и др.) и веб-конференции (Cisco WebEx, GoToMeeting и др.). С помощью веб-конференций удобно организовывать консультации, семинары, групповые проекты. Кроме возможности видеть и слышать участников конференции, есть возможность организации совместной работы с материалами различного типа. Умения по транслированию в реальном времени и публикация мультимедийных материалов на сайте особенно актуальны для будущих культурологов, одна из профессиональных задач которых является организация социального взаимодействия.

Четвертое направление — создание информационного ресурса (Wix, Sites.google, LiveJournal, Blogger, Wordpress и др.). Для организации и проведения различных мероприятий, можно создать собственный блог.

Пятое направление — составление маршрутов и виртуальных экскурсий (WikiMapia, Google Maps и др.). Умения находить, отмечать, снабжать фотографиями, комментировать объекты на карте обязательны для будущих культурологов, так как они должны знать и популяризировать культурные достопримечательности различных стран.

Сетевые сервисы в настоящее время быстро развиваются. В данной классификации рассмотрены только основные виды социальных сетевых сервисов, которые могут быть использованы в управленческой и маркетинговой деятельности культуролога-менеджера.

Выводы. Сетевые сервисы при организации руководства и исследований в профессиональной деятельности культуролога можно использовать для совместного поиска и хранения информации, создания и использования медиа-

материалов, редактирования и использования карт и схем; организации коллективной работы над проектами и др.

Ключевые понятия: сетевое сообщество, социальный сетевой сервис, сервисы для хранения информации в Сети, сервисы для организации совместной работы с документами, сервисы для организации видео и аудио трансляций и веб-конференций, сервисы для создания сайта, блога, геосервисы.

Тема 4. Системы управления базами данных

Лекции 7

Организация баз данных

Информация и информационные системы

В деятельности любого предприятия, учреждения, организации существенную роль играют данные. Данные фиксируются в определенной форме, пригодной для последующей обработки, хранения и передачи (на бумаге, магнитных дисках, магнитных лентах, компакт-дисках и т.п.).

Из данных извлекается необходимая информация. Данные в этом смысле можно рассматривать как сырье (ресурс) для производства информации. В результате обработки данные приобретают смысл, т.е. становятся информацией.

Под информацией понимают любые сведения о каком-либо событии, процессе, объекте, которые можно воспринимать, передавать, хранить или использовать.

На предприятии или в учреждении необходима информация для управления финансовыми, трудовыми и материальными ресурсами. Для управления финансовыми ресурсами надо иметь следующую информацию: источники денежных поступлений и их объемы, сколько денег израсходовано и на что, сколько средств предстоит получить и сколько осталось.

Управлять трудовыми ресурсами можно, когда известна информация о числе сотрудников, их специальности или профессии, должностном окладе, местонахождения рабочего места, прошлых достижениях сотрудников, сегодняшнем положении, возможности продвижения по службе.

Для управления материальными ресурсами требуется знать: какие материалы есть в наличии, откуда они поступают, какое количество их требуется, сколько уже израсходовано, сроки поставки материалов и другую информацию.

Информация, предназначенная для управления учреждением, условно может быть разбита на три уровня. К нижнему уровню управления относится оперативная информация. Эта информация используется сотрудниками подразделений учреждения в повседневной работе. Она представляет собой

часто обновляемую, первичную, рутинную информацию. Поэтому в информационных системах в первую очередь подлежит автоматизации обработка оперативной информации.

На среднем уровне управления имеют дело с тактической информацией. Эта информация предназначена для руководителей среднего звена. Тактическая информация получается путем обобщения оперативной информации и может быть представлена в виде отчетов или различных вариантов решения.

К верхнему уровню управления относят стратегическую информацию. Она получается в результате обработки оперативной и тактической информации. Эта информация содержит краткие, но содержательные сводки, отчеты, прогнозы. На ее основе осуществляется долгосрочное планирование работы учреждения.

Каждое предприятие, учреждение, организацию можно рассматривать как информационную систему, состоящую из элементов, связей между ними, по которым циркулирует некоторая информация. Информационная система функционирует на базе некоторой информационной технологии. В понятие информационной технологии входят устройства, носители информации, методы хранения, переработки и обмена информацией. Например, информационные технологии 30-40 г.г. строились на базе телефона, почты, устных сообщений, отчетов, различных форм и бланков, бумаги и т.п.

Задачей разработчиков автоматизированных информационных систем (АИС) является включение в существующие информационные системы со своей информационной технологией элементов автоматизации на всех уровнях и создание на базе персональных компьютеров и вычислительных сетей новой информационной технологии.

Базы данных и их модели

Первые АИС создавались на основе так называемого позадачного метода. При таком методе решались вопросы автоматизации задач оперативного уровня. Выбирались самые очевидные, формализуемые (имеющие алгоритмы решения) задачи, автоматизация которых сразу давала максимальную эффективность. Это были задачи расчета заработной платы, снабжения, кадры и т.п.

Но вскоре разработчики АИС столкнулись с трудностями и проблемами, которые оказались непреодолимыми при таком подходе. Требовался новый более качественный метод. Были сформулированы стандартные требования к организации данных в АИС. Основные из них приведены ниже:

– интеграция данных. Это означает, что все данные должны накапливаться и храниться централизованно, создавая в реальном масштабе времени обновляемую модель предметной области. Предметная область (ПО) – часть реального мира, подлежащая автоматизации;

– максимально возможная независимость прикладных программ от данных, т.е. отделение логической модели ПО от физического представления в памяти компьютера или, говоря другими словами, обеспечение логической и физической независимости данных;

– безопасность данных. Под безопасностью данных понимают защиту данных от случайного или преднамеренного доступа к ним лиц, не имеющих на это право;

– минимальная избыточность данных, т.е. требования новых приложений должны удовлетворяться за счет существующих данных, а не путем создания новых файлов;

– обработка непредсказуемых запросов должна быть обеспечена с помощью высокоуровневого языка запроса или пользовательского интерфейса генерации отчетов.

Выполнение этих требований привело к созданию единого для многих приложений блока данных, который мы будем называть базой данных – БД, и разработке одной управляющей программы для манипулирования данными на физическом уровне, называемой системой управления базами данных – СУБД.

Отметим, что не каждый блок данных является БД. БД – это совокупность данных, обладающих следующими качествами:

– интегрированностью, направленной на решение общих задач;

– модельностью, т.е. структурированностью, отражающей некоторую часть реального мира;

– взаимосвязанностью;

– независимостью описания данных от прикладных программ, поскольку данные и их описания хранятся совместно в БД.

Аналогично не всякая управляющая программа работы с БД является СУБД. СУБД – это пакет программ, позволяющий:

– обеспечить пользователей языковыми средствами описания и манипулирования данными;

– обеспечить поддержку логических моделей данных. Модель данных определяет логическое представление физических данных;

– обеспечить операции создания и манипулирования логическими данными (выбор, вставка, обновление, удаление и т.п.) и одновременное выполнение этих операций над физическими данными;

– обеспечить защиту и целостность данных, поскольку при коллективном режиме работы многих пользователей возможно использование общих физических данных. Поэтому необходимо обеспечить защиту от некорректных обновлений пользователями, защиту от несанкционированного доступа, защиту данных от разрушений при сбоях оборудования.

Иногда для совокупности баз данных употребляют термин банк данных. Некоторые специалисты вместо базы данных используют термин банк данных. Часто в литературе термин банк данных дается без четкого определения. В нашем пособии термин банк данных включает в себя БД, СУБД, аппаратные средства, обслуживающие службы и некоторые другие компоненты.

Пользователя БД интересует ее информационное и смысловое содержание. Детали организации физического хранения данных нас интересовать не будут. Поэтому в представлении данных в БД мы будем выделять два уровня абстракции: информационную модель и физическую модель.

Информационная модель должна отображать ПО в терминах, понятных и привычных для пользователя. Обычно это информация об интересующих его фактах, явлениях, событиях, предметах и связях между ними.

Проектировщики информационной модели термином сущность называют объект любой природы, о котором надо хранить информацию в БД. Например, множество студентов можно назвать сущностью 'студент'. Один объект сущности называют экземпляром сущности. Свойства, характеризующие сущность, называют атрибутами. Примерами атрибутов сущности 'студент' являются Номер зачетной книжки, Фамилия, Имя, Отчество, Факультет, Курс, Группа и т.п. Между различными сущностями ПО и их атрибутами могут существовать межсущностные и межатрибутные связи, информационно важные для пользователя БД.

Информационная модель ПО с выделенными в ней сущностями, атрибутами и связями должна быть описана для представления в компьютере. Это описание делается средствами модели данных, которую поддерживает СУБД, и называется внутренней схемой информационной модели.

СУБД поддерживает некоторую модель данных и отображает ее в соответствующие структуры физической базы данных. Средствами модели данных СУБД строится логическая внутренняя схема информационной модели ПО. Прикладная программа, которая пишется в терминах модели данных СУБД, поддерживает логику внешней информационной модели ПО для пользователя, основываясь на внутренней схеме ПО.

Модели данных, поддерживаемые СУБД, делят на сетевые, иерархические и реляционные. Соответственно различают иерархические, сетевые и реляционные СУБД. Сетевые и иерархические СУБД получили наибольшее распространение на больших и мини- ЭВМ.

В иерархических СУБД данные представляются в виде древовидной структуры. Дерево представляет собой иерархию элементов, называемых узлами. На самом верхнем уровне иерархии имеется только один узел – корень. Каждый узел, кроме корня, связан с одним узлом на более высоком уровне,

называемым исходным узлом для данного узла. Ни один элемент не имеет более одного исходного. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне. Они называются порожденными.

Если порожденный элемент в отношении между данными имеет более одного исходного элемента, то это отношение нельзя описать как древовидную структуру. Его описывают в виде сетевой структуры. Любой элемент в сетевой структуре может быть связан с любым другим элементом. Такие структуры используются для представления данных в сетевых СУБД.

Иерархические и сетевые модели данных имеют тенденцию к усложнению описания ПО по мере роста БД и требуют высокой квалификации пользователя. Была найдена другая модель для представления данных – реляционная. Более того было показано, что при помощи нормализации иерархические и сетевые модели данных могут быть приведены к реляционной. Практически все СУБД, предназначенные для персональных компьютеров, поддерживают реляционную модель данных.

Реляционные базы данных

В основе реляционной модели данных лежит понятие отношения (англ. relation). Отношение удобно представляется в виде двумерной таблицы при соблюдении определенных ограничивающих условий. Таблица привычна для пользователя, понятна и обозрима, ее легко запомнить.

Основоположник теории реляционных баз данных Е.Ф.Кодд показал, что набор отношений (таблиц) может быть использован для хранения данных об объектах реального мира и моделирования связей между ними. Например, для хранения сущности ‘слушатель’ используют отношение СТУДЕНТ, в котором свойства сущности располагаются в столбцах таблицы:

СТУДЕНТ

ФИО	Факультет	Курс	Группа
Сидорова А.И.	ФКиСКД	3	308
Василевский В.И.	ФКиСКД	2	212
Дубова В.А.	ТБКиСИ	3	314
Алексеенко С.А.	ФМИ	1	116
Королева В.С.	ТБКиСИ	2	213

Столбцы отношения называют атрибутами, атрибутам присваивают имена. Список имен атрибутов отношения называют схемой отношения. Список значений атрибутов отношения называют картежом. Схема отношения СТУДЕНТ записывается так:

СТУДЕНТ (Фамилия, Факультет, Курс, Группа).

Реляционная БД – это набор взаимосвязанных таблиц. Набор взаимосвязанных таблиц на физическом уровне (на внешних носителях информации) хранится в виде файла БД. Соответствие между элементами файла БД, таблицы, отношения и сущности, когда файл БД содержит одну таблицу, может быть показано следующим образом:

Файл БД	Таблица	Отношение	Сущность
запись	строка	кортеж	экземпляр сущности
поле	столбец	атрибут	атрибут

Важным достоинством реляционной модели данных является то, что к схемам отношений можно применять соответствующие операции и тем самым получать новые схемы отношения, которые ранее в файле БД не были представлены. Таким способом СУБД позволяет получать ответы на незапланированные прикладными программами информационные запросы.

Лекция 8

Нормализация отношений

В зависимости от содержания отношений реляционных БД мы будем различать два типа отношений: объектные и связные. Объектное отношение хранит данные об объектах. Рассмотренное выше отношение СТУДЕНТ является примером объектного отношения.

В объектном отношении один из атрибутов однозначно идентифицирует объект. Такой атрибут называют ключом отношения. В отношении СТУДЕНТ ключом может быть атрибут Фамилия. Для удобства ключ размещают в первом столбце таблицы. Ключ может состоять из нескольких атрибутов или быть частью одного атрибута.

Основное ограничение реляционной модели БД состоит в следующем. В объектном отношении не должно быть строк с одинаковыми ключами. Это ограничение позволяет обеспечить целостность БД.

Связное отношение хранит ключи двух или более объектных отношений, по этим ключам устанавливаются связи между объектами. Пусть в БД имеются два отношения: ФИРМА (Название, Адрес) и ТОВАР (Наименование, Цена). Из них может быть получено связное отношение ПОСТАВЛЯЕТ (Фирма, Товар).

Связное отношение кроме связываемых ключей может иметь и другие атрибуты, которые будут функционально зависеть от этой связи. Например, отношение ПОСТАВЛЯЕТ может иметь следующий вид: ПОСТАВЛЯЕТ(Фирма, Товар, Цена). Ключи в связных отношениях

называются внешними ключами, поскольку они являются первичными ключами других отношений.

Каждому внешнему ключу должна соответствовать строка какого-либо объектного отношения. Невыполнение этого ограничения может привести к нарушению ссылочной целостности данных. Ключ должен ссылаться на объект, который существует.

Отношения в БД должны быть нормализованы. Это означает, что каждый атрибут должен быть простым - содержать неделимые значения. В приведенном ниже отношении условие нормализации не выполнено.

ПРЕПОДАВАТЕЛЬ

Фамилия	Кафедра	Предмет	Телефон	
			рабочий	домашний
Голубев	Информатика	Гипертекстовые системы	20-08-35	70-01-32
Погорелова	Психология	Практическая психология	20-08-35	50-85-30
Михайлов	Методика воспитания	Методика воспитания	20-15-91	53-33-45
Новожилов	Информатика	Информационные технологии	20-08-35	50-85-30
Антипов	Педагогика	Педагогика	20-13-04	55-93-14

Это отношение можно нормализовать разбиением сложного атрибута Телефон на два простых: Телефон рабочий и Телефон домашний. Схема нормализованного отношения выглядит так:

ПРЕПОДАВАТЕЛЬ (Фамилия, Кафедра, Предмет, Телефон рабочий, Телефон домашний).

Отношение, у которого все атрибуты простые, называется приведенным к первой нормальной форме (1НФ).

Перечислим условия и ограничения, накладываемые на отношения реляционной моделью данных, которые позволяют таблицы считать отношениями:

1. Не может быть одинаковых первичных ключей, т.е. все строки таблицы должны быть уникальны.

2. Все строки таблицы должны иметь одну и ту же структуру, т.е. одно и то же количество атрибутов с соответственно совпадающими именами.

3. Имена столбцов таблицы должны быть различны, а значения столбцов должны быть однотипными.

4. Значения атрибутов должны быть простыми, следовательно, отношения не могут иметь в качестве компонент другие отношения.

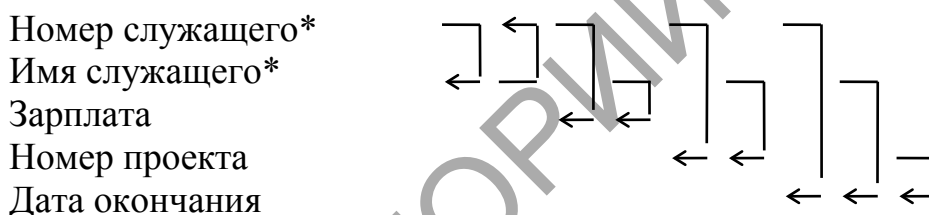
5. Должна соблюдаться ссылочная целостность для внешних ключей.

6. Порядок следования строк в таблице несущественен - он лишь влияет на скорость доступа к строке.

Задавая отношения над элементами данных, проектировщик БД определяет, какие из атрибутов объекта являются зависимыми. Термин функциональная зависимость означает следующее. Пусть имеется два атрибута: А и В. Если в любой момент времени каждому значению А соответствует не более чем одно значение атрибута В, говорят, что В функционально зависит от А. Функциональная зависимость обозначается так:

$A \rightarrow B$.

Рассмотрим следующее отношение: СЛУЖАЩИЙ (Номер служащего, Имя служащего, Зарплата, Номер проекта, Дата окончания). Функциональные зависимости между атрибутами СЛУЖАЩИЙ для наглядности представим в виде диаграммы.



В нашем примере звездочками отмечены основные атрибуты. Основные атрибуты являются элементами возможных ключей отношения. Атрибут Имя служащего функционально зависит от атрибута Номер служащего и, в свою очередь, атрибут Номер служащего функционально зависит от атрибута Имя служащего. Атрибут Зарплата функционально зависит от атрибута Номер служащего или Имя служащего. Аналогично атрибут Номер проекта функционально зависит от атрибута Номер служащего или Имя служащего. Атрибут Дата окончания обладает функциональной зависимостью от атрибутов: Номер служащего, Имя служащего и Номер проекта.

Атрибут может функционально зависеть не от одного какого-то атрибута, а от целой группы атрибутов. Рассмотрим, например, отношение, содержащее информацию о расходе рабочего времени программиста:

РАБОТА ПРОГРАММИСТА (Номер программиста, Номер программы, Имя программиста, Имя программы, Количество часов).

Атрибут Количество часов функционально зависит от составного ключа (Номер программиста, Номер программы) или от одного из следующих возможных ключей: (Номер программиста, Имя программы), (Имя программиста, Номер программы) или (Имя программиста, Имя программы).

Заранее предполагается, что среди программистов нет однофамильцев и что две программы не могут иметь одинаковых имен.

Атрибут (или набор атрибутов) B из отношения R называется полностью зависимым от другого набора атрибутов A отношения R , если B функционально зависит от всего множества A , но не зависит ни от какого подмножества A .

Например, в отношении РАБОТА ПРОГРАММИСТА атрибут Количество часов является полностью зависимым от составного ключа (Номер программиста, Номер программы), так как он задает количество рабочего времени, затраченного данным программистом на конкретную программу. При этом ни один из атрибутов Номер программиста или Номер программы в отдельности не определяет значение атрибута Количество часов.

Говорят, что отношение задано во второй нормальной форме (2НФ), если оно является отношением в первой нормальной форме, и каждый атрибут, не являющийся основным атрибутом в этом отношении, полностью зависит от любого возможного ключа этого отношения.

Если все возможные ключи отношения содержат по одному атрибуту, как это было в ранее рассмотренном отношении СЛУЖАЩИЙ, то это отношение задано во второй нормальной форме. Если ключи состоят более чем из одного атрибута, отношение, заданное в 1НФ, может не быть отношением в 2НФ. Отношение РАБОТА ПРОГРАММИСТА задано в 2НФ, потому что единственный атрибут Количество часов, не являющийся основным, полностью зависит от каждого возможного ключа.

Рассмотрим пример отношения, которое не является отношением в 2НФ:

ИСТОЧНИК СНАБЖЕНИЯ (Номер поставщика, Номер партии товара, Имя поставщика, Сведения о поставщике, Цена).

У этого отношения один возможный ключ, который является составным: (Номер поставщика, Номер партии товара). Атрибут Имя поставщика не входит в ключ, так как одной и той же фирме в разных районах могут быть присвоены различные номера поставщика. Таким образом, атрибут Номер поставщика не определяется значением атрибута Имя поставщика. В этом отношении атрибуты Имя поставщика и Сведения о поставщике, не будучи основными, функционально зависят от атрибута Номер поставщика, который является подмножеством составного ключа.

Нарушение условий 2НФ приводит к ряду неудобств:

1. Графа Сведения о поставщике не может быть заполнена до фактической поставки конкретной партии товара поставщиком, либо необходимо задать какой-нибудь фиктивный номер партии.

2. Если поставщик временно задержал поставку некоторой партии, то удаление кортежа, соответствующего данному значению атрибута Номер поставщика, вызовет удаление сведений о нем.

3. Если требуется изменить значение атрибута Сведения о поставщике, то придется внести одни и те же изменения сразу в несколько кортежей.

Устранение подобных трудностей достигается разложением отношения на два отношения, представленных в 2НФ, говоря другими словами, приведением отношения к 2НФ. Наше отношение ИСТОЧНИК СНАБЖЕНИЯ можно разложить на два отношения:

ПОСТАВЩИК (Номер поставщика, Имя поставщика, Сведения о поставщике) и ТОВАР (Номер поставщика, Номер партии товара, Цена). Легко убедиться, что отношения ПОСТАВЩИК и ТОВАР находятся в 2НФ.

В ряде случаев и вторая нормальная форма порождает неудобства. Для их устранения используется еще один шаг нормализации, преобразующий 2НФ в третью нормальную форму (3НФ). На этом шаге ликвидируется так называемая транзитивная зависимость атрибутов.

Если для атрибутов А, В, С выполняются условия $A \rightarrow B$ и $B \rightarrow C$, но обратная зависимость отсутствует, то говорят, что С зависит от А транзитивно. Отношение находится в 3НФ, если оно находится в 2НФ и в нем отсутствуют транзитивные зависимости неосновных атрибутов от каждого возможного ключа.

В отношении СЛУЖАЩИЙ, ранее представленном диаграммой, атрибут Дата окончания зависит от атрибута Номер проекта, который в свою очередь зависит от атрибута Номер служащего. Таким образом, Дата окончания транзитивно зависит от атрибута Номер служащего. Это отношение можно привести к 3НФ, расщепив его на два отношения:

СЛУЖАЩИЙ (Номер служащего, Имя служащего, Зарплата, Номер проекта), ПРОЕКТ (Номер проекта, Дата окончания).

Следует отметить, что уже известны и другие нормальные формы отношений: усиленная 3НФ - нормальная форма Бойса-Кодда, четвертая нормальная форма и пятая нормальная форма. Они позволяют устранить зависимости ключей от неосновных атрибутов, независимые многозначные зависимости и т.д. Рассмотрение этих форм выходит за пределы данной публикации.

При проектировании БД процессу нормализации отношений отводится следующее место:

– вначале составляются исходные отношения проекта БД с использованием объектно-связной модели для отображения объектов предметной области и связей между ними;

– затем производится нормализация, т.е. разложение исходных отношений и назначение ключей новых отношений в соответствии с правилами нормализации;

– далее схемы нормализованных отношений описываются средствами СУБД и вводятся в компьютер.

Лекция 9

Жизненный цикл автоматизированных информационных систем

По мнению специалистов, из-за многообразия типов АИС единой модели, которая описывала бы их жизненный цикл, не существует. В литературе был предложен целый спектр моделей, на противоположных концах которых находятся так называемые каскадная и спиральные модели. По-видимому, оптимум находится где-то посередине между ними. Такая комбинированная модель жизненного цикла АИС содержит следующие стадии:

- 1) разработка стратегии автоматизации;
- 2) оценивание реализуемости;
- 3) анализ требований;
- 4) разработка технического задания;
- 5) логическое проектирование;
- 6) физическое проектирование;
- 7) программирование;
- 8) отладка и испытание;
- 9) внедрение;
- 10) сопровождение;
- 11) анализ опыта эксплуатации.

Результатом выполнения стадии 1 являются документы, инициирующие разработку, в которых должны быть сформулированы стратегические цели и общий замысел автоматизации. Стадии 1, 7-11 являются каскадными, т.е. они выполняются последовательно друг за другом. Стадии 2-6 являются спиральными – в процессе их выполнения происходят многократные встречи проектировщиков АИС с заказчиками, сопровождающиеся многократными уточнениями и корректировками разрабатываемых документов. Далее стадии 2-6 будем называть технологией проектирования SSADM (Structured Systems Analysis and Design Method). Разработка этой технологии началась с середины 70-х годов в Великобритании и в 1993 году она была официально принята в качестве государственного стандарта Великобритании.

Основополагающие принципы технологии SSADM приведены ниже:

– постоянное вовлечение представителей будущих пользователей в процесс выработки решений на протяжении всего проектирования АИС;

– четкая структуризация технологического процесса, взаимная увязка всех стадий, этапов и проектных процедур, явная регламентация ролей всех участников разработки;

– эффективный контроль за ходом разработки со стороны руководителей проекта, встроенный контроль качества проектирования по формализованным критериям, возможность применения существующих технологий автоматизированного управления разработкой;

– стыковка с технологиями, реализованными в существующих системах программирования и управления БД;

– формализация процесса разработки, обеспечивающая широкое применение средств автоматизации проектирования.

Технологическая стадия 2 не является обязательной и может быть пропущена, если ранее были проведены достаточно глубокие исследования при выработке стратегии автоматизации. Основной целью стадии является предварительное технико-экономическое обоснование проекта и разработка концепции будущей АИС.

Стадия 3 состоит из двух этапов: предпроектного обследования и выбора варианта автоматизации. На этапе предпроектного обследования определяют основные требования к новой АИС. Для этого изучают существующую систему обработки информации, составляя с участием пользователей ее логическое описание в терминах потоков данных, задач и информационных объектов. Определяют границы существующей системы, внешние объекты и функции пользователей. При этом, анализируя ее недостатки, формулируют основные требования к новой АИС, которые отражают в Каталоге требований.

На этапе выбора варианта автоматизации путем упорядочения требований по важности выбирают различные их подмножества и составляют описания нескольких вариантов разрабатываемой АИС. При этом также выполняют технико-экономические расчеты, позволяющие на этой стадии сравнить варианты и при активном участии будущих пользователей обоснованно выбрать среди них оптимальный.

На стадии 4 полностью определяют требования к выбранному варианту построения АИС, которые, как и на этапе предпроектного обследования стадии 3, формулируют в терминах потоков данных, задач и информационных объектов. Отличие состоит в том, что речь идет не о существующей, а о новой системе, причем ее описывают значительно подробнее. Кроме того, в терминах событий и данных разрабатывают требования к динамике функционирования АИС. Характерно, что на этой стадии предусмотрена также разработка демонстрационного прототипа. Это позволяет частично реализовать достоинства спиральной модели жизненного цикла без присущего ей усложнения контроля за ходом разработки со стороны проекта.

На стадии 5 обосновывают выбор технической и программной сред реализации создаваемой АИС и параллельно с этим проектируют диалоговое взаимодействие пользователей с системой, а также разрабатывают и увязывают между собой постановки задач.

Стадия 6 является заключительной в технологии проектирования SSADM. На ней разрабатывают описания данных на физическом уровне, выполняют их оптимизацию, уточняют постановки задач и готовят руководящие указания по генерации баз данных и разработке программного обеспечения к выбранной среде реализации.

Таким образом, основным продуктом, создаваемым по технологии SSADM, является комплект документов, на основе которых может быть реализована разрабатываемая АИС на компьютере с использованием системы программирования и СУБД, выбранных на стадии 5.

Последовательность выполнения технологических стадий и состав разрабатываемых на них проектных документов четко регламентирован 13 методиками, составляющими методическое обеспечение технологии SSADM. Перечень их с краткой характеристикой приводится ниже:

№ п/п	Название методики	Характер основных проектных документов	Номера стадий, на которых применяется методика
1.	Определение требований к АИС	Табличная форма	2, 3, 4, 5, 6
2.	Моделирование информационных потоков	Схема, табличная форма	2, 3
3.	Логическое моделирование данных	Схема, табличная форма	2, 3, 4, 5
4.	Определение функций и задач	Схема, табличная форма	3, 4
5.	Динамическое моделирование данных	Схема, табличная форма	4, 5
6.	Реляционный анализ данных	Табличная форма	3, 4, 5
7.	Выбор вариантов автоматизации	Пояснительная записка, схема	3
8.	Разработка демонстрационного прототипа	Видеограмма, схема	4
9.	Выбор вариантов технической реализации	Пояснительная записка, табличная форма	5
10.	Проектирование диалогового	Табличная форма,	4, 5

	взаимодействия	схема	
11.	Логическое проектирование процедур обработки	Табличная форма, схема	5
12.	Физическое проектирование БД	Табличная форма, схема	6
13.	Физическое проектирование процедур обработки	Табличная форма, схема	6

В рамках типового технологического процесса применение отдельных методик, например, “Выбор вариантов автоматизации”, ограничено одной технологической стадией. Однако вследствие итерационного характера разработки основных проектных документов большинство методик используется на протяжении нескольких стадий.

Основное назначение рассматриваемой технологии проектирования АИС заключается в том, чтобы разработчики могли выявить скрытые в проекте противоречия и устранить ошибки задолго до стадии отладки и испытания.

Тема 5. Защита информации

Лекция 10

Защита информации в информационных системах

Основные вопросы

1. Виды угроз безопасности информационных систем.
2. Защиты информации в экономических информационных системах.
3. Виды защиты в информационных системах маркетинговой деятельности.

Цель: изучение методов и способов защиты информации.

Виды угроз безопасности информационных систем

Наряду с интенсивным развитием вычислительных средств и систем передачи информации все более актуальной становится проблема обеспечения ее безопасности. Меры безопасности направлены на предотвращение несанкционированного получения информации, физического уничтожения или модификации защищаемой информации.

Зарубежные публикации последних лет показывают, что возможности злоупотреблений информацией, передаваемой по каналам связи, развивались и совершенствовались не менее интенсивно, чем средства их предупреждения. В этом случае для защиты информации требуется не просто разработка частных механизмов защиты, а организация комплекса мер, т.е. использование специальных средств, методов и мероприятий с целью предотвращения потери информации. В этом смысле сегодня рождается новая современная техно-

логия – технология защиты информации в компьютерных информационных системах и в сетях передачи данных.

Несмотря на предпринимаемые дорогостоящие методы, функционирование компьютерных информационных систем выявило наличие слабых мест в защите информации. Неизбежным следствием стали постоянно увеличивающиеся расходы и усилия на защиту информации. Однако для того, чтобы принятые меры оказались эффективными, необходимо определить, что такое угроза безопасности информации, выявить возможные каналы утечки информации и пути несанкционированного доступа к защищаемым данным.

Под угрозой безопасности понимается действие или событие, которое может привести к разрушению, искажению или несанкционированному использованию информационных ресурсов, включая хранимую, передаваемую и обрабатываемую информацию, а также программные и аппаратные средства.

Угрозы принято делить на случайные, или непреднамеренные, и умышленные. Источником первых могут быть ошибки в программном обеспечении, выходы из строя аппаратных средств, неправильные действия пользователей или администрации и т.п. Умышленные угрозы преследуют цель нанесения ущерба пользователям АИТ и, в свою очередь, подразделяются на активные и пассивные.

Пассивные угрозы, как правило, направлены на несанкционированное использование информационных ресурсов, не оказывая при этом влияния на их функционирование. Пассивной угрозой является, например, попытка получения информации, циркулирующей в каналах связи, посредством их прослушивания.

Активные угрозы имеют целью нарушение нормального процесса функционирования системы посредством целенаправленного воздействия на аппаратные, программные и информационные ресурсы. К активным угрозам относятся, например, разрушение или радиоэлектронное подавление линий связи, вывод из строя ПЭВМ или ее операционной системы, искажение сведений в базах данных либо в системной информации и т.д. Источниками активных угроз могут быть непосредственные действия злоумышленников, программные вирусы и т.п.

К основным угрозам безопасности информации относят:

- раскрытие конфиденциальной информации;
- компрометация информации;
- несанкционированное использование информационных ресурсов;
- ошибочное использование ресурсов;
- несанкционированный обмен информацией;
- отказ от информации;
- отказ от обслуживания.

Средствами реализации угрозы раскрытия конфиденциальной информации могут быть несанкционированный доступ к базам данных, прослушивание каналов и т.п. В любом случае получение информации, являющейся достоянием некоторого лица (группы лиц), другими лицами наносит ее владельцам существенный ущерб.

Компрометация информации, как правило, реализуется посредством внесения несанкционированных изменений в базы данных, в результате чего ее потребитель вынужден либо отказаться от нее, либо предпринимать дополнительные усилия для выявления изменений и восстановления истинных сведений. В случае использования скомпрометированной информации потребитель подвергается опасности принятия неверных решений со всеми вытекающими последствиями.

Несанкционированное использование информационных ресурсов, с одной стороны, является средством раскрытия или компрометации информации, а с другой – имеет самостоятельное значение, поскольку, даже не касаясь пользовательской или системной информации, может нанести определенный ущерб абонентам и администрации. Этот ущерб может варьировать в широких пределах – от сокращения поступления финансовых средств до полного выхода АИТ из строя.

Ошибочное использование информационных ресурсов, будучи санкционированным, тем не менее может привести к разрушению, раскрытию или компрометации указанных ресурсов. Данная угроза чаще всего является следствием ошибок в программном обеспечении АИТ.

Несанкционированный обмен информацией между абонентами может привести к получению одним из них сведений, доступ к которым ему запрещен, что по своим последствиям равносильно раскрытию содержания маркетинговой информации.

Отказ от информации состоит в непризнании получателем или отправителем информации фактов ее получения или отправки. В условиях маркетинговой деятельности это, в частности, позволяет одной из сторон расторгать заключенные финансовые соглашения «техническим» путем, формально не отказываясь от них и нанося тем самым второй стороне значительный ущерб.

Отказ в обслуживании представляет собой весьма существенную и распространенную угрозу, источником которой является сама АИТ. Подобный отказ особенно опасен в ситуациях, когда задержка с предоставлением ресурсов абоненту может привести к тяжелым для него последствиям. Так, отсутствие у пользователя данных, необходимых для принятия решения, в течение периода, когда это решение еще может быть эффективно реализовано, может стать причиной его нерациональных или даже антимонопольных действий.

Основными типовыми путями несанкционированного доступа к информации, сформулированными на основе анализа зарубежной печати, являются:

- перехват электронных излучений;
- принудительное электромагнитное облучение (подсветка) линий связи с целью получения паразитной модуляции;
- применение подслушивающих устройств (закладок);
- дистанционное фотографирование;
- перехват акустических излучений и восстановление текста принтера;
- хищение носителей информации и документальных отходов;
- чтение остаточной информации в памяти системы после выполнения санкционированных запросов;
- копирование носителей информации с преодолением мер защиты;
- маскировка под зарегистрированного пользователя;
- мистификация (маскировка под запросы системы);
- использование программных ловушек;
- использование недостатков языков программирования и операционных систем;
- включение в библиотеки программ специальных блоков типа «Троянский конь»;
- незаконное подключение к аппаратуре и линиям связи;
- злоумышленный вывод из строя механизмов защиты;
- внедрение и использование компьютерных вирусов.

Необходимо отметить, что особую опасность в настоящее время представляет проблема компьютерных вирусов, ибо эффективной защиты против них разработать не удалось. Остальные пути несанкционированного доступа поддаются надежной блокировке при правильно разработанной и реализуемой на практике системе обеспечения безопасности.

Методы и средства защиты информации в экономических информационных системах

При разработке АИТ возникает проблема по решению вопроса безопасности информации, составляющей коммерческую тайну, а также безопасности самих компьютерных информационных систем.

Современные АИТ обладают следующими основными признаками:

- содержат информацию различной степени конфиденциальности;
- при передаче данных имеют криптографическую защиту информации различной степени конфиденциальности;
- отражают иерархичность полномочий субъектов, открывают доступ к программам, к АРМ, файл-серверам, каналам связи и информации системы; необходимость оперативного изменения этих полномочий;

- организуют обработку информации в диалоговом режиме, в режиме разделения времени между пользователями и в режиме реального времени;
- обеспечивают управление потоками информации как в локальных сетях, так и при передаче по каналам связи на далекие расстояния;
- регистрируют и учитывают попытки несанкционированного доступа, события в системе и документах, выводимых на печать;
- обеспечивают целостность программного продукта и информации в АИТ;
- устанавливают наличие средств восстановления системы защиты информации, а также обязательный учет магнитных носителей;
- создают условия для физической охраны средств вычислительной техники и магнитных носителей.

Организационные мероприятия и процедуры, используемые для решения проблемы безопасности информации, решаются на всех этапах проектирования и в процессе эксплуатации АИТ. Существенное значение при проектировании придается предпроектному обследованию объекта. На этой стадии проводятся следующие действия:

- устанавливается наличие конфиденциальной информации в разрабатываемой АИТ, оцениваются уровень конфиденциальности и объемы такой информации;
- определяются режимы обработки информации (диалоговый, телеобработки и реального времени), состав комплекса технических средств, общесистемные программные средства и т.д.;
- анализируется возможность использования имеющихся на рынке сертифицированных средств защиты информации;
- определяется степень участия персонала, функциональных служб, научных и вспомогательных работников объекта автоматизации в обработке информации, характер их взаимодействия между собой и со службой безопасности;
- вводятся мероприятия по обеспечению режима секретности на стадии разработки системы.

Среди организационных мероприятий по обеспечению безопасности информации важное место принадлежит охране объекта, на котором расположена защищаемая АИТ (территория здания, помещения, хранилища информационных носителей). При этом устанавливаются соответствующие посты охраны, технические средства, предотвращающие или существенно затрудняющие хищение средств вычислительной техники, информационных носителей, а также исключают несанкционированный доступ к АИТ и линиям связи.

Функционирование системы защиты информации от несанкционированного доступа как комплекса программно-технических средств и организационных (процедурных) решений предусматривает:

- учет, хранение и выдачу пользователям информационных носителей, паролей, ключей;
- ведение служебной информации (генерация паролей, ключей, сопровождение правил разграничения доступа);
- оперативный контроль за функционированием систем защиты секретной информации;
- контроль соответствия общесистемной программной среды эталону;
- приемку включаемых в АИТ новых программных средств;
- контроль за ходом технологического процесса обработки финансово-кредитной информации путем регистрации анализа действий пользователей;
- сигнализацию опасных событий и т.д.

Следует отметить, что без надлежащей организационной поддержки программно-технических средств защиты информации от несанкционированного доступа и точного выполнения; предусмотренных проектной документацией процедур в должной мере не решить проблему обеспечения безопасности информации, какими бы совершенными эти программно-технические средства ни были.

Создание базовой системы защиты информации в АИТ основывается на следующих принципах:

– Комплексный подход к построению системы защиты при ведущей роли организационных мероприятий. Он означает оптимальное сочетание программных аппаратных средств и организационных мер защиты, подтвержденное практикой создания отечественных и зарубежных систем защиты.

– Разделение и минимизация полномочий по доступу к обрабатываемой информации и процедурам обработки. Пользователям предоставляется минимум строго определенных полномочий, достаточных для успешного выполнения ими своих служебных обязанностей, с точки зрения» автоматизированной обработки доступной им конфиденциальной информации.

– Полнота контроля и регистрации попыток несанкционированного доступа, т.е. необходимость точного установления идентичности каждого пользователя и протоколирования его действий для проведения возможного расследования, а также невозможность совершения любой операции обработки информации в АИТ без ее предварительной регистрации.

– Обеспечение надежности системы защиты, т.е. невозможность снижения ее уровня при возникновении в системе сбоев, отказов,

преднамеренных действий нарушителя или непреднамеренных ошибок пользователей и обслуживающего персонала.

– Обеспечение контроля за функционированием системы защиты, т.е. создание средств и методов контроля работоспособности механизмов защиты.

– «Прозрачность» системы защиты информации для общего, прикладного программного обеспечения и пользователей АИТ.

– Экономическая целесообразность использования системы защиты. Он выражается в том, что стоимость разработки и эксплуатации систем защиты информации должна быть меньше стоимости возможного ущерба, наносимого объекту в случае разработки и эксплуатации АИТ без системы защиты информации.

Проблема создания системы защиты информации включает две взаимодополняющие задачи:

- разработку системы защиты информации (ее синтез);
- оценку, разработанной системы защиты информации.

Вторая задача решается путем анализа технических характеристик защиты информации с целью установления соответствия ее требованиям, предъявляемым к таким системам.

В настоящее время подобные задачи решаются практически исключительно экспертным путем с помощью сертификации средств системы защиты информации и ее аттестации в процессе внедрения.

Раскроем основное содержание представленных средств и методов защиты информации, которые представляют основу механизмов защиты.

Препятствие – метод физического преграждения пути злоумышленнику к защищаемой информации (к аппаратуре, носителям информации и т.д.).

Управление доступом – метод защиты информации с помощью регулирования использования всех ресурсов компьютерной информационной системы банковской деятельности (элементов баз данных, программных и технических средств). Управление доступом включает следующие функции защиты:

– идентификацию пользователей, персонала и ресурсов системы (присвоение каждому объекту персонального идентификатора);

– опознание (установление подлинности) объекта или субъекта по предъявленному им идентификатору;

– проверку полномочий (соответствия дня недели, времени суток, запрашиваемых ресурсов и процедур установленному регламенту);

– разрешение и создание условий работы в пределах установленного регламента;

– регистрацию (протоколирование) обращений к защищаемым ресурсам;

– реагирование (сигнализация, отключение, задержка работ, отказ в запросе) при попытках несанкционированных действий.

Маскировка – метод защиты информации путем ее криптографического закрытия. Этот метод широко применяется за рубежом как при обработке, так и при хранении информации, в том числе на дискетах. При передаче информации по каналам связи большой протяженности данный метод является единственно надежным.

Регламентация – метод защиты информации, создающий такие условия автоматизированной обработки, хранения и передачи защищаемой информации, при которых возможности несанкционированного доступа к ней сводились бы к минимуму.

Принуждение – метод защиты, когда пользователи и персонал системы вынуждены соблюдать правила обработки, передачи и использования защищаемой информации под угрозой материальной, административной или уголовной ответственности.

Побуждение – метод защиты, побуждающий пользователя и персонал системы не разрушать установленные порядки за счет соблюдения сложившихся моральных и этических норм (как регламентированных, так и «неписанных»).

