

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН
ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ имени НИЗАМИ

У.Ю. Юлдашев

Информационные технологии
Часть 2

Ташкент – 2007

В пособии «Информационные технологии» рассмотрены основные понятия и средства информатики (часть 1) и её практический аспект, представленный новыми информационными технологиями (часть 2).

Данное пособие написано в соответствии с действующими программами общеобразовательных дисциплин «Информатика», «Информационные технологии» и рассчитано на учащихся профессиональных колледжей, а также на студентов бакалавриата всех направлений.

Материал пособия может быть полезен слушателям факультетов повышения квалификации преподавателей.

Автор: доктор технических наук, профессор У.Ю.Юлдашев

Рецензенты:

доктор педагогических наук, профессор А.А.Абдукадыров
кандидат физико–математических наук, доцент С.И.Махкамов

Рекомендовано Учёным советом ТГПУ имени Низами,
Протокол № 3 - 23, от 30 ноября 2006 г.

© ТГПУ имени Низами

Глава 1. Информационные технологии

В сфере современного промышленного производства известны многочисленные технологические процессы (например, в автомобилестроении) с гарантированным результатом и воспроизводимым технологическим циклом. Безусловно, это результат создания эффективных технологий и технологизации соответствующих производственных сфер.

Термин «Технология» при переводе с греческого означает искусство, мастерство, умение. В более широком толковании технология – это комплекс научных и инженерных знаний, реализованных в приёмах труда, а также в комплексах технологических условий (материальных, технических, энергетических, трудовых) и способах их соединения для гарантированного создания конечной продукции. Технологии свойственны расчленённость процесса на стадии (этапы) и, одновременно, системная его полнота (целостность). В зависимости от уровня технологии сохраняется (или нет) необходимость контроля состояния продукта на этапах. В так называемых «высоких технологиях» нет необходимости контроля изделия на промежуточных этапах технологической цепочки – достаточно лишь обеспечить технологические условия и конечный продукт будет гарантированно создан.

Технологизация – это целенаправленные воздействия человека посредством разработки и внедрения технологий в те или иные процессы и преобразования их в интересах гарантированного достижения pragматических результатов производства и эффективности его в целом.

Переработка информации, её хранение и передача сопровождали человечество на всех этапах его развития, принимая различные формы: от примитивных до самых современных информационных технологий.

Появление языка, а затем и речи, не потребовало специального инструментария усиления интеллектуальной мощи человека: достаточным оказался его мозг. Язык и речь оказались носителями информации, человеческого опыта и знаний. Их накопление в виде устных рассказов, хранение в памяти и передача от поколения к поколению осуществлялись за счёт природных возможностей человека.

1.1. Информационные технологии в информационном обществе

В истории развития цивилизации известны периоды (этапы), когда в информационной сфере происходили кардинальные изменения благодаря созданию искусственных усилителей природных возможностей человека по хранению, переработке и передаче информации.

Первый из этих этапов связан с изобретением письменности, когда появилась возможность хранения и передачи знаний от поколения к поколению. С созданием письменности человек, пожалуй впервые, получил технологическую поддержку в сфере обработки информации.

Второй этап (середина XVI в.) связан с изобретением книгопечатания, которое радикально изменило общество, культуру, организацию деятельности. Книгопечатание стимулировало развитие наук, ускорило темпы накопления профессиональных знаний. Знания, овеществлённые через трудовой процесс, станки, машины, технологии становились источником новых идей.

Третий этап (конец XIX в.) обусловлен изобретением электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно передавать и накапливать большую по объёму информацию.

Четвертый этап характеризуется как подлинная информационная революция. Начало этого этапа относится к концу 40-х годов XX века, когда были созданы универсальные ЭВМ. С 70-х гг. ядро информационных технологий образуют микротехнология и персональные компьютеры. Возникло микропроцессорное направление, которое и по сей день определяет развитие вычислительной техники.

Пятый этап (конец XX века) обусловлен усилением содержательной переработки информации с целью выработки предложений для принятия управлеченческих решений. В основе содержательной переработки информации находится разработка таких моделей и алгоритмов, которые позволяют преобразование информации текущего состояния в параметры управления. Появление компьютера - великое достижение человечества, но являясь средством запоминания, накопления и быстрой обработки информации, он не определяет правил целевой переработки её в интересах управления. К концу XX века были разработаны модели (математические, логико-лингвистические, вербальные и др.) и алгоритмы управления техническими (автоматизированные и автоматические производства) и социальными (проектные организации, конструкторские бюро и др.) системами.

В основе любого производства, протекающего в обществе, лежат целенаправленные действия, которые невозможны без управления. Управление есть целенаправленный процесс переработки информации и представляет собой логическую составляющую производства. Во второй половине прошлого века дальнейший рост физической составляющей производства начал вступать в противоречие с неудовлетворимым состоянием логической составляющей. Совершенствование логической составляющей, возрастание её доли сулило большие возможности роста эффективности производства. Оценившие эту тенденцию страны свои людские ресурсы своевременно переориентировали в сферу информационно - логического производства и, благодаря этому преуспели в экономическом развитии.

Однако к концу XX века стало понятно, что путь увеличения числа работающих в сфере информационно-логического производства, в сфере управления исчерпал себя. Понадобилось создание средств повышения эффективности управления, усиления интеллектуальных возможностей руководителя. Для этого потребовалось автоматизировать процессы

содержательной переработки информации на основе достижений теории и практики управления, использования передовых информационных технологий.

Особенность пятого этапа состоит не только в резком повышении эффективности управления, но и в глубокой информатизации самой физической составляющей производства, что сделало материальный продукт информационно ёмким (увеличение доли инноваций, дизайна и маркетинга в его стоимости). Появились сначала в составе производственных, а затем и самостоятельные процессы, «исходным сырьём» и «конечным продуктом» в которых является информация.

Появление самостоятельных информационных процессов стало следствием того, что целенаправленная деятельность человека или коллектива людей стала зависеть от их информированности и способности эффективно перерабатывать имеющуюся информацию.

Эти обстоятельства меняют представления о технологии. В современном понимании технология – это управление процессами, направленное на гарантированное создание новых объектов: материальных и (или) информационных. Управление в современных технологиях осуществляется не только с целью преобразования состава, структуры и формы вещества, но и для фиксации, обработки и получения новой информации. К таким можно отнести технологии планирования и управления, научных исследований и разработок, проектирования, банковских операций, криминалистики, медицины, образования и др.

Кординальные изменения в сфере обработки информации вызвали к жизни концепцию *информационного общества*. В таком обществе процесс компьютеризации даст людям доступ к надежным источникам информации, избавит их от рутинной работы, обеспечит высокий уровень автоматизации обработки информации в производственной и социальной сферах. Движущей силой развития общества должно стать производство информационного продукта, а материальное производство и производство энергии будет возложено на машины.

В информационном обществе изменятся не только производство, но и весь уклад жизни, система ценностей, возрастет значимость культурного досуга. По сравнению с индустриальным обществом, где все направлено на производство и потребление товаров, в информационном обществе акцент делается на производстве и потреблении интеллекта, что приведёт к увеличению доли умственного труда. От человека потребуется способность к творчеству, возрастет спрос на знания.

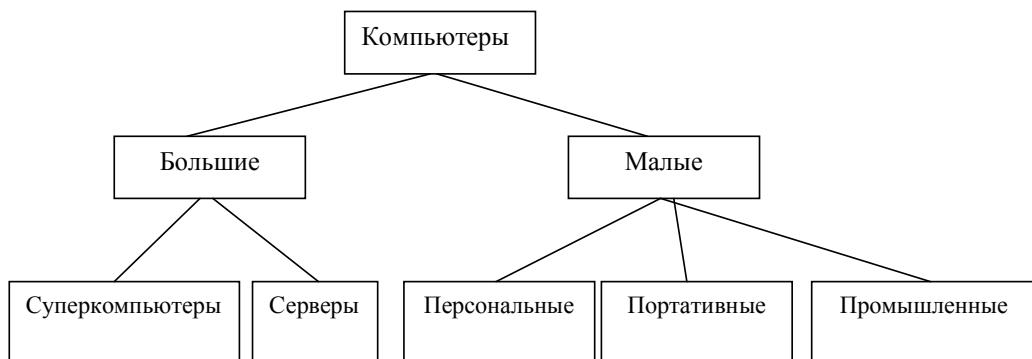
Материальной и технологической базой информационного общества станут различного рода системы на базе компьютерной техники и компьютерных сетей, информационных технологий, телекоммуникационной связи. Ближе всех на пути к информационному обществу стоят страны с развитой информационной индустрией, к числу которых следует отнести США, Японию, Англию, Германию и др.

1.2. Классификация компьютеров, применяемых в ИТ

Формально информационные технологии можно представить в безкомпьютерном варианте. Однако объективно современные информационные технологии функционируют в среде компьютерной техники и коммуникационных средств.

Применяемые в ИТ компьютеры по своим характеристикам отличаются друг от друга. Правильный выбор компьютера во многом предопределяет эффективность ИТ. Решению проблемы способствует классификация компьютеров, применяемых в различных ИТ. Возможны различные критерии классификации. Рассмотрим классификацию компьютеров по обобщенному параметру, где в разной степени учтено несколько характерных признаков:

- назначение и роль компьютеров в системе обработки информации;
- условия взаимодействия человека и компьютера;
- габариты компьютера;
- ресурсные возможности компьютера.



В *суперкомпьютерах* используется мультипроцессорный принцип обработки информации. Основная идея мультипроцессорной обработки – расчленение решаемой задачи на несколько подзадач или частей. Каждая часть решается на своем процессоре.

Сервер – мощный компьютер в вычислительных сетях, который обеспечивает обслуживание подключенных к нему компьютеров и выход в другие сети. На сервере хранятся большие объемы информации, которыми пользуются подключенные к нему компьютеры. На диске сервера располагают совместно используемые программы, базы данных и т.п. Сервер является мощным высокопроизводительным компьютером с большой оперативной и дисковой памятью, высокой надежностью.

Название класса портативных компьютеров происходит от латинского слова «*porto*» - «ношу» и означает, что эти компьютеры легко переносимы. Первые портативные компьютеры назывались лэптопами, т.е. наколенными.

Современное же название их ноутбук происходит от (notebook - блокнот). Ноутбуки различают по габаритам, по весу и по конструкции. В частности, по конструкции ноутбуки можно классифицировать следующим образом.

«Всё в одном». К их числу относятся мощные настольные ноутбуки. В них встроены все дисководы, сетевая карта и факс – modem, все возможные порты расширения, включая USB порт; размер экрана не менее 14»; вес от 4 кг. Существует также вариант ноутбука класса «всё в одном», но только с меньшим экраном и меньшими габаритами, весом до 3кг.

«Модульная конструкция» (Bay module). В целях снижения веса и габаритов из корпуса ноутбука вынесены в виде модулей дисководы, которые подключаются по мере необходимости; размер экрана от 12».

«Slim Notebook» с док – станцией. Это легкий высокопроизводительный ноутбук с встроенными: сетевой картой, факс – modemом, инфракрасным портом и др.; весит 1,6 кг. Если подключить его к док – станции, то становятся доступными все возможные дисководы и порты расширения.

Мини – ноутбук. Имеет технические параметры, сравнимые с обычными ноутбуками, и стандартное для них программное обеспечение. Дисководы соединяются через мини- док – станцию. Но следует иметь в виду, что малые габариты и малый вес (от 850 гр.) обусловили малый размер клавиатуры и экрана , что создаёт неудобства при работе с клавиатурой.

Промышленные компьютеры предназначены для использования в производственных условиях. Они встраиваются в технологический процесс производства какой-нибудь продукции, осуществляют управление технологическими линиями.

1.3. Новые информационные технологии

Информационная индустрия связана с созданием технических и программных средств, методов, технологий для производства новых знаний. Экономические потребности общества явились главным и определяющим стимулом развития такой индустрии. В свою очередь социальное воздействие техники и технологий на общество идёт через производительность труда, путём передачи техническим средствам трудовых функций человека.

Важнейшей составляющей информационной индустрии сегодня являются **новые информационные технологии** (НИТ), т.е. технологизированные процессы получения новой информации на основе широкой компьютеризации с применением современных алгоритмических и программных средств содержательной переработки исходной информации, компьютерных, периферийных, коммуникационных устройств сбора, хранения, обработки и передачи данных.

Виды обеспечений НИТ.

Средства, обеспечивающие сбор, производование, накопление, хранение и передачу информации в НИТ можно объединить в следующие виды обеспечений: аппаратное (АО), программное (ПО), информационное (ИО), организационное (ОО), лингвистическое (ЛО) и др.

Наличие первых четырёх видов обеспечений является необходимым условием функционирования НИТ.

К аппаратному обеспечению НИТ относят :

- компьютер высокой производительности, с развитым графическим интерфейсом, стандартным ПО и минимально необходимым набором устройств ввода - вывода;
- дополнительные внешние устройства ввода - вывода данных, в том числе ввода и манипулирования текстовой и графической информации;
- внешние устройства для хранения и транспортировки больших объёмов информации;
- сетевое оборудование ;
- преобразователи графической и звуковой информации в цифровую форму и обратно;
- средства создания аудиовизуальной информации и манипулирования ею;
- средства связи и телекоммуникаций и др.

Программное обеспечение НИТ представляет собой комплекс программ и соответствующей документации, который обеспечивает решение информационных задач с применением аппаратных средств.

По аналогии с классификацией НИТ программное обеспечение делится на ПО информационных технологий обработки данных (ИТОД) и ПО информационных технологий управления (ИТУ).

Программное обеспечение ИТОД включает в себя стандартное и нестандартное ПО персональных компьютеров.

Стандартное (т.е. поставляемое вместе с персональным компьютером) ПО делится на системное и прикладное. Системное ПО составляют: операционная система (Windows и др.); системы программирования (алгоритмические языки программирования, программы архивированного хранения данных и др.).

Прикладное ПО представляет собой программный инструментарий, основу которого составляет офисный программный комплекс(текстовый редактор, презентационные программы, электронные таблицы и др.), а также электронные календари, электронные записные книжки, электронные игры и др.

Нестандартное ПО составляют инсталлируемые при необходимости: драйвера внешних устройств, программы перевода с одного языка на другой и с одного алфавита на другой, программы распознавания, электронные словари, в том числе толковые словари, издательские настольные системы, инструментальные программные комплексы создания экспертных систем, мультимедийных материалов, в том числе электронных учебных ресурсов и т.п.

Программное обеспечение ИТУ включает в себя :

ПО систем функционального назначения, т.е. средства автоматизации повторяющихся операций управленческого труда на уровне исполнительской деятельности (финансовая, бухгалтерия, маркетинг и т.п.);

ПО управления в составе специального математического обеспечения управления, включающего в себя модели, методы, алгоритмы и программы содергательной переработки информации, также представляемого в вариантах ПО АСУ, ПО САПР, ПО АОС и др.;

ПО управления в составе специального математического обеспечения управления на основе переработки знаний и представляемого в вариантах ПО ИАСУ, ПО ИСАПР, ПО ИАОС и др.;

ПО автоматических систем управления в виде экспертных систем, например, экспертных обучающих систем (ЭОС).

Информационное обеспечение НИТ. Понятие информационное обеспечение (ИО) характерно как для автоматизированных, так и для неавтоматизированных систем. Документы, «делопроизводство» характерны для ИО неавтоматизированных систем. Однако понятие ИО стало широко применяться в связи с машинным хранением и обработкой данных. ИО - это определённым образом структурированная информация, предназначенная для машинного хранения (точнее компьютерная организация такого хранения) и использования прикладными программами. ИО включает в себя совокупность единой системы классификации и кодирования информации для использования в современных информационных технологиях.

Организационное обеспечение НИТ - совокупность людей, методов и средств организации и использования информационных технологий в какой-либо конкретной области.

Правовое обеспечение НИТ - это совокупность правовых норм, определяющих юридический статус использования информационной технологии. Развитие рыночных отношений в информационной деятельности поставило вопрос о защите информации как объекта интеллектуальной собственности и имущественных прав на нее. Таким образом, главной целью правового обеспечения является укрепление законности на информационное обеспечение.

Глава 2. Информационные системы

В зависимости от характера решаемых задач (при непременном использовании компьютерных и программных средств) и типа используемых технических устройств сегодня можно создавать специальные виды НИТ, т.н. информационные системы. В качестве подсистемы некой головной системы они призваны обеспечить последнюю необходимой информацией.

Напомним, что систему можно определить как состоящую из подсистем. Те, в свою очередь, делятся на подсистемы и так до подсистем нижнего уровня, которые называют элементами. Так что структура системы есть иерархия вложенных подсистем и совокупность элементов, образующих целостность и единство.

Добавление к понятию "система" слова "информационная" отражает цель создания и функционирования информационной системы.

Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач из любой области.

Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом для информационных систем.

Наконец, выходной продукцией информационной системы является информация, на основе которой принимаются решения.

Информационная система — это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Информационные системы могут быть классифицированы на:

1. Информационные технологии обработки данных для решения структурированных задач, когда известны входные данные, алгоритмы и другие процедуры их обработки. Такие технологии применяются для автоматизации повторяющихся операций в сфере управления и оказания услуг, например, автоматизированная система обслуживания вкладчиков банка.
2. Информационные технологии поддержки принятия решений, Цель таких технологий - удовлетворение информационных потребностей лиц, связанных с принятием решений. В идеале видятся НИТ автоматического решения управленческих задач на основе обработки знаний.

По степени автоматизации принятия управленческих решений различают:

- автоматизированные (т.е. с участием человека) системы: управления (АСУ), проектирования (САПР), обучения (АОС);
- интеллектуальные автоматизированные системы: управления (ИАСУ), проектирования (ИСАПР), обучения (ИАОС) , в которых принят реальный, и открытый к эволюционному развитию интеллектуальности уровень автоматизации;
- автоматические (т.е. без участия лиц, принимающих решения) системы управления, проектирования, обучения в форме экспертных систем.

Целью любой ИС является обработка данных об объектах и предоставление нужной информации о них.

Наиболее примечательной чертой НИТ является их системность. На передний план выступают не отдельные информационные устройства, а системы аппаратного и программного обеспечений.

Информационная система определяется как совокупность средств автоматизированной обработки данных, программного обеспечения, обслуживающего персонала и имеет целью обеспечение компонент НИТ информацией и информационным сервисом. Информация в них понимается как совокупность сведений, воспринимаемых из окружающей среды, выдаваемых в окружающую среду, либо сохраняемых внутри информационной системы.

Внедрение информационных систем может способствовать:

- получению рациональных вариантов решения управленческих задач за счет применения математических методов, моделирования и т.д.;
- освобождению работников от рутинной работы за счет ее автоматизации;
- обеспечению достоверности информации;
- замене бумажных технологий безбумажными;
- совершенствованию структуры потоков информации и системы документооборота;
- уменьшению затрат на производство продуктов и услуг;
- предоставлению потребителям специальных услуг

Фактографические и документальные ИС. Если мы рассмотрим совокупность некоторых объектов, то сможем выделить объекты, обладающие одинаковыми свойствами. Такие объекты выделяют в отдельные классы. Внутри выделенного класса объекты можно упорядочивать как по общим правилам классификации, например по алфавиту, так и по некоторым конкретным общим признакам, например по цвету или материалу. Группировка объектов по определенным признакам значительно облегчает поиск и отбор информации.

Информационные системы можно условно разделить на фактографические и документальные.

В фактографических ИС регистрируются факты, конкретные значения данных (атрибутов) об объектах реального мира. Основная идея таких систем заключается в том, что все сведения об объектах (напр., фамилии людей и названия предметов, числа, даты) сообщаются компьютеру в каком-то заранее обусловленном формате. Информация, с которой работает фактографическая ИС, имеет четкую структуру, позволяющую машине отличать одно данное от другого, например, фамилию от должности человека, дату рождения от роста и т.п. Поэтому фактографическая система способна давать однозначные ответы на поставленные вопросы.

Документальные ИС обслуживают принципиально иной класс задач, которые не предполагают однозначного ответа на поставленный вопрос. Базу данных таких систем образует совокупность неструктурированных текстовых документов (статьи, книги, рефераты и т.д.) и графических объектов, снабженная тем или иным формализованным аппаратом поиска. Цель такой системы - выдать в ответ на запрос пользователя список документов или объектов, в какой-то мере удовлетворяющих сформулированным в запросе условиям.

Указанная классификация ИС в известной мере устарела, так как современные фактографические системы часто работают с неструктурированными блоками информации (текстами, графикой, звуком, видео), снабженными структурированными описателями.

Подсистемы обработки данных.

Важнейшей структурной подсистемой ИС является система обработки данных, т.е. система, состоящая из совокупности данных в виде информационных массивов и набора формальных программных процедур

обработки, обеспечивающих сбор, хранение, выборку, манипуляцию и коммуникацию данных. Возможны два пути формирования информационных массивов: исходя из потребностей отдельных задач (файловая организация), для каждой задачи пишутся программы ввода, сортировки и т.п.; второй путь связан с применением т.н. баз данных, основанных на стандартизации представления информации в системе, использовании стандартного программного обеспечения без жёстких ограничений на организацию хранения и поиск данных. Первый путь в известной мере устарел. Второй путь направлен на создание унифицированного программного обеспечения, пригодного для обработки массивов информации вне зависимости от их конкретного содержания. Исследования такой направленности привели к концепции *баз данных и систем управления базами данных* (СУБД).

База данных представляет собой совокупность хранимых данных, организованных по определенным правилам (в соответствии с выбранной моделью БД); допускает машинную обработку данных различного типа и обеспечивает независимость данных от использующих их программ; позволяет снизить избыточность данных, архивацию их и т.п.

Модели баз данных. БД может быть основана на одной модели или на совокупности нескольких моделей. Любую модель данных можно рассматривать как объект, который характеризуется своими свойствами (параметрами), и над ней, как над объектом, можно производить какие-либо действия.

Благодаря развитию информатизации и компьютеризации, сетевого взаимодействия стали востребованы принципиально новые подходы в хранении, использовании и управлении информацией. Концепция соединенных и совместно использующих ресурсы компьютеров носит название *сетевого взаимодействия*. Сетью называется группа соединенных компьютеров и других устройств. Самая простая сеть (network) состоит как минимум из двух компьютеров, соединенных друг с другом кабелем. Это позволяет им использовать данные совместно. Компонентами сети являются:

- серверы (server) – компьютеры, предоставляющие свои ресурсы сетевым пользователям;
- клиенты (client) – компьютеры, осуществляющие доступ к сетевым ресурсам, предоставляемым сервером;
- среда (media) - способ соединения компьютеров;
- совместно используемые данные;
- совместно используемые периферийные устройства;
- ресурсы – файлы, принтеры и другие элементы, используемые в сети.

В БД в классическом виде была реализована идея централизации. Все вычислительные ресурсы были сконцентрированы в едином комплексе, там же хранились и обрабатывались огромные массивы данных. Достоинства централизованной архитектуры очевидны — это простота защиты информации, администрирования и ряд других.

Однако появление ПК сделало возможным в составе сети компьютеров иметь вычислительные и информационные ресурсы на собственном рабочем месте и управлять ими по собственному разумению.

Появилась идея переноса части системы непосредственно на рабочее место для выполнения на персональном компьютере, поручив ему выполнять ту часть программы, которая отвечает за интерфейс с пользователем, а функции обработки данных оставить на центральном компьютере. Таким образом, система стала распределенной - одна ее часть выполнялась на центральном компьютере, другая — на персональном, который был объединен в сеть с центральным. Подобная модель взаимодействия компьютеров и программ в сети положила начало технологии "клиент-сервер".

Требования корпоративного доступа к ресурсам и появление локальных вычислительных сетей на базе ПК привели к созданию наиболее многочисленных на сегодня решений на базе технологии "клиент-сервер". Часть функций закрепляется за сервером, а часть - за клиентом. На сервер возлагаются обязанности по оперативному исполнению транзакций, поддержке целостности данных, обеспечению безопасности хранения и доступа, обеспечению пользовательских соединений. На стороне клиента остается пользовательский интерфейс.

Итак, управление хранимой информацией производит сервер базы данных, который (помимо самого компьютера, предоставляющего свои ресурсы сетевым пользователям) состоит из пакета программ, утилит и компонентов, и хранилища информации – базы данных. Сервер базы данных принимает команды от клиента – запросы, обрабатывает их, обращаясь к базе данных и, при необходимости, возвращает клиенту результат. Клиентом может являться любая программа, установившая связь с сервером (с помощью компонентов, предоставляемых сервером базы данных).

2. 1. Информационные «Инtranет – системы »

Возросшая потребность корпоративного доступа к информационным ресурсам на фоне развития сетевых компьютерных технологий, появления Интернет и Web-технологий способствовали совершенствованию информационных систем, функционирующих в архитектуре «клиент-сервер». В частности, этапом в эволюции информационных систем стало применение Инtranет в корпоративных информационных системах.

Традиционные системы "клиент-сервер" — это системы, поставляющие данные. Им свойственны:

- на сервере порождаются данные, а не информация;
- клиентам, т.е. ПК из сервера передаются данные, которые там интерпретируются и преобразуются в информацию;
- система является распределенной - одна ее часть исполняется сервером, другая — на персональном компьютере (клиенте);

- для обмена данными между клиентом и сервером используется закрытый корпоративный протокол, специфичный для данной информационной системы.

Недостатки традиционных систем "клиент-сервер" отсутствуют в информационных «Инtranет-системах». Последние объединили в себе лучшие качества централизованных систем и традиционных систем "клиент-сервер".

«Инtranет-системам» свойственны:

- только на сервере порождается конечный продукт — информация в форме, предназначенной для пользователя, а не полуфабрикат в виде данных;
- информация передается на ПК в виде, пригодном для восприятия человеком;
- для обмена информацией между клиентом и сервером используется протокол открытого стандарта.

Может сложиться впечатление, что имеет место возврат к ранним БД - централизованной структуре систем поставки информации. На самом деле имеет место возврат, но уже на качественно ином, более высоком уровне, на новом витке спирали. Здесь клиент - это графический терминал для потребления информации, т.е. сетевой компьютер, снабженный специализированным программным обеспечением — программой навигации. Вся потребляемая информация порождается на сервере. Доступ к информации осуществляется через одну и ту же программу, не требующую локальных данных. Компьютер на рабочем месте целиком настраивается из центра и нет необходимости выполнять какие-то действия по его конфигурированию.

Обобщая вышесказанное, можно следующим образом характеризовать Инtranет технологии. Инtranет — это применение технологии Интернет в рамках корпоративных информационных систем. Благодаря Инtranет предприятие имеет возможность реализовать внутри (intra) своей сети технологии Интернет.

Основной причиной, стимулирующей предприятие создать интрасеть, является необходимость более эффективного управления, сбора и предоставления информации. Встроенные же механизмы безопасности, возможности обеспечения координации работы, доступа к информации сотрудников вне зависимости от уровня (разнородности) их рабочих мест (рабочих станций) делают интрасеть привлекательной.

В условиях интенсификации производства для обеспечения координации работы сотрудников необходимо использование все большего количества документов, инструкций, руководств, меморандумов и прочих документов. Возрастающее количество документов и изменяющихся их версий создает проблемы актуальности и доступности информации. В таких случаях оправданным является применение механизмов публикации документов на Web-серверах компаний. В интрасети любой пользователь с рабочей станции посредством Web-браузера может получить доступ к

любым самым последним версиям документов, как только они будут помещены на Web-сервер.

