

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени А. А. КУЛЕШОВА»

С. Н. Батан, Л. В. Батан,
О. В. Малашук

ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Курс лекций



Могилев
МГУ имени А. А. Кулешова
2016

УДК 004(075.8)
ББК 32.81я73
Б28

*Печатается по решению редакционно-издательского совета
МГУ имени А. А. Кулешова*

Рецензент

кандидат физико-математических наук, доцент
Могилевского государственного университета продовольствия
В. Л. Тутов

Батан, С.Н.

Б28 Основы информационных технологий : курс лекций / С. Н. Батан, Л. В. Батан, О. В. Малашук. – Могилев : МГУ имени А. А. Кулешова, 2016. – 120 с. : ил.
ISBN 978-985-568-146-6

Курс лекций предназначен для студентов первой и второй ступени высшего образования. Данный курс разработан с учетом современных достижений в области информационных технологий. Лекции помогут овладеть основами информационной культуры, освоить технологии работы в операционной системе и с основными прикладными пакетами, изучить принцип работы компьютерных сетей, сетевых средств поиска и обмена информацией, ознакомиться с методами защиты информации, овладеть компьютерным моделированием и информационными технологиями поддержки принятия решений.

**УДК 811.112.2 (075.8)
ББК 81.43.24**

ISBN 978-985-568-146-6

© Батан С.Н., Батан Л.В., Малашук О.В., 2016
© МГУ имени А. А. Кулешова, 2016

Глава 1

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1. Информация. Информационные технологии и системы

Строгого научного определения понятия “информация” нет. В различных науках слово “информация” понимается по-разному. Термин “**информация**” происходит от латинского “informatio”, означающего разъяснение, осведомление, содержание сообщения, сведения с учетом их передачи в пространстве и времени. С содержательной точки зрения “информация” – это сведения о ком-то или о чем-то, а с формальной точки зрения – набор знаков и сигналов.

Информация характеризуется источниками ее возникновения, потребителями, средой распространения и средствами ее доставки.

Источники – это живые существа, документы на любых физических носителях информации. *Среда распространения* – это окружающее нас пространство и технические средства связи (коммуникаций). *Средства, обеспечивающие доступность информации* – это информационно-поисковые системы (ИПС) и их лингвистическое обеспечение. *Потребитель информации* – это живое существо, техническое устройство, в т.ч. имеющее какой-либо физический носитель информации.

Информатика – наука о законах и методах (технологиях) получения, измерения, накопления, хранения, переработки и передачи информации с помощью математических и технических средств.

В информатике выделяют два основных научных направления: теоретическая и прикладная информатика. Некоторые науки, взаимодействуя с информатикой, создают собственные “отраслевые информатики”, использующие соответствующие им информационные технологии. С 1980-х гг. появляются такие “**отраслевые информатики**”, как: историческая, социальная, правовая, экономическая и др.

В условиях все возрастающих объемов информации появилась необходимость использовать способы хранения, обработки и передачи информации (информационные технологии), удаленные от человека. **Формирование информационного пространства** предполагает широкомасштабную компьютеризацию процессов переработки информации во всех сферах деятельности

и активное использование телекоммуникационных систем информационного обмена.

Технологии, предназначенные для решения информационных задач с помощью различных методов и программно-технических средств, называют *информационными*.

Информационная технология – процесс, использующий совокупность средств и методов сбора, обработки и передачи первичной информации для получения информации нового качества о состоянии объекта, т.е. информационного продукта. **Информационный продукт** используется, в частности, для принятия решений.

Информационные технологии состоят из этапов, каждый из них включает операции, а последние состоят из элементарных действий, таких, как нажатие какой-нибудь клавиши, выбор позиции в меню и т.д.

Выделяют следующие виды данных, обрабатываемых при помощи ИТ: текстовые, табличные, графические, звуковые, видео и мультимедийные данные. В информационных технологиях широкое распространение получили интегрированные офисные программы, включающие: текстовые и табличные процессоры; СУБД и др.

Существует разница между понятиями “информационная система” и “информационная технология”. *Информационная технология* является процессом, состоящим из четко регламентированных операций по преобразованию информации (сбор данных, их регистрация, передача, хранение, обработка, использование). Компьютерная *информационная система* является человеко-машинной системой обработки информации с целью организации, хранения и передачи информации. Например, технология, работающая с текстовым редактором, не является информационной системой.

Программные средства информационных систем можно разделить на 3 категории:

- базовое программное обеспечение;
- инструментальные средства;
- прикладные средства.

К *базовому программному обеспечению* относятся локальные и сетевые операционные системы (Windows, Unix, Solaris, OS/2, Linux и др.), а также сервисные средства для расширения возможностей операционных систем, обеспечения работоспособности компьютеров и выполнения процедур обслуживания автоматизированных информационных систем (АИС). К сервисным средствам относятся файловые менеджеры, антивирусные программы, архиваторы файлов, утилиты для тестирования компьютеров и сетей и т.п.

Инструментальные средства предназначены для изготовления новых программ. Это алгоритмические языки, переводчики (трансляторы: интерпретаторы и компиляторы) к ним.

Прикладные программные средства включают:

Программные средства (ПС) общего назначения (офисные, коммуникаций, мультимедиа, издательские системы, переводчики и пр.).

Методо-ориентированные ПС (статистика, бухгалтерский учет, планирование, управление проектами, экспертные системы, искусственный интеллект и др.).

Проблемно-ориентированные ПС (локальные, комплексные, интегрированные, корпоративные и др.).

2. Операционные системы

Операционная система (ОС) – это комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами вычислительной системы и прикладными программами, а с другой стороны – предназначены для управления устройствами, управления вычислительными процессами, эффективного распределения вычислительных ресурсов между вычислительными процессами и организации надежных вычислений. Это определение применимо к большинству современных операционных систем общего назначения.

В большинстве вычислительных систем ОС является основной, наиболее важной (а иногда и единственной) частью системного программного обеспечения. С 1990-х гг. наиболее распространенными операционными системами являются системы семейства Windows и системы класса UNIX (особенно Linux и Mac OS).

Основные функции ОС:

- Исполнение запросов программ (ввод и вывод данных, запуск и остановка других программ, выделение и освобождение дополнительной памяти и др.).
- Загрузка программ в оперативную память и их выполнение.
- Стандартизированный доступ к периферийным устройствам (устройства ввода-вывода).
- Управление оперативной памятью (распределение между процессами, организация виртуальной памяти).
- Управление доступом к данным на энергонезависимых носителях (таких, как жесткий диск, оптические диски и др.), организованным в той или иной файловой системе.
- Обеспечение пользовательского интерфейса.
- Сохранение информации об ошибках системы.
- Дополнительные функции:
- Параллельное или псевдопараллельное выполнение задач (многозадачность).

- Эффективное распределение ресурсов вычислительной системы между процессами.
- Разграничение доступа различных процессов к ресурсам.
- Организация надежных вычислений (невозможности одного вычислительного процесса намеренно или по ошибке повлиять на вычисления в другом процессе), основана на разграничении доступа к ресурсам.
- Взаимодействие между процессами: обмен данными, взаимная синхронизация.
- Защита самой системы, а также пользовательских данных и программ от действий пользователей (злонамеренных или по незнанию) или приложений.
- Многопользовательский режим работы и разграничение прав доступа.

3. Языки программирования

Первые языки (*языки машинных команд*) появились примерно в середине XX в. Они были достаточно примитивными и представляли собой упорядоченные последовательности единиц и нулей, понятные компьютеру. Программист должен был знать числовые коды всех машинных команд, сам распределять память под команды, программы и данные. Чтобы облегчить общение человека с электронно-вычислительной машиной (ЭВМ, компьютером) были созданы языки программирования типа *Ассемблер*. Переменные величины стали изображаться символическими именами. Числовые коды операций заменились на мнемонические обозначения, которые легче запомнить. Язык программирования приблизился к человеческому, и отделился от языка машинных команд.

Один из первых языков программирования – Фортран (**Formula Translation**) был создан в середине 50-х гг. Благодаря своей простоте и тому, что на этом языке накоплены большие библиотеки программ, Фортран достаточно распространен и используется для инженерных и научных расчетов, решения задач физики и других наук с развитым математическим аппаратом.

В 1968 г. был объявлен конкурс на лучший язык программирования для обучения студентов. Победитель Алгол-68, не получил широкого распространения, но был создан прецедент для развития структурного программирования и создания учебных языков. Так, Никлаус Вирт создал язык Паскаль – простой, удобный, с наличием мощных средств структурирования данных. Паскаль получил широкое развитие и в настоящее время считается одним из самых используемых (Turbo, Free, PascalABC). Для обучения младших школьников

Сеймур Пайперт разработал язык Лого. В школах долгое время использовался Бейсик в режиме непосредственного диалога. И сегодня он – самый простой для освоения (Dark, Small Basic).

Необходимость разработки больших программ, управляющих работой ЭВМ, потребовала создания специального языка программирования Си в начале 70-х гг. В нем заложены возможности непосредственного обращения к некоторым машинным командам и участкам памяти. Используется как инструментальный язык для разработки операционных систем, трансляторов, БД и других системных и прикладных программ. Язык известен своей эффективностью, экономичностью, и переносимостью. Часто программы, написанные на Си, сравнимы по скорости с программами, написанными на языке Ассемблера. При этом они имеют лучшую наглядность.

Существуют различные *классификации языков программирования*. По наиболее распространенной классификации все языки программирования делят на языки низкого, высокого и сверхвысокого уровня.

Низкого уровня: машинные языки и языки символического кодирования: (Автокод, Ассемблер). Операторы этого языка – это те же машинные команды, но записанные мнемоническими кодами, а в качестве операндов используются не конкретные адреса, а символические имена. Ориентированы на определенный тип компьютера, т. е. машинно-зависимы.

Высокого уровня: Фортран, Алгол, Кобол, Паскаль, Бейсик, Си, Пролог и т.д. Машинно-независимы, т.к. они ориентированы не на систему команд той или иной ЭВМ, а на систему операндов, характерных для записи определенного класса алгоритмов. Однако программы, занимают больше памяти и медленнее выполняются, чем программы на машинных языках. К языкам сверхвысокого уровня можно отнести лишь Алгол-68 и APL. Повышение уровня этих языков произошло за счет введения сверхмощных операций и операторов.

Другая классификация делит языки на вычислительные и языки символьной обработки. К первому типу относят Фортран, Паскаль, Алгол, Бейсик, Си, ко второму типу – Лисп, Пролог, Снобол и др.

Можно выделить два основных направления развития языков программирования: *процедурное* и *непроцедурное*. Процедурное возникло на заре вычислительной техники и получило широкое распространение. Программа явно описывает действия, которые необходимо выполнить, а результат задается только способом получения его при помощи некоторой процедуры (опр. последовательности действий).

Среди процедурных выделяют *структурные* и *операционные языки*. В структурных языках одним оператором записываются целые алгоритмические структуры: ветвления, циклы и т.д. В операционных языках для этого используются несколько операций. Широко распространены следующие

структурные языки: Паскаль, Си, Ада, ПЛ/1. Среди операционных известны Фортран, Бейсик, Фокал.

Непроцедурное (декларативное) программирование появилось в начале 70-х гг., но его развитие началось в 80-е гг., когда был разработан японский проект создания ЭВМ пятого поколения, цель которого – создания интеллектуальных машин. К непроцедурному программированию относятся *функциональные* и *логические* языки. В функциональных программах описывается вычисление некоторой функции. Обычно эта функция задается как композиция других, более простых, те в свою очередь разлагаются на еще более простые и т.д. Один из основных элементов в функциональных языках – рекурсия. Присваивания и циклов в классических функциональных языках нет.

В логических языках программа вообще не описывает действий. Она задает данные и соотношения между ними. После этого системе можно задавать вопросы. Машина перебирает известные и заданные в программе данные и находит ответ на вопрос. Порядок перебора не описывается в программе, а неявно задается самим языком. Классическим языком логического программирования считается Пролог. Построение логической программы вообще не требует алгоритмического мышления, программа описывает статические отношения объектов, а динамика находится в механизме перебора и скрыта от программиста.

Можно выделить еще один класс языков – *объектно-ориентированные* языки высокого уровня. На таких языках не описывают подробной последовательности действий для решения задачи, хотя они содержат элементы процедурного программирования. Объектно-ориентированные языки, благодаря богатому пользовательскому интерфейсу, предлагают человеку решить задачу в удобной для него форме. Пример – VBA, Object Pascal.

Языки описания сценариев (Perl, Python, Rexx) предполагают стиль программирования, отличный от характерного для языков системного уровня. Они предназначены не для написания приложения с нуля, а для комбинирования компонентов, набор которых создается заранее при помощи других языков. Развитие Internet способствует распространению языков описания сценариев. Так, для написания сценариев широко употребляется язык Perl, а среди разработчиков Web-страниц популярен JavaScript.

4. Технологии программирования

С развитием языков программирования развивались и технологии, используемые при написании программного кода. Первые программы писались сплошным текстом. Единственным средством, доступным программисту для создания логики были *условные переходы*. Следующий шаг – *процедурный*

подход: фрагменты программного кода объединяются в отдельные блоки (подпрограммы). Появилась возможность неоднократно использовать одну и ту же подпрограмму (процедуру или функцию) в одной программе.

Далее акцент при разработке программ сместился от проектирования процедур в сторону организации данных. Набор связанных процедур вместе с данными, которые они обрабатывают, часто называют *модулем*. Появилось *модульное программирование*.

Следующий шаг *объектно-ориентированное программирование* (ООП). Здесь программист оперирует не просто процедурами и функциями, а целыми объектами. *Объект* – это совокупность свойств, методов, событий.

Например, кнопка (окна приложения) обладает: свойствами (цвет, текст, шрифт текста...); событиями (нажатие на кнопку); методами – процедуры или функции, которые будут выполняться при наступлении определенного события).

Таким образом кнопку можно рассматривать как автономный объект, способный работать в различных условиях.

Укажем основные понятия ООП. *Абстрагирование* – способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. *Абстракция* – это набор всех таких характеристик. *Инкапсуляция* – это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя.

Класс – это модель объекта, он описывает устройство объекта. Говорят, что объект – это экземпляр класса. Класс также может представляться некоторым объектом при выполнении программы. Обычно классы разрабатывают таким образом, чтобы их объекты соответствовали объектам предметной области.

Наследование – это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским. Новый класс — потомком, наследником или производным классом.

Объект – сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса.

Полиморфизм – это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

Глава 2

ОСНОВНЫЕ ПРОГРАММНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

1. Текстовые редакторы и издательские системы

Каждый работающий на компьютере время от времени использует его для работы с текстом. Рассмотрим подробнее режимы работы текстовых редакторов.

Ввод и редактирование текста – это основной режим работы ТР, причем редактирование понимается как любое изменение в набранном тексте.

Под *форматированием* текста понимается изменение его внешнего вида с целью создания более эффективного и привлекательного документа. Различают форматирование *символов*, форматирование *абзацев*, форматирование *страниц* разделов.

Для *работы с фрагментами* можно использовать буфер обмена. Начиная с Word 2000 он имеет расширенные возможности и позволяет хранить до 24 фрагментов

Поиск и замена текста.

Работа с файлами заключается в создании, сохранении и открытии файла.

Широкие возможности текстовых редакторов позволили компьютеру вытеснить пишущие машинки из делопроизводства, а использование компьютерных издательских систем во многом изменило организацию подготовки рукописи к изданию, автоматизировало труд людей нескольких типографских профессий – верстальщика, наборщика, корректора и др.

В издательских технологиях обработки текстов выделяют художественное и техническое редактирование.

Вот некоторые понятия технического редактирования.

Шрифты – основное изобразительное средство издательских систем. Шрифты различают по *гарнитуре* (рисунку), *начертанию*, *кеглю* (размеру).

Гарнитурой называется совокупность шрифтов одного рисунка во всех начертаниях и кеглях. Полный комплект гарнитуры содержит шрифты всех начертаний и кеглей, а в каждом кегле – русский и латинский (и, если надо, другие) алфавиты прописных и строчных букв, а также относящиеся к ним знаки.

Буквы располагаются по *базовой линии*. Расстояние между строками называют *интерлиньяжем*.

Размер шрифта, определенный как расстояние между нижним и верхним выносными элементами, называют *кеглем* и измеряют в пунктах. Один пункт равен 0,376 мм. Шрифты могут быть прямыми и наклонными. Наклонный вариант шрифтов часто называют *курсивом*.

Шрифт на компьютере – это файл или группа файлов, обеспечивающих вывод текста на печать со стиливыми особенностями шрифта.

Таблица представляет собой сложную наборную форму, в которую могут входить тексты, цифры, графический материал, формулы или любое сочетание этих элементов.

2. Основы работы в Word

Word – наиболее популярный текстовый процессор, позволяющий создавать профессионально оформленные документы. При запуске Word автоматически создается новый документ (его рабочее название Документ1), готовый для ввода текста.

Если вы не знаете, какие операции применимы к выделенному объекту в документе, то вызовите *контекстное меню* (щелкнув на объекте правой клавишей мыши). В него будут включены основные команды, предназначенные для работы с выделенным объектом.

Начиная с версии 2000, в Word для каждого открытого документа создается отдельное окно и выводится кнопка на панели задач Windows. Такой подход обеспечивает более быстрое переключение между документами.

Word позволяет отображать документ в режиме, который в наибольшей степени подходит для выполняемых операций. Кнопки для выбора основного режима просмотра находятся в правой части строки состояния. Также управлять режимами отображения документа можно на вкладке **Вид**.

Режим разметки страницы позволяет увидеть страницу в точности так, как она будет напечатана, включая колонки, колонтитулы и номера страниц. Можно прокрутить страницу за пределы основного текста и просмотреть колонтитулы, номера страниц и размеры полей. В режиме просмотра разметки страницы можно производить все операции по вводу и редактированию текста.

Режим чтения используется для чтения текста документа в удобном для пользователя масштабе.

Веб-документ – режим просмотра электронного документа позволяет просмотреть документ в том виде, в каком он будет отображен в браузере. В этом режиме слева от текста появляется схема документа, которая позволяет быстро переходить к любому заголовку документа.

Режим структуры документа используется для просмотра документа в соответствии со структурой заголовков. В режиме структуры отображаются структурные уровни абзацев документа и открывается специальная вкладка **Структура**, кнопки которой позволяют повышать и понижать уровни абзацев, а также передвигать их вверх или вниз.

Черновик применяется для отображения основного текста в том виде, в каком он будет напечатан. В этом режиме форматирование символов и абзацев отображается полностью, разрывы строк и страниц, позиции табуляции и выравнивание также отображаются без искажений. В то же время область вне основного текста, т.е. поля, колонтитулы и номера страниц не отображаются. За счет этого операции редактирования и перемещения точки вставки выполняются несколько быстрее.

Перемещение по документу

Вертикальная и горизонтальная полосы (линейки) прокрутки позволяют с помощью мыши прокручивать документ. Бегунок полосы прокрутки показывает место отображаемой в данный момент части документа относительно полной длины документа. При перемещении бегунка по вертикальной полосе прокрутки рядом с ним появляется номер страницы.

Чтобы переместить точку вставки с помощью мыши, прокрутите документ так, чтобы увидеть нужное место; переместите туда указатель мыши и щелкните им.

Клавиши со стрелками, а также другие клавиши управления курсором – <Page Up>, <Page Down>, <Home>, <End> перемещают точку вставки обычным образом.

Эти же клавиши, но в комбинации с клавишей <Ctrl>, позволяют выполнять и другие операции перемещения.

Таблица 1 – Некоторые приемы перемещения с помощью клавиатуры

Перемещение	Комбинация клавиш
На одно слово влево	<i>Ctrl</i> + ←
На одно слово вправо	<i>Ctrl</i> + →
В конец строки	<i>End</i>
В начало строки	<i>Home</i>
В конец документа	<i>Ctrl</i> + <i>Home</i>
В начало документа	<i>Ctrl</i> + <i>End</i>

Выбор объектов перехода

Важным средством Word является Выбор объектов, которое позволяет перемещаться по документу от одного объекта к другому. Чтобы воспользоваться средством **Выбор объектов**:

Нажмите кнопку Выбор объекта перехода на вертикальной полосе прокрутки. Появится палитра кнопки **Выбор объекта перехода**.

Нажмите кнопку, определяющую тип просматриваемых, объектов.

Нажмите одну из кнопок с двойными стрелками, расположенными рядом с кнопкой **Выбор объекта перехода**, чтобы перейти к предыдущему или следующему объекту выбранного типа.

Чтобы отменить режим просмотра объектов и вернуться к обычному режиму листания страниц, нажмите кнопку **Страницы** в меню объектов перехода.

После выбора типа просматриваемых объектов кнопки на вертикальной полосе прокрутки, расположенные выше и ниже кнопки **Выбор объекта перехода**, изменят свой цвет и название. Из черных стрелочки на кнопках станут синими, что служит признаком того, что установлен режим просмотра объектов определенного типа, а в названии кнопок появится указание типа объектов, для которых установлен режим просмотра.

Стрелочки останутся черными, если нажать кнопку **Страницы**, потому что переход к следующей и предыдущей страницам является функцией по умолчанию для этих кнопок.

Если нажать кнопку **Перейти** или **Найти**, то Word откроет соответствующую вкладку диалогового окна **Найти и заменить**.

3. Форматирование документов Word

Минимальным формируемым фрагментом текста является **символ**.

Символ – это отдельная буква, цифра, знак пунктуации или специальный знак.

Под форматом символа понимаются параметры его начертания – используемый шрифт и его атрибуты: размер, начертание, цвет, регистр, заливка, анимация и др. Все эти параметры можно назначить как отдельному символу, так и всему документу.

Параметры форматирования можно комбинировать.

Шрифт (гарнитура) – это способ начертания символов. Все буквы, цифры и проч. Знаки, относящиеся к данному шрифту, имеют характерный для него вид.

Размер (кегель) – измеряется в пунктах (1пт = 0,375 см).

Стиль – набор “запомненных” команд форматирования.

Абзац – это фрагмент текста, возможно пустой, который заканчивается символом (маркером) **конца абзаца** ¶. Этот символ вводится нажатием клавиши <Enter>. Для удобства символ конца абзаца обычно не отображается на экране. Символ конца абзаца хранит в себе форматирование абзаца.

В быту абзац – это последовательность предложений, выражающих отдельную мысль. В Word абзацами являются также: заголовок статьи, пункт списка, пустая строка между абзацами.

Для абзацев существуют свои параметры форматирования:

- Выравнивание.
- Отступы.
- Позиции табуляции.
- Интервалы.
- Линии, рамки и заливка.

Списки. В общем случае список представляет собой набор абзацев – элементов списка, отформатированных особым образом и снабженных номерами или специальными маркерами, Word позволяет максимально автоматизировать процесс создания списков различных видов. Возможно использование списков трех типов: маркированные, нумерованные и многоуровневые.

Маркированные списки. В таких списках перед каждым элементом ставится специальный символ или рисунок, отмечающий элемент списка (маркер). Для всех элементов одного списка используются одинаковые маркеры. Маркированные списки обычно применяются для перечисления параметров, порядок следования которых не важен (например, список свойств какого-либо продукта).

Нумерованные списки. В отличие от маркированных списков здесь вместо маркеров используется последовательность чисел или символов. Каждый элемент такого списка имеет свой индивидуальный номер, что удобно при перечислении свойств или операций, порядок следования которых важен (например, список пошаговых процедур).

Многоуровневые списки. Этот список представляет собой иерархическую структуру из списков вышеописанных типов. Подобные структуры используются при создании разнообразных технических или юридических документов, требующих многоуровневой нумерации элементов.

Word позволяет распространять *установки форматирования страницы* на отдельные части документа, которые называются **разделами**. Установки форматирования страниц включают такие параметры, как размеры полей, вертикальное выравнивание на странице, разрывы страниц, колонтитулы, формат и ориентацию бумаги.

Оглавлением называется список заголовков данного документа, перечисленных в том порядке, в котором они встречаются в документе. Как правило, в оглавление включаются все заголовки определенного уровня. Например, можно указать в оглавлении все главы и разделы документа, но не указывать подразделы. Для каждого заголовка указывается фактический номер страницы, на которой он находится. Вставка оглавления возможна, если ваш документ обладает структурой, и в нем использовались встроенные заголовки Word.

Печать документов

Напечатанный документ традиционно считается окончательным результатом работы в текстовом процессоре. Помимо основных средств печати документов, Word предоставляет много дополнительных возможностей: просмотр перед печатью одной или нескольких страниц документа и изменение параметров страницы прямо в процессе просмотра; печать всего открытого документа или только его части; печать документа в виде черновика или в окончательном варианте. Можно также печатать скрытый текст и коды полей или сразу несколько документов, не открывая их.

Все эти дополнительные возможности можно реализовать в диалоговом окне **Печать**. Например, иногда возникает необходимость изменить масштаб печатаемого материала, чтобы напечатать несколько страниц документа на одном листе для экономии бумаги.

4. Таблицы в Word

Word обладает богатым арсеналом возможностей для работы с таблицами. Он предоставляет широкий набор средств, облегчающих и ускоряющих выполнение многих стандартных операций с таблицами, таких, как создание, редактирование, форматирование и т. д.

Таблица состоит из строк и столбцов ячеек. Ячейки могут содержать как текстовые, так и графические данные (например, иллюстрации к тексту). Наиболее удобно использовать таблицы для представления хорошо структурированных данных, например списков, расписаний, финансовой информации.

Таблицы позволяют легко и быстро форматировать содержащиеся в них данные, сортировать их и даже производить некоторые вычисления. С помощью таблиц можно разместить текст в несколько колонок, изменить направление текста (например, расположив его по вертикали), выровнять абзацы текста и рисунки.

Таблица может быть вставлена в любое место документа. Она может быть достаточно длинной, в том числе размещаться на нескольких страницах, в этом случае имеется возможность повторять заголовки таблицы на каждой странице (вкладка **Макет, Данные**). Структура таблицы в общем случае может быть сколь угодно сложной. Совершенно необязательно, чтобы таблица представляла собой прямоугольную сетку из одинаковых ячеек.

Для вставки таблицы в документ используют кнопку **Таблица**, расположенную на вкладке **Вставка** и выбирают соответствующее число строк и столбцов.

Иногда возникает потребность представить существующий текст в табличной форме. Вместо того чтобы переносить имеющийся текст в таблицу

вручную, лучше выделить текст и воспользоваться командой **Преобразовать в таблицу** из подменю кнопки **Таблица**. Такой текст должен быть предварительно подготовлен к преобразованию. Для этого на месте будущих границ ячеек нужно установить символ-разделитель, например, нажимая клавишу Tab.

Word также позволяет выполнить и обратную операцию – преобразование таблицы в текст (вкладка **Макет, Данные**). При этом тоже потребуется указать символ-разделитель, который будет отделять информацию из разных столбцов таблицы.

5. Графические объекты в Word

Word предоставляет богатые возможности использования графических объектов. Вот некоторые из них:

- 1) вставка готовых рисунков и клипов;
- 2) создание рисунков SmartArt;
- 3) вставка отдельных фигур и рисование с помощью инструментов Word;
- 4) создание текстовых эффектов (объектов Word Art);
- 5) построение диаграмм;
- 6) создание надписей;
- 7) вставка формул.

В документ Word можно вставлять оцифрованные с помощью сканера фотографии, картинки, созданные различными графическими программами или взятые из готовых библиотек.

Документ Word имеет три слоя, в каждый из которых можно поместить текст или рисунок. Слой текста – тот, в котором появляется основной текст документа. Кроме того, имеется еще два специфических слоя: слой под текстом и слой над текстом. Рисованные объекты можно помещать в любой из этих слоев.

При вставке в текст документа рисунка, последний *привязывается* к одному из абзацев (ближайшему к точке вставки) и при изменении текста перемещается вместе с абзацем. Чтобы рисунок всегда оставался на своем месте необходимо отменить привязку. С другой стороны, при перемещении рисунка, точка привязки перемещается к ближайшему абзацу. Чтобы рисунок размещался на одной странице с текстом следует заблокировать точку привязки. (**Формат рисунка** вкладка **Положение**, кнопка **Дополнительно**).

Одним из типов объектов для вставки в документ Word является диаграмма. Можно создать диаграмму, отражающую данные некоторой таблицы.

Word предоставляет возможность создавать рисунки различной степени сложности: от элементарных геометрических фигур до сложных объемных фигур. Для создания рисунков используют кнопку **Фигуры** на вкладке **Вставка**.

Для размещения рисованных объектов можно использовать специальную область рисования – **Полотно**, с помощью которого в документе отводится место под рисунок.

Полотно вставляется в слой текста, в то место, где находился курсор (точка вставки) при выборе пункта **Новое полотно** из подменю кнопки **Фигуры**.

Полотно можно форматировать, перемещать в различные слои документа, назначать параметры обтекания текстом. При работе с **Полотном** доступны некоторые дополнительные операции.

Объекты WordArt. При создании документов часто возникает необходимость привлечь внимание читателя к каким-либо разделам, выделить важные моменты. Можно, например, эффектно оформить заголовок статьи. В состав Microsoft Office входит программа WordArt, с помощью которой текст документа можно преобразовать в графический образ. Вставить объекты WordArt можно с помощью соответствующей кнопки на вкладке **Вставка**.