Рассмотренные методы обеспечения безопасности реализуются на практике за счет применения различных средств защиты. К основным средствам защиты, используемым для создания механизма защиты, относятся следующие.

Технические средства представляют электрические, электромеханические и электронные устройства. Вся совокупность указанных средств делится на аппаратные и физические. Под аппаратными техническими средствами принято понимать устройства, встраиваемые непосредственно в вычислительную технику, или устройства, которые сопрягаются с подобной аппаратурой по стандартному интерфейсу:

Физическими средствами являются автономные устройства и системы (замки на дверях, где размещена аппаратура, решетки на окнах, электронно-механическое оборудование охранной сигнализации и др.).

Программные средства – это программное обеспечение, специально предназначенное для выполнения функций защиты информации.

Организационные средства защиты представляют собой организационно-технические и организационно-правовые мероприятия, осуществляемые в процессе создания и эксплуатации вычислительной техники, аппаратуры телекоммуникаций. Организационные мероприятия охватывают все структурные элементы аппаратуры на всех этапах ее жизненного цикла

(проектирование компьютерной информационной системы банковской деятельности, монтаж и наладка оборудования, испытание, эксплуатация).

Морально-этические средства защиты реализуются в виде всевозможных норм, которые сложились традиционно или складываются по мере распространения вычислительной техники и средств связи в обществе. Подобные нормы большей частью не являются обязательными как законодательные меры, однако несоблюдение их ведет обычно к потере авторитета и престижа человека. Наиболее показательным примером таких норм является Кодекс профессионального поведения членов Ассоциаций пользователей ЭВМ США.

Законодательные средства защиты определяются законодательными актами страны, регламентирующими правила пользования, обработки и передачи информации ограниченного доступа и устанавливающими меры ответственности за нарушение этих правил.

Все рассмотренные средства защиты разделены на формальные (выполняющие защитные функции строго по заранее предусмотренной процедуре без непосредственного участия человека) и неформальные (определяемые целенаправленной деятельностью человека либо регламентирующие эту деятельность).

Для реализации мер безопасности используются различные механизмы шифрования (криптографии). Криптография – это наука об обеспечении секретности и/или аутентичности (подлинности) передаваемых сообщений.

Сущность криптографических методов заключается в следующем.

Готовое к передаче сообщение – будь то данные, речь либо графическое изображение того или иного документа, – обычно называется открытым, или незащищенным, текстом (сообщением). В процессе передачи такого сообщения по незащищенным каналам связи оно может быть легко перехвачено или отслежено подслушивающим лицом посредством умышленных или неумышленных действий. Для предотвращения несанкционированного доступа к сообщению оно зашифровывается, преобразуясь в шифrogramму, или закрытый текст. Санкционированный пользователь, получив сообщение, дешифрует или раскрывает его посредством обратного преобразования шифrogramмы, вследствие чего получается исходный открытый текст.

Метод преобразования в криптографической системе определяется используемым специальным алгоритмом, действие которого определяется уникальным числом или битовой последовательностью, обычно называемым шифрующим ключом.

Каждый используемый ключ может производить различные зашифрованные сообщения, определяемые только этим ключом. Для большинства систем закрытия схема генератора ключа может представлять

собой либо набор инструкций, команд, либо часть, узел аппаратуры (hardware), либо компьютерную программу (software), либо все это вместе. Однако в любом случае процесс шифрования/дешифрования определяется единственно выбранным специальным ключом. Таким образом, чтобы обмен зашифрованными сообщениями проходил успешно, и отправителю, и получателю необходимо знать правильную ключевую установку и хранить ее в тайне.

Следовательно, надежность любой системы закрытой связи определяется степенью секретности используемого в ней ключа. Тем не менее, этот ключ должен быть известен другим пользователям сети, чтобы они могли свободно обмениваться зашифрованными сообщениями. Криптографические системы также помогают решить проблему аутентификации (установления подлинности) принятой информации, поскольку подслушивающее лицо, пассивно перехватывающее сообщение, будет иметь дело только с зашифрованным текстом. В то же время истинный получатель, приняв сообщения, закрытые известным ему и отправителю ключом, будет надежно защищен от возможной дезинформации.

Шифрование может быть симметричным и асимметричным. Первое основывается на использовании одного и того же секретного ключа для шифрования и дешифрования. Второе характеризуется тем, что для шифрования используется один общедоступный ключ, а для дешифрования – другой, являющийся секретным, при этом знание общедоступного ключа не позволяет определить секретный ключ.

Наряду с шифрованием внедряются следующие механизмы безопасности:

- цифровая (электронная) подпись;
- контроль доступа;
- обеспечение целостности данных;
- обеспечение аутентификации;
- постановка графика;
- управление маршрутизацией;
- арбитраж или освидетельствование.

Механизмы цифровой подписи основываются на алгоритмах асимметричного шифрования и включают две процедуры: формирование подписи отправителем и ее опознавание (верификацию) получателем. Первая процедура обеспечивает шифрование блока данных либо его дополнение криптографической контрольной суммой, причем в обоих случаях используется секретный ключ отправителя. Вторая процедура основывается на использовании общедоступного ключа, знания которого достаточно для опознавания отправителя.

Механизмы контроля доступа осуществляют проверку полномочий объектов АИТ (программ и пользователей) на доступ к ресурсам сети. При доступе к ресурсу через соединение контроль выполняется как в точке инициации, так и в промежуточных точках, а также в конечной точке.

Механизмы обеспечения целостности данных применяются к отдельному блоку и к потоку данных. Целостность блока является необходимым, но не достаточным условием целостности потока и обеспечивается выполнением взаимосвязанных процедур шифрования и дешифрования отправителем и получателем. Отправитель дополняет передаваемый блок криптографической суммой, а получатель сравнивает ее с криптографическим значением, соответствующим принятому блоку. Несовпадение свидетельствует об искажении информации в блоке. Однако описанный механизм не позволяет вскрыть подмену блока в целом. Поэтому необходим контроль целостности потока, который реализуется посредством шифрования с использованием ключей, изменяемых в зависимости от предшествующих блоков.

Различают одностороннюю и взаимную аутентификацию. В первом случае один из взаимодействующих объектов проверяет подлинность другого, во втором случае проверка является взаимной.

Механизмы постановки графика, называемые также механизмами заполнения текста, используются для засекречивания потока данных. Они основываются на генерации объектами АИТ фиктивных блоков, их шифровании и организации передачи по каналам сети. Тем самым нейтрализуется возможность получения информации посредством наблюдения за внешними характеристиками потоков, циркулирующих по каналам связи.

Механизмы управления маршрутизацией обеспечивают выбор маршрутов движения информации по коммуникационной сети таким образом, чтобы исключить передачу секретных сведений по скомпрометированным (небезопасным) физически ненадежным каналам.

Механизмы арбитража обеспечивают подтверждение характеристик данных, передаваемых между объектами АИТ, третьей стороной (арбитром). Для этого вся информация, отправляемая или получаемая объектами, проходит и через арбитра, что позволяет ему впоследствии подтверждать упомянутые характеристики.

В АИТ при организации безопасности данных используется комбинация нескольких механизмов.

Виды защиты в информационных системах маркетинговой деятельности.

В практической маркетинговой деятельности применение мер и средств защиты информации включает следующие самостоятельные направления:

- защиту информации от несанкционированного доступа;
- защиту информации в системах связи;

- защиту юридической значимости электронных документов;
- защиту конфиденциальной информации от утечки по каналам побочных электромагнитных излучений и наводок;
- защиту информации от компьютерных вирусов и других опасных воздействий по каналам распространения программ;
- защиту от несанкционированного копирования и распространения программ и ценной компьютерной информации.

Для каждого направления определяются основные цели и задачи.

Под несанкционированным доступом понимается нарушение установленных правил разграничения доступа, последовавшее в результате случайных или преднамеренных действий пользователей либо других субъектов системы разграничения, являющейся составной частью системы защиты информации.

Субъекты, совершившие несанкционированный доступ к информации, называются нарушителями. С точки зрения защиты информации несанкционированный доступ может иметь следующие последствия: утечку обрабатываемой конфиденциальной информации, ее искажение или разрушение в результате умышленного нарушения работоспособности АИТ.

Нарушителями могут быть:

- штатные пользователи АИТ;
- сотрудники-программисты, сопровождающие системное, общее и прикладное программное обеспечение системы;
- обслуживающий персонал (инженеры);
- другие сотрудники, имеющие санкционированный доступ к АИТ (в том числе подсобные рабочие, уборщицы и т.д.).

Доступ к АИТ посторонних лиц, не принадлежащих к указанным категориям, исключается организационно-режимными мерами.

Под каналом несанкционированного доступа к информации понимается последовательность действий лиц и выполняемых ими технологических процедур, которые либо выполняются несанкционированно, либо обрабатываются неправильно в результате ошибок персонала и/или сбое оборудования, приводящих в конечном итоге к факту несанкционированного доступа. Выявление всех каналов несанкционированного доступа проводится в ходе проектирования путем анализа технологии хранения, передачи и обработки информации, определенного порядка проведения работ, разработанной системы защиты информации и выбранной модели нарушителя.

Защита конфиденциальной и ценной информации от несанкционированного доступа и модификации призвана обеспечить решение одной из наиболее важных задач защиты имущественных прав владельцев и пользователей ПЭВМ – защиту собственности, воплощенную в

обрабатываемой с помощью ПЭВМ информации от всевозможных злоумышленных покушений, которые могут нанести существенный экономический и другой материальный и нематериальный ущерб.

Центральной проблемой защиты информации от несанкционированного доступа является разграничение функциональных полномочий и доступа к информации, направленное не только на предотвращение возможности потенциального нарушителя «читать» хранящуюся в ПЭВМ информацию, но и на предотвращение возможности нарушителя модифицировать ее штатными и нештатными средствами.

Требования по защите информации от несанкционированного доступа направлены на достижение (в определенном сочетании) трех основных свойств защищаемой информации:

- конфиденциальность (засекреченная информация должна быть доступна только тому, кому она предназначена);
- целостность (информация, на основе которой принимаются важные решения, должна быть достоверной, точной и защищена от возможных непреднамеренных и злоумышленных искажений);
- готовность (информация и соответствующие информационные службы должны быть доступны, готовы к обслуживанию всегда, когда в них возникает необходимость).

В основе контроля доступа к данным лежит система разграничения доступа между пользователями АИТ и информацией, обрабатываемой системой. Для успешного функционирования любой системы разграничения доступа необходимо решение следующих задач:

- невозможность обхода системы разграничения доступа действиями, находящимися в рамках выбранной модели;
- гарантированная идентификация пользователя, осуществляющего доступ к данным (аутентификация пользователя).

Одним из эффективных методов увеличения безопасности АИТ является регистрация. Система регистрации и учета, ответственная за ведение регистрационного журнала, позволяет проследить за тем, что происходило в прошлом, и соответственно перекрыть каналы утечки информации. В регистрационном журнале фиксируются все осуществленные или неосуществленные попытки доступа к данным или программам. Содержание регистрационного журнала может анализироваться как периодически, так и непрерывно.

В регистрационном журнале ведется список всех контролируемых запросов, осуществляемых пользователями системы.

Система регистрации и учета осуществляет:

регистрацию входа-выхода субъектов доступа в систему/из системы либо регистрацию загрузки и инициализации операционной системы и ее программного останова (регистрация выхода из системы или останов не проводится в моменты аппаратного отключения АИТ), причем в параметрах регистрации указывается:

время и дата входа-выхода субъекта доступа в систему/из системы или загрузки/останова системы; результат попытки входа – успешный или неуспешный (при попытке несанкционированного доступа), идентификатор (код или фамилия) субъекта, предъявляемый при попытке доступа;

регистрацию и учет выдачи печатных (графических) документов на «твердую копию»;

регистрацию запуска/завершения программ и процессов (заданий, задач), предназначенных для обработки защищаемых файлов;

регистрацию попыток доступа программных средств (программ, процессов, задач, заданий) к защищаемым файлам;

учет всех защищаемых носителей информации с помощью их любой маркировки. Учет защищаемых носителей должен проводиться в журнале (картотеке) с регистрацией их выдачи/приема; должно проводиться несколько видов учета (дублирующих) защищаемых носителей информации.

Защита информации в системах связи направлена на предотвращение возможности несанкционированного доступа к конфиденциальной и ценной информации, циркулирующей по каналам связи различных видов. Данный вид защиты преследует достижение тех же целей: обеспечение конфиденциальности и целостности информации. Наиболее эффективным средством защиты информации в неконтролируемых каналах связи является применение криптографии и специальных связных протоколов.

Защита юридической значимости электронных документов оказывается необходимой при использовании систем и сетей для обработки, хранения и передачи информационных объектов, содержащих приказы, платежные поручения, контракты и другие распорядительные, договорные, финансовые документы. Их общая особенность заключается в том, что в случае возникновения споров (в том числе и судебных) должна быть обеспечена возможность доказательства истинности факта того, что автор действительно фиксировал акт своего волеизъявления в отчуждаемом электронном документе. Для решения данной проблемы используются современные криптографические методы проверки подлинности информационных объектов, связанные с применением так называемых «цифровых подписей». На практике вопросы защиты значимости электронных документов решаются совместно с вопросами защиты компьютерных информационных систем.

Защита информации от утечки по каналам побочных электромагнитных излучений и наводок – важный аспект защиты конфиденциальной и секретной информации в ПЭВМ от несанкционированного доступа со стороны посторонних лиц. Данный вид защиты направлен на предотвращение возможности утечки информативных электромагнитных сигналов за пределы охраняемой территории. При этом предполагается, что внутри охраняемой территории применяются эффективные режимные меры, исключающие возможность бесконтрольного использования специальной аппаратуры перехвата, регистрации и отображения электромагнитных сигналов. Для защиты от побочных электромагнитных излучений и наводок широко применяется экранирование помещений, предназначенных для размещения средств вычислительной техники, а также технические меры, позволяющие снизить интенсивность информативных излучений самого оборудования (ПЭВМ и средств связи).

В некоторых ответственных случаях может потребоваться дополнительная проверка вычислительного оборудования на предмет возможного выявления специальных закладных устройств финансового шпионажа, которые могут быть внедрены с целью регистрации или записи информативных излучений компьютера, а также речевых и других несущих уязвимую информацию сигналов.

Защита информации от компьютерных вирусов и других опасных воздействий по каналам распространения программ приобрела за последнее время особую актуальность. Масштабы реальных проявлений «вирусных эпидемий» оцениваются сотнями тысяч случаев «заражения» персональных компьютеров. Хотя некоторые из вирусных программ оказываются вполне безвредными, многие из них имеют разрушительный характер. Особенно опасны вирусы для компьютеров, входящих в состав однородных локальных вычислительных сетей. Некоторые особенности современных компьютерных информационных систем создают благоприятные условия для распространения вирусов. К ним, в частности, относятся:

необходимость совместного использования программного обеспечения многими пользователями;

трудность ограничения в использовании программ;

ненадежность существующих механизмов защиты;

разграничения доступа к информации в отношении противодействия вирусу и т.д.

Имеют место два направления в методах защиты от вирусов:

применение «иммуностойких» программных средств, защищенных от возможности несанкционированной модификации (разграничение доступа, методы самоконтроля и самовосстановления);

использование специальных программ-анализаторов, осуществляющих постоянный контроль за возникновением отклонений в деятельности прикладных программ, периодическую проверку наличия других возможных следов вирусной активности (например, обнаружение нарушений целостности программного обеспечения), а также входной контроль новых программ перед их использованием (по характерным признакам наличия в их теле вирусных образований).

Защита от несанкционированного копирования и распространения программ и ценной компьютерной информации является самостоятельным видом защиты имущественных прав, ориентированной на охрану интеллектуальной собственности, воплощенной в виде программ ПЭВМ и ценных баз данных. Данная защита обычно осуществляется с помощью специальных программных средств, подвергающих защищаемые программы и базы данных предварительной обработке (вставка парольной защиты, проверки по обращению к устройствам хранения ключа и ключевым дискетам, блокировка отладочных прерываний, проверка рабочей ПЭВМ по ее уникальным характеристикам и т.д.), которая приводит исполнимый код защищаемой программы и базы данных в состояние, препятствующее его выполнению на «чужих» машинах.

Для повышения защищенности применяются дополнительные аппаратные блоки (ключи), подключаемые к разъему принтера или к системной шине ПЭВМ, а также шифрование файлов, содержащих исполнимый код программы. Общим свойством средств защиты программ от несанкционированного копирования является ограниченная стойкость такой защиты, ибо, в конечном счете, исполнимый код программы поступает на выполнение в центральный процессор в открытом виде и может быть прослежен с помощью аппаратных отладчиков. Однако это обстоятельство не снижает потребительские свойства средств защиты до нуля, так как основная цель их применения – максимально затруднить, хотя бы временно, возможность несанкционированного копирования ценной информации.

Контроль целостности программного обеспечения проводится с помощью внешних средств (программ контроля целостности) и внутренних средств (встроенных в саму программу).

Контроль целостности программ внешними средствами выполняется при старте системы и состоит в сравнении контрольных сумм отдельных блоков программ с их эталонными суммами. Контроль целостности программ внутренними средствами выполняется при каждом запуске программы на выполнение и состоит в сравнении контрольных сумм отдельных блоков программ с их эталонными суммами. Такой контроль используется в программах для внутреннего пользования.

Одним из потенциальных каналов несанкционированного доступа к информации является несанкционированное искажение прикладных и специальных программ нарушителем с целью получения конфиденциальной информации. Подобные действия могут преследовать цель изменения правил разграничения доступа или обхода их (при внедрении в прикладные программы системы защиты) либо организацию незаметного канала получения конфиденциальной информации непосредственно из ; прикладных программ (при внедрении в прикладные программы). Одним из методов противодействия этому является контроль целостности базового программного обеспечения, осуществляемый с помощью специальных программ. Однако данный метод недостаточен, поскольку предполагает, что программы контроля целостности не могут быть подвергнуты модификации нарушителем.

При защите коммерческой информации, как правило, используются любые существующие средства и системы защиты данных от несанкционированного доступа, но в каждом случае следует реально оценивать важность защищаемой информации и ущерб, который может нанести ее утрата.

Чем выше уровень защиты, тем она дороже. Сокращение затрат идет в направлении стандартизации технических средств. В ряде случаев исходя из конкретных целей и условий рекомендуется применять типовые средства, прошедшие аттестацию, даже если они уступают индивидуально изготовленным по некоторым параметрам.

Защита информации может решаться разными методами, но наибольшей надежностью и эффективностью обладают (а для каналов связи являются единственно целесообразными) системы и средства, построенные на базе криптографических методов. В случае использования иных методов большую сложность составляет доказательство достаточности реализованных мер и обоснование надежности системы защиты от несанкционированного доступа.

Необходимо иметь в виду, что подлежащие защите сведения могут быть получены «противником» не только за счет проникновения в ЭВМ, которые с достаточной степенью надежности могут быть предотвращены (например, все данные хранятся только в зашифрованном виде), но и за счет побочных электромагнитных излучений и наводок на цепи питания и заземления ЭВМ, а также проникновения в каналы связи. Все без исключения электронные устройства, блоки и узлы ЭВМ в той или иной мере излучают побочные сигналы, причем они могут быть достаточно мощными и распространяться на расстояния от нескольких метров до нескольких километров. Наибольшую опасность представляет получение «противником» информации о ключах. Восстановив ключ, можно предпринять успешные действия по завладению зашифрованными данными, которые, как правило, оберегаются менее серьезно, чем соответствующая открытая информация. С этой точки зрения выгодно

отличаются именно аппаратные и программно-аппаратные средства защиты от несанкционированного доступа, для которых побочные сигналы о ключевой информации существенно ниже, чем для чисто программных реализаций.

Итак, определяющим фактором при выборе и использовании средств защиты является надежность защиты.

Лекция 11

Электронно-цифровая подпись

Основные вопросы

1. Понятие электронно-цифровой подписи.
2. Задачи электронно-цифровой подписи.
3. Схемы применения ЭЦП.
4. Юридически значимые системы электронного документооборота.
5. Инфраструктура открытых ключей.
6. Управление внедрением СЭД с ЭЦП.

Цель. Изучение возможностей электронно-цифровой подписи в системах электронного документооборота.

В настоящее время электронно-цифровая подпись (ЭЦП) получает все большее распространение в корпоративных информационных системах.

Функционирование системы электронного документооборота (СЭД) с ЭЦП обеспечивают четыре компонента: программно-аппаратный комплекс СЭД, имеющий в составе модуль “ЭЦП и шифрование”, программно-аппаратные комплексы службы штампов времени и удостоверяющего центра, а также несколько криптопровайдеров.

По общему мнению, собственноручная подпись на бумажном документе решает следующие задачи:

- убедить читателя в том, что человек, подписавший документ, сделал это сознательно (*подпись достоверна*);
- доказать, что именно этот человек, и никто другой, сознательно подписал документ (*подпись неподдельна*);
- будучи частью документа, защитить ее от мошеннического переноса в другой документ (*подпись невозможно использовать повторно*);
- защитить и сам документ (*подписанный документ невозможно изменить*);

- обеспечить материальность подписи и документа, гарантирующую, что человек, подписавший документ, не сможет утверждать впоследствии, что документ подписан не им (*от подписи нельзя отказаться*).

Однако, как показывает практика, собственноручная подпись на бумажном документе по самой своей природе оставляет лазейки для мошенников. Недаром для затруднения их действий на бланки документов наносят специальные защитные знаки, применяют нумерацию и скрепление листов, а кроме того, наряду с самой подписью используют собственноручное написание фамилии, имени, отчества на документе и т. п. Одним словом, при всех ее достоинствах собственноручная подпись обладает и целым рядом недостатков.

Как результат проникновения компьютерных технологий во все сферы человеческой деятельности возникла потребность реализовать аналог собственноручной подписи человека в электронном виде. Эта задача была успешно решена. В основе решения лежат разработанные в середине 1970-х гг. криптографические алгоритмы с открытым ключом, которые базируются на сложном математическом аппарате.

При этом ЭЦП устранила большинство проблем, свойственных подписи на бумажном документе, и обеспечила электронному документу следующие важнейшие характеристики:

- *подлинность* - подтверждение авторства документа;
- *целостность* - документ не может быть изменен после подписания;
- *неотрицание авторства (неотрекаемость)* - автор впоследствии не сможет отказаться от своей подписи.

Наиболее широкое применение сегодня ЭЦП находит в документационном обеспечении управления (ДОУ), в платежных системах, электронной торговле и бухгалтерии. Из перечисленных направлений наиболее востребованной и сложной является задача автоматизации ДОУ организаций - главная цель создания систем электронного документооборота (СЭД). Именно на нем мы и сосредоточим свое внимание в статье. Однако прежде необходимо уточнить, что понимается под использованием ЭЦП, имея в виду две основные его схемы:

- *подписание электронного сообщения при его передаче и проверка подписи отправителя после получения, то есть защищенная передача документа*. Часто подобную схему воспринимают как юридически значимый документооборот, что является глубоким заблуждением. Защита электронного сообщения посредством ЭЦП - вещь, безусловно, полезная и нужная, но для обеспечения полноценного документооборота совершенно недостаточная;

- *использование ЭЦП во всем жизненном цикле электронного документа - при его создании, согласовании, утверждении, ознакомлении с ним и т. д.* Только в том случае, когда автоматизируется полный жизненный цикл документа и ЭЦП является его неотъемлемой частью, можно говорить об

использовании полноценной, т. е. юридически значимой системы электронного документооборота.

Начиная работать с ЭЦП, пользователь делает запрос средствами СЭД на получение ключевой пары; затем запрашивает сертификат для только что сгенерированного открытого ключа; при необходимости подписать документ (только что созданный или полученный из СЭД) его данные передаются в криптопровайдер для генерации ЭЦП, после чего на ЭЦП ставится штамп времени; после получения документа из СЭД получатель передает его для проверки в криптопровайдер, где проверяется не только сама ЭЦП, но также сертификат и штамп времени.

Далее будем рассматривать юридически значимые СЭД. Такие системы наиболее востребованы в крупных холдингах, государственных структурах управления, кредитных учреждениях, биржах, страховых компаниях - там, где необходимо в электронном виде документировать принятые решения и нести материальную ответственность в связи с ними.

Электронный документ – это документ, подготовленный с использованием системы электронного документооборота, зафиксированный на материальном носителе в виде объекта СЭД и снабженный реквизитами, с помощью которых можно идентифицировать место, время создания и автора документа.

Основная отличительная черта СЭД с поддержкой ЭЦП от СЭД без таковой поддержки состоит в том, что электронные документы, снабженные ЭЦП, являются доказательствами: они документируют решение или какой-либо факт. Если при возникновении конфликтной ситуации существует электронный документ, подписанный ЭЦП, то на его основе можно провести расследование внутри организации, а при необходимости - и с привлечением третьей стороны (например, в арбитражном суде). СЭД, не предусматривающие ЭЦП, такой возможности не предоставляют.

Пользователи СЭД без ЭЦП вынуждены доверять системе, системным администраторам, другим участникам работы с документами, причем без каких-либо веских на то оснований. При наличии же ЭЦП основания для доверия есть - это криптографические алгоритмы и протоколы.

Доказательность электронных документов имеет два важных следствия:

- возникает возможность полностью отказаться от бумажных документов при условии, что это не противоречит действующему законодательству (некоторые типы документов требуется иметь в бумажном виде). Это позволяет избежать дублирования информации на различных носителях, обеспечивает надежное хранение данных и предотвращает утечку конфиденциальной информации;

- отпадает надобность в физической передаче сотрудникам бумажных документов, что многократно ускоряет процессы принятия решений по документам и доведения решений руководства до сотрудников.

Принятие закона "Об электронно-цифровой подписи" стало точкой отсчета для введения в оборот термина "юридически значимый электронный документооборот". Сейчас этот термин трактуется весьма широко, что часто вызывает недоразумения во взаимоотношениях заказчиков и разработчиков СЭД. Чтобы лучше понять этот термин, необходимо развеять некоторые заблуждения, касающиеся применения ЭЦП в СЭД и обеспечения юридической значимости документа:

по сути электронный документ без ЭЦП существовать не может. Если ЭЦП не применяется, то можно говорить лишь об электронном образе документа и не более того. Никакие графические образы подписи на документе (например, отсканированный бумажный оригинал) не могут рассматриваться как аналог собственноручной подписи. Более того, подобная практика вредна, поскольку создает иллюзию защищенности системы и ее участников;

важно понимать, что ЭЦП не обеспечивает конфиденциальность электронного документа. Эту задачу решает шифрование, которое, в свою очередь, никакого отношения к обеспечению юридической значимости документа не имеет. В случае совместного использования ЭЦП и шифрования нужно учитывать, что для того, чтобы ЭЦП была юридически значимой, пользователь должен видеть и понимать, что он подписывает. Поэтому необходимо вначале создать ЭЦП, а уж затем зашифровать документ, который перед проверкой подписи должен быть расшифрован;

с определенностью можно констатировать, что на сегодняшний день нормативно-правовая и техническая база для реализации юридически значимого электронного документооборота в масштабах государства в нашей стране еще не создана. Тем не менее, даже в рамках имеющегося законодательства и технической базы можно реализовать внутренний юридически значимый электронный документооборот в работе отдельно взятой организации или нескольких компаний, входящих в одну структуру, - корпоративный электронный документооборот.

Уже сегодня многие внутренние документы организации можно перевести в электронный вид (например, служебные записки, заявки на выделение денежных средств, различные внутренние отчеты, поручения и т. п.). Необходимо разработать нормативно-правовую базу организации, регламентирующую применение ЭЦП. Такой регламент обеспечит электронным документам юридическую силу - возможность представлять их в суде в качестве доказательства. Безусловно, небольшая правоприменительная практика вносит некоторые ограничения в применение ЭЦП, но не является

принципиальным барьером для построения в отдельной компании внутреннего юридически значимого электронного документооборота.

Как следует из вышеизложенного, ЭЦП - это аналог собственноручной подписи человека, применяемый в электронных документах. Электронно-цифровая подпись создается с помощью закрытого ключа - уникальной последовательности символов, которая известна его владельцу и предназначена для создания ЭЦП в электронных документах с использованием соответствующих средств.

Получатели электронного документа, подписанного ЭЦП, имеют возможность проверить действительность подписи посредством открытого ключа и убедиться в том, что документ подлинный, а ЭЦП принадлежит именно тому лицу, которое в нем указано. Открытый ключ - это уникальная последовательность символов, которая математически связана с закрытым ключом ЭЦП. Открытый и закрытый ключи образуют так называемую ключевую пару.

Открытый ключ доступен любому пользователю информационной системы в составе сертификата ключа. Сертификат ключа является аналогом документа, удостоверяющего личность (например, паспорта). Это документ на бумажном носителе либо электронный документ с ЭЦП уполномоченного лица (сотрудника) удостоверяющего центра. Сертификат ключа помимо открытого ключа ЭЦП содержит идентификационные данные владельца. Сертификат передается пользователю СЭД и выполняет две задачи: подтверждает подлинность ЭЦП и идентифицирует владельца сертификата ключа подписи.

И в том и в другом случае используются средства электронно-цифровой подписи - программно-аппаратный комплекс, обеспечивающий реализацию хотя бы одной из следующих функций: создание ЭЦП в электронном документе с использованием закрытого ключа ЭЦП; подтверждение с использованием открытого ключа ЭЦП подлинности ЭЦП в электронном документе; создание закрытых и открытых ключей ЭЦП.

Аналогом третейского судьи, которому доверяют все участники документооборота, является удостоверяющий центр - организационная структура, осуществляющая деятельность по управлению сертификатами ключей и поддержке их использования в различных подсистемах корпоративной информационной системы. Удостоверяющий центр может быть внешней организацией или подразделением той или иной компании.

Еще один участник процесса - криптопровайдер. Это программный или аппаратно-программный модуль, реализующий один или несколько криптографических алгоритмов и предоставляющий свои функции внешним системам.

Аналогом даты на бумажном документе, которую собственноручно проставляет лицо, подписывающее документ, является штамп времени. Речь идет о свидетельстве третьей доверенной стороны - организационной единицы, носящей название службы штампов времени. СЭД передает туда так называемое хеш-сообщение, которое получается в результате криптографического преобразования документа. На это сообщение служба ставит штамп (средствами своего программно-аппаратного обеспечения), удостоверяющий, что электронный документ существовал на данный момент времени. В результате к хеш-сообщению добавляется значение, указывающее, когда службой штампов времени был получен запрос на проставление штампа времени. Проставляемое значение служба штампов времени подписывает собственной ЭЦП и возвращает документ обратно в СЭД.

Совокупность аппаратно-программного обеспечения, а также персонала, политик и процедур, необходимых для создания, хранения, распределения, управления жизненным циклом и использования сертификатов открытых и связанных закрытых ключей, называется инфраструктурой открытых ключей (ИОК).

Выбирая СЭД с поддержкой ЭЦП, следует обратить внимание на особенности реализации выбранной системы. Рассмотрим ключевые аспекты, которые необходимо учитывать.

Не только содержание, но и форма. Во многих СЭД в качестве электронного документа рассматривается файл какого-либо типа (например, Microsoft Word или Adobe Acrobat), прилагаемый к регистрационной карточке. Хотелось бы обратить внимание на одно обстоятельство: подписание просто файлов (содержательной части документов) не представляет большого интереса для организации. Строго говоря, не вся информация в документе является "неструктурированной", документ кроме содержательной части содержит реквизиты, которые можно и нужно выделять в отдельную структуру, чтобы в дальнейшем осуществлять по ним поиск и классификацию документов. Во многих ситуациях полезно подписывать не только содержание, но и форму. В этом случае можно отображать на экране компьютера и распечатывать электронный документ в том виде, в котором он был подписан, что позволит избежать всяческих конфликтных ситуаций.

Все нюансы бумажного документооборота. ЭЦП должна быть равнозначна собственноручной подписи и учитывать все нюансы бумажного делопроизводства. А именно - необходимо, чтобы СЭД позволяла подписать часть документа, поставить подпись в электронном документе последовательно (подписывается документ и все имеющиеся ЭЦП), параллельно (подписывается документ и все ЭЦП нижележащих уровней).

Подпись "заверяю" - последовательная подпись первого уровня - охватывает только содержательную часть документа (это подпись автора документа). Подписи "согласовано 1" и "согласовано 2" (визы согласующих) - это параллельные подписи второго уровня. Они охватывают содержательную часть документа и подпись первого уровня и при этом не зависят друг от друга. Подпись "утверждаю" (виза руководителя) - последовательная подпись третьего уровня - охватывает содержательную часть документа и все предыдущие подписи.

Формат электронного документа. Для полноценной реализации ЭЦП система электронного документооборота должна поддерживать формат документа, являющийся канонической формой, к которой будет приводиться любой "электронный документ" в СЭД. С точки зрения удобства работы и перспективы развития и интеграции предпочтительны СЭД, которые для описания формата документа используют язык XML. В настоящий момент международными организациями ведется работа по созданию стандарта формата электронного документа.

Штамп времени. Важно отметить, что при работе с ЭЦП неизбежно возникает проблема, обусловленная тем, что срок действия любого сертификата ограничен определенным промежутком времени. По истечении срока действия сертификата все созданные при его помощи ЭЦП теряют свое значение, поскольку невозможно определить, была ли ЭЦП создана, когда сертификат еще действовал, или когда срок его действия уже закончился. А это автоматически означает недействительность ЭЦП.

Поэтому внимания заслуживают лишь СЭД, интегрированные со службой штампов времени, которая позволяет в один из атрибутов системы помещать штамп, фиксирующий момент создания ЭЦП. При таком решении имеется возможность проверять ЭЦП с учетом того, действовал ли сертификат в момент создания этой ЭЦП, а не в момент проверки.

Однако и этот штамп заверяется ЭЦП службы штампов времени, сертификат которой также имеет ограниченный срок действия. Для нивелирования данной проблемы существует стандартизированная методика, которая должна быть реализована в СЭД. Ее суть заключается в том, что, когда срок действия сертификата службы штампов времени подходит к концу, СЭД запрашивает для старого сертификата и набора служебной информации штамп времени с ЭЦП на новом сертификате, срок действия которого только начинается. Таким образом, благодаря действительности нового сертификата можно доказать, что до момента заверения действовал и старый сертификат.

Архивная копия электронного документа. СЭД должна позволять участнику системы получить архивную копию подписанного электронного документа, которая может быть представлена в качестве доказательства в

случае возникновения конфликтной ситуации. Разумеется, необходимо учитывать ограничения политики безопасности, касающиеся конфиденциальных документов.

Создание ЭЦП под документом должно выполняться пользователем осознанно. Не допускается подписывать документ в автоматическом режиме. Система обязательно должна задать вопрос, будет ли пользователь подписывать документ. Во многих существующих СЭД этому не уделяется должное внимание. Более того, некоторые разработчики, увлекаясь автоматизацией, специально реализуют автоматическую простановку ЭЦП под документом, что является абсолютно неверным подходом.

Организационные факторы внедрения СЭД с ЭЦП

- Понимание того, для чего необходима ЭЦП и какие выгоды можно получить от ее применения.
- Политика информационной безопасности.
- Регламент бумажного делопроизводства.
- Нормативно-правовая база применения ЭЦП в СЭД.
- Квалифицированные специалисты в области информационной безопасности.

Технические факторы внедрения СЭД с ЭЦП

- В организации развернута инфраструктура открытых ключей либо имеется договор о предоставлении услуг удостоверяющего центра сторонней организацией.
- Развернута служба штампов времени.
- Обеспечивается доверенная среда выполнения программного обеспечения СЭД на рабочих местах пользователей и серверах.
- На рабочих местах развернуты необходимые средства криптографической защиты.

Отдельный вопрос - делегирование должностных полномочий одного пользователя системы другому. Очень часто руководители передают свое право подписания электронного документа доверенному лицу, при этом параллельно подписывают бумажный экземпляр документа, который в данном случае и является оригиналом. Вопрос не простой. Следует получить квалифицированную консультацию у юриста, в каких случаях делегирование допустимо и не противоречит действующему законодательству и каким образом этот факт должен быть оформлен документально. Просто передавать другому сотруднику свой закрытый ключ для создания ЭЦП недопустимо, поскольку такая ситуация трактуется как компрометация ключа, а следовательно, полученная под документом ЭЦП не является легитимной.

Если говорить о реализации СЭД, то технически делегирование реализуется как выпуск удостоверяющим центром специальных сертификатов,

имеющих ограниченное применение (указывается в сведениях об отношениях) и срок действия; при этом в реквизитах ЭЦП указывается, от кого делегированы данные полномочия. Для многих читателей термин "сведения об отношениях" почти наверняка покажется странным. Этим термином обозначается свойство сертификата, позволяющее ограничить область его применения. Например, сотрудник имеет право поставить подпись под служебной запиской, но не имеет права подписать финансовый документ.

Бизнес-логика СЭД должна различать сертификаты, выпущенные для делегирования полномочий. Проверьте, чтобы эти возможности поддерживала СЭД и обеспечивающая ее инфраструктура.

При внедрении СЭД рекомендуется заранее проработать все вопросы, так или иначе затрагивающие создание в организации системы защищенного юридически значимого электронного документооборота.

Регламент. Для обеспечения юридической значимости электронных документов в организации должен быть разработан и утвержден регламент применения ЭЦП. Разработку регламента следует поручить квалифицированным специалистам и привлечь (в качестве консультантов) юристов. При этом необходимо доработать внутреннюю нормативно-правовую базу компании, с тем чтобы регламент не противоречил другим нормативным актам. Для органов государственной власти кроме перечисленного требуется, чтобы средства ЭЦП и удостоверяющий центр были сертифицированы в установленном законом порядке.

Осведомленность. Участники системы электронного документооборота должны хорошо понимать, что такое ЭЦП, как ее использовать и что она им дает. Это может быть обеспечено проведением семинаров по применению ЭЦП для руководителей подразделений. А уже затем руководители должны довести до своих подчиненных, какие выгоды получает компания и каждый сотрудник от внедрения ЭЦП.

При развертывании в организации СЭД с ЭЦП необходимо решить, какую инфраструктуру открытого ключа использовать: разворачивать ли собственную (внутренняя) или прибегнуть к услугам сторонней компании (внешняя).

Как правило, крупные компании и холдинги реализуют ИОК собственными силами, т. е. службы являются внутренними. Это решение имеет ряд преимуществ, поскольку позволяет полностью контролировать процесс функционирования ИОК. Для средних организаций экономически целесообразным может оказаться использование услуг других компаний. При этом между организацией и компанией-поставщиком заключается договор на предоставление услуг удостоверяющего центра и службы штампов времени.

Для территориально распределенных организаций может быть целесообразным использование собственной иерархической структуры

удостоверяющих центров. В этом случае удостоверяющий центр будет иметь корневой и подчиненные центры регистрации. Технически центры регистрации оснащены аппаратно-программными комплексами, реализующими функции работы с сертификатами ключей. В иерархической структуре удостоверяющего центра каждый центр регистрации (узел, к которому подключаются конечные пользователи) имеет собственный сертификат ключа, выпущенный и подписанный вышестоящим центром регистрации. При этом корневой центр регистрации использует сертификат ключа, подписанный собственным закрытым ключом (так называемый самоподписанный сертификат), так как у него нет вышестоящего центра регистрации. Установка самоподписанного сертификата корневого центра регистрации должна осуществляться по процедуре, исключающей подмену этого сертификата.

Кроме того, необходимо учитывать количество запросов, поступающих в удостоверяющий центр в единицу времени, и рассчитывать производительность сетевого оборудования, с тем чтобы обеспечить приемлемое время обработки запроса.

Выбор компонентов ИОК. Важным этапом является определение критериев, по которым следует выбирать компоненты ИОК. Необходимо изучить предлагаемые инфраструктуры открытых ключей: программное обеспечение удостоверяющего центра, аппаратно-программные или программные реализации систем для служб штампов времени, аппаратно-программные или программные криптопровайдеры. Выбор компонентов не слишком широк, и получить профессиональную консультацию вполне возможно. При этом необходимо иметь в виду, что имеющиеся аппаратно-программные комплексы импортного производства не обеспечивают работу с российскими криптографическими алгоритмами. Поддержка этих алгоритмов будет нужна, если возникнет потребность в сертификации СЭД.

Из практики внедрения СЭД следует отметить, что наиболее остро встает вопрос организации развертывания службы штампов времени, состоящей из двух частей: сервера штампов времени и доверенного источника времени. Конкретная реализация службы зависит от потребности организации в длительном хранении электронных документов, подписанных ЭЦП. Как правило, сервер штампов времени, с которым необходима периодическая синхронизация, является внешним по отношению к организации и, таким образом, становится "точкой отказа". Необходимо проработать вопрос обеспечения доступа к резервному серверу доверенного времени. На данный момент вопрос с доверенным источником времени на уровне государства никак не решен.

Выбор СЭД. Определившись с инфраструктурой, нужно потратить значительные усилия на выбор СЭД, поскольку предложений очень много.

Принципиальные требования к СЭД уже обсуждались выше. Здесь отметим следующее.

Разработчики подавляющего большинства СЭД, представленных на рынке, заявляют, что поддерживают юридически значимый электронный документооборот, однако термин этот у каждого трактуется по-своему. Например, ЭЦП проставляется без использования штампов времени, но документ объявляется юридически значимым. Как правило, заявление о поддержке юридически значимого документооборота является преувеличением, поскольку многие важные вопросы использования ЭЦП остаются неразрешенными. В этой связи следует критически относиться к подобным заявлениям и уточнять: что же понимает под юридически значимым документооборотом конкретный производитель СЭД.

Желательно выбрать поставщика СЭД, имеющего опыт разработки регламента применения ЭЦП.

Во многих случаях в СЭД отсутствуют средства для разбора конфликтных ситуаций, связанных с действительностью ЭЦП или отрицанием авторства, что технически не позволяет представить доказательства в суде. В СЭД должна существовать четкая процедура разбора конфликтных ситуаций и представления доказательств третьей стороне.