Использование интрасети (публикация документов на Web-серверах) имеет еще ряд других преимуществ. Первое состоит в том, что документы в сети могут обновляться автоматически, в режиме реального времени. Например, Вы публикуете документ для отдела продаж, содержащий информацию о ценах на продукцию Вашей компании и конкурентов. Для предоставления оптимальных в любой момент времени скидок Ваш отдел продаж должен обладать самой актуальной информацией. Для обеспечения такой актуальности Вы можете создать автоматизированный сценарий (script) на Web-сервере, по которому он будет, например, раз в 15 минут обновлять публикуемый документ со списками текущих цен. Сотрудники отдела продаж автоматически будут получать обновленный список цен для своей работы – им просто не надо будет предпринимать никаких действий по обновлению информации. Выигрыш очевиден – актуальность информации, плюс снижение непроизводительной работы сотрудников.

В дополнение, можно получать немедленную обратную связь об опубликованных документах. Если документы предоставляются сотрудникам в виде файлов хранящихся на файловом сервере, то Вы никогда не сможете быть уверены в том, что информация оказалась прочитанной тем или иным сотрудником. Если же документ опубликован на Web-сервере, то Вы можете в любой момент времени получить сведения о том, кто, когда и сколько раз обращался к опубликованным документам.

Развитие организации постоянно требует новых возможностей в предоставлении и удобстве использования информации ее сотрудниками. И в ответ на эти требования, документы, публикуемые с использованием World Wide Web получили возможность содержать графическую, аудио- и видеоинформацию. Очень большое количество организаций в настоящее время используют приложения, позволяющие осуществлять доступ к базам данных компаний прямо из Web-браузера, убирая необходимость создавать и использовать специализированные программы доступа к этим базам данных. Например, такой продукт, как Novell GroupWise позволяет сотрудникам организации работать со своей электронной почтой и ежедневником прямо из Web-браузера.

Сотрудники получают доступ к информационному хранилищу организации. Они работают быстрее и эффективнее, так как им не нужно звонить по телефонам, бегать по кабинетам в поисках документов, отвлекать коллег от работы, ожидать, пока то или иное предписание поступит в отдел – достаточно лишь запустить навигатор и "перейти" по ссылкам в требуемую точку информационного хранилища.

Еще одним фактором, придающим значимость построению интрасетей, является тот факт, что доступ к опубликованной информации может осуществляться через Интернет. Сотрудники, находящиеся в командировке, поставщики и заказчики могут получать доступ к опубликованной в Вашей интрасети неконфиденциальной информации.

В дидактических целях материал об Инtranет был иллюстрирован примерами из сферы бизнеса. Однако использование Инtranет в других производственных или социальных системах возможно и это сулит большие выгоды.

2. 2. Системы искусственного интеллекта

В большинстве современных систем управления остро стоит вопрос о придании им свойства принятия решений, т.е. реализация в них каких-то моделей принятия решений естественным интеллектом. Методы решения этой проблемы сформировались в научно-техническое направление, называемое искусственный интеллект (ИИ).

Наибольший интерес представляют кибернетические модели ИИ, т.е. такие, в которых осуществляется переработка информации с целью принятия решений. Так что ИИ - некоторая модель процесса принятия решений человеком.

Под системой искусственного интеллекта (СИИ) подразумевается система, заменяющая естественный интеллект в профессиональных сферах. СИИ может быть представлена тремя блоками: модель внешнего мира (ВшП), внутреннее представление (ВтП) и преобразователя из ВшП во ВтП.

ВтП - особенность СИИ. Центральное место в ВтП занимает интеллектуальный банк данных (ИБД), в котором сосредоточена вся информация о внешнем мире. Условно ИБД разделен на базу целей (БЦ), базу знаний (БЗ), базу данных (БД). БЦ содержит информацию о целевом поведении ПО и обеспечивает побуждения к действию. БЗ содержит закономерности ПО, позволяющие выводить новые факты, сведения о структуре БД, а также сведения, необходимые для перевода входных запросов во внутренний язык.

Интеллектуальность предполагает наличие нижеследующих признаков:

1. Наличие в системе собственной внутренней модели внешнего мира. Это дает возможность семантической и прагматической интерпретации запроса.
2. Возможность пополнения, изменения, удаления знаний в рамках понятия ведения баз знаний, а также используются идеи вывода новых знаний и расширения за этот счет баз знаний.
3. Возможность вывода, генерации решений. Традиционно решение задачи предполагает наличие: постановки задачи, целевой установки и способа перехода от первого ко второму. В нашем же случае предполагается наличие лишь первых двух компонентов. Интеллектуальная система сама автоматически вырабатывает алгоритм и генерирует соответствующую рабочую программу.
4. Разрешение проблемы нечеткости и размытости самого процесса принятия решений. Неопределенность лежит в основе принятия решений.

Классификация СИИ.

ИИ как научное направление имеет целью воссоздание с помощью ЭВМ “разумных рассуждений и действий”.

Исследования по проблемам ИИ выявили 2 направления:

- попытка моделировать функции мозга;
- разработка методов и систем, позволяющих усилить деятельность человека.

Математические модели, широко применяемые в информатике, универсальны. Но в этом и недостаток матмоделей, так как они не содержат в себе конкретные знания об объекте. Так что смысл (семантика) модели остается в голове конечного пользователя (КП). Отсюда (при традиционной технологии) необходимость в диалоге : КП - математик - ЭВМ.

Разработка логико-лингвистических моделей (ЛЛМ) привела к становлению новой технологии использования ЭВМ. ЛЛМ носят семантический характер и конкретны. Конкретность выражается в описательной (вербальной) форме ЛЛМ. В плохо формализованных областях ЛЛМ заменяют матмодели. Именно ЛЛМ привели к появлению баз знаний (БЗ).

Данные и знания. В информатике известен термин «данные». В процессе эволюции понятие данных усложнилось. Можно условно выделить вехи такой эволюции:

1. Внутренняя интерпретируемость. Раньше содержимое данных было доступно лишь программисту. Создание баз данных позволило ЭВМ отвечать на запросы по поводу содержимого её памяти, т.е. появилась возможность внутренней интерпретации данных.
2. Наличие структуры связей данных, в том числе внутренней структуры связей, характерной для фреймов и внешней структуры связей, характерной для семантической сети.
3. Шкалирование. Связность данных задается не только отношениями, но и задавая различные шкалы. Наиболее простая из шкал - это метрическая шкала. С помощью шкал можно устанавливать соотношения и порядок данных.
4. Активность. Традиционно программа играла роль активатора данных. В СИИ знания (данные) являются активатором для программы.

Таким образом, налицо процесс усложнения данных и эволюционное превращение их в знания.

В исследованиях по ИИ можно выделить 4 области:

1. Системы, имитирующие отдельные творческие процессы. Это направление включает в себя составление программ для решения игровых задач, доказательства теорем, распознавание изображений и др.
2. Системы, основанные на знаниях. Это важная с точки зрения практики и бурно развивающаяся область ИИ. Использование ЛЛМ

обеспечивает результативность и эффективность автоматизированных систем и повышает их до уровня интеллектуальных АСУ, САПР, АОС и т.д.

3. Новая архитектура ЭВМ. Это область машин пятого поколения. Благодаря наличию в них интеллектуального интерфейса КП получает возможность непосредственного общения с ЭВМ, а не через плечо программиста.
4. Интеллектуальные роботы. Это область третьего поколения роботов - автономных транспортных средств, способных совершать операции по достижению целей, поставленных человеком. Они ориентированы на использование знаний о внешней среде и так как они ситуативны, то требуется обработка информации в реальном масштабе времени.

Модели представления знаний.

Важной составной частью СИИ является модель представления знаний. Модель представления знаний (МПЗ) о ПО сообщается компьютеру посредством специальным образом организованных структур данных. Разнообразие возможных при этом структур и методов их обработки порождают проблему выбора в некотором смысле лучшей модели представления внешнего мира. Одновременно разработка СИИ предполагает сделать знания объектом обработки на компьютере. Для этого необходимо также решить проблему перехода от внешнего представления объектов ПО во внутреннее по отношению к компьютеру. При этом внутренняя модель мира должна порождать информацию, которой свойственны: внутренняя интерпретируемость, связность, активность, характеризующие эволюцию данных знания.

Традиционно различают *декларативный* и *процедуральный* способы представления знаний. Декларативному способу свойственно отделение синтаксических знаний от семантических, а описание состояний представляет собой множество утверждений, не имеющих глубоких связей с конкретной ПО. Такую универсальность декларативного способа представления можно расценивать в целом как положительный момент, но независимость процедуры поиска решений от ПО не позволяет использовать семантический аспект знаний. Процедуральный способ реализует описание состояний в виде процедур, в которые вложены знания о ПО. При необходимости такие процедуры позволяют генерировать все возможные состояния, а использование специфики ПО позволяет существенно сократить поиск решений. Процедуральный способ приспособлен к наращиванию знаний об объекте, что во многом предопределяет мощность СИИ.

Многообразие моделей представления знаний можно разделить на логические и эвристические.

Логическая модель базируется на аппарате математической логики, исчислении высказываний и предикатов. Она основана на формализмах, включающих понятия алфавита символов, конечные последовательности которых образуют выражения. Некоторые подмножества последних

составляют формулы теории. Процедура построения формул имеется. Подмножество априорно истинных формул называют аксиомами. Конечное множество отношений между формулами является правилами вывода. Имеется последовательность формул, каждая из которых есть либо аксиома, либо следствие некоторых формул. Языки логического типа опираются в основном на исчисление предикатов. Знания здесь представляются как некоторый набор предикатов. Достоинством исчисления предикатов в представлении знаний является наличие набора мощных процедур вывода, доказательства теорем, а также возможность пополнения текстовых описаний др. Одновременно ориентированность исчисления предикатов при интеллектуализации на ограниченное число мощных правил вывода затрудняет представление многообразных и зависимых знаний ПО. Включение компонент адаптивного изменения правил в семиотических моделях призваны ослабить недостатки логических моделей. Отметим также, что инструментарий представления и пополнения знаний не удобен для конечного пользователя. Наиболее распространенными языками программирования, основанными на логической модели представления, являются ЛИСП и ПРОЛОГ.

Группу эвристических моделей представления знаний составляют сетевые, реляционные, фреймовые, продукционные и др. модели.

Предметная область в сетевой модели представления знаний рассматривается как совокупность сущностей (объектов) и отношений (связей) между ними. Объекты при этом являются вершинами (узлами) сети, а отношения, направленными поименованными ребрами. Таким образом система знаний отображается сетью ориентированным графом, составленным из поименованных вершин и ребер. Ребро и связываемые им вершины образуют подграф сети, несущий минимальную информацию, факт наличия связи определенного типа между соответствующими объектами. В данной модели представления важна классификация типов связей. Реализация модели базируется на использовании бинарных отношений. Для поиска и манипулирования используются операции сопоставления с образцом. Запрос представляется в виде сети и поиск ответа на запрос реализуется сопоставлением запроса с фрагментом семантической сети.

Несмотря на очевидные достоинства этой модели, такие как большие выразительные способности, естественность системы знаний, близость структуры сети к семантической структуре фраз естественного языка, сильным ограничивающим фактором ее использования является сложность методов обработки информации в сети и получение правдоподобного вывода, что вызывает трудности в создании соответствующего программного обеспечения.

Основной элемент информации в реляционной базе называется записью. Каждая запись имеет несколько атрибутов и соответствующих им значений. Множество записей одинакового формата образуют таблицу или отношение. Множество отношений образуют базу данных. В результате операции поиска нужная информация извлекается из базы данных. Язык

манипулирования дает возможность описывать поиск и ведение базы данных.

Фрейм достаточно жесткий фрагмент семантической сети иерархического вида, предназначенный для описания объекта (ситуации) среды. Все, что касается объекта или ситуации и важно для решаемых задач, не размывается по сети, а представляется во фрейме. Фрейм состоит из слотов, каждый из которых в свою очередь может быть фреймом. Для обработки фрейма используется его “ассоциированная” процедура, что позволяет использовать некоторые важные механизмы представления знаний и, в первую очередь, процедурные знания. Концептуальная установка фреймовых моделей представления позволяет наиболее полно реализовать характерные для знаний структурированность, внутреннюю интерпретируемость, связанность и активность. Может быть, в этом причины популярности фреймов. Однако будем помнить, что фреймовая модель является семантической сетью, а наличие слотов у фреймов дает последним возможность ссылаться друг на друга, чем сильно усложняются и порождаются проблемы движения в такой сети.

В случаях отсутствия формализованности знаний они могут быть реализованы в некоторых процедурах решения задач, в том числе и в программах ЭВМ. Продукционные модели представления содержат указание на такие процедуры и условия их применения. Основной конструкцией этих моделей является форма: ЕСЛИ <условие>, ТО <действие>.

В условии задается совокупность фактов, представляющих конкретную ситуацию, и при выполнении данного условия производится действие. Правила такого вида могут производить операции над самими правилами. В этом случае такие правила называют метаправилами. Преимущество этой модели в простоте формирования базы знаний на полноту и непротиворечивость.

Для систем искусственного интеллекта характерна активность данных, сообщенных системе заранее или возникающих в результате выполнения предыдущего шага. Эту особенность СИИ наиболее полно учитывают модели представления в виде управляемых образцами модулей. Сущность таких моделей представления знаний состоит в следующем. Создается множество модулей, обеспечивающих решение класса задач выбранной ПО, а также рабочая память для хранения временных знаний. С каждым модулем ассоциируется выражение образец, являющийся целью применения модуля. Модули, имеющие деление на предусловия (исследование базы данных) и действия модификации данных, называют правилами. На каждом шаге данные рабочей памяти сопоставляются с образцом модуля. Исходное выражение при этом воспринимается правилом (например, условное выражение ЕСЛИ условие, ТО действие), сопоставляется с его образцом, являющимся левой частью этого правила и преобразует его, представляя полученные при сопоставлении значения переменных в правую часть правила. Концепция моделей представления знаний в виде управляемых образцами модулей имеет ряд преимуществ

перед другими. Однако заметим, что эти преимущества обретают силу, если задействована одновременно эффективная система автоматизации программирования (вплоть до автоматической генерации рабочих программ).

2. 3. Экспертные системы

Искусственный интеллект (ИИ) имеет более чем 50 летнюю историю. В начале 70-х годов в области ИИ имел место следующий концептуальный прорыв: эффективность программы для решения задач зависит от знаний, которыми она обладает, а не только от формализмов и схем вывода.

Чтобы сделать программу интеллектуальной, её нужно снабдить множеством высококачественных специальных знаний о предметной области (ПО). Понимание этого факта привело к развитию специализированных программных систем, каждая из которых является экспертом в ПО. Такие программы получили название экспертных систем (ЭС). Последние сегодня во многом определяют развитие систем искусственного интеллекта (СИИ).

ЭС позволяют использовать компьютер там, где формализация с помощью математических моделей затруднена или невозможна.

В ЭС ЛЛМ приобретают решающую роль. Наиболее распространена в ЭС такая разновидность ЛЛМ как продукция, т. е. правила вида ЕСЛИ <условие>, ТО <действие> .

Важно понять как организованы и представляются знания в ЭС. Знания понимают как информацию, которая необходима программе, чтобы она вела себя “интеллектуально”. Эта информация принимает форму **фактов** или **правил**, причём, факты играют роль элементарных «единиц знаний» (т.е. простых утверждений о характеристиках объектов), а правила служат для выражения связей, зависимостей между фактами.

Знания в ЭС организованы так, чтобы знания о ПО отделить от других типов знаний системы, например, таких как общие знания о том как решать задачи и др.

Выделенные знания о ПО называют **базой знаний**, тогда как общие знания о нахождении решений задач называют **механизмом вывода**.

База знаний ЭС содержит факты (данные) и правила, использующие эти факты как основу для принятия решений.

Механизм вывода содержит *интерпретатор*, определяющий каким образом применять правила и *диспетчер*, устанавливающий порядок применения этих правил.

Отсутствие простого и общего механизма вывода решений затрудняет конструирование и использование его.

Некоторые языки высокого уровня, предназначенные для построения ЭС, имеют механизм вывода, в некотором смысле встроенный в язык как его часть. Другие языки требуют, чтобы создатель ЭС спроектировал и реализовал механизм вывода.

Оба подхода имеют свои недостатки и достоинства. Первый облегчает работу создателя ЭС . В тоже время у него меньше свободы выбора. Второй

позволяет учесть специфику ПО, что повышает эффективность поиска решений.

В ЭС, основанных на правилах, предметные знания представляются набором правил, которые проверяются на группе фактов или знаний о текущей ситуации. Когда левая часть правила удовлетворяет фактам, то указание в части «То» выполняется. В этом случае говорят, что правило выполнено.

Процесс сопоставления с фактами частей «Если» правил может порождать **цепочку вывода**.

Существует два важных способа использования правил: *прямая цепочка рассуждений* и *обратная цепочка рассуждений*.

Глава 3. Базовые информационные технологии

В формировании НИТ трудно переоценить роль технических средств. Вместе с тем в становлении и совершенствовании НИТ важное значение принадлежит программным средствам. Их симбиоз с соответствующими техническими устройствами породил множество информационных технологий. Мы различаем **базовые и прикладные** информационные технологии.

Базовые информационные технологии существуют «сами по себе», изначально создаются для сбора, накопления, переработки и передачи различных видов информации без конкретного указания её происхождения. К их числу можно отнести: технологии обработки текстов; презентационные технологии; технологии электронных таблиц; технологии распознавания; технологии баз данных; гипертекст технологии; сканер технологии; технологии перевода с одного языка на другой; технологии перевода текста с одной графики на другую; Zip-драйвер технологии; сетевые компьютерные технологии; аудио-технологии; видео-технологии; технологии цифрового фото и др.

Базовые информационные технологии в зависимости от их комплексации образуют иерархию. Вышеперечисленные технологии можно условно отнести к базовым технологиям нижнего уровня. В качестве примера рассмотрим несколько базовых ИТ нижнего уровня.

3.1. Технологии архивирования данных

При эксплуатации компьютера по самым разным причинам возможна порча или потеря информации на магнитных дисках. Это может произойти из-за физической порчи магнитного диска, неправильной корректировки или случайного уничтожения файлов, разрушения информации компьютерным вирусом и т.д. Для того чтобы уменьшить потери в таких ситуациях, следует иметь архивные копии используемых файлов и систематически обновлять копии изменяемых файлов. Для сохранения копий можно использовать

обычные команды копирования. Однако при этом архивные копии занимают столько же места, сколько занимают исходные файлы.

Проблемы больших объёмов данных возникают при переносе информации с одного компьютера на другой. Проблемы обостряются при пересылке информации средствами электронной почты, когда увеличение объёма данных не только вызывает рост затрат на оплату времени передачи, но и уменьшает надёжность передачи: чем больше файл, тем больше вероятность сбоя на телефонной линии.

Можно ли уменьшить объём файла, не теряя содержащуюся в нём информацию? Уменьшить объём файла, не теряя содержащуюся в нём информацию, призваны программы, называемые архиваторами. Архиватором (упаковщиком) называется программа, позволяющая за счёт применения специальных методов сжатия информации создавать копии файлов меньшего размера, также объединять копии нескольких файлов в один *архивный* файл, из которого можно при необходимости извлечь файлы в их первоначальном виде.

Самый простой метод архивирования информации: если мы (зная заранее или проанализировав содержащийся в файле текст) имеем список типовых слов и заменяем каждое из них на соответствующий ему байт (или на двухбайтовое число), то объём текста заметно сократится при сохранении возможности преобразования «закодированного» текста в исходный вид. Чтобы уменьшить размер графического файла, можно с помощью некоей программы-конвертора переписать его содержимое, заменяя подобные последовательности всего двумя байтами, первый из которых представляет собой «образец», а второй – количество одинаковых байтов в последовательности в исходном файле. Очевидно, что при этом файлы рисунков, в значительной мере состоящих из фона, могут сжаться довольно ощутимо. В случае необходимости другая программа-конвертор расшифрует содержимое файла, считав информацию из закодированного файла и повторив байт-образец нужное число раз.

Для создания архивных копий удобно использовать архиваторы, которые позволяют не только сэкономить место на архивных дискетах, но и объединять группы совместно используемых файлов в один архивный файл, что позволяет гораздо легче разбираться в архиве файлов. Архивный файл представляет собой набор из одного или нескольких файлов, помещенных в сжатом виде в единый файл, из которого их можно при необходимости извлечь в первоначальном виде. Архивный файл содержит оглавление, позволяющее узнать, какие файлы содержатся в архиве.

Различными разработчиками были созданы специальные программы для архивации файлов: ARJ, ZIP и т.д. Разные программы отличаются форматом архивных файлов, скоростью работы, степенью сжатия файлов, при помещении в архив, удобством пользователя. Достаточно популярные архиваторы WinZIP, WinRAR представляют собой полноценные Windows приложения. Они имеют типовой оконный интерфейс. Например, WinZIP корректно работает с длинными именами файлов, а благодаря богатым

возможностям по интеграции установленных приложений WinZIP может быть вызван не только из меню *Пуск* (как и всякое другое приложение), но и двойным щелчком на пиктограмме архивного файла (больше всего она похожа на зажатый в штангенциркуле кубик), а так же через всплывающее при щелчке правой кнопкой мыши на пиктограмме файла или папки дополнительное меню, в которое при инсталляции WinZIP добавляется соответствующий новый пункт. Формат ZIP является одним из стандартных в сети Интернет.

3.2 Вирус – технологии

Компьютерные вирусы были и остаются одной из наиболее распространенных причин потери информации. Термин «компьютерный вирус» впервые употребил сотрудник Лехайского университета (США) Ф.Коэн в 1984 г. на 7-й конференции по безопасности информации, проходившей в США.

Что же такое компьютерный вирус? Компьютерный вирус – это специально написанная компьютерная программа, которую сознательно «внедряют» в другие программы. Программа, внутри которой находится вирус, называется «заражённой». Когда такая программа начинает работу, то сначала управление получает вирус. Вирус находит и «заражает» другие программы, а также выполняет какие-нибудь вредные действия (например, портит файлы или таблицу размещения файлов на диске, «засоряет» оперативную память и т.д.). Для маскировки вируса действия по заражению других программ и нанесению вреда могут выполняться не всегда, а, скажем, при выполнении некоторых условий. Если не предпринимать мер по защите от вируса, то последствия заражения компьютера могут быть очень серьезными.

По среде обитания вирусы можно разделить на: файловые, загрузочные, макро, сетевые.

Файловые вирусы либо различными способами внедряются в выполняемые файлы (наиболее распространенный тип вирусов), либо создают файлы-двойники.

Загрузочные вирусы записывают себя либо в загрузочный сектор диска (boot-сектор), либо в сектор, содержащий системный загрузчик винчестера (Master Boot Record), либо меняют указатель на активный boot-сектор.

Макро-вирусы заражают файлы-документы и электронные таблицы нескольких популярных редакторов.

Сетевые вирусы используют для своего распространения протоколы или команды компьютерных сетей и электронной почты.

Другой тип вредоносных программ - это программы шпионы. Троян – это программа, предоставляющая удаленный доступ к чужому компьютеру, позволяющая проводить различные манипуляции на нем, отсылать конфиденциальную информацию (пароли, номера кредитных карт, Интернет - аккаунты, логин компьютера и т.д.). Трояны, как правило, не заражают другие файлы, а являются автономными отдельными программами, которые

после попадания на компьютер, копируют себя в системные папки под неприметными или заслуживающими доверия именами (winrun32.dll.exe). Затем они прописывают себя в ветви реестра отвечающей за запуск программ при загрузке операционной системы.

Каким бы ни был вирус, пользователю необходимо знать основные методы защиты от компьютерных вирусов. Для защиты от вирусов можно использовать профилактические меры, позволяющие уменьшить вероятность заражения вирусом. Несмотря на то, что общие средства защиты информации очень важны для защиты от вирусов, все же их недостаточно. Необходимо применение специализированных антивирусных программ.

Антивирусные программы можно разделить на: детекторы, доктора (фаги), ревизоры, доктора-ревизоры, фильтры и вакцины (иммунизаторы).

Программы-детекторы позволяют обнаруживать файлы, зараженные одним из нескольких известных вирусов. Эти программы проверяют, имеется ли в файлах на указанном пользователем диске специфическая для данного вируса комбинация байтов. При ее обнаружении в каком-либо файле на экран выводится соответствующее сообщение. Многие детекторы имеют режимы лечения или уничтожения зараженных файлов. Следует подчеркнуть, что программы-детекторы могут обнаруживать только те вирусы, которые ей известны. Большинство программ-детекторов имеют функцию «доктора», т.е. они пытаются вернуть зараженные файлы или области диска в исходное состояние. Те файлы, которые не удалось восстановить, как правило, делаются неработоспособными или удаляются. К программам-детекторам относится, например, Norton AntiVirus.

С помощью программы-ревизора можно в любой момент сравнить состояние программ и системных областей дисков с исходными. О выявленных несоответствиях сообщается пользователю. Чтобы проверка состояния программ и дисков проходила при каждой загрузке операционной системы, необходимо включить команду запуска программы-ревизора в командный файл AUTUEXEC.BAT. Это позволяет обнаружить заражение компьютерным вирусом, когда оно еще не успело нанести большого вреда. Более того, та же программа-ревизор сможет найти поврежденные вирусом файлы.

Традиционные антивирусные программы эффективны в обнаружении известных вирусов и некоторых троянских программ, но не справляются с нарастающим валом современных атак. Поэтому можно говорить о необходимости дополнения антивирусных средств следующими новыми технологиями.

Antivirus

Антивирусные системы - наиболее распространенные средства защиты настольных компьютерных систем, которые используются в начальной фазе жизненного цикла атаки.

Virus Prevention System

Технология предотвращения вирусных атак (Virus Prevention System, VPS), в отличие от традиционных антивирусных средств, решает проблему

борьбы с неизвестными вирусами. Это технология следующего поколения, которая дополняет обычные антивирусные механизмы и анализирует поведение программ до их выполнения в информационной системе на предмет обнаружения и блокирования враждебных действий.

Application Control

Следующий вид защиты обеспечивает технология контроля деятельности приложений (Application Control, AC), завершающая линейку средств информационной защиты уровня приложений. Технология AC может предотвращать атаки в фазе их выполнения или раньше.

Следующий этап защиты это **ФАЙЕРВОЛЛ**:

Персональный межсетевой экран (Personal Firewall, PFW) - наиболее распространенная и понятная форма защиты настольных компьютерных систем для подавляющего числа пользователей. Благодаря набору правил фильтрации пакетов PFW может уменьшить, но не ликвидировать риск того, что компьютер подвергнется нападению из внешней сети. Путем блокирования доступа к портам, IP-адресам, сетевым протоколам и службам PFW-технология может предотвращать только небольшое число хорошо известных атак (SYNFlood, IPSpoofing, Ping of Death, WinNuke и т.д.). Несмотря на свои бесспорные достоинства, технологии межсетевого экранирования не справляются с новыми, постоянно развивающимися угрозами, примерами реализации которых можно назвать известные всем атаки: Nimda, Code Red, Slammer и т.д.

В Windows имеется свой файервол так называемый брандмауэр. Если на компьютере установлен пакет обновления Service Pack 2 (SP2) для Windows XP, новый брандмауэр Windows по умолчанию включен. Для защиты компьютера от вирусов и других угроз нарушения безопасности рекомендуется не выключать его.

Наиболее эффективный и распространенный антивирус и файервол на сегодня - это Kaspersky Internet Security 6.

Kaspersky Internet Security 6.0 обеспечивает комплексную защиту персонального компьютера и рабочих станций, работающих под управлением операционной системы Windows, от всех типов электронных угроз — и вредоносных программ, и хакерских атак, и нежелательных электронных рассылок.