Объекты WordArt, вставленные в документ, можно впоследствии изменять средствами программы WordArt, используя соответствующую вкладку на ленте. Можно установить способ обтекания текста, форматировать объект WordArt, изменять форму текста, вращать объект и др. В целом работа с объектами WordArt не отличается от работы с другими графическими объектами.

Использование **надписей** является наиболее эффективным средством при компьютерной верстке. Текст, заключенный в надпись, можно свободно перемещать по странице и форматировать независимо от основного текста. Можно вставлять в надпись различные типы объектов, включая таблицы, рисунки, диаграммы, фигурный текст, формулы и др.

При изменении размера надписи текст в ней автоматически разбивается на строки, так, чтобы соответствовать размерам надписи. При добавлении текста в надпись размер надписи не изменяется, поэтому часть текста может оказаться скрытой.

Надпись можно перемещать по странице документа. Как и другие объекты, надпись привязывается к абзацу, первоначально к тому, возле которого была создана.

Форматирование текста надписи аналогично подобным операциям для основного текста документа. Однако существуют два специальных приема: поворот текста (**Направление текста**) и изменение просвета между текстом и границей надписи.

Если при подготовке документа необходимо получить свободу в перемещении блоков текста на странице, изменении размера и формы текста, то в этих целях удобно использовать надписи.

6. Компьютерная графика и графические редакторы

Различают четыре вида компьютерной графики: растровая, векторная, трехмерная и фрактальная графика. Они отличаются принципами формирования изображения при отображении на экране монитора или при печати на бумаге.

Растровую графику применяют при разработке электронных (мультимедийных) и полиграфических изданий. Большинство растровых редакторов ориентированы не столько на создание изображений, сколько на их обработку. Для этого используют отсканированные иллюстрации, подготовленные художником на бумаге, или фотографии. В Интернете применяют растровые иллюстрации в тех случаях, когда надо передать полную гамму оттенков цветного изображения.

Программные средства для работы с **векторной графикой** наоборот предназначены, в первую очередь, для создания иллюстраций и в меньшей степени для их обработки. Такие средства широко используют в рекламных агентствах, дизайнерских бюро, редакциях и издательствах.

Трехмерная графика широко используется в инженерном программировании, компьютерном моделировании физических объектов и процессов, в мультипликации, кинематографии и компьютерных играх.

Программные средства для работы с **фрактальной графикой** предназначены для автоматической генерации изображений путем математических расчетов. Создание фрактальной художественной композиции состоит не в рисовании или оформлении, а в программировании. **Фрактал** – это рисунок, который состоит из подобных между собой элементов. Построение фрактального рисунка осуществляется по какому-то алгоритму или путем автоматической генерации изображений при помощи вычислений по конкретным формулам. Изменения значений в алгоритмах или коэффициентов в формулах приводит к модификации этих изображений. Главное преимущество фрактальной графики – в файле фрактального изображения сохраняются только алгоритмы и формулы. Фрактальную графику часто используют в развлекательных программах.

Для работы с графикой используют специальные программы – **ГР**. Растровые ГР: Paint, Photoshop. Векторный: CorelDraw.

Основным (наименьшим) **элементом растрового изображения** является точка (пиксел). Каждый пиксел растрового изображения имеет свойства: размещение и цвет. Чем больше количество пикселей и чем меньше их раз-

меры, тем лучше выглядит изображение. Большие объемы данных – основная проблема при использовании растровых изображений. Второй недостаток растровых изображений связан с невозможностью их увеличения для рассмотрения деталей. Поскольку изображение состоит из точек, то увеличение изображения приводит только к тому, что эти точки становятся крупнее и напоминают мозаику.

Photoshop. Назначение – создание фотореалистических изображений, работа с цветными сканированными изображениями, ретуширование, цветокоррекция, коллажирование, трансформации, цветоделение и др. Позволяет легко изменять цветовое представление документов (в градациях серого, черно-белого, RGB или CMYK и др.). Photoshop – это программа растровой графики, есть элементы векторной графики. Все инструменты художника!

Слои позволяют работать с одним элементом рисунка, не затрагивая другие элементы. Слои можно представить, как стопку прозрачных пленок, на которых можно рисовать, и если на пленке нет изображения, то можно видеть сквозь нее низлежащие. Можно легко изменить рисунок, меняя порядок и свойства слоев.

Название активного слоя отображено в шапке рисунка, а в палитре слоев напротив активного слоя изображена кисточка. Выбрать слой для работы можно в палитре слоев, кликнув на его изображение.

Когда создается новый слой, он появляется над выбранным слоем в палитре слоев.

Можно выборочно скрывать и показывать содержимое слоев.

Можно полностью или частично запереть слои для защиты их содержимого (если работа над ним закончена). Или запереть изменение содержимого, при этом оставив возможность двигать слой.

В векторной графике основным элементом изображения является линия (при этом не важно, прямая это линия или кривая).

Как и все объекты, линии имеют свойства. К этим свойствам относятся: форма линии, ее толщина, цвет, характер линии (сплошная, пунктирная и т.п.). Замкнутые линии имеют свойство заполнения. Внутренняя область замкнутого контура может быть заполнена цветом, текстурой, картой. Простейшая линия, если она не замкнута, имеет две вершины, которые называются узлами. Узлы тоже имеют свойства, от которых зависит, как выглядит вершина линии и как две линии сопрягаются между собой.

Линия – это элементарный объект векторной графики. Все, что есть в векторной иллюстрации, состоит из линий. Например объект четырехугольник можно рассматривать как четыре связанные линии. Из-за такого подхода векторную графику часто называют объектно-ориентированной графикой. Объекты векторной графики хранятся в памяти в виде набора параметров, но

на экран все изображения все равно выводятся в виде точек (просто потому, что экран так устроен). Перед выводом на экран каждого объекта программа производит вычисления координат экранных точек в изображении объекта, поэтому векторную графику еще называют вычисляемой графикой. Аналогичные вычисления производятся и при выводе объектов на принтер.

7. Основные понятия компьютерной графики

Разрешение экрана – это свойство компьютерной системы (зависит от монитора и видеокарты) и операционной системы (зависит от настроек Windows). Измеряется в пикселах (точках) и определяет размер изображения, которое может поместиться на экране целиком.

Разрешение принтера – это свойство принтера, выражающее количество отдельных точек, которые могут быть напечатаны на участке единичной длины. Оно измеряется в единицах dpi (точки на дюйм) и определяет размер изображения при заданном качестве или, наоборот, качество изображения при заданном размере.

Разрешение изображения – это свойство самого изображения. Оно тоже измеряется в точках на дюйм – dpi и задается при создании изображения в графическом редакторе или с помощью сканера. Так, для просмотра изображения на экране достаточно, чтобы оно имело разрешение 72 dpi, а для печати на принтере – не меньше как 300 dpi. Значение разрешения изображения хранится в файле изображения.

Физический размер изображения определяет размер рисунка по вертикали (высота) и горизонтали (ширина) может измеряться как в пикселах, так и в единицах длины (миллиметрах, сантиметрах, дюймах). Он задается при создании изображения и хранится вместе с файлом. Если изображение готовят для демонстрации на экране, то его ширину и высоту задают в пикселах, чтобы знать, какую часть экрана оно занимает.

Если изображение готовят для печати, то его размер задают в единицах длины, чтобы знать, какую часть листа бумаги оно займет.

При работе с цветом используются понятия: глубина цвета (его еще называют цветовое разрешение) и цветовая модель.

Для кодирования цвета пиксела изображения может быть выделено разное количество бит. От этого зависит то, сколько цветов на экране может отображаться одновременно. Чем больше длина двоичного кода цвета, тем больше цветов можно использовать в рисунке. **Глубина цвета** – это количество бит, которое используют для кодирования цвета одного пиксела. Для кодирования двухцветного (черно-белого) изображения достаточно выделить

по одному биту на представление цвета каждого пиксела. Выделение одного байта позволяет закодировать 256 различных цветовых оттенков. Два байта (16 битов) позволяют определить 65536 различных цветов. Этот режим называется High Color. Если для кодирования цвета используются три байта (24 бита), возможно одновременное отображение 16,5 млн цветов (True Color). От глубины цвета зависит размер файла изображения.

Под **цветовым охватом** следует понимать диапазон цветов, который может быть воспроизведен, зафиксирован или описан каким-либо образом. Цветовой охват у разных устройств различен. Каждый из охватов может быть выражен моделью цвета.

С помощью **цветовой модели** можно дать математическое описание цвета. Составной частью модели является **канал**. С помощью задания различных значений совокупности каналов описывается тот или иной цвет.

Наиболее известны следующие модели: RGB, CMYK, Lab, HSB.

Модель RGB – трехканальная модель, каждому каналу которой соответствует свой цвет – Красный (R), Зеленый (G), Синий (B). Концепция этой модели сводится к тому, что цвета каналов являются базовыми, а все остальные цвета получаются путем смешения в той или иной пропорции двух основных цветов. Полученные цвета называются аддитивными. Результирующие цвета получаются светлее базовых. Модель нашла применение в таких устройствах как сканер и экран монитора.

Модель CMY также трехканальная, но каналы соответствуют вторичным цветам – голубому (C), желтому (Y) и пурпурному (M). Начальные буквы аббревиатуры соответствуют английским названиям перечисленных цветов: Cyan, Yellow, Magenta. Эти цвета называются вторичными или отраженными. Получены путем вычитания первичного цвета RGB из белого. Белый минус красный создает голубой цвет. Белый – зеленый = пурпурный. Белый – синий = желтый. Концепция также основана на смешении цветов, но уже не первичных, а вторичных. Полученные цвета (красный, зеленый, синий) в результате парного смешения отраженных цветов называют субтрактивными.

На практике более широкое применение нашла **модель CMYK**. Она описывает реальный процесс цветной печати на офсетной машине и цветном принтере. В области черного и темных цветов наносятся не цветные, а черная краска. Это четвертый базовый канал, он введен для описания реального процесса печати. Буква K в аббревиатуре соответствует последней букве из английского слова black (черный).

Модели RGB и CMYK являются аппаратно-зависимыми. То есть при воспроизведении одного и того же изображения на разных устройствах цветопередача будет отличаться. Даже, если это однотипные устройства (например, мониторы разных марок). Цвет в таких моделях зависит от параметров устройств воспроизведения: свойств бумаги, на которой осуществляется печать, качества

краски, свойств люминофора и др. часто приходится иметь дело одновременно с двумя моделями. Так для отображения на экране монитора оптимальной является модель RGB, а для офсетной печати – CMYK. В таком случае приходится применять системы коррекции цвета – специальные программные технологии, которые преобразуют изображение из одной модели в другую.

Однако существуют цветовые модели, которые не требуют цветокоррекции. Одной из таких моделей является аппаратно-независимая **модель Lab**. Концепция этой модели сводится к тому, что любой цвет описывается с помощью трех каналов: L – светлота (Lightness, яркость), а – изменение цвета в диапазоне от зеленого до красного, b – изменение цвета от синего до желтого. Яркость в модели Lab полностью отделена от цвета, что делает модель удобной для регулировки основных параметров изображения – контрастности, тона и др. Lab имеет наибольший охват из перечисленных цветовых моделей, т.к. включает в себя их охваты.

Не менее популярной является и **модель HSB**, название которой происходит от сокращения английских слов Hue (Тон), Saturation (Насыщенность) и Brightness (Яркость). Концепция этой модели сводится к тому, что задавая значения трем каналам, которые соответствуют тону (собственно цветом), насыщенности (процент содержания белого цвета) и яркости (процент содержания черной краски) краски, можно получить тот или иной цвет или оттенок.

8. Электронные таблицы Excel

Электронная таблица – это средство информационных технологий, позволяющее решать целый комплекс задач:

1. Выполнение вычислений в области делопроизводства: расчетные ведомости, сметы расходов и т.п. Решение ряда математических задач численными методами – в табличной форме. Решения многих вычислительных задач на ЭВМ, которые раньше можно было осуществить только путем программирования.
2. Математическое моделирование: использование математических формул позволяет представить взаимосвязь между различными параметрами некоторой реальной системы. ЭТ представляет собой удобный инструмент для организации численного эксперимента: подбор параметров, прогноз поведения моделируемой системы, анализ зависимостей, планирование. Дополнительные удобства для моделирования дает возможность графического представления данных.
3. Использование электронной таблицы в качестве базы данных. Конечно, по сравнению с СУБД электронные таблицы имеют меньшие возможности, но некоторые операции манипулирования данными в них реализованы. Это сортировка, поиск информации по заданным условиям.

Электронная таблица – компьютерный эквивалент обычной таблицы, в клетках (ячейках) которой записаны данные различных типов: тексты, даты, формулы, числа.

Главное достоинство электронной таблицы – это возможность мгновенного пересчета всех данных по формулам при изменении значения любого операнда.

Рабочая область электронной таблицы состоит из строк и столбцов, имеющих свои имена.

Имена строк – это их номера. Нумерация строк начинается с 1 и заканчивается максимальным числом, установленным для данной программы.

Имена столбцов — это буквы латинского алфавита сначала от А до Z, затем от AA до AZ, BA до BZ и т. д.

Ячейка – область, определяемая пересечением столбца и строки электронной таблицы. Ячейка является наименьшей структурной единицей для размещения данных внутри рабочего листа.

Каждая ячейка может содержать данные в виде текста, числовых значений, формул или параметров форматирования. При вводе данных Excel автоматически распознает тип данных и определяет перечень операций, которые могут с ними проводиться. По своему содержанию ячейки делятся на исходные (влияющие) и зависимые. В последних записаны формулы, которые имеют ссылки на другие ячейки таблицы. Ячейка, выбранная с помощью указателя, называется активной или текущей ячейкой.

Адрес ячейки предназначен для определения местонахождения ячейки в таблице.

9. Типы данных в ячейке

Все данные, которые обрабатывает Excel, делятся на следующие три типа: числа, формулы и текст.

К **числовому** типу данных относятся:

- числовая константа, например: 125; 125.5; 3.46E+03;
- *дата*, например: 32.12.97; 6 апр 97; дек 97;
- *время*, например: 14:30; 14:30:59; 1:30:55 PM.
- *дата и время*, например: 28 октябрь, 1995 23:45:36

Все числовые данные после ввода в ячейку по умолчанию выравниваются по правому краю.

Числа в электронных таблицах Excel могут быть записаны в обычном числовом или экспоненциальном формате, например: 195,2 или 1.952E + 02. По умолчанию числа выравниваются в ячейке по правому краю. Это объясняется тем, что при размещении чисел друг под другом (в столбце таблицы)

удобно иметь выравнивание по разрядам (единицы под единицами, десятки под десятками и т.д.).

Формула должна начинаться со знака равенства (=) и может включать в себя числа, имена ячеек, функции (Математические, Статистические, Финансовые, Дата и время и т.д.) и знаки математических: операций. Например, формула «=A1+\$B2» обеспечивает сложение чисел, хранящихся в ячейках A1 и B2, а формула «=A1*5» – умножение числа, хранящегося в ячейке A1, на 5. При вводе формулы в ячейке отображается не сама формула, а результат вычислений по этой формуле. Саму формулу можно увидеть в Строке формул. При изменении исходных значений, входящих в формулу, результат пересчитывается немедленно.

C1		fx =A1*B1	
	A	B	C
1	3	5	15

Любые данные, которые программой не распознаются в качестве числовых (константа, дата, время), интерпретируются как **текстовые**. *Текстом* в электронных таблицах Excel является последовательность символов, состоящая из букв, цифр и пробелов, например запись «32 Мбайт» является текстовой. По умолчанию текст выравнивается в ячейке по левому краю. Это объясняется традиционным способом письма (слева направо).

Особенности ввода данных

Данные любого типа, введенные в ячейку, могут быть представлены в определенном формате. Для этого используется команда контекстного меню **Формат ячеек**. Форматирование содержимого ячейки не приводит к его изменению, а определяет внешнее представление введенного значения (см. табл. 6.1). Ячейки можно форматировать как до, так и после ввода в них значений.

Таблица 6.1 – Представление данных в различных форматах

Представление	Формат
1	Общий
1,00	Числовой
1,00р.	Денежный
1,0р.	Финансовый
1.01.1900	Дата
0:00:00	Время
100,0%	Процент
1	Дробный
1.00E+00	Экспоненциальный
1	Текстовый
0001	Дополнительный

Вначале число «1» введено и ячейки в формате по умолчанию Общий, а затем каждая ячейка подвергалась форматированию по команде Формат ячеек.

Если при вводе данных в ячейки не был задан определенный числовой формат, то по умолчанию применяется формат Общий.

В формате Общий в ячейке отображается максимум 10 цифр. Если ширина ячейки недостаточна для отображения введенного числового значения, то Excel автоматически представляет (но не преобразует!) его в экспоненциальной форме.

При недостаточной ширине ячейки для вводимого числа, но с использованием числового формата, отличного от формата Общий, Excel представит вместо числа несколько символов номера, например: #####. В этом случае, чтобы отобразить содержимое ячейки, необходимо увеличить ширину столбца.

При вводе в ячейку значения даты для разделения ее частей используется наклонная черта (/), точка или дефис (—), например, 1.02.1998 или 1/02/98. Значения времени также можно ввести непосредственно с клавиатуры, отделяя друг от друга часы, минуты и секунды двоеточием. Если в ячейке одновременно должны быть указаны значения даты и времени, то при вводе их необходимо разделить пробелом, например:

1—ФЕВ—98 17:09

Excel представляет и обрабатывает значения даты и времени как десятичные числа. Каждой дате из диапазона 01.01.1900 по 31.12.2078 ставится в соответствие {десятичное} число от 1 до 65380. Так дате 01.02.1900 в формате Числовой (Number) соответствует целое число 32. Аналогично любому значению времени суток ставится в соответствии {десятичная} дробь из интервала от 0 до 1. Например, значение времени суток 23:45:36, представленное в числовом формате, есть десятичная дробь 0.99. Справедливо и обратное преобразование. Если десятичное число 32.99, введенное в ячейку с использованием формата Числовой, отформатировать с применением формата Дата, то оно будет выглядеть следующим образом: 01.02.1990 23:45:36. Благодаря такому преобразованию значения даты и времени используются при вычислениях. Если эти значения используются в формулах, то они должны быть записаны в кавычках, например, =«5/12/94» – «3/5/94».

Особенности ввода текстовых данных

Длина текста в ячейке не может превышать 255 символов. Для каждого символа в ячейке можно задать свои параметры форматирования. Если введенный текст полностью не помещается в текущую ячейку, то он будет отображен (но не помещен!) поверх соседней ячейки справа. Последняя в этом случае должна быть пустой – в противном случае, не уместяющийся текст будет обрезан по правому краю своей ячейки. Полный текст, введенный в ячейку, всегда виден в строке формул. Длинный текст можно разбить в ячейке на несколько строк нажатием комбинации клавиш <Alt> + <Enter>.

Ввод и использование формул

Формула представляет собой сочетание констант, операторов, ссылок, функций, имен диапазонов и круглых скобок, используемых для изменения последовательности вычислений. Пример записи формулы:

= СУММ (A1 : A15) / \$C\$3 + ЦЕНА * (150 + КОЛИЧ)

В этом примере СУММ имя функции; A1, A15, \$C\$3 ссылки; A1 : A15 диапазон (массив) ячеек; ЦЕНА, КОЛИЧ – имена диапазонов ячеек; 150 – константа. В формулах могут использоваться арифметические операторы +, —, *, /, % (определение значения процента); ^ (возведение в степень); операторы сравнения =, <, >, <=, >=, <> (не равно) и оператор объединения текстовых данных &. Кроме того, Excel распознает три адресных оператора для указания диапазонов ячеек:

1. Оператор диапазона (двоеточие). Например, по формуле =СУММ(B1:D1) вычисляется сумма содержимого ячеек B1, C1 и D1, входящих в состав диапазона B1:D1. Оператор диапазона можно использовать для указания ссылок на ячейки всего столбца (например, A:A), на ячейки всей строки (например, 1:1), на ячейки всего рабочего листа (A:XFD или 1:1048576).

2. Оператор объединения диапазонов (точка с запятой). Например, по формуле =СУММ(C5:C7; A3:C3) вычисляется сумма двух несмежных диапазонов C5:C7 и A3:C3.

3. Оператор пересечения диапазонов (пробел). Например, по формуле =СУММ(A5:B11 B8:C10) вычисляется сумма содержимого ячеек диапазона B8:B10.

Абсолютные и относительные ссылки

В формулах используются ссылки на адреса ячеек. Существуют два основных типа ссылок: **относительные** и **абсолютные**. Различия между ними проявляются при копировании формулы из активной ячейки в другую ячейку.

По умолчанию, ссылки на ячейки в формулах рассматриваются как *относительные*. Это означает, что при копировании формулы адреса в ссылках автоматически изменяются в соответствии с относительным расположением исходной ячейки и создаваемой копии.

Относительные ссылки имеют следующий вид: A1, B3.

Абсолютная ссылка в формуле используется для указания фиксированного адреса ячейки. При перемещении или копировании формулы абсолютные ссылки не изменяются. В абсолютных ссылках перед неизменяемым значением адреса ячейки ставится знак доллара (например, \$A\$1).

Если символ доллара стоит перед буквой (например: \$A1), то координата столбца абсолютная, а строки – относительная. Если символ доллара стоит перед числом (например, A\$1), то, наоборот, координата столбца относитель-

ная, а строки – абсолютная. Такие ссылки называются **смешанными**. Таким образом, *смешанные ссылки* – это ссылки, которые при копировании изменяются частично.

Пусть, например, в ячейке C1 записана формула $=A\$1+\$B1$, которая при копировании в ячейку D2 приобретает вид $=B\$1+\$B2$.

Относительные ссылки при копировании изменились, а абсолютные – нет.

Рассмотрим примеры копирования формул с разными типами ссылок.

	A	B	C	D	E
1	Наценка	30 %			
2					
3	Товары	Цена	Стоимость		
4	Тетрадь	1000	$=b4+b4*\$b\$1/100$		
5	Линейка	2000			
6	Ручка	900			
7	Карандаш	500			
8					

После ввода формулы в ячейку C4 отобразится результат – 1300

Заполним аналогичными формулами остальные ячейки столбца «Стоимость». Для этого можно активизировать ячейку с формулой (ячейку c4), установить указатель мыши на квадратике в правом нижнем углу этой ячейки и, удерживая левую клавишу мыши протащить указатель до конца списка товаров.

После копирования формулы в ячейку c5 она примет вид $=B5+B5*\$B\$1/100$. Результат – 2600. И т.д.

В формуле могут использоваться *внешние ссылки* – ссылки на ячейки других листов в текущей рабочей книге или на ячейки листов других рабочих книг.

Использование функций в формулах

Excel располагает большим количеством встроенных функций, которые облегчают выполнение вычислений в таблице.

Функцию в составе формулы можно записывать путем ввода символов с клавиатуры или с использованием *Мастера функций*. Для вызова мастера можно нажать кнопку *Вставка функции*, расположенную слева от строки формул  (существуют и другие способы вызова мастера). На экране появится первое из двух окон *Мастера функций*. В списке *Категория* выбирается категория, к которой относится функция (если определить категорию затруднительно, используют пункт Полный алфавитный перечень), а в списке *Функция* – конкретная функция данной категории. После выбора функции под списком функций можно увидеть ее формат, краткое описание действий и гиперссылку на справку по этой функции.

На втором шаге мастера в диалоговом окне *Аргументы функции* необходимо ввести все требуемые аргументы. Каждый аргумент вводится в специально предназначенное для него поле. Поля для ввода аргументов могут быть обязательными и необязательными для заполнения. Первые выделяются полужирным шрифтом. При вводе аргументов рядом с полем появляется их значение, а в нижней части окна – значение всей формулы. Затем в скобках необходимо указать аргументы.

После завершения второго шага *Мастера функций* в ячейке с функцией появится ее значение.

Примеры часто используемых функций

1) СУММ(Аргументы) – математическая функция – определяет сумму всех числовых значений в списке.

СУММ(B5:E7)

СУММ(a3..e3;250)

2) СРЗНАЧ(Список) – статистическая – среднее арифметическое.

3) МАКС(Список) – статистическая – максимальное значение в списке.

4) МИН(Список) – статистическая – минимальное значение в списке.

5) ЕСЛИ(Условие, Истинно, Ложно) – логическая функция – проверяет истинность заданного логического условия. Если условие истинно (выполняется), то результатом будет значение аргумента “Истинно”. Если условие ложно (не выполняется), результатом будет значение аргумента “Ложно”.

ЕСЛИ(b4<100,10,200) – если в ячейке b4 содержится значение меньше 100, то в ячейке с формулой отобразится значение 10; иначе (≥ 100) – 200.

Логические формулы, кроме операторов сравнения, могут содержать также следующие логические функции:

6) НЕ(Логическое_значение) – логическая – меняет логическое значение своего аргумента на противоположное (отрицание).

7) И(Список) – логическая – возвращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА; возвращает значение ЛОЖЬ, если хотя бы один аргумент имеет значение ЛОЖЬ.

8) ИЛИ(Список) – логическая – возвращает ИСТИНА, если хотя бы один из аргументов имеет значение ИСТИНА; возвращает ЛОЖЬ, если все аргументы имеют значение ЛОЖЬ.

10. Построение диаграмм

Электронные таблицы позволяют представлять числовые данные в виде диаграмм или графиков. Диаграммы бывают различных типов (столбчатые, круговые и т.д.); выбор типа диаграммы зависит от характера данных.

Диаграммы графически представляют данные числового типа. Числовым данным рабочего листа соответствуют элементы диаграммы, которые изображаются различными геометрическими фигурами (полосы, линии, столбики, сектора, точки и т.п.). При изменении исходных данных автоматически изменяются размеры или местоположение элементов диаграммы. При изменении элемента диаграммы (увеличить или уменьшить высоту столбика, изменить местонахождение точки и т.п.) автоматически изменяются соответствующие числовые значения в таблицах.

Для построения диаграммы необходимо выделить на листе необходимые ячейки и диапазоны ячеек. Очень часто диаграмма строится по таблице, состоящей из нескольких столбцов и строк. Как правило, заголовки столбцов отображают смысл данных. Поэтому рекомендуется выделять столбцы вместе с заголовками. Если столбцов несколько и формат первого столбца отличается от формата всех остальных столбцов, то среда Excel автоматически считает, что в этом столбце расположены данные для подписей по оси X.

Если для построения диаграммы используются не все ячейки таблицы, а только некоторые, то для выделения отдельных ячеек необходимо щелкнуть на первой ячейке, а затем, нажав клавишу Ctrl, на остальных ячейках.

Инструменты построения диаграмм расположены на вкладке *Вставка* в группе *Диаграммы*. В этой группе расположены кнопки построения конкретных типов диаграмм, а также кнопка *Другие диаграммы*, которая позволяет ознакомиться со всеми типами диаграмм и выбрать для построения подходящий тип.

После выбора типа диаграммы на листе автоматически появляется диаграмма. На ней отображаются объекты диаграммы с некоторым форматированием, заданным по умолчанию. Довольно часто данные на диаграмме отражаются не так, как нужно пользователю. Поэтому далее необходимо приступить к редактированию диаграммы. Кроме того, может потребоваться изменить форматирование объектов. Вместе с диаграммой на ленте появляется меню *Работа с диаграммами* и три вкладки: *Конструктор*, *Макет* и *Формат*, предназначенные для изменения диаграммы.

11. Печать в Excel

Область печати – это один или несколько диапазонов ячеек, выделенных для печати, если не требуется печатать весь лист. Если на листе задана область печати, то будет напечатана только она. (*Разметка страницы*, *Печатать заголовок* (или *Поля*, *Настраиваемые поля*) см. вкладку *Лист*).

Разрыв страницы – разделитель страницы для печати. В Excel разрывы вставляются автоматически в зависимости от размера бумаги, полей, масштаба.

Добавление колонтитулов

Обычно колонтитулы содержат номера страниц, даты, имена авторов, рисунки. Чтобы добавить верхний и/или нижний колонтитул/ы используйте вкладку на ленте *Разметка страницы*, кнопка *Параметры листа* (вкладка *Колонтитулы*).

Кроме колонтитулов на каждой странице можно выводить данные с листа – *заголовки* строк или столбцов, которые выводятся вверх или слева на каждой странице при печати. Заголовки строк – номера строк, отображаемые в левой части листа Excel; заголовки столбцов – буквы или числа, отображаемые в верхней части столбцов листа.

Также можно использовать в качестве заголовков строк или столбцов данные с листа – подписи строк или столбцов. Чтобы воспользоваться этой возможностью на вкладке *Разметка страницы* нажмите кнопку *Печатать заголовки*, откроется вкладка *Лист*.

Печать диаграмм

Внедренная диаграмма печатается вместе с данными листа, на котором она находится. Позицию диаграммы на листе для печати можно изменить, изменяя размер диаграммы и перемещая ее мышью в режиме разметки страницы.

Для диаграммы, размещенной на отдельном листе, можно изменять размер области диаграммы, масштабировать, изменять ее позицию на листе для печати, а также просматривать ее в окне предварительного просмотра.

12. Работа со списками данных в Excel

Общие сведения о списках

Список – это таблица Excel, данные в которой расположены, как в базе данных. В такой таблице каждый столбец имеет свой заголовок, который называется *именем поля*. Все имена полей располагаются в первой строке таблицы. На каждое имя отводится одна ячейка. Отдельный столбец такой таблицы называется *полем данных*, а каждая строка – *записью*. Запись состоит из элементов, число которых равно числу полей данных. Все записи имеют одинаковую структуру. На рис. 2.1 представлен пример таблицы Excel, оформленной в виде списка.