Не исключено, что будет принято решение заказать разработку системы или существенно доработать существующую в организации СЭД. Последний подход не многим отличается по трудоемкости от заказной разработки, но значительно уступает в эффективности решения.

Управление внедрением. Внедрению должна предшествовать разработка стратегии автоматизации делопроизводства организации. В этом документе следует четко сформулировать цели внедрения СЭД, определить принципы построения и этапы внедрения СЭД в организации. Развертывание ЭЦП является составной частью процесса внедрения СЭД, и рассматривать этот процесс как независимый не рекомендуется. Внедрение же СЭД является ответственным делом, и подходить к нему следует с особой тщательностью.

При внедрении СЭД очень важна разработка нормативно-правовой базы. Этот процесс выполняется поэтапно (параллельно с внедрением самой системы) с учетом особенностей делопроизводства в организации и выбранной СЭД.

Как показывает практика, СЭД с ЭЦП можно внедрить безболезненно, если ЭЦП является составной частью архитектуры системы. Компания получит безусловную выгоду, если ЭЦП частично внедрена в СЭД, например в таких подсистемах, как "ведение договоров", "заявки на денежные средства" и т. п., но максимальную выгоду от применения СЭД можно извлечь только при полномасштабном развертывании ЭЦП.

Доверие к инфраструктуре. Обеспечение доверия к внешнему окружению СЭД является ключевым моментом развертывания ЭЦП в организации. Особое внимание следует обратить на доверенную (т. е. исключющую подмену) установку на рабочем месте пользователя самоподписанного сертификата корневого центра регистрации.

Желательно также обеспечить доверенную среду, в которой будет работать СЭД. Для получения полностью доверенной среды необходимо использовать ряд аппаратно-программных компонентов, обеспечивающих (возможно их поэтапное подключение):

- доверенную загрузку операционной системы, например с использованием "электронного замка";
- регулярное обновление антивирусного ПО (как на серверах организации, так и на рабочих станциях);
- централизованную установку ПО на рабочих местах пользователя.

Необходимо также получить гарантии от разработчиков СЭД на отсутствие недокументированных функций в системе.

Если все эти вопросы тщательно проработаны, то организационные и технические сложности процесса внедрения ЭЦП будут минимальными.

Изначально СЭД проектировались без учета применения ЭЦП, они моделировали работу с бумажными документами, от которых организации не собирались отказываться. По мере осознания того, что ЭЦП использовать нужно, разработчики стали встраивать в существующие СЭД функции ЭЦП, рассматривая их как дополнительные. Однако в результате получались решения с ограниченными возможностями, поскольку полноценное встраивание ЭЦП требовало слишком больших переделок в существующих системах.

Сегодня, на очередном витке развития информационных технологий, разработчики вновь создаваемых СЭД уже не рассматривают ЭЦП как некое дополнение и учитывают необходимость ее применения уже на этапе разработки архитектуры системы.

Мировой опыт развития СЭД показывает, что перспективы применения ЭЦП в электронном документообороте и смежных областях весьма впечатляющи. Наблюдается бурное развитие технологий потокового сканирования и распознавания графических образов, что позволяет перевести практически любые бумажные документы в электронный вид и обеспечить эффективный полнотекстовый поиск по ним. Развивается ИОК. В сочетании с общей тенденцией к ускорению принятия решений по документам и потребностью именно в юридически значимых электронных документах это приводит к тому, что электронная цифровая подпись становится востребованной, как никогда ранее.

Тема 6. Планирование, проведение и обработка результатов научных исследований

Лекция 1. (2 часа)

Табличные процессоры и системы математических вычислений. Программные средства автоматизации проектирования

Основные вопросы

Электронные таблицы.

Назначение, возможности электронных таблиц.

Системы математических вычислений.

Назначение и возможности систем математических вычислений.

Программные средства автоматизированного проектирования.

Цель: Ознакомиться с табличными процессорами, системами математических вычислений, программными средствами автоматизации проектирования.

Электронные таблицы – это двумерные массивы, состоящие из столбцов и строк. С помощью электронных таблиц можно выполнять различные экономические, бухгалтерские и инженерные расчеты, а также строить разного рода диаграммы, проводить сложный экономический анализ, моделировать и оптимизировать решение различных хозяйственных ситуаций и т. д.

Основные возможности табличных процессоров:

- создание и редактирование электронных таблиц;
- создание многотабличных документов;
- оформление и печать электронных таблиц;
- построение различных типов диаграмм и графиков;
- проведение однотипных расчетов над большими наборами данных;
- работа с электронными таблицами как с базами данных: сортировка данных, выборка данных;
- создание итоговых и сводных таблиц;
- использование при построении таблиц информации из внешних баз данных;
- решение задач подбора значений параметров;
- решение оптимизационных задач;
- обработка результатов экспериментов;
- решение экономических задач;
- разработка макрокоманд, настройка среды под потребности

пользователя и т.д.

Кроме этого, табличные процессоры дают возможность:

- представлять числа в таблицах в различных форматах;
- защищать клетки таблиц от несанкционированных действий;
- скрывать столбцы с данными;
- создавать командные файлы, с помощью которых можно составлять

программы на простом языке высокого уровня;

- устанавливать связи с другими программными продуктами.

Среди табличных процессоров наиболее распространенными являются различные версии Microsoft Excel, Lotus, QuattroPro, SuperCalc, OpenOffice.org Calc.

Для выполнения математических действий в традиционном для математики символьном виде были разработаны *системы математических вычислений*, позволяющие осуществлять аналитические преобразования, а результат получать не в виде одного числа, а в виде формулы.

Успехи вычислительной математики постоянно совершенствующиеся, позволяя решать любую математическую задачу применительно к любой отрасли знаний. Результат вычислений представляется одним конечным числом в арифметическом виде, то есть при помощи десятичных цифр. Иногда результат представляется множеством (массивом, матрицей) таких чисел, но существо представления от этого не меняется – результат в виде конечного десятичного арифметического числа.

Однако, подавляющее большинство результатов нетривиальных математических вычислений в классической математике традиционно записывается в символьной форме: с использованием специальных общеизвестных чисел, таких как, например, пи, с помощью радикалов и тд. Считается, что в противном случае имеет место принципиальная потеря точности.

Классический пример, выражение: $\sin^2(x) + \cos^2(x) = 1$.

В компьютерных вычислениях либо будет предпринята попытка вычислить это выражение с неизбежными ошибками округления, либо будет выдано сообщение о неопределенности аргумента x и всякие дальнейшие действия будут прекращены.

Таким образом возникла необходимость выполнять преобразования традиционными для математики способами: дробно-рациональные преобразования, подстановки, упрощения, решение уравнений, дифференцирование и т.п, что и привело к созданию *систем математических вычислений*.

Данные об особенностях существующих систем математических вычислений приведены в таблице.

Система	Назначение и возможности	Недостатки
Mathcad 13, Mathcad 14	<p>Система универсального назначения в основном для непрофессиональных математиков и целей образования всех ступеней. Продуманный интерфейс представления данных в традиционной математической форме и изумительная графика на всех этапах работы, включая ввод. Ввод с помощью выбора из панелей инструментов или из меню практически без использования клавиатуры. Мощный и исчерпывающий набор операторов и функций. Множество примеров, электронных книг и библиотек, готовых решений практических задач. Ядро символьных вычислений импортировано из СКМ Maple. Предоставление серверных услуг профессионального пакета. Легкость переноса документа в другие приложения</p>	<p>Достаточно примитивные средства программирования. Дороговизна электронных книг и библиотек, отсутствие русифицированных версий самого пакета и дополнительных библиотек (книг). Затруднена символьная обработка дифференциальных уравнений. Не создается итоговый исполняемый *.exe-файл; для запуска документа необходимо наличие пакета СКМ Mathcad. Затруднения при выполнении тригонометрических преобразований</p>
Maple V R4/R5/R6	<p>Университетское высшее образование и научные расчеты. Мощное ядро символьных вычислений – возможности аналогичны СКМ Mathcad, содержащее до 3000 функций. Мощнейшая графика. Удобная справочная система. Средства форматирования документов</p>	<p>Повышенные требования к аппаратным ресурсам. Отсутствие синтеза звуков. Ориентация на опытных пользователей и специалистов по математике. Все недостатки аналитических действий аналогичны СКМ Mathcad</p>
Mathematica 5/7	<p>Высшее образование и научные расчеты. Наиболее развитая система символьной математики. Единственная СКМ, обеспечивающая символьное решение дифференциальных уравнений. Совместимость с разными компьютерными платформами. Уникальная трехмерная графика. Поддержка синтеза звука. Развитые</p>	<p>Высокие требования к аппаратным ресурсам. Чрезмерная защита от копирования. Слабая защита от некорректных задач. Ориентация на опытных пользователей. Ввод задач на уникальном языке функционального программирования.</p>

	<p>средства форматирования документов. Программный синтез звуков.</p>	<p>Непривычная индикация функций запуска вычислений.</p>
<p>MATLAB 7.*</p>	<p>Образование (в том числе техническое), научные расчеты, численное моделирование, и расчеты, ориентированные на применение матричных методов, при этом скаляр рассматривается как матрица 1x1. Уникальные матричные средства, обилие численных методов, описательная (дескрипторная) графика, высокая скорость вычислений, легкость адаптации к задачам пользователя благодаря множеству пакетов расширения системы. Развитый язык программирования с возможностями объектно-ориентированного программирования (ООП), совместимость с алгоритмическим языком Java</p>	<p>Очень высокие требования к аппаратным ресурсам. Практически отсутствует возможность символьных вычислений. Относительно высокая стоимость. Ввод задач на уникальном языке программирования</p>

Интегрированная среда MathCad является системой универсального назначения и наиболее приспособлена для решения широкого спектра, а точнее –практически любых математических задач, в основном непрофессиональными математиками, а также для эффективного использования во всех областях сферы образования.

Все действия в MathCad сразу оформляются в виде документа, состоящего из рабочих листов, на которых помещается описание алгоритма, рабочие формулы, комментарии, иллюстрации, графики, таблицы. Форма такого документа максимально приспособлена для печати, передачи по сети Internet и не требует дополнительного редактирования. С другой стороны, этот документ, имеющий расширение .mcd, содержит в скрытом виде всю программу вычислений. Он может быть импортирован как для целей издания, так и для продолжения и совершенствования программных вычислений. Весь документ или отдельные его части могут быть заблокированы для редактирования путем задания пароля.

Встроенные в среду MathCad электронные книги (e-Books) содержат примеры, справки и типовые расчеты из различных областей науки, техники, экономики. Любой фрагмент из этих книг можно скопировать на рабочий лист

документа и выполнить.

Библиотеки и пакеты расширений, ориентированные на решение различных прикладных задач, поставляются и устанавливаются разработчиком отдельно.

Мощный интерфейс MathCad не требует программирования при вводе заданий и индикации результатов – все это выполняется в традиционной форме на общепринятом языке математических символов и формул без применения каких-либо специальных команд или операторов.

Основные возможности

Численные методы вычислений

Решение уравнений и систем уравнений, как линейных, так и нелинейных. Нахождение корней многочлена.

Решение неравенств.

Вычисление определенного интеграла.

Вычисление несобственных интегралов.

Вычисление кратных интегралов.

Численные методы дифференцирования.

Численное решение обыкновенных дифференциальных уравнений – задача Коши.

Численное решение обыкновенных дифференциальных уравнений – решение краевой задачи.

Решение дифференциальных уравнений в частных производных.

Вычисление суммы и произведения членов ряда.

Исследование функций и численное определение экстремумов функций одной и нескольких переменных, построение асимптот.

Решение оптимизационной задачи методом линейного программирования.

Символьные вычисления

1. Выполнение точных вычислений с представлением результатов в традиционной математической форме – с записью ответа в форме радикала и специальных иррациональных чисел;

Символьные преобразования математических выражений целиком или их фрагментов:

1. разложение выражений в более простые;

2. приведение подобных:

- разложение на множители;

- приведение к общему знаменателю;

- вынесение общего множителя;

- разложение на элементарные дроби;

- вычисление коэффициентов полиномов;
- выполнение подстановок.
- 3. Аналитическое решение уравнений и систем уравнений.
- 4. Дифференцирование в символьной форме определение производных любых порядков.
- 5. Аналитическое определение первообразной.
- 6. Построение касательной и нормали к плоской кривой и к поверхности.
- 7. Аналитическое вычисление определенного интеграла.
- 8. Символьное вычисление кратных интегралов.
- 9. Решение неравенств.
- 10. Аналитическое вычисление предела.
- 11. Аналитическое вычисление суммы ряда конечного или бесконечного.
- 12. Аналитическое вычисление произведения членов ряда конечного или бесконечного.
- 13. Аналитическое вычисление суммы/произведения членов ряда конечного или бесконечного, когда пределы и шаг изменения индекса члена ряда задаются (например, сложить четные числа от 10 до 10+k.)
- 14. Разложение в ряд Тейлора.
- 15. Разложение в ряд Фурье.
- 16. Символьное преобразование Фурье и Лапласа – прямое и обратное.
- 17. Операции с матрицами в символьной форме: умножение и сложение матриц, поиск обратной матрицы, вычисление определителя, поиск собственных значений и собственных векторов.

Работа с матрицами и матричные вычисления

1. Элементарные матричные действия: создание, импорт, заполнение матриц, задание матриц специального вида, умножение, сложение, транспонирование и сортировка матрицы в целом или ее фрагмента. Выполнение векторизации – однотипных действий над всеми элементами матрицы.

2. Вычисление определителя, размерности, ранга и следа матрицы, скалярное и векторное умножение векторов, вычисление якобиана, например, для перехода к другим системам координат в тройном интеграле. Вычисление собственных значений и собственных векторов, поиск максимального и минимального элемента матрицы.

3. Матричные преобразования: скалярное и векторное умножение векторов, поиск обратной матрицы и решение системы алгебраических линейных уравнений, всевозможные разложения матрицы на произведение матриц специального вида: двух треугольных – верхней и нижней (LU-преобразование), треугольной и ее же транспонированной (разложение

Холецкого), ортогональной и верхней треугольной (QR-разложение), сингулярное разложение.

4. Интегрирование среды MathCad с матричной математической системой MATLAB и возможность использования ее аппарата открывает удивительные возможности эффективного решения матричных задач неограниченной сложности.

Решение дифференциальных уравнений

Программирование

Составление программ и выполнение расчетов на упрощенном процедурном алгоритмическом языке с возможностью использования всех процедурных конструкций: условных операторов, циклов, массивов, модуль-функций, модуль-процедур.

Комплексные числа

1. Представление комплексных чисел в традиционной форме, возможность выполнения основных арифметических действия с ними.

2. Возможность автоматического получения результатов многих вычислений в виде комплексного числа (например, всех корней многочлена).

3. Возможность задания комплексного аргумента для многих библиотечных функций и получение математически корректного результата.

Обработка данных и финансовые расчеты

Теория вероятностей и математическая статистика

Математическое моделирование

Специальные возможности по прикладным инженерным и научным расчетам

1. Обработка электрических сигналов и расчет электронных устройств.

2. Виртуальная генерация электрических сигналов и их обработка.

Система автоматизированного проектирования (САПР) - это совокупность средств и методов для осуществления автоматизированного проектирования. Она состоит из нескольких составных частей, называемых техническим, математическим, программным, лингвистическим, информационным, методическим и организационным обеспечением ссылка на источники литературы.

Техническое обеспечение САПР представляет собой совокупность взаимосвязанных и взаимодействующих технических средств, предназначенных для выполнения автоматизированного проектирования. Техническое обеспечение делится на группы средств программной обработки данных, подготовки и ввода данных, средств отображения и документирования, архива проектных решений, средств передачи данных.

Средства программной обработки данных представлены процессорами и запоминающими устройствами, т.е. устройствами ЭВМ, в которых реализуются преобразования данных и программное управление вычислениями. Средства подготовки, ввода, отображения и документирования данных служат для общения человека с ЭВМ. Средства архива проектных решений представлены внешними запоминающими устройствами. Средства передачи данных используются для организации связей между территориально разнесенными ЭВМ и терминалами (оконечными пунктами).

Математическое обеспечение САПР включает в себя математические модели (ММ) проектируемых объектов, методы и алгоритмы проектных процедур, используемые при автоматизированном проектировании. Элементы математического проектирования САПР чрезвычайно разнообразны. К ним относятся принципы построения функциональных моделей, методы численного решения алгебраических и дифференциальных уравнений, постановки экстремальных задач, поиска экстремума и т.д. Специфика предметных областей проявляется прежде всего в ММ проектируемых объектов, она заметна и в способах решения задач структурного синтеза. Формы представления математического обеспечения также довольно разнообразны, но его практическое использование происходит после реализации в программном обеспечении ссылка на источники литературы.

Программное обеспечение САПР объединяет собственно программы для систем обработки данных на машинных носителях и программную документацию, необходимую для эксплуатации программы. Программное обеспечение (ПО) делится на общесистемное, базовое и прикладное (специальное). Общесистемное ПО предназначено для организации функционирования технических средств, т.е. для планирования и управления вычислительным процессом, распределения имеющихся ресурсов, и представлено операционными системами ЭВМ и ВС. Общесистемное ПО обычно создается для многих приложений и специфики САПР не отражает. Базовое и прикладное ПО создаются для нужд САПР. В базовое ПО входят программы, обеспечивающие правильное функционирование прикладных программ.

В прикладном ПО реализуется математическое обеспечение для непосредственного выполнения проектных процедур. Прикладное ПО обычно имеет форму пакетов прикладных программ (ППП), каждый из которых обслуживает определенный этап процесса проектирования или группу однотипных задач внутри различных этапов.

Информационное обеспечение САПР объединяет всевозможные данные, необходимые для выполнения автоматизированного проектирования. Эти

данные могут быть представлены в виде тех или иных документов на различных носителях, содержащих сведения справочного характера о материалах, комплектующих изделиях, типовых проектных решениях, параметрах элементов, сведения о состоянии текущих разработок в виде промежуточных и окончательных проектных решений, структур и параметров проектируемых объектов и т.п. Основная часть информационного обеспечения САПР - банк данных, представляющий собой совокупность средств для централизованного накопления и коллективного использования данных в САПР. Банк данных (БНД) состоит из базы данных и системы управления базой данных.

База данных (БД) - сами данные, находящиеся в запоминающих устройствах ЭВМ и структурированные в соответствии с принятыми в данной БД правилами. Система управления базой данных (СУБД) - совокупность программных средств, обеспечивающих функционирование БД. С помощью СУБД производятся запись данных в БД, их выборка по запросам пользователей и прикладных программ, обеспечивается защита данных от искажений, несанкционированного доступа и т.п.

Лингвистическое обеспечение САПР представлено совокупностью языков, применяемых для описания процедур автоматизированного проектирования и проектных решений. Основная часть лингвистического обеспечения - языки общения человека с ЭВМ.

Методическое обеспечение САПР составляют документы, характеризующие состав, правила отбора и эксплуатации средств автоматизированного проектирования. Допускается более широкое толкование понятия методического обеспечения, при котором под методическим обеспечением подразумевают совокупность математического, лингвистического обеспечения и названных документов, реализующих правила использования средств проектирования ссылка на источники литературы.

Организационное обеспечение САПР включает положения, инструкции, приказы, штатные расписания, квалификационные требования и другие документы, регламентирующие организационную структуру подразделений проектной организации и их взаимодействие с комплексом средств автоматизированного проектирования ссылка на источники литературы.

В области классификации САПР используется ряд устоявшихся англоязычных терминов, применяемых для классификации программных приложений и средств автоматизации САПР по отраслевому и целевому назначению.

По отраслевому назначению

MCAD (англ. mechanical computer-aided design) — автоматизированное проектирование механических устройств. Это машиностроительные САПР, применяются в автомобилестроении, судостроении, авиакосмической промышленности, производстве товаров народного потребления, включают в себя разработку деталей и сборок (механизмов) с использованием параметрического проектирования на основе конструктивных элементов, технологий поверхностного и объемного моделирования (SolidWorks, Autodesk Inventor, КОМПАС, CATIA);

EDA (англ. electronic design automation) или ECAD (англ. electronic computer-aided design) — САПР электронных устройств, радиоэлектронных средств, интегральных схем, печатных плат и т. п., (Altium Designer, OrCAD);

AEC CAD (англ. architecture, engineering and construction computer-aided design) или CAAD (англ. computer-aided architectural design) — САПР в области архитектуры и строительства. Используются для проектирования зданий, промышленных объектов, дорог, мостов и проч. (Autodesk Architectural Desktop, AutoCAD Revit Architecture Suite, Bentley MicroStation, Bentley AECOsim Building Designer, Piranesi, ArchiCAD).

По целевому назначению

По целевому назначению различают САПР или подсистемы САПР, которые обеспечивают различные аспекты проектирования.

CAD (англ. computer-aided design/drafting) — средства автоматизированного проектирования, в контексте указанной классификации термин обозначает средства САПР, предназначенные для автоматизации двумерного и/или трехмерного геометрического проектирования, создания конструкторской и/или технологической документации, и САПР общего назначения. CADD (англ. computer-aided design and drafting) — проектирование и создание чертежей.

CAGD (англ. computer-aided geometric design) — геометрическое моделирование.

CAE (англ. computer-aided engineering) — средства автоматизации инженерных расчетов, анализа и симуляции физических процессов, осуществляют динамическое моделирование, проверку и оптимизацию изделий. CAA (англ. computer-aided analysis) — подкласс средств CAE, используемых для компьютерного анализа.

CAM (англ. computer-aided manufacturing) — средства технологической подготовки производства изделий, обеспечивают автоматизацию программирования и управления оборудования с ЧПУ или ГАПС (Гибких автоматизированных производственных систем). Русским аналогом термина

является АСТПП — автоматизированная система технологической подготовки производства.

CAPP (англ. computer-aided process planning) — средства автоматизации планирования технологических процессов, применяемые на стыке систем CAD и CAM.

Многие системы автоматизированного проектирования совмещают в себе решение задач, относящихся к различным аспектам проектирования CAD/CAM, CAD/CAE, CAD/CAE/CAM. Такие системы называют комплексными, или интегрированными.

Лекция 2. (2 часа)

Планирование научного эксперимента.

Обработка результатов научного исследования

Основные вопросы

Методы планирования научного эксперимента.

Методы статистического анализа данных

Цель: Изучить этапы планирования научного эксперимента и ознакомиться с методами обработки результатов научного исследования.

Эксперимент представляет собой систему операций, воздействий и наблюдений, направленных на получение информации об объекте при исследовательских испытаниях.

Опыт – отдельная элементарная часть эксперимента.

От сравнительно пассивного наблюдения эксперимент отличается активным воздействием исследователя на объект изучения. В ходе эксперимента должны быть выполнены следующие операции: определение цели эксперимента; теоретическое обоснование условий эксперимента, способствующих наиболее полному и всестороннему проявлению заданных свойств и связей изучаемого объекта; разработка методики эксперимента; разработка экспериментальных средств исследования; наблюдение и измерение изучаемых характеристик объекта; статистическая обработка результатов экспериментов, вывод математических зависимостей, построение диаграмм, графиков, схем; анализ экспериментальных результатов, их обобщение. Планирование эксперимента – это процедура определения числа опытов и условий их проведения, необходимых и достаточных для решения поставленной задачи с требуемой точностью.

Различают однофакторный и многофакторный эксперименты.

Однофакторный эксперимент заключается в изучении поведения каждого фактора в отдельности и включает: выделение нужных факторов; стабилизацию мешающих факторов; поочередное варьирование интересующих исследователя факторов.

Стратегия многофакторного эксперимента состоит в том, что варьируются все переменные сразу и каждый эффект оценивается по результатам всех опытов, проведенных в данной серии экспериментов.

Планирование эксперимента (experimental design techniques) – комплекс методов математической статистики, направленных на постановку опытов и проведение рациональных измерений, подверженных случайным ошибкам.

Общая схема проведения эксперимента выглядит следующим образом. Со случайными ошибками измеряются некоторые выходные переменные изучаемой системы, зависящие от неизвестных значений параметров и известных значений переменных-факторов, а также из возможных взаимодействий. Основная цель планирования эксперимента – достижение максимальной точности измерений при минимальном количестве произведенных опытов и сохранении статистической достоверности результатов.

Основные этапы планирования эксперимента:

1. Установление цели эксперимента – постановка целей и задач проведения эксперимента;
2. Уточнение условий проведения эксперимента – выбор оборудования, сроков работ, способа проведения эксперимента и т.п.;
3. Выбор входных и выходных параметров – выбор зависимой измеряемой переменной, определение случайных и детерминированных независимых переменных;
4. Установление необходимой точности результатов измерений – выбор компромисса между минимальным числом испытаний и статистической достоверностью получаемых результатов;
5. Составление плана и проведение эксперимента – количество и порядок испытаний, задание совокупности значений задаваемых переменных-факторов и их взаимодействий в эксперименте;
6. Статистическая обработка результатов эксперимента – применение методов математической статистики для обработки результатов, построение математической модели эксперимента;
7. Формулирование выводов.

Планирование эксперимента широко используют в промышленной статистике, социологических, психологических и маркетинговых исследованиях. В условиях статистического эксперимента одна из основных

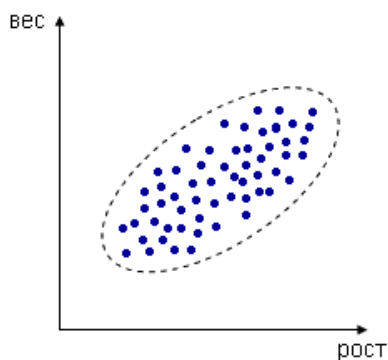
задач заключается в определении максимального количества информации о влиянии изучаемых факторов на изучаемый процесс с помощью наименьшего числа наблюдений.

Методы анализа данных экспериментов опираются на теорию вероятностей и методы математической статистики: корреляционный анализ, регрессионный анализ, дисперсионный анализ.

Корреляционный анализ - метод, позволяющий обнаружить зависимость между несколькими случайными величинами.

Допустим, проводится независимое измерение различных параметров у одного типа объектов. Из этих данных можно получить качественно новую информацию - о взаимосвязи этих параметров.

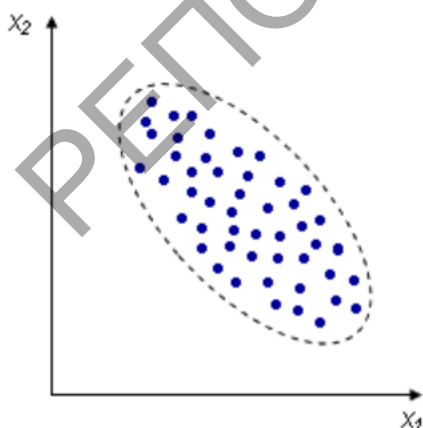
Например, измеряем рост и вес человека, каждое измерение представлено точкой в двумерном пространстве:



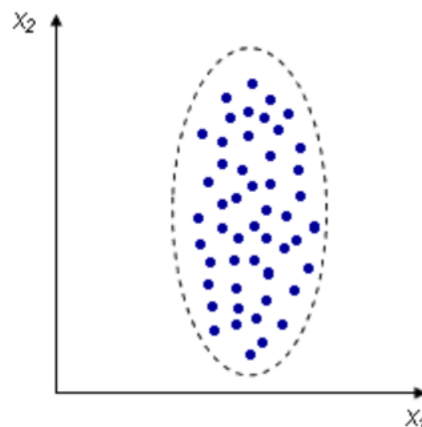
Несмотря на то, что величины носят случайный характер, в общем наблюдается некоторая зависимость - величины коррелируют.

В данном случае это положительная корреляция (при увеличении одного параметра второй тоже увеличивается). Возможны также такие случаи:

Отрицательная корреляция:



Отсутствие корреляции:



Для получения численной характеристики вводится коэффициент корреляции. Он рассчитывается следующим образом:

Есть массив из n точек $\{x_{1,i}, x_{2,i}\}$

Рассчитываются средние значения для каждого параметра:

$$\bar{x}_1 = \frac{\sum x_{1j}}{n}, \quad \bar{x}_2 = \frac{\sum x_{2j}}{n}$$

И коэффициент корреляции: коэффициент корреляции

$$r = \frac{\sum (x_{1j} - \bar{x}_1) \cdot (x_{2j} - \bar{x}_2)}{\sqrt{\sum (x_{1j} - \bar{x}_1)^2} \cdot \sqrt{\sum (x_{2j} - \bar{x}_2)^2}}$$

Коэффициент корреляции r изменяется в пределах от -1 до 1. В данном случае это линейный коэффициент корреляции, он показывает линейную взаимосвязь между x_1 и x_2 : r равен 1 (или -1), если связь линейна. В случае, если этот коэффициент положителен, имеет место прямая корреляция, если отрицателен, обратная. Чем больше коэффициент корреляции по своему абсолютному значению тем интенсивнее выражена линейная связь между переменными.

Рассмотрим две непрерывные переменные $x=(x_1, x_2, \dots, x_n)$, $y=(y_1, y_2, \dots, y_n)$. Разместим точки на двумерном графике рассеяния и скажем, что мы имеем линейное соотношение, если данные аппроксимируются прямой линией.

Если мы полагаем, что y зависит от x , причём изменения в y вызваны именно изменениями в x , мы можем определить линию регрессии (регрессия y на x), которая лучше всего описывает прямолинейное соотношение между этими двумя переменными.

Статистическое использование слова "регрессия" исходит из явления, известного как регрессия к среднему, приписываемого сэру Френсису Гальтону (1889).

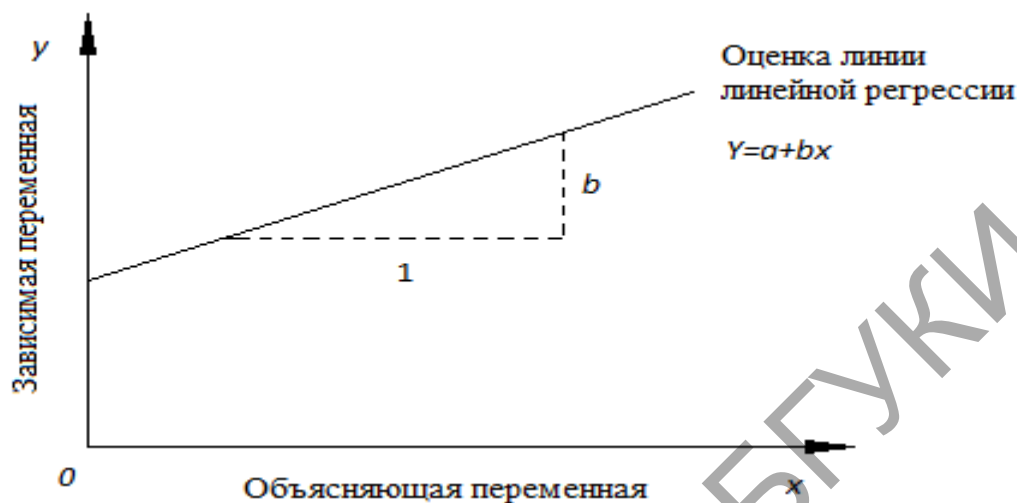
Он показал, что, хотя высокие отцы имеют тенденцию иметь высоких сыновей, средний рост сыновей меньше, чем у их высоких отцов. Средний рост сыновей "регрессировал" и "двигался вспять" к среднему росту всех отцов в популяции. Таким образом, в среднем высокие отцы имеют более низких (но всё-таки высоких) сыновей, а низкие отцы имеют сыновей более высоких (но всё-таки довольно низких).

Математическое уравнение, которое оценивает линию простой (парной) линейной регрессии имеет вид:

$$Y = a + bx,$$

где x называется независимой переменной или предиктором, Y – зависимая переменная или переменная отклика. Это значение, которое мы ожидаем для y (в среднем), если мы знаем величину x , т.е. это «предсказанное значение y », a – свободный член (пересечение) линии оценки; это значение Y , когда $x=0$, b –

угловой коэффициент или градиент оценённой линии; она представляет собой величину, на которую Y увеличивается в среднем, если мы увеличиваем x на одну единицу. a и b называют коэффициентами регрессии оценённой линии, хотя этот термин часто используют только для b .



Парную линейную регрессию можно расширить, включив в нее более одной независимой переменной; в этом случае она известна как множественная регрессия.

Тема 7. Методы оптимизации и информационные системы в управлении, образовании, науке и творческой деятельности

Лекция 1. (2 часа)

Модели и задачи оптимизации

Основные вопросы

Оптимизация как этап вычислительного эксперимента.

Модели оптимизации.

Оптимизация различных предметных областях.

Цель: Изучить модели и задачи оптимизации, а так же их применение в различных предметных областях.

Оптимизация - процесс нахождения экстремума (глобального максимума или минимума) определённой функции или выбора наилучшего (оптимального) варианта из множества возможных.

Наиболее надёжным способом нахождения наилучшего варианта является сравнительная оценка всех возможных вариантов (альтернатив). Если число альтернатив велико, при поиске наилучшей обычно используют методы

математического программирования. Применить эти методы можно, если есть строгая постановка задачи:

- задан набор переменных
- установлена область их возможного изменения (заданы ограничения)
- определён вид целевой функции (функции, экстремум которой нужно найти) от этих переменных: есть количественная мера (критерий) оценки степени достижения поставленной цели.

Число всевозможных конкретных моделей задач оптимизации очень велико, так же как и число проблем, для разрешения которых они разработаны.

Теория игр. Одна из важнейших переменных, от которой зависит успех организации, - конкурентоспособности. Очевидно, способность прогнозировать действия конкурентов означает преимущество для любой организации. Теория игр – метод моделирования оценки воздействия принятого решения на конкурентов.

Теорию игр изначально разработали военные с тем, чтобы в стратегии можно было учесть возможные действия противника. В бизнесе игровые модели используются для прогнозирования реакции конкурентов на изменение цен, новые компании поддержки сбыта, предложения дополнительного обслуживания, модификацию и освоение новой продукции. Если, например, с помощью теории игр руководство устанавливает, что при повышении цен конкуренты не сделают того же, оно, вероятно, должно отказаться от этого шага, чтобы не попасть в невыгодное положение в конкурентной борьбе.

Теория игр используется не так часто, как другие модели. К сожалению, ситуации реального мира зачастую очень сложны и на столько быстро изменяются, что невозможно точно спрогнозировать, как отреагируют конкуренты на изменение тактики фирмы. Тем не менее, теория игр полезна, когда требуется определить наиболее важные и требующие учета факторы в ситуации принятия решений в условиях конкурентной борьбы. Эта информация важна, поскольку позволяет руководству учесть дополнительные переменные или факторы, могут повлиять на ситуацию, и тем самым повышает эффективность решения.

Модель теории очередей. Модель теории очередей или модель оптимального обслуживания используется для определения оптимального числа каналов обслуживания по отношению потребности в них. К ситуациям, в которых модели теории очередей могут быть полезны, можно отнести звонки людей в авиакомпанию для резервирования места и получения информации, ожидание в очереди на машинную обработку данных, мастеров по ремонту оборудования, очередь грузовиков под разгрузку на склад, ожидание клиентами банка свободного кассира. Если, например, клиентам приходится слишком

долго ждать кассира, они могут решить перенести свои счета в другой банк. Подобным образом, если грузовикам приходится слишком долго дожидаться разгрузки, они не смогут выполнить столько поездок за день, сколько положено. Таким образом, принципиальная проблема заключается в уравнивании расходов на дополнительные каналы обслуживания (больше людей для разгрузки грузовиков, больше кассиров, больше клерков, занимающихся предварительной продажей билетов на самолёты) и потерь от обслуживания на уровне ниже оптимального (грузовики не смогут сделать лишнюю остановку из-за задержек под разгрузкой, потребители уходят в другой банк или обращаются к другой авиакомпании из-за медленного обслуживания).

Модели очередей снабжают руководство инструментом определения оптимального числа каналов обслуживания, которые необходимо иметь, чтобы сбалансировать издержки в случаях чрезмерно малого и чрезмерно большого их количества.

Модели управления запасами. Модель управления запасами используется для определения времени размещения заказов на ресурсы и их количества, а также массы готовой продукции на складах. Любая организация должна поддерживать некоторый уровень запасов во избежание задержек на производстве и в сбыте.

Цель данной модели – сведение к минимуму отрицательных последствий накопления запасов, что выражается в определённых издержках. Эти издержки бывают трех основных видов: на размещение заказов, на хранение, а также потери, связанные с недостаточным уровнем запасов. В этом случае продажа готовой продукции или предоставление обслуживания становятся невозможными, а также возникают потери от простоя производственных линий, в частности, в связи с необходимостью оплаты труда работников, хотя они не работают в данный момент.

Поддержание высокого уровня запасов избавляет от потерь, обуславливаемых их нехваткой. Закупка в больших количествах материалов, необходимых для создания запасов, во многих случаях сводит к минимуму издержки на размещение заказов, поскольку фирма может получить соответствующие скидки и снизить объем «бумажной работы». Однако эти потенциальные выгоды перекрываются дополнительными издержками типа расходов на хранение, перегрузку, выплату процентов, затрат на страхование, потерь от порчи, воровства и т.д.

Модель линейного программирования применяют для определения оптимального способа распределения дефицитных ресурсов при наличии

конкурирующих потребностей. Линейное программирование обычно используют специалисты штабных подразделений для разрешения производственных трудностей.

Типичные варианты применения линейного программирования в управлении производством:

1) укрупненное планирование производства (составление графиков производства, минимизирующих общие издержки с учетом издержек в связи с изменением ставки процента, заданных ограничений по трудовым ресурсам и уровням запасов);

2) планирование ассортимента изделий (определение оптимального ассортимента продукции, в котором каждому ее виду свойственны свои издержки и потребности в ресурсах);

3) маршрутизация производства изделия (определение оптимального технологического маршрута изготовления изделия, которое должно быть последовательно пропущено через несколько обрабатывающих центров, причем каждая операция центра характеризуется своими издержками и производительностью);

4) управление технологическим процессом (сведение к минимуму выхода стружки при резке стали, отходов кожи или ткани в рулоне или полотнище);

5) регулирование запасов (определение оптимального сочетания продуктов на складе или в хранилище);

6) календарное планирование производства (составление календарных планов, минимизирующих издержки с учетом расходов на содержание запасов, оплата сверхурочной работы и заказов на стороне);

7) планирование распределения продукции (составление оптимального графика отгрузки с учетом распределения продукции между производственными предприятиями и складами, складами и магазинами розничной торговли);

8) определение оптимального местоположения нового завода (определение наилучшего пункта местоположения путем оценки затрат на транспортировку между альтернативными местами размещения нового завода и местами его снабжения и сбыта готовой продукции);

9) календарное планирование транспорта (минимизация издержек подачи грузовиков под погрузку и транспортных судов к погрузочным причалам);

10) распределения рабочих (минимизация издержек при распределении рабочих по станкам и рабочим местам);

11) перегрузка материалов (минимизация издержек при маршрутизации движения средств перегрузки материалов, например, автопогрузчиков, между отделениями завода и доставке материалов с открытого склада к местам их

переработки на грузовых автомобилях разной грузоподъемности с разными ТЭХ).

Имитационное моделирование. Все описанные выше модели подразумевают применение имитации в широком смысле, поскольку все являются заменителями реальности. Тем не менее как метод моделирования, имитация конкретно обозначает процесс создания модели и ее экспериментальное применение для определения изменений реальной ситуации. Главная идея имитации состоит в использовании некоего устройства для имитации реальной системы для того, чтобы исследовать и понять ее свойства, поведения и характеристики. Аэродинамическая труба - пример физически осязаемой имитационной модели, используемой для проверки характеристик разрабатываемых самолетов и автомобилей. Специалисты по производству и финансам могут разрабатывать модели, позволяющие имитировать ожидаемый прирост производительности и прибыли в результате применения новой технологии или изменения состава рабочей силы.

Имитация используется в ситуациях, слишком сложных для математических методов типа линейного программирования. Это может быть связано с чрезмерно большим числом переменных, трудностью математического анализа определенных зависимостей между переменными или высоким уровнем неопределенности.

Итак, имитация – это часто весьма практичный способ подстановки модели на место реальной системы или натурального прототипа. Эксперименты на реальных или прототипных системах стоят дорого и продолжаются долго, а релевантные переменные не всегда поддаются регулированию. Экспериментируя на модели системы, можно установить, как она будет реагировать на определенные изменения или события, в то время когда отсутствует возможность наблюдать эту систему в реальности. Если результаты экспериментирования с использованием имитационной модели свидетельствует о том, что модификация ведет к улучшению, руководитель может с большей уверенностью принимать решение об осуществлении изменения в реальной системе.

Экономический анализ. Почти все руководители воспринимают имитацию как метод моделирования. Однако многие из них никогда не думали, что экономический анализ – очевидно наиболее распространенный метод – это тоже одна из форм построения модели. Экономический анализ вбирает в себя почти все методы оценки издержек и экономических выгод, а также относительной рентабельности деятельности предприятия. Типичная «экономическая» модель основана на анализе безубыточности, методе принятия решений с определением точки, в которой общий доход уравнивается

с суммарными издержками, т.е. точки, в которой предприятие становится прибыльным.

Объем производства, обеспечивающий безубыточность, можно рассчитать почти по каждому виду продукции или услуге, если соответствующие издержки удастся определить. Это может быть число сидений в самолете, которые должны быть заняты пассажирами, число посетителей в ресторане, объем сбыта нового типа автомобиля.

В дополнение к моделированию, имеется ряд методов, способных оказать помощь руководителю в поиске объективно обоснованного решения по выбору из нескольких альтернатив той, которая в наибольшей мере способствует достижению целей.