По своим возможностям и эффективности заслуживает внимания антивирусная система NOD 32.

3.3. Технологии БД и СУБД

Как было отмечено выше, в компьютеризованных информационных технологиях есть проблема организации хранения данных и программного управления информацией. Исторически складывалось так, что программы управления данными разрабатывались применительно к каждой задаче, т. е. создавались свои файлы, свои программы ввода, сортировки, выдачи результатов. В результате: сколько задач столько файлов и наборов

программ; избыточность данных; большое число носителей и проблемы их использования; рост объёмов программирования. Таковы издержки файловой организации данных. Чтобы избежать эти проблемы пошли по пути унифицированного представления информации и унифицированного ПО, пригодного для обработки данных вне зависимости от их конкретного содержания.

База Данных (БД) обеспечивает хранение информации и представляет собой совокупность данных, организованных по определенным правилам. БД позволяет структурировать, хранить и обрабатывать данные различного типа. В широком смысле о базе данных можно сказать, что это совокупность сведений о конкретных объектах реального мира. Умение работать с базами данных сегодня является одним из важнейших навыков в работе с компьютером.

Модели баз данных

БД может быть основана на одной модели или на совокупности нескольких моделей. Любую модель данных можно рассматривать как объект, который характеризуется своими свойствами (параметрами), и над ней, как над объектом, можно производить какие-либо действия.

Существуют три основных типа моделей данных – реляционная, иерархическая и сетевая.

Реляционная модель

Термин «реляционный» (от латинского *relatio* – отношение) указывает прежде всего на то, что такая модель хранения данных построена на взаимоотношении составляющих ее частей. В простейшем случае она представляет собой двухмерный массив или двухмерную таблицу, а при создании сложных информационных моделей составит совокупность взаимосвязанных таблиц. Каждая строка такой таблицы называется записью, а столбец – полем.

Реляционная модель данных имеет следующие свойства:

- Каждый элемент таблицы – один элемент данных.
- Все поля в таблице являются однородными, т.е. имеют один тип.
- Каждое поле имеет уникальное имя.
- Одноковые записи в таблице отсутствуют.
- Порядок записей в таблице может быть произвольным и может характеризоваться количеством полей, типом данных.

Иерархическая модель

Иерархическая модель БД представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое дерево (граф). Данная модель характеризуется такими параметрами, как уровни, узлы, связи. Принцип работы модели таков, что несколько узлов более низкого уровня соединяются при помощи связи с одним узлом более высокого уровня.

Узел – информационная модель элемента, находящегося на данном уровне иерархии.

Свойства иерархической модели данных:

- Несколько узлов низшего уровня связано только с одним узлом высшего уровня.
- Иерархическое дерево имеет только одну вершину (корень), не подчиненную никакой другой вершине.
- Каждый узел имеет свое имя (идентификатор).
- Существует только один путь от корневой записи к более частной записи данных.

Сетевая модель

Сетевая модель БД похожа на иерархическую. Она имеет те же основные составляющие (узел, уровень, связь), однако характер их отношений принципиально иной. В сетевой модели принята свободная связь между элементами разных уровней.

Основы разработки базы данных

Прежде чем приступать к работе с базой данных, в первую очередь необходимо выбрать модель представления данных. Она должна отвечать следующим требованиям:

- Наглядность представления информации;
- Простота ввода информации;
- Удобство поиска и отбора информации;
- Возможность использования информации, введенной в другую базу;
- Возможность быстрой перенастройки базы данных (добавление новых полей, новых записей, их удаление).

Компьютерную базу данных можно создать несколькими способами:

- С помощью алгоритмических языков программирования, таких как Basic, Pascal, C++ и т.д. Данный способ применяется для создания уникальных баз данных.
- С помощью прикладной среды, например Visual Basic. С его помощью можно создавать базы данных, требующие каких-то индивидуальных особенностей построения.
- С помощью специальных программных сред, которые называются системами управления базами данных (СУБД).

СУБД называют программную систему, предназначенную для создания в компьютере общей БД, используемой для решения множества задач. Подобные системы служат для поддержания БД в актуальном состоянии и обеспечивают эффективный доступ пользователей к содержащимся в ней данным в рамках предоставленных пользователям полномочий. Языковые средства современных СУБД включают:

- язык описания данных, предназначенный для описания логической структуры данных;
- язык манипулирования данными, обеспечивающий выполнение

основных операций над данными – ввод, модификации и выборку;

- структурированный язык запросов (SQL), обеспечивающий управление структуры БД и манипулирования данными, а также являющийся стандартным средством доступа к удаленным БД ;
- язык запросов по образцу (QBE), обеспечивающий визуальное конструирование запросов к БД.

Ниже представлены основные этапы обобщенной технологии работы с СУБД:

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

При формировании новой таблицы БД работа с СУБД начинается с создания структуры таблицы. Этот процесс включает определение перечня полей, из которых состоит каждая запись таблицы, а также типов и размеров полей. Практически все используемые СУБД хранят данные следующих типов: текстовый, числовой, календарный, логический, примечание. СУБД, предназначенные для работы в Windows, могут формировать поля типа объекта OLE, которые используются для хранения рисунков, графиков, таблиц. Создание структуры таблицы не связано с заполнением таблиц данными, поэтому эти две операции можно разнести во времени.

Практически все СУБД позволяют вводить и корректировать данные в таблице двумя способами:

- с помощью представляемой по умолчанию стандартной формы в виде таблиц;
- с помощью экранных форм, специально созданных для этого пользователем.

СУБД, работающие с Windows, позволяют вводить в созданные экранные формы рисунки, узоры, кнопки. Возможно построение форм, наиболее удобных для работы пользователя, включающих записи различных связанных таблиц БД.

Обрабатывать информацию, содержащуюся в таблицах БД, можно путем использования запросов. Запрос представляет собой инструкцию на отбор записей. Большинство СУБД разрешают использовать запросы следующих типов:

- запрос–выборка, предназначенный для отбора данных, хранящихся в таблицах, и не изменяющих эти данные;
- запрос–изменение, предназначенный 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, позволяющий 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.

Практически любая СУБД позволяет вывести на экран и принтер информацию, содержащуюся в БД, из режимов таблицы или формы. Каждый пользователь, работающий с СУБД, имеет возможность использования специальных средств построения отчетов для вывода данных. Используя специальные средства создания отчетов, пользователь получает следующие дополнительные возможности вывода данных:

- включать в отчет выборочную информацию из таблиц БД;
- добавлять информацию, не содержащуюся в БД;
- при необходимости выводить итоговые данные на основе информации БД;
- располагать выводимую в отчет информацию в любом удобном для пользователя виде (вертикальное или горизонтальное расположение полей);
- включать в отчет информацию из разных связанных таблиц БД.

В процессе создания компьютерной модели можно выделить некоторые стадии, типичные для любой СУБД.

Стадия 1. Запуск СУБД, создание нового файла базы данных или открытие созданной ранее базы.

В процессе выполнения данной стадии необходимо запустить СУБД, создать новый файл (новую базу) или открыть существующую.

Стадия 2. Создание исходной таблицы или таблиц.

Создавая исходную таблицу, необходимо указать имя и тип каждого поля. Имена полей не должны повторяться внутри одной таблицы. В процессе работы с БД можно дополнять таблицу новыми полями. Созданную таблицу необходимо сохранить, дав ей имя, уникальное в пределах создаваемой базы.

Стадия 3. Создание экранных форм.

Первоначально необходимо указать таблицу, на базе которой будет создаваться форма. Ее можно создавать при помощи Мастера форм или самостоятельно, указав, какой вид она должна иметь (например, в виде столбца или таблицы). При создании формы можно указывать не все поля, которые содержит таблица, а только некоторые из них. Имя формы может совпадать с именем таблицы, на базе которой она создана. На основе одной таблицы можно создать несколько форм, которые могут отличаться видом или количеством используемых из данной таблицы полей. После создания форму необходимо сохранить. Созданную форму можно редактировать, изменяя местоположение, размеры и формат полей.

Стадия 4. Заполнение БД.

Процесс заполнения БД может проводиться в двух видах: в виде таблицы и в виде формы. Числовые и текстовые поля можно заполнять в виде таблицы, а поля типа МЕМО и ОЛЕ – в виде формы.

Работа с созданной базой данных

Работа с БД включает в себя такие действия, как:

- Поиск необходимых сведений;
- Сортировка данных;
- Отбор данных;
- Вывод на печать;

По характеру использования СУБД разделяют на многопользовательские и персональные.

Многопользовательские СУБД позволяют создавать информационные системы, функционирующие в архитектуре «клиент-сервер». К многопользовательским СУБД относятся Microsoft SQL server, Oracle, InterBase, SyBase, Informix и др.

Персональная СУБД обеспечивает возможность создания локальных БД, работающих на одном компьютере. К персональным СУБД относятся Paradox, dBase, Foxpro, Access и др.

Рассмотрим все этапы создания и принципы работы с базами данных на примере СУБД Microsoft Access.

3. 4. Технологии СУБД ACCESS

Продолжая обсуждение проблем СУБД, необходимо прежде всего договориться о терминологии данных: запись, таблицы, столбец, строка, формы, страницы, макросы, модули, запросы, отчеты. Пусть, например, требуется хранить сведения о деталях, поступивших на склад. Деталь как объект реального мира будет отображена в базе данных? Для того, чтобы ответить на этот вопрос, необходимо знать, какие признаки или стороны детали будут актуальны, необходимы для работы. Среди них могут быть название детали, ее вес, размер, цвет, дата изготовления, материал, из которого она сделана и т.д. В традиционной терминологии объекты реального мира, сведения о которых хранятся в базе данных, называются сущностями, а их актуальные признаки – атрибутами.

Каждый признак конкретного объекта есть значение атрибута. Так, деталь "двигатель" имеет значение атрибута "вес", равное "50", что отражает тот факт, что данный двигатель весит 50 килограммов.

Было бы ошибкой считать, что в базе данных отражаются только физические объекты. Она способна вобрать в себя сведения об абстракциях, процессах, явлениях, то есть обо всем, с чем сталкивается человек в своей деятельности. Так, например, в базе данных можно хранить информацию о заказах на поставку деталей на склад (хотя он суть не физический объект, а процесс). Атрибутами сущности "заказ" будут название поставляемой детали, количество деталей, название поставщика, срок поставки и т.д.

Объекты реального мира имеют друг с другом множество сложных связей. Однако этого недостаточно, так как могут понадобиться

дополнительные сведения о производителе конкретной детали - его адрес, номер телефона и т.д. Следовательно, база данных должна содержать не только сведения о деталях и заказах на поставку, но и сведения о их производителях. Более того, база данных должна отражать связи между деталями и производителями (каждая деталь выпускается конкретным производителем) и между заказами и деталями (каждый заказ оформляется на конкретную деталь). Отметим, что в базе данных нужно хранить только актуальные, значимые связи.

Таким образом, в широком смысле слова база данных - это совокупность описаний объектов реального мира и связей между ними, актуальные для конкретной прикладной области. В дальнейшем мы будем исходить из этого определения, которое будет уточняться по ходу изложения. Итак, мы получили представление о том, что хранится в базе данных. Теперь необходимо понять, как сущности, атрибуты и связи отображаются на структуре данных. Это определяется моделью данных.

Традиционно все СУБД классифицируются в зависимости от модели данных, которая лежит в их основе. Принято выделять иерархическую, сетевую и реляционную модели данных. Соответственно говорят об иерархических, сетевых, реляционных СУБД. По распространенности и популярности реляционные СУБД сегодня вне конкуренции. Она была разработана Коддом еще в 1969-70 годах на основе математической теории отношений и опирается на систему понятий, важнейшими из которых являются таблица, отношение, строка, столбец, первичный ключ, внешний ключ.

Реляционной считается такая база данных, в которой все данные представлены для пользователя в виде прямоугольных таблиц значений данных, и все операции над базой данных сводятся к манипуляциям с таблицами. Таблица состоит из строк и столбцов и имеет имя, уникальное внутри базы данных. Таблица отражает тип объекта реального мира (сущность), а каждая ее строка - конкретный объект. Так, таблица «Деталь» содержит сведения о всех деталях, хранящихся на складе, а ее строки суть набор значений атрибутов каждой конкретной детали. Каждый столбец таблицы - это совокупность значений конкретного атрибута объекта. Так, столбец «Материал» представляет собой множество значений "Сталь", "Олово", "Цинк", "Никель" и т.д. В столбце «Количество» содержатся целые положительные числа. Значения в столбце «Вес» суть вещественные числа, равные весу детали в килограммах. Каждый столбец имеет имя, которое обычно записывается в верхней части таблицы. Оно должно быть уникальным в таблице, однако различные таблицы могут иметь столбцы с одинаковыми именами. Любая таблица должна иметь по крайней мере один столбец; столбцы расположены в таблице в соответствии с порядком следования их имен при ее создании. В отличие от столбцов, строки не имеют имен; порядок их следования в таблице не определен, а количество логически не ограничено. Так как строки в таблице не упорядочены, невозможно выбрать строку по ее позиции - среди них не существует

"первой", "второй", "последней". Любая таблица имеет один или несколько столбцов, значения в которых однозначно идентифицируют каждую ее строку. Такой столбец (или комбинация столбцов) называется первичным ключом (primary key). В таблице «Деталь» первичный ключ - это столбец «Номер» детали. В нашем примере каждая деталь на складе имеет единственный номер, по которому из таблицы «Деталь» извлекается необходимая информация.

Следовательно, в этой таблице первичный ключ - это столбец «Номер» детали. В нем значения не могут дублироваться - в таблице «Деталь» не должно быть строк, имеющих одно и то же значение в столбце «Номер» детали.

Если таблица удовлетворяет этому требованию, она называется отношением (relation). Взаимосвязь таблиц является важнейшим элементом реляционной модели данных. Она поддерживается внешними ключами (foreign key).

Помимо таблиц, в базе данных могут храниться и другие объекты, такие как экранные формы, отчеты (reports), представления (views) и даже прикладные программы, работающие с базой данных.

Для пользователей информационной системы недостаточно, чтобы база данных просто отражала объекты реального мира. Важно, чтобы такое отражение было однозначным и непротиворечивым. В этом случае говорят, что база данных удовлетворяет условию целостности (integrity).

Система управления базами данных Access (СУБД Access) входит, в стандартный набор прикладных программ пакета Microsoft Office.

Система управления базой данных является универсальным программным инструментом создания и обслуживания баз данных и приложений пользователя в самых разных предметных областях. В СУБД поддерживаются различные модели данных.

Модель данных – это метод логической организации данных, используемый СУБД. Наиболее известными являются иерархическая, сетевая и реляционная модели.

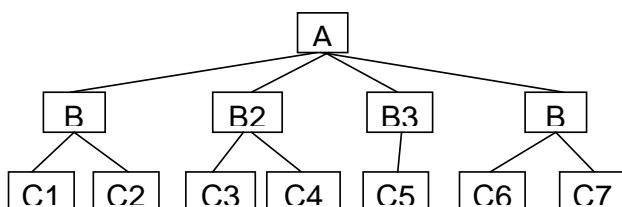


Рис.1. Иерархическая модель.

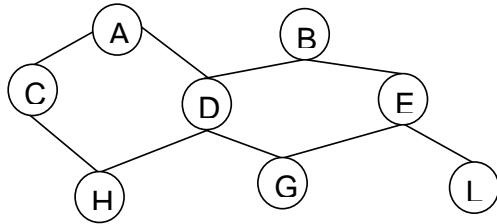


Рис.2. Сетевая модель.

Поле 1	Поле 2	Поле 3
A1	B1	C1
A2	B2	C2
A3	B3	C3

Рис. 3. Реляционная модель.

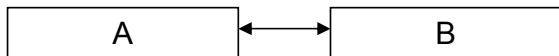
База данных является организованной на машинном носителе совокупностью взаимосвязанных данных и содержит сведения о различных сущностях одной предметной области – реальных объектах, процессах, событиях или явлениях.

Реляционная база данных представляет собой множество взаимосвязанных двумерных таблиц – реляционных таблиц, называемых также отношениями, в каждой из которых содержатся сведения об одной сущности автоматизируемой предметной области.

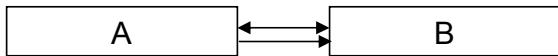
Структура реляционной таблицы определяется составом **полей**. Каждое поле отражает определенную характеристику сущности. Для поля указывается тип, размер и ряд других свойств. Содержание таблицы заключено в ее строках, однотипных по структуре. Каждая строка таблицы содержит данные о конкретном экземпляре сущности и называется **записью**. Структура записи определяется составом входящих в нее полей. Для однозначного определения (идентификации) каждой записи таблица должна иметь **уникальный (первичный) ключ**. По значению ключа таблицы отыскивается единственная запись в таблице. Ключ может состоять из одного или нескольких полей таблицы.

В реляционной базе данных связь двух таблиц характеризуется отношениями записей типа **один к одному** (1:1) или **один ко многим** (1:M). Отношение 1:1 предполагает, что каждой записи одной таблицы соответствует одна запись в другой. Отношение 1:M предполагает, что каждой записи первой соответствует много записей во второй, но каждой записи второй таблицы соответствует только одна запись в первой.

Один к одному



Один ко многим



Многие ко многим

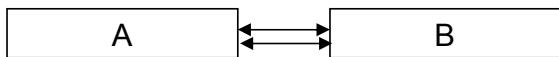


Рис. 4. Типы связей.

Этапы проектирования базы данных.

Этап проектирования реляционной базы данных заключается в определении ее логической структуры, т.е. состава реляционных таблиц и межтабличных связей.

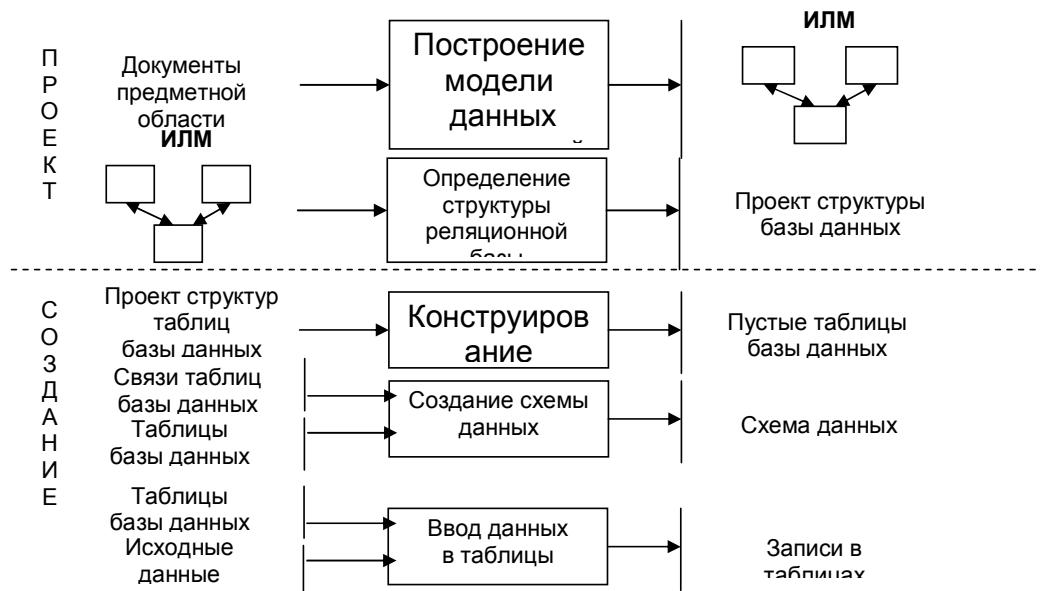


Рис. 5. Этапы проектирования и создания базы данных.

Информационно-логическая модель (ИЛМ) отображает данные предметной области в виде совокупности информационных объектов и связей между ними.

Информационный объект – это информационное описание некоторой сущности предметной области – реального объекта, процесса, явления или события.

Информационный объект образуется совокупностью логически взаимосвязанных **реквизитов**, представляющих качественные и количественные характеристики сущности.

Информационный объект имеет множество реализаций – **экземпляров** объекта.

Экземпляр объекта должен однозначно определяться среди множества экземпляров, т.е. идентифицироваться значением уникального ключа информационного объекта.

Простой ключ состоит из одного реквизита, а **составной ключ** – из нескольких реквизитов.

Этапы проектирования базы данных:

- определение цели создания базы данных;
- определение таблиц, которые должна содержать база данных;
- определение необходимых в таблице полей;
- задание индивидуального значения каждому полю;
- определение связей между таблицами;
- обновление структуры базы данных;
- добавление данных и создание других объектов базы данных;
- использование средств анализа.

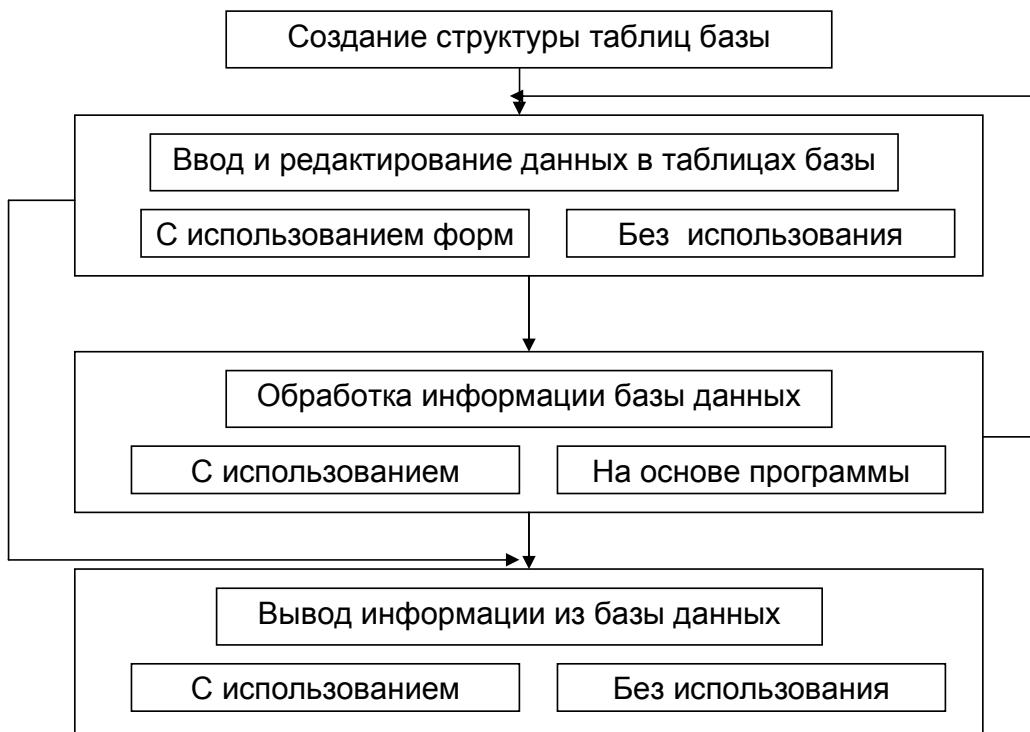


Рис. 6. Обобщенная технология работы..

Структурные элементы базы данных в Access.

Таблицы создаются пользователем для хранения данных об одной сущности – одном информационном объекте модели данных предметной области.

Запросы служат для выборки нужных данных из одной или нескольких связанных таблиц.

Схемы данных определяют, с помощью каких полей таблицы они связываются между собой.

Формы являются основным средством создания диалогового интерфейса приложения пользователя.

Отчеты предназначены для формирования выходных документов любых форматов, содержащих результаты решения задач пользователя, и вывода их на печать.

Страницы доступа к данным являются диалоговыми Web-страницами, которые поддерживают динамическую связь с базой данных и позволяют просматривать, редактировать и вводить данные в базу, работая в окне браузера.

Макросы позволяют автоматизировать некоторые действия в приложении пользователя.

Модули содержат процедуры на языке Visual Basic for Applications.

Создание БД. Работа с таблицами.

Чтобы начать работу в СУБД Access необходимо ее запустить. Это можно сделать так: **Пуск – Программы – Microsoft Access**. Появившееся окно состоит из:

- Страна заголовка.
- Страна меню.
- Панели инструментов.
- Область задач.
- Окно базы данных.
- Страна состояния.

Создание базы данных. После запуска Microsoft Access выводит область задач **Создание файла**, которая предоставляет возможность начать создание базы данных или открыть ранее созданную базу данных. Если область задач не выводится, можно начать создание базы данных с помощью команды **Файл – Создать**.

В области задач представлено четыре строки:

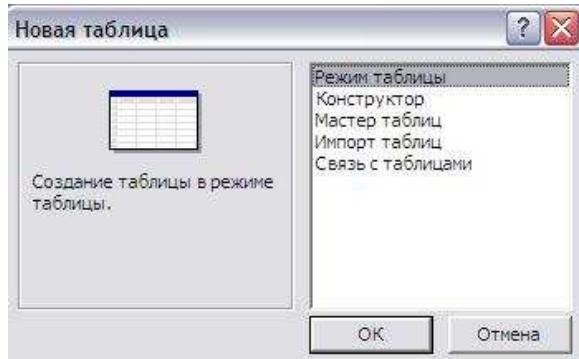
Новая база данных позволяет приступить к созданию новой оригинальной базы данных с пустого файла.

Пустая страница доступа к данным позволяет создавать в режиме конструктора Web-страницы специального типа на основе таблиц и запросов.

Проект (существующие данные) и **Проект (новые данные)** позволяют создавать проект-приложение пользователя, которое работает с базой данных, размещенной на SQL-сервере.

В области задач в группе **Создание из имеющегося файла** строка **Выбор файла** позволяет выбрать существующий файл в качестве основы для создания нового файла.

В области задач в группе **Создание с помощью шаблона** строка **Общие шаблоны** открывает окно с двумя вкладками. На вкладке **Общие** значками представлены все перечисленные средства создания баз данных и приложений. Вкладка **Базы данных** открывает возможность создать с помощью мастера на основе существующих шаблонов одну из типовых баз данных.

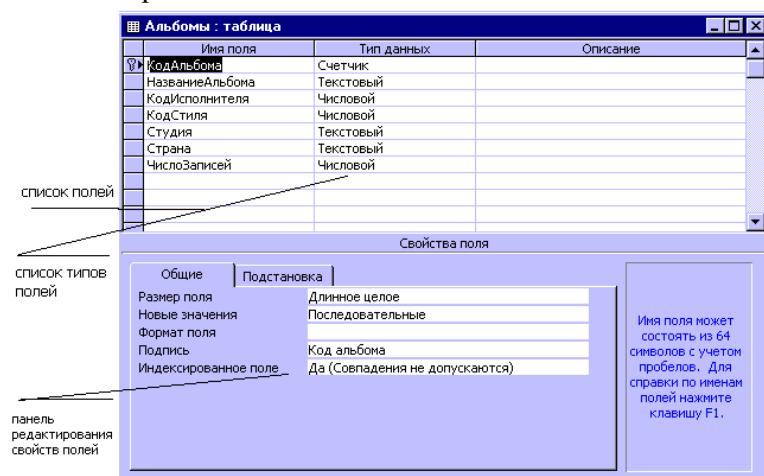


Microsoft Access хранит все объекты в одном файле. Поэтому необходимо в начале работы создать файл пустой базы данных. Для создания файла новой базы данных надо в области задач выбрать строку **Новая база данных**. Появится диалоговое окно **Файл новой базы данных**. В раскрывающемся списке **Папка** нужно выбрать каталог, в котором будет размещен файл, задать имя файла и нажать на кнопку **Создать**.