	A	B	C	D	E	F	G	H	I
1	Фамилия	Имя	Год рождения	Год смерти	Страна проживания	Опер	Балетов	Оперетт	Симфоний
2	Гайдн	Йозеф	1732	1809	Австрия	30	0	1	100
3	Моцарт	Вольфганг	1756	1791	Австрия	17	1	2	50
4	Госсек	Франсуа	1734	1829	Франция	20	0	0	29
5	Шостакович	Дмитрий	1906	1975	Россия	2	3	1	11
6	Бетховен	Людвиг	1770	1827	Германия	1	1	0	9
7	Шуберт	Франц	1797	1828	Австрия	7	0	6	9
8									

Рис. 2.1. Пример списка

Данные в списке должны иметь постоянный формат. Первая строка в списке содержит названия столбцов или имена полей как в базах данных. Наличие пустых строк и столбцов в списке недопустимо.

Для безошибочного выполнения операций над данными списка рекомендуется на рабочем листе создавать только один список. Таким образом, размер списка ограничен размерами рабочего листа.

Записи списка могут создаваться и редактироваться обычным образом – на рабочем листе, или с использованием диалогового окна, называемого **формой данных**.

К средствам, которые предназначены для обработки и анализа данных в списке относятся команды вкладки Данные: Сортировка, Фильтр, Итоги, Форма, Проверка данных. При выполнении этих команд, редактор автоматически распознает список как базу данных и осуществляет обработку и анализ данных в списке как в базе данных.

Сортировка

Под **сортировкой списка** понимается упорядочение его записей в алфавитном, числовом, хронологическом или другом (заданном пользователем) порядке. Сортировка может производиться по содержимому одного поля или нескольких полей одновременно. Причем в каждом поле может быть указан свой (убывающий или возрастающий) порядок сортировки.

Для выполнения сортировки используют соответствующие кнопки на вкладке Данные на Ленте. Для сортировки по одному полю (столбцу): по убыванию  или по возрастанию .

Для сортировки по нескольким полям используют кнопку «Сортировка» (рис. 2.2 а) и в диалоговом окне (рис. 2.2 б) указывают уровни, порядок и необходимые параметры (рис. 2.2 в).

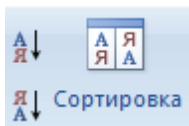


Рис. 2.2 а. Кнопка «Сортировка»

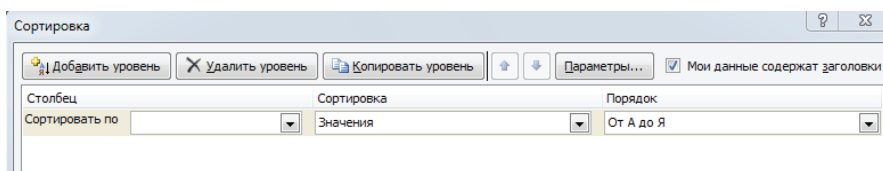


Рис. 2.2. б. Сортировка по нескольким полям

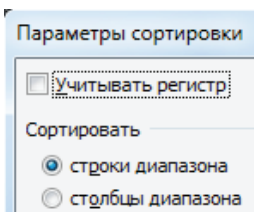
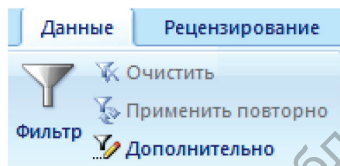


Рис. 2.2 в. Дополнительные параметры сортировки

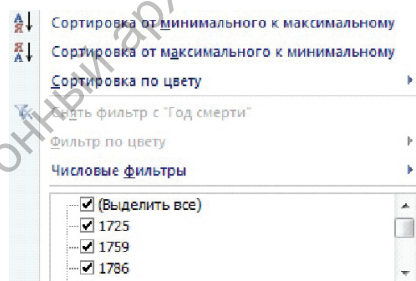
К средствам фильтрации относятся:

- Автофильтр (Фильтр).
- Расширенный фильтр (Дополнительно).

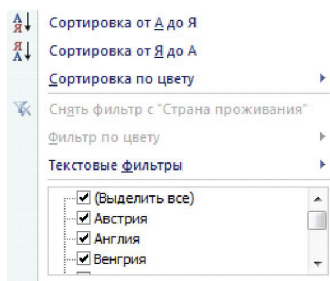


Автофильтр предназначен для простых условий отборов строк, а расширенный фильтр для более сложных условий отбора записей.

Как только команда *Фильтр (Автофильтр)* будет выполнена, в первой строке таблицы рядом с именем каждого поля появятся кнопки со стрелками. Нажатие любой из этих кнопок приводит к открытию соответствующего диалогового окна. Вид диалогового окна зависит от типа данных поля (см. рис. 2.3 а, б).



а) для числового поля



б) для текстового поля

Рис. 2.3. Автофильтр

Настраиваемый фильтр (рис. 2.4) позволяет наложить до двух условий на значения поля, при этом можно использовать символы шаблона «?» и «*».

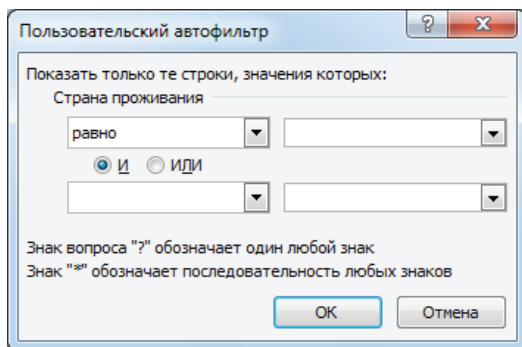


Рис. 2.4. Диалоговое окно «Пользовательский фильтр»

Расширенный фильтр рекомендуется использовать в следующих случаях:

- к ячейкам одного столбца необходимо применить три и более условий отбора;
- в условии отбора используется значение, полученное в результате вычисления заданной формулы.

При использовании расширенного фильтра условия отбора задаются в отдельном диапазоне ячеек текущего рабочего листа, который называется *диапазоном критериев*. Он может размещаться в любом месте рабочего листа. Рекомендуется отделять диапазон критериев от фильтруемого списка по крайней мере одной пустой строкой (столбцом). В этом случае исключается ошибочное восприятие программой ячеек диапазона критериев в качестве элементов списка.

Диапазон критериев оформляется следующим образом: в первой строке записываются (или копируются) имена полей списка, для которых задаются условия отбора, во второй и последующих строках вводятся непосредственно сами условия отбора. Причем, если два условия записаны в одной строке таблицы, то они связаны логическим И, если в разных строках – логическим ИЛИ.

Рассмотрим примеры формирования диапазонов критериев:

1. Требуется найти в списке английских, австрийских и венгерских композиторов.

Сформируем диапазон условий:

Страна проживания
Англия
Австрия
Венгрия

2. Требуется найти французских композиторов, написавших не менее 10 опер

Страна проживания	Опер
Франция	>9

3. Требуется найти композиторов, родившихся в XIX веке

Год рождения	Год рождения
>1800	<=1900

4. Требуется найти композиторов, проживших более 80 лет:

Возраст
=D2-C2>80

Замечание. Если в условии отбора используется значение, полученное по формуле, то подпись в первой строке диапазона условий не должна совпадать с именем какого-либо поля списка.

После создания диапазона критериев для применения расширенного фильтра необходимо:

1. Активизировать любую ячейку списка.
2. На вкладке **Данные** нажать кнопку **Дополнительно**
3. В диалоговом окне (рис. 2.5) указать:

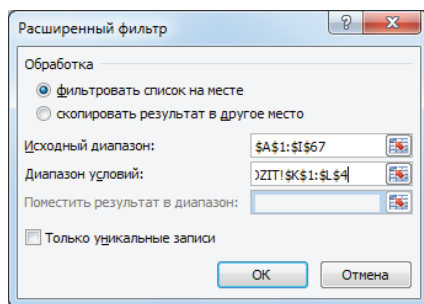


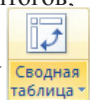
Рис. 2.5. Диалоговое окно «Расширенный фильтр»

- a. Исходный диапазон (выделяется автоматически).
- b. Диапазон условий
- c. Способ Обработки: **фильтровать список на месте** (по умолчанию) или **скопировать результат в другое место** – указать левую верхнюю ячейку.

13. Формирование сводной информации в Excel

Excel содержит средства формирования сводной информации для проведения анализа данных. Сводная информация может быть получена:

- объединением данных промежуточных итогов;
- методом консолидации;
- формированием сводных таблиц (кнопка находится на вкладке **Вставка**).



консолидация

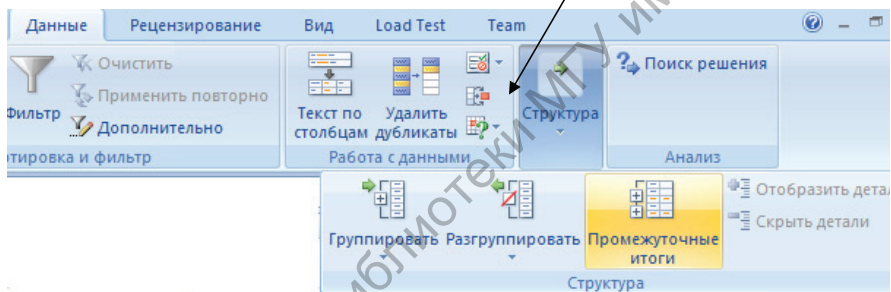


Рис. 2.6. Средства формирования сводной информации

Промежуточные итоги

Простейший способ получения сводных данных – с помощью команды **Промежуточные итоги** вкладки **Данные**.

Для выполнения этой команды необходимо:

- при помощи сортировки упорядочить записи по тому полю, при изменении значений которого будут подводиться промежуточные итоги;
- в диалоговом окне Промежуточные итоги задать нужные параметры.

Пример. Подведение промежуточных итогов по симфониям для каждой страны.

Вначале отсортируем данные по столбцу «Страна проживания». Затем откроем диалоговое окно «Промежуточные итоги» (рис. 2.7).

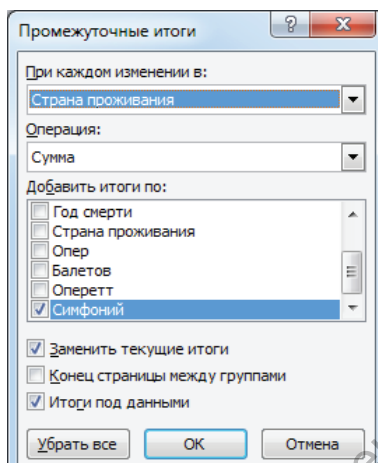


Рис. 2.7. Диалоговое окно «Промежуточные итоги»

После выполнения команды **Промежуточные итоги** создается структура (рис. 2.8), в которой данные разбиты на несколько уровней. С помощью уровней структуры можно управлять выводом данных соответствующего уровня из таблицы на экран, указывая, выводить данные или скрывать.

1	2	3	A	B	C	D	E	F	G	H	I
	1		Фамилия	Имя	Год рождения	Год смерти	Страна проживания	Опер	Балетов	Оперетт	Симфоний
	2		Гайды	Иозеф	1732	1809	Австрия	30	0	1	100
	3		Зуппе	Франц	1819	1895	Австрия	0	0	40	0
	4		Миллекер	Карл	1842	1899	Австрия	0	0	20	0
	5		Моцарт	Вольфганг	1756	1791	Австрия	17	1	2	50
	6		Целлер	Карл	1842	1898	Австрия	0	0	5	0
	7		Штраус	Иоганн	1825	1899	Австрия	0	0	16	0
	8		Шуберт	Франц	1797	1828	Австрия	7	0	6	9
	9						Австрия Итог				159
	10		Воан-Уильямс	Ралф	1872	1958	Англия	5	3	0	9
	11						Англия Итог				9
	12		Барток	Бела	1881	1945	Венгрия	1	2	0	1
	13		Кальман	Имре	1882	1953	Венгрия	0	0	20	2
	14		Легар	Франц	1870	1948	Венгрия	1	0	30	0
	15						Венгрия Итог				3
	16		Бетховен	Людвиг	1770	1827	Германия	1	1	0	9
	17		Брух	Макс	1838	1920	Германия	2	0	0	3
	18		Вагнер	Рихард	1813	1883	Германия	13	0	0	2

Рис. 2.8. Результат операции «Промежуточные итоги»

Для удаления промежуточных итогов необходимо в диалоговом окне команды Промежуточные итоги выбрать кнопку Убрать все. При удалении промежуточных итогов из списка удаляется и структура таблицы.

Консолидация данных в Excel

При консолидации данных объединяются значения из нескольких диапазонов данных. Инструмент **Консолидация** (по расположению) позволяет объединить таблицы, находящиеся на разных рабочих листах и даже в разных книгах. Но такие таблицы должны иметь **идентичную структуру**.

Для проведения консолидации данных по положению или категории:

1. Щелкните левый верхний угол области, в которой требуется разместить консолидированные данные.
2. Выполните команду **Данные, Консолидация**.
3. Выберите из раскрывающегося списка **Функция** функцию, которую требуется использовать для консолидации данных.
4. Щелкните поле **Ссылка**, укажите первый диапазон данных для консолидации и нажмите кнопку **Добавить**.

Повторите этот шаг для всех диапазонов.

5. Если таблицу консолидации требуется обновлять автоматически при каждом изменении данных в каком-либо исходном диапазоне, установите флажок **Создавать связи с исходными данными** (не работает, если результат консолидации находится на том же листе, что и исходный список).

6. Если консолидация выполняется по положению, оставьте все поля в группе **Использовать в качестве имен** пустыми. В Microsoft Excel подписи исходных строк и столбцов не копируются в консолидированные данные. Если требуется скопировать подписи в консолидированные данные, сделайте это вручную.

Если консолидация выполняется по категории, в группе **Использовать в качестве имен** установите флажки, соответствующие расположению подписей в исходных диапазонах: в верхней строке, в левом столбце или в верхней строке и в левом столбце одновременно. Все подписи, не совпадающие с подписями в других исходных областях, в консолидированных данных будут расположены в отдельных строках или столбцах.

Пусть имеются таблицы с наборами продуктов (обратите внимание, что набор продуктов в таблицах различается, но структура таблиц одинаковая).

Объем продаж за январь 2001			
Наименование	стоимость за 1 шт.	кол-во штук	сумма по продукту
Продукт1	1000 руб.	10000 шт.	10000000 руб.
Продукт2	2000 руб.	20000 шт.	40000000 руб.
Продукт3	3000 руб.	30000 шт.	90000000 руб.
Продукт5	5000 руб.	50000 шт.	250000000 руб.
Продукт6	6000 руб.	60000 шт.	360000000 руб.
Продукт7	7000 руб.	70000 шт.	490000000 руб.
Продукт8	8000 руб.	80000 шт.	640000000 руб.
Продукт9	9000 руб.	90000 шт.	810000000 руб.
Продукт10	10000 руб.	10000 шт.	100000000 руб.

Объем продаж за февраль 2001			
Наименование	стоимость за 1 шт.	кол-во штук	сумма по продукту
Продукт1	10 руб.	44000 шт.	440000 руб.
Продукт2	20 руб.	25000 шт.	500000 руб.
Продукт3	30 руб.	14000 шт.	420000 руб.
Продукт4	140 руб.	35000 шт.	4900000 руб.
Продукт5	50 руб.	42000 шт.	2100000 руб.
Продукт6	60 руб.	17000 шт.	1020000 руб.
Продукт7	70 руб.	19000 шт.	1330000 руб.
Продукт8	80 руб.	45000 шт.	3600000 руб.
Продукт9	90 руб.	32000 шт.	2880000 руб.
Продукт10	100 руб.	10000 шт.	1000000 руб.

Объем продаж за март 2001			
Наименование	стоимость за 1 шт.	кол-во штук	сумма по продукту
Продукт1	10 руб.	50000 шт.	500000 руб.
Продукт2	20 руб.	25000 шт.	500000 руб.
Продукт3	30 руб.	12000 шт.	360000 руб.
Продукт4	40 руб.	31000 шт.	1240000 руб.
Продукт5	50 руб.	19000 шт.	950000 руб.
Продукт9	90 руб.	61000 шт.	5490000 руб.
Продукт10	100 руб.	16000 шт.	1600000 руб.

Рис. 2.9. Примеры листов с данными идентичной структуры

На рабочем листе Квартал необходимо просуммировать данные за три месяца. Для этого:

1. Сделайте активной ячейку на рабочем листе **Квартал**.
2. Воспользуйтесь кнопкой **Консолидация** (см.рис. 2.6).
3. В диалоговом окне (рис.2.10 а) в списке **Функция** выберите операцию для обработки консолидируемых данных – **Сумма**.
4. В поле ввода **Ссылка** укажите первый диапазон ячеек, данные которого должны подвергнуться консолидации (Наиболее удобно отмечать диапазон ячеек при помощи мыши) и нажмите кнопку **Добавить** (рис. 2.10 б).

Консолидация

Функция: **Сумма**

Ссылка: "Продажа 01.01"!\$A\$3:\$D\$12

Список диапазонов:

Добавить

Удалить

Использовать в качестве имен

☒ подписи верхней строки

☒ значения левого столбца

☒ Создавать связи с исходными данными

ОК

Закрыть

а)

Консолидация

Функция: **Сумма**

Ссылка: "Продажа 01.01"!\$A\$3:\$D\$12

Список диапазонов:

Добавить

Удалить

Использовать в качестве имен

☒ подписи верхней строки

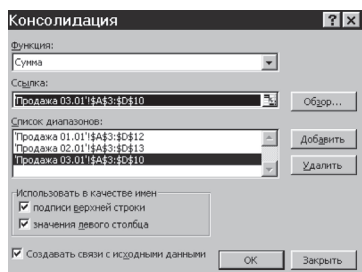
☒ значения левого столбца

☒ Создавать связи с исходными данными

ОК

Закрыть

б)



в)

Рис. 2.10. Диалоговое окно «Консолидация»

Аналогичным образом включите остальные диапазоны.

5. Так как верхняя строка и левый столбец содержат заголовки таблицы, включим их в итоговую таблицу, установив соответствующие флажки в области **Использовать в качестве имен**.

6. Если между исходными данными должна быть установлена динамическая связь, необходимо включить флажок **Создавать связи с исходными данными** (рис. 2.10 в).

7. Нажмите кнопку ОК и через некоторое время появится итоговая таблица (рис. 2.11).

При необходимости таблицу можно отредактировать.

1	2	A	B	C	D
	1			Объем продаж за 1 квартал 1998	
	2				
	3			кол-во штук	сумма по продукту
+	7	Продукт1		104000 шт,	10940000 руб,
+	11	Продукт2		70000 шт,	41000000 руб,
+	15	Продукт3		56000 шт,	90780000 руб,
+	18	Продукт4		66000 шт,	6140000 руб,
+	22	Продукт5		111000 шт,	253050000 руб,
+	25	Продукт6		77000 шт,	361020000 руб,
+	28	Продукт7		89000 шт,	491330000 руб,
+	31	Продукт8		125000 шт,	643600000 руб,
+	35	Продукт9		183000 шт,	818370000 руб,
+	39	Продукт10		36000 шт,	102600000 руб,
	40				
	41				

Рис. 2.11. Результат консолидации

Сводные таблицы

Сводная таблица представляет собой интерактивный метод быстрого суммирования больших объемов данных. Она используется для подробного анализа числовых данных и для ответов на непредвиденные вопросы по данным. Сводная таблица позволяет:

- организовывать запросы к большим массивам данных дружественными по отношению к пользователю способами;
- подводить промежуточные итоги;
- применять статистические функции к числовым данным;
- суммировать данные по категориям и подкатегориям;
- создавать дополнительные вычисления и формулы.

Интерфейс сводной таблицы организован так, что позволяет развертывать и свертывать уровни представления данных для привлечения внимания к результатам.

Чтобы начать создание сводной таблицы следует

1. Выполнить команду *Вставка/Таблицы/Сводная таблица/Сводная таблица*. Откроется диалоговое окно *Создание сводной таблицы* (рис. 2.12).

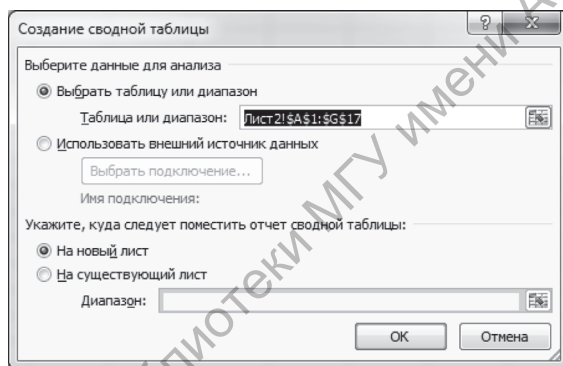


Рис. 2.12. Диалоговое окно *Создание сводной таблицы*

2. Установить переключатель в положение *Выбрать таблицу или диапазон*, если данные берутся из той же книги.

3. В поле *Таблица или диапазон* ввести имя диапазона, по которому будет строиться сводная таблица.

4. Установить переключатель в положение *На новый лист*, так как удобнее располагать сводную таблицу на отдельном листе. Нажать ОК.

После выполнения указанных действий на листе появится макет сводной таблицы и область задач *Список полей сводной таблицы*. Одновременно на ленте появляется дополнительное меню *Работа со сводными таблицами*, которое имеет вкладки *Параметры* и *Конструктор*.

Макет сводной таблицы состоит из следующих областей:

- *Поля страниц* – здесь размещаются поля, значения которых обеспечивают отбор записей на первом уровне. На странице может быть размещено несколько полей, между которыми устанавливается иерархия связи – сверху вниз. Страницу определять необязательно;

- *Поля столбцов* – поля размещаются слева направо, обеспечивая группировку данных сводной таблицы по иерархии полей. При условии существования области страницы или строки определять столбец не обязательно;
- *Поля строк* – поля размещаются сверху вниз, обеспечивая группировку данных таблицы по иерархии полей. При условии существования области страницы или столбцов определять строку необязательно;
- *Элементы данных* – поля, по которым подводятся итоги, согласно выбранной функции. Область определять обязательно.

5. Разместить поля данных в области сводной таблицы.

Справа от сводной таблицы располагается окно *Список полей сводной таблицы*. По замыслу разработчиков на одном и том же макете сводной таблицы пользователь может, меняя заголовки полей в областях, анализировать данные с разными целями.

Размещение полей выполняется путем их перетаскивания из окна *Список полей сводной таблицы* в определенную область макета. Каждое поле размещается в сводной таблице только в одной из областей. Поля, помещенные в области сводной таблицы, отмечаются в окне *Список полей сводной таблицы* полужирным начертанием и флажком. Если флажок снять, поле автоматически удалится из сводной таблицы.

6. Изменить функцию подсчета итоговых значений.

В области элементов данных формируются итоговые значения в соответствии с выбранной функцией. При первичном формировании элементам данных автоматически присваивается функция *Количество*, которая подсчитывает количество ненулевых значений по полю. Однако гораздо чаще пользователя интересуют другие функции, например, *Сумма*, *Среднее*.

Функцию итоговых данных можно изменять. Для быстрой смены функции можно выбрать в контекстном меню команду *Параметры полей значений* или *Итоги по*, а затем в меню *Дополнительно* – итоговую функцию.

14. Сервисные инструментальные средства:

Файловые менеджеры

Файловый менеджер – это программа, которая предоставляет интерфейс пользователя для работы с файловой системой и файлами.

Файловый менеджер позволяет выполнять наиболее частые операции над файлами: создание, открытие/проигрывание/просмотр, редактирование, перемещение, переименование, копирование, удаление, изменение атрибутов и свойств, поиск файлов и назначение прав. Помимо основных функций, многие файловые менеджеры включают ряд дополнительных возможностей,

например, таких, как работа с сетью (через FTP, NFS и т. п.), резервное копирование, управление принтерами и др.

Выделяют различные типы файловых менеджеров. Например, стандартный для Windows файловый менеджер Проводник относят к так называемым навигационным. Двухпанельные (Total Commander, Far Manager) – в общем случае имеют две равноценных панели для списка файлов, дерева каталогов и т. п. Существуют также трехмерные файловые менеджеры (NavScope).

Средства сжатия данных

Характерной особенностью большинства типов данных является их избыточность. Степень избыточности данных зависит от типа данных. Например, для видеоданных степень избыточности в несколько раз больше чем для графических данных, а степень избыточности графических данных, в свою очередь, больше чем степень избыточности текстовых данных. Другим фактором, влияющим на степень избыточности является принятая система кодирования.

Для человека избыточность данных часто связана с качеством информации, поскольку избыточность, как правило, улучшает понятность и восприятие информации. Однако, когда речь идет о хранении и передаче информации средствами компьютерной техники, то избыточность играет отрицательную роль, поскольку она приводит к возрастанию стоимости хранения и передачи информации. Особенно актуальной эта проблема станет в случае обработки огромных объемов информации при незначительных объемах носителей данных. В связи с этим постоянно возникает проблема уменьшения избыточности или сжатия данных.

Если методы сжатия данных применяются к готовым файлам, то часто вместо термина “сжатие данных” употребляют термин “архивация данных”, сжатый вариант данных называют архивом, а программные средства, которые реализуют методы сжатия называются архиваторами.

Программы-архиваторы предназначены для создания архивов. Архивные файлы имеют повышенную плотность записи информации и соответственно, эффективнее используют носители информации.

В зависимости от того, в каком объекте размещены данные, подлежащие сжатию различают:

1. Сжатие (архивация) файлов: используется для уменьшения размеров файлов при подготовке их к передаче каналами связи или к транспортированию на внешних носителях маленькой емкости;
2. Сжатие (архивация) папок: используется как средство уменьшения объема папок перед долгим хранением, например, при резервном копировании;
3. Сжатие (уплотнение) дисков: используется для повышения эффективности использования дискового пространства путем сжатия данных при записи их на носителе информации (как правило, средствами операционной системы).

Существует много практических алгоритмов сжатия данных, но все они базируются на трех теоретических способах уменьшения избыточности данных. Первый способ состоит в изменении содержимого данных, второй – в изменении структуры данных, а третий – в одновременном изменении как структуры, так и содержимого данных.

Если при сжатии данных происходит изменение их содержимого, то метод сжатия называется **необратимым**, то есть при восстановлении (разархивировании) данных из архива не происходит полное восстановление информации. Такие методы часто называются методами сжатия с регулирурованными потерями информации. Понятно, что эти методы можно применять только для таких типов данных, для которых потеря части содержимого не приводит к существенному искажению информации. К таким типам данных относятся видео- и аудиоданные, а также графические данные. Методы сжатия с регулирующими потерями информации обеспечивают значительно большую степень сжатия, но их нельзя применять к текстовым данным. Примерами форматов сжатия с потерями информации могут быть:

- JPEG – для графических данных;
- MPG – для видеоданных;
- MP3 – для аудиоданных.

Если при сжатии данных происходит только изменение структуры данных, то метод сжатия называется **обратимым**. В этом случае, из архива можно восстановить информацию полностью. Обратимые методы сжатия можно применять к любым типам данных, но они дают меньшую степень сжатия по сравнению с необратимыми методами сжатия. Примеры форматов сжатия без потери информации:

- GIF, TIFF – для графических данных;
- AVI – для видеоданных;
- ZIP, ARJ, RAR, CAB, LH – для произвольных типов данных.

Существует много разных практических методов сжатия без потери информации, которые, как правило, имеют разную эффективность для разных типов данных и разных объемов. Однако в основе этих методов лежат три теоретических алгоритма:

- алгоритм RLE (Run Length Encoding);
- алгоритмы группы KWE(KeyWord Encoding);
- алгоритм Хаффмана.

На практике программные средства сжатия данных синтезируют эти три “чистых” алгоритма, поскольку их эффективность зависит от типа и объема данных.

Вот программы-архиваторы, использующиеся на практике: ARJ, RAR, ZIP. Все современные архиваторы предоставляют пользователю полный спектр услуг для работы с архивами, основными из которых являются:

- создание нового архива;
- добавление файлов в существующий архив;
- распаковывание файлов из архива;
- создание самораспаковывающихся архивов (self-extractor archive);
- создание распределенных архивов фиксированного размера для носителей маленькой емкости;
- защита архивов паролями от несанкционированного доступа;
- просмотр содержимого файлов разных форматов без предварительного распаковывания;
- поиск файлов и данных внутри архива;
- проверка на вирусы в архиве к распаковыванию;
- выбор и настройка коэффициента сжатия.

Системы автоматизированного перевода

Различают электронные словари и программы перевода языка.

Электронные словари – это средства для перевода отдельных слов в документе. Используются профессиональными переводчиками, которые самостоятельно переводят текст.

Программы автоматического перевода используют текст на одном языке и выдают текст на другом, то есть автоматизируют перевод. При автоматизированном переводе невозможно получить качественный исходный текст, поскольку все сводится к переводу отдельных лексических единиц. Но, для технического текста, этот барьер снижен.

Программы автоматического перевода целесообразно использовать:

- при абсолютном незнании иностранного языка;
- при необходимости быстрого ознакомления с документом;
- для создания черновика, который потом будет подправлен полноценным переводом.

К популярным программам-переводчикам относят **Prompt** и **Lingvo**. Также в сети Интернет существуют он-лайн переводчики, предоставляющие услуги перевода как отдельных слов, так и фрагментов текста.