Платежная матрица. Суть каждого принимаемого руководством решения – выбор наилучшей из нескольких альтернатив по конкретным установленным заранее критериям. Платежная матрица – это один из методов статистической теории решений, метод, который может оказать помощь руководителю в выборе одного из нескольких вариантов. Он особенно полезен, когда руководитель должен установить, какая стратегия в наибольшей мере будет способствовать достижению целей. Платеж представляет собой денежное вознаграждение или полезность, являющиеся следствием конкретной стратегии в сочетании с конкретными обстоятельствами. Если платежи представить в форме таблицы (или матрицы), мы получаем платежную матрицу. Слова «в сочетании с конкретными обстоятельствами» очень важны, чтобы понять, когда можно использовать платежную матрицу и оценить, когда решение, принятое на ее основе, скорее всего будет надежным. В самом общем виде матрица означает, что платеж зависит от определенных событий, которые фактически совершаются. Если такое событие или состояние природы не случается на деле, платеж неизбежно будет иным. В целом платежная матрица полезна, когда:

- 1) имеется разумно ограниченное число альтернатив или вариантов стратегии для выбора между ними.
- 2) то, что может случиться, с полной определенностью не известно.
- 3) результаты принятого решения зависят от того, какая именно выбрана альтернатива и какие события в действительности имеют место.

Кроме того, руководитель должен располагать возможностью объективной оценки вероятности релевантных событий и расчета ожидаемого значения такой вероятности. Руководитель редко имеет полную определенность, но также редко он действует в условиях полной неопределенности. Почти во всех случаях принятия решений руководителю приходится оценивать вероятность или возможность события. Вероятность можно определить объективно, как поступает игрок в рулетку, ставя на

нечетные номера. Выбор ее значения может опираться на прошлые тенденции или субъективную оценку руководителя, который исходит из собственного опыта действий в подобных ситуациях.

Дерево решений – это схематическое представление проблемы принятия решений. Как и платежная матрица, дерево решений дает руководителю возможность учесть различные направления действий, соотнести с ними финансовые результаты, скорректировать их в соответствии с приписанной им вероятностью, а затем сравнить альтернативы. Концепция ожидаемого значения является неотъемлемой частью метода дерева решений.

Дерево решений можно строить под сложные ситуации, когда результаты одного решения влияют на последующие решения. Таким образом, дерево решений – это полезный инструмент для принятия последовательных решений.

Многие допущения, из которых исходит руководитель, относятся к условиям в будущем, над которыми руководитель почти не имеет никакого контроля. Однако такого рода допущения необходимы для многих операций планирования. Ясно, что чем лучше руководитель сможет предсказать внешние и внутренние условия применительно к будущему, тем выше шансы на составление осуществимых планов.

Прогнозирование – это метод, в котором используются как накопленный опыт, так и текущие допущения насчет будущего с целью его определения.

Разновидности прогнозов:

1) экономические прогнозы используются для предсказания общего состояния экономики и объема сбыта для конкретной компании или по конкретному продукту.

2) Прогнозы развития технологии позволят предсказать, разработки каких новых технологий можно ожидать, когда это может произойти, насколько экономически приемлемыми они могут быть.

3) Прогнозы развития конкуренции позволяют предсказывать стратегию и тактику конкурентов.

4) Прогнозы на основе опросов и исследований дают возможность предсказать, что произойдет в сложных ситуациях, используя данные многих областей знаний. Например, будущий рынок автомобилей можно оценить только с учетом надвигающегося изменения состояния экономики, общественных ценностей, политической обстановки, технологии и стандартов по защите окружающей среды от загрязнений.

5) Социальное прогнозирование, которым в настоящее время занимается всего несколько крупных организаций, используется для предсказания изменений в социальных установках людей и состояния общества

Методы прогнозирования:

- неформальные методы;
- количественные;
- качественные.

К неформальным методам относят:

- вербальная информация (информация получаемая из радио- и телепередач, от потребителей, поставщиков, конкурентов, на торговых совещаниях, в профессиональных организациях, от юристов, бухгалтеров, финансовых ревизоров и консультантов. Такая информация затрагивает все основные факторы внешнего окружения, представляющие интерес для организаций. Она имеет откровенно переменчивый характер, ее легко получить, и часто на нее вполне полагаются. Иногда, впрочем, данные могут оказаться неточными, устаревшими или страдающими расплывчатостью. Если такое происходит, и руководство использует некачественную информацию для формулирования целей организации, количество проблем при осуществлении целей может быть значительным);

- письменная информация (газеты, торговые журналы, информационные бюллетени, профессиональные журналы и годовые отчеты. Хотя эта информация легко доступна, она страдает теми же недостатками, что и вербальная информация, а именно, она может быть не свежей и не особенно глубокой);

- промышленный шпионаж (иногда он оказывается успешным способом сбора данных о действиях конкурентов, и эти данные затем использовались для переформулирования целей организации. Поэтому руководители должны защищать данные, имеющие статус их интеллектуальной собственности).

Количественные методы можно использовать для прогнозирования, когда есть основание считать, что деятельность в прошлом имела определенную тенденцию, которую можно продолжить в будущем, и когда имеющейся информации достаточно для выявления статистически достоверных тенденций или зависимостей. Кроме того, руководитель обязан знать, как использовать количественную модель, и помнить, что выгоды от принятия более эффективного решения должно перекрыть расходы на создание модели. Два типичных метода количественного прогнозирования – это анализ временных рядов и каузальное (причинно-следственное) моделирование.

Анализ временных рядов. Иногда называемый проецированием тренда, анализ временных рядов основан на допущении, согласно которому случившееся в прошлом дает достаточно хорошее приближение в оценке будущего. Этот анализ является методом выявления образцов и тенденций прошлого и продления их в будущее. Данный метод анализа часто используется для оценки спроса на товары и услуги, оценки потребности в запасах,

прогнозирования структуры сбыта, характеризующегося сезонными колебаниями, и потребности в кадрах.

Каузальное (причинно-следственное) моделирование. Каузальное моделирование – наиболее хитроумный и математически сложный количественный метод прогнозирования из числа применяемых сегодня. Он используется в ситуациях с более чем одной переменной. Каузальное моделирование – это попытка спрогнозировать то, что произойдет в подобных ситуациях, путем исследования статистической зависимости между рассматриваемыми факторами и другими переменными.

Когда количество информации недостаточно или руководство не понимает сложный метод, или когда количественная модель получается чрезмерно дорогой, руководство может прибегнуть к качественным моделям прогнозирования. При этом прогнозирование будущего осуществляется экспертами, к которым обращаются за помощью. Четыре наиболее распространенных качественных методов прогнозирования – это мнение жюри, совокупное мнение сбытовиков, модель ожидания потребителя и метод экспертных цен.

Мнение жюри. Этот метод заключается в соединении и усреднении мнений экспертов в релевантных сферах. Неформальной разновидностью этого метода является «мозговой штурм», во время которого участники сначала пытаются генерировать как можно больше идей. Только после прекращения процесса генерирования некоторые идеи подвергаются оценке. Этот метод может отнимать много времени, но зачастую дает полезные результаты, особенно когда организация нуждается во множестве новых идей и альтернатив.

Совокупное мнение сбытовиков. Опытные торговые агенты часто прекрасно предсказывают будущий спрос. Они близко знакомы с потребителями и могут принять в расчет их недавние действия быстрее, чем удастся построить количественную модель. Кроме того, хороший торговый агент на определенном временном отрезке зачастую «чувствует» рынок по сути дела точнее, чем количественные модели.

Модель ожидания потребителя. Прогноз, основанный на результатах опроса клиентов организации. Их просят оценить собственные потребности в будущем, а также новые требования. Собранные все полученные таким путем данные и сделав поправки на пере- или недооценку, исходя из собственного опыта, руководитель зачастую оказывается в состоянии точно предсказать совокупный спрос.

Метод экспертных оценок. Этот метод представляет собой процедуру, позволяющую группе экспертов прийти к согласию. Эксперты заполняют подробные вопросники по поводу рассматриваемой проблемы. Они также

записывают свои мнения о ней. Каждый эксперт затем получает свод ответов других экспертов, и его просят заново рассмотреть свой прогноз, и если он не совпадает с прогнозами других, просят объяснить, почему это так. Процедура повторяется обычно три или четыре раза, пока эксперты не приходят к единому мнению.

В настоящее время для решения оптимальных задач применяются следующие методы:

- методы исследования функций классического анализа;
- методы, основанные на использовании неопределенных множителей Лагранжа;
- вариационное исчисление;
- динамическое программирование;
- принцип максимума;
- линейное программирование;
- нелинейное программирование.
- геометрическое программирование.

Как правило, нельзя рекомендовать какой-либо один метод, который можно использовать для решения всех без исключения задач, возникающих на практике. Одни методы в этом отношении являются более общими, другие - менее общими. Наконец, целую группу методов (методы исследования функций классического анализа, метод множителей Лагранжа, методы нелинейного программирования) на определенных этапах решения оптимальной задачи можно применять в сочетании с другими методами, например динамическим программированием или принципом максимума.

Некоторые методы специально разработаны или наилучшим образом подходят для решения оптимальных задач с математическими моделями определенного вида. Так, математический аппарат линейного программирования, специально создан для решения задач с линейными критериями оптимальности и линейными ограничениями на переменные и позволяет решать большинство задач, сформулированных в такой постановке. Так же и геометрическое программирование предназначено для решения оптимальных задач, в которых критерий оптимальности и ограничения представляются специального вида функциями позиномами.

Динамическое программирование хорошо приспособлено для решения задач оптимизации многостадийных процессов, особенно тех, в которых состояние каждой стадии характеризуется относительно небольшим числом переменных состояния. Однако при наличии значительного числа этих переменных, т. е. при высокой размерности каждой стадии, применение метода

динамического программирования затруднительно вследствие ограниченных быстродействия и объема памяти вычислительных машин.

Наилучшим путем при выборе метода оптимизации, наиболее пригодного для решения соответствующей задачи, следует признать исследование возможностей и опыта применения различных методов оптимизации. Ниже приводится краткий обзор математических методов решения оптимальных задач и примеры их использования. Здесь же дана лишь краткая характеристика указанных методов и областей их применения, что до некоторой степени может облегчить выбор того или иного метода для решения конкретной оптимальной задачи.

Методы исследования функций классического анализа представляют собой наиболее известные методы решения несложных оптимальных задач, с которыми известны из курса математического анализа. Обычной областью использования данных методов являются задачи с известным аналитическим выражением критерия оптимальности, что позволяет найти не очень сложное, также аналитическое выражение для производных. Полученные приравниванием нулю производных уравнения, определяющие экстремальные решения оптимальной задачи, крайне редко удается решить аналитическим путем, поэтому, как, правило, применяют вычислительные машины. При этом надо решить систему конечных уравнений, чаще всего нелинейных, для чего приходится использовать численные методы, аналогичные методам нелинейного программирования.

Дополнительные трудности при решении оптимальной задачи методами исследования функций классического анализа возникают вследствие того, что система уравнений, получаемая в результате их применения, обеспечивает лишь необходимые условия оптимальности. Поэтому все решения данной системы (а их может быть и несколько) должны быть проверены на достаточность. В результате такой проверки сначала отбрасывают решения, которые не определяют экстремальные значения критерия оптимальности, а затем среди остающихся экстремальных решений выбирают решение, удовлетворяющее условиям оптимальной задачи, т. е. наибольшему или наименьшему значению критерия оптимальности в зависимости от постановки задачи.

Методы исследования при наличии ограничений на область изменения независимых переменных можно использовать только для отыскания экстремальных значений внутри указанной области. В особенности это относится к задачам с большим числом независимых переменных (практически больше двух), в которых анализ значений критерия оптимальности на границе допустимой области изменения переменных становится весьма сложным.

Метод множителей Лагранжа применяют для решения задач такого же класса сложности, как и при использовании обычных методов исследования функций, но при наличии ограничений типа равенств на независимые переменные. К требованию возможности получения аналитических выражений для производных от критерия оптимальности при этом добавляется аналогичное требование относительно аналитического вида уравнений ограничений.

В основном при использовании метода множителей Лагранжа приходится решать те же задачи, что и без ограничений. Некоторое усложнение в данном случае возникает лишь от введения дополнительных неопределенных множителей, вследствие чего порядок системы уравнений, решаемой для нахождения экстремумов критерия оптимальности, соответственно повышается на число ограничений. В остальном, процедура поиска решений и проверки их на оптимальность отвечает процедуре решения задач без ограничений.

Множители Лагранжа можно применять для решения задач оптимизации объектов на основе уравнений с частными производными и задач динамической оптимизации. При этом вместо решения системы конечных уравнений для отыскания оптимума необходимо интегрировать систему дифференциальных уравнений.

Следует отметить, что множители Лагранжа используют также в качестве вспомогательного средства и при решении специальными методами задач других классов с ограничениями типа равенств, например, в вариационном исчислении и динамическом программировании. Особенно эффективно применение множителей Лагранжа в методе динамического программирования, где с их помощью иногда удается снизить размерность решаемой задачи.

Методы вариационного исчисления обычно используют для решения задач, в которых критерии оптимальности представляются в виде функционалов и решениями которых служат неизвестные функции. Такие задачи возникают обычно при статической оптимизации процессов с распределенными параметрами или в задачах динамической оптимизации.

Вариационные методы позволяют в этом случае свести решение оптимальной задачи к интегрированию системы дифференциальных уравнений Эйлера, каждое из которых является нелинейным дифференциальным уравнением второго порядка с граничными условиями, заданными на обоих концах интервала интегрирования. Число уравнений указанной системы при этом равно числу неизвестных функций, определяемых при решении оптимальной задачи. Каждую функцию находят в результате интегрирования получаемой системы.

Уравнения Эйлера выводятся как необходимые условия экстремума функционала. Поэтому полученные интегрированием системы дифференциальных уравнений функции должны быть проверены на экстремум функционала.

При наличии ограничений типа равенств, имеющих вид функционалов, применяют множители Лагранжа, что дает возможность перейти от условной задачи к безусловной. Наиболее значительные трудности при использовании вариационных методов возникают в случае решения задач с ограничениями типа неравенств.

Заслуживают внимания прямые методы решения задач оптимизации функционалов, обычно позволяющие свести исходную вариационную задачу к задаче нелинейного программирования, решить которую иногда проще, чем краевую задачу для уравнений Эйлера.

Динамическое программирование служит эффективным методом решения задач оптимизации дискретных многостадийных процессов, для которых критерий оптимальности задается как аддитивная функция критериев оптимальности отдельных стадий. Без особых затруднений указанный метод можно распространить и на случай, когда критерий оптимальности задан в другой форме, однако при этом обычно увеличивается размерность отдельных стадий.

По существу метод динамического программирования представляет собой алгоритм определения оптимальной стратегии управления на всех стадиях процесса. При этом закон управления на каждой стадии находят путем решения частных задач оптимизации последовательно для всех стадий процесса с помощью методов исследования функций классического анализа или методов нелинейного программирования. Результаты решения обычно не могут быть выражены в аналитической форме, а получаются в виде таблиц.

Ограничения на переменные задачи не оказывают влияния на общий алгоритм решения, а учитываются при решении частных задач оптимизации на каждой стадии процесса. При наличии ограничений типа равенств иногда даже удается снизить размерность этих частных задач за счет использования множителей Лагранжа. Применение метода динамического программирования для оптимизации процессов с распределенными параметрами или в задачах динамической оптимизации приводит к решению дифференциальных уравнений в частных производных. Вместо решения таких уравнений зачастую значительно проще представить непрерывный процесс как дискретный с достаточно большим числом стадий. Подобный прием оправдан особенно в тех случаях, когда имеются ограничения на переменные задачи и прямое решение

дифференциальных уравнений осложняется необходимостью учета указанных ограничений.

При решении задач методом динамического программирования, как правило, используют вычислительные машины, обладающие достаточным объемом памяти для хранения промежуточных результатов решения, которые обычно получают в табличной форме.

Принцип максимума применяют для решения задач оптимизации процессов, описываемых системами дифференциальных уравнений. Достоинством математического аппарата принципа максимума является то, что решение может определяться в виде разрывных функций; это свойственно многим задачам оптимизации, например задачам оптимального управления объектами, описываемыми линейными дифференциальными уравнениями.

Нахождение оптимального решения при использовании принципа максимума сводится к задаче интегрирования системы дифференциальных уравнений процесса и сопряженной системы для вспомогательных функций при граничных условиях, заданных на обоих концах интервала интегрирования, т. е. к решению краевой задачи. На область изменения переменных могут быть наложены ограничения. Систему дифференциальных уравнений интегрируют, применяя обычные программы на цифровых вычислительных машинах.

Принцип максимума для процессов, описываемых дифференциальными уравнениями, при некоторых предположениях является достаточным условием оптимальности. Поэтому дополнительной проверки на оптимум получаемых решений обычно не требуется.

Для дискретных процессов принцип максимума в той же формулировке, что и для непрерывных, вообще говоря, несправедлив. Однако условия оптимальности, получаемые при его применении для многостадийных процессов, позволяют найти достаточно удобные алгоритмы оптимизации.

Линейное программирование представляет собой математический аппарат, разработанный для решения оптимальных задач с линейными выражениями для критерия оптимальности и линейными ограничениями на область изменения переменных. Такие задачи обычно встречаются при решении вопросов оптимального планирования производства с ограниченным количеством ресурсов, при определении оптимального плана перевозок (транспортные задачи) и т. д.

Для решения большого круга задач линейного программирования имеется практически универсальный алгоритм - симплексный метод, позволяющий за конечное число итераций находить оптимальное решение подавляющего большинства задач. Тип используемых ограничений (равенства или неравенства) не сказывается на возможности применения указанного

алгоритма. Дополнительной проверки на оптимальность для получаемых решений не требуется. Как правило, практические задачи линейного программирования отличаются весьма значительным числом независимых переменных. Поэтому для их решения обычно используют вычислительные машины, необходимая мощность которых определяется размерностью решаемой задачи.

Методы нелинейного программирования применяют для решения оптимальных задач с нелинейными функциями цели. На независимые переменные могут быть наложены ограничения также в виде нелинейных соотношений, имеющих вид равенств или неравенств. По существу методы нелинейного программирования используют, если ни один из перечисленных выше методов не позволяет сколько-нибудь продвинуться в решении оптимальной задачи. Поэтому указанные методы иногда называют также прямыми методами решения оптимальных задач.

Для получения численных результатов важное место отводится нелинейному программированию и в решении оптимальных задач такими методами, как динамическое программирование, принцип максимума и т. п. на определенных этапах их применения.

Названием “методы нелинейного программирования” объединяется большая группа численных методов, многие из которых приспособлены для решения оптимальных задач соответствующего класса. Выбор того или иного метода обусловлен сложностью вычисления критерия оптимальности и сложностью ограничивающих условий, необходимой точностью решения, мощностью имеющейся вычислительной машины и т.д. Ряд методов нелинейного программирования практически постоянно используется в сочетании с другими методами оптимизации, как, например, метод сканирования в динамическом программировании. Кроме того, эти методы служат основой построения систем автоматической оптимизации - оптимизаторов, непосредственно применяющихся для управления производственными процессами.

Геометрическое программирование есть метод решения одного специального класса задач нелинейного программирования, в которых критерий оптимальности и ограничения задаются в виде полиномов - выражений, представляющих собой сумму произведений степенных функций от независимых переменных. С подобными задачами иногда приходится сталкиваться в проектировании. Кроме того, некоторые задачи нелинейного программирования иногда можно свести к указанному представлению, используя аппроксимационное представление для целевых функций и ограничений.

Специфической особенностью методов решения оптимальных задач (за исключением методов нелинейного программирования) является то, что до некоторого этапа оптимальную задачу решают аналитически, т. е. находят определенные аналитические выражения, например, системы конечных или дифференциальных уравнений, откуда уже отыскивают оптимальное решение. В отличие от указанных методов при использовании методов нелинейного программирования, которые, как уже отмечалось выше, могут быть названы прямыми, применяют информацию, получаемую при вычислении критерия оптимальности, изменение которого служит оценкой эффективности того или иного действия.

Важной характеристикой любой оптимальной задачи является ее размерность n , равная числу переменных, задание значений которых необходимо для однозначного определения состояния оптимизируемого объекта. Как правило, решение задач высокой размерности связано с необходимостью выполнения большого объема вычислений. Ряд методов (например, динамическое программирование и дискретный принцип максимума) специально предназначен для решения задач оптимизации процессов высокой размерности, которые могут быть представлены как многостадийные процессы с относительно невысокой размерностью каждой стадии.

Лекция 2. (2 часа)

Системы, основанные на знаниях и системы поддержки принятия решений

Основные вопросы:

Искусственный интеллект.

задачи решаемые с помощью искусственного интеллекта.

Экспертные системы.

Система поддержки принятия решений.

Цель: Изучить основные направления развития систем, основанных на знаниях и систем поддержки принятия решений

Исследования в области искусственного интеллекта начали проводиться в 60-х годах. Под искусственным интеллектом сегодня понимают науку, изучающую возможности создания для ЭВМ таких программ, которые решают задачи, требующие определенных интеллектуальных усилий, а не рутинных процедур при выполнении их человеком. Задачей этой науки является

воссоздание с помощью искусственных устройств, в основном ЭВМ, разумных суждений и действий. К настоящему времени на основе исследований в области искусственного интеллекта возникла новая отрасль индустрии – разработка интеллектуальных систем.

С самого начала исследования в **области искусственного интеллекта развивались в двух направлениях**: • познание искусственного интеллекта и законов его функционирования. • создание искусственных систем, которые способны не хуже (а возможно, и лучше) выполнять ту работу, которую традиционно относят к сфере интеллектуального труда.

Исследования в этой области сконцентрированы на разработке и внедрении компьютерных программ, способных эмулировать (имитировать, воспроизводить) те области деятельности человека, которые требуют мышления, определенного мастерства и накопленного опыта. К ним относятся задачи принятия решений, распознавания образов и понимания человеческого языка. Эта технология уже успешно применяется в некоторых областях техники и жизни общества — органической химии, поиске полезных ископаемых, медицинской диагностике.

Центральная парадигма интеллектуальных технологий сегодня – это обработка знаний. Системы, ядром которых является БЗ или модель предметной области, описанная на языке сверхвысокого уровня, приближенном к естественному, называют интеллектуальными. Чаще всего ИС применяются для решения сложных задач, где основная сложность решения связана с использованием слабоформализованных знаний специалистов-практиков и где логическая (или смысловая) обработка информации превалирует над вычислительной.

На пути разработки таких систем существуют две основные трудности. Во-первых, в большинстве случаев, выполняя какие-либо действия, человек четко не осознает, как он это делает. Другими словами, он не знает точного алгоритма выполнения таких действий, как понимание текста, принятие решения в тех или иных условиях. Во-вторых, несмотря на постоянное совершенствование и развитие по уровню компетентности в рассматриваемой деятельности ЭВМ все еще далеки от человека: они работают по соответствующей программе.

Задачи, решаемые методами искусственного интеллекта, обладают двумя характерными особенностями:

- в них, как правило, используется информация в символьной форме, такая как буквы, слова, знаки, рисунки, в отличие от традиционных ЭВМ, которые обрабатывают данные в числовой форме;

- в этих задачах предполагается наличие выбора, а именно, отсутствие в них алгоритма означает только то, что необходимо сделать выбор между многими вариантами, и часто в условиях неопределенности.

Все существующие в настоящее время интеллектуальные системы можно условно разбить на два класса: специализированные и общего назначения.

К специализированным ИС относятся те, которые выполняют решение фиксированного набора задач, predetermined при проектировании системы. Для использования таких систем их требуется снабдить данными и знаниями, соответствующими конкретной проблемной или предметной областям.

К интеллектуальным системам общего назначения относятся такие, которые не только исполняют заданные процедуры, но на основе метапроцедур поиска генерируют и исполняют процедуры решения новых конкретных задач. В данных системах пользователь (эксперт) формирует знания (данные и правила), описывающие выбранную проблемную или предметную область. Затем на основании этих знаний, заданной цели и исходных данных метапроцедуры системы генерируют и исполняют процедуру решения конкретной задачи. Данная технология позволяет специалисту в некоторой проблемной области, не знающему программирования, разрабатывать гибкие прикладные системы.

Одним из основных типов интеллектуальных систем являются ЭС. Они появились в рамках исследований по ИИ в тот период, когда эта наука переживала серьезный кризис, и требовался существенных прорыв в развитии практических приложений. Пришла идея моделировать конкретные знания специалистов-экспертов. До сих пор единственным критерием интеллектуальности является наличие механизмов работы со знаниями.

Термин *«системы, основанные на знаниях»* появился в 1976 году одновременно с первыми системами, аккумулирующими опыт и знания экспертов. Это были экспертные системы для медицины и химии. Они ставили диагноз при инфекционных заболеваниях крови и расшифровывали данные масс-спектрографического анализа.

ЭС позволяют формализовать конкретные содержательные знания об объектах управления и протекающих в них процессах, то есть ввести в ЭВМ логико-лингвистические модели наряду с математическими. Логико-лингвистическое моделирование расширяет область применения ЭВМ за счет трудно или совсем неформализуемых ранее областей знаний (диспетчерское управление, управление гибким автоматизированным производством, управление боевыми действиями и т.п.).

ЭС эффективны в специфических «экспертных» областях, где важен опыт специалистов. Разработка ЭС направлена на использование ЭВМ для обработки информации в тех областях науки и техники, где традиционные математические методы моделирования малопригодны.

Знания и данные

Чем же отличаются знания от данных?

Данные – это информация, полученная в результате наблюдений или измерений отдельных свойств, характеризующих объекты, процессы и явления предметной области, то есть это конкретные факты, такие как температура воздуха, высота здания, фамилия и т.д.

Знания же основаны на данных, полученных эмпирическим путем. Они представляют собой результат опыта и мыслительной деятельности человека, направленной на обобщение этого опыта, полученного в результате практической деятельности. (Данные о высокой температуре у человека не позволяют решить задачу выздоровления. Однако знания о том, что температуру можно снизить тем или иным средством приближают решение задачи.)

Знания – это связи и закономерности предметной области (принципы, модели, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в данной области. Т.о. знания – это хорошо структурированные данные, или данные о данных, или метаданные.

Знания, которыми обладает специалист в какой-либо области, можно разделить на формализованные (точные) и неформализованные (неточные). Формализованные знания формулируются в книгах и руководствах в виде общих и строгих суждений, отражающих универсальные знания. Неформализованные знания, как правило, не попадают в книги в связи с их субъективностью, приблизительностью. Знания такого рода являются обобщением многолетнего опыта работы и интуиции специалистов.

К неформализованным задачам обычно относят те, которые обладают одной или несколькими из следующих особенностей:

- алгоритмическое решение задачи неизвестно или не может быть использовано из-за ограниченности вычислительных ресурсов;
- задача не может быть определена в числовой форме;
- цели задачи не могут быть выражены в терминах точно определенной целевой функции.

Кроме того, неформализованные задачи обычно обладают следующими особенностями:

- ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о самой решаемой задаче и проблемной области, к которой она относится;
- ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных;
- большой размерностью пространства решений задачи, из чего вытекает необходимость большого числа шагов перебора при поиске решения;
- динамически изменяющимися данными и знаниями.

Также знания делят на процедурные и декларативные. Любые знания можно представить как декларативно, так и процедурно, но, в зависимости от решаемых задач, одни знания удобно иметь в декларативном представлении, а другие в процедурном. Например, факт в декларативном представлении — это просто утверждение, что факт истинен.

Первичными были процедурные знания, т.е. знания, «растворенные» в алгоритмах. Они управляли данными. Для их изменения требуется изменять текст программ. Однако с развитием информатики и ПО все большая часть знаний сосредотачивалась в структурах данных (таблицах, списках, абстрактных типах данных), т.е. увеличивалась роль декларативных знаний.

Процедурное представление факта — это множество инструкций, выполнение которых дает результат, согласующийся с фактом. Если факты, заносимые в БЗ, независимы и неизменяемы, то декларативный подход легче понимается пользователем и довольно просто поддерживается системой, благодаря его модульности. Эксперты и пользователи предпочитают иметь дело с декларативными представлением. Процедурный подход является более эффективным в аналитическом плане, но его сложнее поддерживать в системе. Разработчики ЭС предпочитают именно процедурное представление, т.к. результат работы процедуры всегда легко проверить, прослеживая последовательность ее выполнения. В принципе всегда можно преобразовать декларативное представление в процедурное и наоборот.

Характеристики и особенности экспертных систем (ЭС)

Мы под ИС будем понимать следующее: ИС — это программа для компьютера, которая оперирует со знаниями в определенной предметной области с целью выработки рекомендаций или решения проблем.

Для экспертных систем приняты два основных определения:

- это программы, которые манипулируют знаниями в целях получения удовлетворительного или эффективного решения в узкой предметной области;
- это информационные системы, построенные на принципах искусственного интеллекта, способные в некоторой предметной области

решать задачи, принимать решения и получать выводы, которые может сделать только очень квалифицированный специалист (эксперт).

Экспертная система может полностью взять на себя функции, выполнение которых обычно требует привлечения опыта человека-специалиста, или играть роль ассистента для человека, принимающего решение. Другими словами, система, требующая принятия решения, может получить его непосредственно от программы или через промежуточное звено — человека, который общается с программой. Тот, кто принимает решение, может быть экспертом со своими собственными правами, и в этом случае программа может «оправдать» свое существование, повышая эффективность его работы. Альтернативный вариант — человек, работающий в сотрудничестве с такой программой, может добиться с ее помощью результатов более высокого качества. Вообще говоря, правильное распределение функций между человеком и машиной является одним из ключевых условий высокой эффективности внедрения экспертных систем.

Как и человек-эксперт, экспертная система использует символическую логику и эмпирические правила, чтобы найти решение. Эти системы, как и человек, могут ошибаться, но в отличие от обычных программ, экспертные системы могут учиться на своих ошибках, накапливая и обобщая знания в некоторой проблемной области. У такой искусственной экспертизы есть и преимущества перед человеком: она постоянна, непротиворечива, легко передает свои знания, документируется и уточняется.

ЭС не отвергают и не заменяют традиционного подхода к разработке ПО, они отличаются от традиционных программ тем, что ориентированы на решение трудноформализуемых задач.

ЭС отличаются от традиционных программ тем, что ориентированы на решение неформализованных задач и обладают следующими особенностями:

- алгоритм решения не известен заранее, а строится самой ЭС с помощью символических рассуждений, базирующихся на эвристических приемах;
- ясность полученных решений, то есть ЭС «осознает» в терминах пользователя, как она получила решение;
- способность анализа и объяснения своих действий и знаний;
- способность приобретения новых знаний от пользователя-эксперта, не знающего программирования, и изменения в соответствии с ним своего поведения;
- обеспечение, как правило, естественно-языкового интерфейса с пользователем.

Экспертная система отличается от прочих прикладных программ наличием следующих признаков.

- Моделирует не столько физическую (или иную) природу определенной проблемной области, сколько механизм мышления человека применительно к решению задач в этой проблемной области. Это существенно отличает экспертные системы от систем математического моделирования или компьютерной анимации. Нельзя, конечно, сказать, что программа полностью воспроизводит психологическую модель специалиста в этой предметной области (эксперта), но важно, что основное внимание все-таки уделяется воспроизведению компьютерными средствами методики решения проблем, которая применяется экспертом, т.е. выполнению некоторой части задач так же (или даже лучше), как это делает эксперт.

- Система, помимо выполнения вычислительных операций, формирует определенные соображения и выводы, основываясь на тех знаниях, которыми она располагает. Знания в системе представлены, как правило, на некотором специальном языке и хранятся отдельно от собственно программного кода, который и формирует выводы и соображения. Этот компонент программы принято называть базой знаний.

- При решении задач основными являются эвристические и приближенные методы, которые, в отличие от алгоритмических, не всегда гарантируют успех. Такие методы являются приблизительными в том смысле, что, во-первых, они не требуют исчерпывающей исходной информации, и, во-вторых, существует определенная степень уверенности (или неуверенности) в том, что предлагаемое решение является верным.

Среди систем искусственного интеллекта экспертные системы выделяются двумя особенностями:

- практической направленностью (решают задачи в узкой предметной области);

- понятностью пользователю всех действий экспертной системы благодаря тому, что экспертная система ведет с ним диалог на подмножестве естественного языка и способна объяснить все свои действия.

Важность экспертных систем в современных процессах проектирования решений определяется, в основном, тремя обстоятельствами:

- технология экспертных систем в значительной степени расширяет круг решаемых практически значимых задач, что в свою очередь приносит значительный экономический эффект;

- технология экспертных систем представляет собой одно из важнейших средств решения глобальных проблем проектирования сложных систем при традиционном программировании, таких как длительность и высокая стоимость разработки, высокая стоимость сопровождения их программного обеспечения, повторная используемость программ;

— объединение технологий, экспертных систем и традиционного программирования добавляет новые качества программным продуктам проектирования как в содержательной стороне процесса проектирования, так и в обеспечении, лучшей графики, лучшего интерфейса и взаимодействия с пользователем продукта.

Итак, ЭС предназначены для выполнения специальных, трудно формализуемых задач, решение которых возможно при учете опыта квалифицированных специалистов, являющихся экспертами. Эксперты не только участвуют в выявлении знаний из области разработки, изготовления и эксплуатации изделий АТ, необходимых для решения поставленной задачи, но и выработке правил, на основании которых могут быть предложены приемлемые варианты ее решения. Эти приемы или методы, существенно сокращающие время поиска путем отсечения неперспективных ветвей, не просто методом полного перебора, называются эвристиками и являются одной из разновидностей правил.

Система поддержки принятия решений, СППР, Decision Support System, DSS - компьютерная автоматизированная система, целью которой является помощь людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности.

СППР возникли в результате слияния управленческих информационных систем и систем управления базами данных.

Система поддержки принятия решений предназначена для поддержки многокритериальных решений в сложной информационной среде. При этом под многокритериальностью понимается тот факт, что результаты принимаемых решений оцениваются не по одному, а по совокупности многих показателей (критериев) рассматриваемых одновременно. Информационная сложность определяется необходимостью учета большого объема данных, обработка которых без помощи современной вычислительной техники практически невыполнима. В этих условиях число возможных решений, как правило, весьма велико, и выбор наилучшего из них "на глаз", без всестороннего анализа может приводить к грубым ошибкам.

Система поддержки решений СППР решает две основные задачи:

- выбор наилучшего решения из множества возможных (оптимизация),
- упорядочение возможных решений по предпочтительности (ранжирование).

В обеих задачах первым и наиболее принципиальным моментом является выбор совокупности критериев, на основе которых в дальнейшем будут оцениваться и сопоставляться возможные решения (будем называть их также альтернативами). Система СППР помогает пользователю сделать такой выбор.

Для анализа и выработок предложений в СППР используются разные методы. Это могут быть: - информационный поиск,

- интеллектуальный анализ данных,
- поиск знаний в базах данных,
- рассуждение на основе прецедентов,
- имитационное моделирование,
- эволюционные вычисления и генетические алгоритмы,
- нейронные сети,
- ситуационный анализ,
- когнитивное моделирование и др.

Некоторые из этих методов были разработаны в рамках искусственного интеллекта. Если в основе работы СППР лежат методы искусственного интеллекта, то говорят об интеллектуальной СППР или ИСППР.

Близкие к СППР классы систем — это экспертные системы и автоматизированные системы управления.

Система позволяет решать задачи оперативного и стратегического управления на основе учетных данных о деятельности компании.

Система поддержки принятия решений представляет собой комплекс программных инструментальных средств для анализа данных, моделирования, прогнозирования и принятия управленческих решений, состоящий из собственных разработок корпорации и приобретаемых программных продуктов (Oracle, IBM, Cognos).

Теоретические исследования в области разработки первых систем поддержки принятия решений проводились в технологическом институте Карнеги в конце 50-х начале 60-х годов XX века. Объединить теорию с практикой удалось специалистам из Массачусетского технологического института в 60-х годах. В середине и конце 80-х годов XX столетия стали появляться такие системы, как EIS, GDSS, ODSS. В 1987 году компания Texas Instruments разработала для United Airlines Gate Assignment Display System. Это позволило значительно снизить убытки от полетов и отрегулировать управление различными аэропортами, начиная от Международного аэропорта О'Харе в Чикаго и заканчивая Stapleton в Денвере, штат Колорадо. В 90-х годах сфера возможностей СППР расширялась благодаря внедрению хранилищ данных и инструментов OLAP. Появление новых технологий отчетности сделало СППР незаменимой в менеджменте.

Классификации СППР

По взаимодействию с пользователем выделяют три вида СППР:

- пассивные помогают в процессе принятия решений, но не могут выдвинуть конкретного предложения;

- активные непосредственно участвуют в разработке правильного решения;

- кооперативные предполагают взаимодействие СППР с пользователем. Выдвинутое системой предложение пользователь может доработать, усовершенствовать, а затем отправить обратно в систему для проверки. После этого предложение вновь представляется пользователю, и так до тех пор, пока он не одобрит решение.

По способу поддержки различают:

- модельно-ориентированные СППР, используют в работе доступ к статистическим, финансовым или иным моделям;

- СППР, основанные на коммуникациях, поддерживают работу двух и более пользователей, занимающихся общей задачей;

- СППР, ориентированные на данные, имеют доступ к временным рядам организации. Они используют в работе не только внутренние, но и внешние данные;

- СППР, ориентированные на документы, манипулируют неструктурированной информацией, заключенной в различных электронных форматах;

- СППР, ориентированные на знания, предоставляют специализированные решения проблем, основанные на фактах.

По сфере использования выделяют:

- общесистемные

- настольные СППР.

Общесистемные работают с большими СХД и применяются многими пользователями. Настольные являются небольшими системами и подходят для управления с персонального компьютера одного пользователя.

Архитектура СППР

Функциональные СППР. Являются наиболее простыми с точки зрения архитектуры. Они распространены в организациях, не ставящих перед собой глобальных задач и имеющих невысокий уровень развития информационных технологий. Отличительной особенностью функциональных СППР является то, что анализу подвергаются данные, содержащиеся в файлах операционных систем. Преимуществами подобных СППР являются компактность из-за использования одной платформы и оперативность в связи с отсутствием необходимости перегружать данные в специализированную систему. Из недостатков можно отметить следующие: сужение круга вопросов, решаемых с помощью системы, снижение качества данных из-за отсутствия этапа их очистки, увеличение нагрузки на операционную систему с потенциальной возможностью прекращения ее работы.

СППР, использующие независимые витрины данных. Применяются в крупных организациях, имеющих несколько подразделений, в том числе отделы информационных технологий. Каждая конкретная витрина данных создается для решения определенных задач и ориентирована на отдельный круг пользователей. Это значительно повышает производительность системы. Внедрение подобных структур достаточно просто. Из отрицательных моментов можно отметить то, что данные многократно вводятся в различные витрины, поэтому могут дублироваться. Это повышает затраты на хранение информации и усложняет процедуру унификации. Наполнение витрин данных достаточно сложно в связи с тем, что приходится использовать многочисленные источники. Отсутствует единая картина бизнеса организации, вследствие того что нет окончательной консолидации данных.

СППР на основе двухуровневого хранилища данных. Используется в крупных компаниях, данные которых консолидированы в единую систему. Определения и способы обработки информации в данном случае унифицированы. На обеспечение нормальной работы подобной СППР требуется выделить специализированную команду, которая будет ее обслуживать. Такая архитектура СППР лишена недостатков предыдущей, но в ней нет возможности структурировать данные для отдельных групп пользователей, а также ограничивать доступ к информации. Могут возникнуть трудности с производительностью системы.

СППР на основе трехуровневого хранилища данных. Такие СППР применяют хранилище данных, из которого формируются витрины данных, используемые группами пользователей, решающих сходные задачи. Таким образом, обеспечивается доступ, как к конкретным структурированным данным, так и к единой консолидированной информации. Наполнение витрин данных упрощается ввиду использования проверенных и очищенных данных, находящихся в едином источнике. Имеется корпоративная модель данных. Такие СППР отличает гарантированная производительность. Но существует избыточность данных, которая ведет к росту требований на их хранение. Кроме того, необходимо согласовать подобную архитектуру с множеством областей, имеющих потенциально различные запросы.

Структура СППР

Выделяют четыре основных компонента:

- информационные хранилища данных;
- средства и методы извлечения, обработки и загрузки данных (ETL);
- многомерная база данных и средства анализа OLAP;
- средства Data Mining.

Динамическое моделирование. Особый класс систем стратегического управления и поддержки принятия решений представляют собой системы, позволяющие осуществлять динамическое моделирование процессов. При использовании методов динамического моделирования деятельность компании описывается в виде математической модели, в которой все бизнес-задачи и процессы представляются как система взаимосвязанных вычисляемых показателей.

Решаемые вопросы

СППР позволяет облегчить работу руководителям предприятий и повысить ее эффективность. Они значительно ускоряют решение проблем в бизнесе. СППР способствуют налаживанию межличностного контакта. На их основе можно проводить обучение и подготовку кадров. Данные информационные системы позволяют повысить контроль над деятельностью организации. Наличие четко функционирующей СППР дает большие преимущества по сравнению с конкурирующими структурами. Благодаря предложениям, выдвигаемым СППР, открываются новые подходы к решению повседневных и нестандартных задач.

Использование системы позволяет найти ответы на множество вопросов, возникающих у руководителей компании, например:

У генерального директора:

- На сколько процентов выполнен план по продажам, доходу, прибыли, расходам;
- Какова доля рынка, принадлежащего компании;
- Каковы тенденции развития сегмента рынка, на котором представлена компания;
- Каковы ключевые показатели производительности компании в текущем периоде;
- Каковы тенденции изменения ключевых показателей производительности компании со временем.

У руководителя отдела по работе с партнерами:

- Какие из партнеров приносят наибольший доход, прибыль;
- Какие проекты, группы продуктов лучше всего продает данный партнер;
- Каковы тенденции изменения продаж через партнеров.