При работе с БД в первую очередь открывается **окно базы данных**. В этом окне в разделе **Объекты** вертикальным рядом представлены все объекты, которые могут быть созданы в БД: таблицы, запросы, формы, отчеты, страницы, макросы, модули. При нажатии кнопки в рабочем поле окна отображается список имен объектов данного типа. Объекты различных

типов могут объединяться в группы, которые представлены в разделе **Группы**.

Создание таблиц БД состоит из двух этапов. На первом этапе определяется ее структура: состав полей, их имена,



последовательность размещения полей в таблице, тип данных каждого поля, размер поля, ключи, индексы таблицы и другие свойства полей. На втором этапе производится создание записей таблицы и заполнение их данными.

Для создания новой таблицы в окне база данных выбирается объект **Таблица** и нажимается кнопка **Создать**. В открывшемся окне Новая таблица выбирается один из режимов создания таблицы.

Определение структуры новой таблицы в режиме конструктора.

При выборе режима конструктора появляется окно **Таблица1: таблица**, в которой определяется структура таблицы базы данных.

Для определения поля в окне **Таблица**, задаются **Имя поля**, **Тип данных**, а также свойства поля в разделе **Свойства поля**.

Имя поля. Каждое поле в таблице должно иметь уникальное имя. Оно является комбинацией из букв, цифр, пробелов и специальных символов, за исключением [! . `]. Имя не может начинаться с пробела. Максимальная длина имени – 64 символа.

Тип данных. Тип данных определяется **значениями**, которые предполагается вводить в поле, и операциями, которые будут выполняться с этими значениями. Раскрывающийся список возможных типов данных вызывается нажатием кнопки списка при выборе типа данных каждого поля.

1. **Текстовый** – тип данных по умолчанию. Текст или цифры, не участвующие в расчетах. Число символов не должно превышать 255.
2. **Числовой** тип служит для ввода числовых данных.
3. Тип для ввода даты или времени - **Дата/время**.
4. Для ввода логических данных, имеющих только два значения (Да или Нет; 0 или 1; Истина или Ложь и т.п.), служит специальный тип – **логический**. Длина такого поля всегда равна 1 байту.
5. Особый тип поля – **денежный**. В этом случае компьютер изображает числа вместе с денежными единицами, различает рубли и копейки, фунты и пенсы, доллары и центы.
6. В современных БД можно хранить не только числа и буквы, но и картинки, музыкальные клипы и видеозаписи. Поле для таких объектов называют **полем объекта OLE**.
7. Если нужно вставить в поле длинный текст, для этого служит поле типа **МЕМО**. В нем можно хранить до 65535 символов. Особенность поля МЕМО состоит в том, что реально эти данные хранятся не в поле, а в другом месте, а в поле хранится только указатель на то, где расположен текст.
8. Очень интересно поле **счетчик**. Оно имеет свойство автоматического наращивания. Это поле удобно для нумерации записей.
9. **Мастер подстановок.** Выбор этого типа данных запускает мастер подстановок. Мастер строит для поля список значений на основе полей из другой таблицы. Значения в такое поле будут вводиться из списка.

Общие свойства полей. Общие свойства задаются для каждого поля на вкладке Общие и зависят от выбранного типа данных. Для отображения

свойств поля необходимо установить курсор на строке соответствующего поля.

Размер поля задает максимальный размер сохраняемых в поле данных.

Для поля с типом данных Текстовый задается размер от 1 до 255 знаков (по умолчанию – 50 знаков).

Для поля с типом данных **Числовой** можно задать:

- **Байт** (для целых чисел от 0 до 255);
- **Целое** (для целых чисел от -32768 до +32767, занимает два байта);
- **Длинное целое** (для целых чисел от -2 147 483 648 до +2 147 483 647, занимает четыре байта);
- **Дробные с плавающей точкой 4 байта** (для чисел от $-3,4 \cdot 10^{38}$ до $+3,4 \cdot 10^{38}$ с точностью до семи знаков);
- **Дробные с плавающей точкой до 8 байт** (для чисел от $-1,797 \cdot 10^{308}$ до $+1,797 \cdot 10^{308}$ с точностью до 15 знаков);
- **Действительное.**

Формат поля является форматом отображения заданного типа данных и задает правила представления данных при выводе их на экран или печать. В Access определены встроенные стандартные форматы отображения для полей с типами данных Числовой, Дата/Время, Логический и Денежный. Для указания конкретного формата отображения необходимо выбрать в раскрывающемся списке одно из значений свойства Формат поля.

Число десятичных знаков задает для числового и денежного типов данных число знаков после запятой. Можно задать число от 0 до 15.

Подпись поля задает текст, который выводится в таблицах, формах, отчетах.

Значение по умолчанию определяет текст или выражение, которое автоматически вводится в поле при создании новой записи.

Условие на значение позволяет осуществлять контроль ввода, задает ограничения на вводимые значения, запрещает ввод при нарушении условий и выводит текст, заданный свойством Сообщение об ошибке.

Сообщение об ошибке задает текст сообщения, выводимый на экран при нарушении ограничений, заданных свойством Условие на значение.

Определение первичного ключа. Каждая таблица в реляционной БД должна иметь уникальный (первичный) ключ, однозначно определяющий каждую запись. Ключевое поле должно содержать уникальные значения и не может содержать значения нуль. Если для таблицы определен первичный ключ, то Access предотвращает дублирование ключа или ввод значений нуль в эти поля. Ключ может быть простым и составным, включающим несколько полей. Для определения ключа выделяются составляющие ключ поля и выполняется команда **Правка – Ключевое поле**.

После определения структуры таблицы ее необходимо сохранить. Для этого используется команда **Файл – Сохранить**. В окне Сохранение вводится имя таблицы. Длина имени таблицы не должна превышать 64 знаков.

После сохранения таблицы становится доступным режим, позволяющий перейти ко второму этапу создания таблицы – созданию записей. Переход в этот режим, называемый режимом таблицы, возможен только при сохранении структуры таблицы и осуществляется нажатием кнопки **Вид**.

Создание новой таблицы в режиме таблицы. В режиме таблицы пользователь может создать таблицу, не определяя предварительно ее структуру.

Создание новой таблицы в режиме таблицы осуществляется выбором строки **Создание таблицы путем ввода данных** в рабочем поле окна БД или строки **Режим таблицы** в окне **Новая таблица**.

После выбора этого режима сразу открывается пустая таблица, в которую можно ввести данные. При сохранении этой таблицы Access проанализирует данные и автоматически присвоит соответствующий тип данных каждому полю, т.е. автоматически создаст структуру таблицы. Таблица имеет 10 столбцов и 21 строку. Полям таблицы по умолчанию присваиваются имена Поле 1, Поле 2 и т.д.

Любое поле можно переименовать дважды щелкнув кнопкой мыши на области выделения столбца, содержащей его имя.

Если требуется создать таблицу, содержащую более 20 полей, то можно вставить столбцы, для этого выполняется команда **Вставка – Столбец**.

Для удаления столбца его необходимо выделить, щелкнув кнопкой мыши на его заголовке, и выполнить команду **Правка – Удалить столбец**.

Создание таблицы с помощью мастера таблиц. Мастер создания таблиц автоматически создает таблицу по одному из шаблонов. Начать работу мастера можно, выбрав строку **Создание таблицы с помощью мастера** в рабочем поле объекта **Таблицы**. Пользователю предлагается для выбора более 50 образцов таблиц, предназначенных для использования в различных целях. Каждая таблица шаблона содержит соответствующий набор полей, из которых пользователь может выбрать нужные для включения в создаваемую таблицу. Включаемые в таблицу поля могут быть переименованы. Мастер определит ключ таблицы. После создания таблицы мастером можно в любое время доработать структуру таблицы в режиме конструктора.

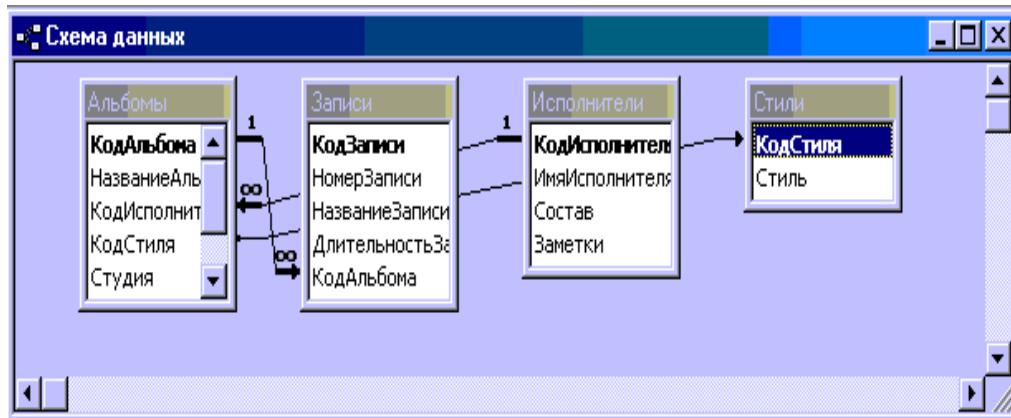
Связывание данных. Ввод данных в таблицы.

 Для создания связи между таблицами используется окно **Схема данных**, которое открывается щелчком на одноименной кнопке панели инструментов или командой **Сервис – Схема данных**. Если ранее никаких связей между таблицами базы не было, то при открытии окна **Схема данных** одновременно открывается окно **Добавление таблицы**, в котором можно выбрать нужные таблицы для включения в структуру межтабличных связей. Если связи между таблицами уже были заданы, то для введения в схему данных новой таблицы надо щелкнуть правой кнопкой мыши на схеме данных и в контекстном меню выбрать пункт **Добавить таблицу**.

Связь между полями устанавливают путем перетаскивания имени поля из одной таблицы в другую на соответствующее ему связное поле. После

перетаскивания открывается диалоговое окно **Связи**, в котором можно задать свойства образующейся связи. Включение флашка **Обеспечение условия целостности данных** позволяет защититься от случаев удаления записей из одной таблицы, при которой связанные с ними данные других таблиц останутся без связи.

Определение связей между таблицами.



1. Закрыть все открытые таблицы. Создавать или изменять связи между открытыми таблицами нельзя.

2. Переключиться в окно базы данных.

3. Нажать кнопку **Схема данных** на панели инструментов.

4. С помощью окна **Добавление таблицы** отобразить таблицы БД, для которых определяются связи.

5. Для связывания полей выбрать поле в одной таблице и перетащить его на соответствующее поле во второй таблице.

Для связывания сразу нескольких полей нужно переместить их при нажатой клавише **CTRL**.

В большинстве случаев связывают ключевое поле (представленное в списке полей полужирным шрифтом) одной таблицы с соответствующим ему полем (часто имеющим то же имя), которое называют полем внешнего ключа во второй таблице. Связанные поля не обязательно должны иметь одинаковые имена, но они должны иметь одинаковые типы данных (из этого правила существует два исключения) и иметь содержимое одного типа. Кроме того, связываемые поля числового типа должны иметь одинаковые значения свойства **Размер поля**.

6. В диалоговом окне **Схема данных**, отображенном на экране, нужно проверить имена полей, представленные в двух колонках. В случае необходимости внести изменения. Если необходимо, то нужно установить параметры связи.

7. Для создания связи надо нажать кнопку **Создать**.

8. Для каждой пары таблиц, которые необходимо связать, выполнить шаги 5-7.

При закрытии окна схемы данных на экран будет выведено сообщение, нужно ли сохранять макет. Не зависимо от того, будет он сохранен или нет, связи, созданные в базе данных, будут сохранены.

Если необходимо просмотреть все связи, определенные в базе данных, надо нажать кнопку **Показать все связи** на панели инструментов. Для просмотра связей только для определенной таблицы нужно выбрать таблицу, а затем на панели инструментов нажать кнопку **Показать прямые связи**.

Если в структуру таблицы необходимо внести изменения, необходимо щелкнуть таблицу правой кнопкой и выбрать команду Конструктор таблиц.

Допускается определение связей не только для таблиц, но и для запросов. Однако для запросов целостность данных все не обязательна.

Для связывания таблицы самой с собой или для связывания поля таблицы с другим полем той же таблицы следует дважды добавить таблицу. Такая ситуация возникает при определении поля с подстановкой значений из той же таблицы.

Ввод записей в режиме **таблицы**. Корректность вводимых данных (соответствие заданному типу поля, размеру и условию на значение, которые определены в свойствах полей в режиме конструктора) проверяется автоматически при их вводе. Отслеживается уникальность значений ключевых полей.

Отменить ввод значения в поле до перехода к другому полю можно, нажав клавишу <Esc> или выполнив команду **Правка – Отменить ввод**. Переход от одного поля к другому можно выполнить клавишей <Tab> или переводом курсора.

Добавление записи в таблицу начинается с заполнения пустой строки, размещенной в конце таблицы и помеченной звездочкой (*). Переход к этой записи можно выполнить также по команде **Правка – Перейти – Новая запись** или нажав кнопку **Новая запись** панели инструментов **Таблица в режиме таблицы**. Корректируемая запись помечается слева символом карандаша.

Изменение значений в полях записи осуществляется непосредственно в ячейках таблицы. Введенное значение проверяется Access при попытке перевода курсора в другое поле. Если значение не является допустимым, появляется предупреждающее сообщение. Для удаления записи в таблице ее нужно сделать текущей и нажать кнопку панели инструментов **Удалить запись** или выполнить соответствующую команду **Правка – Удалить запись**.

Одним из способов ввода подчиненных записей только при наличии связанной записи в главной таблице может быть использование в подчиненной таблице поля со списком, включающим значения поля главной таблицы.

Создадим в подчиненной таблице для поля** поле со списком, построенным на основе поля* главной таблицы. Для этого откроем подчиненную таблицу в режиме конструктора. Установим курсор в поле** на столбце **Тип данных**, откроем список и выберем строку **Мастер подстановок**. В открывшемся окне мастера **Создание подстановки** устанавливается переключатель **Объект «столбец подстановки»** будет использовать значения из таблицы или запроса. При этом будет построен список на основе значений из таблиц. В следующем окне мастера выбирается главная таблица, на значениях из которых будет строиться список поля. В следующих окнах мастера выбирается нужная ширина столбца списка и его подпись. На этом создание мастером поля со списком завершается. Параметры списка, подготовленные мастером, отображаются в окне свойств поля** на вкладке **Подстановка**. Теперь при вводе данных в подчиненную таблицу можно воспользоваться списком, в котором отобразятся все значения поля*, содержащиеся в главной таблице. Ввод значений в поле** осуществляется путем выбора нужного значения из списка.

Наличие поля со списком не запрещает ввод в поле** значений, не указанных в списке. Однако если при определении параметров списка в окне конструктора в разделе **Свойства поля** на вкладке **Подстановка** выбрать для свойства **Ограничиться списком** значение **Да**, то ввод непредусмотренных в списке значений сделается невозможным.

Отображение записей подчиненных таблиц в главной таблице.

В таблице базы данных Access можно одновременно просматривать данные вложенных таблиц на нескольких уровнях. Такие вложенные таблицы называются подтаблицами. Вложенная таблица должна быть связанной с основной таблицей. В пределах таблицы можно вложить до восьми уровней подтаблиц. Однако каждая таблица или подтаблица может иметь только одну вложенную подтаблицу.

Access автоматически создает подтаблицу в таблице, имеющей отношение типа 1:1 или являющейся стороной один в отношении 1:M, если в схеме БД между ними определена связь и свойство таблицы **Имя подтаблицы** имеет значение **Авто**. Отношение определяется по совпадению полей первичного ключа и внешнего (ключ связи подчиненной таблицы) ключа в связанных таблицах. При просмотре таблицы отображается столбец со значками плюс в каждой записи. Достаточно щелкнуть на значке плюс в записи этой таблицы, чтобы отобразились связанные с этой текущей записью записи подчиненной таблицы. При этом плюс на значке преобразуется в минус. Щелчком на минусе подчиненные записи закрываются. Таким образом могут быть открыты подчиненные записи каждой записи главной таблицы. Если открыть все записи главной таблицы, то все записи подчиненной таблицы, разбитые на подмножества, связаны с конкретными записями главной таблицы. Открыть или закрыть все подчиненные записи можно, воспользовавшись командой **Формат – Подтаблица – Развернуть все** или **Свернуть все**.

Поиск и замена значений в полях. Для поиска значений в текущем или во всех полях таблицы необходимо в режиме таблицы нажать кнопку панели инструментов **Найти** или воспользоваться командой меню **Правка – Найти**.

В появившемся диалоговом окне **Поиск и замена** на вкладке **Поиск** в поле **Образец** вводится значение, которое требуется найти в полях таблицы. В раскрывающемся списке **Поиск** выбирается область поиска. Если выбрать в диалоговом окне вкладку **Замена**, оно дополнится раскрывающимся списком **Заменить на**, позволяющим ввести в найденные поля новое значение.

Сортировка записей. Сортировка записей по возрастанию или убыванию значений одного поля (поля сортировки) может быть выполнена как в режиме таблицы, так и в режиме формы. Для этого курсор устанавливается в поле сортировки и нажимается соответствующая кнопка на панели инструментов.

- 
- кнопка **Сортировка по возрастанию**;
- 
- кнопка **Сортировка по убыванию**.

Для сортировки записей можно также использовать команду **Записи – Сортировка**.

Использование фильтра. Фильтр – это набор условий, применяемый для отбора подмножества записей из таблицы, формы или запроса.

Простейшим способом задания условия отбора записей является выделение в таблице некоторого значения поля или его части. Выбирается команда **Записи – Фильтр – Фильтр по выделенному**. В результате фильтрации в таблице будут отображены только те записи, в которых есть выделенное значение. Каждое новое выделение значения накладывает дополнительные условия на отбор записей существующим фильтром. Таким образом, данный способ позволяет не только создать новый фильтр, но и дополнить фильтр, созданный любым другим способом, новыми условиями отбора.

Для отбора записей, поля которых не содержат выделенного значения, необходимо после выделения значения выбрать команду **Записи – Фильтр – Исключить выделенное**. Если необходимо работать со всеми записями таблицы, то действие фильтра можно отменить, воспользовавшись командой **Записи – Удалить фильтр**. При этом фильтр снимается с таблицы, но сохраняется, и в любой момент его можно применить повторно, используя команду **Записи – Применить фильтр**.

Запросы. Запросы являются одним из основных инструментов выборки записей, обновления таблиц и обработки данных в таблицах БД. Результат выполнения запроса – это новая, чаще всего временная, таблица, которая существует до закрытия запроса. Структура такой таблицы определяется выбранными из одной или нескольких таблиц полями. Записи формируются путем объединения записей, участвующих в запросе таблиц. Простейшие

запросы могут быть созданы с помощью мастера. Любой запрос можно создать в режиме конструктора. Конструктор предоставляет удобное для пользователя диалоговое графическое средство формирования запроса по образцу, с помощью которого легко может быть построен сложный запрос.

С помощью запроса можно выполнить следующие виды обработки данных:

- выбрать записи, удовлетворяющие условиям отбора;
 - включить в результирующую таблицу запроса заданные пользователем поля;
 - произвести вычисления в каждой из полученных записей;
 - сгруппировать записи с одинаковыми значениями в одном или нескольких полях в одну запись с одновременным выполнением над другими полями групповых функций;
 - произвести обновление полей в выбранном подмножестве записей;
- создать новую таблицу базы данных, используя данные из существующих таблиц;
- удалить выбранное подмножество записей из таблицы базы данных;
 - добавить выбранное подмножество записей в другую таблицу.

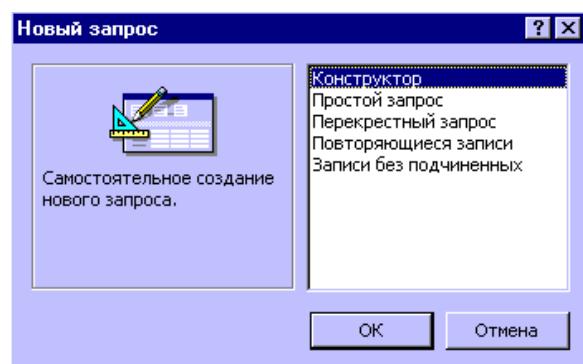
В Access может быть создано несколько видов запроса.

Запрос на выборку – выбирает данные из взаимосвязанных таблиц и других запросов. Результатом его является таблица, которая существует до закрытия запроса. На основе этого вида запроса могут строиться запросы других видов.

Запрос на создание таблицы – также выбирает данные из взаимосвязанных таблиц и других запросов, но, в отличие от запроса на выборку, сохраняет результат в новой постоянной таблице.

Запросы на обновление, добавление, удаление являются запросами действия, в результате выполнения которых изменяются данные в таблицах.

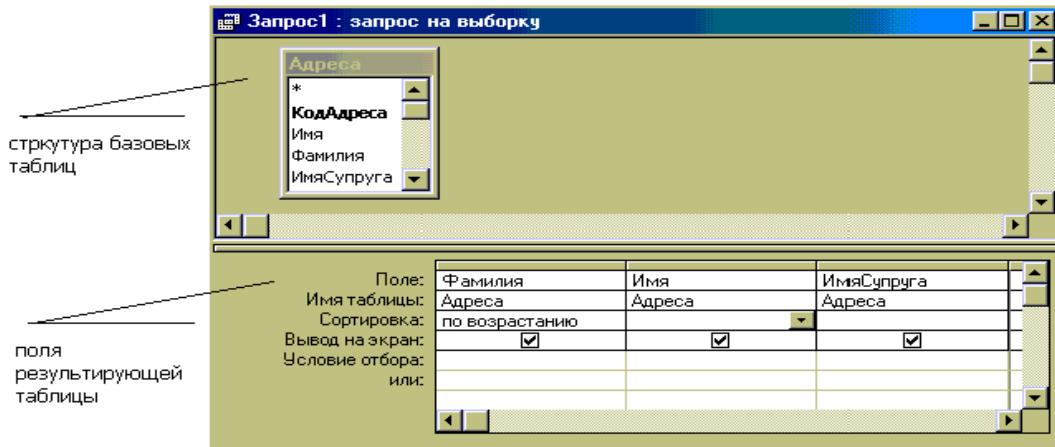
Создание запроса.



Для создания запроса необходимо в окне База данных выбрать вкладку Запросы и нажать кнопку Создать. В открывшемся окне Новый запрос выбрать строку Конструктор.

Можно перейти к созданию нового запроса в режиме конструктора, выбрав на вкладке Запросы строку Создание запроса в режиме конструктора.

Далее в открывшемся окне Добавление таблицы выбираются используемые в запросе таблицы и нажимается кнопка Добавить. В



результате появится окно конструктора запросов – Запрос N: запрос на выборку. Окно конструктора разделено на две панели. Верхняя панель содержит схему данных запроса, которая включает выбранные для данного запроса таблицы, представленные списками полей. Нижняя панель является бланком запроса по образцу, который необходимо заполнить:

- в строку Поле вклюаются имена используемых в запросе полей;
- в раскрывающемся списке Сортировка выбирается порядок сортировки записей результата;
- в строке Вывод на экран устанавливаются флагки для полей, которые должны быть включены в результирующую таблицу;
- в строке Условие отбора задаются условия отбора записей;
- в строке или (ог) задаются альтернативные условия отбора записей.

Для включения требуемых полей таблиц в соответствующие столбцы запроса можно воспользоваться следующими приемами:

- в первой строке бланка запроса Поле щелчком мыши вызвать появление кнопки списка и выбрать из списка нужное поле. Список содержит все поля таблиц, представленных в бланке запроса;
- переместить с помощью мыши требуемое поле из списка полей таблицы в схеме данных запроса в первую строку бланка запроса;
- дважды щелкнуть на имени поля таблицы в схеме данных запроса;
- для включения в запрос всех полей таблицы можно переместить с помощью мыши в соответствующую строку бланка запроса все поля из списка полей таблицы в схеме данных запроса или дважды щелкнуть на символе * в этом списке.

Условие отбора записей. Условием отбора является выражение, которое состоит из операторов сравнения и операндов. В качестве операндов выражения могут использоваться: литералы, константы, идентификаторы (ссылки).

Литералы – конкретные значения, воспринимаемые Access так, как они записаны в качестве литералов могут быть использованы числа, текстовые строки, даты.

Константы – неизменяющиеся значения, которые определены в Access, например, True, False, Null, Да, Нет.

Идентификатор – ссылка на значение поля, элемента управления или свойства. Идентификаторами могут быть имена полей таблиц, форм, отчетов и т.д., которые должны заключаться в квадратные скобки.

Операторами сравнения и логическими операторами, использование которых допускается в выражении условия отбора, являются:

=,<,>,<>,<=,>=, Between, In, Like, And, Or, Not.

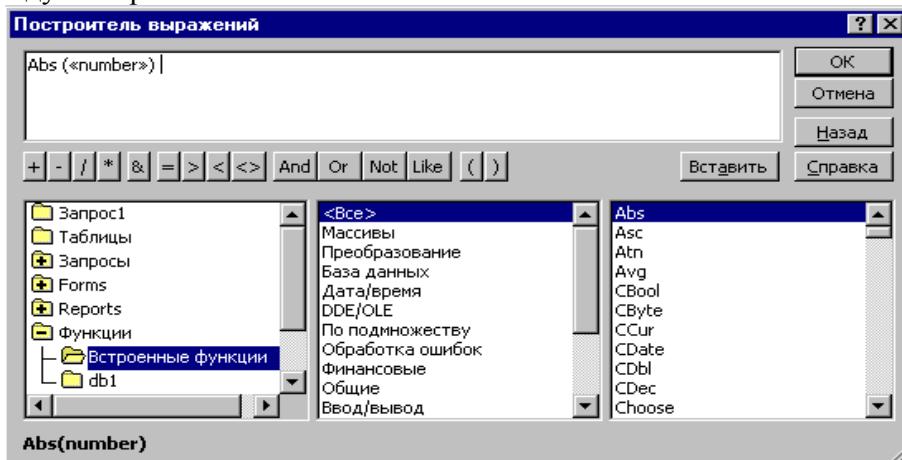
Эти операторы определяют операцию над одним или несколькими операндами. Если выражение в условии отбора не содержит оператора, то по умолчанию используется оператор =. Текстовые значения, если они содержат пробелы или знаки препинания, вводятся в двойных кавычках. В противном случае кавычки можно не вводить, они будут добавлены автоматически.

Оператор Between позволяет задать интервал для числового значения и даты. Например: Between 10 And 100.

Оператор In позволяет выполнить проверку на равенство любому значению из списка, который задается в круглых скобках. Например: In ("Математики";"Информатики";"Истории")

Оператор Like позволяет использовать образцы, использующие символы шаблона, при поиске в текстовых полях. Например: Like "Иванов*"

Сформировать условие отбора можно с помощью построителя выражений. Перейти в окно Построитель выражений можно, нажав кнопку Построить на панели инструментов конструктора запросов или выбрав команду Построить в контекстно-зависимом меню.



Вычисляемые поля. В запросе над полями могут производиться вычисления. Результат вычисления образует вычисляемое поле в таблице запроса. При каждом выполнении запроса производится вычисление с использованием текущих значений полей.

Помимо имен полей, в выражениях вычисляемых полей могут использоваться литералы и встроенные функции. Выражение вводится в бланк запроса в пустое поле строки Полье. Например:

Выражение1: [Цена]*[Количество]

Где Цена и Количество – имена полей.

Имя вычисляемого поля – Выражение 1 становится заголовком столбца в таблице с результатами выполнения запроса. Это имя можно изменить.