Программы распознавания текста

Теория распознавания образов – раздел кибернетики, развивающий теоретические основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и других объектов, которые характеризуются конечным набором некоторых свойств и признаков. Такие задачи решаются довольно часто, например, при переходе или проезде улицы по сигналам светофора. Распознавание цвета загоревшейся лампы светофора и знание правил дорожного движения позволяет принять правильное решение о том, можно или нельзя переходить улицу в данный момент.

В процессе эволюции многие животные с помощью зрительного и слухового аппарата достаточно хорошо решили задачи распознавания образов. Создание искусственных систем распознавания образов остается сложной теоретической и технической проблемой. Необходимость в таком распознавании возникает в самых разных областях – от военного дела и систем безопасности до оцифровки всевозможных аналоговых сигналов. Традиционно задачи распознавания образов включают в круг задач *искусственного интеллекта*.

Системы оптического распознавания текста – OCR-системы (optical character recognition) предназначены для ввода печатного текста для печатных и электронных изданий. В русскоговорящих странах наиболее известной программой этого класса ABBYY Fine Reader.

Программа позволяет извлекать текстовые данные из цифровых изображений (фотографий, результатов сканирования, PDF-файлов). Полученное в результате распознавания содержимое может быть сохранено в различных форматах файлов, включая Microsoft Word, Excel, PDF и djvu. Последние версии программы поддерживают распознавание текста на нескольких языках и имеют встроенную проверку орфографии для этих языков. Существует также онлайн-версия Fine Reader.

Основные этапы преобразование документа в электронный вид OCR-системами:

- сканирование и предварительная обработка изображения;
- анализ структуры документа;
- распознавание;
- проверка результатов;
- реконструкция документа (воссоздание его исходного вида);
- экспорт.

15. Системы математических вычислений

С развитием информационных технологий появились так называемые системы компьютерной математики, или математические пакеты, которые облегчают выполнение различных математических задач, помогают проверить решение задачи с помощью компьютерной программы. Намного сокращается время выполнения задач различной сложности. Для сотен тысяч специалистов в различных отраслях промышленности, занятых инженерными и научными исследованиями, системы компьютерной математики обеспечили превосходную среду для организации вычислений. Они имеют чрезвычайно широкий набор средств, переводящих сложные математические алгоритмы в программы, так называемые элементарные функции, алгебраические и логические операции. Большинство упражнений из курса высшей математики может

быть решено с помощью всего лишь одной команды. Можно вычислять интегралы, решать дифференциальные уравнения, обыкновенные уравнения и системы линейных уравнений. Предоставлен широкий выбор работы с матрицами, векторами. Возможно построение двумерных и трехмерных графиков.

Существует несколько математических пакетов, таких, как Mathcad, MATLAB, Mathematica, Maple, Statistica, SPSS и др.

Сравнительный анализ популярных систем компьютерной математики представлен в таблице 2.1.

Таблица 2.1 – Сравнение математических пакетов

Критерий сравнения	Mathcad	MATLAB	Mathematica
Интерфейс	Типа “wysiwyg”. Палитры инструментов для построения выражений.	Три окна: командное окно, все переменные и их типы и окно подсказок. Строка приглашения, обозначается знаком “>>”. Все функции приходится вводить с клавиатуры.	Строка приглашения в отличие от MATLAB разделена на две области: ввода и вывода, которые составляют вместе область всего выражения. Область ввода можно редактировать. Также имеется палитра с греческими буквами, различными символами и панель матанализа.
Работа с массивами и матрицами	Достаточный набор функций для операций с матрицами и векторами. Некоторые на палитре, другие – вводить с клавиатуры или вставлять из меню Вставка->Функции .	Матрицы и вектора формируются при помощи списка элементов. Функции вводятся с клавиатуры.	
Решение уравнений и систем			
Математические операторы	Приведен в таблице целый ряд операторов, как простых типа сложения, так и вычисления суммы, произведения, интегралов и производных и т.д., которые можно вводить с клавиатуры или вставлять из соответствующей палитры.	Здесь в отличие от Mathcad все операторы вводятся с клавиатуры в виде отдельных символов и функций. Дан относительно подробный список операторов.	Также, как и в MATLAB операторы приходится вводить с клавиатуры, но некоторые можно найти и на палитре инструментов.
Встроенные функции	Построены по принципу всех функций: название функции и параметры в скобках. Функции: упрощения выражения, раскрытия скобок, тригонометрические и др.	Функции вводятся с клавиатуры.	Множество функций различного.
Программирование	Предоставлены шаблоны для создания программ и подпрограмм. В качестве выходного значения указывается последнее значение, вычисленное программой. Можно использовать функции, описанные ранее. Программы пишутся в том же файле, что и все вычисления.	Программы создаются в виде отдельных М-файлов. Если написать программу какой-нибудь функции, то эту функцию можно будет использовать как стандартную.	Можно создавать различные функции и оперировать с ними. Внутри одного блока ввода можно создавать программы. Результатом будет последнее вычисленное значение.

Критерий сравнения	Mathcad	MATLAB	Mathematica
Графические возможности	Графики строятся на основе имеющихся шаблонов. Основные виды: график в декартовой плоскости, в полярной системе координат, трехмерный в виде гладкой поверхности, в виде контурных кривых и т.д. Сначала задается функция графика, диапазон, затем строится сам график, который можно редактировать.	Функция графика создается из командной строки. Графики создаются на формах в определенной системе координат. В команде построения можно указывать свойства графика.	Функция, которая рисует график, заканчивается на "PLOT" в двухмерном случае, и "PLOT 3D" в трехмерном случае. Чтобы построить график, нужно сначала задать функцию. Также можно и редактировать график.

Как видно, все математические пакеты сходны между собой. У них похожие принципы построения вычислений, графиков функций. И в Mathcad, и в MATLAB, и в Mathematica есть списки встроенных функций и операторов. Но имеются и отличия, например в интерфейсе, методах программирования.

16. Система подготовки презентаций Power Point

Презентация, созданная средствами Power Point, представляет собой комплект подготовленных для показа слайдов.

Первый слайд презентации называется *титульный слайд*, он содержит два слота: для ввода заголовка и подзаголовка. Каждый следующий слайд по умолчанию имеет тип «заголовок и текст». Вообще имеется несколько типов авторазметки слайдов. Тип слайда выбирается после нажатия кнопки «Макет» на Главной вкладке. Понять, каким образом спроектирован тот или иной слайд, можно из его названия или графического изображения-пиктограммы.

Добавить слайд в презентацию можно щелкнув кнопку **Создать слайд** на главной вкладке или нажав комбинацию клавиш **Ctrl+Б**. После этого можно выбрать желаемый макет слайда.

Укажем *основные элементы слайдов*:

- Текст
- Таблицы
- Диаграммы
- Рисунки, клипы
- Рисунки SmartArt

Кроме обычных диаграмм, на слайдах можно использовать так называемые рисунки SmartArt (в Office 2007; в более ранних версиях для этой цели использовались организационные диаграммы), которые особенно удобны, например, при представлении некоторой структуры или схемы.

Рисунки SmartArt служат для наглядного представления данных, их можно быстро и легко создать на основе различных макетов, чтобы эффективно

донести сообщение или идею. Если на слайде уже имеется текст, то его можно преобразовать в рисунок SmartArt. Кроме того, в рисунок SmartArt можно добавить анимацию.

При создании рисунка SmartArt предлагается выбрать его тип, например **Процесс, Иерархия, Цикл** или **Связь**. Тип соответствует категории рисунка SmartArt и содержит несколько различных макетов. Можно легко изменить выбранный макет рисунка SmartArt. Большая часть текста и другого содержимого – цвета, стили, эффекты и форматирование текста – автоматически переносится в новый макет.

Поскольку можно быстро и легко переключать макеты, уместно будет попробовать несколько разных типов и найти тот, который лучше всего иллюстрирует информацию.

Для настройки рисунка SmartArt можно использовать добавление отдельных фигур, удаление фигур, перемещение фигур, изменение размера фигур и форматирование текста.

Использование анимации

В PowerPoint предусмотрена возможность использования специальных эффектов при переходе от одного слайда к другому. Благодаря этому следующий слайд может отображаться на экране не просто так, а постепенно, прорисовываясь, например, в шахматном порядке, вертикальными полосами и т.д. Выбрать нужный эффект для текущего слайда можно на вкладке «Анимация». Применить этот эффект ко всем слайдам данной презентации можно щелкнув по кнопке «Применить ко всем». Также можно настроить скорость перехода и добавить звуковой эффект.

Можно также настроить особенности *появления* (вход), *выделения* и *исчезновения* (выход) отдельных элементов слайда (кнопка «Настройка анимации»).

Демонстрация презентации

Для демонстрации презентации перед аудиторией из 5-7 человек достаточно обычного компьютера. При работе с большей аудиторией используются различные технические средства (например, проектор и проекционный экран), позволяющие увеличить изображение. Различные возможности демонстрации презентации размещены на вкладке «Показ слайдов».

Начать демонстрацию презентации можно с начала или с текущего слайда. Для управления показом во время демонстрации презентации можно использовать клавиатуру и мышь. Чтобы акцентировать внимание на важных деталях слайда, во время показа можно использовать указатель. Вид и цвет указателя выбирается в контекстном меню после щелчка правой кнопкой мыши на слайде.

Если предполагается демонстрация презентации без выступления, то показ можно отрепетировать. *Репетиция* позволяет определить длительность презентации и задать длительность демонстрации каждого слайда на основании результатов репетиции. Для проведения репетиции презентации используйте кнопку «Настройка времени» на вкладке «Показ слайдов».

Во время репетиции для перехода к следующему слайду используйте клавиатуру или мышь. Программа сохраняет время отображения на экране каждого слайда и общее время показа презентации.

Когда будет закрыт последний слайд, вы увидите диалоговое окно, в котором можно применить записанное в ходе репетиции время отображения каждого слайда.

Бесконечный показ

Дополнительные возможности демонстрации презентации можно указать в специальном диалоговом окне после щелчка по кнопке «Настройка демонстрации» на вкладке «Показ слайдов».

Создание произвольных показов.

В приложении PowerPoint предусмотрена возможность создания в одном файле презентации нескольких отличающихся друг от друга слайд-шоу. Эти слайд-шоу называются *пользовательскими показами*. Например, требуется создать презентацию, которая информирует о достижениях компании, как для всех ее сотрудников, так и для руководящего состава. Можно создать презентацию, представляющую все достижения компании, а конфиденциальную информацию скрыть от всех, кроме администрации. В таком случае создается пользовательский показ, который опускает отдельные слайды.

Для создания пользовательского показа нужно на вкладке «Показ слайдов» нажать кнопку «Произвольный показ» и в диалоговом окне нажать кнопку «Создать». Далее следует указать название создаваемого показа и выбрать те слайды презентации, которые должны быть включены в данный показ.

Пользовательских показов одной презентации можно создать сколько угодно. Каждый произвольный показ – это часть полной презентации, набор отдельных ее слайдов.

Основные правила разработки учебных презентаций

Оптимальный объем – не более 20 слайдов. Зрительный ряд из большого числа слайдов вызывает утомление, отвлекает от сути изучаемой темы.

Структура слайда должна быть одинаковой на всей презентации.

Слайд. По возможности используйте верхнюю часть слайда, т.к. с последних рядов нижняя часть экрана бывает не видна. Не полностью заполненный слайд лучше, чем переполненный.

Заголовки должны быть короткими. Точка в конце заголовка не ставится.

Графика чаще всего раскрывает концепции или идеи гораздо эффективнее текста. Если можно заменить текст информативной иллюстрацией, лучше это сделать. Визуализация сильно помогает аудитории. Помещайте картинки левее текста: мы читаем слева-направо, так что смотрим вначале на левую сторону слайда. При использовании скриншотов (копий экрана) лишние элементы (панели инструментов, меню, пустой фон и т.д.) необходимо обрезать. Чтобы на изображении не остались красные (зеленые) подчеркивания ошибок, следует использовать скриншоты в режиме просмотра перед печатью.

Текст. Текст должен быть читабельным, его должно быть легко прочитать с самого дальнего места. Минимальный размер текста – 24 пт. Лучше разместить короткие тезисы, даты, имена, термины, которые часто переспрашиваются при записи материала. Текст слайда не должен повторять текст, который выступающий произносит вслух. Зрители прочтут его быстрее, чем расскажет выступающий, и потеряют интерес к его словам.

Наиболее важный материал, требующий обязательного усвоения, лучше **выделить ярче** для включения ассоциативной зрительной памяти. Тени уменьшают четкость без увеличения информативности.

Цвета. Не используйте больше 2-3 цветов на слайде. Один и тот же элемент на разных слайдах должен быть одного цвета.

Фон. Плохо смотрятся фоны, содержащие активный рисунок. Обычно вместо того, чтобы использовать сплошной цвет лучше выбрать хорошую текстуру и нейтральный фон.

Анимация не должна быть слишком активной. Для смены слайдов лучше используйте один и тот же анимационный эффект.

Звуковое сопровождение должно отражать суть или подчеркивать особенность темы слайда, презентации. Звуковые и визуальные эффекты не должны отвлекать внимание от основной информации.

Глава 3

СЕТЕВЫЕ ТЕХНОЛОГИИ И ИНТЕРНЕТ

1. Компьютерные сети

Компьютерная сеть – это совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к имеющимся ресурсам сети: программным, техническим, информационным.

По степени территориальной рассредоточенности основных элементов сети (абонентских систем, узлов связи) различают глобальные, региональные и локальные компьютерные сети.

Глобальные компьютерные сети (ГКС, Wide Area Network, WAN) объединяют абонентские системы, рассредоточенные на большой территории, охватывающей различные страны и континенты. Они решают проблему объединения информационных ресурсов всего человечества и организации доступа к ним. Для взаимодействия абонентов используются телефонные линии связи, радиосвязь, системы спутниковой связи.

Региональные компьютерные сети (РКС, Regional Area Network, RAN) объединяют абонентские системы, расположенные в пределах отдельного региона – города, административного района; как правило, имеют выход в ГКС.

Локальные компьютерные сети (ЛКС, Local Area Network, LAN) объединяют абонентские системы, расположенные в пределах небольшой территории (этаж здания, здание, несколько зданий одного и того же предприятия).

Отдельный класс представляют **корпоративные компьютерные сети** (ККС), которые являются технической базой компаний, корпораций, организаций и т. д. Такая сеть играет ведущую роль в реализации задач планирования, организации и осуществления производственно-хозяйственной деятельности корпорации.

Глобальная компьютерная сеть

Первые исследования в области соединения удаленных компьютеров были проведены в начале 60-х гг. После Карибского кризиса в США были начаты работы по организации надежной системы передачи информации и созданию первой компьютерной сети.

Проект сети компьютеров на основе коммутации пакетов был разработан Агентством Министерства обороны США по перспективным исследовани-

ям (ARPA). Создаваемая сеть получила название ARPANET. В 1969 г. началась техническая реализация проекта, в результате которой в сеть ARPANET были включены четыре удаленных компьютера четырех научных учреждений США.

Для включения в сеть большего числа компьютеров необходимо было выработать некоторый единый набор правил, определяющих способ взаимодействия узлов сети ARPANET. Набор таких правил получил название “протокол”.

Первым единым протоколом для сети ARPANET стал Network Control Protocol (NCP) – 1970 год. Создание стандартного протокола позволило начать разработку прикладных программ для использования в сети. Одной из первых стала программа электронной почты. Она позволила участникам проекта эффективно обмениваться информацией.

Передача файлов, удаленная регистрация и электронная почта произвели огромный эффект и сделали сеть необычайно популярной. В 1973 г. через трансатлантический телефонный кабель к сети были подключены первые иностранные организации из Великобритании и Норвегии, сеть стала международной.

Локальная компьютерная сеть

К классу ЛКС относятся сети предприятий, фирм, банков, офисов, учебных заведений и т.д. Также существуют локальные сети, узлы которых разнесены географически на расстояния более 12 500 км (космические станции и орбитальные центры). Несмотря на такие расстояния, подобные сети все равно относят к локальным. Принципиальным отличием ЛКС от других классов сетей является наличие своей штатной системы передачи данных.

Все сети имеют некоторые *общие компоненты*, функции и характеристики. В их числе:

- *серверы* – компьютеры, предоставляющие свои ресурсы сетевым пользователям;
- *клиенты* – компьютеры, осуществляющие доступ к сетевым ресурсам, предоставляемым сервером;
- *среда* – способ соединения компьютеров.

Локальные сети разделяются на два типа:

- одноранговые сети;
- сети на основе сервера.

В одноранговой сети все компьютеры равноправны: нет иерархии среди компьютеров и нет выделенного сервера. Как правило, каждый компьютер функционирует и как клиент, и как сервер; иначе говоря, нет отдельного компьютера, ответственного за администрирование всей сети. Все пользователи самостоятельно решают, что на своем компьютере можно сделать общедоступным по сети и кому.

Одноранговые сети называют также *рабочими группами*. Рабочая группа – это небольшой коллектив, поэтому в одноранговых сетях чаще всего не более 10 компьютеров.

Если к сети подключено более 10 пользователей, то одноранговая сеть, где компьютеры выступают в роли и клиентов, и серверов, может оказаться недостаточно производительной. Поэтому большинство сетей использует выделенные серверы. Выделенным называется такой компьютер, который функционирует только как сервер. Они специально оптимизированы для быстрой обработки запросов от сетевых клиентов и для управления файлами и каталогами. Сети на основе серверов в настоящее время являются промышленным стандартом.

Сетевой администратор – человек, ответственный за работу локальной сети или ее части. В его обязанности входит обеспечение и контроль физической связи, настройка активного оборудования, настройка общего доступа и программ, обеспечивающих стабильную работу сети.

Существуют и комбинированные типы сетей, совмещающие лучшие качества одноранговых сетей и сетей на основе сервера. Как правило, в сетях на основе выделенных серверов пользователи могут и самостоятельно предоставлять друг другу доступ к своим локальным ресурсам время от времени.

2. Сеть Интернет и сетевые протоколы

В протоколе NCP не предполагалось наличие какого-либо механизма взаимодействия с сетями другой, нежели ARPANET, архитектуры. Эта явилось препятствием для объединения нескольких независимых компьютерных сетей с различной архитектурой. Кроме того, при объединении сетей, необходимо было учесть возможность временного нарушения связи или выхода из строя фрагмента сети. Оставшаяся часть сети должна продолжать нормально функционировать.

В результате, были сформулированы основные принципы построения глобальной сети:

- 1) для включения в сеть отдельной подсети не должно производиться никаких дополнительных изменений;
- 2) пакеты в сети передаются на основе принципа негарантированной доставки: если пакет не смог достигнуть пункта назначения, то через короткое время он должен быть передан снова;
- 3) для соединения подсетей используются маршрутизаторы – устройства, упрощающие прохождение потока пакетов;
- 4) сеть должна быть децентрализованной – единое, централизованное управление объединенной сетью отсутствует.

В 1973 г. был разработан новый протокол, поддерживающий межсетевое взаимодействие, – TCP (Transmission Control Protocol).

Протокол TCP был способен предоставить достаточно широкий спектр транспортных услуг, таких, как передача файлов или удаленная регистрация. Но при передаче речи потери пакетов невозможно исправить средствами TCP. Функции восстановления должны были выполняться на уровне приложений. Этот факт привел к необходимости разделения протокола TCP на два протокола: IP для адресации и передачи отдельных пакетов и TCP для разделения сообщений на пакеты, обеспечения целостности и восстановления потерянных пакетов. Кроме того, для приложений, которые по техническим причинам не могли использовать TCP, был добавлен альтернативный протокол, названный UDP (User Datagram Protocol – протокол пользовательских датаграмм). Часть UDP предназначалась для передачи данных в сетях без установления соединения и обеспечивала прямой доступ к IP. Объединенный протокол получил название TCP/IP. В 1980 году протокол TCP/IP был принят в качестве стандарта для сети ARPANET.

Переход на использование TCP/IP позволил разделить сеть ARPANET на две различные сети: MILNET, предназначенную для военных целей, и ARPANET – для использования исследовательскими и научными организациями.

3. Семиуровневая модель структуры протоколов связи

Для обеспечения обмена информацией между компьютерными сетями или между компьютерами данной сети в 1978 г. Международная организация по стандартизации (англ. **ISO** – International Organization for Standardization) разработала многоуровневый комплект протоколов, известный как семиуровневая эталонная модель взаимодействия открытых систем (англ. **OSI** – open systems interconnection). Она является основой для анализа существующих сетей, создания новых сетей и стандартов. Схема работы данной модели представлена на рисунке 3.1.

Основная идея этой модели заключается в том, что каждому уровню отводится конкретная роль. Благодаря этому общая задача передачи данных разделяется на отдельные легко обозримые задачи. Необходимые соглашения для связи одного уровня с выше- и нижерасположенными называют *протоколом*.

Таким образом, конечные пользователи не должны заботиться о проблемах совместимости, присущих системам, включающим устройства различных производителей. Обеспечение обмена информацией происходит при использовании аппаратных и программных средств, соответствующих стандартам *OSI*, изготовленных разными фирмами.



Рис. 3.1. Семиуровневая модель протоколов взаимодействия открытых систем

Отдельные уровни модели *OSI* удобно рассматривать как группы программ, предназначенных для выполнения конкретных функций. Нумерация уровней протоколов идет снизу вверх, а их названия указаны на рисунке 3.1. Функциональные уровни взаимодействуют на строго иерархической основе: каждый уровень обеспечивает сервис для вышестоящего уровня, запрашивая, в свою очередь, сервис у нижестоящего уровня.

4. Адресация в сети Интернет

Следующим сильным стимулом к развитию Интернета было увеличение размеров сети и связанные с этим проблемы управления. Одной из таких проблем стала адресация в сети. Первоначально существовало ограниченное число узлов, и соответствия адресов и соответствующих им имен узлов можно было хранить в одной таблице или даже запомнить. Запомнить же огромное множество комбинаций IP-адресов уже было невозможно. Для того чтобы упростить использование сети пользователями Интернета, с каждым узлом ассоциировались имена и, таким образом, не было необходимости запоминать

числовые адреса. Появление большого количества независимо управляемых сетей привело к тому, что хранить адреса в одной таблице было невозможно, тогда появилась **система доменных имен**.

Увеличение размеров Интернета потребовало также изменения возможностей маршрутизаторов. В результате изначальный единый алгоритм маршрутизации был заменен иерархической моделью маршрутизации, что определило современную структуру сети Интернет.

Каждый компьютер, подключенный к сети Интернет, имеет уникальный цифровой **IP-адрес**, на основании которого протокол IP передает пакеты в глобальной сети. IP-адрес компьютера состоит из четырех байтов и записывается в виде четырех десятичных чисел, разделенных точками. Примером IP-адреса может служить следующая комбинация: 169.254.100.123.

Человеку неудобно использовать числовые IP-адреса, поэтому создан механизм, ставящий в соответствие IP-адресам символьные имена. Для этой цели используется система **доменных имен** (DNS). Доменный адрес может содержать латинские буквы, цифры, точки и некоторые другие специальные знаки. Доменная адресация возникла в Интернет для удобства пользователей: гораздо легче запомнить доменный адрес (например, www.microsoft.com), чем четыре числа IP-адреса.

Система DNS имеет иерархическую структуру. Составные части отделяются друг от друга точкой. Младшая часть доменного имени соответствует конечному узлу сети.

Совокупность имен, у которых несколько старших частей доменного имени совпадают, называется доменом. Например, имена www.microsoft.com и ftp.microsoft.com принадлежат домену microsoft.com.

Доменные имена назначаются компьютерам, которые постоянно подключены к сети, специальной организацией InterNIC. Вся сеть разбивается на участки по названиям доменов. Самым главным является корневой домен, который управляется InterNIC, далее следуют домены первого, второго и третьего уровней. Домены первого уровня назначаются для каждой страны (по географическому признаку). Например, для России домен первого уровня – ru, для США – us, для Беларуси – by и т.д. Кроме того, несколько имен доменов первого уровня закреплено для различных типов организаций:

- com – коммерческие организации (например, ibm.com);
- edu – образовательные организации (например, spb.edu);
- gov – правительственные организации (например, loc.gov);
- org – некоммерческие организации (например, w3.org);
- net – организации, поддерживающие сети (например, ripn.net).

5. Основные сервисы сети Интернет

На основе протокола TCP/IP разработаны многие сетевые прикладные сервисные протоколы, среди которых следует отметить следующие:

- File Transfer Protocol (FTP) – протокол передачи файлов;
- Telnet (TERminal NETwork) – протокол удаленного доступа, то есть дистанционного исполнения команд на удаленном компьютере;
- Simple Mail Transfer Protocol (SMTP) – простой протокол пересылки электронИнтернетной почты;
- Unix-Unix-CoPy (UUCP) – копирование файлов между двумя компьютерами (изначально под управлением операционной системы UNIX);
- HyperText Transfer Protocol (HTTP) – протокол передачи гипертекста (используется при передаче сообщений в World Wide Web);
- Network News Transfer Protocol (NNTP) – протокол передачи новостей (телеконференций).

Эти протоколы формируют в сети соответствующие прикладные процессы, а задача протокола TCP – обеспечить передачу данных между этими процессами. Одновременно в сети может выполняться несколько процессов.

Рассмотрим наиболее распространенные услуги сети.

Электронная почта

Электронная почта или e-mail принадлежит к числу наиболее популярных прикладных сервисов Интернета. В качестве основного протокола работы с электронной почтой используется протокол прикладного уровня SMTP.

Любое почтовое сообщение состоит из конверта (заголовка) сообщения и тела сообщения. Конверт содержит информацию, необходимую для доставки и обработки сообщения. Тело сообщения содержит информацию, которую отправитель передает получателю. Структура конверта обычно определяется локальным почтовым программным обеспечением. В конверте содержится информация о структуре и составе передаваемых данных, дате формирования сообщения, имени и адресе отправителя, имени и адресе получателя (или группы получателей), теме сообщения и др.

Система новостей

Система новостей USENET или телеконференции – это сервис Интернет, представляющий собой распределенную систему ведения дискуссий, механизм распространения сетевых новостей. В отличие от электронной почты в системе новостей необходимо централизованное хранение статей на USENET-серверах, так как к банку статей должны иметь доступ все пользователи сети, подсоединенные к этому серверу.

В системе телеконференций принцип электронной почты получил дальнейшее развитие. Если в системе электронной почты сообщения адресуются «один к одному» и каждому пользователю предоставляется индивидуальный «почтовый ящик», то в системе телеконференций адресация осуществляется по принципу «один ко всем» и на всех участников телеконференции выделяется один ящик.

Новости разделены по иерархически организованным тематическим группам. Имеются как глобальные иерархии так и иерархии локальные для какой-либо организации, страны или сети. Набор групп, получаемых локальным сервером USENET, определяется администратором этого сервера и наличием этих групп на других серверах, с которыми обмениваются новостями локальный сервер.

Списки рассылки

Списки рассылки или почтовые списки – это практически единственный сервис, не имеющий собственного протокола и работающий только через электронную почту. Идея его организации состоит в следующем: в сети выделяется адрес электронной почты, который является общим адресом многих пользователей-подписчиков определенного списка рассылки. Каждый список рассылки посвящается какой-то конкретной теме. Пользователи-подписчики могут посылать свои сообщения по общему адресу, и тогда эти сообщения рассылаются всем, кто подписался на данный список рассылки. Имеются общедоступные почтовые списки, а некоторые списки имеют ограничения на участие в них. Например, списки рассылки создаются организациями для оповещения своих клиентов, преподавателями и кураторами, организующими дистанционное обучение. Они требуются также, когда обсуждается какой-то вопрос, интересующий слишком малый круг людей, чтобы заводить для него отдельную группу в новостях USENET.

Передача файлов

Это один из первых широко используемых серверов Интернет, в основе которого лежит протокол передачи файлов FTP, обеспечивающий доступ к файлам в файловых архивах.

FTP – сервис прямого доступа, требующий подключения компьютера в сеть Интернет. Однако возможен доступ и через электронную почту, для чего имеются серверы, которые по запросу могут прислать по электронной почте запрашиваемые файлы. Большая часть обычных веб-браузеров может извлекать файлы, расположенные на FTP-серверах. Полностью функциональный FTP-клиент может быть запущен, например, в Firefox как расширение FireFTP.

Служба Telnet

Telnet – это сетевой протокол для реализации текстового интерфейса по сети. Название «Telnet» имеют также некоторые утилиты, реализующие кли-

ентскую часть протокола. Назначение протокола Telnet заключается в том, чтобы позволить терминальным устройствам и терминальным процессам взаимодействовать друг с другом.

Программа Telnet, разработанная для UNIX и поддерживаемая всеми современными версиями Windows, позволяет установить связь с удаленным компьютером и использовать его в интерактивном режиме. То есть программа Telnet позволяет пользователю общаться с удаленным компьютером, как со «своим», и временно получить в свое распоряжение все его ресурсы. Для доступа нужно заранее договориться о получении права на вход (имени и пароля).

Всемирная компьютерная паутина

Всемирная компьютерная паутина (World Wide Web) на сегодняшний день является самым популярным и удобным средством работы с информацией в глобальной сети. В качестве основного протокола системой WWW используется протокол HTTP.