У руководителя финансового департамента:

- Сколько каждый проект стоит моему предприятию;
- Сколько стоит поддержка продаваемых проектов;
- Какие проекты в этом году стоят больше, чем в прошлом;
- Как расходы различных подразделений и компании в целом соотносятся с доходами.

У руководителя департамента бюджетного планирования и контроля:

- Насколько точно различные подразделения компании соблюдают установленный бюджет;
- Каковы тенденции расходов по различным подразделениям, статьям бюджета.

У руководителя департамента закупок:

- Какие из моих поставщиков предлагают наилучшее соотношение цена/качество;
- Какие из поставщиков доставляют товары быстрее остальных Медленнее остальных;
- Как часто происходят задержки поставок от того или иного поставщика;
- Каких поставщиков выбрать для поставок крупных/небольших партий продукта.

У руководителя планового отдела (отдела стратегического планирования):

- Насколько предприятие выполняет план по продажам, доходам, прибыли;
- Какие области бизнеса вносят положительный вклад, а какие - отрицательный;
- Каков прогноз ключевых показателей производительности на следующий период (месяц, квартал, год).

У руководителя отдела сервисного обслуживания:

- Каково среднее время выполнения заявки на обслуживание;
- Каковы расходы на выполнение одной заявки;
- Каково среднее время до первой поломки данной модели.

У руководителя отдела кадров:

- Какова производительность персонала, прошедшего определенное обучение перед теми, кто его не проходил;
- Каковы тенденции ежегодного роста персонала компании в различных регионах, подразделениях;
- Каково прогнозируемое количество персонала на следующий год;
- Каковы прогнозы по поводу состава;
- Какие сотрудники нуждаются в обучении;
- Каким набором навыков должен обладать сотрудник чтобы хорошо выполнять свои обязанности.

У руководителя отдела анализа качества:

- Какие проекты доставляются вовремя, а какие - с запозданием;
- Имеют ли определенные клиенты или проекты недопустимо долгий срок поставки;

- Изменилось ли время доставки определенных продуктов со временем;
- Насколько быстрее или медленнее стала поставка продуктов (услуг) в определенный сегмент рынка;
- Каковы основные причины отказа от продукта (услуги).

Процесс создания системы управленческой отчетности, анализа данных и поддержки принятия решений состоит из следующих этапов:

- Анализ существующих на предприятии информационных потоков и процедур управления предприятием;
- Выявление показателей, влияющих на финансово-экономическое состояние предприятия и отражающих эффективность ведения бизнеса (на основе данных из уже используемых систем);
- Выработка процедур, обеспечивающих получение управленческим персоналом необходимой информации в нужное время, в нужном месте и в нужном виде;
- Настройка программных средств многомерного анализа;
- Обучение персонала Заказчика работе с программными средствами многомерного анализа.

Итог – продуманные решения опирающиеся на информационный фундамент, адекватные действия, квалифицированное исполнение и как результат успех всего предприятия.

3 ПРАКТИЧЕСКИЙ РАЗДЕЛ

3.1 Описание лабораторных работ

Тема 1. Современные информационные технологии Лабораторная работа 1 (4 часа). Основы Visual Basic для Excel

Цель работы: изучение возможностей Visual Basic для Excel

Задание 1. Следуя инструкциям приведенным ниже изучить возможности написать простейшую программу на Visual Basic и макрос для Excel

Методические рекомендации по выполнению

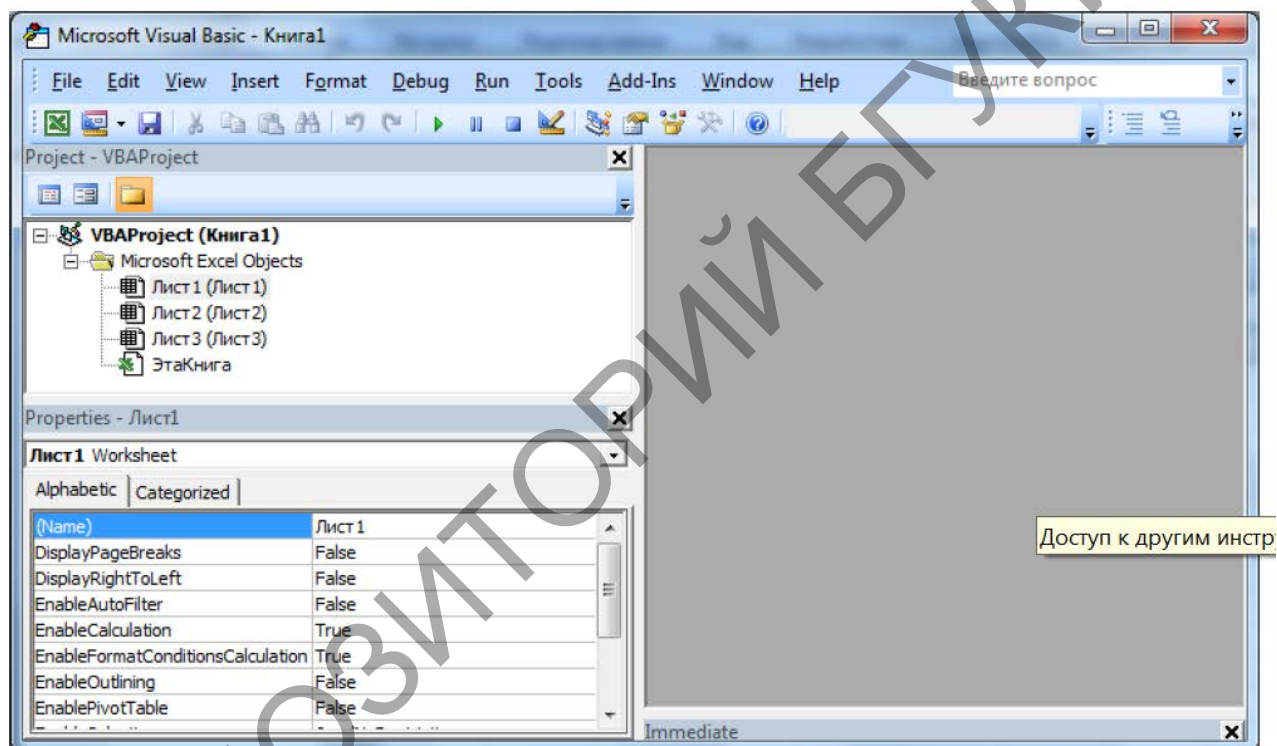
Для входа в среду VB используется вкладка Разработчик окна Excel, которая находится в одном ряду с вкладками Главная, Вставка и др. При ее отсутствии следует выполнить такие операции: • нажать кнопку Office (в левом верхнем углу), кликнув на ней; • нажать кнопку Параметры Excel; • в открывшемся окне Параметры Excel нажать кнопку Основные; • в разделе Основные параметры работы с Excel включить опцию Показывать вкладку «Разработчик» на ленте, поставив соответствующую галочку; • нажать кнопку ОК. При этом в окне Excel появится вкладка Разработчик. В среду Visual Basic войдем следующим образом: • в окне Excel (с книгой) активизируем вкладку Разработчик, кликнув на ней; • выполним команду Visual Basic, кликнув на соответствующей пиктограмме. Эту последовательность операций можно записать так: Разработчик > Visual Basic. В результате откроется окно Microsoft Visual Basic, в котором можно выполнять различные действия: ввод и редактирование текста программы, а также ее отладку и выполнение. (Выполнять программу можно и находясь в окне Excel, но об этом позже.) Программу еще называют приложением, проектом или макросом. Она находится в книге Excel. Рассмотрим основные элементы среды Visual Basic.

1. Строка меню. В ее состав входят все меню, которые используются при создании программы. Среди меню есть стандартные, присущие многим окнам Windows: File (Файл), Edit (Правка), View (Вид), Tools (Сервис), Help (Помощь) и др. Имеются также меню, при помощи которых пользователь компьютера может создавать, запускать и отлаживать программу: Insert (Вставка), Run (Запуск), Debug (Отладка) и др.

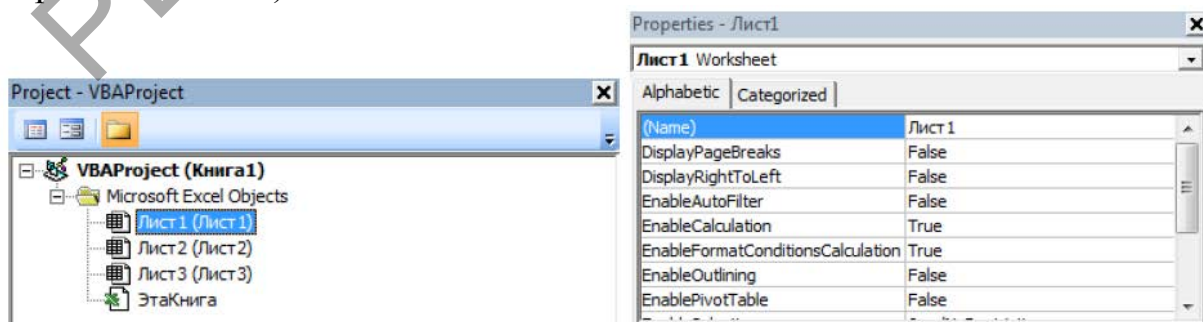
2. Контекстное меню. Служит для удобства выполнения действий, относящихся к той области окна VB, в которой находится указатель

мышки. Окно VB со стандартной панелью инструментов и окнами проекта и свойств. Чтобы вызвать контекстное меню, надо поместить указатель мышки в нужную область экрана и кликнуть правой кнопкой мышки. После этого следует кликнуть левой кнопкой мышки на нужной команде открывшегося списка.

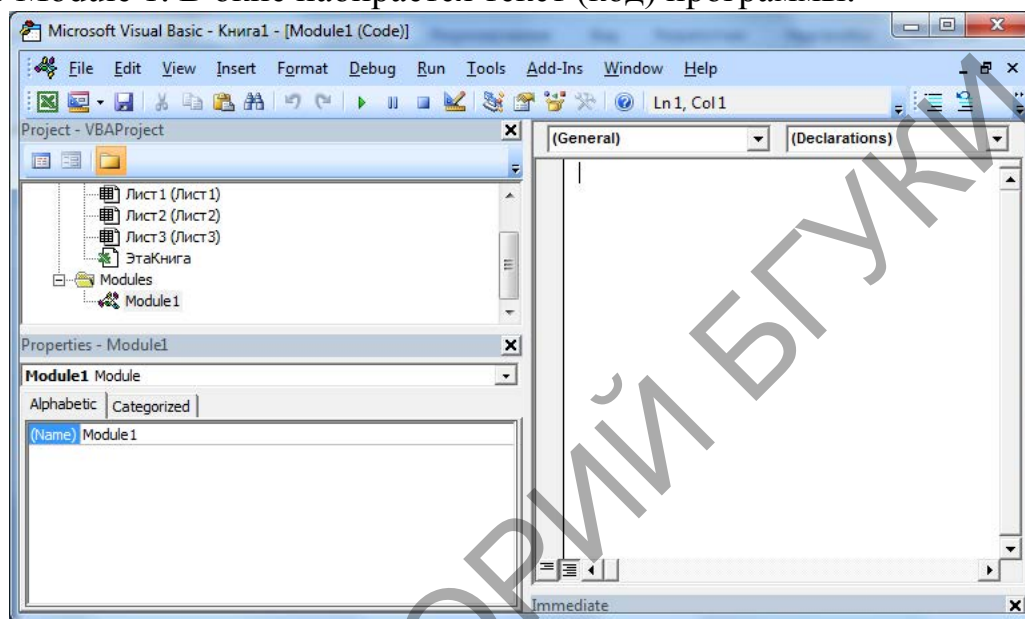
3. Панели инструментов: Standard (Стандарт), Edit (Правка), Debug (Отладка) и др. По умолчанию отображается только стандартная панель инструментов Standard (Стандарт). Для того чтобы добавить или удалить какую-либо панель, следует выполнить View (Вид) > Toolbars (Панели инструментов) и кликнуть на нужной команде открывшегося списка. Галочка свидетельствует о наличии панели на экране.



В окне Excel выполним Разработчик > Visual Basic. В результате появится окно VB (рис. 1.1), включающее в себя окна проекта (Project Explorer) и свойств (Properties Window).



Если окно проекта отсутствует, то для его вывода на экран следует кликнуть на соответствующем пункте меню View (Вид). Окно свойств нам потребуется позже. 2. В окне проекта кликнем на строке VBAProject (Книга1), где Книга1 – имя активной книги. При этом названная строка выделится. 3. Выполним Insert (Вставка) > Module (Модуль). При этом изменится содержимое окна проекта — появится строка, соответствующая вставленному модулю Module 1. Кроме того, откроется пустое окно, которое называется окном кода, соответствующим модулю Module 1. В окне набирается текст (код) программы.



Для открытия окна кода, соответствующего модулю, вставленному ранее, надо дважды кликнуть на имени этого модуля в окне проекта. Для удаления модуля следует выполнить такие операции: 1) кликнуть правой кнопкой мышки на имени этого модуля в окне проекта, например, на Module1; 2) в открывшемся контекстном меню выполнить команду Remove (Удалить); 3) кликнуть на кнопке No (Нет) в открывшемся окне с вопросом, экспортировать модуль перед удалением или нет. Как уже говорилось, в окне кода набирается текст программы. Первая и последняя строки (операторы) программы стандартные:

```
Sub имя ()  
End Sub
```

Имя программы имя назначается ее разработчиком. Оно должно удовлетворять следующим условиям: • первый символ имени должен быть буквой; • имя может содержать только буквы, цифры и символ соединения _; • имя не должно содержать более 255 символов. Между первой и последней строками набираются остальные строки (операторы) программы. При этом можно пользоваться привычными командами редактирования (как в текстовом процессоре Word), а также буфером обмена. Ввод строки оканчивается нажатием клавиши Enter. В качестве 1 примера рассмотрим программу расчета

длины гипотенузы прямоугольного треугольника с катетами $a = 3$ и $b = 4$, основанную на теореме Пифагора. В окно кода введем (с клавиатуры) следующий текст программы:

```
Sub Пифагор()  
a = 3  
b = 4  
c = Sqr(a ^ 2 + b ^ 2)  
End Sub
```

Слово Пифагор набрано русскими буквами; все остальные буквы – английские

Видно, что операция возведения в степень выглядит, как в Excel; Sqr — функция извлечения квадратного корня, как в английской версии Excel (в русской версии Excel функция извлечения квадратного корня называется КОРЕНЬ). Если для наглядности оператор (следует разместить на нескольких строках, то для переноса следует использовать символ пробела с последующим символом соединения _ (после ввода этих символов надо нажать клавишу Enter). Если несколько операторов следует разместить на одной строке, то между ними надо поставить двоеточие и символ пробела. Апостроф (запятая в верхней части строки) означает, что следующая за ним информация (до конца строки) является комментарием, т. е. набором символов, который никак не влияет на выполнение программы. Таким образом, нашу программу можно записать в виде

```
Sub Пифагор()  
a = 3: b = 4  
c = Sqr(a ^ 2 + b ^ 2) 'согласно теореме Пифагора  
MsgBox ("Длина гипотенузы = " & c)  
End Sub
```

После набора текста программы следует этап отладки, т. е. выявления и исправления ошибок в программе. При этом используется отладчик. Команды отладчика можно увидеть, если в среде VB открыть меню Debug (Отладка). Рассмотрим основные команды отладчика. 1. Step Into (Шаг с заходом) — выполнение одного оператора программы или его части. Эта команда также выполняется при нажатии клавиши F8. Ее можно использовать для пошагового выполнения программы. 2. Run To Cursor (Выполнить до текущей позиции) — выполнение программы до мигающего курсора. (Для установки мигающего курсора нужно кликнуть в нужном месте программы.) Рассматриваемая команда выполняется также при нажатии клавиш Ctrl + F8. 3. Toggle Breakpoint (Точка останова) — установка или ликвидация точки останова. Такой точкой отмечается строка в программе, перед которой выполнение программы временно прекращается. Установка или ликвидация точки останова

производится в том месте, где находится мигающий курсор. Эта команда также выполняется при нажатии клавиши F9. Для установки или ликвидации точки останова можно кликнуть на серой левой границе окна кода напротив нужной строки. 4. Clear All Breakpoints (Снять все точки останова) — ликвидация всех точек останова. 5. Add Watch (Добавить контрольное значение) — визуализация текущих значений выбранных переменных. Методика использования этой команды будет рассмотрена позже. Следующие две команды входят в состав меню Run (Запуск). 1. Run (Запуск) — запуск программы и переход от одной точки останова к другой. Если точек останова нет, то программа выполняется полностью. Эта команда изображена стрелкой на панелях инструментов, в частности на стандартной панели. 2. Reset (Сброс) — прекращение выполнения программы. Эта команда изображена квадратом на панелях инструментов. При остановках во время выполнения программы стрелкой слева и желтым цветом выделяется оператор, который еще не выполнен. Значение той или иной переменной можно увидеть, подведя к ней указатель мышки. Рассчитанное значение переменной c в программе Пифагор можно увидеть с помощью отладчика. Для этого выполним следующие операции.

Для того чтобы окно вывода значения нам не мешало, прокомментируем строку с этим оператором. Установите в строку MsgBox и затем нажмите на панели инструментов кнопку - Comment Block.

1. Отметим точкой останова последнюю строку программы, например, кликнув на серой левой границе окна кода напротив этой строки/

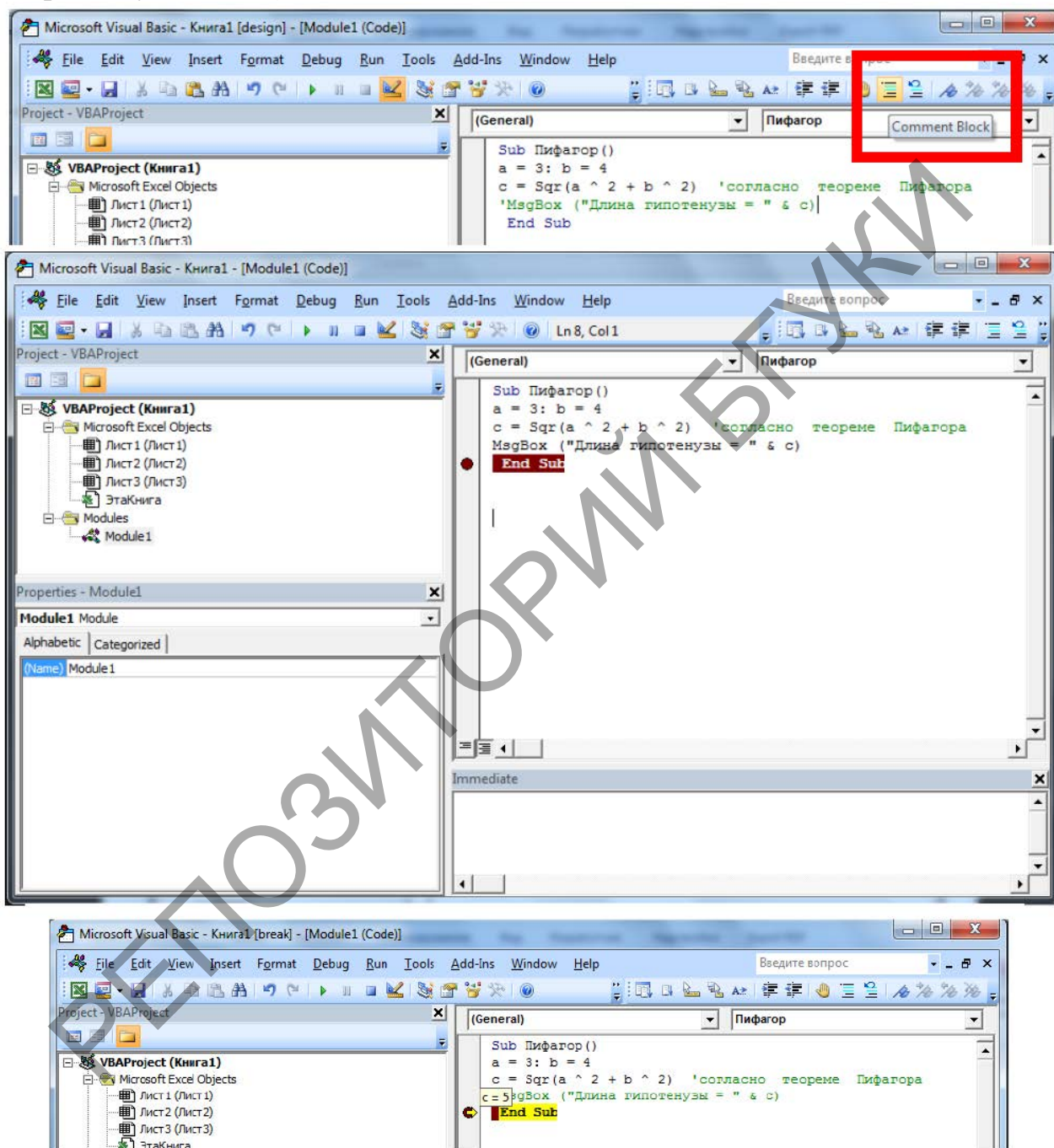
2. Кликнем по стрелке на панели инструментов, для выполнения программы до точки останова.

3. В открывшемся окне Macros (Макрос) нажмем кнопку Run (Выполнить).

4. Подведем указатель мышки к переменной c в тексте программы; при этом высветится $c = 5/$

5. Кликнем по стрелке на панели инструментов, для окончания выполнения программы. Выполнение программы означает последовательное выполнение ее операторов: сначала компьютер задает значения a и b , а затем рассчитывает значение c . В этом легко убедиться, если выполнить следующее. Задание: уберите точку останова (кликнув на ней) и произведите пошаговое выполнение программы Пифагор с помощью клавиши F8, наблюдая за изменением значений переменных a , b и c . Перед первым нажатием клавиши F8 мигающий курсор должен находиться в тексте программы. Выполняемая программа находится в оперативной памяти компьютера. По-английски оперативная память называется «main memory», что переводится как «основная память». Такое название объясняется тем, что оперативное запоминающее

устройство непосредственно связано с процессором и содержит данные, непосредственно участвующие в операциях. Процессор выполняет операции (в частности арифметические) и управляет работой компьютера (в частности считыванием информации из оперативной памяти и записью информации в оперативную память).



Запись макроса

Если нажать кнопку Макрос на вкладке Разработчик, открывается диалоговое окно Макрос, которое предоставляет доступ к подпрограммам или макросам VBA, которые можно использовать из определенного документа или приложения. Кнопка Visual Basic открывает редактор Visual Basic, где можно

создавать и редактировать код VBA. Также на вкладке Разработчик в Word 2010 и Excel 2010 размещена кнопка Записать макрос, которая автоматически создает код VBA, позволяющий воспроизводить действия, выполненные пользователем в приложении. Запись макроса — это великолепное средство, которое можно

использовать для изучения VBA. Читая сгенерированный код, можно понять язык VBA и объединить знания пользователя и программиста Office 2010. Единственное, что нужно помнить — это то, что сгенерированный код может быть запутанным, так как редактор макросов использует некоторые предположения о намерениях пользователя, которые не всегда точны. Запись макроса

1. Откройте Excel 2010, создайте рабочую книгу и щелкните вкладку Разработчик на ленте. Нажмите кнопку Записать макроса и примите все параметры по умолчанию в диалоговом окне Запись макроса, в том числе Macro1 в качестве имени макроса и This Workbook в качестве расположения.

2. Нажмите кнопку ОК, чтобы начать запись макроса. Заметьте, что текст кнопки изменяется на Остановить запись. Нажмите эту кнопку, когда выполните все записываемые действия.

3. Щелкните ячейку B1 и введите классическую первую строку программиста: Hello World. Прекратите ввод текста и посмотрите на кнопку Остановить запись; она отключена, так как Excel 2010 ждет, пока пользователь завершит ввод значения ячейки.

4. Щелкните ячейку B2, чтобы завершить действие в ячейке B1, а затем нажмите кнопку Остановить запись.

5. Щелкните Макросы на вкладке Разработчик, выберите макрос Macro1, если он еще не выбран, и щелкните Правка, чтобы просмотреть код Macro1 в редакторе Visual Basic. Созданный макрос должен выглядеть следующим образом. Код

Код

```
Sub Macro1()  
.  
' Macro1 Macro  
.  
.  
    Range("B1").Select  
    ActiveCell.FormulaR1C1 = "Hello World"  
    Range("B2").Select  
End Sub
```

Учтите схожие моменты с предыдущим примером кода, в котором выбирается ячейка A1, и отличия от него. В этом примере кода выбирается ячейка B1, а затем строка "Hello World" записывается в активную ячейку.

Кавычки вокруг текста обозначают строковое значение (в противоположность числовому значению).

Вспомните, что чтобы отобразить кнопку Остановить запись, нужно было щелкнуть ячейку B2. Это действие также задается в строке кода. Средство записи макроса записывает каждое нажатие клавиши.

Строки кода, начинающиеся на апостроф и выделенные зеленым цветом — это комментарии, которые поясняют код или напоминают другим программистам его предназначение. VBA игнорирует любую строку или ее часть, которые начинаются с одинарной кавычки. Написание понятных и подходящих комментариев в коде очень важно, но в этой статье данная тема не затрагивается. В следующих ссылках на этот код в данной статье эти четыре строки комментариев отсутствуют. Когда средство записи макроса генерирует код, оно использует сложный алгоритм для определения нужных методов и свойств. Если данное свойство незнакомо, пользователю могут помочь множество ресурсов. Например, в записанном макросе средство записи макроса сгенерировало код, который ссылается на свойство `ForumulaR1C1`.

Использование справки разработчика Выберите `ForumulaR1C1` в записанном макросе и нажмите F1. Система справки запустит быстрый поиск, определит, что нужные темы есть в разделе "Разработчик" справки Excel 2010, и укажет свойство `FormulaR1C1`. Можно щелкнуть ссылку, чтобы прочитать описание свойства, но прежде обратите внимание на ссылку Справочник по объектной модели Excel рядом с нижней частью окна. Щелкните ее, чтобы просмотреть список объектов, которые Excel 2010 использует в объектной модели для описания рабочих листов и их компонентов. Щелкните любой из них, чтобы просмотреть свойства и методы, которые применяются к этому объекту, а также ссылки на другие связанные с ними параметры. Во многих записях справки содержатся короткие примеры кода, которые могут быть полезны. Например, можно перейти по ссылкам в описании объекта `Borders`, чтобы узнать, как задать границу в VBA.

```
Worksheets(1).Range("A1").Borders.LineStyle = xlDouble
```

Редактирование кода

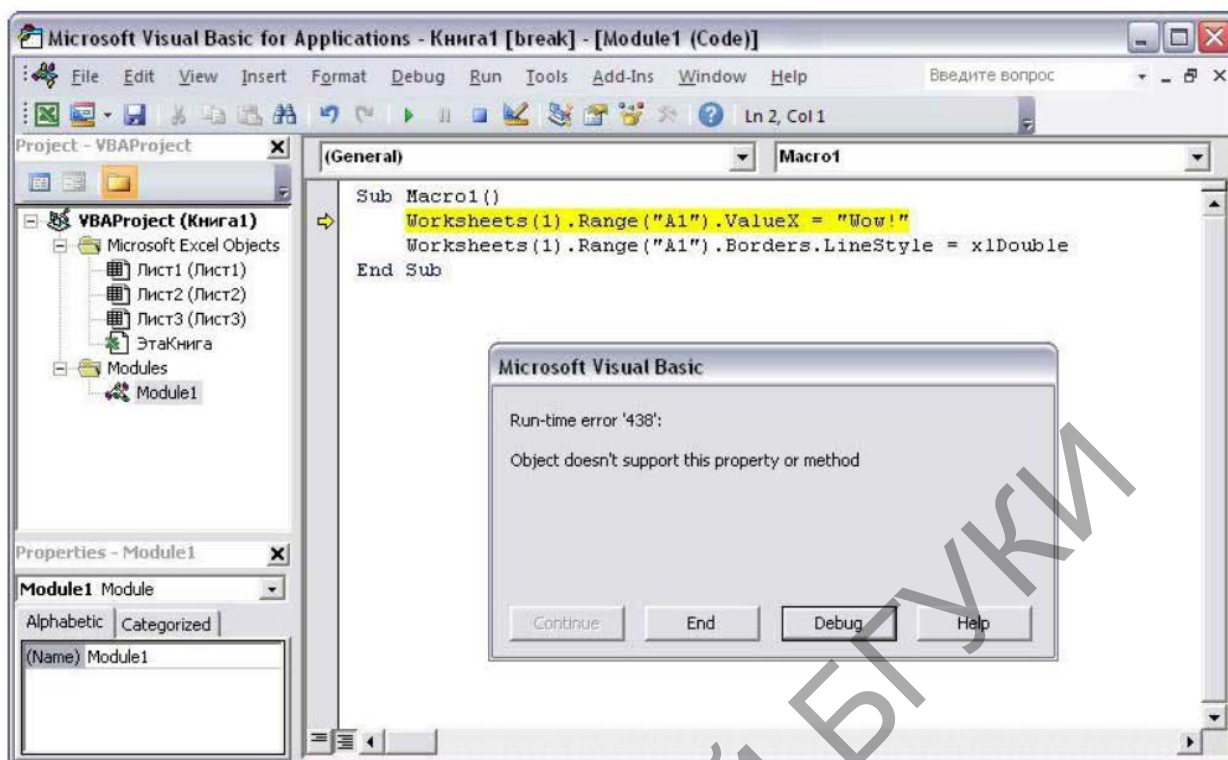
Код с границами отличается от записанного макроса. В объектной модели есть несколько способов адресации любого объекта, ячейки A1 в этом примере, что может вызывать затруднения. Иногда самый лучший способ изучить программирование — вносить небольшие изменения в определенный рабочий

код и смотреть, что получается. Попробуйте сделать это сейчас. Откройте Macro1 в редакторе Visual Basic и измените код следующим образом: Код

```
Sub Macro1()  
Worksheets(1).Range("A1").Value = "Wow!"  
Worksheets(1).Range("A1").Borders.LineStyle = xlDouble  
End Sub
```

Не нужно сохранять код, чтобы запустить его, поэтому вернитесь в документ Excel 2010, щелкните Макросы на вкладке Разработчик, выберите Macro1 и нажмите кнопку Запустить. Теперь в ячейке A1 содержится текст Wow!, а вокруг него размещена граница из двойных линий.

Существует два основных вида ошибок программирования: синтаксические ошибки, которые нарушают грамматические правила языка программирования и ошибки времени выполнения, которые синтаксически правильны, но вызывают сбой, когда VBA пытается выполнить код. Хотя исправлять синтаксические ошибки неприятно, их легко обнаружить; редактор Visual Basic выдает сопровождающееся звуковым сигналом сообщение об ошибке и меняет цвет текста, если при вводе кода найдена синтаксическая ошибка. Например, в VBA строковые значения должны быть заключены в двойные кавычки. Чтобы узнать, что происходит при использовании одинарных кавычек, вернитесь в редактор Visual Basic и замените строку "Wow!" на 'Wow!' (т. е. слово Wow в одинарных кавычках). Если щелкнуть следующую строку, среагирует редактор Visual Basic. Сообщение об ошибке "Ошибка компиляции. Ожидается: выражение" не особо помогает, но строка, содержащая ошибку, становится красной, что говорит о синтаксической ошибке в этой строке. В результате программа не будет запускаться. Нажмите кнопку ОК и измените текст на "Wow!". Ошибки времени выполнения обнаружить сложнее, так как синтаксически все выглядит правильно, но при попытке выполнить код возникает сбой. Например, откройте редактор Visual Basic и измените имя свойства Value на ValueX в макросе, намеренно вызывая ошибку времени выполнения, так как у объекта Range нет свойства ValueX. Вернитесь в документ Excel 2010, откройте диалоговое окно Макросы и запустите Macro1 еще раз. Появится сообщение Visual Basic, в котором описывается ошибка: "Объект не поддерживает это свойство метода". Хотя текст и так понятен, нажмите кнопку "Отладка" для получения дополнительных сведений. После возвращения в редактор Visual Basic будет включен специальный режим отладки, в котором желтым цветом выделяется строка кода с ошибкой. Как и ожидалось, желтым выделена строка со свойством ValueX.



Можно внести изменения в исполняемый код VBA, поэтому измените ValueX на Value и нажмите маленькую зеленую кнопку воспроизведения в меню Отладка. Программа должна запускаться без ошибок. РЕЗЮМЕ Только что вы записали макрос, прочитали документацию по объектной модели и написали простую программу на VBA, которая что-то делает. Поздравляем!

Задание 2. Написать макрос выводящий в ячейки листа 10 членов последовательности Фибоначчи.

Задание 3. Написать программу, рассчитывающую площадь круга с заданным радиусом и выводящим значение в ячейку.

Тема 2. Основные программные средства информационных технологий

Лабораторная работа 1

Подготовка документов сложной структуры

Цель работы: изучение возможностей современных текстовых редакторов по подготовке документов сложной структуры

Задание 1. Подготовка документа сложной структуры в текстовом редакторе

Методические рекомендации по выполнению

Для работы используйте любой готовый документ сложной структуры (например, курсовую работу).

1. Установите следующие параметры страницы: размер бумаги – А4, ориентация – книжная, поля: левое – 2 см; правое – 1 см; верхнее – 2 см; нижнее – 2 см, интервал от края до верхнего колонтитула 1,2 см (в верхнем колонтитуле будет размещаться номер печатной страницы).

2. Выполните форматирование большого документа сложной структуры (например, курсовой работы) с помощью стилей.

2.1. Создайте стили для форматирования основных структурных элементов документа (заголовков глав, разделов, подразделов и основного текста) в соответствии с требованиями к оформлению дипломных работ. Выписка из требований представлена в файле *pr3_0.docx*. При создании нового стиля следует учитывать:

- соблюдайте иерархию: заголовок для глав – заголовок для параграфа (раздела) – заголовок для подпараграфа (подраздела) – заголовок для основного текста; все стили добавьте в шаблон; это позволит в дальнейшем совмещать набор и правильное форматирование текста;

- стиль основного текста должен быть основан на стиле – *Обычный*, стиль следующего абзаца – этот же;

- заголовок подпараграфа – основан на стиле *Заголовок 3*, следующий стиль – созданный для основного текста;

- заголовок параграфа – основан на стиле *Заголовок 2*, следующий стиль – созданный для подпараграфа;

- заголовок главы – основан на стиле *Заголовок 1*, следующий стиль – созданный для параграфа.

2.2. При необходимости исправьте титульный лист.

2.3. На 2 страницу документа поместите оглавление (вкладка – *Оглавление*, меню *Ссылки*).

2239724

2.4. Удалите нумерацию с первых двух страниц. Для этого в конце страницы 2 установите *Разрыв раздела* (Разметка страницы) и установите разные колонтитулы для разделов (вкладка *Работа с колонтитулами* – *Конструктор* кнопка *Как в предыдущем разделе*).

3. Результат выполнения задания покажите преподавателю.

4. Проверьте работу иерархии стилей. Введите любой текст. Сначала заголовок, отформатируйте его созданным стилем для заголовков. Затем с новой строки введите название параграфа, подпараграфа, основной текст ...

Задание 2. Создание и обновление ссылок на список использованной литературы

Методические рекомендации по выполнению

1. В отформатированном файле выделить существующий список использованной литературы.

2. Выполнить команду меню *Ссылки - Список литературы - Сохранить выделенный фрагмент в коллекцию списков литературы*.

3. В диалоговом окне *Создание нового стандартного блока* задать следующие значения полей:

- имя – ввести имя списка литературы;
- коллекция – Списки литературы;
- категория – Общие, Описание (не обязательное поле);
- сохранить в – Building Blocks.dotx;
- параметры – Вставить содержимое в тот же абзац;
- нажать кнопку *Ок*;

Перекрестные ссылки предназначены для связи объекта ссылки с абзацем, заголовком структурной части документа, закладкой, сноской, рисунком, таблицей, формулой. В зависимости от типа объекта изменяется содержание ссылки.

4. В тексте выделите числа в квадратных скобках, которые указывают на номер ссылки в списке источников литературы

5. Выполнить команду меню *Вставка - Перекрестная ссылка*;

- выбрать *Тип ссылки*;
- выбрать вид ссылки – *Вставить ссылку на*;
- указать способ вставки – *Как гиперссылка*;
- добавить слова «*выше*» или «*ниже*»;
- нажать кнопку *Вставить*.

6. Обновление значений перекрестных ссылок. Обновление – изменение значения полей в результате подстановки новых данных. Значения перекрестных ссылок можно обновить по одному и в совокупности все имеющиеся в документе.

Добавьте новые источники в конец списка литературы. В тексте поставьте ссылку на источник и создайте перекрестную ссылку как описано выше.

7. Выполните сортировку данных списка литературы.

Способ обновления вручную:

- нажать клавишу *F9* для отдельно выделенной ссылки;
- для обновления всех ссылок нужно сначала их выделить *Правка - Выделить все*, затем нажать клавишу *F9*.

Способ автоматического обновления:

– при печати: установить обновление полей (*Сервис* → *Параметры Word* → *Экран* → *Обновлять поля перед печатью, Обновлять связанные данные перед печатью*).

Лабораторная работа 2

Шаблоны документов в текстовых редакторах

Цель работы: изучение шаблонов современных текстовых редакторов и их использование в деятельности культуролога-менеджера

Задание. Создание резюме

Методические рекомендации по выполнению

1. Используя встроенные шаблоны Microsoft Word, создайте резюме.
2. Добавьте новые поля, например «Рекомендации». Заголовки новых полей отформатируйте с помощью инструмента «Формат по образцу».
3. Выполните поиск шаблонов для резюме в сети интернет и заполните его.

Лабораторная работа 3

Системы подготовки презентаций

Цель работы: приобретение основных умений и навыков создания гармоничной презентации с элементами нелинейности и интерактивности, используя триггеры и гипертекст.

Методические рекомендации по выполнению

Общие рекомендации по оформлению контента в презентации

Стиль

- соблюдайте единый стиль оформления;
- избегайте стилей, которые будут отвлекать от содержания;
- вспомогательная информация не должна преобладать над основной;
- используйте на всех страницах одинаковый стиль оформления (рамки, разделительные линии, размещение заголовков и др.).

Фон

- фон должен быть однотонным;
- лучше использовать один фон для всех страниц (исключения: титульная и заключительная);
- лучше использовать светлые холодные тона;
- если необходимо имитировать традиционную меловую доску, то можно использовать темно-зеленый или темно-коричневый цвет фона (при этом цвет основного текста должен быть белым);

– не рекомендуется использовать красный фон (только в исключительных случаях).

Цвет

– на одной странице рекомендуется использовать не более трех цветов: один для фона, один для заголовка, один для текста;

– лучше использовать для фона и текста контрастные цвета;

– необходимо обратить внимание на цвет гиперссылок (до и после использования).

Анимационные эффекты

– допускается использование анимации для акцентирования, привлечения внимания к информации на странице;

– не стоит злоупотреблять различными анимационными эффектами, они не должны отвлекать внимание от содержания страницы;

– на одной странице можно использовать не более одного эффекта;

– не следует применять «движущиеся» строки по горизонтали и вертикали;

Содержание информации

– используйте короткие слова и предложения;

– минимизируйте количество предлогов, наречий, прилагательных;

– заголовки должны быть короткими и простыми;

– заголовки должны привлекать внимание учащихся.

Расположение информации на странице

– предпочтительно горизонтальное расположение информации;

– наиболее важная информация должна располагаться в центре экрана;

– если на слайде располагается графический объект (рисунок, фотография и т.п.), надпись должна располагаться под ним.

Шрифты

– для заголовков – не менее 22;

– для основной информации не менее 18;

– шрифты без засечек легче читать с большого расстояния (Arial, Tahoma, Verdana и т.п.);

– не рекомендуется смешивать разные типы шрифтов в одном учебном контенте, тем более на одной странице;

– для выделения информации следует использовать жирный шрифт, курсив, другой цвет букв или подчеркивание;

– не рекомендуется злоупотреблять прописными буквами (они читаются хуже строчных);

– не следует использовать узкое и (или) курсивное начертание шрифта для основного текста.

Выделение информации

– используйте средства выделения для акцентирования внимания учащихся на ключевых моментах (понятиях, объектах): 1. рамки, границы, заливка 2. штриховка, стрелки 3. рисунки, диаграммы, схемы для иллюстрации наиболее важных фактов

– не злоупотребляйте средствами выделения информации

Объем информации

– не стоит заполнять страницу большим объемом информации: одновременно можно запомнить не более двух, трех, четырех фактов, выводов, определений;

– старайтесь передать одну мысль в одной строке;

– в каждой строке должно быть максимум 6 слов (40 символов);

– на одной странице должно быть не более 4-7 строк текста, без учета заголовков;

– в списках должно быть не более 5-6 элементов;

– наибольшая эффективность достигается тогда, когда ключевые пункты отображаются по одному на каждой странице;

– количество страниц с текстовой и/или знаковой информацией не должно быть больше 15-20% общего количества страниц учебного контента (без учета динамических видеоматериалов).

Вид подачи информации

– Для обеспечения разнообразия следует использовать разные виды представления контента: текст, таблицы, диаграммы, графики, рисунки, фотографии.

Задание Создайте презентацию из 10 слайдов с элементами нелинейности и интерактивности, используя триггеры и гипертекст

Порядок выполнения

1. Запустите программу PowerPoint
2. После титульного слайда создайте слайд с содержанием вашей презентации.

3. Создайте слайды по содержанию.

4. Организуйте переходы посредством гиперссылки с раздела содержания на слайд и возврата к содержанию:

4.1. Выделить объект, пункт меню ВСТАВКА – ГИПЕРССЫЛКА, выбрать место в документе и указать слайд.

4.2. Создать кнопку на слайде начала раздела, пункт меню ВСТАВКА – ГИПЕРССЫЛКА, выбрать место в документе и указать слайд Содержание.