Параметры запроса. При решении практических задач пользователь должен иметь возможность вводить значение поля в диалоговом режиме в процессе выполнения запроса. Для этого необходимо определить параметр **запроса**. Имя параметра запроса может задаваться непосредственно в поле Условие отбора в квадратных скобках. При выполнении запроса это имя появится в диалоговом окне Введите значение **параметра**. Если в запрос вводится несколько параметров, то порядок их ввода через диалоговые окна определяется порядком расположения полей с параметрами в бланке запроса. Чтобы иметь возможность ввести несколько значений для одного поля при выполнении запроса, можно в условии отбора этого поля определить несколько параметров.

Корректировка данных средствами запросов.

Запрос на обновление.

Для обновления данных в полях базовых таблиц может быть использован запрос на обновление. Изменения вносятся в группу записей, отбираемых с помощью указанных пользователем условий отбора. Значения для изменений в полях определяются в бланке запроса в поле **Обновление**. Чтобы создать запрос на обновление, необходимо сначала создать запрос на выборку, который затем преобразуется в окне конструктора запросов в запрос на обновление при выборе команды **Обновление** меню, появляющегося при нажатии кнопки **Тип запроса** на панели инструментов конструктора запросов или команды **Запрос – Обновление**. После выполнения этой команды в бланке запроса появляется строка **Обновление**.

Для отбора обновляемых записей следует включить в бланк запроса поля, требующие обновления, а также поля, по которым задаются условия отбора записей. Процедура задания условия отбора аналогична созданию запроса на выборку.

После выполнения команды **Запрос – Запуск** открывается диалоговое окно с сообщением о числе обновляемых записей и вопросом о продолжении операции обновления.

Если необходимо просмотреть содержимое только обновленных полей, то после выполнения запроса следует переключиться в режим таблицы. Для этого выбирается команда **Вид – Режим таблицы** или используется кнопка **Вид** панели инструментов. Для просмотра старого содержимого обновляемых полей в режиме таблицы необходимо переключиться до выполнения запроса.

Запрос на добавление. С помощью запроса на добавление производится добавление записей из таблицы результата запроса в таблицу БД. Поэтому

поля добавляемых записей в запросе должны соответствовать полям существующих записей таблицы БД.

Структура записи таблицы может не полностью совпадать со структурой записи таблицы, в которую добавляются записи.

Первоначально запрос на добавление создается как запрос на выборку для одной или нескольких взаимосвязанных таблиц. Затем в окне конструктора запросов он превращается в запрос на добавление выбором команды **Запрос – Добавление**, при этом открывается диалоговое окно **Добавление**. В этом окне в поле с раскрывающимся списком вводится имя или выбирается имя таблицы, в которую необходимо добавить записи.

Если таблица-приемник добавляемых записей находится в открытой БД, необходимо отметить переключатель **в текущей базе данных**. Для таблицы, находящейся в другой БД, следует отметить переключатель **в другой базе данных** и ввести имя файла БД. После преобразования запроса в запрос на добавление в его бланке появляется строка **Добавление**.

При формировании добавляемых записей следует иметь в виду, что в бланк запроса должны быть включены поля, соответствующие полям таблицы, в которую будет производиться добавление. Если в таблице, куда добавляются записи, есть ключ, ключевые поля должны быть обязательно включены в бланк запроса. Кроме того, в бланк запроса могут быть включены поля, по которым задаются условия отбора.

Для указания в строке **Добавление** имен полей таблицы-получателя, в которые будут добавляться значения из соответствующих полей таблицы запроса, необходимо в каждом поле с раскрывающимся списком этой строки открыть список полей и выбрать нужное имя. Если выбранные поля имеют одни и те же имена в обеих таблицах, имена в строку **Добавление** вносятся автоматически.

Запрос на удаление. Запрос на удаление позволяет удалить записи из одной таблицы или из нескольких взаимосвязанных таблиц, для которых установлен флажок **каскадное удаление связанных записей**. В запросе указываются таблицы, из которых должны удаляться записи, и задаются условия отбора удаляемых записей.

Удалено: -

Первоначально запрос на удаление создается как запрос на выборку, в схему данных которого включаются взаимосвязанные таблицы, из которых требуется удалить записи и для полей которых задаются условия отбора. Затем запрос в режиме конструктора преобразуется в запрос на удаление командой **Запрос – Удаление**. После преобразования запроса в запрос на удаление в его бланке появляется строка **Удаление**. Затем формируется бланк запроса. Для задания таблицы, записи которой следует удалить, необходимо переместить с помощью мыши символ * из списка полей соответствующей таблицы в бланк запроса. В строке **Удаление** в этом поле появляется значение **Из**. Для задания условий отбора удаляемых записей перемещаются с помощью мыши в бланк запроса те поля, для которых устанавливаются условия отбора. В строке **Удаление** в полях с этими

именами появляется значение **Условие**. Строку **Условие отбора** для этих полей следует заполнить необходимыми условиями.

Использование групповых операций в запросах. Групповые операции позволяют выделить группы записей с одинаковыми значениями в указанных полях и использовать для этих групп одну из статистических функций. В Access предусмотрены девять статистических функций:

Sum – сумма значений некоторого поля для группы;

Avg – среднее от всех значений поля в группе;

Max, Min – максимальное и минимальное значение поля в группе;

Count – число значений поля в группе без учета пустых значений;

StDev – среднеквадратичное отклонение от среднего значения поля в группе;

Var – дисперсия значений поля в группе;

First, Last – значение поля из первой или последней записи в группе.

Для создания запроса с использованием групповых операций формируется запрос на выборку. В бланк запроса включаются поля, по которым необходимо произвести группировку, и поля, по которым следует произвести статистические вычисления.

После этого выполняется команда **Вид – Групповые операции**. В бланке запроса появится строка **Групповая операция**, все поля которой имеют значение **Группировка**. Для групповых вычислений по некоторому полю следует в этом поле заменить значение **Группировка** на нужную статистическую функцию. Для выбора необходимой функции используется раскрывающийся список этого поля.

Формы.

Проектирование форм. Access предоставляет широкие возможности по конструированию графического диалогового интерфейса пользователя для работы с БД. Основой такого интерфейса являются формы. Через формы может осуществляться первоначальная загрузка данных во взаимосвязанные таблицы БД, просмотр данных, а также их корректировка.



Если для БД создана схема данных, состоящая из связанных однотипными отношениями таблиц, то для обеспечения корректного ввода взаимосвязанных данных в такую БД могут быть созданы формы. Поскольку

источником данных для этих форм будут первичные документы, то при создании макета таких форм целесообразно соблюдать условие его максимальной внешней приближенности к оригиналу.

Технология создания целостной базы, в которой между таблицами установлены связи и определены параметры поддержания целостности, предполагает упорядочение загрузки взаимосвязанных таблиц для обеспечения пользователя удобным интерфейсом.

Чтобы иметь возможность отображать, вводить или корректировать данные таблиц через форму, необходимо сначала спроектировать и сконструировать ее.

В процессе разработки технологии загрузки БД и проектирования форм следует выполнить приведенную ниже последовательность действий:

1. Определение документов-источников в немашинной сферы, содержащих необходимые данные для загрузки таблиц БД.
2. Определение таблиц-объектов загрузки на отдельных этапах ввода данных и соответствующего документа-источника.
3. Определение содержания и последовательности этапов загрузки.
4. Определение подсхемы данных для каждого этапа загрузки (фрагмента схемы данных), обеспечивающей построение экранной формы для ввода данных из документа.
5. Определение общей структуры формы, т.е. ее макета в соответствии со структурой входного документа и полученной подсхемой данных. При этом:

- для составления формы выбирается таблица, которая будет источником записей основной части этой формы, и задается название формы;
- выбираются таблицы, которые будут источником записей подчиненных форм, включаемых в составную форму, и определяется название каждой подчиненной формы;
- распределяется пространство формы для размещения основной части формы и включаемых подчиненных форм.

6. Определение состава и размещение реквизитов (полей и надписей) для каждого из частей составной формы. При этом:

- ключевые поля таблицы-источника основной части необходимо размещать в основной части формы;
- в подчиненной форме следует предусмотреть только те ключевые поля таблицы-источника подчиненной формы, которых нет в таблице-источнике основной части.

Конструирование форм. Для конструирования форм в Access используется конструктор форм. Форма в режиме конструктора всегда имеет раздел **Область данных**. Разделы **Заголовок формы** и **Примечание формы** могут быть отображены по команде **Вид – Заголовок/примечание формы**. В процессе конструирования разделы формы наполняются различными графическими объектами и элементами управления.

При конструировании форм используется **Панель элементов**, которая вызывается из меню **Вид**. Она позволяет создавать элементы управления в форме и осуществлять действия при конструировании.



Выбор объекта – выделение по умолчанию элемента, раздела или формы.

Мастер – включение или отключение мастера создания элементов управления;

Надпись – создание текстов, постоянных заголовков, примечаний, инструкций, не связанных с другими элементами управления.

Поле – установление связи с полем таблицы или запроса, а также создание вычисляемых полей.

Группа переключателей – размещение набора флажков, переключателей или выключателей.

Переключатель – выбор альтернативных значений.

Флажок – выбор набора из возможных значений.

Поле со списком – объединение поля и раскрывающегося списка значений, которые могут вводиться как непосредственно в поле, так и выбором из списка.

Список – создание постоянно отображаемого списка значений, которые являются единственным источником ввода данных в поле.

Кнопка – создание командной кнопки, с помощью которой может быть выполнен переход по записям, открыта форма, напечатан отчет и другие функции Access.

Рисунок – отображение не редактируемого рисунка, не являющегося объектом OLE.

Свободная рамка объекта – отображение свободного объекта OLE, который остается неизменным при переходе по записям.

Присоединенная рамка объекта – отображение объектов OLE, сохраненных в поле базового источника записей формы.

Разрыв страницы – создание нового экрана в форме, новой страницы в печатной форме (отчете).

Вкладка – создание вкладок в форме, на каждой из которых могут размещаться свои элементы управления.

Подчиненная форма/отчет – вывод данных из таблиц, связанных с таблицей-источником основной части формы.

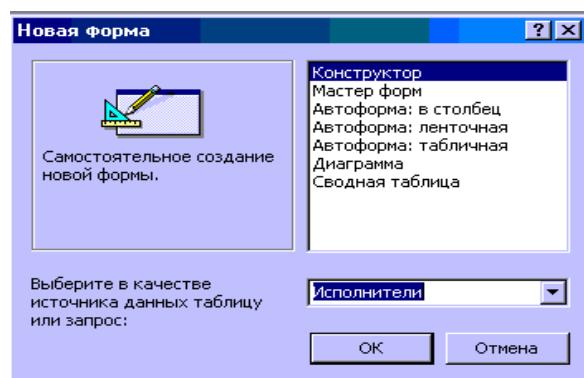
Линия – разграничение разделов в форме (отчете).

Прямоугольник – создание рамки при оформлении.

Другие элементы – открытие обширного списка дополнительных элементов, при выборе из которых в форме будет создан соответствующий элемент.

Режим конструктора форм. Создание формы в режиме конструктора полностью возлагается на пользователя. Чтобы начать создание формы в режиме конструктора, необходимо в окне БД, в области объекты выбрать

строку **Формы**. Для выбора режима конструктора при создании новой формы можно предварительно нажать кнопку **Создать**. После нажатия этой кнопки открывающееся диалоговое окно **Новая форма** предоставляет возможность выбрать режим создания формы – **Конструктор**.



выбрать режим создания формы – **Мастер форм**, который может создавать форму для одной таблицы или для нескольких взаимосвязанных таблиц. Мастер позволяет пользователю задать поля, которые необходимо включить в форму, и выбрать стиль ее оформления

Простейшую форму на основе только одной таблицы можно создать выбором одного из режимов **Автоформа: в столбец**, **Автоформа: ленточная**, **Автоформа: табличная**. Для построения автоформы необходимо предварительно в окне **Новая форма** выбрать таблицу-источник.

Форма, созданная мастером, так же как и форма, созданная любой командой Автоформа, может быть отредактирована в соответствии с требованиями пользователя. Редактирование выполняется в режиме конструктора форм.

Редактирование формы. Для уточнения местоположения, размера, шрифта и других параметров отображения элементов формы, а также для изменения текста надписей необходимо перейти в режим конструктора форм. При открытой форме переход в режим конструктора осуществляется нажатием кнопки **Вид** на панели инструментов **Режим формы**. В раскрывающемся списке кнопки **Вид** можно выбрать необходимый режим представления формы.

После выбора режима конструктора в окне Access появляется панель **Конструктор форм** и **Панель элементов**. Панель форматирования **Формат (форма/отчет)** может быть вызвана при активном окне формы по команде **Вид – Панели инструментов – Формат (форма/отчет)**.

Некоторые кнопки панели инструментов **Конструктор форм**.



Вид – выбор режима отображения формы из списка, включающего варианты: Конструктор, Режим формы, Режим таблицы, Сводная таблица, Сводная диаграмма.

Добавить гиперссылку – создание элемента управления для перехода к объектам БД, к документам.

Список полей – вызов списка доступных полей для выделенной области формы.

Панель элементов – открытие панели элементов.

Автоформат – настройка и применение встроенных форматов к форме, разделу или элементам управления.

Программа – открытие модуля формы в редакторе VBA.

Свойства – открытие окна свойств для выделенной формы, раздела, элемента управления.

Построить – вызов диалогового окна, позволяющего выбрать и запустить построитель выражений, редактор макросов, программ VBA для выделенной формы или элемента управления.

Окно базы данных – активизация окна БД.

Новый объект – открытие списка объектов с возможностью создания нового объекта БД.

Кнопки панели инструментов Формат (форма/отчет).



Объект – позволяет выбрать и выделить из списка доступные разделы и элементы управления формы.

Шрифт – осуществляет выбор шрифта для выделенного элемента управления.

Размер – позволяет выбрать размер шрифта для выделенного элемента управления.

Начертание – устанавливает для выделенного элемента управления начертание: полужирное, наклонное, подчеркнутое.

Выравнивание – выравнивает содержимое выделенного элемента управления по левому краю, по центру, по правому краю.

Цвет заливки/фона – позволяет выбрать цвет фона для выделенного элемента управления или раздела.

Цвет текста – позволяет выбрать цвет текста.

Цвет линии/границы – позволяет выбрать цвет границы для выделенного элемента управления.

Толщина линии/границы – позволяет выбрать толщину линии и границы.

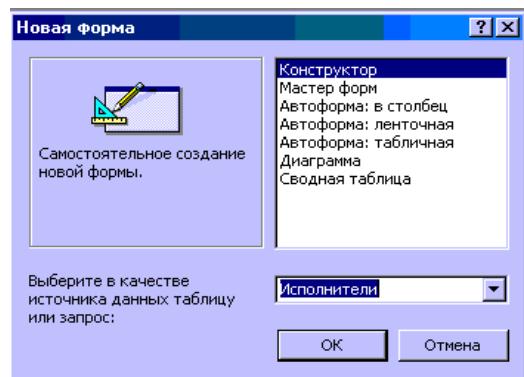
Оформление – позволяет выбрать вариант оформления выделенного раздела.

Редактирование формы и ее элементов может быть произведено не только графическими средствами, но и путем изменения свойств. Для этого открывается диалоговое окно свойств выделенного элемента по команде **Вид – Свойства** или с помощью контекстного меню.

После завершения редактирования формы она может быть сохранена. Если редактируемая форма ранее не сохранялась, то выполняется команда **Файл – Сохранить**.

Составная многотабличная форма создается для работ с несколькими взаимосвязанными таблицами. Многотабличная форма может состоять из основной части и одной или нескольких подчиненных включаемых форм. Многотабличная форма может быть создана в режиме конструктора или с помощью мастера форм. При использовании мастера форм подчиненная форма строится только для таблицы, которая является подчиненной по отношению к таблице – источнику основной части. Подчиненная форма отображает данные из всех записей подчиненной таблицы, которые связаны с записью главной таблицы, отображаемой в основной части формы. Мастер форм позволяет создать связанные формы, которые не включаются непосредственно в форму. При этом вместо подчиненной формы включается лишь кнопка, представляющая связанный форму.

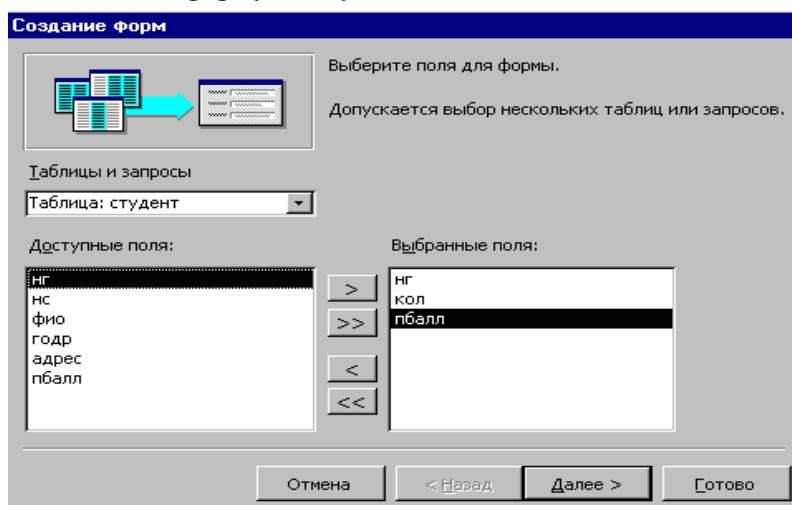
Составная многотабличная форма, построенная мастером форм, может не включать подчиненных и связанных форм. Такая форма создается, если необходимо отображать записи подчиненной таблицы, дополненные полями из одной или нескольких главных таблиц.



Создание многотабличной формы.
Для создания формы в окне БД в области объекты выбирается строка Формы и нажимается кнопка Создать.

В диалоговом окне Новая форма выбирается режим создания Мастер форм. В качестве источника данных основной части формы можно сразу выбрать из списка таблицу (или запрос). В первом шаге мастера последовательно выбираются

таблицы из списка Таблицы и запросы, начиная с главной таблицы, и включаемые в форму Доступные поля каждой из таблиц.

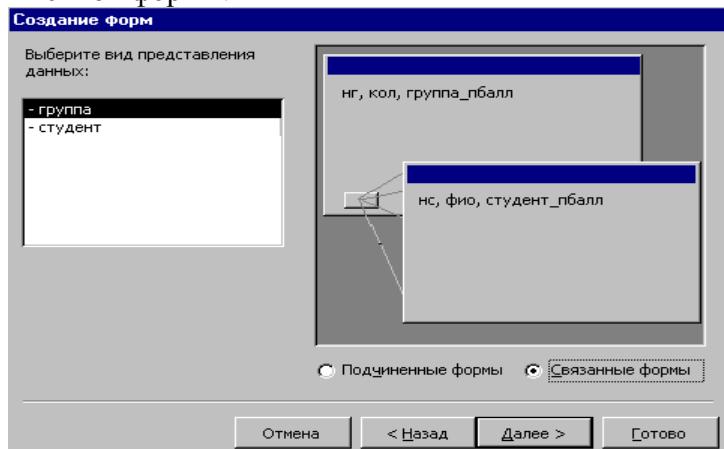


На следующем шаге в списке Выберите вид представления данных необходимо выделить имя таблицы, которая является источником основной части формы.

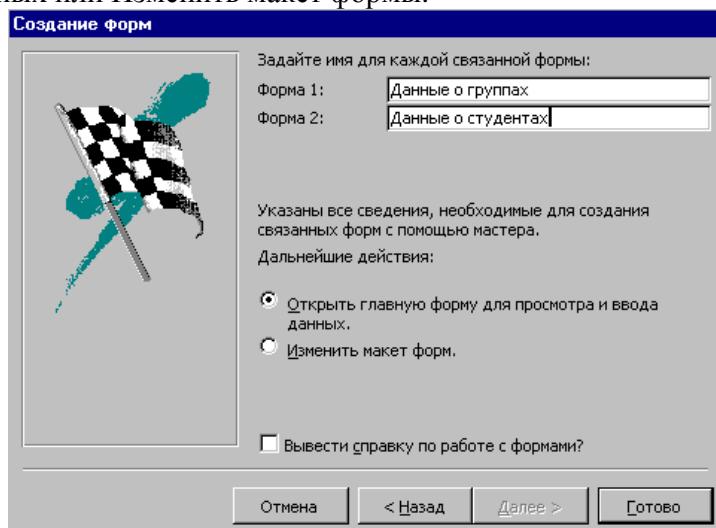
Если таблица ранее уже была выбрана в окне Новая форма, то она уже будет выделена. Также выбирается один из двух возможных типов подключения формы:

- для непосредственного включения подчиненной формы необходимо отметить переключатель Подчиненные формы;
- для включения кнопки, вызывающей связанную форму, нужно отметить переключатель Связанные формы.
- Если выбран первый тип, далее мастер предоставляет выбрать для подчиненной формы вид – ленточный или табличный стиль оформления, который определяет общий вид формы, отображение надписей и значений полей в форме.

В последнем шаге мастера можно изменить имена основной и подчиненной форм.

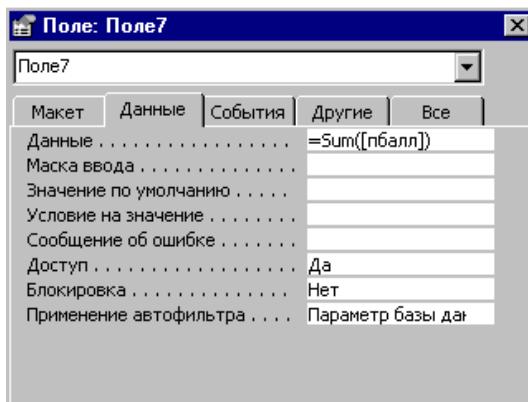


В том же шаге с помощью включения соответствующих переключателей можно выбрать дальнейшие действия: Открыть форму для просмотра и ввода данных или Изменить макет формы.



Если выбран первый переключатель, то автоматически выводится форма с данными связанных с ней таблиц. После нажатия кнопки Готово мастер завершает создание формы. Форма открывается в режиме формы. При выборе второго переключателя после завершения работы мастера форма будет отображена в режиме конструктора, позволяющем произвести нужную доработку.

Вычисления в форме. Чтобы произвести вычисления на основе данных в каждой записи, в форме необходимо создать вычисляемый элемент управления, в который записывается выражение. Для создания вычисляемого элемента управления открыть форму в режиме конструктора и, используя кнопку Поле на панели элементов, разместить этот свободный элемент управления в нужном месте области данных. Затем в элемент управления Свободный необходимо ввести выражение. Выражение должно начинаться со знака " $=$ ". В качестве операндов выражения чаще всего используются имена полей, константы, а в качестве операторов – знаки арифметических операций. Выражение может быть введено на вкладке Данные в поле Данные в окне свойств созданного элемента управления. Можно воспользоваться построителем выражений, используя появляющуюся кнопку Построить сразу за полем свойства Данные, который позволяет упростить создание выражения.



Вычисление итоговых значений. Вычисление итоговых значений для записей формы выполняется с помощью встроенных статистических функций, записываемых в качестве выражения в вычисляемых элементах управления. В однотабличной форме вычисляемый элемент управления, создаваемый для расчета итогового значения по всем записям, необходимо размещать в области примечания формы.

Отчеты. Средства Access по разработке отчетов предназначены для конструирования макета отчета, по которому может быть осуществлен вывод данных в виде выходного печатного документа.

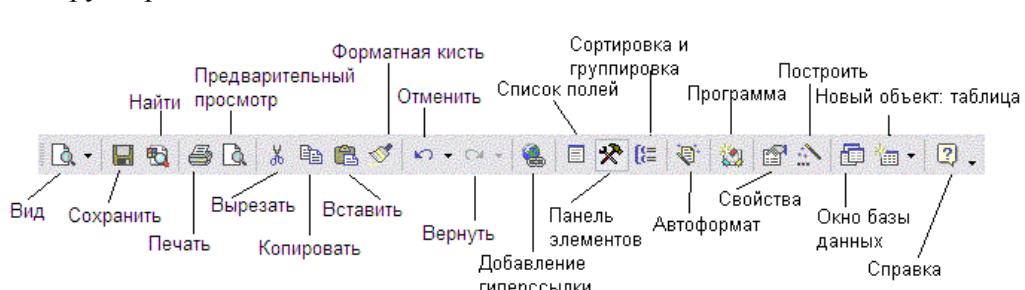
Отчет может создаваться с помощью мастера или в режиме конструктора отчетов. Режим конструктора отчетов во многом похож на режим конструктора форм.

Структура отчета состоит из пяти разделов: заголовка отчета, верхнего колонтитула, области данных, нижнего колонтитула, примечания отчета. В заголовок помещается текст. В верхний и нижний колонтитулы обычно помещают заголовка, номера страниц и даты. В области данных размещаются поля таблиц БД или запросов. В процессе конструирования разделы отчета заполняются элементами с помощью команд меню или кнопок панели инструментов конструктора отчета и панели элементов.



Поля с неповторяющимися значениями размещают в области данных. Поля с повторяющимися значениями, по которым можно проводить группировку записей, целесообразно размещать в добавленном разделе **Заголовок группы**. Элементами разделов отчета, кроме полей таблиц или запросов, на которых строится отчет, являются также тексты подписей, кнопки управления, внедряемые объекты, линии и т.п. Для каждого из элементов имеются соответствующие кнопки на панели элементов. Панель элементов, которая активно используется при конструировании отчетов, аналогична панели элементов, используемой при разработке форм.

В процессе конструирования отчета используется панель инструментов конструктора отчетов.



Создание однотабличного отчета в режиме конструктора. Выбрать в окне БД объект **Отчеты**. Нажать кнопку **Создать**. В открывшемся окне выбрать вариант **Конструктор**. Если в этом окне в качестве источника данных отчета выбрать таблицу (запрос), то одновременно с открытием нового отчета в режиме конструктора откроется дополнительное окно со списком полей **источника**. В открывшемся окне конструктора отображается макет создаваемого отчета. Если раздел **Заголовок отчета** отсутствует, то его можно включить командой **Вид – Заголовок/примечание отчета**.

Если в отчете необходима группировка данных по какому-нибудь полю, то используется кнопка **Сортировка и группировка**. Далее задаются необходимые параметры в открывшемся диалоговом окне

Сортировка и Группировка. В окне в столбце **Поле/выражение** необходимо записать имя поля для группировки или выбрать это поле из раскрывающегося списка полей источника. В области **Свойства группы** в строках **Заголовок группы** и **Примечание группы** следует выбрать значение **Да**. После определения группировки в окне конструктора отчета появляются дополнительные разделы **Заголовок группы** и **Примечание группы**.

Для задания сортировки в следующей строке диалогового окна достаточно записать (выбрать) имя соответствующего поля. В области **Свойства группы** этого поля в строках **Заголовок группы** и **Примечание группы** по умолчанию установлены значения **Нет**, что и устанавливает по этому полю только сортировку.

Размещение данных в разделе отчета. Процедура размещение полей из таблиц включает два этапа:

1. **Размещение поля Группировки.** Значение поля **Группировки** должно быть представлено один раз в области заголовка группы. Если список полей источника не открыт, нажимается кнопка **Список полей**. Затем поле группировки перетаскивается из списка полей в раздел **Заголовок группы**.