Большая часть документов в WWW хранится в формате HTML. HTML – это язык гипертекстовой разметки, используемый для создания веб-документов. Язык HTML представляет собой набор команд, в соответствии с которыми браузер отображает содержимое документа. В языке HTML реализован механизм гипертекстовых ссылок, который обеспечивает связь одного документа с другими. Эти документы могут находиться на том же сервере, что и страница, с которой на них делается ссылка, а могут быть размещены на другом сервере. Для просмотра веб-документов используются специальные программы-клиенты, называемые браузерами.

Общение пользователей в режиме реального времени

Служба Web Chat (веб-чат) обеспечивает прямое общение пользователей в режиме реального времени. Сообщения можно отправлять как одному конкретному пользователю, так и нескольким участникам беседы. Никаких программ, кроме браузера, запускать на своем компьютере не нужно. Именно поэтому эта технология общения пользуется очень большой популярностью. В отличие от телеконференций, которые открыты всему миру, в чате общение происходит между пользователями в пределах одного канала.

Служба ICQ («интернет-пейджер») предназначена для обмена короткими текстовыми сообщениями между пользователями, одновременно находящимися на связи. Она также выполняет поиск IP-адреса пользователя, подключенного в данный момент к Сети. При каждом подключении пользователя к сети программа ICQ (ICQ-клиент), запускаемая на компьютере, определяет его временный адрес и сообщает его на ICQ-сервер. Каждый пользователь, который уже зарегистрировался на ICQ-сервере, имеет персональный иденти-

фикационный номер UIN (Universal Internet Number), по которому его могут найти другие пользователи.

Важная особенность служб мгновенных сообщений – автоматическое определение присутствия активных пользователей в сети. Основные протоколы служб мгновенных сообщений: IRC, MSN, OSCAR (AIM, ICQ) и другие.

6. Поиск информации в Интернет

Поиск информации – одна из самых востребованных на практике задач, которую приходится решать любому пользователю Интернета. Существуют три основных способа поиска информации в Интернет: указание адреса страницы; передвижение по гиперссылкам; обращение к поисковой системе (поисковому серверу).

Указание адреса страницы – это самый быстрый способ поиска, но его можно использовать только в том случае, если точно известен адрес документа или сайта, где расположен документ. Не стоит забывать возможность поиска по открытой в окне браузера web-странице (Правка-Найти на этой странице...).

Передвижение по гиперссылкам – это наименее удобный способ, так как с его помощью можно искать документы, только близкие по смыслу текущему документу.

Обращение к поисковой системе. Пользуясь гипертекстовыми ссылками, можно бесконечно долго путешествовать в информационном пространстве сети Интернет, переходя от одной web-страницы к другой, но если учесть, что в мире созданы многие миллионы web-страниц, то найти на них нужную информацию таким способом вряд ли удастся. На помощь приходят специальные *поисковые системы* (их еще называют *поисковыми машинами*). Адреса поисковых серверов хорошо известны всем, кто работает в Интернете. В настоящее время в русскоязычной части Интернет популярны следующие поисковые серверы: **Google** (google.ru), **Яндекс** (yandex.ru) и **Rambler** (rambler.ru)

Поисковая система – это веб-сайт, предоставляющий возможность поиска информации в Интернете. Большинство поисковых систем ищут информацию на сайтах Всемирной паутины, но существуют также системы, способные искать файлы на ftp-серверах, товары в интернет-магазинах, а также информацию в группах новостей Usenet. По принципу действия поисковые системы делятся на два типа: *поисковые каталоги* и *поисковые индексы*.

Поисковые каталоги служат для тематического поиска. Информация на этих серверах структурирована по темам и подтемам. Имея намерение осветить какую-то узкую тему, нетрудно найти список web-страниц, ей посвященных. Каталог – это структурированный набор ссылок на сайты с кратким их

описанием. Каталог, в котором ссылки на сайты внутри категорий сортируются по популярности сайтов, называется рейтинг (или топ).

Поисковые индексы работают как алфавитные указатели. Клиент задает слово или группу слов, характеризующих его область поиска, – и получает список ссылок на web-страницы, содержащие указанные термины. Первой поисковой системой для Всемирной паутины был «Wandex», уже не существующий индекс, разработанный в 1993 г.

Поисковые индексы автоматически, при помощи специальных программ, сканируют страницы Интернета и индексируют их, то есть заносят в свою огромную базу данных. Поисковый робот («веб-паук») – программа, являющаяся составной частью поисковой системы и предназначенная для обхода страниц Интернета с целью занесения информации о них (ключевые слова) в базу поисковика. По своей сути паук больше всего напоминает обычный браузер. Он сканирует содержимое страницы, забрасывает его на сервер поисковой машины, которой принадлежит и отправляется по ссылкам на следующие страницы.

В ответ на запрос, где найти нужную информацию, поисковый сервер возвращает список гиперссылок, ведущих к web-страницам, на которых нужная информация имеется или упоминается. Обширность списка может быть любой, в зависимости от содержания запроса.

<http://www.google.ru/> – лидер поисковых машин Интернета. Google занимает более 70% мирового рынка. Сейчас регистрирует ежедневно около 50 млн поисковых запросов и индексирует более 8 млрд веб-страниц. Google может находить информацию на 115 языках.

По одной из версий, Google – искаженное написание английского слова googol. «Googol (гугол)» – это математический термин, обозначающий единицу со 100 нулями. Использование этого термина компанией Google отражает задачу организовать огромные объемы информации в Интернете. Интерфейс Google содержит довольно сложный язык запросов, позволяющий ограничить область поиска отдельными доменами, языками, типами файлов и т. д.

<http://www.yandex.ru/> – Яндекс – российская система поиска в Сети.

Слово «Яндекс» (состоящее из буквы «Я» и части слова index; обыгран тот факт, что русское местоимение «Я» соответствует английскому «I»). Поиск Яндекса позволяет искать по Рунету документы на русском, украинском, белорусском, румынском, английском, немецком и французском языках с учетом морфологии русского и английского языков и близости слов в предложении. Отличительная особенность Яндекса – возможность точной настройки поискового запроса. Это реализовано за счет гибкого языка запросов.

Время от времени алгоритмы, отвечающие за релевантность выдачи результата запроса, меняются, что приводит к изменениям в результатах поисковых запросов. В частности, эти изменения направлены против поискового спама, приводящего к нерелевантным результатам по некоторым запросам.

<http://www.rambler.ru/> – Rambler Media Group – интернет-холдинг, включающий в качестве сервисов поисковую систему, рейтинг-классификатор ресурсов российского Интернета, информационный портал.

Rambler создан в 1996 г. Поисковая система Рамблер понимает и различает слова русского, английского и украинского языков. По умолчанию поиск ведется по всем формам слова.

7. Перспективные технологии на основе Интернета

IP-телефония

Под IP-телефонией понимается технология, позволяющая использовать Интернет или любую другую IP-сеть в качестве средства организации и ведения телефонных разговоров и передачи факсов в режиме реального времени. Существует техническая возможность оцифровать звук или факсимильное сообщение и переслать его аналогично тому, как пересылаются цифровые данные. И в этом смысле IP-телефония использует Интернет (или любую другую IP-сеть) для пересылки голосовых или факсимильных сообщений между двумя пользователями компьютера в режиме реального времени.

Общий принцип действия телефонных серверов IP-телефонии таков: с одной стороны, сервер связан с телефонными линиями и может соединиться с любым телефоном мира. С другой стороны, сервер связан с Интернетом и может связаться с любым компьютером в мире. Сервер принимает стандартный телефонный сигнал, оцифровывает его (если он исходно не цифровой), значительно сжимает, разбивает на пакеты и отправляет через Интернет по назначению с использованием протокола TCP/IP. Для пакетов, приходящих из Сети на телефонный сервер и уходящих в телефонную линию, операция происходит в обратном порядке. Обе составляющие операции (вход сигнала в телефонную сеть и его выход из телефонной сети) происходят в режиме реального времени.

Для того чтобы осуществить связь с помощью телефонных серверов, организация или оператор услуги должны иметь серверы в тех местах, куда и откуда планируются звонки. Стоимость такой связи на порядок меньше стоимости телефонного звонка по обычным телефонным линиям. Особенно велика эта разница для международных переговоров.

Электронная коммерция

Электронная коммерция подразумевает использование технологий глобальных компьютерных сетей для ведения бизнеса. Благодаря широкому распространению технологии World Wide Web в течение последних лет сеть Интернет из академической сети превратилась в популярную среду для общения,

рекламы и бизнеса. Хотя электронная коммерция существовала и ранее, но именно популярность и доступность Интернета сделала возможным широкое использование электронной коммерции.

Электронная коммерция позволяет изменить практически все процессы, происходящие в современном бизнесе, интегрируя их в единое целое. Потребители могут искать, заказывать и оплачивать товары, используя Интернет, обмениваться информацией о товарах и услугах с другими пользователями. Правительственные организации могут использовать Интернет для сбора налоговых деклараций и распространения официальной информации.

Системы электронных платежей

Существуют две основные области применения систем электронных платежей – это обмен данными (заказы, счета и др.) и электронный перевод денежных средств, который осуществляется между банками и имеет достаточно большой объем и относительно небольшое количество передаваемых полей, а также долгосрочные связи.

Электронные платежные средства. Существует три типа электронных платежных средств. Первый тип представляет собой комбинацию обычных и электронных платежей. Например, платеж осуществляется традиционным способом, а подтверждение владельцу высылается по электронной почте. Или, напротив, платеж делается электронным способом, а подтверждение приходит по обычной почте. Второй тип является расширением традиционного способа передачи денежных средств. Сюда входит передача номеров кредитных карточек электронным способом и использование так называемых смарт-карт, которые могут хранить и передавать всю информацию о своем владельце. В данном случае вся операция может происходить в электронном виде.

Третий тип включает различные виды цифровой наличности и электронных денег. Разница между первыми двумя типами и третьим заключается в том, что в последнем случае действительно переводятся деньги, а не только информация о них. Например, если передается только номер кредитной карточки – это платеж второго типа, а если сообщение само несет в себе некую сумму денег – это платеж третьего типа.

Электронные деньги – это некоторые зашифрованные серийные номера, представляющие реальную сумму денег. В то же время они являются вполне полноценными денежными средствами в том смысле, что могут быть обменяны на обычные деньги.

Смарт-карты являются небольшими устройствами, которые могут хранить информацию, необходимую для совершения транзакций в рамках электронной коммерции. Смарт-карты могут хранить электронную наличность, информацию о владельце, электронные ключи и другую информацию.

Микроплатежи – это особый вид электронных денег, необходимый для оплаты микротранзакций. Стоимость одной транзакции может быть очень мала, и вследствие этого ее может быть невозможно выразить в обычных денежных единицах, поэтому возникла необходимость в микроплатежах. Важно отметить, что микро платежи являются особенностью электронной коммерции.

8. Основы разработки веб-документов. Язык HTML

Для создания веб-страниц разработан стандартный язык разметки гипертекста – HTML (Hyper Text Markup Language). Подчеркнем, что HTML не является языком программирования. Изначально предполагалось, что веб-документы могут содержать текстовую и графическую информацию вместе с элементами управления, такими, как ссылки на другие документы. Впоследствии появилась возможность включать в документы аудио- и видеофрагменты.

Создать веб-документ, можно в обычном текстовом редакторе (например, в Блокноте), сохранив его в файле с расширением `htm` или `html`, а просмотреть – в любом браузере.

Инструкции HTML называют *тегами*, или *метками* (иногда дескрипторами). Они выделяются угловыми скобками (`<` и `>`). Каждый тег имеет свое название, являющееся ключевым словом. Спецификация тега обеспечивает его параметрами, которые обычно называют *атрибутами*. Атрибуты могут иметь значения (аргументы). Названия тегов и атрибуты можно писать в любом регистре, однако, желательно использовать строчные буквы (согласно стандарту XHTML). Атрибуты, если они имеются, пишутся за названием тега в произвольном порядке через пробел. Формат тега имеет следующий вид:

`<НАЗВАНИЕ_ТЕГА [АТРИБУТ1 [= значение1] ... [АТРИБУТ N [= значение N]] >`

Квадратные скобки показывают, что заключенные в них элементы не обязательны. Существуют теги без атрибутов (например, тег перехода на новую строку `<BR>`), а также атрибуты без аргументов (например, `NORESIZE` в теге `<FRAME>`).

Даже если для тега предусмотрены атрибуты, во многих случаях их можно не указывать, используя значение по умолчанию.

Атрибут `ID` присущ любому тегу, но если он не используется, то по умолчанию принимается, что его значение — пустая строка.

Пример. Тег `<IMG SRC = "picture.jpg">` дает браузеру команду вывести на экран графический объект из файла `picture.jpg`. Здесь ключевое слово `IMG` – название тега, `SRC` – атрибут, а `"picture.jpg"` – аргумент (значение).

Теги можно писать в одной строке или в нескольких, делая переносы в любом месте. Невидимый служебный символ перевода строки и возврата каретки, вставляемый в редакторе при нажатии клавиши Enter, не воспринимается браузером как разделитель тегов. Можно считать, что браузер воспринимает последовательность тегов как единую строку символов.

Большинство тегов являются *контейнерными*. Это означает следующее.

1. Начальному тегу <ТЕГ...> обязательно соответствует заключительный (конечный) тег </ТЕГ>.

2. Внутри контейнерного тега можно разместить другие теги.

В связи с этим можно говорить о вложенности одних тегов в другие.

Например, теги тела документа <BODY>, ссылки <A...>, раздела <DIV>, таблицы <TABLE> и др. являются контейнерными, то есть им соответствуют заключительные теги </BODY>, , </DIV>, </TABLE>. Говоря «внутри тега <ТЕГ>», мы имеем в виду то, что расположено между тегами <ТЕГ> и </ТЕГ>. В приведенном ниже примере показано, что внутри тега ссылки <A> можно разместить тег графического изображения , создав таким образом графическую ссылку:

```
<A HREF = "yandex.ru"><IMG SRC = "banner.gif"></A>
```

В HTML-коде существуют теги, которые никак не проявляются в окне браузера какими бы то ни было визуальными эффектами (например, <HEAD>, <META>, <SCRIPT>). Вместе с тем существует множество тегов, которым соответствуют определенные видимые элементы документа (веб-страницы). Например, тег

 соответствует изображению, Тегу <BUTTON> – кнопка, тегу <INPUT> – поле ввода данных, переключатель или кнопка, в зависимости от значения атрибута TYPE.

Пример

```
<HTML>
<H1>Моя веб-страница</H1>
<IMG SRC="pict.jpg"> ,
<A HREF = "msu. mogilev.by">Сайт университета</a>
<FORM>
<INPUT TYPE="text" VALUE="">
<p>
<BUTTON>Нажми здесь</ BUTTON >
</FORM>
</HTML>
```

Приведенный HTML-код формирует веб-страницу (рис. 3.2), содержащую заголовок первого уровня (<H1>), изображение (), ссылку (<A>) и форму (<FORM>), которая содержит поле ввода данных (<INPUT>) и кнопку (<BUTTON>).

Моя веб-страница



[Сайт университета](#)

Нажми здесь

Рис. 3.2. Пример размещения объектов на веб-странице

С помощью специальных тегов можно форматировать текстовые данные: управлять шрифтом, задавать абзацы и способ выравнивания, в том числе взаимное расположение текста и изображений.

Если не использовать специальных средств позиционирования, то элементы в документе будут располагаться в порядке появления в HTML-коде соответствующих им тегов. При этом если не использовать теги перевода строки (`<BR>`) или абзаца (`<P>`), то элементы будут располагаться рядом друг с другом по горизонтали или по вертикали, в зависимости от размеров окна браузера и их собственных размеров.

Теги `<BR>` и `<P>` являются довольно слабыми средствами форматирования.

Один из часто применяемых способов разметки документа основан на использовании группы тегов, определяющих **таблицу** (`<TABLE>`, `<TR>`, `<TD>` и др.). Таблицы можно создавать не обязательно с видимыми границами (рамками). С помощью таблицы можно просто разбить все поле документа на прямоугольные ячейки одинаковых или различных размеров и размещать в них нужные элементы (тексты, изображения, кнопки, ссылки и т. п.). Во многих случаях этого способа разметки вполне достаточно.

Для группирования элементов уровня блока и текста используется тэг `<div>...</div>`.

Его атрибуты:

`align` – задает выравнивание содержимого тега `<div>`.

`title` – добавляет всплывающую подсказку к содержимому.

Также для этого тега доступны универсальные атрибуты и события. Закрывающий тег обязателен.

Структура HTML-документа

Пример HTML-документа:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
<html>
<head>
<title>Приветик</title>
</head>
<body>
<p align="center">
Всем привет!</p>
</body>
</html>
```

Первая строка (она необязательна, но желательна) показывает версию языка HTML, которой пользуется автор документа.

HTML-код, формирующий веб-документ, начинается тегом <HTML> и заканчивается тегом </HTML>. Таким образом, <HTML> является контейнерным тегом, причем самого верхнего уровня. Между тэгами <html> и </html> размещают собственно HTML-документ. Он состоит из двух разделов – **заголовка** (между тэгами <head> и </head>) и **тела** (между тэгами <body> и </body>) (в некоторых случаях раздел body не используется и может быть использован раздел frameset).

Все, что находится между парой меток <title> ... </title>, толкуется браузером как название документа. Обычно браузер показывает название текущего документа в заголовке окна и печатает его в левом нижнем углу каждой страницы при выводе на принтер. К тому же в браузере это интерпретируется как название закладки по умолчанию.

В теле веб-документа метки вида <H_i> (где i – цифра от 1 до 6) описывают **заголовки** шести различных уровней. Заголовок первого уровня – самый крупный, шестого уровня – самый мелкий.

Все, что заключено между <P> и </P>, воспринимается как один **абзац**.

Метки <H_i> и <P> могут содержать дополнительный атрибут выравнивания ALIGN, который может иметь одно из четырех значений:

- CENTER – выравнивание по центру;
- RIGHT – выравнивание по правому краю;
- LEFT – выравнивание по левому краю;
- JUSTIFY – выравнивание по ширине (по обоим краям).

Например:

<H1 ALIGN=CENTER>Выравнивание заголовка по центру</H1>

или

<P ALIGN=RIGHT>Образец абзаца с выравниванием по правому краю</P>

Метка
 используется, если необходимо перейти на **новую строку**, не прерывая абзаца.

Метка <HR> описывает **горизонтальную линию**, которая позволяет делить информацию страницы на разделы. Эта метка может дополнительно включать атрибуты SIZE (определяет толщину линии в пикселях) и/или WIDTH (определяет размах линии в процентах от ширины экрана).

Форматирование шрифта

Тэги выделения фрагментов текста позволяют управлять отображением отдельных символов и слов. Обязательны как открывающий, так и закрывающий тэг.

<tt> Моноширинный шрифт

<i> Курсив

 Полужирный шрифт

<big> Увеличенный шрифт

<small> Уменьшенный шрифт

<strike> Зачеркнутый шрифт

<s> Зачеркнутый шрифт

<u> Подчеркнутый шрифт

<sup> Верхний индекс

<sub> Нижний индекс

Тэг предоставляет возможности управления размером (size), начертанием (face) и цветом (color) текста.

Пример, .

Диапазон установки размера текущего шрифта – от 1 до 7.

Примеры записи цвета:

Цвет		Цвет	
черный	black	фиолетовый	purple
белый	white	желтый	yellow
красный	red	коричневый	brown
зеленый	green	оранжевый	orange
бирюзовый	azure	лиловый	violet
синий	blue	серый	gray

Форматированный текст: <PRE> ... </PRE>

Браузеры игнорируют множественные пробелы и символы конца строки. Однако, текст, заключенный между метками <PRE> и </PRE> (preformatted – предварительно форматированный), выводится браузером на экран как есть – со всеми пробелами, символами табуляции и конца строки. Это можно использовать при создании таблиц.

Списки

Текст, расположенный между метками и , воспринимается как **маркированный** список. Каждый новый элемент списка следует начинать с метки . Обратите внимание, что у метки нет парной закрывающей метки. Например, чтобы создать вот такой список:

Алеша Попович;
Добрыня Никитич;
Илья Муромец

необходим вот такой HTML-код:

```
<li>  
  <li> Алеша Попович;  
  <li> Добрыня Никитич;  
  <li> Илья Муромец  
  <li>
```

Для задания вида маркера в метке можно использовать атрибут **type**. Его возможные значения: circle (окружность), disk (круг), square (квадрат).

Текст, расположенный между метками и , воспринимается как **нумерованный** список. Например:

```
<ol>  
  <li> Алеша Попович;  
  <li> Добрыня Никитич;  
  <li> Илья Муромец  
</ol>
```

получится вот такой список:

1. Алеша Попович;
2. Добрыня Никитич;
3. Илья Муромец

Для тэга можно использовать атрибуты **type** для задания типа списка и **start**, задающий начальный номер списка:

```
<ol type=A>  
<ol type=a>
```

```
<ol type=I>  
<ol type=1 start=3>
```

Элемент любого списка может содержать в себе целый список любого вида. Так можно создавать **многоуровневые списки**. Число уровней вложенности в принципе не ограничено, однако злоупотреблять вложенными списками не следует.

Гиперссылки

В отличие от обыкновенного текста, который можно читать только от начала к концу, гипертекст позволяет осуществлять мгновенный переход от одного фрагмента текста к другому. Системы помощи многих программных продуктов устроены именно по гипертекстовому принципу.

В HTML переход от одного фрагмента текста к другому задается с помощью метки вида:

```
<A HREF="[адрес перехода]">выделенный фрагмент текста</A>
```

В качестве параметра [адрес перехода] может использоваться несколько типов аргументов. Самое простое — это задать имя другого HTML-документа, к которому нужно перейти. Например:

```
<A HREF="index.html">Перейти к оглавлению</A>
```

Такой фрагмент HTML-текста приведет к появлению в документе выделенного фрагмента [Перейти к оглавлению](#), при нажатии на который в текущее окно будет загружен документ `index.html`.

При этом, если в адресе перехода не указан каталог, переход будет выполнен внутри текущего каталога. Если в адресе перехода не указан сервер, переход будет выполнен на текущем сервере. На практике часто бывает необходимо дать ссылку на документ, находящийся на другом сервере. В этом случае следует указать полный путь к документу – URL-адрес.

При необходимости можно задать переход не просто к некоторому документу, но и к определенному месту внутри этого документа. Для этого необходимо создать в документе, к которому будет задан переход, некоторую опорную точку, или **анкер**.

Допустим, что необходимо осуществить переход из файла `1.html` к словам “Переход закончен” в файле `2.html` (файлы находятся в одном каталоге). Прежде всего, необходимо создать вот такой анкер в файле `2.html`:

```
<A NAME="AAA">Переход закончен</A>
```

Слова “Переход закончен” при этом никак не будут выделены в тексте документа.

Затем в файле `1.html` (или в любом другом) можно определить переход на этот анкер:

```
<A HREF="2.html#AAA">Переход к анкеру AAA</A>
```

Переход к этому анкеру можно определить и внутри самого документа 2.html — достаточно только включить в него вот такой фрагмент:

Переход к анкеру AAA

На практике это очень удобно при создании больших документов. В начале документа можно поместить оглавление, состоящее из ссылок на анкеры, расположенные в заголовках разделов документа.

Во избежание недоразумений рекомендуется задавать имена анкером латинскими буквами. Большинство браузеров отличают большие буквы от маленьких. То есть, если имя анкера определено как AAA, ссылка на анкер aaa или AaA не выведет Вас на анкер AAA, хотя документ, скорее всего, будет загружен корректно.

Возможны ссылки и на другие виды ресурсов:

Отправить письмо

Если пользователь совершит переход по такой ссылке, у него на экране откроется окно ввода исходящего сообщения его почтовой программы. В строке To: ("Куда") окна почтовой программы будет указано user@mail.box.

Выгрузить файл

Такая ссылка, если ей воспользоваться, запустит протокол передачи файлов и начнет выгрузку файла file.ext, находящегося в каталоге directory на сервере server, на локальный диск пользователя.

Изображения в HTML-документе

Тэг позволяет вставить изображение в документ. Изображение появится в том месте документа, в котором записан этот тэг. Команда записывается с одиночным (непарным) тэгом.

Допустим, нам нужно включить в документ изображение, записанное в файл picture.gif, находящийся в одной папке с HTML-документом:

Тэг может иметь дополнительные атрибуты, указанные в таблице 3.1.

Таблица 3.1 – Атрибуты тега IMG

Атрибут	Формат	Описание
ALT		Если браузер не воспринимает изображение, вместо него появляется заменяющий текст.
BORDER		Задаёт толщину рамки вокруг изображения. Измеряется в пикселях.
ALIGN		Выравнивает изображение относительно текста: по верхней части изображения – TOP, по нижней – BOTTOM, по средней – MIDDLE.

Атрибут	Формат	Описание
HEIGHT		Задаёт вертикальный размер изображения внутри окна браузера.
WIDTH		Задаёт горизонтальный размер изображения внутри окна браузера.
VSPACE		Добавляет верхнее и нижнее пустые поля.
HSPACE		Добавляет левое и правое пустые поля.

Если ссылка на изображение находится между метками и , изображение фактически становится кнопкой, при нажатии на которую происходит переход по ссылке.

Цветовая гамма HTML-документа

Цветовая гамма HTML-документа определяется атрибутами, размещёнными внутри метки <BODY>:

`bgscolor` – определяет цвет фона документа.

`text` – определяет цвет текста документа.

`link` – определяет цвет выделенного элемента текста, при нажатии на который происходит переход по гипертекстовой ссылке.

`vlink` – определяет цвет ссылки на документ, который уже был просмотрен ранее.

`alink` – определяет цвет ссылки в момент, когда на нее указывает курсор мыши и нажата ее правая кнопка, то есть непосредственно перед переходом по ссылке.

Цвет кодируется последовательностью из трех пар символов. Каждая пара представляет собой шестнадцатеричное значение насыщенности заданного цвета одним из трех основных цветов (красным, зеленым и синим) в диапазоне от нуля (00) до 255 (FF).

Примеры:

`bgscolor=#FFFFFF`

Цвет фона. Насыщенность красным, зеленым и синим одинакова – FF (это шестнадцатеричное представление числа 255). Результат — белый цвет.

`text=#000000`

Цвет текста. Насыщенность красным, зеленым и синим одинакова – 00 (ноль). Результат – черный цвет.

`link=#FF0000`

Цвет гиперссылки. Насыщенность красным – FF (255), зеленым и синим – 00 (ноль). Результат – красный цвет.

Кроме того, метка <BODY> может включать атрибут background=" [имя файла] ", который задает изображение, служащее фоном для текста и других изображений. Как и любое другое изображение, фон должен быть представлен в формате GIF или JPEG. Браузеры заполняют множественными копиями изображения-фона все пространство окна, в котором открыт документ.

Цвет фона и изображение-фона никак не отображаются на бумаге при выводе HTML-документа на печать.

Таблицы

Таблица начинается с метки <table> и заканчивается меткой </table>.

Метка <table> может включать несколько атрибутов.

Таблица может иметь заголовок (<caption> ... </caption>), хотя он не является обязательным. Метка <caption> может включать атрибут align. допустимые значения: <caption align=top> (заголовок помещается над таблицей) и <caption align=bottom> (заголовок помещается под таблицей).

Каждая строка таблицы начинается с метки <TR> и заканчивается меткой </TR>. Метка <TR> может включать дополнительные атрибуты.

Каждая ячейка таблицы начинается с метки <td> и заканчивается меткой </td>.

Если ячейка таблицы пуста, вокруг нее не рисуется рамка. Если ячейка пуста, а рамка нужна, в ячейку можно ввести символьный объект (non-breaking space – неразрывающий пробел).

Любая ячейка таблицы может содержать в себе другую таблицу.

&-последовательности

Поскольку символы "<" и ">" воспринимаются браузерами как начало и конец HTML-меток, возникает вопрос, как показать эти символы на экране. В HTML это делается с помощью &-последовательностей (эскейп-последовательностей).

<	<
>	>
& (амперсанд)	&

Двойные кавычки ("") кодируются последовательностью ";

Точка с запятой – обязательный элемент &-последовательности. Кроме того, все буквы, составляющие последовательность, должны быть в нижнем регистре (т.е., маленькие).

Вообще говоря, &-последовательности определены для всех символов из второй половины ASCII-таблицы, куда входят и русские буквы.

Комментарии

Браузеры игнорируют любой текст, помещенный между `<!--` и `-->`. Это удобно для размещения комментариев. `<!--` Это комментарий `-->`

9. Каскадные таблицы стилей

Вскоре после появления HTML выяснилось, что он не вполне отвечает потребностям разработчиков и пользователей сайтов. Основные требования были следующими.

1. Нужны были более мощные и гибкие средства форматирования.
2. Требовалась возможность вставки в HTML-документ объектов сторонних производителей.
3. Требовались средства для управления содержимым HTML-документа, загруженного в браузер, а также для обработки действий пользователя (манипуляции мышью, нажатия на клавиши).

Иначе говоря, было необходимо, чтобы HTML-документ являлся динамическим и интерактивным, мог взаимодействовать с пользователем. В результате появились так называемые каскадные таблицы стилей – CSS (Cascading Style Sheets).

CSS – это язык стилей, определяющий отображение HTML-документов. Например, CSS работает со шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими другими вещами.

Таким образом, если HTML используется для структурирования содержимого страницы. CSS используется для форматирования этого структурированного содержимого.

Конкретные преимущества CSS:

- управление отображением множества документов с помощью одной таблицы стилей;
- более точный контроль над внешним видом страниц;
- различные представления для разных носителей информации (экран, печать, и т. д.);
- сложная и проработанная техника дизайна.

С помощью стилей можно определить параметры форматирования текстов, задать фильтры (визуальные эффекты) для текстовых и графических элементов, а также задать три координаты позиционирования элементов.

Пусть нам нужен красный цвет фона web-страницы:

В HTML это можно сделать так:

```
<body bgcolor="#FF0000">
```

С помощью CSS того же самого результата можно добиться так:

```
body {background-color: #FF0000;}
```

Как видно, эти коды более или менее идентичны в HTML и CSS. Этот пример также демонстрирует фундаментальную модель CSS:

```
selector {property: value;}
```

↑ ↑ ↙

selector		value/значение:
/селектор:		Значением свойства
К какому		background-color
HTML-тэгу		может быть, например
(тэгам)		red ("#FF0000")
применяется	property	
свойство	/свойство:	Свойство, которое, к примеру,
(например,		может быть цветом фона
"body")		("background-color")

10. Развитие языка HTML. XML и XHTML-документы

XML (eXtensible Markup Language) – это расширяемый язык разметки. В XML основное внимание сосредоточено на данных.

XML-документ представляет собой обычный текстовый файл, в котором при помощи специальных маркеров создаются элементы данных, последовательность и вложенность которых определяет структуру документа и его содержание. XML позволяет расширять множество тегов. Вместо вставки специальных тегов или комментариев, объясняющих назначение конкретного поля, имя поля само может быть информативным как для приложений, так и для человека.

Формат HTML является стандартом верстки документов оригинального образа, коими являются макеты сайтов. XML лучше подходит для хранения данных сложной структуры и различных манипуляций, связанных с импортом/экспортом данных, между различными программами и системами управления базами данных, у которых нет прямой совместимости. XML позволяет выполнить экспорт правил и накопленных ранее данных посредством формирования специального текстового файла (файлов) даже в другую операционную систему.

Языки HTML и XML предназначены для решения разных задач. Учитывая прекрасную структуру XML и место, реально занимаемое HTML в практике веб-дизайна, было решено объединить HTML и XML. В результате появился XHTML. В отличие от HTML, XHTML – это язык разметки, который базируется на основе XML. Появление XHTML означает, что HTML переопределяется как приложение XML. Теперь HTML должен подчиняться правилам XML.

11. Сценарии

Сценариями называют программы, работающие с объектами HTML-документа. Сценарии можно писать непосредственно в HTML-документе, а также в отдельных текстовых файлах, которые вызываются из HTML-документа. Проще всего размещать сценарий непосредственно в HTML-документе.

Размещение сценария в отдельном файле не сложно, но не всегда оправданно. Необходимо накопить достаточное количество программных заготовок, чтобы использовать их в последующих проектах в качестве библиотек.

Сценарий в HTML-документе можно разместить несколькими способами. Стандартным является размещение сценария в контейнерном теге `<SCRIPT>`, то есть между тегами `<SCRIPT>` и `</SCRIPT>`. Встречая тег `<SCRIPT>`, браузер “понимает”, что за ним начинается код сценария. Заключительный тег `</SCRIPT>` указывает браузеру, что код сценария закончился. Все, что находится вне этих тегов, браузер воспринимает как HTML-код. Контейнер `<SCRIPT>` может располагаться в любом месте HTML-документа и даже не один раз. От его расположения иногда может зависеть функционирование всего HTML-документа. Контейнерный тег `<SCRIPT>`, объемлющий код сценария, может содержать следующие атрибуты.

- **LANGUAGE** – язык сценария; возможные значения:
«JavaScript», «JScript»;
«VBScript», «VBS».

Если атрибут **LANGUAGE** не указан, то в Internet Explorer подразумевается JScript.

- **SRC** – указывает файл (имя или URL-адрес), содержащий код сценария. Этот атрибут используется в том случае, если сценарий расположен не в HTML-документе, а в отдельном файле.

Примеры

```
<SCRIPT LANGUAGE="JavaScript">
```

```
... // код сценария
```

```
</SCRIPT>
```

```
<SCRIPT LANGUAGE="JScript" SRC="myscripts.js"></SCRIPT>
```

Современные браузеры распознают еще и версию языка.

Если сценарий располагается в отдельном файле, то в нем теги `<SCRIPT>` и `</SCRIPT>` не пишутся. Файлы со сценариями на JavaScript являются обычными текстовыми файлами с расширением `.js`. В отдельных файлах обычно размещают библиотеки функций (определения функций), а также сценарии, использующиеся в нескольких HTML-документах одного или нескольких

сайтов. Сценарий, загруженный из внешнего файла, можно представить себе просто как его вставку в HTML-документ.

Вот пример:

```
<HTML>
...
<SCRIPT>
function myfunc(){
...
}
</SCRIPT>
<SCRIPT SRC = «mylibrary1.js»></SCRIPT>
<SCRIPT SRC = «myprogram.js»></SCRIPT>
...