5. Создать тест по содержанию презентации с использованием триггеров:

5.1. Выделить объект, добавить к объекту анимацию, задать исполнение анимации по триггеру (переключателю) с назначением либо этого, либо другого объекта слайда. Анимация должна быть задана по щелчку. Добавить анимацию к объекту с параметром После предыдущего.

6. Оформить презентацию в соответствии с перечисленными рекомендациями. Тему презентации выбрать по содержанию программы дисциплины.

Лабораторная работа 4

Способы хранения и обработки графической информации

Цель: углубление основных знаний и умений по созданию графической продукции в контексте задач социокультурной деятельности.

Краткая теория

Мультимедиа (смотри семинар «Программно-компьютерные среды для работы с медиаприложениями»).

Компьютерная графика – область деятельности, связанная с созданием и обработкой цифровых изображений.

Компьютерная графика – область информатики, занимающаяся проблемами получения, представления, обработки и отображения различных изображений (рисунков, чертежей, мультипликации) при помощи компьютера.

Компьютерная графика – процесс создания, обработки и вывода изображений разного рода с помощью компьютера.

Компьютерная графика – вид искусства.

Растровая графика – компьютерная графика, в которой изображение представляется двумерным массивом точек (элементов раstra), цвет и яркость каждой из которых задается независимо. Программные средства: Adobe Photoshop, Corel Photo-Paint, Microsoft Paint, GNU Image Manipulation Program и др.

Векторная графика – компьютерная графика, в которой изображения описываются в виде математических объектов – «контуров». Каждый контур представляет собой независимый объект, который можно перемещать, масштабировать и изменять. Векторную графику часто называют также объектно-ориентированной графикой. Программные средства: CorelDRAW, Adobe Illustrator и Xara Designer и др.

Фрактальная графика – компьютерная графика, в которой изображения описываются в виде математических процедур построения «фракталов». Фрактал – это бесконечно самоподобная геометрическая фигура. Бесконечное самоподобие означает, что в любом фрагменте фигуры найдется

часть, подобная всей фигуре. Структура фрактала не меняется при изменении масштаба. Поэтому в памяти компьютера хранится информация только об этой структуре и алгоритм ее масштабирования. Программные средства: Fractint, Fractal Explorer, Arophysis и др.

Формат – способ организации цифровых данных в файле. Графические форматы служат для цифрового кодирования изображений.

По типу «хранимой» графики различают следующие форматы:

- растровые (PNG, TIFF, GIF, BMP, JPEG и др.);
- векторные (AI, CDR, DXF и др.);
- смешанные (EPS, PDF и др.).

Название формата обычно совпадает с расширением файла.

Задание 1

1. Проанализируйте возможности графических программ. Сделайте вывод о целесообразности их применения для создания, заполнив таблицу

Анализируемый объект	Растровая графика	Векторная графика
Основной элемент изображения		
Изменение качества изображения при масштабировании		
Соотношение цвета и формы (отделены или неотделены)		
Фотореалистичность		
Форматы		
Графические программы		
Примеры применения в деятельности культуролога		

2. Продумайте тематику социокультурной программы. Разработайте концепцию медиатекста социокультурной программы. Продумайте цветовое, шрифтовое и композиционное решение графической продукции.

3. В программе векторной графике создайте логотип социокультурного субъекта или объекта социокультурной программы.

4. В программе растровой графики создайте афишу и пригласительный билет социокультурной программы с включением логотипа.

5. Обоснуйте выбор средств выражения основной идеи

Средство	Решение	Смысловая нагрузка, цель
Цветовое решение фона шрифта		
Тип шрифта, размер		
Элементы композиции 1. 2.		

Задание 2. Проанализируйте возможности использования различных видов графики и графических форматов в профессиональной деятельности культуролога.

Задача 1.

1. Проанализируйте возможности растровой, векторной и фрактальной графики. По результатам анализа сделайте вывод о целесообразности использования различных видов графики в профессиональной деятельности культуролога.

2. Используя графические редакторы Microsoft Paint, Adobe Photoshop, CorelDraw, проанализируйте их возможности.

3. Проанализируйте свойства и возможности цветовых моделей, используемых в компьютерной графике. Заполните таблицу 1.1.

Таблица 1.1

Анализ цветовых моделей

	RGB	CMYK	LAB	HSB
Способ получения цвета				
Для какого класса устройств предназначена				
Аппаратная зависимость				
Цветовой охват				
Область применения				

1. Проанализируйте возможности, которые поддерживают различные графические форматы. Заполните таблицу 1.2.

Анализ графических форматов

Формат	Слои	Цветовая модель	Прозрачность	Сохранение анимации
BMP				
JPG				
GIF				
PNG				
TIFF				
PSD				

Задача 3.

1. Найдите и изучите информационные ресурсы, содержанием которых являются базы графических изображений и технологии их создания.

2. Составьте аннотированное описание каждого из изученных информационных ресурсов.

3. Проанализируйте информационные ресурсы по возможности их использования в профессиональной деятельности культуролога (результаты представить в виде таблицы).

Задача 4. Оформите отчет.

Задача 5. Оцените ваши личные успехи при выполнении данной работы и предъявите результаты преподавателю.

Литература

1. Гурский, Ю. Компьютерная графика Photoshop CS5, CorelDRAW X5, Illustrator CS5. Трюки и эффекты / Ю. Гурский, А. Жвалевский, В. Завгородний. – СПб.: Питер, 2011. – 704 с.

2. Немцова, Т.И. Компьютерная графика и Web-дизайн. Практикум по информатике (+ CD-ROM) / Т.И. Немцова, Ю.В. Назарова. – М.: Форум: Инфра-М, 2010. – 288 с.

3. Петров, М.Н. Компьютерная графика (+ CD-ROM) / М.Н. Петров. – СПб.: Питер, 2011. – 544 с.

4. Тимофеев, С.М. Новейшая энциклопедия компьютерной графики / С.М. Тимофеев. – М.: Эксмо-пресс. – 2009. – 512 с.

Вопросы для самоконтроля

1. Какое программное обеспечение вы будете использовать для создания транспаранта большого размера? Обоснуйте свой выбор.

2. На вашем компьютере установлены: Microsoft Paint, Adobe Photoshop, CorelDraw. Какое программное обеспечение вы будете использовать для создания анимации?

3. Приведите примеры использования графических редакторов в профессиональной деятельности культуролога?

4. Какие из сетевых информационных ресурсов, содержащих базы данных графических медиатекстов и материалы по их созданию, будут наиболее полезны в вашей профессиональной деятельности?

5. В каких случаях вы будете использовать редакторы фрактальной графики в своей профессиональной деятельности?

Лабораторная работа 5

Способы представления и обработки звуковой информации

Цель. Углубить основные навыки записи и микширования звука с другими звуковыми фрагментами в контексте задач социокультурной деятельности.

Задание. Создайте звуковой медиатекст

Порядок выполнения

1. Заполните таблицу основных характеристик звука.

Параметр	Пояснение	Единицы измерения
Громкость		
Высота		
Тембр		
Основной тон		
Обертон		

2. Продумайте содержание текста.

3. Подключите микрофон к компьютеру. Настройте громкость громкости микрофона для записи. Для этого нужно в правом нижнем углу рабочего стола нажать на иконке с изображением динамика. Появится стандартный микшер Windows, в котором нужно в меню выбрать "параметры".

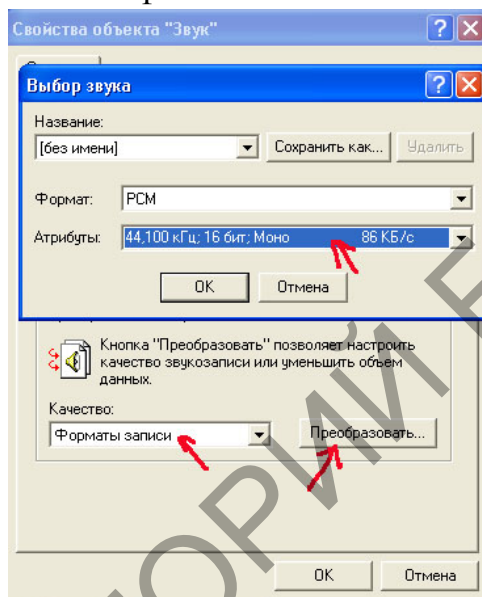
4. В пункте Свойства выбираем Запись.

5. В появившемся окне измененного микшера отображены все способы и источники записи, в данном случае нам нужно записать голос с

использованием микрофона, соответственно выбираем Микрофон (Microphone) и устанавливаем громкость на максимум (за исключением особо редких случаев при использовании особо чувствительных микрофонов голос может искажаться).

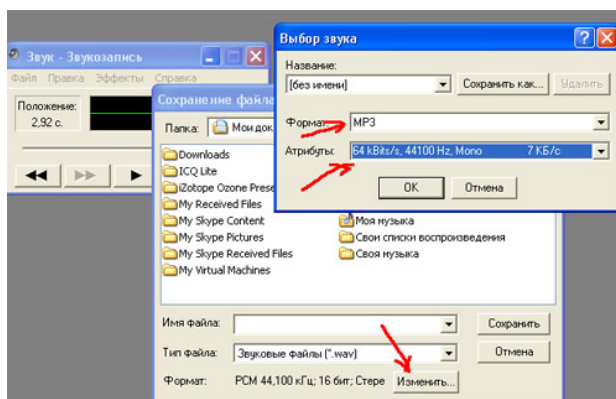
6. Если на компьютере не установлены никакие специальные программы, используем встроенный редактор записи в Windows. Пуск-Все программы- Стандартные-развлечения-звукозапись.

7. Выбираем в меню файл-свойства-качество-Форматы записи: В



появившемся окошке форматов оставляем формат PCM, но выбираем качество голоса с качеством повыше, чем это установлено по умолчанию: например так, как показано на рисунке. Нажимаем Запись в главном окошке программы

8. Перекодируем записанный файл в формат mp3 с помощью той же программы. Для этого Сохранить как, обращаем внизу внимание на кнопку преобразователь форматов Изменить, и выбираем формат mp3 с атрибутами показанными на рисунке (64kbts/сек, 44100 Hz, mono, 7 КБ/с).



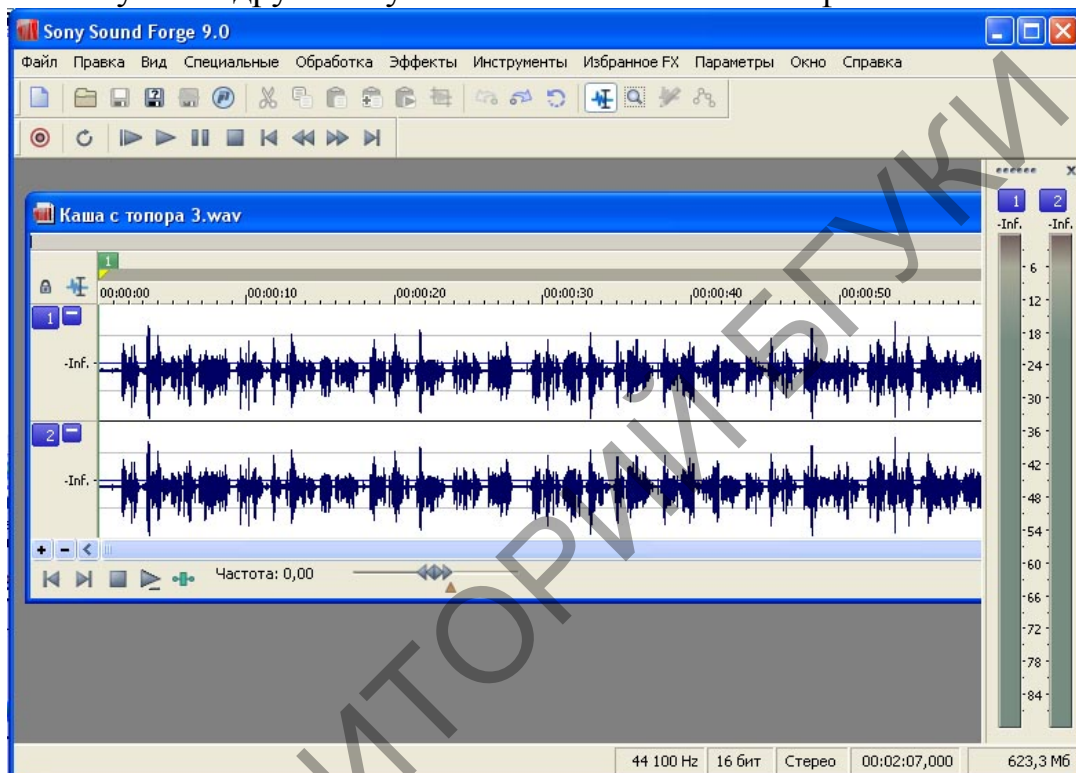
9. Сохраняем файл в формате mp3.

10. Sound Forge. Приемы работы с голосовым файлом.

10.1. Открываем голосовой файл. Удалим смещение по оси амплитуды.

Удаление смещения по оси амплитуды Process (Обработка) -> DC Offset (Смещение DC).

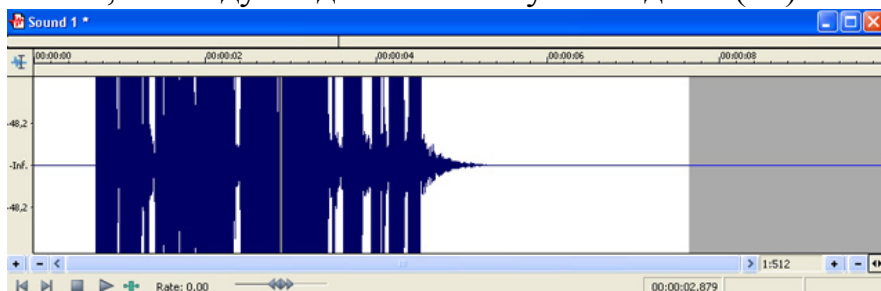
Примечание: Если вы работаете с файлом и не знаете, содержит ли он смещение по оси амплитуды, всегда используйте функцию DC Offset перед началом любых других операций, связанных с редактированием. Если вы этого не сделаете, смещение по оси амплитуды может послужить причиной появления шумов и других звуковых аномалий в вашем файле.



10.2. Удалим посторонний шум от микрофона.

Способ 1. Есть шум. Уберем его. Эффекты (Effects) – Noise Gate (Пороговый шумоподавитель). Предустановка – Пороговый шумоподавитель 2.

Примечание: Основное его назначение - сводить к нулю шум, который равен или ниже заданного диапазона децибел. Делайте этот эффект до тех пор, пока шума между словами не будет вовсе, а если начала и окончания слов обрывистые, то следует сделать отмену последнего(их) шага(ов).



Способ 2. Инструменты - Noise Reduction (Подавление шума).

Как правило, диапазон шумовых помех носит среднечастотный характер. Поэтому выбираем стандартную вилку (FFT Size): 2048.. Выделим отрезок молчания в голосовом треке, откроем плагин Noise Reduction, поставим галочку перед строчкой Capture noiseprint (Снять отпечаток шума) и нажмем Preview (Просмотр).

Плагин обработал данный отрезок одновременно запомнив настройки. Теперь открываем плагин и жмем Preview. Правильно варьируя ползунки, можно очистить запись от нежелательных шумов и при этом не нанести вреда голосу.

Способ 3. Плагин Waves X-Noise.

В записи вручную (самый лучший и правильный метод) удаляете из голосового трека элементы дыхания, большие промежутки тишины между словами или предложениями.

Добавление фрагментов тишины — создание паузы между фразами:

Auto Trim/Crop, Insert Silence и Mute.

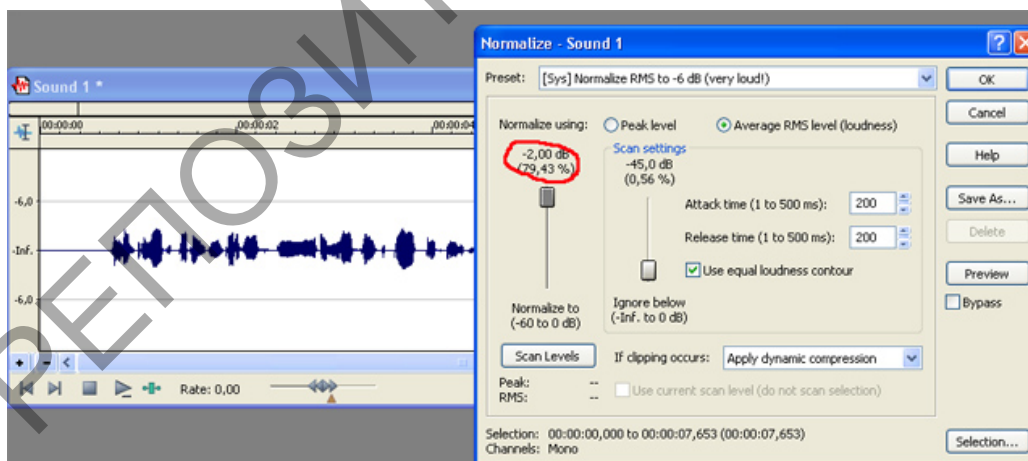
Обработка Process -> Auto Trim/Crop Авто подрезка/обрезка.

Обработка Process -> Insert Silence Вставить тишину.

Приглушение звука - функция Mute просто стирает все выделенные данные, превращая их в тишину.

10.3. Нормализация и компрессия. Нормализация и компрессия позволяет услышать даже самые тихо сказанные слова и буквы.

Меню Process (Обработка) пункт Normalize (Нормализовать).



Пресеты:

Невнятное начало – Максимизация пикового значения, для работы с голосом - [Sys] Нормализация RMS до -10 дБ (речь) и [Sys] Нормализация RMS до -6 дБ (очень громко!).

Нормализация по пиковым значениям (Peak Level).

10.4. Исправление плохих элементов дикции.

Функция Fade in/out (Обработка – Постепенное изменение уровня сигнала).

В некоторых случаях поможет понижение с помощью эквалайзера уровня частоты 90 Гц для того, чтобы звуки "б" и "п" не были взрывными. (Обработка – Эквалайзер Параграфический).

10.5. Обрабатываем эквалайзером. Открываем Обработка – Эквалайзер – Графический (Graphic EQ). Пробуем предустановки. Обычно нужно высокие частоты увеличивать, для этого следует применить пресет [Sys] Повышение высоких частот выше 7 кГц на 6 дБ (Boost high frequencies above 7 kHz by 6 dB). Прослушиваем обрабатываемый файл и добиваемся хорошего звучания.

10.6. Обработка компрессором для выравнивания звучания голоса. Эффекты – Динамическое представление – Графическое.

Стандартные установки компрессора для голоса следующие: Threshold Порог -10-15db, Ratio Отношение 1.5-2, Attac Атака 1-3ms, Release Время Освобождения 30-40ms.

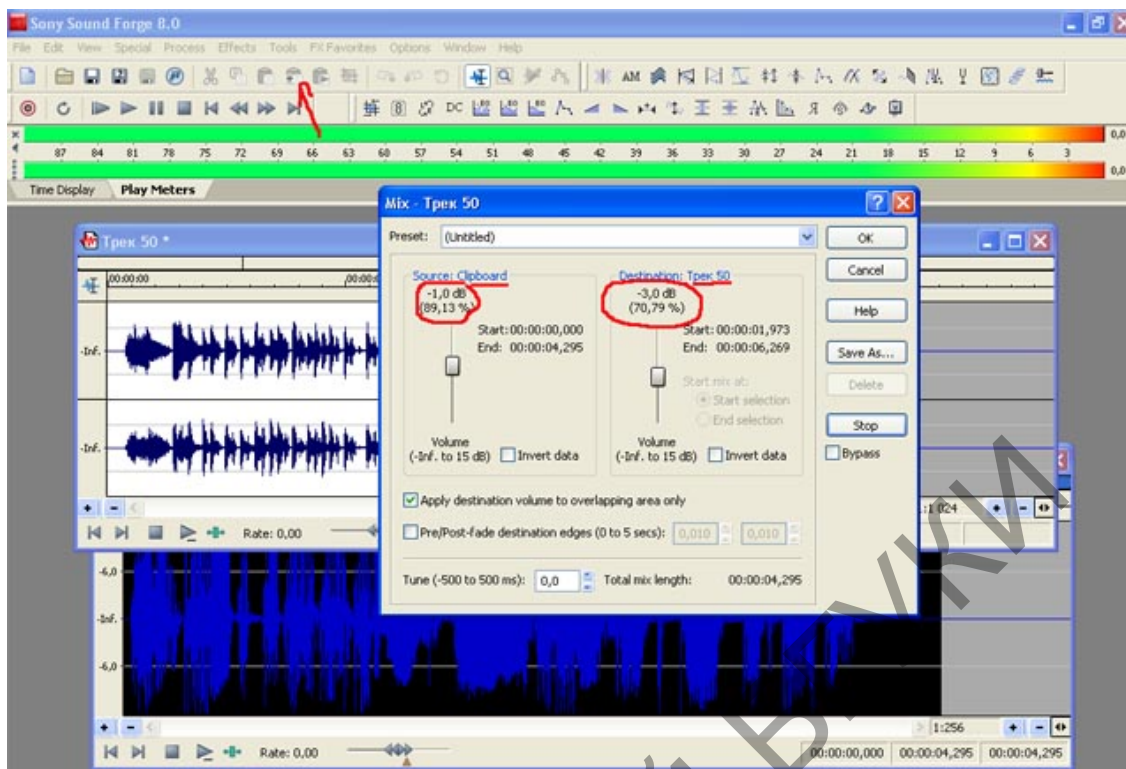
10.7. Ревербация. Пресет "Long hall" Длинный зал или "Cathedral" Кафедральный собор. Попробуйте различные режимы реверберации - Концертный зал, Теплый сектор, и крутите вверх-вниз вторую и третью шкалу. Добивайтесь наилучшего звучания.

10.8. Добавляем динамику. Для этого нужно несколько ускорить трек, сохраняя форманты и текущий тембр голоса. Избранное FX – Sony – ExpressFX Time Stretch. Или Обработка – Протяженность времени.

10.9. Подложим под голос музыку. Открываем любой файл с музыкой. Битовую глубину и частоту дискретизации меняем под параметры голосового файла. Обработка- Конвертер битовой глубины.

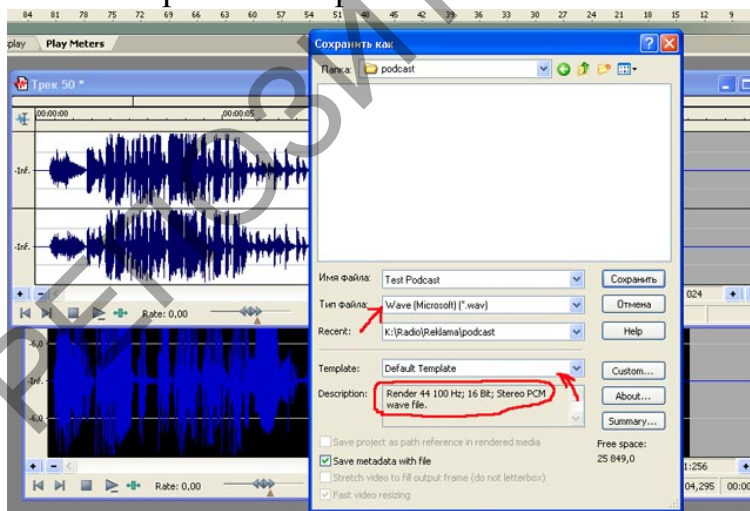
Уменьшаем его громкость на 4 децибела. Обработка – Громкость. -4.

Выделяем кусок с голосом, копируем, переходим в окно с музыкой, ставим курсор в том месте, откуда вы хотели бы слышать голос, жмем в меню на кнопку Mix (Микширование).



Левый вертикальный микшер показывает громкость при наложении того, что у нас в буфере обмена (в нашем случае голоса), правый микшер показывает уровень при миксе файла, на который накладывают (то есть громкость музыки в конкретном случае).

Можно снизить уровень музыки в момент микса еще на 3 децибела.
10.10.Сохраните в mp3.



Перед сохранением проверьте частоты - 44100 hz/16 bit. Если будет другое значение - проигрывание записи через встроенный в страничку плеер будет либо ускоренным, либо замедленным (если значение больше или меньше, соответственно).

Лабораторная работа 6

Системы видеомонтажа

Цель: углубление основных умений по разработке сценария и созданию рекламного ролика в контексте задач социокультурной деятельности.

Порядок выполнения

Задание. Создайте рекламный ролик в контексте задач социокультурной деятельности с помощью программы Adobe Premier

1. Создайте новый проект (*File – New – Project*). Свойства проекта определите в предлагаемых окнах настроек: предложенные по умолчанию (*Load Preset*), собственные (*Custom Settings*).

Примечание. Для телевизионных программ основные характеристики видео будут следующими:

- разрешение от 720×480 пикселей и выше;
- частота кадров в секунду – 25 (в системе PAL) или 30 (в системе NTSC);
- глубина цвета – 32 бит.

Мультимедийные продукты, содержащие видео, имеют обычно такие характеристики:

- разрешение 360×240 пикселей;
- частота кадров – 15 (в секунду);
- глубина цвета – 16 бит.

Для размещения видеофайлов в Интернете необходимо, чтобы они были небольшого размера, поэтому часто приходится жертвовать качеством картинки. Обычные параметры видеоизображений в Интернете таковы:

- разрешение 160×120 или 180×120 пикселей;
- частота кадров – 10 (в секунду);
- глубина цвета – 8 бит.

2. Импортируйте три файла: изображение, звук и видео (*File – Import*).

Примечание. Для разъединения звука и видео: выделяют требуемые файлы, выбирают меню *Clip (Clip)* и команду *Разъединить (Unlink)*.

3. Создайте две текстовые надписи (одна – название, вторая – исполнители). Откройте окна для работы с надписями (*Window – Title*).



Текстов

ые надписи автоматически появятся в окне *Проект (Project)*.

4. Переместите файл с первой надписью из окна *Проект (Project)* на временную шкалу (*TimeLine*). Поместите его на временной шкале с 1 до 10 секунды. Вторую надпись разместите с 30 до 40 секунды.

5. Переместите видеофайл из окна *Проект (Project)* на временную шкалу. Поместите его на временной шкале с 10 до 20 секунды.

Примечание. При необходимости используйте панель с инструментами монтажа рисунка.

	1	Инструмент выбора (Selection Tool) позволяет выделить один клип, трек, сделать активным какое-либо из окон
	2	Инструмент выбора трека (Track Select Tool) позволяет выбрать все клипы на дорожке, которые расположены правее текущего положения курсора
	3	Инструмент монтажа со смещением (Ripple Edit Tool). В этом случае при вставке или удалении клипа происходит изменение длительности всей последовательности в большую или меньшую сторону на величину, равную длительности помещенного или удаленного клипа
	4	Инструмент монтажа с наложением (Rolling Edit Tool). При вставке клипа методом наложения общая длительность всей последовательности сохраняется, однако изменяется граница между клипами за счет наложения одного клипа на другой
	5	Инструмент масштабирования клипов (Rate Stretch Tool) изменяет длительность клипов за счет скорости воспроизведения. Следует осторожно применять этот инструмент для редактирования, чтобы избежать потери качества исходного материала
	6	Инструмент разрезания клипов (Razor Tool) делит один клип на два
	7	Инструмент монтажа с прокруткой (Slip Tool) изменяет входной и выходной маркеры редактируемого клипа. Длительность самого клипа при этом не изменяется
	8	Инструмент монтажа с совмещением (Slide Tool) изменяет входной и выходной маркеры за счет наложения на соседние клипы
	9	Инструмент «Перо» (Pen Tool)

6. Аналогично поместите изображение с 20 до 30 секунды.

7. Переместите звук из окна Проект (Project) на временную шкалу. Поместите его на временной шкале с 20 до 40 секунды (параллельно



изображению и титрам)

Примечание. При необходимости создайте переходы (эффекты, которые применяются при смене одного изображения (или звукового фрагмента) на другой). Используйте окно Эффекты (*Effects*). Изменить длительность перехода можно двумя способами:

– Установить требуемое значение *длительности перехода (Duration)* с клавиатуры во вкладке *Управление эффектом (Effect Controls)*.

– Навести мышку на конец или начало перехода в окне *Монтажный стол (Timeline)*; курсор приобретет соответствующую форму. Потянуть вправо (для увеличения) или влево (для уменьшения) за границу перехода для изменения его длительности.

8. Просмотрите весь проект с помощью окна *Программа (Program)*.

9. Сохраните проект (*File – Save*) и экспортируйте проект в SWF-формат (*File – Export – Movie*).

Примечание. Проекты имеют расширение *.prproj. В проекте хранятся ссылки на исходные файлы. Помните об этом при переносе файла проекта с одного компьютера на другой.

РЕПОЗИТОРИЙ БГУИМ

Тема 3. Сетевые технологии и Интернет

Лабораторная работа 1

Организационная структура Интернета

Цель: ознакомление с основными технологиями информационных коммуникаций на примере глобальной компьютерной сети Интернет.

Краткие теоретические сведения

Технология – способ организации деятельности, включающий ее средства, формы и методы, позволяющий гарантированно получать нужный результат.

Коммуникация – опосредованное (средством коммуникации) и целесообразное взаимодействие субъектов коммуникации с помощью объекта коммуникации - передаваемого сообщения (от коммуниканта к реципиенту).

Технологии информационных коммуникаций – это способ организации коммуникации, включающий совокупность средств коммуникации (коммуникационный интерфейс), форм коммуникации и методов коммуникации, позволяющий гарантированно осуществлять передачу сообщений между коммуникантом и реципиентом.

Компьютерная сеть – система коммуникации между компьютерами. Кроме самих компьютеров включает коммуникационный интерфейс: комплекс аппаратных и программных средств, обеспечивающих коммуникацию - обмен информацией (сообщениями) между компьютерами.

Компоненты компьютерной сети: аппаратный интерфейс: компьютеры, коммуникационное оборудование и т.д.; программный интерфейс: операционная система, сетевые приложения и т.д.; информационное наполнение: пересылаемые данные и команды.

Типология компьютерных сетей:

по масштабу: LAN (Local Area Network) - локальная вычислительная сеть; CAN (Campus Area Network) - сеть, объединяющая близко расположенные LAN; MAN (Metropolitan-Area Network) - сеть городского масштаба; WAN (Wide Area Network) - широкомасштабная сеть; GAN (Global-Area Network) – Интернет;

по организации работы: равноправная сеть – сеть, в которой нет единого устройства управления и хранения данных; распределенная сеть – сеть, в которой сервер (машина, программа или устройство), обеспечивает сервис, но не управляет сетью; сеть с централизованным управлением – сеть, в которой один из компьютеров выполняет функции организации

взаимодействия и т.д.;

по физической топологии (топология - способ соединения компьютеров в сети, физическая топология определяет расположение узлов и соединений): «шина» (Bus); «кольцо» (Ring); «звезда» (Star) и др.

по логической топологии (логическая топология определяет потоки данных и порядок получения права на их передачу, основана на физической топологии).

Классификация устройств в сети: узел (node) – любое устройство в сети, имеющее свой идентификатор; сервер (server) – компьютер, предоставляющий свои ресурсы другому; клиент (client) или рабочая станция – компьютер, потребляющий ресурсы.

Интернет (Internet) - Всемирная сеть, объединяющая глобальные, в т.ч. национальные, и локальные сети с разными типами используемых каналов, разной пропускной способностью и т.п.

Протокол – это набор правил и соглашений, используемых при передаче данных («язык общения» компьютеров в сетях).

Основополагающим протоколом сети Internet является протокол **TCP/IP**. TCP (Transmission Control Protocol) – протокол управления передачей данных. Определяет способ разбивки информации на пакеты и отправки по каналам связи. TCP располагает пакеты в нужном порядке, а также проверяет каждый пакет на наличие ошибок при передаче. Каждый информационный пакет содержит IP-адрес (IP – Internet Protocol) компьютера-отправителя и компьютера-получателя. Протокол IP (Internet Protocol) - это стандарт, описывающий систему адресации в сети Интернет. Каждый узел Интернет имеет уникальный IP-адрес. Формат IP-адреса - четыре числа от 0 до 999, например, 192.168.10.2. (Сейчас идет переход на формат IP6, состоящий из шести чисел).

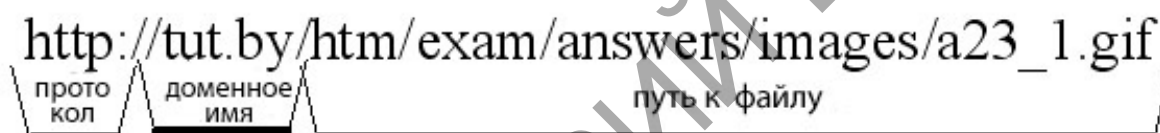
Примеры протоколов

Название протокола	Расшифровка	Назначение
HTTP	Hyper Text Transfer Protocol	Протокол передачи гипертекста
FTP	File Transfer Protocol	Протокол передачи файлов
SMTP	Simple Mail Transfer Protocol	Простой протокол отправки электронных писем

POP3	Post Office Protocol 3	Протокол получения электронных писем
NNTP	News Net Transfer Protocol	Протокол телеконференций

Система имен доменов (DNS – Domain Name System) – это распределенная база данных, которая используется приложениями TCP/IP, для установления соответствия между доменными именами узлов и IP адресами. Функции DNS: определение IP адреса(ов) узла по его доменному имени; определение доменного имени узла по его IP адресу. IP-адрес или соответствующее ему доменное имя позволяют однозначно идентифицировать компьютер в сети Internet.

Для однозначной идентификации любого ресурса в сети Internet используют **URL (Uniform Resource Locator)** – унифицированный указатель ресурсов.



World Wide Web (WWW, "Всемирная паутина") - основной инструмент Интернет, её главный информационный сервис. World Wide Web (WWW, "Всемирная паутина") - гипертекстовая информационная система поиска ресурсов Интернет и доступа к ним. Система WWW построена на протоколе передачи гипертекста HTTP. Контент системы WWW – это WWW-страницы, называемых сайтами (англ. site - участок). WWW-страницы (сайты) - гипермедийные документы системы World Wide Web. Создаются с помощью языка разметки гипертекста HTML (Hypertext markup language).

Браузер (англ. browse - листать, просматривать) - программа, с помощью которой пользователь организует взаимодействие с WWW-серверами и другими сервисами и ресурсами Интернет.

Передача файлов по протоколу FTP. С помощью данного сервиса возможно, используя соответствующую FTP-программу на своем компьютере, подключиться к удаленной машине (FTP-серверу), ознакомиться с перечнем доступных файлов и скопировать их на свой компьютер. FTP позволяет пересылать по сети файлы любого типа – тексты, изображения, исполняемые программы, файлы с записями звуковых фрагментов и т. д.

Поиск информации в сети

Язык запросов – это искусственный язык, содержащий специальные

символы и операторы, на котором создаются запросы, которые обрабатываются информационно-поисковыми системами.

Ключевые слова (key-words) - слова, которые наиболее точно отражают суть того, что необходимо найти.

Релевантным (англ. relevant – «подходящий») называется документ, полностью соответствующий введенному запросу. Релевантность поиска – это его полнота (найденны все релевантные документы) и точность (не найдено ничего лишнего). Коэффициент полноты поиска определяют как отношение количества найденных релевантных документов к общему количеству существующих в базе данных релевантных документов. Коэффициент точности поиска определяют как отношение количества найденных релевантных документов к количеству всех найденных документов.

Пертигентность – это соотношение объема полезной для пользователя информации к объему полученной. Средства повышения пертигентности: уточнение формулировок запросов, ранжирование по весовым критериям, ограничение числа выданных в результате поиска документов.

Одним из средств повышения пертигентности поиска в поисковых системах являются «расширенный» (форма, которую нужно заполнить, ответив на дополнительные вопросы) и «сложный» (использование булевых операторов, поиск с помощью логических операторов) поиск.

Сохранение web-страниц. Для сохранения найденной информации имеют значение два обстоятельства: тип браузера и вид, в котором вы хотите сохранить документ.

Рассмотрим варианты сохранения на примере Microsoft Internet Explorer (IE). IE позволяет сохранить документ как: web-страницу полностью (со всеми иллюстрациями, которые разместятся в отдельной папке, что довольно удобно); web-архив (с включенными иллюстрациями); web-страницу, один файл (без иллюстраций, только HTML); текстовый файл (только текст документа). При сохранении в IE возможно указание кодировки страницы. При сохранении файлов других типов (doc, ppt, pdf и др.) браузер автоматически начнет «скачивание» файла после подтверждения.

Для «скачивания» информации из Интернета без помощи браузера существуют специальные утилиты (ReGet, FlashGet и др.). При их использовании после обрыва связи решается проблема восстановления перекачки.

Для сохранения полученной из Интернета информации также возможно использование текстового редактора: найденная информация на web-странице копируется в буфер обмена, вставляется из буфера в текстовый редактор и сохраняется его средствами.

Задание 1. Познакомьтесь с основными понятиями, связанными с технологиями информационных коммуникаций в разделе.

Краткие теоретические сведения

Задание 2. Изучите язык запросов на примере различных поисковых систем. Используйте для поиска материала функцию расширенного поиска и справку.

Результаты изучения операторов поисковых систем оформите в виде таблицы:

ОПЕРАТОРЫ ПОИСКОВЫХ СИСТЕМ – СПЕЦИАЛЬНЫЕ СИМВОЛЫ				
Назначение оператора	Поисковая система			
	google.co m	yandex.ru	yahoo.co m	altavista.c om
Поиск точных слов или фразы				
Поиск страниц, содержащих одно из слов				
Исключение страницы, содержащие слово или фразу				
Поиск страниц, содержащих все слова				
Отключение морфологического поиска				
Ограничение поиска определенным типом файла				
Поиск страниц, содержащих ссылки на указанную страницу				
Ограничение поиска страницами, написанные на определенном языке				
Ограничение поиска страницами, содержащими указанные слова в заголовке страниц				

Задание 3. Для отработки навыков поиска информации решите предложенные задачи.

№1. Как доехать? Предложите гастрольный тур коллектива по двум, трём городам и разработайте маршрут передвижения: поезда, автобусы, самолёты и т.д. Результаты поиска оформите в виде таблицы:

Маршрут передвижения (пункты и время отправления и прибытия)				
№ поезда (рейса)	Название поезда (рейса)	Время отправления	Время прибытия	Время в пути

Пример решения задачи № 1.

Пояснения: Вокальный ансамбль из Бобруйска имеет гастрольный тур по городам Беларуси (Минск-Брест). Первое выступление в 19:00 в Минске. Концерт длится 2,5 часа. Следующий концерт в Бресте в 10:00 следующего дня. Подготовьте маршрут передвижения вокального ансамбля.

Маршрут передвижения вокального ансамбля из Бобруйска через Минск в Брест (гастрольный тур)				
№ поезда (рейса)	Название поезда (рейса)	Время отправления	Время прибытия	Время в пути
330К	Бобруйск- Минск	15:06	17:21	2 часа 15 минут
027Б	Минск- Брест	23:28	02:38	3 ч. 10 мин

№2. Где лучше купить книгу? Найдите стоимость любой книги культурологической тематики в трёх разных Интернет-магазинах. Результаты поиска оформите в виде таблицы:

Где лучше купить книгу? (название книги)			
	Название	URL	Цена

Пример оформления библиографического описания книги:

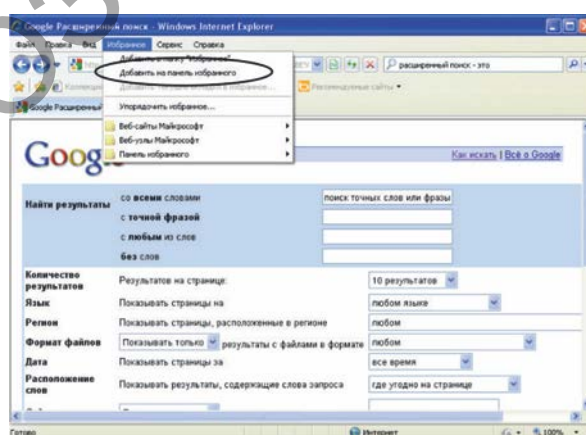
Шиманский, М.Н. Моя вера - моя Беларусь/ М.Н. Шиманский. - Минск: Историко-художественная лит-ра, 2010. – 270 с.

№3. Где можно прочесть книгу? Найдите любую книгу культурологической тематики в электронных библиотеках. Выясните в каких электронных библиотеках её можно прочесть. Результаты поиска оформите в виде таблицы:

Где можно прочесть книгу? (название книги)	
Название библиотеки	URL

№ 4. Найдите информационные ресурсы, посвящённые культурологической тематике (порталы, блоги, форумы и т.д.). Опишите основное контентное наполнение и назначение каждого из них. Результаты поиска оформите в виде таблицы:

Название веб-сайта	URL	Контент и назначение



Задание 4. Изучите алгоритм работы с центром управления избранным и отработайте навыки использования браузера и сохранения информации.

Избранное - позволяет просматривать избранные веб-сайты, RSS-каналы (📡) и журнал в одном месте. Чтобы отобразить меню "Избранное", нажмите

кнопку "Избранное": ★. Изучите особенности добавления ссылок в папку «Избранное» и на панель избранного.

Изменить масштаб просматриваемой страницы возможно с помощью меню Вид.

Для отработки навыков использования браузера и сохранения информации выполните предложенные задания:

1.1 Найдите пять поисковых систем. Каждую поисковую систему сохраняйте в Избранном в папке Поисковые системы.

1.2 Найдите пять каталогов по предложенной теме. Каждый каталог сохраняйте в Избранном в папке Каталоги.