2. **Формирование табличной части отчета.** В соответствии с проектом макета необходимо разместить поля для табличной части так, чтобы их имена располагались в области данных, а подписи полей для формирования заголовка столбца – в области **Заголовок группы**. Тогда в области данных после формирования отчета будут размещены строки табличной части, а в области заголовка – название столбцов. При этом следует иметь в виду, что после перетаскивания полей из списка источника в область данных, они поместятся туда вместе с подписями. Для перемещения подписей в область **Заголовок группы** их можно вырезать и, сделав текущей область заголовка, вставить. Подписи также можно создать заново, воспользовавшись кнопкой панели элементов **Надпись**. Вычисляемое поле с итоговым значением по группе записей размещается в разделе **Примечание группы**. Для включения расчетного реквизита используется кнопка **Поле** на панели элементов, которая позволит сформировать элемент **Свободный**. Далее нужно открыть свойства создаваемого элемента и на вкладке **Данные** ввести в строку **Данные** необходимое выражение для выполнения групповой операции. При необходимости, следует выбрать в раскрывающемся списке **Число** десятичных знаков нужную точность расчета. На вкладке **Макет**, в раскрывающемся списке **Формат** поля выбирается требуемый формат.

Вычисляемое поле с итоговым значением по группе записей размещается в разделе **Примечание группы**. Для включения расчетного реквизита используется кнопка **Поле** на панели элементов, которая позволяет сформировать элемент **Свободный**. Далее нужно открыть свойства создаваемого элемента и на вкладке **Данные** ввести в строку **Данные** необходимое выражение для выполнения групповой операции. При необходимости следует выбрать в раскрывающемся списке **Число**

десятичных знаков нужную точность. На вкладке **Макет** в раскрывающемся списке **Формат** поля выбирается требуемый формат.

Для добавления в отчет текущей даты можно использовать встроенную функцию **=Now()** из категории **Дата/Время**. Сначала в области **Заголовок отчета** с помощью кнопки **Поле** создаётся свободный элемент. В окне его свойств на вкладке **Данные** записывается значение **=Now()**. На вкладке **Макет**, в раскрывающемся списке **Формат поля**, выбирается значение **Длинный формат даты**. Подпись этого поля можно удалить. Для добавления номера страницы в области нижнего колонтитула создается поле как свободный элемент. В окне его свойств на вкладке **Данные** в строку **Данные** записывается выражение **=[Page]**. Добавить в отчет поле текущей даты и времени можно, выполнив в режиме конструктора команду **Вставка/Дата и время**. Установка в диалоговом окне **Дата** и время флагков **Формат даты** и/или **Формат времени** позволяет вставить текущую дату и/или текущее время и выбрать нужный вариант форматов. В случае отсутствия раздела заголовка поле текущих даты и времени вносится в раздел данных. В результате в отчете будет создано поле, в окне свойств которого на вкладке **Данные** в строке **Данные** будет записано выражение **=Date()**.

Добавить в отчет поле нумерации страниц можно в режиме конструктора с помощью команды **Вставка – Номера страниц**. В диалоговом окне **Номера страниц** выбираются параметры, определяющие формат, расположение и выравнивание номеров страниц. Вывод номеров страниц можно задать, формируя соответствующее выражение в окне свойств поля на вкладке **Данные** в строке **Данные**. Такое выражение можно получить в построителе, открыв там папку **Общие выражения** и далее – категорию **Номер страницы**. При этом построитель предлагает выражение **=”Страница”&[Page]**. Если выбрать категорию **Страница N из M** построитель предлагает выражение **=”Страница”&[Page]&”из”&[Pages]**.

Создание отчета с помощью автоотчета.

Автоотчет является средством создания отчета, в котором выводятся все поля и записи из базовой таблицы или запроса.

1. В окне базы данных выбрать вкладку **Отчеты**.
2. Нажать кнопку **Создать**.
3. В диалоговом окне **Новый отчет** выбрать один из следующих мастеров:
 - автоотчет: в столбец — каждое поле образует отдельную строку с заголовком слева;
 - автоотчет: ленточный — поля каждой записи образуют отдельную строку. Заголовки печатаются сверху на каждой странице.
4. Выбрать таблицу или запрос, содержащие данные, выводящиеся в отчете.
5. Нажать кнопку **OK**.

Microsoft Access применяет последний автоформат, использованный для создания отчета. Если ранее не создавался отчет с помощью мастера или не

использовалась команда Автоформат из меню Формат, тогда будет применен стандартный автоформат.

Также можно создать отчет с полями, размещенными в один столбец, базовым источником данных которого являются таблица или запрос, выбранные в окне базы данных. Для этого надо выбрать команду Автоотчет в меню Вставка или нажать кнопку раскрытия списка рядом с кнопкой

Новый объект на панели инструментов и выбрать команду Автоотчет. Отчеты, созданные подобным образом, содержат только сами записи (в них нет верхнего и нижнего колонтитулов).



Разработка многотабличных отчетов. Необходимо создать два однотабличных отчета на основе главной и подчиненной таблиц.

Для вывода в отчет данных из подчиненной таблицы в виде табличной части необходимо создать отчет, который будет включен в основную часть отчета в качестве подчиненного.

Для включения подчиненного отчета в основной можно использовать графический способ. Для этого достаточно в окне БД выделить имя подчиненного отчета и перетащить его с помощью мыши в область данных отчета, построенного на основе главной таблицы.

Для включения подчиненного отчета можно также воспользоваться кнопкой панели элементов **Подчиненная форма/отчет**. При включенной кнопке панели элементов **Мастер элементов** в области данных основного отчета создается элемент **Свободный** с надписью **Внедренный**. В окне свойств этого элемента на вкладке **Данные** в раскрывающемся списке **Объект-источник** необходимо выбрать имя встраиваемого отчета.

Разработка с помощью мастера отчетов на основе подчиненной таблицы с включением данных из главных таблиц. В окне БД выбирается объект **Отчеты**. Нажать на кнопку **Создать**. В окне **Новый отчет** выбрать строку

Мастер отчетов и таблицу – основной источник для формирования отображаемых в отчете записей (подчиненную таблицу). После выбора мастера открывается окно **Создание отчетов**, в котором необходимо выбрать включаемые в отчет поля. В следующем окне мастера в поле **Выберите тип представления данных** по умолчанию будет выделено имя таблицы, которая была выбрана в качестве источника записей в окне **Новый отчет**. После этого открывается окно мастера, в котором задаются уровни группировки, которые позволяют сгруппировать записи для вывода в отчет, имеющие одинаковые значения в заданном поле группировки. В следующем сеансе диалогового окна мастера **Создание отчетов** задается порядок сортировки записей. Мастер позволяет производить сортировку по четырем полям в порядке возрастания или убывания значений поля. Порядок сортировки определяется надписью на кнопке, которая меняется при нажатии. Чтобы произвести подсчет итоговых значений в соответствии с заданной ранее группировкой, используется кнопка **Итоги**. Для числовых полей в открывшемся окне **Итоги** можно выбрать статистическую функцию для подсчета итоговых значений по сгруппированным строкам отчета. Далее в окне мастера выбирается один из видов макета отчета. В очередном сеансе выбирается один из возможных стилей оформления.

3.5. Мультимедиа технологии

Современный мультимедиа ПК в полном “вооружении” напоминает домашний стереофонический Hi-Fi комплекс, объединенный с дисплеем. Он укомплектован активными стереофоническими колонками, микрофоном и дисководом для оптических компакт-дисков *CD-ROM (CD – Compact Disc, компакт-диск; ROM – Read only Memory, память только для считывания)*. Кроме того, внутри компьютера укрыто новое для ПК устройство – аудиоадаптер, позволивший перейти к прослушиванию чистых стереофонических звуков через акустические колонки с встроенными усилителями.

Между тем исторически считалось, что для каждого информационного средства характерен свой особый вид представления информации. Для книг это текст, графика; для радиовещания — это средства записи и воспроизведения звука, речь, музыка; для телевидения и кино — движущиеся изображения, музыка, речь. С развитием вычислительной техники создалась возможность иметь комплексный объект, объединяющий в себе вышеперечисленные способы представления информации. Такие объекты стали называть *мультимедийными*.

Термин “МУЛЬТИМЕДИА” можно перевести на русский язык как “много сред”. Мультимедиа — это специальная технология, позволяющая с помощью программного обеспечения и технических средств объединить на компьютере обычную информацию (текст и графику) со звуком и движущимися изображениями. Мультимедийные возможности современных программных систем в отображении информации значительно отличаются от

привычных последовательных (линейных). Это отличие заключается не только в возросшем количестве средств представления — текст, графика, анимация, видео, звук, но и в иной форме организации данных, обусловленной возможностями гиперпространства.

Технологию мультимедиа составляют:

- аппаратные средства компьютера, обеспечивающие доступ к данным и воспроизведение мультимедийной информации;
- программные средства, обслуживающие доступ и воспроизведение;
- носители информации мультимедиа.

Мультимедиа — это интерактивные системы, обеспечивающие работу с неподвижными изображениями и движущимся видео, анимированной компьютерной графикой и текстом, гипертекстом, речью и высококачественным звуком.

Появление систем мультимедиа подготовлено как требованиями практики, так и развитием теории. Однако, резкий рывок, произошедший в этом направлении за последние годы, обеспечен прежде всего развитием технических и системных средств. Это и прогресс в развитии персональных компьютеров, связанный с резко возросшими: объемом памяти, быстродействием, графическими возможностями, характеристиками внешней памяти. Это и достижения в области видеотехники, накопителей информации на компакт дисках и их массовое применение.

Аудио технологии.

Мультимедиа определяют как многосредность. Одной из таких сред является аудиосреда. Мультимедиа технологии обеспечивают свой продукт высококачественным звуком, т.е. в диапазоне от 20 до 20000 Гц, и с частотой определения уровня (сколько раз в секунду делаются выборки реального звука) аналогового сигнала от 4 до 48 Кгц.

При поступлении на синтезатор некоторой управляющей информации по ней формируется соответствующий выходной сигнал. Современные аудиоадAPTERы синтезируют музыкальные звуки двумя способами: методом *частотной модуляции FM (Frequency Modulation)* и с помощью *волнового синтеза* (выбирая звуки из таблицы звуков, *Wave Table*).

Частотный синтез (FM) появился в 1974 году (PC-Speaker). В 1985 году появился AdLib, который, используя частотную модуляцию, был способен играть музыку. Новая звуковая карта Sound Blaster уже могла записывать и воспроизводить звук. Стандартный FM-синтез имеет средние звуковые характеристики, поэтому на картах устанавливаются сложные системы фильтров против возможных звуковых помех.

Второй способ обеспечивает более натуральное звучание.

Суть технологии WT-синтеза состоит в следующем. На самой звуковой карте устанавливается модуль ПЗУ с “зашитыми” в него образцами звучания настоящих музыкальных инструментов — сэмплами, а WT-процессор с помощью специальных алгоритмов даже по одному тону инструмента воспроизводит все его остальные звуки. Кроме того многие производители

оснащают свои звуковые карты модуляторами ОЗУ, так что есть возможность не только записывать произвольные сэмплы, но и подгружать новые инструменты.

Кстати, управляющие команды для синтеза звука могут поступать на звуковую карту не только от компьютера, но и от другого, например, *MIDI (Musical Instruments Digital Interface)* устройства. MIDI-сообщение содержит ссылки на ноты, а не запись музыки как таковой. В частности, когда звуковая карта получает подобное сообщение, оно расшифровывается (какие ноты каких инструментов должны звучать) и отрабатывается на синтезаторе. В свою очередь компьютер может через MIDI управлять различными "интеллектуальными" музыкальными инструментами с соответствующим интерфейсом.

Для электронных синтезаторов обычно указывается число одновременно звучащих инструментов и их общее число (от 20 до 32). Также важна и программная совместимость аудиоадаптера с типовыми звуковыми платформами (SoundBlaster, Roland, AdLib, Microsoft Sound System, Gravis Ultrasound и др.).

В качестве примера рассмотрим состав узлов в недавнем прошлом одного из мощных аудиоадаптеров – Sound Blaster AWE 32 Value. Он содержит: два микрофонных малошумящих усилителя с автоматической регулировкой усиления для сигналов, поступающих от микрофона; два линейных усилителя для сигналов, поступающих с линии, с проигрывателя звуковых дисков или музыкального синтезатора. Кроме того, сюда входят программно-управляемый электронный микшер, обеспечивающий смешение сигналов от различных источников и регулировку их уровня и стереобаланса, 20-голосый синтезатор музыкальных звуков частотной модуляции FM, программно - управляемый волновой (табличный) синтезатор музыкальных звуков и звуковых эффектов (16 каналов, 32 голоса, 128 инструментов), аналого-цифровой 16-разрядный преобразователь для превращения аналогового сигнала с выхода микшера в цифровой сигнал, систему сжатия цифровой информации с возможностью применения расширенного звукового процессора ASP. Наконец, аудиоадаптер имеет цифро-аналоговый преобразователь (ЦАП) для превращения цифровых сигналов, несущих информацию о звуке, в аналоговый сигнал, адаптивный электронный фильтр на выходе ЦАП, снижающий помехи от квантования сигнала, двухканальный усилитель мощности по 4 Вт на канал с ручным и программно-управляемым регулятором громкости и MIDI-разъем для подключения музыкальных инструментов.

Как видно из этого перечня, аудиоадаптер – достаточно сложное техническое устройство, построенное на основе использования последних достижений в аналоговой и цифровой аудиотехнике.

В современные звуковые карты входит *цифровой сигнальный процессор DSP (Digital Signal Processor)* или *расширенный сигнальный процессор ASP (Advanced Signal Processor)*. Они используют совершенные

алгоритмы для цифровой компрессии и декомпрессии звуковых сигналов, для расширения базы стереозвука, создания эха и обеспечения объемного (квадрофонического) звучания.

Видео технологии.

В компьютере все данные хранятся в цифровой форме, в то время как телевизионные, видео- и большинство аудиоаппаратуры имеет дело с аналоговым сигналом. Поэтому простейший и наиболее дешевый путь построения первых систем мультимедиа состоял в стыковке разнородной аппаратуры с компьютером, предоставлении компьютеру возможностей управления этими устройствами, совмещении выходных сигналов компьютера, видео- и аудиоустройств и обеспечении их нормальной совместной работы.

Благодаря специальному устройству (genlock) возможно совмещение изображения, сгенерированного компьютером (как то анимированная или неподвижная графика, текст, титры) с “живым” видео. Если же использовать TV-кодер, то компьютерное изображение можно преобразовать в форму TV-сигнала и далее записать на видеокассету и т.д. Так было в начале мультимедиа технологий (ММТ).

Недостатком ММТ этого периода являлась невозможность редактирования “живого”, аналогового видеоизображения. Для того, чтобы это стало возможным, его необходимо оцифровать и ввести в память компьютера. Для этого служат так называемые *платы захвата* (*capture board, frame grabbers*). Оцифровка аналоговых сигналов порождает огромные массивы данных.

Оцифрованный кадр может затем быть изменен, отредактирован обычным графическим редактором, могут быть убраны или добавлены детали, изменены цвета, масштабы, добавлены спецэффекты, типа мозаики, инверсии и т.д. Обработанные кадры могут быть записаны на диск в каком-либо графическом формате и затем использоваться в качестве реалистического неподвижного фона для компьютерной анимации. Возможна также покадровая обработка исходного изображения и вывод обратно на видеопленку.

Запись последовательности кадров в цифровом виде требует от компьютера больших объемов внешней памяти: для запоминания одной секунды полноцветного полноэкранного видео требуется 20–30 Мбайт, а оптический диск емкостью 600 Мбайт в этом случае вместит менее полминуты изображения. Кроме того есть проблема вывода на экран последовательности кадров в приемлемом темпе.

Обе эти проблемы решаются с помощью методов сжатия / развертки данных, которые позволяют сжимать информацию перед записью на внешнее устройство, а затем считывать и разворачивать в реальном режиме времени при выводе на экран. Далее результирующий продукт помещают на СД.

Алгоритмы сжатия применимы не только к видеоизображениям, но и к компьютерной графике, что дает возможность применять на обычных персональных компьютерах новый для них вид анимации, а именно

покадровую запись рисованных мультфильмов большой продолжительности. Эти мультфильмы могут храниться на диске, а при воспроизведении считываться, распаковываться и выдаватьсь на экран в реальном времени, обеспечивая те же необходимые для плавного изображения 25–30 кадров в секунду.

Технология MPEG.

Важную роль в совершенствовании мультимедиа сыграла так же разработка методов быстрого и эффективного сжатия/развертки данных, нашедших своё отражение в технологии MPEG.

MPEG - это аббревиатура от Moving Pictures Experts Group. Так называется комитет по стандартизации методов цифровой компрессии потоков видеоданных международной организации ISO/IEC (International Standards Organization/International Electrotechnical Commission). Изначально задача этого комитета состояла в разработке формата хранения и проигрывания аудио, видеоданных с компакт дисков CD-ROM. Так появился первый стандарт MPEG-1, ориентированный на низкоскоростные каналы передачи информации около 1Mbps (здесь и далее Kbps – Kilobit per second, Mbps – Megabit per second, MBps – MegaByte per second). По мере расширения задач передачи видео, повышения пропускной способности каналов и роста требований к визуальному качеству получаемых изображений появились MPEG-2, MPEG-4 и даже MPEG-7. К примеру, MPEG-4 предназначен для цифровой передачи видеоданных по телефонным линиям (Интернет, видеоконференции) в условиях жестко ограниченной пропускной способности, но использует самую продвинутую схему кодирования с разделением изображения на такие независимые объекты, как фон, текст, 2D/3D графика, “разговаривающие” человеческие лица, двигающиеся тела и т.д.

MPEG-2 и нелинейный монтаж. Что касается MPEG-2, то изначально он был нацелен на решение задачи передачи телевизионных изображений и допускает 4 уровня (Levels) разрешения кадра и 5 базовых профилей (Profiles) кодирования сигналов яркости и цветности.

Эффективное сжатие видеинформации зиждется на двух основных идеях: подавление несущественных для визуального восприятия мелких деталей пространственного распределения отдельных кадров и устранение временной избыточности в последовательности этих кадров. Соответственно говорят о пространственной и временной компрессии.

В первой из них используется экспериментально установленная малая чувствительность человеческого восприятия к искажениям мелких деталей изображения. Глаз быстрее замечает неоднородность равномерного фона, чем искривление тонкой границы или изменение яркости и цвета малого участка.

Временная MPEG-компрессия использует высокую избыточность информации в изображениях, разделенных малым интервалом. Действительно, между смежными изображениями обычно меняется только малая часть сцены, например, происходит плавное смещение небольшого

объекта на фоне фиксированного заднего плана. В этом случае полную информацию о сцене нужно сохранять только выборочно для опорных изображений. Для остальных достаточно передавать только разностную информацию: о положении объекта, направлении и величине его смещения, о новых элементах фона (открывающихся за объектом по мере его движения). Причем эти разности можно формировать не только по сравнению с предыдущими изображениями, но и с последующими (поскольку именно в них по мере движения объекта открывается часть фона, ранее скрытая за объектом).

Таким образом, в MPEG кодировке принципиально формируются три типа кадров: I (Intra), выполняющие роль опорных и сохраняющие полный объем информации о структуре изображения; P (Predictive), несущие информацию об изменениях в структуре изображения по сравнению с предыдущим кадром (типов I или P); B (Bi-directional), сохраняющие только самую существенную часть информации об отличиях от предыдущего и последующего изображений (только I или P). Последовательности I-, P-, B-кадров объединяются в фиксированные по длине и структуре группы кадров - GOP (Group of Pictures). Каждая GOP обязательно начинается с I и с определенной периодичностью содержит P кадры. Ее структуру описывают как M/N, где M – общее число кадров в группе, а N – интервал между P-кадрами. Так, типичная для Video-CD и DVD IPB группа 15/3 имеет следующий вид: IBBPBPBPBPBPBP. Здесь каждый B кадр восстанавливается по окружающим его P кадрам (в начале и конце группы – по I и P), а в свою очередь каждый P кадр – по предыдущему P (или I) кадру. В то же время I кадры самодостаточны и могут быть восстановлены независимо от других, но являются опорными для всех P и тем более B кадров группы. Соответственно у I и P наименьшая степень компрессии, у B – наибольшая. Установлено, что по размеру типичный P кадр составляет 1/3 от I, а B – 1/8 часть. В результате MPEG последовательность IPPP (GOP 4/1) обеспечивает двукратное уменьшение требуемого потока данных (при том же качестве) по сравнению с последовательностью только из I кадров, а использование GOP 15/3 позволяет достичь 4-кратного сжатия.

Компрессия звука. До сих пор речь шла только о компрессии изображений. Но полноценное видео подразумевает и звуковую составляющую. Подход к сжатию информации тот же – отбрасывание части, не очень существенной для восприятия человеческим ухом. MPEG стандарт разрешает 3 уровня (Layer) компрессии аудио. Layer 1 использует наиболее простой алгоритм с минимальной компрессией, что предполагает 192 Kbps на канал. Алгоритм Layer 2 более сложный, зато и степень компрессии больше, всего 128 Kbps на канал. Мощный алгоритм сжатия цифрового звука CD-качества (в 11 раз без различаемых человеческим ухом потерь) Layer 3 обеспечивает максимально возможное качество звука при жестких ограничениях потока – не более 64 Kbps на канал. В основном он предназначен для Интернет. Его значение столь велико, что он получил особое сокращенное наименование MP3, что означает MPEG Layer 3.

Появилось множество Интернет-сайтов, содержащих сотни тысяч MP3-файлов с популярной музыкой. С помощью специальных программ проигрывания (Real Audio) MP3-музыку можно в реальном времени слушать через Интернет, ее можно копировать. Уже появились портативные MP3-плееры (например, Diamond Rio).

MPEG-2 в задачах нелинейного монтажа. Термин нелинейный монтаж не соответствует сути процесса, а лишь отражает одну из его характеристик. На самом деле речь идет о монтаже видеофильмов, осуществляемом в цифровом виде на компьютерах. При этом исходные видеофрагменты подвергаются обязательной оцифровке и записи на винчестер в виде соответствующих файлов. В отличие от накопителей на магнитных лентах, доступ к любому из этих файлов-фрагментов не требует утомительной перемотки (а этот процесс линейный), т.е. все кадры видео доступны в произвольном порядке. Это важное свойство и обусловило название цифрового монтажа как нелинейного, хотя возможности цифровой обработки намного шире и богаче.

Известно множество алгоритмов, осуществляющих компрессию без потери информации, но даже самые эффективные из них на типичных изображениях не обеспечивают сжатия более чем в 2 раза.

До недавнего времени в мире систем нелинейного видеомонтажа безраздельно царил M-JPEG. Различные решения отличались степенью компрессии, что соответствовало различным уровням качества результирующего видео. Весьма условно, здесь можно выделить 4 уровня: Стандартное Видео (VHS, C-VHS, Video8), Супер-Видео (SVHS, C-SVHS, Hi8), Цифровое Видео (Betacam SP, DV/DVCAM/DVCPRO, mini-DV, Digital8) и Студийное Видео (Digital-S, DVCPRO50). Для простоты в дальнейшем будем обозначать их как Video, S-Video, DV и Studio-TV, которые требует для хранения и обработки видеофайлов значительных объемов дискового пространства. Например, 1 минута MJPEG-видео требует 120 МБ для качества Video и около 500 МБ для Studio-TV. Но ведь хочется работать с роликами продолжительностью в десятки минут! И вот здесь на арену выступает MPEG2. Уже просто переход к I-кадрам позволяет сэкономить 15% объема, а если использовать P-кадры, то можно выиграть еще вдвое (для групп IPPP), а это уже существенно.

3.6. Сетевые технологии

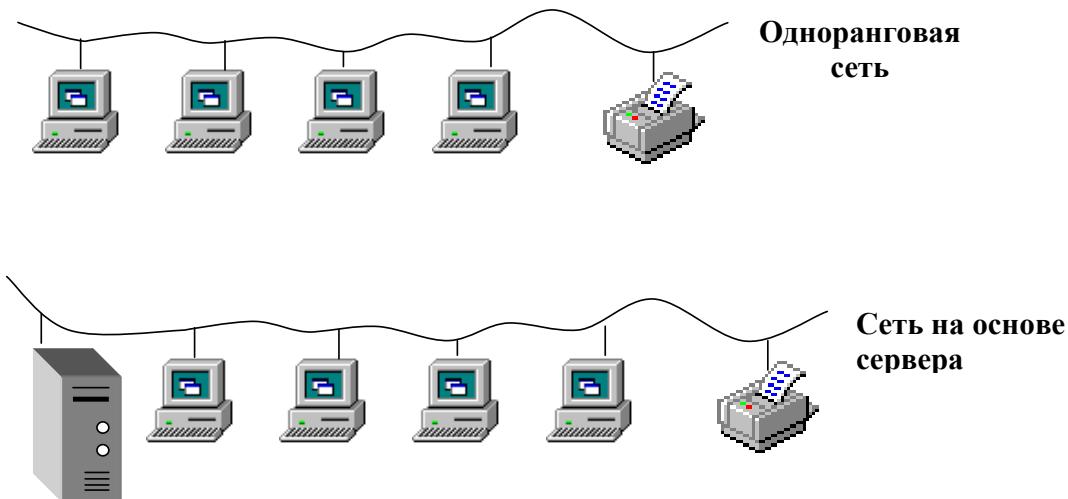
При наличии всего двух компьютеров может возникнуть потребность совместного использования данных, обмена информацией между ними. Такую потребность можно реализовать с помощью так называемых промежуточных носителей информации. Но лучше обеспечить кабельную (или беспроводную) связь между компьютерами, чтобы они обменивались информацией без участия человека. Если подобным образом соединить два близко расположенных компьютера и написать программы для передачи информации, то можно получить простейшую компьютерную сеть. Принципы соединения множества компьютеров в сеть остаются те же, что и

для двух. Компьютеры должны быть соединены с помощью линий связи. Для соединения удаленных компьютеров дополнительно используются специальные электронные устройства (модемы, сетевые карты и др.), и, кроме того, устанавливаются программы, обеспечивающие совместную работу компьютеров в сети.

Итак, **компьютерная сеть** – это комплекс компьютеров, линий связи между ними и программ, обеспечивающих обмен и совместное использование информации.

Компьютерные сети разделяются на два типа:

- Одноранговые сети (peer-to-peer)
- Сети на основе сервера (server based)



В одноранговой сети все компьютеры равноправны, т.е. нет определенного компьютера, ответственного за администрирование всей сети. Все пользователи самостоятельно решают, какие данные на своем компьютере сделать общедоступными по сети.

Одноранговая сеть вполне подходит там, где:

- количество пользователей небольшое ;
- пользователи расположены компактно;
- вопросы защиты данных не критичны;
- в обозримом будущем не ожидается значительного расширения сети.

Сети на основе сервера. С увеличением числа пользователей одноранговая сеть, где компьютеры выступают в роли и клиентов, и серверов, может оказаться недостаточно производительной. В таких ситуациях используется выделенные серверы. На дисках серверов располагаются совместно используемые программы, базы данных и т.д.

Выделенным сервером называется свободный от функции клиента или рабочей станции компьютер, специально подготовленный для быстрой обработки запросов от сетевых клиентов и для управления защитой файлов и каталогов.

Все сети строятся на основе трёх базовых топологий:

- шина (bus)
- звезда (star)
- кольцо (ring)

Если компьютеры подключены вдоль одного кабеля, топология называется шиной. В том случае, когда компьютеры подключены к сегментам кабеля, исходящим из одной точки, или концентратора, топология называется звездой. Если кабель, к которому подключены компьютеры, замкнут в кольцо, такая топология носит название кольца.