</HTML>
```

Здесь в HTML-документе расположены три раздела (секции) сценария. В первом разделе сценария дано определение некоторой функции, два других загружаются из отдельных файлов.

12. Этапы и методы разработки сайтов

Можно выделить следующие основные этапы разработки веб-сайтов:

- 1) определение тематики сайта, его целей и задач;
- 2) проектирование структуры сайта, определение разделов и связей между страницами;
- 3) разработка дизайна сайта, т.е. стиля оформления страниц;
- 4) подготовка материалов (текстов и графики);
- 5) конструирование страниц сайта (создание HTML-кода);
- 6) размещение в сети (публикация);
- 7) тестирование сайта.

Выделяют две основные группы методов конструирования (разработки) сайтов: визуальные и программные. Визуальные не требуют знания языка HTML.

Для визуальной разработки сайтов используют специальные программные средства. Так в состав пакета Microsoft Office входит приложение, предназначенное для разработки веб-сайтов – Front Page. Одним из наиболее удачных приложений для разработки веб-сайтов является Adobe Dreamweaver.

Кроме того, используют специальные языки веб-программирования. Например, язык сценариев Java Script. Написанные на нем конструкции (скрипты) вставляются непосредственно на веб-страницу и интерпретируются браузером. Они используются для создания интерактивных элементов веб-страниц. Например, динамических меню, часов, календарей, систем голосования, счетчиков посещений и т.д.

Для серверного программирования следующие используют языки:

PHP – препроцессор гипертекста,

PERL – практический язык для извлечения данных и составления отчетов.

Глава 4

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

1. Базы данных и СУБД

Везде, где в процессе работы получается большое количество данных, возникает проблема их упорядочивания и управления ими. Наиболее эффективным способом упорядочивания данных являются *базы данных*.

Фактически база данных, наполненная информацией, является файлом. Информация, хранимая в базах данных, используется в качестве исходных данных при решении различных задач.

База данных (БД) – это совокупность данных, организованная в соответствии с определенными правилами и поддерживаемая в памяти компьютера.

База знаний – это БД, разработанная для оперирования знаниями (метаданными). База знаний содержит структурированную информацию, покрывающую некоторую область знаний. Современные базы знаний работают совместно с системами поиска информации, имеют классификационную структуру и формат представления знаний.

Система управления базой данных (СУБД) представляет собой совокупность программного обеспечения, необходимого для создания и использования баз данных. Основное назначение СУБД – обеспечить быстрый доступ к нужной информации.

Каждая БД и СУБД строится на основе некоторой явной или неявной **модели данных**. Используют следующие модели данных: *иерархическую* и *сетевую* (с 60-х гг.) и *реляционную* (с 70-х). Основное различие данных моделей в представлении взаимосвязей между объектами.

Иерархическая модель данных строится по принципу иерархии объектов, то есть один объект является главным (его называют *корневым*), все ниже лежащие – подчиненными. Устанавливается связь «один ко многим», то есть для некоторого главного объекта существует несколько подчиненных объектов. У подчиненных объектов могут быть в свою очередь другие подчиненные объекты. Пример иерархически организованной структуры данных представлен на рисунке 4.1. По принципу иерархической БД построены иерархические файловые системы и реестр ОС Windows.

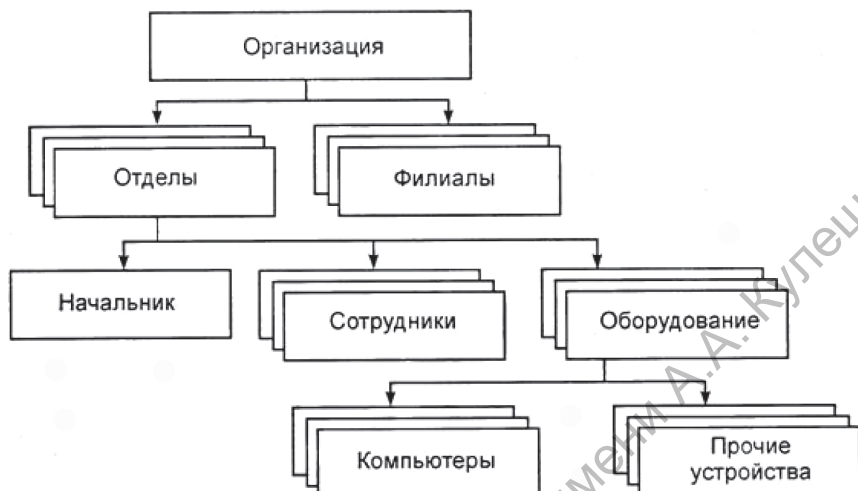


Рис. 4.1. Пример иерархии данных

Сетевая модель данных является расширением иерархического подхода. В сетевой модели любой объект может порождать несколько подчиненных объектов и при этом быть подчиненным для нескольких главных объектов. В сетевой модели возможны связи всех информационных объектов со всеми. Например, каждый преподаватель может обучать много студентов и каждый студент может обучаться у многих преподавателей (рис. 4.2).

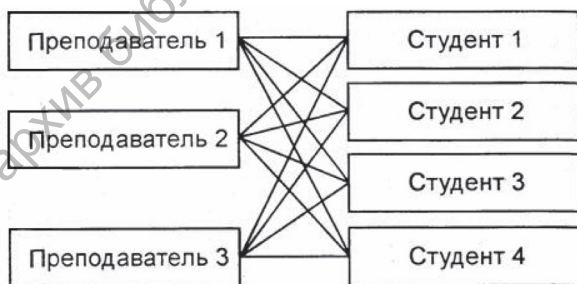


Рис. 4.2. Пример сетевой организации данных

Все современные СУБД поддерживают реляционную модель данных (РМД). **Реляционная модель данных** основана на математическом понятии «отношение» (relation). В качестве неформального синонима термину «отношение» часто используется слово таблица. Все строки, составляющие таблицу в реляционной базе данных, должны иметь первичный ключ.

Таким образом, в реляционной модели данные организованы в виде совокупности таблиц, между которыми установлены связи. Предполагается, что данные об объектах представляются в виде наборов, состоящих из записей. Структура всех записей в наборе одинакова. Элементы данных, из которых состоит каждая запись, называются полями. Кроме того, в состав реляционной модели данных включают теорию нормализации.

Таблица отражает объект реального мира – сущность, а каждая ее строка отражает конкретный экземпляр сущности. Каждый столбец имеет уникальное для таблицы имя. Строки не имеют имен, порядок их следования не определен, а количество логически не ограничено. Одним из основных преимуществ РМД является однородность (каждая строка таблицы имеет один формат). Пользователь сам решает вопрос, обладают ли соответствующие сущности однородностью. Этим решается проблема пригодности модели. Основные элементы РМД показаны на рис. 4.3.

Отношение представляет собой двумерную таблицу, содержащую некоторые данные. Сущность – объект любой природы, данные о котором хранятся в БД. Атрибуты – свойства, характеризующие сущность (столбцы). Степень отношения – количество столбцов. Схема отношения – список имен атрибутов, например, *СОТРУДНИК (№, ФИО, Год рождения, Должность, Кафедра)*. Домен (поле) – совокупность значений атрибутов отношения (тип данных). Кортеж (запись) – строка таблицы. Кардинальность (мощность) – количество строк в таблице.

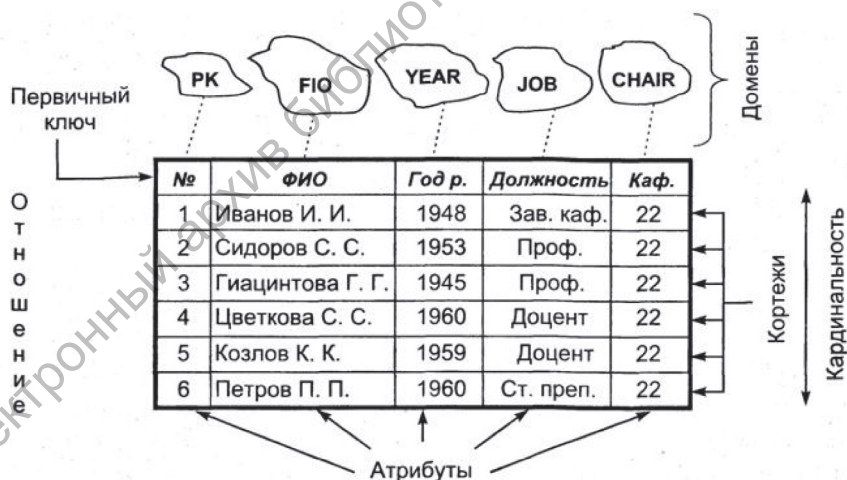


Рис. 4.3. Элементы реляционной модели

Первичный ключ – это атрибут, уникально идентифицирующий строки отношения. Первичный ключ из нескольких атрибутов называется составным.

Первичный ключ не может быть полностью или частично пустым (иметь значение null). Ключи, которые можно использовать в качестве первичных, называются *потенциальными* или *альтернативными* ключами. *Внешний ключ* – это атрибут (атрибуты) одной таблицы, который может служить первичным ключом другой таблицы. Он является ссылкой на первичный ключ другой таблицы (рис. 4.4).

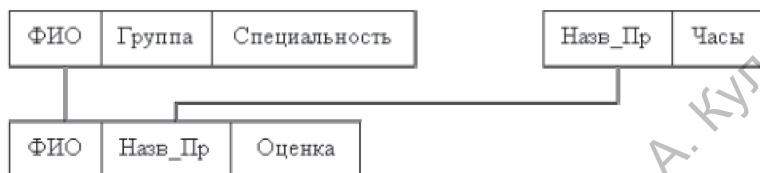


Рис. 4.4. Связь отношений

Отношения *СТУДЕНТ* (*ФИО, Группа, Специальность*) и *ПРЕДМЕТ* (*Назв_Пр, Часы*) связаны отношением *СТУДЕНТ_ПРЕДМЕТ* (*ФИО, Назв_Пр, Оценка*), в котором внешние ключи *ФИО* и *Назв_Пр* образуют составной ключ.

В настоящее время разработано множество разных СУБД. Все они могут быть разделены по способу доступа к базам данных на две большие группы: *настольные* и *серверные*.

Настольные СУБД ориентированы на обслуживание одного пользователя, работающего на определенном компьютере с базами данных в каждый реальный момент времени. К настольным СУБД относят Microsoft Access, Paradox, dBase, FoxPro.

Недостатки настольных СУБД стали очевидными, когда возникла необходимость одновременной работы с ними большого числа пользователей. Поэтому следующим этапом развития СУБД стало создание серверных СУБД.

Серверные СУБД используют принцип централизованного хранения и обработки данных, который основан на архитектуре клиент-сервер. СУБД, хранящая данные, и прикладная программа, интерпретирующая эти данные, являются разными приложениями, т. е. существуют приложение-сервер и приложение-клиент. К серверным СУБД относят Microsoft SQL Server, Informix, Sybase, DB2, Oracle.

2. Файлы баз данных Microsoft Access

Рассмотрим основы работы в СУБД Microsoft Access. Access – это система управления реляционными базами данных (БД). Работа с БД происходит в реальном времени. Поэтому перед тем как приступать к созданию объектов БД необходимо создать файл для их хранения.

В рамках файла БД используются следующие объекты:

- Таблицы для сохранения данных.
- Запросы для поиска и извлечения только требуемых данных.
- Формы для просмотра, добавления и изменения данных в таблицах.
- Отчеты для анализа и печати данных в определенном формате.
- Страницы доступа к данным для просмотра, обновления и анализа данных из базы данных через Интернет или интрасеть.

Данные сохраняются один раз в таблице, но просматриваются из различных расположений. При изменении данных в таблице, происходит их автоматическое обновление везде, где они появляются.

Краткая характеристика объектов БД

Таблицы. Процесс создания БД начинается с проектирования и построения таблиц, которые затем будут наполняться данными.

Структура таблицы определяется в режиме ее конструирования.

Ввод и редактирование данных удобнее осуществлять в режиме таблицы.

Каждая строка таблицы представляет собой запись, а столбец – поле. Запись содержит набор данных об одном объекте (например, в телефонном справочнике: фамилию, адрес, номер телефона), а поле – однородные данные о всех объектах (например, адреса всех занесенных в таблицу абонентов).

Запросы предназначены для отбора данных, удовлетворяющих некоторым условиям. Благодаря запросам можно в любой момент выбрать из БД необходимую информацию.

Формы. Форма представляет собой бланк, подлежащий заполнению, или маску, накладываемую на набор данных. Форма-бланк позволяет упростить процесс заполнения БД. С помощью формы-маски можно ограничить объем информации, доступной пользователю, обращающемуся к БД.

Отчеты служат для отображения итоговых данных из таблиц и запросов в удобном для просмотра виде. В Access существуют разнообразные способы оформления отчетов.

Страницы доступа к данным представляют собой специальный тип Web-страниц, предназначенных для просмотра и работы через Интернет с данными, хранящимися в БД.

Макросы. В состав БД часто включают макросы, автоматизирующие процессы ее заполнения, отбора информации и т.д.

Модули. Для выполнения операций, которые невозможно реализовать с помощью команд Access или макросов, предусмотрены процедуры обработки событий или выполнения вычислений, написанные на языке Visual Basic for Application (VBA). Такие процедуры оформляются в виде модулей.

3. Обобщенная технология работы с базой данных

Можно выделить следующие основные этапы работы с базой данных (рисунк 4.5):

- создание структуры таблиц БД и связей между таблицами;
- ввод и редактирование данных в таблицах;
- обработка данных, содержащихся в таблицах;
- вывод информации из БД.

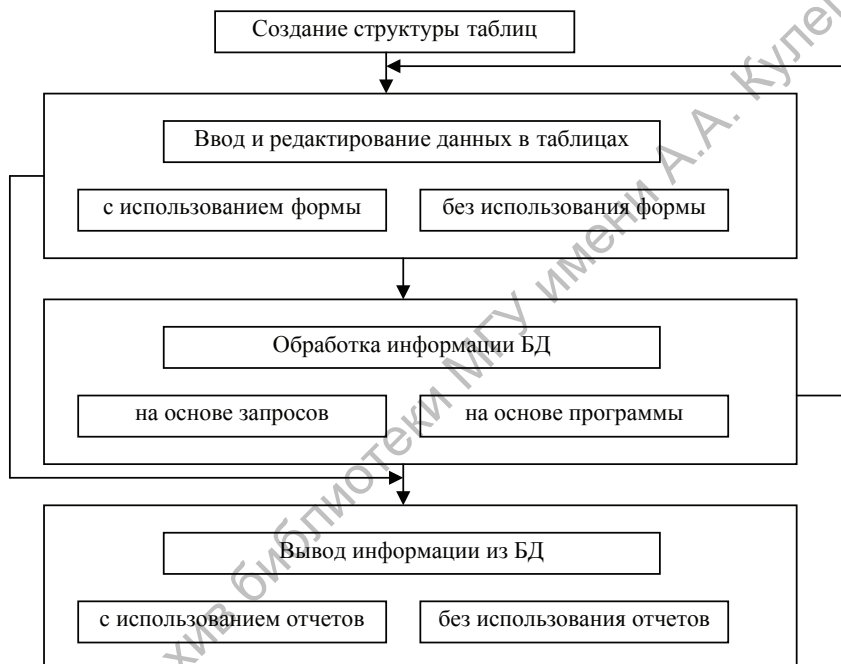


Рис. 4.5. Схема обобщенной технологии работы в СУБД

Создание структуры таблиц БД. Создание каждой новой таблицы начинается с создания структуры таблицы, т.е. определения перечня полей, их типов и размеров. Создание структуры таблицы не связано с ее наполнением данными, поэтому эти две операции можно разнести во времени.

Ввод и редактирование данных. Заполнение таблиц данными возможно как непосредственно вводом данных, так и в результате выполнения программных модулей и запросов.

Вводить и корректировать данные можно двумя способами:

- с помощью представляемой стандартной формы в виде таблицы (на листе данных);
- с помощью *экранных форм*, специально созданных пользователем.

Обработка данных, содержащихся в таблицах. Обрабатывать информацию, содержащуюся в таблицах БД, можно путем использования запросов или в процессе выполнения специально разработанных программных модулей. Запрос представляет собой инструкцию на отбор записей. Запрос-выборка предназначен для отбора данных, хранящихся в таблицах, и не изменяет эти данные. Запрос на выборку является самым распространенным типом запроса. Запрос-изменение предназначен для изменения или перемещения данных. К этому типу запросов относятся: запрос на добавление записей, запрос на удаление записей, запрос на создание таблицы, запрос на обновление. Запрос с параметром позволяет определить одно или несколько условий отбора во время выполнения запроса.

Результатом выполнения запроса является таблица с временным набором данных (динамический набор). Записи динамического набора могут включать поля из одной или нескольких таблиц БД. На основе запроса можно построить отчет или форму.

Вывод информации из БД. СУБД позволяет вывести на экран или принтер информацию, содержащуюся в БД из режимов таблицы или формы. Такой способ вывода данных может использоваться как черновой вариант. Пользователь имеет возможность построения отчетов для вывода данных, при этом можно:

- включать в отчет выборочную информацию из таблиц БД;
- добавлять информацию, не содержащуюся в БД;
- при необходимости выводить итоговые данные;
- располагать информацию в любом удобном виде.

4. Работа с таблицами

Таблицей является совокупность данных, объединенных общей темой. Таким образом для каждой темы отводится отдельная таблица. Это позволяет избежать повторения сохраняемых данных, положительно сказывается на эффективности работы с базой данных и уменьшает вероятность возникновения ошибок ввода.

Структуру создаваемых таблиц следует тщательно продумывать. Существует 3 способа построения таблиц (в порядке предпочтения): в окне конструктора, с помощью мастера таблиц и непосредственно на листе данных.

Каждая таблица состоит из полей и записей. Количество полей в записи, их тип и величина (размер, формат) устанавливаются пользователем в процессе конструирования таблицы.

К созданию новой таблицы можно приступить сразу после открытия БД. Для этого следует:

1) в окне БД открыть вкладку Таблицы;
 2) выполнить щелчок по кнопке Создать. В результате на экране появится окно Новая таблица, в правой части которого перечисляются способы создания таблицы;

3) выбрать способ Конструктор. После нажатия кнопки ОК вы попадете в окно конструктора таблицы.

Проект таблицы состоит из трех колонок: Имя поля, Тип данных и Описание. В первой из них указывается имя поля, во второй – его тип, а в третьей – вводится информация о назначении поля (не обязательно).

Допустимые в Access типы данных

Тип данных	Описание
текстовый	Текст не более 255 символов
МЕМО	Текстовые либо текстовые и числовые данные объемом до 65 535 символов
числовой	числовые данные (целые или дробные)
дата/время	даты и время, допускается ввод дат от 100 по 9999 год
денежный	денежные значения
счетчик	уникальное длинное целое, автоматически увеличиваемое при добавлении в таблицу каждой новой записи. Обычно используется в качестве ключа. Значения в этих полях не могут обновляться.
логический	логические значения (да/нет)
поле объекта OLE	объект, созданный другим приложением (например, таблица MS Excel или документ MS Word, рисунок)
гиперссылка	Гиперссылки, представляющие собой путь к файлу на жестком диске или адрес в сети Интернет.

Дополнительные свойства полей

Название	Назначение
Размер поля	Определяется максимальный размер данных, для хранения которых предназначено данное поле. Поле с текстовым типом данных может иметь размер от 1 до 255 байт (символов), для числового типа данных размер поля может быть следующим: байт: целые числа от 0 до 255. Занимает 1 байт. целое: числа от -32 768 до +32 767. Занимает 2 байта. длинное целое: числа от -2 147 483 648 до +2 147 483 647. Занимает 4 байта. с плавающей точкой (4 байта): хранит числа с точностью до 6 знаков от $-3,4 \times 10^{38}$ до $+3,4 \times 10^{38}$. Занимает 4 байта. с плавающей точкой (8 байт): хранит числа с точностью до 10 знаков от $-1,797 \times 10^{308}$ до $+1,797 \times 10^{308}$. Занимает 8 байт.

Название	Назначение
формат	определяется формат представления данных при выводе на экран или печать
число десятичных знаков	определяется число знаков, выводимых после десятичного разделителя
маска ввода	определяется образец для всех данных, которые вводятся в это поле. Позволяет контролировать правильность данных при вводе.
подпись поля	определяется обозначение, которое должно быть распечатано в форме или отчете. При отсутствии данного значения за основу автоматически будет взято имя поля.
значение по умолчанию	определяется значение, которое автоматически вводится в поле при формировании новой записи данных
условие на значение	определяется выражение, которое устанавливает ограничение на значения при вводе данных. Позволяет контролировать правильность данных при вводе.
сообщение об ошибке	определяется сообщение об ошибке, появляющееся при вводе значения с нарушением наложенного условия
обязательное поле	определяется возможность хранения в поле значения Null
пустые строки	определяется разрешение на ввод пустых строк
индексированное поле	определяется, нужно ли индексировать поле для ускорения операции поиска и сортировки

Ключевые поля

Обязательным условием нормальной работы с БД является однозначная идентификация каждой записи. Поэтому при создании таблицы рекомендуется определить *ключевое поле*.

Ключевое поле – это одно (или несколько полей) таблицы, однозначно определяющее содержимое других полей. Ключевое поле таблицы не может содержать пустых или повторяющихся значений. Ключевое поле можно определить только в режиме конструктора.

Access создает индекс для ключевого поля и использует его для поиска записей и объединения таблиц в запросе. Таблицу, в которой не определен ключ, нельзя использовать при установке связей, кроме того, поиск и сортировка в такой таблице выполняются медленнее.

Определение связей между таблицами

Access позволяет строить БД, таблицы которой могут быть связаны между собой. Связь между таблицами определяется путем добавления связываемых таблиц в окно «Схема данных» с последующим перетаскиванием ключевого поля из одной таблицы в другую. Как правило, связывают ключевое поле одной таблицы с соответствующим ему полем другой таблицы, которое называют *полем внешнего ключа*.

Связанные поля могут иметь разные имена, но у них должны быть одинаковые типы данных и свойства поля.

Тип отношения в создаваемой Microsoft Access связи зависит от способа определения связываемых полей.

Отношение «один-ко-многим» является наиболее часто используемым типом связи между таблицами. В отношении «один-ко-многим» каждой записи в таблице **A** могут соответствовать несколько записей в таблице **B**, но запись в таблице **B** не может иметь более одной соответствующей ей записи в таблице **A**. Отношение «один-ко-многим» создается в том случае, когда только одно из полей является ключевым или имеет уникальный индекс.

Отношение «один-к-одному». При отношении «один-к-одному» запись в таблице **A** может иметь не более одной связанной записи в таблице **B** и наоборот. Отношения этого типа используются нечасто, поскольку большая часть сведений, связанных таким образом, может быть помещена в одну таблицу.

Отношение «один-к-одному» может использоваться для разделения очень широких таблиц, или для отделения части таблицы по соображениям защиты. Отношение «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.

Отношение «многие-ко-многим». При отношении «многие-ко-многим» одной записи в таблице **A** могут соответствовать несколько записей в таблице **B**, а одной записи в таблице **B** несколько записей в таблице **A**. Этот тип связи возможен только с помощью третьей (связующей) таблицы, первичный ключ которой состоит из двух полей, которые являются внешними ключами таблиц **A** и **B**.

Отношение «многие-ко-многим» по сути дела представляет собой два отношения «один-ко-многим» с третьей таблицей.

5. Запросы. Запрос на выборку

Запросы предназначены для отбора данных, удовлетворяющих заданным критериям. Результатом выполнения запроса является набор записей, собранных в таблице, который называется динамическим или временным набором данных.

Типы запросов, создаваемых в Microsoft Access

- Запрос на выборку
- Запрос с параметрами
- Перекрестный запрос
- Запрос на изменение

Запрос на выборку

Запрос на выборку является самым распространенным типом запроса. Он отбирает данные из одной или более таблиц по заданным условиям, а затем отображает их в нужном порядке. Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений. Запрос можно создать с помощью мастера или с помощью **Конструктора запросов**.

Окно конструктора запросов разделено на две части. В верхней части находятся окна таблиц со списками полей. Можно добавлять туда нужные таблицы или запросы, вызвав контекстное меню и выбрав пункт «Добавить таблицу».

В нижней части конструктора запроса находится бланк запроса. Далее для формирования простого запроса на выборку достаточно выполнить следующие шаги:

- 1) добавить нужные поля в бланк запроса;
- 2) установить условия (критерии) отбора записей;
- 3) указать порядок сортировки записей в запросе.

При формировании условий отбора используются операторы сравнения, которые приведены в таблице 4.1, и логические операторы **and** (логическое «И»), **or** (логическое «ИЛИ»), **not** (логическое «НЕ»).

Таблица 4.1. – Операторы сравнения

Обозначение оператора	Действие
=	равно
>	больше, чем
>=	больше или равно
<	меньше, чем
<=	меньше или равно
<>	не равно
Like “...”	проверяет соответствие текстового или Мемо поля заданному шаблону символов. Допускается использование следующих символов шаблона: ? —
Between...And...	проверяет, что значение поля находится внутри заданного диапазона, верхняя и нижняя границы которого разделяются логическим оператором And (значения даты или времени должны быть заключены между символами #)
In (...)	проверяет на равенство любому значению из списка; элементы списка разделяются запятыми

Примеры условий отбора

1. Указание диапазона значений

>120

<=330

Between 144 And 330

2. Текст, часть текста или совпадающие значения

абрикос – отбор записей об абрикосах

“абрикос” or “киви” – отбор записей об абрикосах и киви

a* – отбор записей о фруктах, начинающихся на букву ‘a’

like “[г-к]*” – значения поля начинаются с одной из букв г-к

C?e* – начинается на C, третья буква e

*ми – заканчивается на ‘ми’

ломтик – отбор записей, содержащих слово “ломтик”

C* and *ломт* – значения поля начинаются на ‘C’ и содержат ‘ломт’

3. Значения, не совпадающие с указанным (оператор Not)

Not c* – отбор записей, не начинающихся на “c”

Not абрикос

4. Значения в списке (In)

In (абрикос, киви) – то же, что “абрикос” or “киви” (см. выше)

In (вишня, груша, яблоко)

5. Пустое значение поля (Null)

Is null – записи, в которых указанное поле не заполнено

Is not null – записи, в которых указанное поле заполнено

Вычисления в запросах

Запрос можно использовать для выполнения расчетов и подведения итогов, обобщающих данные из исходных таблиц. Для этих целей используют статистические функции SQL. Статистическую функцию задают в строке «Групповая операция», которая появляется после нажатия кнопки Σ (Групповые операции) панели инструментов. Групповые операции приведены в таблице 4.2.

Посредством статистических функций можно обработать содержимое каждого поля запроса. Результат обработки появляется в динамическом наборе записей запроса.

Таблица 4.2 – Групповые функции

Обозначение	Функция	Назначение
AVG	среднее	вычисляет среднее сгруппированных значений
COUNT	число	находит количество сгруппированных записей (для подсчета числа записей с учетом значений Null —
SUM	сумма	вычисляет сумму сгруппированных значений
MIN	минимум	находит наименьшее из сгруппированных значений
MAX	максимум	находит наибольшее из сгруппированных значений
STDEV	отклонение	вычисляет статистическое стандартное отклонение для сгруппированных значений: оно показывает, насколько широко разбросаны значения данных относительно среднего значения
VAR	дисперсия	вычисляет статистическую дисперсию для сгруппированных значений: равна квадрату стандартного отклонения
FIRST	первая	выбирает первое из сгруппированных значений
LAST	последняя	выбирает последнее из сгруппированных значений

6. Создание запросов других типов

Кроме простых запросов (на выборку) существуют следующие виды QBE запросов:

- параметрические запросы;
- перекрестные запросы;
- запросы-действия.

Параметрические запросы

Это запросы, которые представляют собой варианты базового запроса, отличающиеся друг от друга только другими значениями в условиях отбора. При выполнении запроса с параметром отображается диалоговое окно с приглашением ввести данные, например условие для возвращения записей или значение, которое требуется вставить в поле.

Для создания параметрического запроса в строке “Условия отбора” в квадратных скобках следует ввести обращение.

Перекрестные запросы

Перекрестные запросы позволяют компактно отображать отобранные данные и объединять однородную информацию. Перекрестный запрос выполняет некоторую статистическую функцию (подсчитывает сумму, среднее значение, число значений), после чего результаты группируются в виде таблицы

по двум наборам данных, один из которых определяет заголовки столбцов, а другой – заголовки строк.

Для создания перекрестного запроса в режиме конструктора следует поместить в бланк не менее трех полей и выбрать тип запроса – **Перекрестный**. При этом в бланке запроса отобразятся строки “Групповая операция” и “Перекрестная таблица”.

Например, перекрестный запрос, построенный по данным таблиц **Служащие, Пенсионные фонды, Распределение по фондам**:

	Фамилия	Имя	Название фонда	Процент
Групповая операция	Группировка	Группировка	Группировка	Sum
Перекр.таблица	Заголовки строк	Заголовки строк	Заголовки столбцов	Значение

После запуска запроса появится таблица, в которой для каждого служащего указан его процент в каждом пенсионном фонде.

Перекрестный запрос можно создавать и с помощью мастера. Но мастер создает перекрестный запрос только на основе одной таблицы или одного запроса, подготовленного ранее.

Запросы-действия

Запросы – действия используют для изменения и переноса данных в таблицах, а также для обновления, добавления и удаления групп записей, для создания новой таблицы на основе одной или нескольких таблиц. В запросах-действиях условия, указанные в бланке служат для отбора записей, а дополнительная информация определяет характер изменений.

В окне БД перед именем запроса-действия Access устанавливает ! (восклицательный знак).

Наиболее опасным является запрос на удаление, он удаляет из таблицы все записи, отвечающие заданным критериям. Это может быть полезно для “чистки” БД – например можно удалить все записи, утратившие актуальность (например, дата которых предшествует 1 января 2000 года). Однако ошибка при вводе условия может привести к уничтожению актуальных данных. Поэтому перед выполнением запроса на удаление всегда следует вначале провести запрос на выборку, в котором условия используются только для отбора записей.

Запросы на создание таблицы и добавление могут использоваться для перемещения записей из одной таблицы (или нескольких таблиц) в другую. Запрос на добавление присоединяет записи к существующей таблице. Запрос

на создание таблицы создает резервную копию таблицы, например, перед выполнением запроса на удаление.

Запрос на обновление позволяет изменить значение любого поля БД для записей, отвечающего заданным критериям. В бланке такого запроса появляется строка 'Обновление', она применяется для ввода выражений для ввода выражений, определяющих новое значение. Например, если цена на набор (таблица *Наборы*, поле *Цена*) возросла на 15%, то выражение может выглядеть следующим образом:

[Наборы].[Цена]*0,15

7. QBE и SQL запросы

В Access различают два типа запросов: QBE (Query By Example – запрос по образцу) и SQL (Structured Query Language – структурированный язык запросов). При создании SQL-запросов применяются операторы и функции языка SQL.

QBE-запросы, создаваемые в окне конструктора запроса, Access автоматически преобразует в SQL-запросы. Просмотреть соответствующий SQL-запрос можно просмотрев запрос в **режиме SQL**.

С помощью языка SQL можно составить любое количество сложных запросов, а также управлять обработкой запросов. SQL-запрос представляет собой последовательность инструкций, в которую могут входить выражения и статистические функции SQL.

Пока пользователь работает в Access, у него обычно нет необходимости обращаться к SQL-инструкциям. SQL нужен в том случае, если из Access пользователь делает запрос к другой БД, поддерживающей SQL.

SQL (язык структурированных запросов) – универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

Основным объектом хранения реляционной базы данных является таблица, поэтому все SQL-запросы – это операции над таблицами. В соответствии с этим, запросы делятся на

- запросы, оперирующие самими таблицами (создание и изменение таблиц);
- запросы, оперирующие с отдельными записями (или строками таблиц) или наборами записей.

Язык SQL представляет собой совокупность: операторов; инструкций; и вычисляемых функций. Операторы (и другие зарезервированные слова) в SQL всегда нужно писать прописными буквами.

Операторы SQL делятся на:

- операторы определения данных:

CREATE создает объект БД (саму базу, таблицу, представление, пользователя и т. д.)

ALTER изменяет объект

DROP удаляет объект

- операторы манипуляции данными:

SELECT считывает данные, удовлетворяющие заданным условиям

INSERT добавляет новые данные

UPDATE изменяет существующие данные

DELETE удаляет данные

Транзакция – это группа последовательных операций, которая представляет собой логическую единицу работы с данными. Транзакция либо выполняется целиком и успешно, либо не выполняется вообще. Операторы управления транзакциями: COMMIT – применяет транзакцию. ROLLBACK откатывает все изменения, сделанные в контексте текущей транзакции. SAVEPOINT делит транзакцию на более мелкие участки.

SELECT – оператор языка SQL, возвращающий набор данных (выборку) из базы данных, удовлетворяющих заданному условию.

В большинстве случаев, выборка осуществляется из одной или нескольких таблиц.

При формировании запроса SELECT пользователь описывает ожидаемый набор данных: его вид (набор столбцов) и его содержимое (критерий попадания записи в набор, группировка значений, порядок вывода записей и т.п.).

Запрос выполняется следующим образом: сначала извлекаются все записи из таблицы, а, затем, для каждой записи набора проверяется ее соответствие заданному критерию. Если осуществляется слияние из нескольких таблиц, то сначала составляется произведение таблиц, а уже затем из полученного набора отбираются требуемые записи.

Формат запроса с использованием данного оператора:

SELECT список полей FROM список таблиц WHERE условия...

Основные ключевые слова, относящиеся к запросу SELECT:

WHERE – используется для определения, какие строки должны быть выбраны (выборка) или включены в GROUP BY.

GROUP BY – используется для объединения строк с общими значениями в элементы меньшего набора строк.

ORDER BY — используется для определения, какие столбцы используются для сортировки результирующего набора данных.

Примеры:

SELECT * FROM T; – вернет все столбцы из таблицы T

SELECT C1 FROM T; – столбец C1 из таблицы T

SELECT * FROM T WHERE C1 = 1; – значения всех столбцов всех строк таблицы, у которых значение поля C1 равно 1 (выборка)

SELECT * FROM T ORDER BY C1 DESC; – результат будет отсортирован в обратном порядке

Глава 5

ЗАЩИТА ИНФОРМАЦИИ

1. Защита информации в Интернете

Проведение финансовых операций с использованием Интернета, заказ товаров и услуг, использование кредитных карточек, доступ к закрытым информационным ресурсам, передача телефонных разговоров требуют обеспечения соответствующего уровня безопасности.

Конфиденциальная информация, которая передается по сети Интернет, проходит через определенное количество маршрутизаторов и серверов, прежде чем достигнет пункта назначения. Обычно маршрутизаторы не отслеживают проходящие сквозь них потоки информации, но возможность того, что информация может быть перехвачена, существует. Более того, информация может быть изменена и передана адресату в измененном виде. К сожалению, сама архитектура сети Интернет всегда оставляет возможность для недобросовестного пользователя осуществить подобные действия.

Всегда существует проблема выбора между необходимым уровнем защиты и эффективностью работы в сети. В некоторых случаях пользователями или потребителями меры по обеспечению безопасности могут быть расценены как меры по ограничению доступа и эффективности. Однако такие средства, как, например, криптография, позволяют значительно усилить степень защиты, не ограничивая доступ пользователей к данным.

Принципы защиты информации

Проблемы, возникающие с безопасностью передачи информации при работе в компьютерных сетях, можно разделить на три основных типа:

- **перехват информации** – целостность информации сохраняется, но ее конфиденциальность нарушена;
- **модификация информации** – исходное сообщение изменяется либо полностью подменяется другим и отправляется адресату;
- **подмена авторства информации**. Данная проблема может иметь серьезные последствия. Например, кто-то может послать письмо от вашего имени (этот вид обмана принято называть спуфингом) или Web-сервер может притворяться электронным магазином, принимать заказы, номера кредитных карт, но не высылать никаких товаров.

В соответствии с перечисленными проблемами при обсуждении вопросов безопасности под самим термином «безопасность» подразумевается совокупность трех различных характеристик обеспечивающей безопасность системы:

1. Аутентификация – это процесс распознавания пользователя системы и предоставления ему определенных прав и полномочий. Каждый раз, когда заходит речь о степени или качестве аутентификации, под этим следует понимать степень защищенности системы от посягательств сторонних лиц на эти полномочия.

2. Целостность – состояние данных, при котором они сохраняют свое информационное содержание и однозначность интерпретации в условиях различных воздействий. В частности, в случае передачи данных под целостностью понимается идентичность отправленного и принятого.

3. Секретность – предотвращение несанкционированного доступа к информации. В случае передачи данных под этим термином обычно понимают предотвращение перехвата информации.

2. Криптография

Для обеспечения секретности применяется шифрование, или криптография, позволяющая трансформировать данные в зашифрованную форму, из которой извлечь исходную информацию можно только при наличии ключа.

В основе шифрования лежат два основных понятия: алгоритм и ключ. Алгоритм – это способ закодировать исходный текст, в результате чего получается зашифрованное послание. Зашифрованное послание может быть интерпретировано только с помощью ключа.

Очевидно, чтобы зашифровать послание, достаточно алгоритма. Однако использование ключа при шифровании предоставляет два существенных преимущества. Во-первых, можно использовать один алгоритм с разными ключами для отправки посланий разным адресатам. Во-вторых, если секретность ключа будет нарушена, его можно легко заменить, не меняя при этом алгоритм шифрования. Таким образом, безопасность систем шифрования зависит от секретности используемого ключа, а не от секретности алгоритма шифрования. Многие алгоритмы шифрования являются общедоступными.

Количество возможных ключей для данного алгоритма зависит от числа бит в ключе. Например,

8-битный ключ допускает 256 (28) комбинаций ключей. Чем больше возможных комбинаций ключей, тем труднее подобрать ключ, тем надежнее зашифровано послание. Так, например, если использовать 128-битный ключ, то

необходимо будет перебрать 2128 ключей, что в настоящее время не под силу даже самым мощным компьютерам. Важно отметить, что возрастающая производительность техники приводит к уменьшению времени, требующегося для вскрытия ключей, и системам обеспечения безопасности приходится использовать все более длинные ключи, что, в свою очередь, ведет к увеличению затрат на шифрование.

Поскольку столь важное место в системах шифрования уделяется секретности ключа, то основной проблемой подобных систем является генерация и передача ключа. Существуют две основные схемы шифрования: симметричное шифрование (его также иногда называют традиционным или шифрованием с секретным ключом) и шифрование с открытым ключом (иногда этот тип шифрования называют асимметричным).

При симметричном шифровании отправитель и получатель владеют одним и тем же ключом (секретным), с помощью которого они могут зашифровывать и расшифровывать данные. При симметричном шифровании используются ключи небольшой длины, поэтому можно быстро шифровать большие объемы данных. Симметричное шифрование используется, например, некоторыми банками в сетях банкоматов. Однако симметричное шифрование обладает несколькими недостатками. Во-первых, очень сложно найти безопасный механизм, при помощи которого отправитель и получатель смогут тайно от других выбрать ключ. Возникает проблема безопасного распространения секретных ключей. Во-вторых, для каждого адресата необходимо хранить отдельный секретный ключ. В третьих, в схеме симметричного шифрования невозможно гарантировать личность отправителя, поскольку два пользователя владеют одним ключом.

В схеме шифрования с открытым ключом для шифрования послания используются два различных ключа. При помощи одного из них послание зашифровывается, а при помощи второго – расшифровывается. Таким образом, требуемой безопасности можно добиться, сделав первый ключ общедоступным (открытым), а второй ключ хранить только у получателя (закрытый, личный ключ). В таком случае любой пользователь может зашифровать послание при помощи открытого ключа, но расшифровать послание способен только обладатель личного ключа. При этом нет необходимости заботиться о безопасности передачи открытого ключа, а для того чтобы пользователи могли обмениваться секретными сообщениями, достаточно наличия у них открытых ключей друг друга.

Недостатком асимметричного шифрования является необходимость использования более длинных, чем при симметричном шифровании, ключей для обеспечения эквивалентного уровня безопасности, что сказывается на вычислительных ресурсах, требуемых для организации процесса шифрования.

3. Электронная подпись

Даже если послание, безопасность которого мы хотим обеспечить, должным образом зашифровано, все равно остается возможность модификации исходного сообщения или подмены этого сообщения другим. Одним из путей решения этой проблемы является передача пользователем получателю краткого представления передаваемого сообщения. Подобное краткое представление называют контрольной суммой, или дайджестом сообщения.

Контрольные суммы используются при создании резюме фиксированной длины для представления длинных сообщений. Алгоритмы расчета контрольных сумм разработаны так, чтобы они были по возможности уникальны для каждого сообщения. Таким образом, устраняется возможность подмены одного сообщения другим с сохранением того же самого значения контрольной суммы.

Однако при использовании контрольных сумм возникает проблема передачи их получателю. Одним из возможных путей ее решения является включение контрольной суммы в так называемую электронную подпись.

При помощи электронной подписи получатель может убедиться в том, что полученное им сообщение послано не сторонним лицом, а имеющим определенные права отправителем. Электронные подписи создаются шифрованием контрольной суммы и дополнительной информации при помощи личного ключа отправителя. Таким образом, кто угодно может расшифровать подпись, используя открытый ключ, но корректно создать подпись может только владелец личного ключа. Для защиты от перехвата и повторного использования подпись включает в себя уникальное число – порядковый номер.

4. Аутентификация

Аутентификация является одним из самых важных компонентов организации защиты информации в сети. Прежде чем пользователю будет предоставлено право получить тот или иной ресурс, необходимо убедиться, что он действительно тот, за кого себя выдает.

При получении запроса на использование ресурса от имени какого-либо пользователя сервер, предоставляющий данный ресурс, передает управление серверу аутентификации. После получения положительного ответа сервера аутентификации пользователю предоставляется запрашиваемый ресурс.

При аутентификации используется, как правило, принцип, получивший название “что он знает”, – пользователь знает некоторое секретное слово, которое он посылает серверу аутентификации в ответ на его запрос. Одной из схем аутентификации является использование стандартных паролей. Эта схема является наиболее уязвимой с точки зрения безопасности – пароль может быть

перехвачен и использован другим лицом. Чаще всего используются схемы с применением одноразовых паролей. Даже будучи перехваченным, этот пароль будет бесполезен при следующей регистрации, а получить следующий пароль из предыдущего является крайне трудной задачей. Для генерации одноразовых паролей используются как программные, так и аппаратные генераторы, представляющие собой устройства, вставляемые в слот компьютера. Знание секретного слова необходимо пользователю для приведения этого устройства в действие. Одной из наиболее простых систем, не требующих дополнительных затрат на оборудование, но в то же время обеспечивающих хороший уровень защиты, является S/Key, на примере которой можно продемонстрировать порядок представления одноразовых паролей.

В процессе аутентификации с использованием S/Key участвуют две стороны – клиент и сервер. При регистрации в системе, использующей схему аутентификации S/Key, сервер присылает на клиентскую машину приглашение, содержащее зерно, передаваемое по сети в открытом виде, текущее значение счетчика итераций и запрос на ввод одноразового пароля, который должен соответствовать текущему значению счетчика итераций. Получив ответ, сервер проверяет его и передает управление серверу требуемого пользователем сервиса.