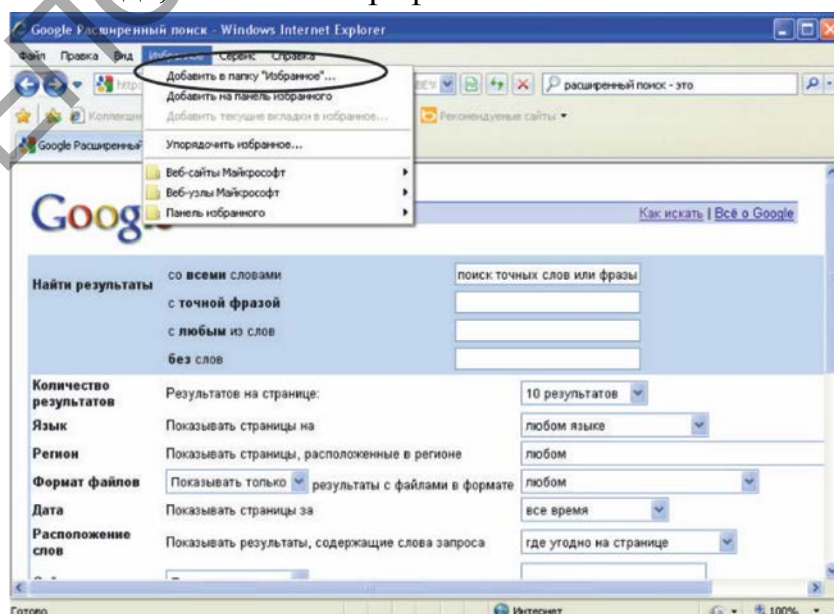
1.3 Используя сохранённые поисковые системы и каталоги, найдите пять ресурсов по предложенной теме. Сохраните их в Избранном в папке Материалы по теме.

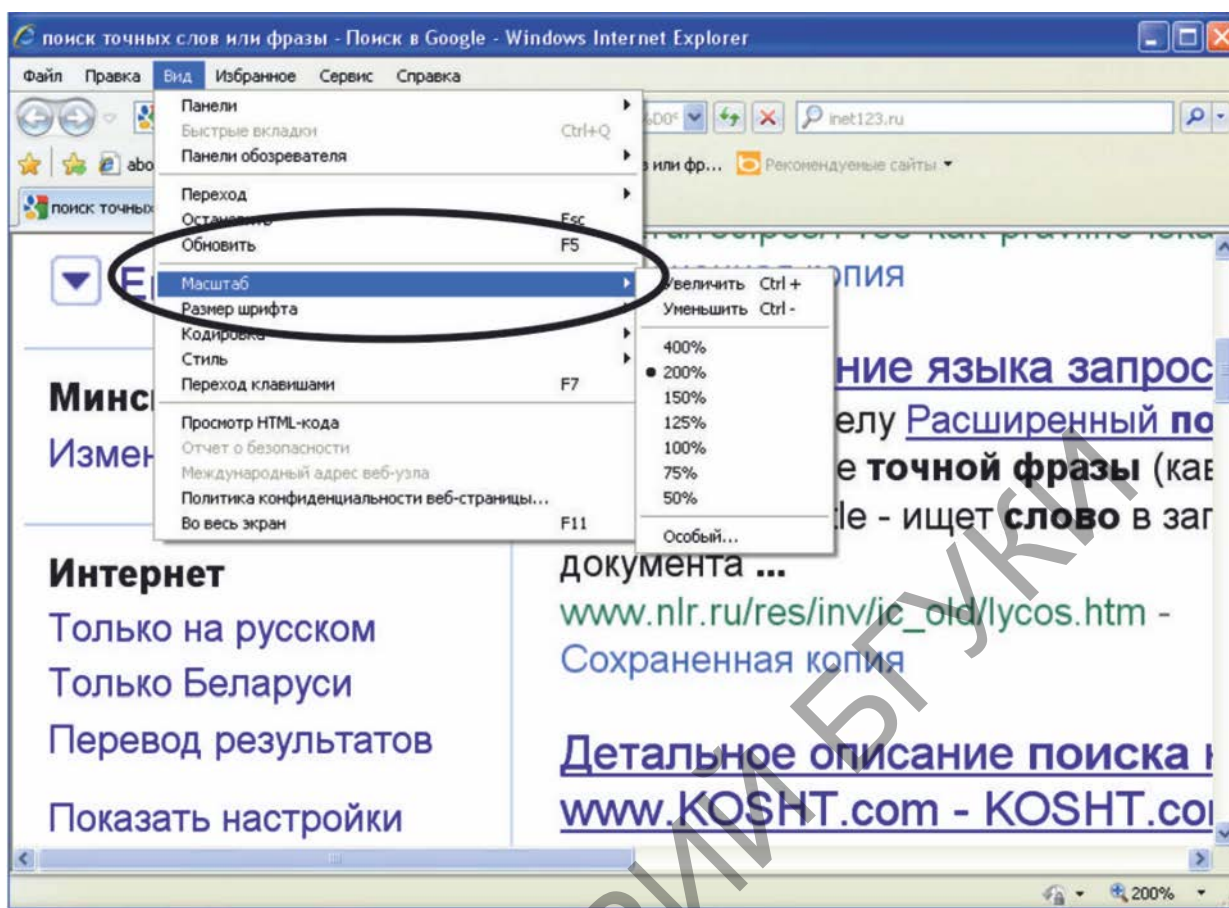
1.4 Используя сохранённые поисковые системы, найдите пять графических изображений по предложенной теме. Сохраните их в личной папке.

1.5 Найдите три иностранных сайта по предложенной теме. Сохраните их в Избранном в папке Материалы по теме. Отметьте, каким странам принадлежат данные сайты.

1.6 Переведите иностранные сайты на русский или белорусский язык. Перевод сохраните в текстовом файле в личной папке. Отредактируйте перевод. Вставьте ссылку на ресурс.

1.7 Используя сохранённые в пункте 1.3 ресурсы, сохраните нужную Вам страницу в личной папке в трех вариантах: 1 – только текстовую информацию; 2 – страницу полностью, вместе с графикой в неизменённом виде; 3 – страницу в неизменённом виде, только без графики.





Дополнительные задания

1. В любой поисковой системе составьте запрос для поиска информации о социальной рекламе. Исключите предложения об услугах, коммерческую рекламу. Сосредоточьте поиск на конкурсах по социальной рекламе.
2. Найдите информацию об особенностях белорусской культуры в материалах телеконференций.
3. Найдите музыкальную композицию, которая могла бы сопровождать Вашу презентацию по предложенной теме. Сохраните её адрес в личной папке.
4. Найдите видеoinформацию предложенного жанра (хроникального, документального, исторического или др.) Сохраните её адрес в личной папке.
5. Найдите компьютерную программу на FTP-сервере. Опишите алгоритм поиска и скачивания информации с FTP-сервера.

Лабораторная работа 2 Основные сервисы Интернета

Цели работы:

– углубление и систематизация имеющихся знания, формирование навыков применения социальных сервисов Интернета в профессиональной деятельности культуролога;

– формирование умения использования Интернета для составления аннотированных списков социальных сервисов профессиональной тематики.

Методические рекомендации по выполнению

Задание 1. Приведите примеры информационных ресурсов по каждому из видов социальных сервисов. Результаты представьте в виде таблицы (табл. 3).

Таблица 3.

Социальные сервисы сети Интернет

Вид социального сервиса Интернета	Название	URL-адрес	Основной контент	Возможность использования в профессиональной деятельности культуролога (поиск, консультирование, реклама и т.д.)
Блог				
Вики				
Географический сервис				
Социальная сеть				
Социальное хранилище				
Форум				

Задание.2. Проанализируйте возможности модератора, участника и владельца в блоге, форуме и социальной сети. По результатам анализа заполните таблицу 4:

Таблица 4. Анализ прав и возможностей модератора, участника и владельца

	Блог	Форум	Социальная сеть
Кто управляет коммуникацией?			
Кто имеет право задавать темы для обсуждения?			
Кто имеет право на редактирование сообщений?			
Кто регламентирует права участников коммуникации?			
Кто осуществляет контроль соблюдения правил общения?			

Задание.3. 1. Найдите и изучите социальные сервисы Интернета искусствоведческой тематики.

2. Составьте аннотированное описание найденных ресурсов (5 наименований).

3. Проанализируйте социальные сети из списка найденных по возможности их использования в профессиональной деятельности культуролога.

Задание.4. Найдите ответы на поставленные вопросы в Интернете. Оформите краткий ответ с указанием источника информации

1. Какое влияние оказал Интернет на функции журналистики?
2. Как классифицируются поисковые запросы?
3. Как определить объект поиска?
4. Какие бывают типы информации?
5. Какие критерии используются при оценке информации?
6. Как осуществляется поиск файлов определённого формата?
7. Как осуществляется поиск программного обеспечения?
8. Как искать рефераты?
9. Какие онлайн-источники информации Вы знаете?
10. Какие офлайн-источники информации Вы знаете?
11. Как узнать последние новости?
12. Как найти требуемую цитату в книге?
13. Как найти толкование того или иного слова в энциклопедиях?
14. Как перевести слово с английского языка на русский?
15. Как перевести слово с чувашского языка на русский?
16. Как найти требуемое изображение?
17. Как узнать прогноз погоды на завтра в Чебоксарах?
18. Для чего нужны каталоги?
19. Что такое «язык запросов» в Google?
20. Какие отличия между языком запросов в Google и в Яндекс?

Задание 5. Оформите отчет. Помните о правильном библиографическом описании источников.

Пример приведен ниже.

Ресурсы удаленного доступа	Козулько, Г. Беловежская пуца должна стать мировым наследием / Г. Козулько // Беловежская пуца – XXI век [Электронный ресурс]. – 2004. – Режим доступа : http://bp21.org.by/ru/art/a041031.html . – Дата доступа : 02.02.2006.
----------------------------	---

<p>Лойша, Д. Республика Беларусь после расширения Европейского Союза: шенгенский процесс и концепция соседства / Д. Лойша // Белорус. журн. междунар. права [Электронный ресурс]. – 2004. – № 2. – Режим доступа : http://www.cenunst.bsu.by/journal/2004.2/01.pdf. – Дата доступа : 16.07.2005.</p>
<p>Статут Международного Суда // Организация Объединенных Наций [Электронный ресурс]. – 2005. – Режим доступа : http://www.un.org/russian/document/basicdoc/statut.htm. – Дата доступа : 10.05.2005.</p>
<p>Cryer, R. Prosecuting international crimes : selectivity and the international criminal law regime / R. Cryer // Peace Palace Library [Electronic resource]. – The Hague, 2003–2005. – Mode of access : http://catalogue.ppl.nl/DB=1/SET=3/TTL=1/SHW?FRST=12. – Date of access : 04.01.2006.</p>

Лабораторная работа № 1

Создание базы данных

Цель работы: формирование умений создавать таблицы, вводить в них данные, устанавливая связи между ними, устанавливая режим целостности данных, добавлять в базу данных новые таблицы.

1. Создайте файл пустой базы данных **Библиотека**.
2. Создайте структуру таблицы **Издательства**, которая содержит следующие поля: **Код издательства**, **Наименование**, **Город**.

Структура таблицы **Издательства**

Имя поля	Тип данных	Свойства поля	Ключ
Код издательства	Числовой	Размер поля – целое, обязательное поле	Да
Наименование	Текстовый	Размер поля – 20	
Город	Текстовый	Размер поля – 15	

После того, как вы набрали имена полей таблицы **Издательства**, указали для них типы данных, установили требуемые свойства и определили ключ, нажмите на кнопку **Закреть** окна таблицы в режиме конструктора. При этом

появится сообщение: **Сохранить изменения макета или структуры таблицы «Таблица1» ?** Ответьте на него утвердительно. После этого появится диалоговое окно **Сохранение**, в котором наберите имя таблицы **Издательства** и нажмите кнопку **ОК**.

3. Аналогичным образом создайте структуру таблицы **Книги** с полями: **Код книги, Название, Автор, Код издательства, Объем, Год издания и Стоимость**.

Структура таблицы **Книги**

Имя поля	Тип данных	Свойства поля	Ключ
Код книги	Числовой	Размер поля – целое, обязательное поле	Да
Название	Текстовый	Размер поля – 25	
Автор	Текстовый	Размер поля – 15	
Код издательства	Числовой	Размер поля – целое, обязательное поле	
Объем	Числовой	Размер поля – целое	
Год издания	Числовой	Размер поля – целое	
Стоимость	Денежный	Формат поля – денежный	

4. Создайте структуру таблицы **Темы**, содержащую поля: **Код книги, Тема**.

Структура таблицы **Темы**

Имя поля	Тип данных	Свойства поля	Ключ
Код книги	Числовой	Размер поля – целое, индексированное поле (совпадения допускаются)	
Тема	Текстовый	Размер поля – 30	

5. Установите связи между таблицами **Издательства, Книги и Темы** базы данных **Библиотека** так, как это показано на рисунке. Обратите внимание на то, что между таблицами базы данных **Библиотека** будут связи одного типа – один ко многим.

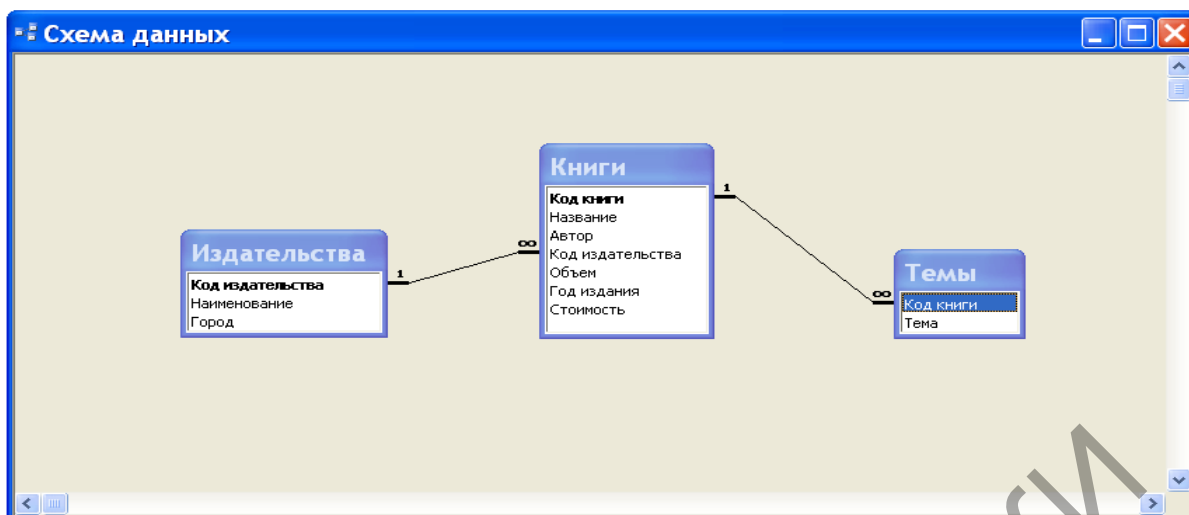


Схема базы данных Библиотека.

Поскольку при установке связей между таблицами мы указали режим обеспечения целостности данных с его подрежимами, то тем самым мы задали порядок, в котором надо вводить данные в таблицы. Вначале надо вводить данные в таблицу **Издательства**, затем – в таблицу **Книги** и лишь после этого – в таблицу **Темы**.

6. Введите в таблицы **Издательства**, **Книги** и **Темы** базы данных **Библиотека** данные, приведенные ниже в таблицах 1-3. Ввод данных в таблицу осуществляется в режиме таблицы. Чтобы открыть таблицу в режиме таблицы достаточно в окне базы данных во вкладке **Таблицы** выполнить двойной щелчок мышью на имени таблицы. Быстро перевести таблицу из режима конструктора в режим таблицы и наоборот можно нажатием кнопки **Вид** на панели **База данных**. Для перехода от одного поля к другому, когда осуществляется ввод данных в таблицу, удобно использовать клавишу **Tab**.

Таблица 1

Издательства

Код издательства	Наименование	Город
1	Наука	Москва
2	Мир	Москва
3	Радио и связь	Минск
4	Машиностроение	Киев
5	Финансы и статистика	Москва

Таблица 2

Книги

Код книги	Название	Автор	Код издательства	Объем	Год издания	Стоимость
1	Педагогика	Беспалько	2	340	1994	24 000,00р.
2	Сборник задач	Сканави	2	634	1992	60 000,00р.
3	Программирование	Арсак	1	273	1989	18 000,00р.
4	Язык Ада	Перминов	3	278	1987	16 000,00р.
5	Операционные системы	Грибанов	3	446	1991	23 000,00р.
6	БД на Паскале	Ульман	4	563	1992	32 000,00р.
7	IBM PC для пользователя	Фигурнов	5	368	1994	22 000,00р.

Таблица 3

Темы

Код книги	Тема	Код книги	Тема
1	Личность человека	5	Управление заданиями
1	Проектирование ППС	5	Управление задачами
1	Технология обучения	5	Управление данными
1	Анализ учебного процесса	6	Операции с поставщиками
2	Уравнения	6	Бухгалтерская книга
2	Прогрессии	6	Платежная ведомость
2	Геометрические задачи	6	Реляционная алгебра
3	Игры с числами	6	Правила нормализации
3	Игры без стратегии	7	Устройства компьютера
3	Комбинаторные задачи	7	Файлы и каталоги
3	Стратегия без игры	7	Диалог пользователя
4	Программные модули	7	Работа с дисками
4	Лексика	7	Программы архивации
4	Предопределенные типы	7	Конфигурирование системы
4	Операторы	7	Обслуживание дисков
5	Структура ОС ЕС	7	Редактирование текстов

Расширение базы данных Библиотека

Иногда в процессе разработки базы данных или в процессе опытной эксплуатации ее возникает необходимость добавления в нее новых таблиц. Очевидно, что спроектированная нами в предыдущей работе база данных **Библиотека** обладает очень ограниченными возможностями. Эта база данных, состоящая из трех таблиц: **Издательства**, **Книги** и **Темы**, не позволяет автоматизировать работу с читателями. В ней отсутствует информация о читателях.

В данной работе мы научимся добавлять таблицы в базу данных с целью расширения ее функциональных возможностей. Создание новых таблиц осуществляется точно так же, как это мы делали в предыдущей работе. Для добавления таблиц в ранее созданную схему данных и установления связи между таблицами используется кнопка **Отобразить таблицу**, размещенная на панели инструментов **Связь**.

1. Откройте базу данных **Библиотека**. Создайте в ней структуру таблицы **Читатели**, которая будет содержать следующие поля: **Код читателя**, **Фамилию**, **Имя**, **Отчество**, **Домашний телефон**, **Домашний адрес**. Типы данных для полей таблицы, их свойства определите самостоятельно по смыслу. В качестве ключа укажите поле **Код читателя**.

2. Аналогичным способом создайте структуру таблицы **Выдача книг**. В эту структуру включите три поля: **Код читателя**, **Код книги**, **Дата заказа**. В этой таблице ключевое поле не задавайте. Для поля **Дата заказа** укажите тип данных – **Дата/время**. Обратите внимание на то, что в последствии ключ **Код читателя** в таблице **Читатели** будет связываться с полем **Код читателя** в таблице **Выдача книг**. Поэтому эти поля должны иметь соответствующие типы данных и свойства.

3. Установите между добавленными таблицами: **Читатели** и **Выдача книг**, а также ранее созданными таблицами: **Издательства**, **Книги** и **Темы**, связи так, как это показано в окне **Схема данных** на рис. 1.

Напомним, что для установления связи между таблицами надо открыть окно **Схема данных**. При его открытии появляется диалоговое окно **Добавление таблицы**, в котором надо выделить имена тех таблиц, между которыми будут устанавливаться связи. После этого нажимают кнопки **Добавить** и **Заккрыть**. Затем в окне **Схема данных** с помощью мыши перетаскивают ключевое поле одной таблицы на соответствующее поле в другой таблице. В появившемся окне **Связи** задают режим **Обеспечение целостности данных** и его подрежимы: **каскадное обновление связанных полей** и **каскадное удаление связанных записей** и нажимают кнопку **Создать**.

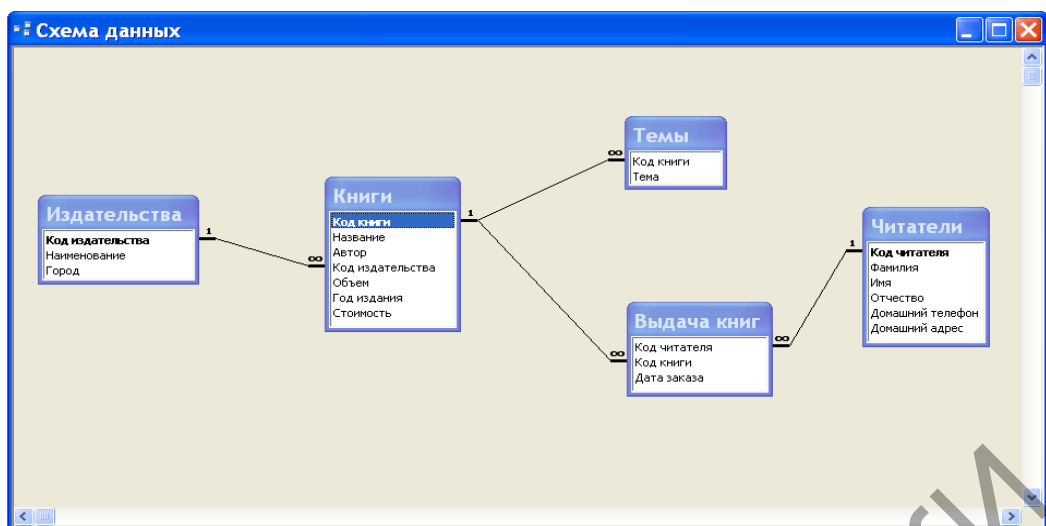


Рис. 3. Схема расширенной базы данных **Библиотека**.

4. Откройте таблицу **Читатели** и введите в нее данные, приведенные в таблице 4.

Таблица 4

Данные для ввода в таблицу **Читатели**

Код читателя	Фамилия	Имя	Отчество	Домашний телефон	Домашний адрес
1	Аксенов	Виктор	Сергеевич	252-88-13	ул. Есенина, 15-19
2	Голубева	Елена	Андреевна	220-99-29	ул. Чкалова, 7-38
3	Васильев	Игорь	Петрович	232-64-78	ул. Богдановича, 102-34
4	Кучеров	Валентин	Степанович	266-24-95	ул. Кнорина, 27-5
5	Мастяница	Вячеслав	Иванович	246-42-25	ул. Плеханова, 34-98
6	Победимская	Лариса	Анатольевна		ул. Чкалова, 9-10
7	Литвин	Борис	Николаевич	239-55-76	пр. Независимости, 46-54
8	Германович	Рита	Мироновна	278-31-51	ул. Казинца, 26-9
9	Бинцаровский	Теодор	Петрович		ул. Корженевская, 1-288

5. Введите в таблицу **Выдача книг** данные, приведенные в таблице 5.

Данные для ввода в таблицу **Выдача книг**

Код читателя	Код книги	Дата заказа	Код читателя	Код книги	Дата заказа
1	1	1.09.15	4	3	7.01.16
1	3	5.07.16	4	4	25.10.15
1	4	21.10.15	5	2	23.04.16
2	1	4.11.15	6	1	18.06.16
3	2	3.08.16	7	3	20.01.16
8	7	25.12.15	9	6	2.02.16

Обратите внимание на то, что, если бы вы попробовали вначале ввести данные в таблицу **Выдача книг**, а затем в таблицу **Читатели**, то MS Access это не позволил бы сделать. Поэтому мы специально раньше установили связи между таблицами, а затем уже вводили данные в таблицы. В этом случае MS Access будет проверять целостность данных.

Лабораторная работа 2

Отбор данных

Цель работы: формирование умения создавать простые запросы для выбора данных.

Запрос в MS Access – это требование предоставить информацию, накопленную в таблицах базы данных. Запрос можно получить с помощью инструментов запроса. Запрос может относиться к одной или к нескольким связанным таблицам. На основании запроса MS Access формирует динамический набор записей. Физически он выглядит как таблица, хотя фактически не является ею. Динамический набор записей является временным (или виртуальным) набором записей и не хранится в базе данных. После закрытия запроса динамический набор записей этого запроса прекращает свое существование.

MS Access поддерживает различные типы запросов, которые можно разбить на шесть основных категорий.

Запрос на выборку. Извлекает данные из одной или нескольких таблиц (основываясь на заданных критериях) и результаты представляет в виде динамического набора записей.

Групповой запрос. Представляет специальную версию запроса на выборку. Позволяет вычислять суммы, подсчитывать количество записей и выполнять расчет итоговых значений. Для этого запроса MS Access добавляет в бланк запроса строку **Групповая операция**.

Запрос на изменение. Позволяет создавать новые таблицы (команда **Создание таблицы**) или изменять данные в существующих таблицах (команды **Удаление**, **Обновление** и **Добавление**). Если в наборе результатов запроса на выборку можно вносить изменения только в одну запись за раз, то запрос на изменение разрешает вносить изменения в несколько записей сразу при выполнении этой операции.

Перекрестный запрос. Отображает результаты статистических расчетов (такие как суммы, количество записей и средние значения). Эти результаты группируются по двум наборам данных в формате перекрестной таблицы. Первый набор выводится в столбце слева и образует заголовки строк, а второй выводится в верхней строке и образует заголовки столбцов.

Запрос SQL. Существуют три типа запросов SQL: запрос на объединение, запрос к серверу и управляющий запрос, которые используются для манипуляций с базами данных SQL. Создаются эти запросы с помощью написания специальных инструкций SQL.

Запрос с ограничением, или Top(n). Этот ограничитель запроса можно использовать только в паре с одним из предыдущих пяти типов запросов. Он позволяет задавать число первых записей или часть общего количества записей в процентах, которую вы хотели бы получить в любом виде запроса.

С помощью запросов можно выполнять следующее: выбирать таблицы, выбирать поля, выбирать записи, сортировать записи, выполнять вычисления, создавать таблицы, создавать формы и отчеты на основе запроса, создавать диаграммы на основе запроса, использовать запрос в качестве источника данных для других запросов (подчиненных запросов) и вносить изменения в таблицы.

Создание запроса и работа с ним выполняется во вкладке **Запросы** окна базы данных. Для работы с запросом можно воспользоваться панелью инструментов **Конструктор запросов**.



Рис. 1. Панель инструментов **Конструктор запросов**.

MS Access допускает два способа создания запроса: с помощью мастера и в режиме конструктора. Для того чтобы приступить к созданию запроса с помощью мастера можно выполнить двойной щелчок мышью на строке **Создание запроса с помощью мастера** во вкладке **Запросы** окна базы данных или щелчок мышью на кнопке **Создать**, а затем выбрать вариант **Простой запрос** в окне диалога **Новый запрос**.

Создание запроса на выборку для сортировки информации

В работе далее для создания запросов будем использовать режим конструктора. Самый быстрый способ запустить этот режим – выполнить двойной щелчок мышью на строке **Создание запроса в режиме конструктора**. При этом появится окно диалога **Добавление таблицы** (см. рис. 2).

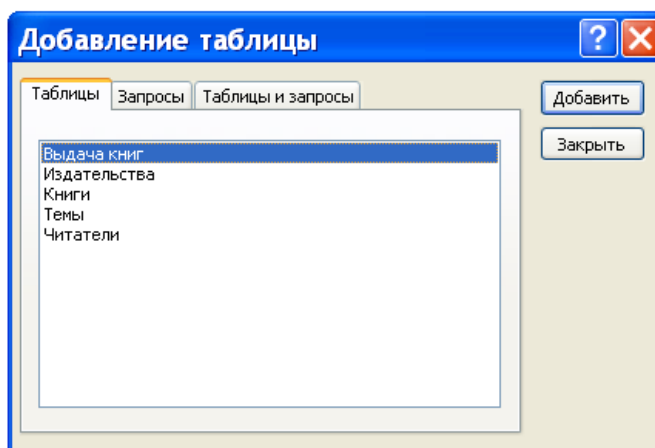


Рис. 2. Окно диалога **Добавление таблицы**.

Создание запроса для сортировки информации рассмотрим на следующем примере. Требуется составить список книг московских издательств, рассортированных по фамилиям авторов. В динамический набор надо включить следующие поля: **Автор**, **Название**, **Наименование** и **Год издания**.

Обратим внимание на то, что в нашем запросе будут использоваться поля из двух таблиц: **Издательства** и **Книги**. Поэтому в окне диалога надо выделить имена этих двух таблиц. Для этих целей щелкните вначале, например, по имени **Издательства**, а затем, удерживая клавишу **CTRL**, щелкните по имени **Книги**. После того как требуемые имена таблиц выделены, надо в окне диалога **Добавление таблицы** щелкнуть мышью по кнопке **Добавить**, а затем – **Заккрыть**. В результате выполнения таких действий в верхней части окна запроса в режиме конструктора появятся списки полей для каждой из выбранных таблиц (см. рис. 3).

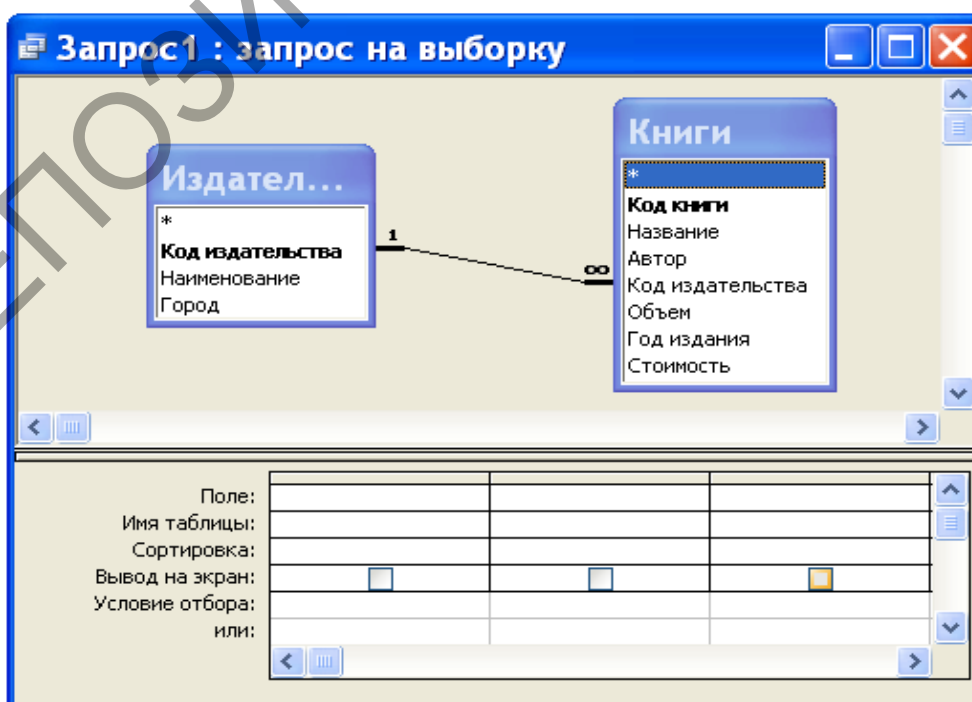


Рис. 3. Окно запроса в режиме конструктора.

Окно запроса в режиме конструктора предназначено для создания новых и изменения существующих запросов. При создании запросов в этом режиме используется механизм запросов по образцу **QBE** (Query by Example). Окно в этом случае состоит из двух частей. В верхней части окна размещаются списки полей, из которых будет формироваться запрос. В нижней части окна располагается бланк **QBE**, в который нужные для запроса поля перемещаются при помощи мыши из списков полей, размещенных в верхней части окна.

Для изменения относительной высоты верхней и нижней частей окна используется специальная разделительная линия. При установке курсора на эту линию курсор приобретает вид двунаправленной стрелки. В это момент разделительную линию можно перемещать вверх или вниз.

Имена полей, которые будут образовывать динамический набор, должны быть в соответствующем порядке размещены в строке бланка **QBE**. Сделать это можно несколькими способами. Самый простой способ состоит в двойном щелчке мышью на имени в списке полей. Указанным способом в строке **Поле** бланка **QBE** поместите поля: **Автор**, **Название**, **Наименование**, **Год издания** и **Город**. Последнее поле нам понадобилось, чтобы задать условие отбора для выбора для выбора книг московских издательств.

В строку **Условие отбора** для поля **Город** наберите текст "Москва" (задание условий отбора подробнее будет рассмотрено ниже). Даже, если вы текст в кавычки не возьмете, MS Access сам это сделает. Условие отбора нам понадобилось для того, чтобы в запросе выбирались не все книги, а только книги, изданные в Москве.

Поскольку по условию задачи поле **Город** не надо выводить на экран, то в строке **Вывод на экран** для этого поля уберите щелчком мыши пометку ("птичку").

Для того чтобы в динамическом наборе записи выводились в алфавитном порядке по фамилиям авторов, надо в строке **Сортировка** для поля **Автор** задать направление сортировки. Выполните щелчок мышью на ячейке в строке **Сортировка** для поля **Автор**. При этом справа в этой ячейке появится кнопка раскрытия списка направления сортировки. Выберите в этом списке направление сортировки – по возрастанию.

Порядок обработки полей при сортировке по нескольким полям определяется их положением в бланке **QBE**: сначала сортируются значения в крайнем левом поле и далее слева направо. После указанных действий бланк **QBE** будет иметь вид, представленный на рисунке 4.

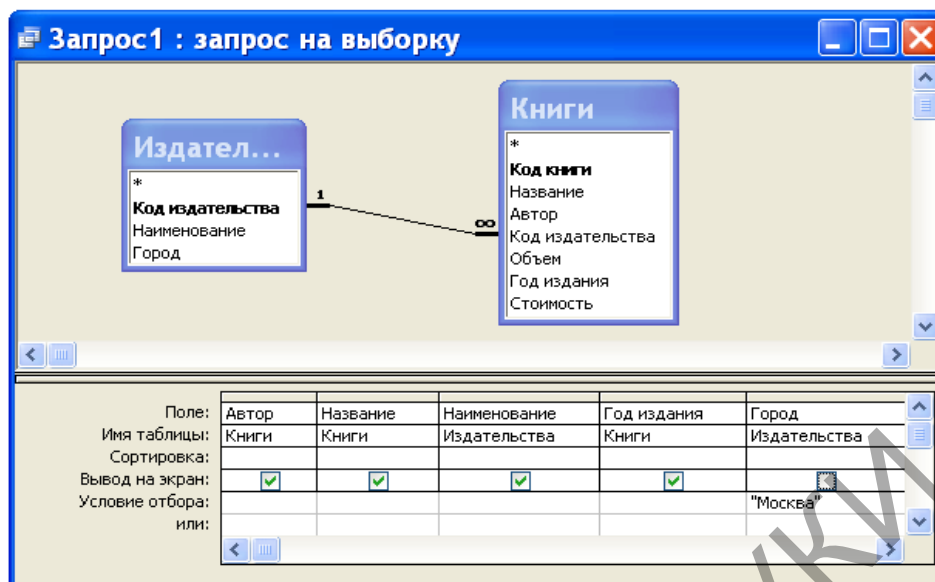


Рис. 4. Вид бланка QBE для решения задачи.

Сейчас выполним созданный нами запрос. Для этого нажмите кнопку **Режим таблицы** на панели инструментов **Конструктор запросов** (первая кнопка – см. рис. 1). После нажатия этой кнопки вы увидите список книг московских издательств, рассортированный в алфавитном порядке по фамилиям авторов (см. рис. 5).

Для того чтобы установить оптимальную ширину столбца списка, надо выполнить двойной щелчок мышью на правой границе столбца в строке заголовков полей. Установите оптимальную ширину для всех столбцов списка книг, как это сделано на рис. 5.

Автор	Название	Наименование	Год издания
Арсак	Программирование	Наука	1989
Беспалько	Педагогика	Мир	1994
Сканави	Сборник задач	Мир	1992
Фигурнов	IBM PC для пользователя	Финансы и статистика	1994

Рис. 5. Результат выполнения запроса.

После того как запрос создан, его можно сохранить. Для этой цели надо выполнить команду **Сохранить запрос** или **Сохранить запрос как** в меню **Файл**. Если мы выполняем сохранение первый раз, то выполнение этих команд приводит к одному и тому же результату – на экране появляется окно диалога, приведенное на рис. 6.

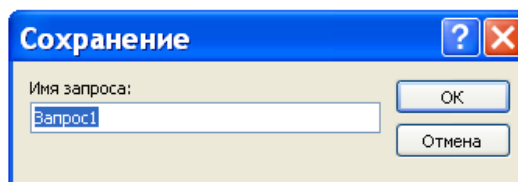


Рис. 6. Окно диалога для сохранения запроса.

Сохраните созданный нами запрос под именем **Список книг московских издательств**. Для этого введите новое имя (старое имя **Запрос1**, которое

предложил Access, после нажатия первой клавиши исчезнет, так что нет необходимости специально его убирать) и нажмите кнопку **ОК**.

Отбор данных

Основное назначение запроса состоит в формировании динамического набора, записи которого удовлетворяют некоторым условиям. Условия отбора записей вводятся как выражения. Выражение указывает, какие записи следует включить в динамический набор при выполнении запроса. Выражения могут быть простыми (например, <30) или сложными (например, Between 100 And 500).

Определить условия отбора можно самостоятельно, введя нужное выражение в ячейку **Условия отбора**, соответствующую данному полю, или воспользоваться построителем выражения. Для определения условия с помощью построителя выражений вначале устанавливают указатель в ячейку **Условия отбора** в бланке **QBE**, в которой следует определить выражение, и нажимают кнопку мыши. После этого нажимают кнопку **Построить** на панели инструментов. На экране появляется диалоговое окно **Построитель выражений**, приведенное рис. 7.

Если в ячейке бланка **QBE**, из которой вызывался построитель, содержится значение, то это значение автоматически копируется в поле построения выражения. Используя построитель выражений, можно вводить символы в область ввода или нажимать кнопки для ввода операторов, а также вставлять ссылки на объекты и другие элементы выражения, выбирая их из папок.

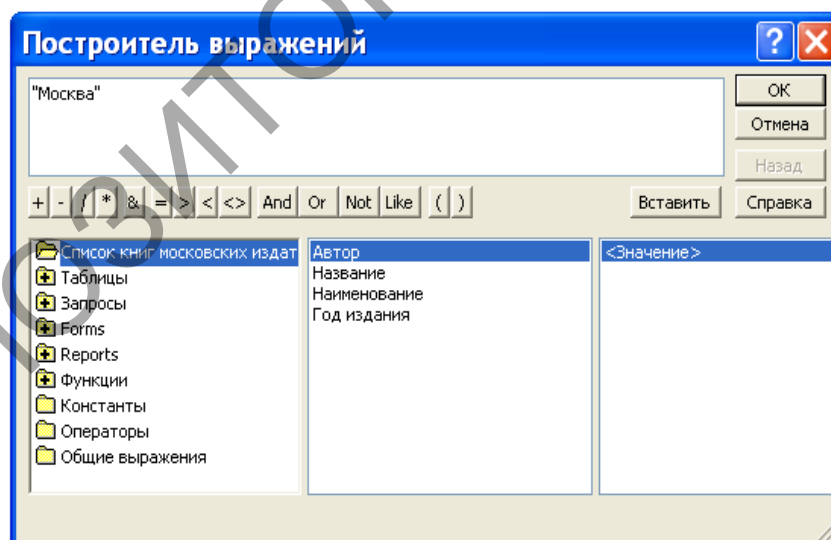


Рис. 7. Окно диалога **Построитель выражений**.

Вставка операторов в выражение из строки операторов, расположенной ниже поля построителя, выполняется щелчком мыши на операторе.

Для вставки элемента поступают следующим образом. В левом нижнем поле построителя выбирают папку, содержащую нужный элемент. В нижнем среднем поле дважды щелкают элемент, чтобы вставить его в поле выражения, или выбирают тип элемента. Если выбран тип в нижнем среднем поле, то значения будут отображаться в нижнем правом поле. Дважды щелкните значение, чтобы вставить его в поле выражения.

Вставьте необходимые операторы в выражение. Для этого поместите указатель мыши в определенную позицию поля выражения и выберите одну из кнопок со знаками операций, расположенных в середине окна построителя. Закончив создание выражения, нажмите кнопку **ОК**.

MS Access скопирует созданное выражение в ту позицию, из которой был вызван построитель выражений. Если в данной позиции уже содержится значение, то исходное значение будет заменено новым выражением.

Следует иметь в виду, что любая часть выражения или все выражение может быть введено в поле выражения непосредственно с клавиатуры. Может также случиться, что выражение можно быстрее ввести в строку **Условие отбора** без использования построителя выражений.

Выражение – комбинация операторов, констант, литералов, значений, функций, названий свойств, имен полей и элементов управления, при оценке которых получается одно значение. Оператор – это символ или слово (например, > или Or), указывающее на операцию, которую следует выполнить над одним или несколькими элементами. Операторы сгруппированы в классы операторов, например, арифметические, сравнения, логические.

В выражениях для условий отбора допускается использование символов шаблона. Символами шаблона являются звездочка (*), знак вопроса (?), знак номера (#), восклицательный знак (!), дефис (-) и квадратные скобки ([]). Эти символы можно использовать в запросах, командах и выражениях для включения всех записей, имен файлов или других элементов, которые начинаются с определенной последовательности букв или удовлетворяют указанному шаблону. Назначение и примеры использования символов шаблонов приведены в таблице 1. При вводе шаблонов можно использовать как прописные, так и строчные буквы. Например, шаблон "ст*" эквивалентен шаблону "Ст*".