В настоящее время в компьютерных сетях используются все известные линии связи от низкоскоростных телефонных линий до высокоскоростных цифровых спутниковых каналов. Самым популярным способом подключения к компьютерным сетям остаётся передача информации по телефонным линиям с помощью модемов. Модем – это устройство для преобразования применяемых в компьютерах цифровых сигналов в используемые в телефонных линиях аналоговые сигналы и наоборот.

Так как при наборе телефонного номера для установки связи двух абонентов на АТС происходит переключение или, другими словами, коммутация линий связи, то телефонные линии часто называют коммутируемыми. С помощью коммутируемых линий связь устанавливается только на некоторое время, необходимое для телефонного разговора двух людей, или передачи информации между компьютерами. В отличие от коммутируемых, так называемые выделенные линии связывают компьютеры постоянно. Выделенные линии могут быть созданы с помощью кабелей или радиосвязи и позволяют обмениваться информацией с огромными скоростями.

Различают локальные и распределенные компьютерные сети. Локальной называется сеть, объединяющая компьютеры, расположенные в одном здании или в соседних зданиях, т.е. поблизости друг от друга. Итак, локальная сеть (иногда используется термин "локальная вычислительная сеть", сокращенно ЛВС) – это группа из нескольких компьютеров, соединенных между собой посредством кабелей (телефонных линий или радиоканалов), используемых для передачи информации между компьютерами. Для соединения компьютеров в локальную сеть необходимо сетевое оборудование и соответствующее программное обеспечение. Локальные компьютерные сети позволяют обеспечить:

- коллективную обработку данных пользователями подключенных в сеть компьютеров и обмен данными между этими пользователями;
- совместное использование программ;
- совместное использование принтеров, модемов и других устройств.

Поэтому организации, имеющие более одного компьютера, объединяют их в ЛВС.

Для объединения компьютеров в локальную сеть требуется:

- обеспечить каждый подключаемый к сети компьютер сетевой платой, которая позволяет компьютеру получать информацию из локальной сети и передавать данные в сеть;
- соединить кабелями компьютеры, а также другие подключенные к сети устройства (принтерами, сканерами и т.д.).

В некоторых типах сетей кабели соединяют компьютеры непосредственно (как электролампочки на елочной гирлянде), в других соединение кабелей осуществляется через специальные устройства-концентраторы (или хабы), коммутаторы и др.

Для обеспечения функционирования компьютерной сети часто выделяется специальный компьютер - сервер, или несколько таких компьютеров. На дисках серверов располагаются совместно используемые программы, базы данных и т.д. Остальные компьютеры сети часто называются рабочими станциями.

Если же соединенные в сеть компьютеры значительно удалены друг от друга, например, находятся в разных городах или странах, то такие сети называют распределенными. Часто к распределенной сети подключаются не отдельные компьютеры, а локальные компьютерные сети. Распределенные сети мирового масштаба называют также глобальными сетями.

Интернет является самой известной глобальной компьютерной сетью. Интернет - это миллионы компьютеров, соединенных проводами, телефонными линиями, радио и спутниковой связью. Часть из них - это подсоединеные большие и официальные компьютерные сети, подобно корпоративной сети IBM и т.д., другие - это компьютеры индивидуальных пользователей, установленные в университетах, лицеях, школах и частных домах.

Глава 4. Базовые информационные технологии высокого уровня

Более высокий уровень в иерархии базовых технологий занимают: Интернет технологии; Интранет технологии; 3Д-графика, FLASH MX и др.

4.1. Интернет технологии

Глобальная компьютерная сеть Интернет, практически охватывающая весь мир, сегодня становится неотъемлемой частью нашей повседневной реальности. Если ранее Интернет использовался в качестве среды передачи файлов и сообщений электронной почты, то сегодня осуществляется сетевой поиск и доступ к распределенным информационным ресурсам. Стоит ли говорить о том, что как только появилась возможность объединять

несколько компьютеров в сеть, пользователи ухватились за эту возможность не только для того, чтобы использовать ресурсы удаленных компьютеров, но и чтобы расширить круг своего общения.

Сегодня необходимо понимание того, что освоение и использование Интернет – это не благое пожелание, а настоятельная необходимость, диктуемая логикой развития науки, техники, культуры, бизнеса.

Введение в Интернет.

Интернет является самой известной глобальной компьютерной сетью. Интернет - это миллионы компьютеров, соединенные проводами, телефонными линиями, радио и спутниковой связью. Часть из них – это подсоединеные большие и официальные компьютерные сети, подобно корпоративной сети IBM и т.д. Другие – это компьютеры индивидуальных пользователей, установленные в университетах, лицеях, школах и частных домах .

Как устроена сеть Internet? Если уподобить Интернет сети автомобильных дорог, то пользование Интернетом похоже на движение автомобилей в большем городе. Каждый автомобиль движется сам по себе, неполадки одного не влияют на движение другого, если некоторые магистрали на ремонте, можно воспользоваться другими дорогами. Подобная организация обеспечивает высокую живучесть сети Интернет: выход из строя даже нескольких десятков компьютеров не скажется заметно на работе всей системы в целом. Всегда существуют запасные варианты путей прохождения информации.

Можно рассматривать Internet как глобальную информационную супермагистраль. Центральная жила Internet – оптоволоконный кабель с очень высокой пропускной способностью, это такая полноводная река информации, к которой примыкают тысячи мелких речушек. Среди прочих особенностей магистральный способ соединения, принятый в Internet, имеет важное достоинство – сеть позволяет осуществлять связь по принципу “многие со многими”, что обеспечивает возможность организации подписных листов и телеконференций –заочных компьютерных клубов, систему всемирной рекламы и информации.

Сегодня все больше и больше людей работают с Internet прямо из дома, используя персональный компьютер, модем и услуги провайдера, обеспечивающего подключение к сети. Что такое персональный компьютер и модем вы знаете, а что такое провайдер? Провайдер – это специальная коммерческая организация (фирма), обеспечивающая своим клиентам доступ в Internet.

О том как осуществляется передача информации в Интернет. В общем случае Internet осуществляет обмен информацией между любыми двумя компьютерами, подключенными к сети. Компьютеры, постоянно подключенные к Internet, часто называют узлами Internet, или сайтами. Узлы, установленные у поставщиков услуг Internet (у провайдеров), обеспечивают доступ пользователей к Internet. Существуют также узлы, специализирующиеся на предоставлении информации.

В Интернете используются два основных понятия: адрес и протокол. Любой компьютер, подключенный к Интернет, имеет свой уникальный адрес. Даже при временном соединении по коммутируемому каналу, компьютеру выделяется уникальный адрес. Таким образом, в любой момент времени все компьютеры, подключенные к Интернету, имеют разные, но уникальные адреса, а понятие адрес в Интернете является важнейшей его компонентой. Дисциплину же «общения» компьютеров в Интернет определяет протокол. Так что же такое протокол? В общем случае протокол - это правила взаимодействия. Сетевой протокол предписывает правила работы компьютеров, которые подключены к сети. Стандартные протоколы заставляют разные компьютеры «говорить на одном языке». Таким образом осуществляется возможность подключения к Интернету разнотипных компьютеров, работающих под управлением различных операционных систем.

В Интернете имеется несколько уровней протоколов, которые взаимодействуют друг с другом. На нижнем уровне используются два основных протокола: IP (протокол Интернета) и TCP (протокол управления передачей). Так как эти два протокола тесно взаимосвязаны, то часто их объединяют, и говорят, что в Интернете базовым протоколом является TCP/IP. Все остальные многочисленные протоколы строятся на основе именно протоколов TCP/IP.

Межсетевой протокол (IP). С помощью линий связи обеспечивается доставка данных из одного пункта в другой. Интернет может доставлять данные во многие точки, разбросанные по всему земному шару. Как это происходит? Различные участки Интернет связываются с помощью системы компьютеров (называемых маршрутизаторами), соединяющих между собой сети. Это могут быть сети Интернет, телефонные линии и т.п.

Маршрутизаторы как почтовые подстанции принимают решения о том, куда направлять «пакеты», то есть маршрутизатор смотрит, куда адресованы данные, и решает, куда их посыпать.

Откуда Интернет знает, куда следует направить данные? Выше мы затрагивали этот вопрос. Работа в Интернет регламентируется определенными правилами, и эти правила называются протоколами. Межсетевой протокол (Internet Protokol, IP) отвечает за адресацию, т.е. гарантирует, что маршрутизатор знает, что делать с данными, когда они поступят.

По целому ряду технических причин (в основном это аппаратные ограничения) информация, посыпаемая по IP сетям, разбивается на порции, называемые пакетами. В одном пакете обычно посыпается от одного до 1500 символов информации.

Одно из достоинств Интернет состоит в том, что для работы на базовом уровне достаточно только межсетевого протокола. Поскольку данные помещаются в IP конверт, то сеть имеет всю информацию, необходимую для перемещения этого пакета из компьютера в пункт назначения. Однако здесь возникает сразу несколько проблем:

- в большинстве случаев объем пересылаемой информации превышает 1500 символов;
- может произойти ошибка;
- последовательность доставки пакетов может быть нарушена.

Поэтому следующий уровень сети даст нам возможность пересылать более крупные порции информации и позаботиться об устранении тех искажений которые вносит сама сеть.

Протокол управления передачей (TCP). Для решения упомянутых выше проблем используется «протокол управления передачей» (Transmission Control Protokol, TCP). Информацию, которую Вы хотите передать, TCP разбивает на порции. Каждая порция нумеруется, чтобы можно было проверить, вся ли информация получена, и расположить данные в правильном порядке. Для передачи этого порядкового номера по сети у протокола есть свой собственный «конверт», на котором «написана» необходимая информация. Порция Ваших данных помещается в конверт TCP, который, в свою очередь, помещается в конверт IP и передается в сеть.

На принимающей стороне программное обеспечение протокола TCP собирает конверты, извлекает из них данные и располагает их в правильном порядке. Если каких-нибудь конвертов нет, программа просит отправителя передать их еще раз. После размещения всей информации в правильном порядке эти данные передаются той прикладной программе, которая использует услуги TCP.

Подключение к Интернет. Подключение к Интернет становится все более простым делом, поскольку постоянно увеличивается количество компаний, которые предоставляют возможность отдельным пользователям и небольшим фирмам работать в Интернет по телефонным каналам.

Доступ через IP. Полное подключение к Интернет – это лучший вариант для домашнего компьютера. Чтобы установить этот тип соединения, вам необходимо договориться с поставщиком услуг Интернет, который будет перенаправлять сетевой протокол TCP/IP по телефонному каналу на ваш компьютер. В этом случае ваш ПК фактически становится частью Интернет (в отличие от простого присоединения к компьютеру, входящему в Интернет). IP-доступ, кроме того, позволяет вам полностью управлять выполняемыми программами.

Предположим, что Вы подключились к Интернет, что дальше? Полезно узнать: какие услуги предоставляет Интернет.

Услуги Internet. Все услуги, предоставляемые Интернет, можно поделить на две категории: обмен информацией между абонентами сети и использование баз данных сети.

К числу услуг связи между абонентами принадлежат:

Электронная почта **e-mail**, служащая для связи внутри Интернет, а также позволяющая связываться с другими сетями. Электронная почта (e-mail) - наиболее широко используемая услуга Internet. Можно посылать и принимать сообщения, поддерживая связь с миллионами людей во всем мире. Электронная почта обеспечивает все то, что предлагает телефон, факс

или письма: официальную деловую корреспонденцию, запрос информации, пересылку рецептов пирога теще, болтовню со старым другом, рассказы о новом увлечении другим энтузиастам. В электронной почте доступны списки рассылки, дающие возможность группового обсуждения конкретных тем, интересующих тех или иных пользователей Internet.

□ **Telnet** – это сетевая программа, которая позволяет с одного компьютера в Internet подсоединиться к другому. Это очень похоже на обычное соединение двух компьютеров по телефонной сети, с той лишь разницей, что для этого не требуется звонить на него с помощью модема. Telnet использует имеющиеся связи между узлами сети, что позволяет обмениваться данными между двумя компьютерами намного быстрее, чем позволяет обычный телефонный modem.

□ **File Transfer Protokol (FTP)** - протокол передачи файлов. Если Ваш коллега говорит: "Эту отличную программу можно получить по Internet!", очевидно он имеет в виду использование FTP или протокол передачи файлов (file transfer protokol), позволяющий передать программный файл из сервера ftp в ваш компьютер. В Internet существуют тысячи серверов ftp, располагающие сотнями тысяч файлов, включая бесплатные программы и графические файлы.

□ **Конференции Usenet** представляют из себя систему, включающую тысячи групп, обменивающихся новостями. Каждая такая группа похожа на электронную доску объявлений, куда каждый может поместить или считать сообщение, посвященное определенной теме. Темы могут быть простыми и узкопрофессиональными или широкими и неопределенными как политика. В Интернет проходят сотни конференций, посвященных, например, компьютерной тематике, в том числе программному обеспечению, и других, о существовании которых Вы, возможно и не подозреваете.

□ **Передача разговора** по Internet. Вы можете вести оживленную беседу с другими людьми так, что они увидят введенный вами с клавиатуры текст сразу после нажатия клавиши Enter. По каналам IRC (Internet Relay Chat - передача разговора по Internet) обсуждаются самые разнообразные, порой даже причудливые темы. По многим каналам люди просто созваниваются и болтают.

□ **Whois** – адресная книга сети Интернет. По запросу абонент может получить информацию о принадлежности удаленного компьютера, о пользователях;

□ **Finger** – получение информации о пользователях удаленного компьютера.

□ **Webster** – сетевая версия толкового словаря английского языка.

□ **Факс-сервис** – дает возможность пользователю отправлять сообщения по факсимильной связи, пользуясь факс-сервером сети.

□ **Шлюзы** – дают возможность абоненту отправлять сообщения в сети, не работающие с протоколами TCP/IP (Fido, Goldnet, AT50).

□ **World Wide Web (WWW)** — наиболее быстро развивающаяся услуга Internet, позволяющая отыскивать информацию практически на

любую тему просто отмечая заинтересовавшие вас слова, фразы или картинки. Когда говорят о Mosaic или Netscape, то тоже имеют в виду работу с WWW. Netscape и Mosaic - это программы работающие с Web. Вы можете использовать Web для чтения текстов, просмотра картинок, прослушивания звуков, проигрывания видеоклипов или загрузки программ в ваш компьютер. Страницы Web содержат все, начиная с перечня лучших программ Internet и кончая online зоопарком.

□ **Gopher** - сходная по назначению услуга, но вместо того, чтобы показывать страницы текста, Gopher показывает списки или меню. Из меню можно выбирать элементы меню до тех пор, пока нужная информация не будет найдена. Подобно WWW, Gopher предоставляет доступ к широкому спектру информации, включая тексты, картинки, звуки, видеоклипы и программное обеспечение. Программы, помогающие использовать Web, могут отображать и меню Gopher. Те же меню отображаются и самой программой Gopher.

□ **WAIS** – ещё более мощное средство получения информации, чем Gopher, поскольку оно осуществляет поиск ключевых слов во всех текстах документов.

Создание всемирной информационной паутины World Wide Web — это попытка объединить в одном информационном инструменте, под единой крышей (в хорошем смысле слова) возможности всех указанных выше средств, да еще и прибавить к ним возможности передачи не только текстов и программ, но и графики, звуков, изображений. И не только взятых каждый в отдельности, но и связанных между собой структурой гипертекста.

Выясним сначала, что же такое гипертекст? Когда мы читаем обычную книгу и встречаем незнакомое слово, то часто оно снабжается специальным значком, а внизу страницы имеется примечание с разъяснением того, что это слово означает,— это простейший случай гипертекста.

В чем же отличие гипертекстов, используемых в WWW? Возможности компьютеров позволяют включать в гипертексты не только печатные материалы, но и графику, звук, видео. Вы можете, читая текст, встретить выделенную другим цветом ссылку и тотчас же по ней получить картинку, связанную, естественно, с данной ссылкой. Затем, остановившись курсором мыши на какой-либо детали картинки, вы щелкаете мышкой и получаете новую картинку, текст или звук. И так можно делать многократно. Конечно, это отвлекает вас от основного текста, но ведь всегда можно вернуться обратно. Зато какое удобство и какая скорость!

Глава 5. Прикладные информационные технологии

Благодаря информатизации общества, стремительному развитию средств информационных и коммуникационных технологий возникает новая информационная среда обитания и жизнедеятельности. Информатизация общества – это глобальный социально-экономический процесс,

характеризующийся интенсивным производством и использованием информации в качестве общественного продукта.

Базовые информационные технологии могут быть адаптированы к требованиям той или иной сферы деятельности и использованы там в той или иной комплексации для создания соответствующих НИТ, например, в науке, технике, образовании, медицине, военном деле, оказании телефонных услуг, торговле и т.д. Эффективность подобных технологий зависит от глубины изучения посредством системного моделирования конкретной сферы деятельности, информатизации и технологизации процессов в ней, правильности выбора критериев оценки, технических и программных средств, базовых ИТ и многих других факторов.

5.1. Информационные технологии в образовании

Информатизация образования является одним из приоритетных направлений процесса информатизации современного общества. Можно прогнозировать, что глобальная информатизация образования будет одной из доминирующих тенденций цивилизации XXI века. Сегодня образование должно обеспечить формирование у людей новых компетенций, знаний и умений, способов деятельности, которые им потребуются в новой информационной среде обитания, в том числе и для получения образования в условиях широкого использования современных информационных технологий обучения. Информационная компетентность как один из основных приоритетов современного образования носит “надпредметный”, общеучебный, общеинтеллектуальный характер. Восстановление и развитие информационной компетентности должно стать первоочередной задачей этапа повышения квалификации специалистов. Основой для формирования информационной компетентности является изложенная в данном учебном пособии образовательная область «Информатика» и отражающая её прикладной аспект «Информационные технологии».

Узбекистан твердо выбрал путь независимости и прогресса, строит свою политику с учетом специфических условий жизни народа, его традиций, исторического и духовного наследия и, одновременно, использует позитивный опыт развитых государств мира, который свидетельствует о том, что уровень развития стран находится в зависимости от состояния системы образования и интеллектуального потенциала населения. Речь идет о развитии человеческих ресурсов нашей страны, т.е. о стратегии опоры не только на богатейшие залежи полезных ископаемых, но и на наращивание и использование интеллектуального потенциала. Вышеприведенные социально-экономические и геополитические факторы породили проблему принципиального реформирования образовательного процесса страны, ориентацию его на обеспечение эквивалентности мировым стандартам получаемых в Узбекистане знаний, навыков и умений, целенаправленного формирования преданности Родине и идеалам независимости, творческого мышления, профессиональной гибкости и мобильности, психологической

устойчивости, гарантирования достижения запланированных результатов образования через его технологизацию.

Главная цель технологизации процесса (явления) – гарантированное получение продукта заданного (спроектированного) образца. **Технологизация образования** – это вызванный (инициированный) необходимостью исполнения социального заказа объективный, сопровождаемый информатизацией и технизацией процесс реорганизации системы образования, включая построение её структуры и системы управления, декомпозицию образовательного процесса на этапы и управляемую пошаговую его реализацию. Современная теория и практика технологизации образования составляют круг задач **педагогических технологий**.

Хотелось бы уберечь читателя от понимания “педагогических технологий” только лишь как технизации образовательного процесса. Педагогические технологии не отменяют теорию и методику, а служат образовательным (педагогическим) целям. Но с другой стороны, нельзя недооценивать роль новых информационных технологий, которые зачастую предлагают качественно новые возможности реализации образовательного процесса.

Приобретаемые в результате обучения знания, навыки, умения, а в результате воспитания – качества, неотделимы от конкретного человека и являются продуктом преднамеренного инвестирования в его образование и здоровье. Главным инвестором в реформирование образования в Республике Узбекистан является государство. Объёмы инвестиций значительны, а конкретная отдача от них возможна через годы. Эти обстоятельства обострили вопросы гарантирования своевременной подготовки кадров, соответствующих государственным стандартам образования.

Известны традиционные ПТ, такие как эмпирические и когнитивные, интерактивные, эвристические, креативные, адаптивные и т.д. Правда, в этот перечень попали и апробированные, в целом полезные методы обучения, но не отвечающие требованиям технологизации. Применявшиеся же на практике педагогические технологии ориентированы преимущественно на активизацию, интенсификацию учебно-воспитательного процесса, иногда с претензиями на его "оптимизацию", но без акцента на гарантированность достижения результатов образования.

А как решается задача гарантированного получения продукта заданного (спроектированного) образца, например, на производстве? В промышленной сфере известны многочисленные примеры технологизации производственных процессов, благодаря которым гарантировано получение конечного продукта. Именно современные технологические процессы определяют уровень научно-технического прогресса в производственной сфере.

Педагогическая наука в целом испытывает на себе воздействие такого прогресса, заинтересованно исследует возможности гарантированного обеспечения конечной продукции, интегрирует достижения научно-

технической мысли. В частности, имеют место попытки проецирования технократического научного сознания в сферу образования, стандартизации процесса обучения. Однако здесь следует знать и помнить, что в отличие от производственной сферы технологизация образования имеет свои сложности и допустимые границы, т.к. образовательный процесс относится к организационным системам, сложность которых значительно выше сложности технических систем. Напомним, что организационная система - это сообщество людей (с собственными эгоистическими целями), объединенных общими целями и использующих общие материальные и финансовые средства для производства материальных и информационных продуктов и услуг.

Система образования Республики Узбекистан рассматривается нами как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных, непосредственно или опосредованно взаимосвязанных и взаимодействующих между собой элементов. В тоже время это «большая система» по масштабу и сложности решаемых ею задач. Ситуация усложнена ещё и тем, что с ней ассоциированы субъекты, объединенные общими целями и использующие общие материальные и финансовые средства, но одновременно обладающие собственными эгоистическими целями, не всегда совпадающими с целями системы в целом. Всё это даёт основание характеризовать систему образования как большую, организационную систему кибернетического типа. Такие системы являются объектом исследования «системного анализа» с целью их эффективной организации и научного управления ими. Системный анализ как научная дисциплина, ориентированная на исследование сложных систем, занимается проблемами принятия решений в условиях, когда выбор альтернативы требует анализа информации различной физической природы. Всякие попытки спроецировать технократическое научное сознание в сферу образования должны учитывать наличие путей разрешения возникающих при этом вышеназванных проблем.

Стратегия реформирования системы образования определена Законом Республики Узбекистан “Об образовании”, Национальной программой по подготовке кадров, директивными документами. Идеология информатизации и технологизации образовательного процесса, государственного управления им нашли своё отражение в концепции “новых педагогических технологий”. Её сущность состоит в:

- понимании образовательного процесса Республики Узбекистан как сложной системы организационного типа;
- тщательном проектировании образовательного процесса на основе корректного преодоления трудностей, связанных с необходимостью системного подхода;
- конструировании образовательного процесса в соответствии с разработанным проектом информационных технологий учебного процесса.

В свете изложенного “педагогические технологии” могут быть определены как методология системного анализа и синтеза современного образовательного процесса.

Итак, система образования Республики Узбекистан моделируется как сложная управляемая система организационного типа, в которой в качестве управляющей части выступает государство, а в качестве объекта управления - система образования Республики Узбекистан. Решение этой проблемы “в лоб” не представляется возможным. Поэтому ее моделирование строим на:

- методологии системного анализа, принципах декомпозиции и агрегирования;

- снижении уровня сложности исходной системы обоснованным переводом её фрагментов в категорию менее сложных, т.е. технических или организационно-технических систем.

Возможен следующий уровень декомпозиции и последующего агрегирования. Процесс подготовки кадров разбивается на: выбор “облика кадра”, т.е. целеполагание (Ц), а также уточнение количественных и качественных параметров выбранного “облика”, т.е. этап проектирования (П); планирование и осуществление учебного процесса, т.е. этап обучения кадра. Допускается также дальнейшая декомпозиция перечисленных этапов на подэтапы.

Разумеется, только декомпозиция не обеспечивает решение проблем, т.к. возникает вопрос адекватности решений. Необходимость же учета системности заставляет вводить связи между этапами, что, в свою очередь, усложняет теперь и декомпозированную систему. Выход видится в замене этих связей на полученные в результате экспертных оценок подмножества показателей качества продукции каждого этапа. Применительно к системе образования моделирование призвано обеспечить такую ситуацию, когда каждомуциальному субъекту-участнику образовательного процесса приходится решать относительно простые задачи этапа или подэтапа. Одновременно обеспечивается воссоединение этапов в новый, связанный процесс, который является эффективным на любом уровне детализации и общности.

Сделаем некоторые выводы из содержательной переработки информации о сфере образования в Республике Узбекистан как объекте информатизации и технологизации.

1. Образовательный процесс республики, безусловно, относится к категории сложных систем и по этой причине должен быть объектом «системного анализа».

2. Результаты такого анализа указывают на то, что система образования республики относится к категории сложных систем организационного типа и по этой причине **принципиально невозможно** новые педагогические технологии реализовать на уровне так называемой «высокой технологии», результаты исполнения этапов которой не требуется перепроверять. А это означает, что в образовании, в том числе в условиях новых информационных технологий обучения, мы **вынуждены** осуществлять контроль знаний

учащихся, ибо на протяжении всего времени изучения предмета показатели контроля знаний учащихся остаются, пожалуй, единственным критерием оценки развития учебного процесса в направлении гарантированного достижения конечных целей.

3. Не совпадающие с целями управления эгоистические цели лиц, ассоциированных с системой образования, наиболее заметны во время контроля знаний обучаемых. По этой причине уровень сложности управления традиционной (экзаменатор - студент) подсистемы контроля знаний высок. Здесь уместно воспользоваться вышеприведённым приёмом снижения уровня сложности систем обоснованным переводом их в категорию менее сложных, т.е. технических или смешанных (организационно-технических) систем. Компьютерное тестирование – показательный пример смешанных (организационно-технических) систем контроля знаний. Объективно мы обязаны обеспечить компьютерное тестирование знаний, разнообразить формы тестирования. В то же время необходимо унифицировать тестовые задания. В настоящее время разработка тестовых заданий является прерогативой преподавателя, т.е. вынесена в процессуальную часть. Это обстоятельство порождает разнобой в составлении тестов и их низкое качество; проблемы в обучении персонала составлению тестовых заданий; потерю времени и материальных средств. Следствием такого положения являются проблемы по выполнению государственных стандартов образования. Выход нам видится в переводе проблемы тестовых заданий по предметам на уровень содержательной части обучения, поручении составления тестовых заданий экспертам. Созданные унифицированные тестовые задания по предметам обучения подлежат рассылке в учреждения образования для использования их в качестве контроля знаний.