5. Защита сетей

В последнее время корпоративные сети все чаще включаются в Интернет или даже используют его в качестве своей основы. Учитывая то, какой урон может принести незаконное вторжение в корпоративную сеть, необходимо выработать методы защиты. Для защиты корпоративных информационных сетей используются брандмауэры.

Брандмауэр – это система или комбинация систем, позволяющие разделить сеть на две или более частей и реализовать набор правил, определяющих условия прохождения пакетов из одной части в другую. Как правило, эта граница проводится между локальной сетью предприятия и Интернетом, хотя ее можно провести и внутри. Однако защищать отдельные компьютеры невыгодно, поэтому обычно защищают всю сеть.

Брандмауэр пропускает через себя весь трафик и для каждого проходящего пакета принимает решение – пропускать его или отбросить. Для того чтобы брандмауэр мог принимать эти решения, для него определяется набор правил.

Брандмауэр может быть реализован как аппаратными средствами (то есть как отдельное физическое устройство), так и в виде специальной программы, запущенной на компьютере.

Как правило, в операционную систему, под управлением которой работает брандмауэр, вносятся изменения, цель которых – повышение защиты

самого брандмауэра. Эти изменения затрагивают как ядро ОС, так и соответствующие файлы конфигурации. На самом брандмауэре не разрешается иметь разделов пользователей, а следовательно, и потенциальных дыр – только раздел администратора. Некоторые брандмауэры работают только в однопользовательском режиме, а многие имеют систему проверки целостности программных кодов.

Брандмауэр обычно состоит из нескольких различных компонентов, включая фильтры или экраны, которые блокируют передачу части трафика. Все брандмауэры можно разделить на два типа:

- **пакетные фильтры**, которые осуществляют фильтрацию IP-пакетов средствами фильтрующих маршрутизаторов;
- **серверы прикладного уровня**, которые блокируют доступ к определенным сервисам в сети.

Таким образом, брандмауэр можно определить как набор компонентов или систему, которая располагается между двумя сетями и обладает следующими свойствами:

- весь трафик из внутренней сети во внешнюю и из внешней сети во внутреннюю должен пройти через эту систему;
- только трафик, определенный локальной стратегией защиты, может пройти через эту систему;
- система надежно защищена от проникновения.

6. Антивирусная защита компьютера

Вирусом называется специально созданная программа, способная самостоятельно распространяться в компьютерной среде. Если вирус попал в компьютер вместе с одной из программ или с файлом документа, то через некоторое время другие программы или файлы на этом компьютере будут заражены. Если компьютер подключен к локальной или глобальной сети, то вирус может распространиться и дальше, на другие компьютеры.

Результаты работы различных вирусов оказываются, как правило, схожими – они повреждают программы и документы, находящиеся на компьютере, что часто приводит к их утрате. Некоторые вирусы способны уничтожать вообще всю информацию на дисках компьютеров.

Для защиты от вирусов можно использовать

- общие средства защиты информации, такие, как резервное копирование важных данных и программ;
- профилактические меры, позволяющие уменьшить вероятность заражения вирусом;
- специализированные программы для защиты от вирусов.

Основным средством защиты информации является резервное копирование наиболее ценных данных. В случае утраты информации по любой из вышеперечисленных причин жесткие диски переформатируются и подготавливаются к новой эксплуатации. На «чистый отформатированный диск устанавливаются все необходимое программное обеспечение, которое тоже берут с дистрибутивных носителей.

Несмотря на то, что общие средства защиты информации очень важны для защиты от вирусов, все же их одних недостаточно. Необходимо и применение специализированных программ для защиты от вирусов.

Антивирусная программа (антивирус) – специализированная программа для обнаружения компьютерных вирусов и восстановления зараженных (модифицированных) файлов, а также для профилактики – предотвращения заражения (модификации) файлов или операционной системы.

Существует несколько основных методов поиска вирусов, которые применяются антивирусными программами: сканирование; эвристический анализ; обнаружение изменений; резидентные мониторы. Антивирусы могут реализовывать все перечисленные выше методики, либо только некоторые из них. Если антивирусная программа обнаруживает вирус в файле, то она удаляет из него программный код вируса. Если лечение невозможно, то зараженный файл удаляется целиком.

Для нормальной работы на компьютере каждый пользователь должен следить за обновлением антивирусов.

Глава 6

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

1. Основные понятия моделирования

Во все времена человек отражает в своем мышлении объекты реальности в виде идеальных мысленных моделей. И действует, исходя из ожидаемого поведения их прототипов. Это этап первого отражения мира в мышлении человека.

С появлением компьютера ситуация изменилась. Человек теперь может передать компьютеру свои знания, создав компьютерную модель реального объекта. Таким образом происходит второе отражение природы – из человеческого мышления в память компьютера. В этом смысле компьютерный мир является *третьей реальностью*.

Эти замечания имеют целью предложить методологию и мировоззренческую основу для понимания моделирования, подчеркнуть его исключительное значение в информатике, его конечную цель. Итак, и компьютер, и программы – это всего лишь средство для моделирования (в самом широком смысле).

Модели и моделирование

В обыденной жизни известно, что модель – это имитация, повторение в масштабе какого-либо реального объекта. Например, глобус – это модель земного шара, модель молекулы воды.

Модель не всегда создается искусственно. Иногда в качестве модели используют другие реальные объекты. В медицине, например, новые методы сложных операций отрабатывают на кроликах, собаках, обезьянах. Организм этих животных для медицинских целей служит моделью человеческого организма. Именно так отрабатывались операции пересадки сердца и других жизненно важных органов.

Необычайная общность понятия модели, тесная связь со свойством отражения в природе ведут к тому, что достаточно точное определение не может быть простым и включает общие категории.

Под **моделью** будем понимать систему не отличимую от реального (моделируемого) объекта в отношении некоторых свойств, полагаемых существенными, и отличающуюся по всем остальным свойствам, которые мы полагаем несущественными. При этом отсутствие в модели несущественных элементов не менее важно, чем присутствие существенных.

Можно сказать, что модель – это упрощенное представление о реальном объекте, процессе или явлении.

Жизнь постоянно ставит перед человеческим обществом разнообразнейшие задачи. Варианты решения возникающих задач все чаще проверяются на моделях, потому что испытания сложных и дорогостоящих систем требуют больших затрат и не всегда возможны в принципе. Например, чтобы изучить сейсмические свойства готового здания, никто не станет устраивать искусственное землетрясение. Гораздо проще и дешевле создать модель здания и испытать ее на вибративном устройстве.

Итак, модель – это объект (материальный или мысленный) или явление, которые для каких-то целей рассматриваются вместо другого объекта-оригинала. **Моделирование** – это процесс создания и использования моделей для решения практических задач.

Модель и информация

Понятие модели тесно связано с понятием информации. Сущность любой модели заключена в том, что с ее помощью можно получить информацию о свойствах моделируемого объекта.

Большинство моделей хранят информацию о внешней форме моделируемого объекта. Глобус несет информацию о внешней форме и особенностях земной поверхности. Сейсмическая модель здания – информацию о внешней форме и прочности здания.

Для одного и того же объекта могут быть построены совершенно непохожие модели. Одна модель может представлять информацию об одних свойствах, другая – о других. Например, карта мира политическая и физическая.

Адекватность модели – это степень совпадения свойств модели и моделируемого объекта. При этом оцениваются те свойства моделируемого объекта, которые модель представляет.

Оценка адекватности модели может быть самой разной: от полной адекватности до частичной адекватности и неадекватности.

При построении модели естественно желание добиться высокой адекватности модели. Однако погоня за высокой адекватностью может привести к чрезмерной сложности модели. Здесь следует искать разумную середину: модель должна иметь достаточную для задачи адекватность и не быть очень сложной.

2. Виды моделей

Среди различных особенностей моделей важнейшими считаются следующие:

- форма представления модели;
- внешние размеры;

- зависимость от времени;
- связь с конкретной отраслью знаний.

1. В зависимости от **формы представления** различают два основных вида моделей: *предметные* (материальные) и *образно-знаковые*.

Модель называется **предметной**, если эта модель является объемным предметом.

Например, глобус – это предметная модель земного шара, электронная модель сердца – также предметная модель. Предметные модели создаются специально или подбираются из имеющихся объектов.

Модель называется **образно-знаковой**, если эта модель является описанием моделируемого объекта в виде образов и знаков. Например, любую фотографию объекта можно считать его моделью. Фотография объемным предметом не является. Это описание объекта в виде образа, поэтому является образно-знаковой моделью. Или учебник истории. С точки зрения информатики он содержит образно-знаковые модели (описания) различных исторических событий.

Существует большое число различных носителей информации. Вид носителя информации определяет вид образно-знаковой модели. Поэтому основными являются следующие виды образно-знаковых моделей:

- мысленные модели;
- документальные модели;
- компьютерные модели.

Перечень видов моделей в зависимости от формы представления можно задать схемой (рис. 2).

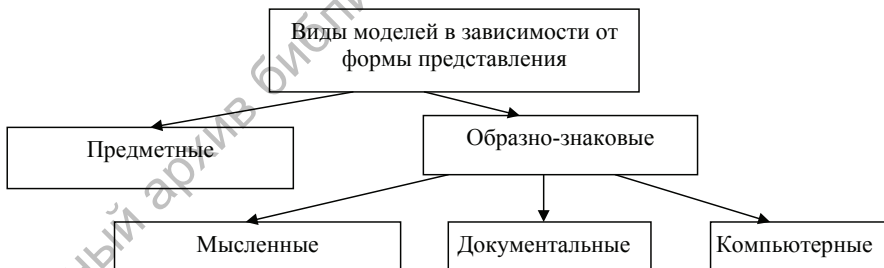


Рис. 6.1. Виды моделей по форме представления

Мысленная модель – это мысленный образ моделируемого объекта в памяти человека.

Документальная модель – это описание или изображение моделируемого объекта на бумаге, картоне или другом плоском носителе. Со многими объектами окружающего мира мы знакомы только по их документальным моделям: фотографиям, описаниям. Документальными моделями являются также географические карты, чертежи машин и механизмов и т. п.

Компьютерная модель – это описание или изображение моделируемого объекта в памяти компьютера. Компьютерные модели создаются с помощью различных программных средств: систем программирования, электронных таблиц, текстовых и графических редакторов.

2. В зависимости от *внешних размеров* модели делятся на масштабные (макеты) и немасштабные.

Модель называется **масштабной**, если внешние размеры модели получены увеличением или уменьшением размеров моделируемого объекта.

Глобус Земли – масштабная модель. Документальная или компьютерная модель также может быть масштабной. Например, географическая карта, чертеж.

Если внешние размеры модели не отражают внешних размеров моделируемого объекта, то такую модель называют **немасштабной**. Немасштабной моделью является любое описание моделируемого объекта в виде текста и формул.

3. Если модель учитывает *зависимость свойств моделируемого объекта от времени*, то модель называют **динамической**. В противном случае модель называют **статической**.

Например, движущиеся модели судов являются динамическими моделями, а модели судов в музеях – статическими. Если динамическая модель полностью имитирует поведение моделируемого объекта во времени, то такая модель называется **имитационной**.

4. Виды моделей *по отраслям знаний*. Каждая наука при описании фактов и явлений пользуется своим языком: для математики – это язык цифр и математических формул, для физики – это язык физических закономерностей. Свои подходы имеют химия, биология, генетика, социология и т. д.

Использование научных знаний в процессе моделирования повышает адекватность моделей. Поэтому при создании моделей обычно используются язык, законы, методы той или иной науки. В зависимости от этого модели разделяют по отраслям знаний: *математические модели, статистические модели, биологические модели, социологические модели* и т. д. Если модель с определенной отраслью знаний не связана, то ее называют *простейшей*.

Чаще других используются математические модели, потому что математика дает возможность учитывать важные числовые свойства объектов (размеры, скорость, углы, дальность, массу, плотность и т. п.).

Вид любой модели определяется отдельно по каждой особенности. Например, электронная модель сердца – это предметная модель (по форме представления), немасштабная (в зависимости от внешних размеров), динамическая (в зависимости от времени), электронная (по отрасли знаний). При рассмотрении моделей могут учитываться и другие их особенности: способ решения (аналитические и численные модели), способ записи (алгоритмические и неалгоритмические модели) и т. д.

Для одного и того же объекта могут быть построены модели разных видов. Например, модель участка земной поверхности можно представить мысленно (мысленная модель), в виде рельефного макета (предметная модель), в виде карты (документальная модель) или в виде графического объекта на экране компьютера (компьютерная модель).

3. Этапы процесса моделирования

Процесс моделирования возникает при решении любой задачи, всегда занимает какое-то время и проходит в несколько этапов. Основными являются следующие этапы:

- 1) постановка задачи;
- 2) оценка имеющейся информации и выбор плана создания модели;
- 3) создание модели;
- 4) проверка адекватности модели;
- 5) получение решения задачи с помощью модели.

Примеры моделирования

Пример 1. Пусть требуется выяснить, как ведет себя новый автомобиль на высокой скорости (этап 1).

Исследованию подлежат силы, которые действуют на автомобиль в скоростном потоке воздуха. Для этих целей можно использовать моделирование в аэродинамической трубе (этап 2).

В аэродинамической трубе создаются условия езды на высокой скорости (этап 3).

Адекватность модели вытекает из условий моделирования (этап 4).

В результате «продувки» автомобиля получается решение задачи (этап 5).

Пример 2. Пусть имеется лист жести. Нужно так раскроить его, чтобы получить жестяной короб максимального объема (этап 1).

В зависимости от размеров листа нужно определить размеры заготовок для короба. При данных условиях имеет смысл создать математическую документальную модель (этап 2).

Создается математическая модель в виде чертежей и уравнений (этап 3).

Адекватность модели зависит от точности чертежей и формул (этап 4).

В результате математических преобразований модели получается решение задачи: способ раскроя (этап 5).

Форму представления модели часто называют *формой реализации модели* и говорят о мысленной реализации, документальной реализации и т. д.

Для решения одних задач достаточно мысленной реализации модели. Для решения других задач нужно строить документальную реализацию модели.

При решении сложных задач могут потребоваться предметные или компьютерные реализации моделей.

В процессе создания сложной модели одна реализация может сменять другую. В разных задачах могут использоваться одна, две или все четыре реализации в любом порядке.

Таким образом, этап создания модели зачастую состоит из нескольких этапов по числу реализации модели. План очередности разных реализаций модели выбирается на втором этапе моделирования.

Моделирование, при котором проходятся все пять этапов, называется **полным**. Если при моделировании пятый этап исключается, то моделирование называют **неполным**.

Существуют задачи, целью которых является создание модели как нового самостоятельного объекта. После проверки адекватности (этап 4) дальнейшее использование модели этапом моделирования уже не считается.

Например, целью спортивного моделизма является создание моделей судов, самолетов, автомобилей. Используются эти модели как независимые объекты, без связи с моделируемым объектом.

К этому же виду относится моделирование во многих задачах научных исследований. Основная цель науки – познать неизвестное явление, описать его, объяснить, т. е. построить модель явления. Построенная модель используется затем при создании более сложных моделей, т. е. рассматривается как новый независимый объект.

4. Компьютерные реализации моделей

Компьютерные реализации моделей обладают универсальными возможностями. С одной стороны, они могут заменить документальные модели (с помощью текстовых и графических редакторов), с другой – заменить предметные.

Моделирование с помощью систем программирования. Математические задачи вычислительного характера и многие другие могут быть решены с помощью составления моделей в виде алгоритмов на каком-либо языке программирования.

Алгоритм (программа), записанный на бумаге, является документальной моделью, алгоритмической по способу записи. Алгоритм в памяти – компьютерная модель и также алгоритмическая.

Моделирование с помощью электронных таблиц. Во многих вычислительных задачах удобно использовать для компьютерной реализации электронные таблицы.

Таким образом, компьютерное моделирование является одним из важнейших компонентов новых информационных технологий.

Глава 7

МЕТОДЫ ОПТИМИЗАЦИИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

1. Информационные технологии поддержки принятия решений

Характеристика и назначение. Главной особенностью информационной технологии поддержки принятия решений является качественно новый метод организации взаимодействия человека и компьютера. Основной целью этой технологии является выработка решения. Выработка решения происходит в результате итерационного процесса (рис. 7.1), в котором участвуют:

- система поддержки принятия решений в роли вычислительного звена и объекта управления;
- человек как управляющее звено, задающее входные данные и оценивающее полученный результат вычислений на компьютере.



Рис. 7.1. Итерационный процесс информационной технологии поддержки принятия решений

Окончание итерационного процесса происходит по воле человека. В этом случае можно говорить о способности информационной системы совместно с пользователем создавать новую информацию для принятия решений.

Дополнительно к этой особенности информационной технологии поддержки принятия решений можно указать еще ряд ее отличительных характеристик:

- ориентация на решение плохо структурированных задач;
- сочетание традиционных методов доступа и обработки компьютерных данных с возможностями математических моделей и методами решения задач на их основе;

- направленность на непрофессионального пользователя компьютера;
- высокая адаптивность, обеспечивающая возможность приспособляться к особенностям имеющегося технического и программного обеспечения, а также требованиям пользователя.

Информационная технология поддержки принятия решений может использоваться на любом уровне управления. Кроме того, решения, принимаемые на различных уровнях управления, часто должны координироваться. Поэтому важной функцией и систем, и технологий является координация лиц, принимающих решения, как на разных уровнях управления, так и на одном уровне.

2. Основные компоненты СППР

Рассмотрим структуру СППР (системы поддержки принятия решений) (рис. 7.2), а также функции составляющих ее блоков, которые определяют основные технологические операции.

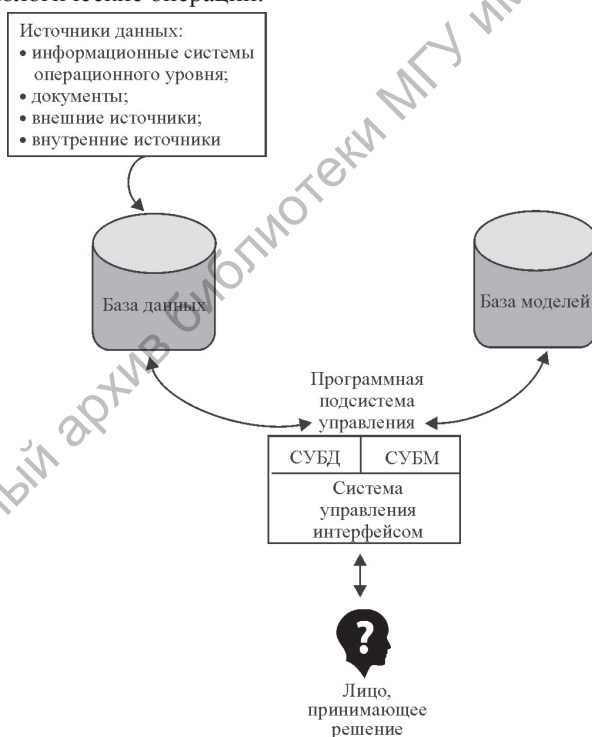


Рис. 7.2. Основные компоненты информационной технологии поддержки принятия решений

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

База данных играет в информационной технологии поддержки принятия решений (СППР) важную роль. Данные могут использоваться непосредственно пользователем для расчетов при помощи математических моделей. Рассмотрим источники данных и их особенности:

1. Часть данных поступает от информационной системы операционного уровня. Чтобы использовать их эффективно, эти данные должны быть предварительно обработаны.

Для этого существуют две возможности:

– использовать для обработки данных об операциях учреждения систему управления базой данных, входящую в состав системы поддержки принятия решений;

– сделать обработку за пределами системы поддержки принятия решений, создав для этого специальную базу данных. Этот вариант более предпочтителен для учреждений, производящих большое количество коммерческих операций. Обработанные данные об операциях учреждения образуют файлы, которые для повышения надежности и скорости доступа хранятся за пределами системы поддержки принятия решений.

2. Помимо данных об операциях учреждения для функционирования системы поддержки принятия решений требуются и другие внутренние данные, например данные о движении персонала, инженерные данные и т. п., которые должны быть своевременно собраны, введены и поддержаны.

3. Важное значение, особенно для поддержки принятия решений на верхних уровнях управления, имеют данные из внешних источников. В числе необходимых внешних данных следует указать данные о конкурентах, национальной и мировой экономике. В отличие от внутренних внешние данные обычно приобретаются у специализирующихся на их сборе организаций.

4. В настоящее время широко исследуется вопрос о включении в базу данных еще одного источника данных – документов, содержащих записи, письма, контракты, приказы и т. п. Если содержание этих документов будет записано в памяти и затем обработано по некоторым ключевым характеристикам (поставщикам, потребителям, датам, видам услуг и др.), то система получит новый мощный источник информации.

Система управления данными (СУБД) должна обладать следующими возможностями:

- составление комбинаций данных, получаемых из различных источников посредством использования процедур агрегирования и фильтрации;

- быстрое прибавление или исключение того или иного источника данных;
- построение логической структуры данных в терминах пользователя;
- использование и манипулирование неофициальными данными для экспериментальной проверки рабочих альтернатив пользователя;
- обеспечение полной логической независимости этой базы данных от других операционных баз данных, функционирующих в рамках учреждения.

База моделей. Целью создания моделей являются описание и оптимизация некоторого объекта или процесса. Использование моделей обеспечивает проведение анализа в системах поддержки принятия решений. Модели, базируясь на математической интерпретации проблемы, при помощи определенных алгоритмов способствуют нахождению информации, полезной для принятия правильных решений.

Например, модель линейного программирования дает возможность определить наиболее выгодную производственную программу выпуска нескольких видов продукции при заданных ограничениях на ресурсы.

Использование моделей в составе информационных систем началось с применения статистических методов и методов финансового анализа, которые реализовывались командами обычных алгоритмических языков. Позже были созданы специальные языки, позволяющие моделировать ситуации типа «что будет, если?» или «как сделать, чтобы?» Такие языки, созданные специально для построения моделей, дают возможность построить модели определенного типа, обеспечивающие нахождение решения при гибком изменении переменных.

Существует множество типов *моделей* и способов их классификации, например, по цели использования, области возможных приложений, способу оценки переменных и т. п.

По цели использования модели подразделяются на *оптимизационные*, связанные с нахождением точек минимума или максимума некоторых показателей (например, управляющие часто хотят знать, какие их действия ведут к максимизации прибыли или минимизации затрат), и *описательные*, описывающие поведение некоторой системы и не предназначенные для целей управления (оптимизации).

По способу оценки модели классифицируются на *детерминистские*, использующие оценку переменных одним числом при конкретных значениях исходных данных, и *стохастические*, оценивающие переменные несколькими параметрами, так как исходные данные заданы вероятностными характеристиками.

Детерминистские модели более популярны, чем стохастические, потому что они менее дорогие, их легче строить и использовать. К тому же часто с их помощью получается вполне достаточная информация для принятия решения.

По области возможных приложений модели разбиваются на *специализированные*, предназначенные для использования только одной системой, и *универсальные* — для использования несколькими системами.

Специализированные модели более дорогие, они обычно применяются для описания уникальных систем и обладают большей точностью.

В системах поддержки принятия решения *база моделей* состоит из *стратегических, тактических и оперативных моделей*, а также математических моделей (рис. 7.3) в виде совокупности модельных блоков, модулей и процедур, используемых как элементы для их построения.

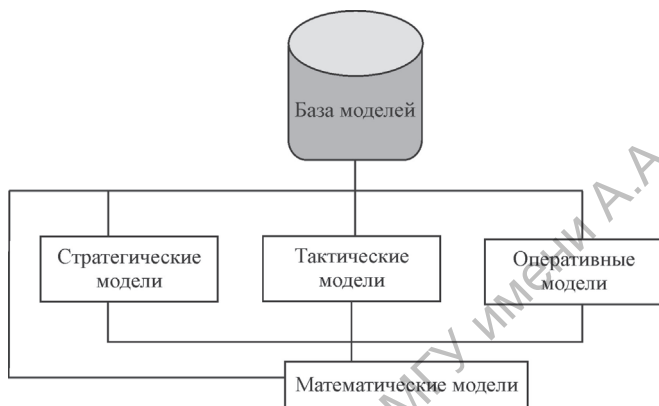


Рис. 7.3. Типы моделей, составляющих базу моделей

Стратегические модели используются на высших уровнях управления для установления целей организации, объемов ресурсов, необходимых для их достижения, а также политики приобретения и использования этих ресурсов. Они могут быть также полезны при выборе вариантов размещения предприятий, прогнозировании политики конкурентов и т. п. Для стратегических моделей характерны значительная широта охвата, множество переменных, представление данных в сжатой агрегированной форме. Часто эти данные базируются на внешних источниках и могут иметь субъективный характер. Горизонт планирования в стратегических моделях, как правило, измеряется в годах. Эти модели обычно детерминистские, описательные, специализированные для использования на одной определенной фирме.

Тактические модели применяются управляющими среднего уровня для распределения и контроля использования имеющихся ресурсов. Среди возможных сфер их использования следует указать финансовое планирование, планирование требований к работникам, планирование увеличения продаж, построение схем компоновки предприятий. Эти модели применимы обычно лишь к отдельным частям фирмы (например, к системе производства и сбыта) и могут также включать в себя агрегированные показатели. Временной горизонт, охватываемый тактическими моделями, – от одного месяца до двух лет. Здесь также могут потребоваться данные из внешних источников, но основное

внимание при реализации данных моделей должно быть уделено внутренним данным фирмы. Обычно тактические модели реализуются как детерминистские, оптимизационные и универсальные.

Оперативные модели используются на низших уровнях управления для поддержки принятия оперативных решений с горизонтом, измеряемым днями и неделями. Возможные применения этих моделей включают в себя ведение дебиторских счетов и кредитных расчетов, календарное производственное планирование, управление запасами и т. д. Оперативные модели обычно используют для расчетов внутрифирменные данные. Они, как правило, детерминистские, оптимизационные и универсальные (т.е. могут быть использованы в различных организациях).

Математические модели состоят из совокупности модельных блоков, модулей и процедур, реализующих математические методы. Сюда могут входить процедуры линейного программирования, статистического анализа временных рядов, регрессионного анализа и т. п. – от простейших процедур до сложных ППП. Модельные блоки, модули и процедуры могут использоваться как по отдельности, так и комплексно для построения и поддержания моделей.

Система управления базой моделей (СУБМ) должна обладать следующими возможностями: создавать новые модели или изменять существующие, поддерживать и обновлять параметры моделей, манипулировать моделями.

Система управления интерфейсом. Эффективность и гибкость информационной технологии во многом зависят от характеристик интерфейса системы поддержки принятия решений. Интерфейс определяет: язык пользователя; язык сообщений компьютера, организующий диалог на экране дисплея; знания пользователя.

Язык пользователя – это те действия, которые пользователь производит в отношении системы путем использования возможностей клавиатуры; электронных карандашей, пишущих на экране; джойстика; «мыши»; команд, подаваемых голосом, и т. п. Наиболее простой формой языка пользователя является создание форм входных и выходных документов. Получив входную форму (документ), пользователь заполняет его необходимыми данными и вводит в компьютер. Система поддержки принятия решений производит необходимый анализ и выдает результаты в виде выходного документа установленной формы.

Значительно возросла за последнее время популярность *визуального интерфейса*. С помощью манипулятора «мышь» пользователь выбирает представленные ему на экране в форме картинок объекты и команды, реализуя таким образом свои действия.

Управление компьютером при помощи человеческого *голоса* – самая простая и поэтому самая желанная форма языка пользователя. Она еще недостаточно разработана и поэтому малопопулярна. Существующие разработки требуют от пользователя серьезных ограничений: определенного набора слов и выражений; специальной надстройки, учитывающей особенности голоса

пользователя; управления в виде дискретных команд, а не в виде обычной гладкой речи. Технология этого подхода интенсивно совершенствуется, и в ближайшем будущем можно ожидать появления систем поддержки принятия решений, использующих речевой ввод информации.

Язык сообщений – это то, что пользователь видит на экране дисплея (символы, графика, цвет), данные, полученные на принтере, звуковые выходные сигналы и т.п. Важным измерителем эффективности используемого интерфейса является выбранная форма диалога между пользователем и системой. В настоящее время наиболее распространены следующие *формы диалога*: запросно-ответный режим, командный режим, режим меню, режим выполнения пропусков в выражениях, предлагаемых компьютером.

Каждая форма в зависимости от типа задачи, особенностей пользователя и принимаемого решения может иметь свои достоинства и недостатки.

Долгое время единственной реализацией языка сообщений был отпечатанный или выведенный на экран дисплея отчет или сообщение. Теперь появилась новая возможность представления выходных данных – машинная графика. Она дает возможность создавать на экране и бумаге цветные графические изображения в трехмерном виде. Использование машинной графики значительно повышает наглядность и интерпретируемость выходных данных и становится все более популярным в информационной технологии поддержки принятия решений.

За последние несколько лет наметилось новое направление, развивающее машинную графику, – мультипликация. Мультипликация оказывается особенно эффективной для интерпретации выходных данных систем поддержки принятия решений, связанных с моделированием физических систем и объектов.

Например, система поддержки принятия решений, предназначенная для обслуживания клиентов в банке, с помощью мультипликационных моделей может реально просмотреть различные варианты организации обслуживания в зависимости от потока посетителей, допустимой длины очереди, количества пунктов обслуживания и т.п.

В ближайшие годы следует ожидать использования в качестве языка сообщений человеческого голоса. Сейчас эта форма применяется в системе поддержки принятия решений сферы финансов, где в процессе генерации чрезвычайных отчетов голосом поясняются причины исключительности той или иной позиции.

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

Совершенствование *интерфейса* системы поддержки принятия решений определяется успехами в развитии каждого из трех указанных компонентов. Интерфейс должен обладать следующими возможностями:

- манипулировать различными формами диалога, изменяя их в процессе принятия решения по выбору пользователя;

- передавать данные системе различными способами;
- получать данные от различных устройств системы в различном формате;
- гибко поддерживать знания пользователя (оказывать помощь по запросу, подсказывать).

3. Информационные технологии экспертных систем

Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки экспертных систем, основанных на использовании искусственного интеллекта. Экспертные системы дают возможность менеджеру или специалисту получать консультации экспертов по любым проблемам, о которых этими системами накоплены знания.

Под искусственным интеллектом обычно понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека.

Решение специальных задач требует специальных знаний. Однако не каждая компания может себе позволить держать в своем штате экспертов по всем связанным с ее работой проблемам или даже приглашать их каждый раз, когда проблема возникла. Главная идея использования технологии экспертных систем заключается в том, чтобы получить от эксперта его знания и, загрузив их в память компьютера, использовать всякий раз, когда в этом возникнет необходимость. Являясь одним из основных приложений искусственного интеллекта, экспертные системы представляют собой компьютерные программы, трансформирующие опыт экспертов в какой-либо области знаний в форму эвристических правил (эвристик). Эвристики не гарантируют получения оптимального результата с такой же уверенностью, как обычные алгоритмы, используемые для решения задач в рамках технологии поддержки принятия решений. Однако часто они дают в достаточной степени приемлемые решения для их практического использования. Все это делает возможным использовать технологию экспертных систем в качестве советующих систем.

Сходство информационных технологий, используемых в экспертных системах и системах поддержки принятия решений, состоит в том, что обе они обеспечивают высокий уровень поддержки принятия решений. Однако имеются три существенных различия. Первое связано с тем, что решение проблемы в рамках систем поддержки принятия решений отражает уровень ее понимания пользователем и его возможности получить и осмыслить решение. Технология экспертных систем, наоборот, предлагает пользователю принять решение, превосходящее его возможности. Второе отличие указанных технологий выражается в способности экспертных систем пояснять свои рассужде-

ния в процессе получения решения. Очень часто эти пояснения оказываются более важными для пользователя, чем само решение. Третье отличие связано с использованием нового компонента информационной технологии – знаний.

Основными компонентами информационной технологии, используемой в экспертной системе, являются (рис. 7.4): интерфейс пользователя, база знаний, интерпретатор, модуль создания системы.

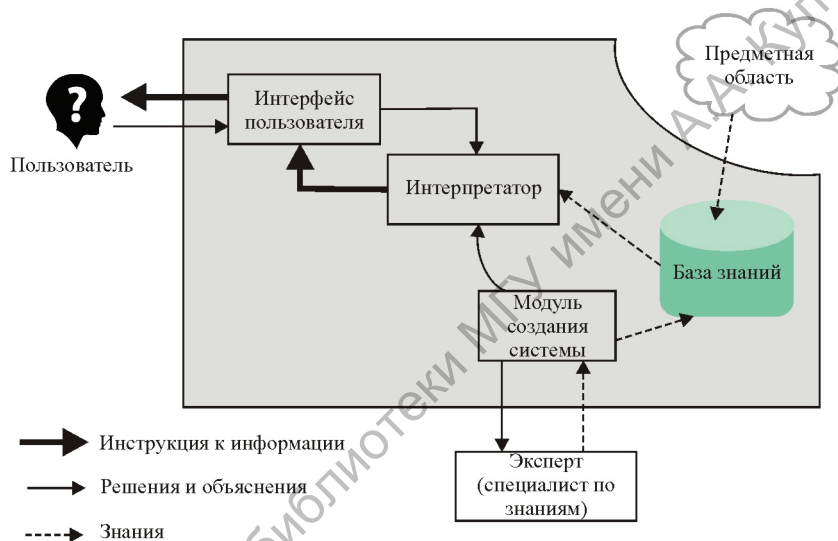


Рис. 7.4. Основные компоненты информационной технологии экспертных систем

Интерфейс пользователя. Менеджер (специалист) использует интерфейс для ввода информации и команд в экспертную систему и получения выходной информации из нее. Команды включают в себя параметры, направляющие процесс обработки знаний. Информация обычно выдается в форме значений, присваиваемых определенным переменным.

Менеджер может использовать четыре метода ввода информации: меню, команды, естественный язык и собственный интерфейс.

Технология экспертных систем предусматривает возможность получать в качестве выходной информации не только решение, но и необходимые объяснения.

Различают два вида объяснений:

- объяснения, выдаваемые по запросам. Пользователь в любой момент может потребовать от экспертной системы объяснения своих действий;
- объяснения полученного решения проблемы. После получения решения пользователь может потребовать объяснений того, как оно было получено. Си-

стема должна пояснить каждый шаг своих рассуждений, ведущих к решению задачи. Хотя технология работы с экспертной системой не является простой, пользовательский интерфейс этих систем является дружелюбным и обычно не вызывает трудностей при ведении диалога.

База знаний содержит факты, описывающие проблемную область, а также логическую взаимосвязь этих фактов. Центральное место в базе знаний принадлежит правилам. Правило определяет, что следует делать в данной конкретной ситуации, и состоит из двух частей: условие, которое может выполняться или нет, и действие, которое следует произвести, если выполняется условие.

Все используемые в экспертной системе правила образуют систему правил, которая даже для сравнительно простой системы может содержать несколько тысяч правил.

Все виды знаний в зависимости от специфики предметной области и квалификации проектировщика (инженера по знаниям) с той или иной степенью адекватности могут быть представлены с помощью одной либо нескольких семантических моделей. К наиболее распространенным моделям относятся логические, продукционные, фреймовые и семантические сети.

Интерпретатор. Это часть экспертной системы, производящая в определенном порядке обработку знаний, находящихся в базе знаний. Технология работы интерпретатора сводится к последовательному рассмотрению совокупности правил (правило за правилом). Если условие, содержащееся в правиле, соблюдается, то выполняется определенное действие, и пользователю предоставляется вариант решения его проблемы.

Кроме того, во многих экспертных системах вводятся дополнительные блоки: база данных, блок расчета, блок ввода и корректировки данных. Блок расчета необходим в ситуациях, связанных с принятием управленческих решений. При этом важную роль играет база данных, где содержатся плановые, физические, расчетные, отчетные и другие постоянные или оперативные показатели. Блок ввода и корректировки данных используется для оперативного и своевременного отражения текущих изменений в базе данных.

Модуль создания системы служит для создания набора (иерархии) правил. Существует два подхода, которые могут быть положены в основу модуля создания системы: использование алгоритмических языков программирования и использование оболочек экспертных систем.

Для представления базы знаний специально разработаны языки Лисп и Пролог, хотя можно использовать и любой известный алгоритмический язык.

Оболочка экспертных систем представляет собой готовую программную среду, которая может быть приспособлена к решению определенной проблемы путем создания соответствующей базы знаний. В большинстве случаев использование оболочек позволяет создавать экспертные системы быстрее и легче в сравнении с программированием.

СОДЕРЖАНИЕ

Глава 1. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.....	3
1. Информация. Информационные технологии и системы	3
2. Операционные системы.....	5
3. Языки программирования	6
4. Технологии программирования	8
Глава 2. ОСНОВНЫЕ ПРОГРАММНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ	10
1. Текстовые редакторы и издательские системы	10
2. Основы работы в Word.....	11
3. Форматирование документов Word	13
4. Таблицы в Word	15
5. Графические объекты в Word	16
6. Компьютерная графика и графические редакторы	18
7. Основные понятия компьютерной графики	20
8. Электронные таблицы Excel.....	22
9. Типы данных в ячейке.....	23
10. Построение диаграмм	28
11. Печать в Excel	29
12. Работа со списками данных в Excel.....	30
13. Формирование сводной информации в Excel	35
14. Сервисные инструментальные средства: Файловые менеджеры	41
15. Системы математических вычислений	45
16. Система подготовки презентаций Power Point	47
Глава 3. СЕТЕВЫЕ ТЕХНОЛОГИИ И ИНТЕРНЕТ	51
1. Компьютерные сети	51
2. Сеть Интернет и сетевые протоколы	53
3. Семиуровневая модель структуры протоколов связи	54
4. Адресация в сети Интернет.....	55
5. Основные сервисы сети Интернет.....	57
6. Поиск информации в Интернет.....	60
7. Перспективные технологии на основе Интернета	62
8. Основы разработки веб-документов. Язык HTML.....	64
9. Каскадные таблицы стилей	74
10. Развитие языка HTML. XML и XHTML-документы	75
11. Сценарии	76
12. Этапы и методы разработки сайтов.....	77

Глава 4. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ	79
1. Базы данных и СУБД	79
2. Файлы баз данных Microsoft Access	82
3. Обобщенная технология работы с базой данных	84
4. Работа с таблицами	85
5. Запросы. Запрос на выборку	88
6. Создание запросов других типов	91
7. QBE и SQL запросы	93
Глава 5. ЗАЩИТА ИНФОРМАЦИИ	95
1. Защита информации в Интернете	95
2. Криптография	96
3. Электронная подпись	98
4. Аутентификация	98
5. Защита сетей	99
6. Антивирусная защита компьютера	100
Глава 6. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ	102
1. Основные понятия моделирования	102
2. Виды моделей	103
3. Этапы процесса моделирования	106
4. Компьютерные реализации моделей	107
Глава 7. МЕТОДЫ ОПТИМИЗАЦИИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ	108
1. Информационные технологии поддержки принятия решений	108
2. Основные компоненты СППР	109
3. Информационные технологии экспертных систем	115

Учебное издание

Батан Сергей Николаевич
Батан Лариса Валерьевна
Малашук Оксана Вячеславовна

ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Курс лекций

Технический редактор *А. Л. Позняков*
Компьютерная верстка *А. Л. Позняков*
Корректор *И. Г. Латушкина*

Подписано в печать 18.05.2016.
Формат 60x84/16. Гарнитура Times New Roman Cyr.
Усл.-печ. л. 7,0. Уч.-изд. л. 6,8. Тираж 87 экз. Заказ № 172.

Учреждение образования «Могилевский государственный университет
имени А. А. Кулешова», 212022, Могилев, Космонавтов, 1
Свидетельство ГРИИРПИ № 1/131 от 03.01.2014 г.

Отпечатано в отделе оперативной полиграфии
МГУ имени А. А. Кулешова. 212022, Могилев, Космонавтов, 1