Таблица 1

Символы шаблона

Символ	Назначение	Пример	Результат отбора
*	Заменяет любую группу символов; может быть первым или последним символом в шаблоне.	ст* *иск	"стол", "станок" и т.п. "иск", "диск", "риск" и т.п.
?	Заменяет любой один символ.	ко?a	"кора", "коса", "коза" и т.п.
#	Заменяет любую одну цифру.	5#4	504, 554, 514 и т.п.
[]	Заменяет любой один символ, указанный в скобках.	ко[рс]а	"кора" и "коса", но не коза
!	Заменяет любой один символ, кроме символов, указанных в скобках.	ко[!рс]а	"коза" и "кожа", но не "кора" и "коса"
-	Заменяет любой один символ из указанного диапазона.	ко[к-м]а	"кока", "кола" и "кома"

После завершения ввода выражения в ячейку строки **Условие отбора** (например, нажатием клавиши Enter, клавиш управления курсором или щелчком мыши в другой ячейке) выполняется синтаксический анализ этого выражения и выражение приводится в соответствие с правилами синтаксиса MS Access. Например, если введено слово Москва, то добавляются прямые кавычки и это слово выводится как "Москва".

Если выражение не содержит оператор, то подразумевается оператор равенства (=). Например, если в ячейку **Условие отбора** для поля **Город** введено слово Москва, то выражение интерпретируется как **Город** = "Москва".

Задание

1. Выведите список книг, цена которых находится в диапазоне от 20 до 30 тыс. рублей. Динамический набор этого запроса должен содержать поля: **Автор, Название, Год издания, Стоимость**. Для задания условия отбора вначале используйте оператор **Between ... And**, а затем операторы **>=, <=, And**. Записи в динамическом наборе расположите по возрастанию цены книг. Сохраните первый запрос под именем **Цена книг из диапазона**, а второй – под именем **Операторы сравнения для поиска цены**.

2. Выведите список читателей, у которых нет домашнего телефона. В список включите следующие поля: **Фамилия, Имя, Отчество, Домашний адрес**. Список рассортируйте в алфавитном порядке по фамилии, имени и отчеству. Для поиска требуемых записей в строке **Условие отбора** для поля **Домашний телефон** используйте выражение **Is Null**. Это выражение предназначено для поиска записей, у которых поле не содержит значение (является пустым). Если требуется отобрать записи, у которых поле имеет значение, то можно использовать выражение **Is Not Null**. Запрос сохраните под именем **Читатели без домашних телефонов**.

3. Выведите список читателей, у которых в домашнем телефоне вторая цифра есть 5 или 6. В динамический набор включите поля: **Фамилия, Имя, Отчество, Домашний телефон**. Условие отбора для поля может иметь следующий вид: **Like "[56]*"**.

Запрос сохраните под именем **Использование символов шаблона**. Измените условие отбора предыдущего запроса так, чтобы он выводил список всех читателей, в номерах телефонов которых вторая цифра не 5 и не 6. Полученный запрос сохраните под именем **Символ отрицания в квадратных скобках**.

4. Создайте запрос, который будет выводить список книг, заказанных читателями в 2007 году. В динамический набор включите следующие поля: **Автор, Название, Наименование, Город**. Для решения задачи вначале используйте функцию **DatePart(interval; date; firstweekday; firstweek)**, а затем **Format(expr; fmt; firstweekday; firstweek)**. Запросы сохраните под именами

Использование функции DatePart и Использование функции Format соответственно.

5. В таблицу **Выдача книг** базы данных **Библиотека** добавьте поле **Дата возврата**. В это поле для записей, приведенных в таблице 2, введите даты возврата. В остальных записях поле **Дата возврата** должно остаться пустым.

Таблица 2

Учет возврата книг

Код читателя	Код книги	Дата заказа	Дата возврата
1	1	01.09.2015	15.10.2016
1	3	05.07.2016	23.09.2016
4	3	07.01.2016	02.03.2016
5	2	23.04.2016	03.05.2016
7	3	20.01.2016	11.04.2016
9	6	02.02.2016	03.03.2016

Составьте запрос, который будет выводить список читателей, которые не сдали своевременно книги (предполагается, что читатель может держать книгу на руках не более 100 дней). В динамический набор включите следующие поля: **Фамилия, Имя, Отчество, Домашний телефон, Автор, Название, Стоимость**. Для решения задачи воспользуйтесь функцией **Date()**. Запрос сохраните под именем **Читатели, не сдавшие своевременно книги**.

Тема 6. Планирование, проведение и обработка результатов научных исследований

Лабораторная работа 1.

Обработка результатов эксперимента

Цель работы: Изучить возможности обработки данных научного эксперимента с помощью Excel.

Задание 1. Пользуясь сервисом «Анализ данных» приложения Excel выполнить следующие задания.

1. Имеются данные по двум экономическим показателям X и Y:

Цена (X)	995	983	1001	1012	1011	1017	978	997	1010	989	900	1100	5000
Спрос (Y)	122	144	114	100	100	90	150	130	95	155	?	?	?

Необходимо:

вычислить коэффициент корреляции;

построить корреляционное поле (диаграмму) на отдельном листе;

построить регрессионную модель (с использованием функции ЛИНЕЙН);

спрогнозировать значение Y для 3-х новых значений X с помощью функции– ПРЕДСКАЗ

Задание 2. Имеются данные научного эксперимента, представленные в таблице. Найти попарные корреляции переменных X_i . Вычислить математическое ожидание и дисперсию для каждой из переменных. Построить модели линейных регрессий для наиболее коррелированных переменных, беря в качестве независимой переменной сначала одну, потом другую переменную. Сравнить результаты.

Тема 7. Методы оптимизации и информационные системы в управлении, образовании, науке и творческой деятельности

Лабораторная работа 1. Решение задач оптимизации

Цель работы: Изучить возможности решения задач оптимизации с помощью Excel.

Задание 1. Требуется определить план выпуска четырех видов продукции, обеспечивающий максимальную прибыль от ее реализации.

На изготовление этой продукции расходуются трудовые ресурсы, сырье и финансы. С учетом рыночного спроса и производственно-технологических возможностей заданы предельные границы выпуска каждого вида продукции.

Эти границы, наличие и нормы расхода ресурсов, а также маржинальная прибыль (разность между выручкой и переменными издержками) на единицу продукции приведены в таблице:

Ресурсы	Продукт 1	Продукт 2	Продукт 3	Продукт 4	Наличие ресурса
Трудовые	1	2	1	2	19
Сырье	7	4	5	4	80
Финансы	5	7	9	8	100
Прибыль	70	60	100	140	-
Нижняя граница	3	1	1	2	
Верхняя граница	5	-	3	4	

Порядок выполнения

В первой строке таблицы находится заголовок, во второй — наименования продуктов.

Третья строка отведена для оптимального решения, которое после вычислений появится в ячейках В3:Е3 (в жирной рамке).

В четвертой строке в ячейках В4:Е4 заданы коэффициенты целевой функции, а ячейка F4, в рамке, зарезервирована для вычисления значения целевой функции.

Строки с 6-й по 15-ю содержат коэффициенты, знаки и правые части ограничений.

В столбце **Лев. часть** после вычислений появятся левые части ограничений, а в столбце **Разница** — разность правых и левых частей.

Предложенная форма таблицы удобна не только для выполнения расчетов, но и для документирования результатов моделирования, а также обмена идеями и улучшения взаимопонимания при работе «в команде» — группе лиц, ответственных за принятие решений.

Наглядность таблицы обеспечивается использованием заголовков, полужирного шрифта, подчеркивания текста и центрирования.

Денежные величины следует представлять в денежном или финансовом формате (с указанием символа используемой валюты).

	A	B	C	D	E	F	G	H	I
1	Производственный план								
2	Продукт	Прод. 1	Прод. 2	Прод. 3	Прод. 4				Наличие
3	Количество					Прибыль			
4	Прибыль/ед.	70,0р.	60,0р.	110,0р.	140,0р.				
5	Ограничения	Расход ресурсов				Лев. Часть		Прав. Часть	Разница
6	Трудовые	1	2	1	2		≤	19	
7	Сырье	7	4	5	4		≤	80	
8	Финансы	5,0р.	7,0р.	9,0р.	8,0р.		≤	100,0р.	
9	Мин. П1	1					≥	3	
10	Мин. П2		1				≥	1	
11	Мин. П3			1			≥	1	
12	Мин. П4				1		≥	2	
13	Мак. П1	1					≤	5	
14	Мак. П3			1			≤	3	
15	Мак. П4				1		≤	4	
16									

Ввод данных для решения задачи линейного программирования

Оформление таблицы облегчают кнопки панели форматирования, в частности: **Объединить и поместить в центре** (для центрирования заголовка, занимающего несколько ячеек), **Денежный формат** (для задания финансового формата), **Увеличить разрядность** или **Уменьшить разрядность** (для задания нужного числа дробных знаков).

Чтобы отобразить знак > или <, наберите знак > или < и затем подчеркните его.

После составления таблицы подберите ширину ее столбцов в соответствии с содержимым ячеек. Для этого выделите весь лист, щелкнув прямоугольник в левом верхнем углу листа (на пересечении заголовков строк и столбцов). Затем дважды щелкните правую границу заголовка любого из столбцов.

Формулы и ячейки, в которые их следует ввести, указаны в таблице:

Ячейка	Формула	Копировать в диапазон ячеек
F4	=СУММПРОИЗВ(\$B\$3:\$E\$3;B4:E4)	F6:F15
I6	=H6-F6	I7:I8; I13:I15
I9	=F9-H9	I10:I12

Чтобы ввести, например, формулу для вычисления целевой функции:

1. Укажите ячейку F4.

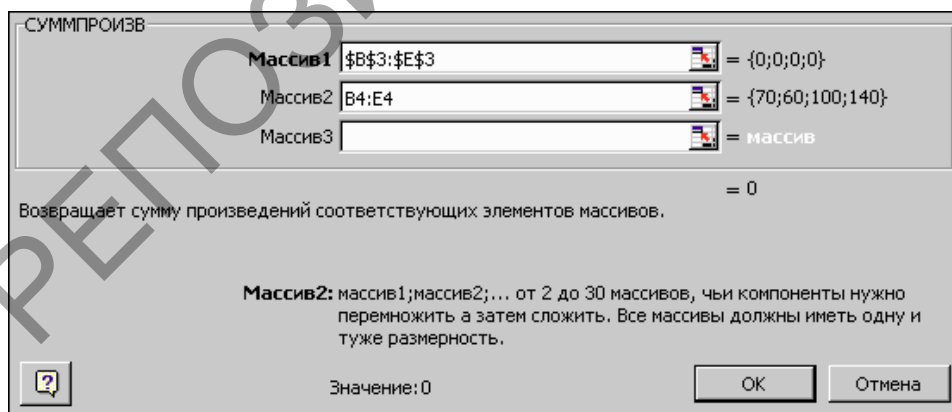
2. В строке формул щелкните кнопку Изменить формулу (со знаком равенства). Раскроется панель формул.

3. В левой части строки формул раскройте список функций и щелкните имя функции СУММПРОИЗВ. Если его там нет, выберите в списке пункт **Другие функции**. В открывшемся окне в категории Математические найдите функцию СУММПРОИЗВ и дважды щелкните ее имя. **Панель формул** примет вид, позволяющий задать аргументы этой функции (рис. 2).

4. Введите аргументы функции СУММПРОИЗВ.

5. Щелкните кнопку ОК.

Сразу после ввода формулы в ячейке F4 появится 0, так как формула вычисляется с нулевыми значениями переменных (ячейки B3:E3 пока пусты).



Ввод формулы целевой функции задачи линейного программирования

Функцией СУММПРОИЗВ позволяет вычислить сумму произведений двух массивов, первый из которых содержит значения переменных, а второй — коэффициенты целевой функции. Чтобы указать соответствующие диапазоны, можно воспользоваться кнопками свертывания, расположенными справа от

полей ввода. Они позволяют временно убрать панель формул с экрана, чтобы удобнее было выделять диапазон на листе. Закончив выделение, щелкните кнопку снова для восстановления панели.

Ссылка на первый диапазон должна быть абсолютной, со знаками доллара перед каждой буквой и цифрой: $\$B\$3:\$E\3 (чтобы изменить относительную ссылку на абсолютную, нажмите клавишу F4 непосредственно после ввода этой ссылки; если это не сделать сразу, то в дальнейшем ссылку сначала понадобится выделить и лишь затем нажать клавишу F4).

Ссылка на второй диапазон B4:E4, напротив, должна быть относительной: что понадобится в дальнейшем при копировании формулы.

После ввода формулы для вычисления целевой функции необходимо задать формулы левых частей ограничений. С этой целью скопируйте формулу из ячейки F4 в ячейки F6:F15. Чтобы копировалась только сама формула (без формата ячейки, к которому относятся, например, рамка и символ используемой валюты) можно воспользоваться специальной вставкой.

Для этого щелкните ячейку F4 правой кнопкой мыши и в контекстном меню выберите команду **Копировать**, затем выделите диапазон F6:F15, щелкните его правой кнопкой мыши и в контекстном меню выберите команду **Специальная вставка**. В группе **Вставить** открывшегося окна выберите параметр формулы и щелкните кнопку ОК. Чтобы отменить движущуюся границу ячейки F4, нажмите клавишу Esc.

При копировании относительная ссылка B4:E4 будет меняться, указывая на массивы коэффициентов соответствующих ограничений, а абсолютная — $\$B\$3:\$E\3 — останется неизменной.

Для проверки формулы в ячейке достаточно дважды щелкнуть эту ячейку. Формула будет отображена на экране, ссылки на ячейки — выделены цветом, а сами ячейки — обведены рамками соответствующих цветов (после просмотра нажмите клавишу Esc).

Чтобы отображались все введенные формулы, выберите в меню **Сервис** команду **Параметры** и в открывшемся окне на вкладке **Вид** в группе **Параметры окна** установите флажок **Формулы**.

Если таблицу с формулами необходимо напечатать, подберите ширину ее столбцов в соответствии с содержимым ячеек, как это было описано ранее.

Аналогично, с помощью копирования, вводятся формулы в ячейки I6:I15 для вычисления разницы правых и левых частей ограничений. Для ограничений со знаком < из правой части ограничения вычитается левая, а для ограничений со знаком > — наоборот.

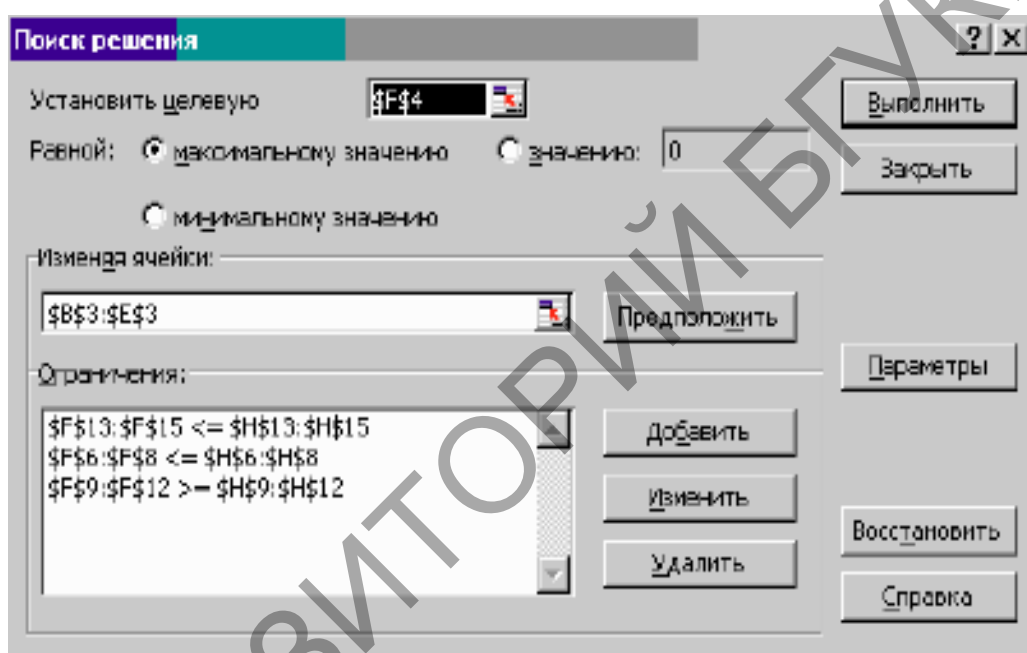
Таким образом, если ограничения выполняются и план допустим, то разница всегда неотрицательна.

После ввода формул вы можете подставлять в ячейки В3:Е3 любые значения (количество выпускаемой продукции), получая соответствующую величину прибыли в ячейке F4 и объем израсходованных ресурсов в столбце Лев.часть.

Столбец Разница покажет, допустим ли соответствующий производственный план, какие ресурсы дефицитны (для них разница равна нулю) и каков остаток недефицитных ресурсов (он равен положительной разнице).

Теперь воспользуемся средством поиска решений.

Для этого в меню **Сервис** выберите команду **Поиск решения**, которая откроет одноименное окно. При решении нашей задачи это окно должно быть заполнено, как показано на рисунке.



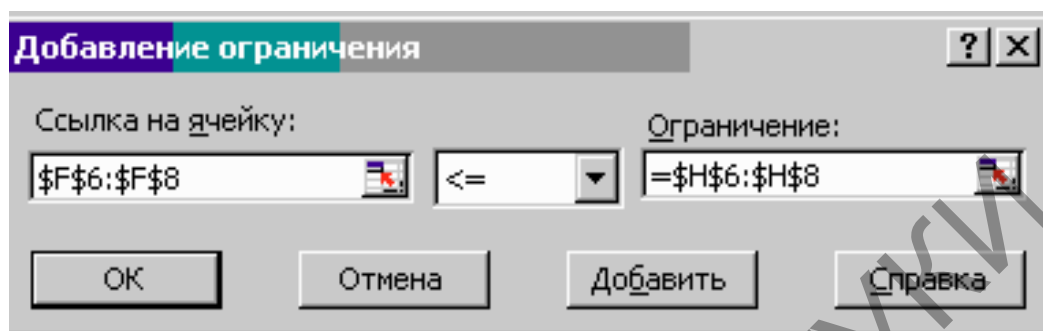
Окно средства поиска решения при решении задачи линейного программирования

В окне **Поиск решения** вы должны задать следующие параметры:

- **Установить целевую ячейку**— для указания целевой ячейки, значение которой необходимо максимизировать, минимизировать или установить равным заданному числу. Эта ячейка должна содержать формулу для вычисления целевой функции.
- **Равной** — для выбора варианта оптимизации значения целевой ячейки (максимизация, минимизация или подбор заданного числа).
- **Изменяя ячейки** — для указания ячеек, значения которых изменяются в процессе поиска решения до тех пор, пока не будут выполнены наложенные ограничения и условие оптимизации значения целевой ячейки. В нашем примере это ячейки, отведенные для значений переменных.
- **Ограничения** — для отображения списка ограничений поставленной задачи.

При вводе ссылок на ячейки (как и при задании аргументов функции СУММПРОИЗВ) удобно пользоваться кнопками свертывания диалогового окна, расположенными справа от соответствующих полей.

Для ввода ограничений следует нажать кнопку **Добавить**, открывающую окно **Добавление ограничения**



Добавление ограничения

В этом окне — два поля ввода:

- **Ссылка на ячейку** — для указания ячейки или диапазона ячеек, на значения которых необходимо наложить ограничение («левая часть ограничения»).

- **Ограничение** — для задания условия, накладываемого на значения ячейки или диапазона, указанного в поле Ссылка на ячейку («правая часть ограничения»). В это поле можно ввести число, формулу, ссылку на ячейку или диапазон.

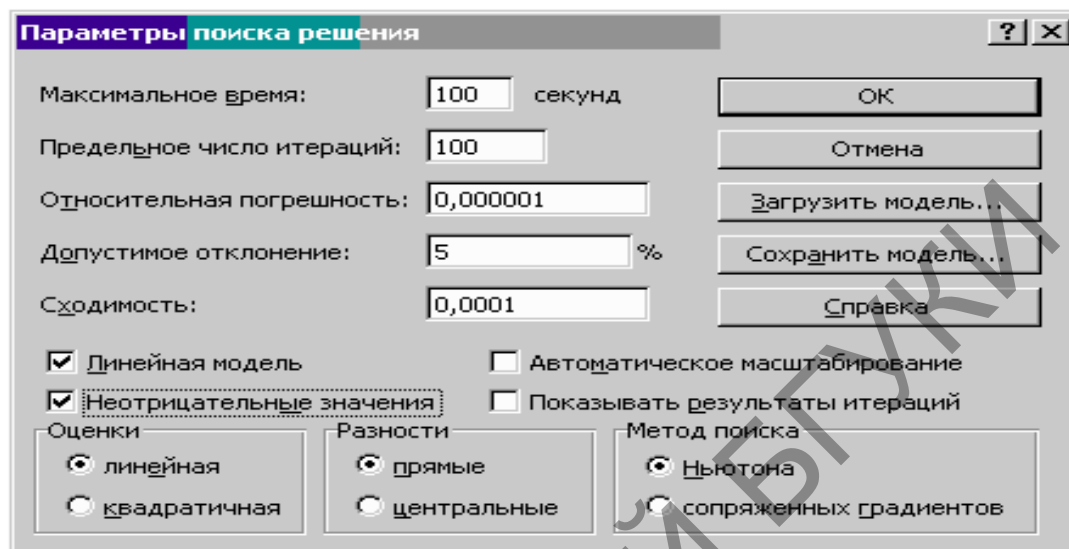
В раскрывающемся списке между этими полями можно выбрать необходимый знак (<=, =, >=), а также пункты **цел** (целое число) или **двоич** (двоичное число — 0 или 1), указывающие вид переменных при целочисленном решении.

Для ввода ссылок на ячейки и диапазоны можно, как и раньше, воспользоваться кнопками свертывания диалогового окна, расположенными справа от полей ввода. Но поскольку окно **Добавление ограничения** и так достаточно невелико, удобнее просто перетащить его вниз экрана, поместив указатель мыши на заголовке. Затем следует щелкнуть в нужном поле этого окна и выделить на рабочем листе ячейку или диапазон, на который делается ссылка.

Каждое ограничение задачи можно вводить индивидуально. Однако когда знаки ограничений одинаковы, удобнее указывать их левые и правые части в виде диапазонов ячеек. После ввода каждого ограничения, кроме последнего, следует нажимать кнопку **Добавить**. При этом поля окна **Добавление ограничения** очищаются и можно вводить следующее ограничение. После ввода последнего из них, щелкните кнопку **ОК**, чтобы вернуться в окно **Поиск**

решения, где вы увидите список всех введенных ограничений. При необходимости их изменения или удаления можно выделить любое из ограничений и нажать кнопку, соответственно, **Изменить** или **Удалить**.

Чтобы задать другие параметры поиска решения, щелкните кнопку **Параметры**, открывающую окно **Параметры поиска решения**,

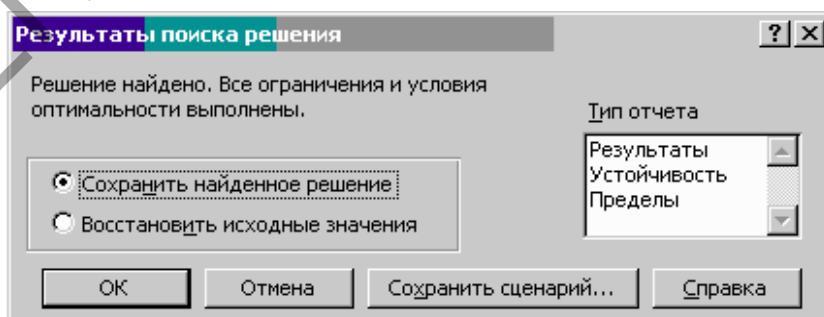


Выбор дополнительных параметров поиска решения

Значения большинства параметров, установленные по умолчанию, годятся для решения большинства задач. Однако специально для решения задач линейного программирования следует установить два флажка:

- **Линейная модель** — для ускорения поиска решения линейной задачи оптимизации и получения результатов после оптимизационного анализа;
- **Неотрицательные значения** — чтобы переменные были неотрицательны.

Вернувшись в окно **Поиск решения**, вы можете приступить к вычислениям. Для этого щелкните кнопку **Выполнить**. Результаты вычислений появятся на рабочем листе, и одновременно откроется окно **Результаты поиска решения** с сообщением о том, найдено решение или нет.



Сообщение о результатах решения задачи линейного программирования

В этом окне можно выбрать следующие параметры:

- **Сохранить найденное решение** — для сохранения найденного решения на рабочем листе.

- **Восстановить исходные значения** — для восстановления исходного вида рабочего листа.

- **Тип отчета** — для выдачи на отдельных рабочих листах отчетов, содержащих анализ полученных результатов. В списке можно выделить один или несколько типов отчета (в случае целочисленного решения доступен лишь один из них— Результаты). Отчет по результатам содержит ту же информацию, что и основной рабочий лист, отчет по пределам не представляет особой ценности, поэтому в дальнейшем подробно анализируется лишь наиболее интересный из отчетов — по устойчивости.

В нашей задаче выберите параметр **Сохранить найденное решение**, выделите в списке тип отчета — **Устойчивость** и щелкните кнопку **ОК**. В рабочей книге появится новый лист с этим отчетом.

Результаты вычислений выглядят на рабочем листе, как показано на рисунке.

	A	B	C	D	E	F	G	H	I
1		Производственный план							
2	Продукт	Прод. 1	Прод. 2	Прод. 3	Прод. 4				Наличие
3	Количество	5	15	3	4	Прибыль			
4	Прибыль/ед.	70,0р.	60,0р.	110,0р.	140,0р.	1330			
5	Ограничения	Расход ресурсов				Лев. Часть		Прав. Часть	Разница
6	Трудовые	1	2	1	2	19	≤	19	0
7	Сырье	7	4	5	4	72	≤	80	8
8	Финансы	5,0р.	7,0р.	9,0р.	8,0р.	94,5	≤	100,0р.	5,5
9	Мин. П1	1				5	≥	3	2
10	Мин. П2		1			1,5	≥	1	0,5
11	Мин. П3			1		3	≥	1	2
12	Мин. П4				1	4	≥	2	2
13	Мак. П1	1				5	≤	5	0
14	Мак. П3			1		3	≤	3	0
15	Мак. П4				1	4	≤	4	0

Результаты решения задачи линейного программирования

В ячейках В3:Е3 выводится оптимальный производственный план, в ячейке F4 — максимальное значение прибыли.

В строках 8-10, содержащих ограничения на ресурсы, в столбце Лев.часть указаны объемы и1080 использованных ресурсов, а в столбце Разница — остаток ресурсов, оставшихся неиспользованными.

Если разница равна нулю, то соответствующий ресурс дефицитен (используется полностью), а если разница положительна — то недефицитен (не используется полностью).

В строках 9–15, содержащих граничные условия, в столбце Лев.часть указаны значения переменных в оптимальном решении, а в столбце Разница —

разность между этими значениями и заданными для переменных верхними и нижними границами.

Если разница равна нулю, то соответствующая переменная принимает свое минимальное или максимальное значение.

4 РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

4.1 Задания для контролируемой самостоятельной работы студентов

Самостоятельная работа студентов направлена на совершенствование их умений и навыков по дисциплине «Основы информационных технологий». Цель самостоятельной работы студентов – способствование усвоению в полном объеме учебного материала дисциплины через систематизацию, планирование и контроль собственной деятельности. Преподаватель дает задания по самостоятельной работе и регулярно проверяет их исполнение.

4.2 Контрольные вопросы

1. Какие протоколы передачи данных вы знаете? Какой протокол является базовым в Интернете?
2. Перечислите все известные Вам средства поиска информационных ресурсов в Сети.
3. Каков коэффициент полноты и точности поиска в идеальной поисковой системе?
4. Какой из критериев качества информационного поиска является более значимым: полнота или точность?
5. Опишите известные Вам технологии сохранения информации из глобальной компьютерной сети Интернет.
6. Какие специальные символы и операторы, которые используются в информационных запросах, Вы знаете?

4.3 Перечень вопросов к кандидатскому зачету

1. Понятия информации и информационных технологий. Информационная деятельность. Информация как отображение окружающего мира.
2. Функционирование информации в обществе. Информационные потребности. Информатизация и компьютеризация.
3. Виды информации. Элементарные схемы передачи и распространения информации.

4. Информационные процессы в культуре, образовании и научных исследованиях.
5. Особенности передачи, сохранения и распространения информации в сфере культуры и искусств.
6. Понятие и виды информационных систем. Аппаратное и программное обеспечение.
7. Классификация информационных технологий, аппаратных и программных средств.
8. Защита информации. Электронная цифровая подпись.
9. Методы и средства защиты информации.
10. Классы безопасности компьютерных систем.
11. Организационно-правовые аспекты защиты информации и авторское право.
12. Концептуальные основы информационной поддержки сферы культуры и искусств.
13. Концептуальные основы информационного обеспечения в сферах культуры и образования.
14. Вычислительная техника: этапы развития, классификация ЭВМ.
15. Структура современных персональных компьютеров. Основные направления повышения скорости обработки информации.
16. Классификация средств компьютерной техники по функциональному назначению. Современные персональные компьютеры.
17. Компьютерные сети, их назначение. Локальные, корпоративные, глобальные сети: обмен информацией и особенности обработки информации.
18. Понятие и классификация программного обеспечения. Операционные системы.
19. Текстовые процессоры и редакционно-издательские системы: функции, использование и назначение. Ввод, обработка и вывод текста.
20. Табличные процессоры. Система функций электронных таблиц. Построение диаграмм.
21. Системы подготовки презентаций: инструментарий, методики, функции.
22. Основы компьютерной графики. Понятие векторной и растровой графики. Методы обработки. Основные форматы графических файлов.
23. Статическая графика. Создание, преобразование и вывод графической информации.
24. Динамическая графика. Понятие анимации. 2d- и 3d-анимация. Этапы создания анимационных материалов.
25. Компьютерная графика в изобразительном искусстве. Средства и методы обработки графических изображений.
26. Основы web-дизайна.
27. Обработка звука. Спектр звука. Методы синтеза спектра.
28. Цифровая обработка звука. Дискретизация и квантование. Теорема Котельникова.

29. Понятие музыкального звука. Средства ввода, обработки и вывода звука.
30. Музыкальные редакторы: функции и назначение.
31. Компьютерные системы обработки музыки.
32. Использование средств компьютерной техники в музыкальном творчестве.
33. Теория и практика компьютерной записи, сохранения и обработки звука.
34. Работа с микрофоном и микшером в процессе записи голоса исполнителя и музыкальных инструментов.
35. Звуковые эффекты (вибрато, дилэй, флэнжер, фэйзер, хорус, реверберация, дисторшн, вокодер).
36. Основные принципы построения виртуальных аудиостудий.
37. Основы создания компьютерной музыки.
38. Мультимедийные системы в культуре и искусствах.
39. Мультимедийные системы в образовании, библиотечном и музейном деле.
40. Перспективные направления использования мультимедийных компьютерных систем в учреждениях культуры.
41. Сервисные инструментальные средства: архиваторы, электронные словари, переводчики, программы распознавания текста.
42. Базы данных. Понятие базы данных как отображения предметной области. Модели баз данных.
43. Понятие системы управления базами данных и ее функции.
44. Проектирование баз данных. Инфологическое моделирование.
45. Этапы разработки баз данных.
46. Основы языка SQL и построение SQL-запросов.
47. Понятие целостности данных. Обеспечение целостности данных в MS Access.
48. Создание структуры таблиц и установление связей между ними в MS Access. Типы данных и связей.
49. Виды запросов в MS Access и их создание: запрос на выборку, запрос с вычисляемыми полями, запрос с параметрами, перекрестный запрос.
50. Перспективные направления разработки автоматизированных информационных систем для отрасли культуры и искусств.
51. Перспективные направления разработки автоматизированных информационных систем для отрасли образования.
52. Применение баз данных в библиотечной деятельности.
53. Отличительные характеристики библиотечных баз данных и особенности их внедрения и использования.
54. Автоматизированные библиотечно-информационные системы: состав, функции и использование.
55. Понятие о коммуникационном формате UNIMARC и его применение в библиотеках Республики Беларусь.

56. Автоматизированные библиотечные информационные системы Республики Беларусь. Основные характеристики, перспективы развития и использования.

57. Компьютерные сети. Глобальная сеть Интернет.

58. Ресурсы Интернета, доступ к ним. Просмотр и навигация в Интернете.

59. Основные методы адресации компьютеров в сети.

60. Провайдеры Интернет-услуг в Беларуси.

61. Информационные ресурсы Интернета в сфере культуры и искусств.

62. Службы сети Интернет: электронная почта, WWW, служба передачи файлов, служба телеконференций и др.

63. Проблемы защиты информации в компьютерных системах.

64. Методы защиты информации в компьютерных системах.

65. Пути решения проблемы информатизации общества: новые технические средства и программные продукты, интеллектуализация средств информационных технологий.

66. Внедрение информационных технологий в образование и культуру. Государственные программы информатизации образования и культуры Беларуси.

4.4 Критерии оценки результатов учебной деятельности студентов

Для выявления и исключения пробелов в знаниях студентов рекомендуется использовать следующие средства:

- 1) фронтальный опрос на лекциях и лабораторных;
- 2) критериально-ориентированные тесты для контроля теоретических знаний современных информационных технологий, основных определений, терминологии;
- 3) выполнение тестовых заданий с произвольной формой ответа для контроля умения анализировать и грамотно излагать и формулировать свои соображения и выводы в данной предметной области;
- 4) выполнение творческих заданий, которые предполагают эвристическую деятельность и поиск неформальных решений.

5 ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

5.1 Учебная программа

Гляков П.В. Основы информационных технологий: Учебная программа учреждения высшего образования по общеобразовательной дисциплине для специальностей второй ступени высшего образования университета / П.В. Гляков, С.А. Гончарова, М.Н. Другакова, . – Минск : БГУКИ, 2014. – 22 с.

5.2 Учебно-методическая карта учебной дисциплины для дневной формы получения высшего образования

№ п/п	Наименование темы	Количество аудиторных часов		Форма контроля знаний
		Лекции	Лабораторные занятия	
1	Современные информационные технологии	4	4	
2	Основные программные средства информационных технологий	8	14	
3	Сетевые технологии и Интернет	8	4	
4	Системы управления базами данных	4	6	
5	Защита информации	4	4	
6	Планирование, проведение и обработка результатов научных исследований	4	2	
7	Методы оптимизации и информационные системы в управлении, образовании, науке и творческой деятельности	4	2	
Итого		36	36	Кандидатский зачет

5.3 Учебно-методическая карта учебной дисциплины для заочной формы получения высшего образования

№ п/п	Наименование темы	Количество аудиторных часов		Форма контроля знаний
		Лекции	Лабораторные занятия	
1	Современные информационные технологии	2		

2	Основные программные средства информационных технологий	2	4	
3	Сетевые технологии и Интернет		2	
4	Системы управления базами данных	2	2	
5	Защита информации		2	
6	Планирование, проведение и обработка результатов научных исследований		2	
Итого		36	36	кандидатский зачет

5.4 Список основной литературы

1. Акулов, О.А. Информатика: базовый курс : учебник для студентов высших учебных заведений, бакалавров, магистров по направлению "Информатика и вычислительная техника" / О.А. Акулов, Н.В. Медведев. – М. : Омега-Л, 2009. – 574 с.
2. Банди, Б. Методы оптимизации. Вводный курс / Б. Банди. – М. : Мир, 1989. – 277 с.
3. Бураўкін, А.Г. Інфармацыйныя тэхналогіі ў мастацтве / А.Г. Бураўкін ; пад рэд. М.А. Ярмаша. – Мінск : БУК, 1999. – 246 с.
4. Бьюли, А. Изучаем SQL / А. Бьюли. – СПб. : Изд. Символ-Плюс, 2007. – 312 с.
5. Гляков, П.В. Базы данных: компьютерный практикум: учебное пособие / П.В. Гляков. – Минск : БГУКИ, 2008. – 130 с.
6. Гляков, П.В. Импорт, экспорт и связывание данных в Microsoft Access : метод. рекомендации / П.В. Гляков. – Минск : РИПО, 2005. – 34 с.
7. Гляков, П.В. Система управления базами данных Access 2.0 : учеб. пособие / П.В. Гляков, С.Н. Карачун. – Минск : РИПО, 1998. – 100 с.
8. Гордеев, А. Операционные системы : учебник для вузов / А. Гордеев. – СПб. : Питер, 2009. – 416 с.
9. Жилинская, Т.С. Медиакультура специалиста : учебно-методическое пособие для студентов высших учебных заведений по специальности 1-210401 Культурология (по направлениям), специализации 1-210401-0204 Информационные системы в культуре / Т. С. Жилинская. – Минск : [б. и.], 2011. – с. 62.
10. Захарова, И.Г. Информационные технологии в образовании : учеб. пособие / И.Г. Захарова. – 4-е изд., стер. – М. : Академия, 2008. – 198 с.

11. Калинин, Н.Н. Численные методы / Н.Н. Калинин. – М. : Наука, 1978. – 512 с.
12. Кузнецов, М.В. Практика создания Web-сайтов / М.В. Кузнецов, И.В. Симдянов. – 2-е издание. – СПб. : БХВ-Петербург, 2009. – 1264 с.
13. Морозевич, А.Н. Информатика / А.Н. Морозевич, А.М. Зеневич. – Минск : Высшая школа, 2006. – 285 с.
14. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы / В.Г. Олифер, Н.А. Олифер. – СПб. : Питер, 2010. – 960 с.
15. Острейковский, В.А. Информатика / В.А. Острейковский. – М. : Высшая школа, 2009. – 510 с.
16. Сеннов, А.С. Access 2010. Учебный курс / А.С. Сеннов. – СПб. : Питер, 2010. – 288 с.
17. Стоцкий, Ю.А. Microsoft Office 2010 / Ю.А. Стоцкий, А.А. Васильев, И.С. Телина. – СПб. : Питер, 2011. – 425 с.
18. Черноруцкий, И.Г. Методы принятия решений / И.Г. Черноруцкий. – СПб. : БХВ-Петербург, 2005. – 416 с.
19. Шаньгин, В. Защита компьютерной информации. Эффективные методы и средства // В. Шаньгин. – Изд. МКД Пресс, 2010. – 544 с.

5.5 Список дополнительной литературы

Дополнительная

1. Батин, Н.В. Основы информационных технологий: учеб.-метод. пособие / Н.В. Батин [и др.], под общ. ред. В.В. Шкурко. – Минск : Институт подготовки научных кадров Национальной академии наук Беларуси, 2008. – 253 с.
2. Трофимов, В.В. Информатика / В.В. Трофимов [и др.] / под ред. В.В. Трофимова. – М. : Юрайт: Высшее образование, 2010. – 910 с.
3. Колосов, С.В. Программирование в среде Delphi / С.В. Колосов. – Минск : БГУИР, 2005. – 166 с.
4. Дьяконов, В. MathCad 8-12 для всех / В. Дьяконов. – М. : СОЛОН-Пресс, 2005. – 632 с.
5. Потемкин, В.Г. MATLAB 6: Среда проектирования инженерных приложений / В.Г. Потемкин. – М. : Диалог-МИФИ, 2003. – 444 с.
6. Хабрейкен, Дж. Изучи Microsoft PowerPoint 2002 за 10 минут / Дж. Хабрейкен. – М. : Изд. Вильямс, 2002. – 224 с.
7. Соломенчук, В. Интернет: краткий курс / В. Соломенчук. – СПб. : Питер, 2000. – 288 с.
8. Барсуков, В.С. Безопасность: Технологии, средства, услуги / В.С. Барсуков. – М. : КУДИЦ-ОБРАЗ, 2003. – 496 с.
9. Как противостоять вирусной атаке / Д. Зегжба [и др.]. – СПб. : ВHV - Санкт-Петербург, 1995. – 320 с.

10. Ливак, Е.Н. Защита информации: учебное пособие в 4 частях / Е.Н. Ливак. – Гродно : ГрГУ, 1998. – 200 с.
11. Лебедев, А.Н. Моделирование в научно-технических исследованиях / А.Н. Лебедев. – М. : Радио и связь, 1989. – 224 с.
12. Марчук, Г.И. Методы вычислительной математики / Г.И. Марчук. – М. : Наука, 1987. – 534 с.
13. Марчук, Г.И. Введение в проекционно-сеточные методы / Г.И. Марчук, В.И. Агошков. – М. : Наука, 1981. – 414 с.
14. Гилл, Ф. Численные методы условной оптимизации / Ф. Гилл, У. Мюррей. – М. : Мир, 1977. – 297 с.
15. Алексеев, В.М. Оптимальное управление / В.М. Алексеев, В.М. Тихомиров, С.В. Фомин. – М. : Наука, 1980. – 429 с.
16. Закон Республики Беларусь от 10 ноября 2008 г. № 455-З "Об информации, информатизации и защите информации", с изменениями и дополнениями, внесенными Законом Республики Беларусь от 4 января 2014 г. № 102-З // Национальный правовой Интернет-портал Республики Беларусь, 10.01.2014 [Электронный ресурс]. – Режим доступа : <http://pravo.by/>. – Дата доступа : 21.01.2015.
17. Гурский, Д.А. Вычисления в MathCAD / Д.А. Гурский. – Минск : Новое знание, 2003. – 814 с.
18. Дьяконов, В.П. Maple 9 в математике, физике и образовании / В.П. Дьяконов. – М. : СОЛОН-Пресс, 2004. – 688 с.
19. Очков, В.Ф. Mathcad 12 для студентов и инженеров / В.Ф. Очков. – СПб. : БХВ-Петербург, 2005. – 464 с.
20. Половко, А.М. Mathcad для студента / А.М. Половко, И.В. Ганичев. – СПб. : БХВ-Петербург, 2006. – 336 с.
21. Титоренко, Г.А. Автоматизированные информационные технологии в экономике / Г.А. Титоренко. – М.: ЮНИТИ, 2008.
22. Быкова, Е.В., Стоянова Е.С. Финансовое искусство коммерции / Е.В. Быкова, Е.С. Стоянова. – М.: Перспектива, 2009.