4. Правильный выбор модели управления образовательным процессом предопределяет эффективное решение поставленных задач, результативность в достижении общих и конкретных целей обучения и воспитания. Модель управления влияет на разработку соответствующей **содержательной части** обучения, определяя совокупность изучаемых дисциплин, общие и конкретные цели обучения, содержание учебного материала, объем часов и т.д. Всё это позволяет сформулировать концепцию преподавания того или иного предмета, которая может быть отражена в типовой программе. В части конструирования обучения НПТ практически не регламентируют методы и формы работы учителя и обучаемого, но через государственные стандарты образования стимулируется деятельность учителя в части диагностики учебного процесса, повышения эффективности управления процессом обучения. Таким образом, проявляется особенность новых педагогических технологий, которая состоит в том, чтобы при определенной заданности содержания обучения в реализации **процессуальной части** была бы предоставлена некоторая свобода (в выборе методов и методик обучения), регламентированная общими и частными целями обучения и воспитания. В значительной степени это связано с усилением pragматического аспекта

(важно, чтобы был результат и польза) новых педагогических технологий. Последнее обстоятельство стимулирует совершенствование частной методики преподавания, отбор и использование эффективных методик, формирования современной инструментовки учебно-воспитательного процесса на основе применения современных информационных технологий. В этом плане поучителен пример резко возросшего в последние годы интереса к известным в прошлом (более 40 лет тому назад) так называемым интерактивным методам обучения. Вызван он необходимостью подготовки конкурентоспособных кадров, имеющих собственное мнение о предмете обсуждения, способных высказывать и отстаивать его. Разумеется, что этот интерес представляет возврат к известным методам на новом витке дидактической спирали. НИТ позволяют активнее использовать такие методы обучения, как дебаты, моделирование, ролевые игры, дискуссионные группы, мозговые атаки, методы Дельфи, методы номинальной группы, форумы, проектные группы. Так, метод "мозговой атаки" представляет собой стратегию взаимодействия, позволяющую группам студентов эффективно генерировать идеи. Этот метод поощряет членов группы мыслить творчески и развивать идеи других членов группы. Основной целью метода «мозговой атаки» является создание фонда идей по определенной теме. При мозговой атаке исключается критицизм, поощряются свободные ассоциативные суждения. Процедура Дельфи представляет собой метод для выработки надежного консенсуса номинальной группы студентов посредством серии анкетных опросов. Термин *номинальная группа* происходит от того, что студенты только номинально представляют собой группу на первоначальной стадии генерации идей. Первоначально каждого участника такой группы просят сформулировать и проранжировать идеи. Затем составляется общий список идей, которые получили самый высокий приоритет у отдельных участников, затем вторые по значимости и т.д. до тех пор, пока список у каждого участника не будет исчерпан. После этого все приглашаются к обсуждению идей. После дискуссии проводится голосование, в ходе которого членов группы просят проранжировать идеи, генерированные в ходе дискуссии.

5. Понимание процесса обучения как информационный процесс порождает необходимость уточнения модели познавательного процесса. Восприятие образовательного процесса как информационного позволяет выделить следующие компоненты в обучении:

- субъект, сообщающий учебную информацию;
- собственно сама учебная информация;
- субъект, принимающий эту информацию;
- процесс познания.

В традиционном обучении процесс познания реализуется в ходе общения обучающего с обучаемым посредством методик обучения, нацеленных на «перекачку» знаний от учителя к ученику. Сегодня процесс познания квалифицируется как информационный. Учебная информация может стать знанием обучаемого, если только она принята, понята и

осознана им как личностно значимая. Отсутствие любого из названной последовательности этапов (условий) означает безуспешность познавательного процесса. Учебная информация может быть принята обучаемым, если он мотивирован. Мотивированность обучаемого формируется внешними (вплоть до принуждения к учению) или внутренними (мотивация учения) мотивами. Мотивацию учения формирует содержание учебной информации и процесс (методика, педагогическое мастерство) обучения. Принятая информация может быть понята, если имеет место соответствие содержания учебной информации тезаурусу обучаемого. В противном случае необходимо повысить уровень знаний обучаемого или понизить сложность изложения учебной информации. Осознание личностно значимой, полезной или бесполезной информации требует от обучаемого определённых интеллектуальных усилий вплоть до изменения структуры собственных представлений. Всё это свидетельствует о том, что результативность процесса обучения зависит от состояния обучаемого. Поэтому сегодня требуются методики, которые позволяют организовать обучение как процесс управления состоянием обучаемого. Предложенная модель познавательного процесса уточняет современные представления о нём педагогической науки, позволяет осуществлять управление познавательной деятельностью и способствует совершенствованию обучения.

6. Конструирование образовательного процесса предполагает выявление и применение новых форм обучения. В настоящее время в сфере образования встает проблема поиска методов и организационных форм обучения, повышающих его педагогическую эффективность. В разрешении этой проблемы весьма перспективным представляется использование «дистанционного образования» путём адаптации его в учебный процесс действующих учреждений образования в форме дополнительного дистанционного обучения (ДДО).

На пути такого решения проблемы возникают трудности, связанные с тем, что:

1. В современной литературе, посвящённой проблемам образования, термины “обучение” и “образование” очень часто употребляются как синонимы. В действительности, это не тождественные понятия, и различие между ними кроется, прежде всего, в качестве достигаемых результатов.

2. ДО, в первую очередь, виртуальная образовательная среда. Но в попытках приспособить её к потребностям обучения порой хотят видеть копию традиционного образования, только реализованную на современной технической базе.

Обучение предполагает чёткое понимание того, чему следует учить. Поэтому оно, так или иначе, направлено на формирование конкретных, ограниченных целями обучения знаний, умений, навыков. При традиционном подходе студент слушает лекции, ведет конспекты, посещает библиотеки, семинары, т.е. он фактически встроен в организованный учебный процесс.

Отличительная особенность ДО состоит в том, что значительная часть времени приходится на самостоятельную познавательную деятельность обучающегося, т.е. самообразование. Самообразование – это целенаправленная, систематическая, индивидуальная, самостоятельная работа по повышению профессионального мастерства и всестороннему развитию личности. В процессе самообразования человек из доступных ему средств обучения и источников информации использует, на его взгляд, адекватные целям, поставленным им самим. Для достижения целей самообразования человеку необходимо обладать очень высокой внутренней мотивацией, самодисциплиной, владеть умениями интеллектуального труда, исследовательскими навыками и иметь доступ к различным источникам информации как традиционным, так и связанным с передовыми средствами новых информационных технологий.

Как известно, обучение представляет собой сложный процесс, в котором участвуют обучающий, обучаемый и имеет место их общение. Проблема общения обучающего с обучаемыми в процессе обучения исследована педагогической наукой и практикой, выявлены формы такого общения, начиная от монолога учителя, диалога и, кончая интерактивным диалогом сторон. В условиях традиционной образовательной системы фактор непосредственного общения сторон в процессе обучения, его значимость бесспорны и этот факт признаётся «по умолчанию».

В условиях дистанционного обучения непосредственное общение утратило свой классический смысл и реализуется как взаимодействие в опосредственных, преимущественно в «отложенных» формах.

Напомним, что дистанционное обучение по реализуемым способам взаимодействия (пары: обучающий – обучаемый) в процессе обучения подразделяются на синхронные и асинхронные. В синхронных – в процессе обучения происходит взаимодействие обучающего и обучаемого в формах интерактивного аудио, текстового, графического, видео диалога, компьютерных телеконференций и др., осуществляемых с помощью телекоммуникационных средств. В асинхронных – обучаемый и обучающийся действуют независимо друг от друга. Асинхронные системы включают в себя как традиционные формы, с использованием печатных материалов, аудио- и видеокассет, так и новые компьютерные технологии, в частности, электронные учебники, автоматизированные обучающие системы, экспертные обучающие системы и др. с использованием электронной почты, «всемирной паутины», файловых архивов и прочих компьютерных технологий. Но при этом учебная информация «отложена» во времени до момента её актуализации и использования. Итак, при реализации дистанционного обучения существует ограничение общения до уровня взаимодействия. Это один фактор проблем.

В традиционном образовании реализуется технология общения обучающего с обучаемым посредством методик обучения. При этом одно несомненно: роль методики преподавания, преподавателя как носителя частной методики продолжает быть высокой. Совершенно очевидно, что

высокая эффективность обучения находится в зависимости от состояния участников процесса, т.е. пары учитель-ученик, их уровня мотивации, педагогического мастерства учителя и т.д.

Реальным путем интенсификации учебного процесса остается включение в него других, не слуховербальных способов предъявления информации. В первую очередь речь идет об информации зрительной и слуховой (невербальной). Дальнейшим шагом на этом пути является комплексное предъявление разнородной, но взаимосвязанной по смыслу информации в одновременном сочетании. Разнопредставленная и взаимосвязанная по смыслу информация в одновременном сочетании и эстетическом воздействии способствует возникновению эмоций и как следствие эмоциональному восприятию учебной информации. Это другой фактор проблем.

Очень желаемым является переход процесса обучения в новое состояние, которое характеризуется как режим самообучения ученика. И тогда помимо традиционных будут востребованы информационные технологии обучения такие, как мультимедиа-курсы на лазерных дисках, Интернет, WEB-курсы, виртуальные библиотеки, дистанционное образование и др. Но все это возможно, если процесс обучения будет реализован как процесс управления состоянием обучаемого. Это ещё один фактор проблем.

Сегодня дистанционное обучение имеет широкую палитру средств доставки учебно-методической информации: Интернет, электронная почта, аудио- и видеотехника, дополняющие традиционные носители учебной и обучающей информации. Для обучаемых должны быть созданы и доступны все необходимые информационные и учебно-методические ресурсы:

- текстовые;
- графические;
- аудио- и видеоматериалы;
- обучающие программы и виртуальные лаборатории;
- методические разработки;
- планы занятий и т.п.

Это очередной фактор проблем, решению которых способствуют:

- формирование и развитие центров информационных, учебно-методических ресурсов на базе локальной вычислительной сети учреждений образования;

- оснащение локальной вычислительной сети учреждений образования высокопроизводительной, компьютерной и коммуникационной техникой;

- внедрение информационной Инtranет-системы со всем набором псевдо Интернет услуг;

- реализация Инtranет-технологии в качестве тренажёра умений и навыков общения с Интернет;

- создание особых типов электронных учебных пособий и обучающих программ. Здесь в первую очередь имеется в виду создание электронных учебных материалов и компьютерных систем контроля знаний обучаемых,

которые позволяют снизить сложность исходного образовательного процесса как организационной системы до уровня образовательного процесса как организационно-технической системы.

При осуществлении самостоятельного учения в виртуальной образовательной среде с опорой на рекомендуемые учреждением образования комплект учебно-методических материалов, учебные планы и др. можно говорить об обучении в форме дополнительного дистанционного обучения.

Таким образом, теория обучения предмету выявляет закономерности функционирования методической системы обучения этому предмету, методика строит их приложения, а технология разрабатывает способы реализации модели этой системы. При таком подходе роль технологии сводится к диагностированию целей и выявлению условий (методов, форм, средств, зависимостей), т.е. к проектированию процесса, осуществление которого позволит достичь намеченные цели.

5.2. Об использовании систем искусственного интеллекта в образовании

Среди современных видов информационных технологий особое место принадлежит СИИ. Разработка СИИ предполагает сделать знания объектом обработки на ЭВМ. В конце 70-х в исследованиях по ИИ сформировалось самостоятельное направление, получившее название "экспертные системы". Задачу данной научно-технической дисциплины Фейгенбаум определил как "...привнесение принципов и инструментария исследований из области ИИ в решение трудных прикладных проблем, требующих знаний экспертов". Существующим ЭС свойственна символьная обработка. В отсутствии символьных машин пятого поколения, благодаря ориентации на конкретную предметную область ЭС могут быть реализованы на ЭВМ традиционной архитектуры.

Особенность нашего подхода состоит в том, что исходно символьную информацию с помощью специальных метакодов предлагается перевести в числовую форму и таким путем представление знаний, а также «интеллектуальные» действия свести к задачам вычислительного характера над арифметическими объектами. Были разработаны алгоритмы числового представления знаний и манипулирования ими, в основу которых положено кодирование знаний простыми числами. Последнее порождает коды в виде больших чисел, которые не умещаются в разрядную сетку существующих ЭВМ, что вызывает сложности для их практической реализации. Нами был разработан комплекс программ, позволивший решить проблему больших чисел. В него входит основной набор операций – сложения, вычитания, умножения, деления, сравнения, а также ряд функций типа выделения целого из десятичной дроби, выделения дробной части. Были использованы алгоритмы так называемой "модулярной" арифметики, включающей в себя операции сложения, вычитания, умножения и деления больших чисел.

Благодаря этому стало возможным по иному организовать работу интеллектуального интерфейса в части автоматического решателя задач и базы знаний, а также практическое использование концепции кодирования знаний простыми числами в реализации СИИ.

Более поздние наши исследования были направлены на третью компоненту интеллектуального интерфейса, т.е. на диалоговый процессор (ДП), в предположении о том, что эффективность экспертных систем зависит не только от уровня автоматического решателя задач и способа организации баз знаний, но и от уровня организации ДП в них. В частности, интеллектуализация ДП расценивается нами как средство настройки запроса на особенности ЭС. В этой связи отметим, что традиционно рассматривалась способность адаптации СИИ к исходной ситуации. В определенных предметных областях, наряду с традиционной адаптацией интеллектуальной системы к ситуации, возникает возможность адаптации запроса к особенностям ЭС, а также возможность так называемой встречной адаптации.

Некоторые результаты этих исследований адаптированы нами к потребностям образования.

5.2.1. Новые информационные технологии в обучении и развитии интеллектуального потенциала одаренной молодежи

Среди многих задач системы образования Республики Узбекистан по подготовке кадров в целом, существует проблема прогнозирования, выявления и работы с одаренной молодежью с целью закрепления и развития её интеллектуального потенциала. Для решения этих проблем предпринимаются различные усилия, в том числе специальная подготовка преподавателей, призванных работать с одаренной молодежью, разработка теории и практики развивающего обучения и воспитания, включая педагогические, психологические, психофизиологические аспекты и др.

Применение информационных технологий, безусловно, повышает уровень работы с одаренными студентами. К числу таких технологий можно отнести автоматизированные обучающие системы (АОС). Но и они имеют ряд недостатков, которые затрудняют, а порой делают невозможным, достижение целей и задач работы с одаренными студентами. Так, присущие АОС предварительная разработка сценария и жесткое следование ему в ходе занятия не позволяют обучаемому импровизировать в процессе обучения, не создают условий для развития творческого мышления у него.

Одним из возможных путей преодоления этих недостатков АОС видится в их интеллектуализации, т.е. в необходимости сделать их “мыслящими”, чтобы обеспечить равноправное общение обучающей системы и обучаемого. Поэтому актуальным является решение проблемы интеллектуализации информатизационных систем обучения и воспитания одаренной молодёжи. Разработке и реализации одного из вариантов решения этой проблемы посвящены наши исследования.

В АОС преобладает пассивный диалог. Между тем, при работе с одаренной молодёжью интерес представляют программы активного диалога, когда действия обучаемого являются определяющими, а изучение материала происходит через познавательную деятельность, от её репродуктивной формы к исследовательской и творческой. Активный диалог «равных» партнеров усиливает мотивацию учащегося к обучению, прививает навыки самостоятельной работы.

Активность ученика в освоении знаний повышается, если ему предоставлена возможность самостоятельно конструировать задания и выполнять их в «содружестве» с интеллектуальной АОС. Это целесообразно делать потому, что ученик может сомневаться в точности своих знаний при выполнении некоторых операций или действий, и будет иметь потребность проверить себя дополнительно именно в требуемом для него аспекте. Вместе с тем важно, чтобы инициатива в изменении направления обучения переходила от одного его участника к другому, ибо системы обучения, в которых изначально заложено лишь одностороннее, машинное управление процессом обучения, в случае работы с одаренной молодёжью не могут считаться эффективными.

Традиционная для нашей школы методика обучения решению задач, когда учитель разбирает типовой пример, а затем ученики решают аналогичные задачи, следуя образцу, не самая лучшая.

Известно, что процесс поиска решений широкого круга задач укладывается в следующую схему: пробные действия и анализ их результатов, выдвижение гипотез, их проверка и коррекция, комбинирование различных идей, проверка найденного решения. Необходимо развивать у учеников навыки реализации этой схемы.

Традиционные же системы, в которых заранее заданы ответы и ход решения задач, подавляют оригинальное, нешаблонное мышление ученика. Такие системы наказывают как ученика, не усвоившего материал, так и ученика, который пытается думать творчески, своеобразно. Если для решения задачи в программе будет заложен жесткий алгоритм, используя который ученик получит нужное решение, то память ученика засоряется огромным объемом процедурных знаний и, вообще, исключает необходимость не только понимать, но и размышлять, думать.

Один из путей преодоления этих проблем - применение методов искусственного интеллекта (ИИ), создание интеллектуальных обучающих систем. Только сочетание естественного интеллекта ученика и искусственного “интеллекта” компьютера может дать ожидаемый педагогический эффект.

Работа квалифицированного педагога, в том числе и специально подготовленного для работы с одарённой молодёжью, плохо формализована, не может быть сведена к строгому алгоритму, однако поддается компьютеризации (компьютерный имитатор педагога) посредством создания экспертных обучающих систем [1].

Экспертно-обучающая система (ЭОС) – это система нового поколения, создаваемая на основе практического использования элементов ИИ для моделирования действий экспертов по дидактике и созданию баз знаний (БЗ) учебного назначения.

ЭОС можно рассматривать, как совокупность трех взаимодействующих ЭС: 1. ЭС по решению задач в изучаемой предметной области (ЭСРЗ), 2. ЭС по диагностике ошибок обучаемого (ЭСДО), 3. ЭС по планированию процесса обучения (ЭСУУ).

В структуре ЭОС можно выделить следующие базы знаний: учебная БЗ для данной ПО; модель обучаемого; БЗ о возможных ошибках обучаемого; БЗ о процессе обучения.

Экспертные знания об изучаемой ПО, управление процессом обучения, анализ ошибок обучаемого в ЭОС отделены от конкретного курса обучения, что позволяет повысить гибкость, адаптивность процесса обучения и организовать многоцелевое использование экспертных знаний.

Эффективность ЭСРЗ определяют не только скоростью, но и результативностью решения задач, т.е. способностью системы доводить решение задачи до конца. Результативности, в частности, способствует упрощение алгоритмов, вплоть до изменения технологии решения задач. Так, традиционная задача определяется как совокупность трех фактов:

- 1) множество известных параметров для решения задачи;
- 2) неизвестное, которое нужно найти;
- 3) алгоритм решения задачи, при выполнении которого из множества известных параметров находится неизвестное.

В отличие от этого будем понимать задачу как множество исходных данных и неизвестное. При этом несколько иначе будем понимать решение задачи. Решение задачи – это такое последовательное расширение постановки задачи в части множества исходных данных, при котором искомое находится одним действием [2].

Под решением одним действием будем понимать такую полноту постановки задачи, при которой в базе знаний обнаруживается правило, содержащее искомое решение в качестве единственного неизвестного.

Если задача не решается в одно действие, то это означает, что для такого решения не хватает данных и следует просто расширить постановку задачи. Суть предлагаемого алгоритма состоит в том, что просмотр базы знаний ведется до первого правила, которое содержит какое-либо единственное неизвестное, следовательно, это неизвестное может быть найдено из этого правила одним действием. Теперь и полученным результатом можно расширить постановку задачи. Таким образом, продолжается расширение постановки задачи до тех пор, пока искомое решение не будет найдено одним действием.

Коротко остановимся на этапах решения задач в предлагаемой ЭОС. Решение задачи условно можно разбить на два этапа. Первый этап – решение задачи в принципе, т.е. выясняется, разрешима ли задача, и второй – решение задачи непосредственно в значениях. Обычно ЭС с прямой цепочкой вывода

решения, такого разделения не делают. Действие выполняется сразу, как только находится нужное правило.

При нашем подходе на первом этапе решения задачи просматривается база знаний и находятся неизвестные, которые могут быть найдены одним действием. Исходная постановка расширяется найденными фактами до тех пор, пока не будет найдено искомое или выяснится, что задача не разрешима. Данный подход избыточен: находятся все неизвестные, которые можно найти, в том числе и те, которые впоследствии непосредственно не используются для решения задачи.

Для того чтобы избежать излишних действий, создаётся рабочее подмножество базы знаний (РПБЗ), т.е. вспомогательная БЗ. В него записываются правила из основной базы знаний, которые использовались на первом этапе. После этого используется обратная цепочка вывода решения по вспомогательной базе знаний. В результате получается конкретная последовательность правил, приводящая к искомому решению, а в завершение производится непосредственное решение задачи в числах.

Приведенные выше соображения реализованы в прототипе ЭОС - “Треугольник” на примере школьного курса планиметрии, раздела решения задач о треугольниках. Данная система позволяет ученику вести активный диалог, предоставляя ему возможность самостоятельно ставить, решать задачу и проверять при этом правильность ее решения. Система имеет два режима работы: решение задачи системой и проверка задачи, решенной учеником. В первом режиме осуществляется решение задачи по описанному выше алгоритму. Во втором режиме ведется диалог для осуществления контроля правильности решения задачи учеником.

Наличие объяснительной функции является важным качеством любой ЭС. ЭОС “Треугольник” обладает таким качеством: ход решения задачи может быть объяснен по желанию пользователя.

Данная система позволяет демонстрировать возможности современных информационных технологий в автоматическом решении задач из некоторой ПО. Для ученика она выступает в роли тренажёра, позволяющего ставить произвольную задачу треугольника, решать и проверять решение такой задачи. Для учителя – это обучающая программа, формирующая и закрепляющая навыки решения задачи. Для разработчиков – это экспертная система, интеллектуальный интерфейс которой (БЗ и автоматический решатель задач) реализует выдвинутую автором оригинальную методологию числового отображения знаний в системах ИИ.

5.2.2. Интеллектуальная система профессионального тренинга

Важным компонентом современных интеллектуальных систем, с которым непосредственно сталкиваются их разработчики и пользователи, является диалоговый процессор (ДП). Эффективность интеллектуальных систем зависит не только от уровня автоматического решателя задач и базы знаний, но и от уровня организации диалогового процессора. В существующих системах имеет место регламентация сообщений пользователя, что существенно сокращает

свободу его действий и снижает эффективность взаимодействия пары человек-машина. Регламентация вызвана сложностями смыслового анализа системой сообщений пользователя.

В классических СИИ ресурсы системы формируются и используются для выявления семантики запроса. Применение в рассматриваемых нами предметно-ориентированных системах известных способов определения смысла сообщения, например, лингвистических процессоров понимания смысла естественных языков, обладающих средствами семантического анализа текстов, мало эффективно. В данном случае не требуется универсальность ДП, т.к. предметная область (ПО) всегда может быть определена. В силу определенного контекста диалога и ожидаемого смысла сообщения пользователя, неоднозначность (омонимия, полисемия, синонимия) диалога минимальна. К тому же структура диалога и отдельных выражений также может быть определена из контекста. В нашем случае нет необходимости определять смысл запроса пользователя (как и вообще, определять, что такое «смысл» и как его понимает машина). Пользователю важно, чтобы система адекватно реагировала на его запрос. Для этого необходимо так организовать ДП, чтобы он в паре с пользователем обеспечил восприятие запроса системой. Фактически речь идет о необходимости интеллектуализации ДП, о выборе такого уровня интеллектуализации, при котором обеспечивается его взаимодействие с пользователем в режиме «равных партнеров». В предметно-ориентированных системах таким способом можно задачу создания сложной интеллектуальной системы свести к созданию вопросно-ответной системы с интеллектуальным ДП, который в паре с пользователем способен распознать ситуацию и привести ее к виду, уже имеющемуся в базе стандартных запросов (БСЗ) системы.

В Национальной программе по подготовке кадров указано на необходимость «разработать, создать и освоить на практике передовые технологии и оборудование профессионального тренинга, а также имитаторов сложных, научно-технических технологических процессов». Одним из путей разработки и создания систем профессионального тренинга видится в использовании современных информационных технологий, компьютерной имитации наставника профессионального обучения, в частности, имитации мастера по ремонту сложной бытовой и электронной техники, посредством создания вопросно-ответных ЭОС. В настоящее время обучение в профессиональном образовании слабо формализовано, т.к. деятельность квалифицированного специалиста (особенно интеллектуальная её часть) не всегда может быть сведена к строгому алгоритму. Однако в таких профессиональных видах обучения, как ремонт телевизоров, стиральных машин и т.п., т.е. узких предметных областях, имеются справочники по ремонту, способные составить основу базы знаний конкретных ПО. Роль обучаемого в таких сферах профессиональной деятельности можно представить себе как попытку в «дружественном диалоге» с компьютером четко сформулировать состояние ремонтируемой техники. Если это сделано, т.е. ситуация сформулирована, как это имеет место в справочнике, то выдаётся диагноз поломки с возможным

перечислением действий по ремонту. Участвуя в подобной работе с ЭОС профессионального тренинга в режиме «равных партнеров», обучаемый приобретает профессиональные умения и навыки, как в правильной формулировке поломок, так и в ремонте бытовой техники.

Предлагаемый алгоритм создания образа запроса и его восприятия системой предусматривает наличие набора ключевых фраз, закодированных простыми числами, набора правильных запросов, закодированных как произведение кодов, составляющих их ключевых слов и параметров. Если установлено полное соответствие между запросом пользователя и одним из образцов запросов в БСЗ и это соответствие подтверждено пользователем, то запрос передается на обработку в систему. В противном случае диалог продолжается, с целью уточнения деталей ситуации. В ответ на запрос ДП предлагает пользователю серию образцов запросов, которые «на его взгляд» наиболее отражают ситуацию. Серию составляют тексты запросов из БСЗ, которые отличаются от введенного пользователем запроса только на одну ключевую фразу. После получения уточнений от пользователя обновленный запрос снова передается в ДП. Диалог ведется до полного распознания запроса.

Литература

1. Абуталиев Ф.Б., Юлдашев У.Ю. Модели гибридных систем проектирования. – Ташкент: АН РУз, НПО Кибернетика, 1993.
2. Кадыров Б.Г., Юлдашев У.Ю. Проблемы развития технологизации образования Республики Узбекистан// Таълимда ахборот технологиялари мавзусидаги Республика илмий-амалий конференция материаллари. - Ташкент, 2000.
3. Макарова Н.В. и др. Информатика.- М.: Финансы и статистика, 1997.
4. Симонович С., Евсеев Г., Алексеев А. Общая информатика. Учебное пособие – М.: Аст-Пресс: Инфорком-Пресс, 1999.
5. Юлдашев У.Ю., Бакиев Р.Р., Закирова Ф.М. Информатика. Касб-хунар колледжлари учун дарслик. – Тошкент, 2002.
6. Юлдашев У.Ю., Закирова Ф.М. Методика преподавания информатики. Учебник для педагогических вузов. – Ташкент, 2005.

ОГЛАВЛЕНИЕ

Глава 1. Информационные технологии.....	3
1.1.Информационные технологии в информационном обществе	3
1.2. Классификация компьютеров, применяемых в ИТ.....	6
1.3. Новые информационные технологии.....	7
Глава 2. Информационные системы.....	9
2.1. Информационные «Инtranет – системы».....	13
2.2. Системы искусственного интеллекта.....	16
2.3. Экспертные системы.....	21
Глава 3. Базовые информационные технологии.....	22
3.1. Технологии архивирования данных.....	22
3.2. Вирус – технологии.....	24
3.3. Технологии БД и СУБД.....	26
3.4.Технологии СУБД ACCESS.....	31
3.5. Мультимедиа технологии.....	63
3.6. Сетевые технологии.....	69
Глава 4. Базовые информационные технологии высокого уровня	72
4.1. Интернет технологии.....	72
Глава 5. Прикладные информационные технологии.....	77
5.1. Информационные технологии в образовании.....	78
5.2. Об использовании систем искусственного интеллекта в образовании.....	87
5.2.1.Новые информационные технологии в обучении и развитии интеллектуального потенциала одаренной молодежи.....	88
5.2.2.Интеллектуальная система профессионального тренинга.....	91
Литература.....	94

Учебно – методическое пособие
Часть 2
(на русском языке)

Редактор, тех. редактор - Т.Тимофеева

Заказ №55 Объём 6,1 п.л.
Тираж 200 Дата разрешения 05 – 07 – 2007
Отпечатано на ризографе ТГПУ имени Низами.