**Билеты**

# Информационные системы и технологии

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

Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации персонального компьютера. В крупных организациях наряду с персональным компьютером в состав технической базы информационной системы может входить мэйнфрейм или суперЭВМ. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

**Информационная технология** – процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта).

Цель информационной технологии – производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-либо действия.

# Этапы развития информационных технологий

**Информационная технология—**процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления. Современная информационная технология опирается на достижения в области компьютерной техники и средств связи.

* 1 этап (до второй половины **XIX** в.) — **«ручная» информационная технология**, инструментарий которой составляли: перо, чернильница, книга. Коммуникации осуществлялись ручным способом путем отправки по почте писем, пакетов, депеш. Основная цель технологии — представление информации в нужной форме.
* 2 этап (с конца **XIX** в.) — **«механическая» технология**, инструментарий которой составляли: пишущая машинка, телефон, диктофон, оснащенная более совершенными средствами доставки почта. Основная цель технологии — представление информации в нужной форме более удобными средствами.
* 3 этап (40 - 60-е гг.) — **«электрическая» технология**, инструментарийкоторой составляли: большие ЭВМ и соответствующее программное обеспечение, электрические пишущие машинки, копировальные аппараты, портативные диктофоны. Изменяется цель технологии — акцент начинает перемещаться с формы представления информации на формирование ее содержания.
* 4 этап (с начала 70-х гг.) — **«электронная» технология**, инструментарием которой становятся большие ЭВМ и создаваемые на их базе автоматизированныесистемы управления (АСУ) и информационно-поисковые системы (ИПС), оснащенные широким спектром базовых и специализированных программных комплексов. Акцент смещается в сторону формирования более содержательной информации.
* 5 этап (с середины 80-х гг.) — **«компьютерная» технология**, основным инструментарием которой является персональный компьютер с широким спектром стандартных программных продуктов разного назначения. На этом этапе происходит процесс персонализацииАСУ (создание систем поддержки принятия решения для разных специалистов). В связи с переходом на микропроцессорную технологию существенным изменениям подвергается бытовая техника, приборы связи и коммуникации, оргтехника. Начинают широко развиваться компьютерные сети (локальные и глобальные). Наблюдается так называемая ***компьютеризация общества***. Под этим термином понимают задачи массового внедрения компьютеров во все области жизни людей, а также последствия, которые будут вызваны этим массовым внедрением компьютеров.
* ***Компьютеризация*** — это сложный социальный процесс, связанный со значительными изменениями в образе жизни населения с целью улучшения качества жизни за счёт увеличения производительности и облегчения условий труда. Этот процесс требует серьёзных усилий на многих направлениях, включая ликвидацию компьютерной неграмотности, формирование культуры использования новых информационных технологий.

# Особенности новых информационных технологий

***Новая информационная технология -***информационная технология с "дружественным" интерфейсом работы пользователя, использующая персональные компьютеры и телекоммуникационные средства.

Прилагательное "компьютерная" подчеркивает, что основным техническим средством ее реализации является компьютер.

Три основных принципа новой (компьютерной) информационной технологии:

* интерактивный (диалоговый) режим работы с компьютером;
* интегрированность с другими программными продуктами;
* гибкость процесса изменения как данных, так и постановок задач.

Для эффективного взаимодействия конечных пользователей с вычислительной системой новые информационные технологии опираются на принципиально иную организацию интерфейса пользователей с вычислительной системой (так называемого дружественного интерфейса), который выражается прежде всего в следующем:

* в обеспечении права пользователя на ошибку благодаря защите информационно-вычислительных ресурсов системы от непрофессиональных действий на компьютере;
* в наличии широкого набора иерархических меню, системы подсказок и обучения и т.п., облегчающих процесс взаимодействия пользователя с ПК;
* в наличии системы "отката", позволяющей при выполнении регламентированного действия, последствия которого по каким-либо причинам не удовлетворили пользователя, вернуться к предыдущему состоянию системы.

Проблемы использования информационных технологи

**Для**информационных технологий является вполне естественным то, что они ***устаревают***и заменяются новыми.

# Общая классификация видов информационных технологий.

В настоящее время классификация ИТ проводится по следующим признакам:

· по способу реализации в автоматизированных информационных системах (АИС),

· по степени охвата задач управления,

· по классам реализуемых технологических операций,

· по типу пользовательского интерфейса,

· по вариантам использования сети ЭВМ,

· по обслуживаемой предметной области и др.

1) ***По способу реализации ИТделятся*** на традиционные и современные ИТ. Традиционные ИТ существовали в условиях централизованной обработки данных, до периода массового использования ПЭВМ. Они были ориентированы главным образом на снижение трудоемкости пользователя (например, инженерные и научные расчеты, формирование регулярной отчетности на предприятиях и др.). Новые (современные) ИТ связаны в первую очередь с информационным обеспечением процесса управления в режиме реального времени.

2) ***По степени охвата информационными технологиями задач управления*** выделяют: электронную обработку данных, автоматизацию функций управления, поддержку принятия решений, электронный офис, экспертную поддержку.

3) ***По классу реализуемых технологических операций ИТ*** *подразделяются: на* работу с текстовым и табличным процессорами, графическими объектами, системы управления БД, гипертекстовые и мультимедийные системы.

**Компьютерная графика** *-* это создание, хранение и обработка моделей объектов и их изображений с помощью ЭВМ.

**Гипертекстовая технология** *-* организация текста в виде иерархической структур Материал текста делится на фрагменты.

**Мультимедиа-технология** *-* программно-техническая организация обмена с компьютером текстовой, графической, аудио и видеоинформацией.

4) ***По типу пользовательского интерфейса*** можно рассматривать ИТ с точки зрения возможностей доступа пользователя к информационным и вычислительным ресурсам. Так, пакетная ИТ исключает возможность пользователя влиять на обработку информации, пока она проводится в автоматическом режиме. В отличие от пакетной диалоговая ИТ предоставляет пользователю неограниченную возможность взаимодействовать с хранящимися в системе информационными ресурсами в реальном масштабе времени, получая при этом всю необходимую информацию для решения функциональных задач и принятия решений.

# Информационные технологии в разных областях

5) ***По обслуживаемым предметным областям*** *ИТ* подразделяются разнообразно. Например, только в экономике ими являются, бухгалтерский учет, банковская, налоговая и страховая деятельность и др.

Рассмотрим типичные применения информационных технологий, применяемых в управленческой системе предприятия:

*Бухгалтерский учет* - классическая область применения информационных технологий и наиболее часто реализуемая на сегодняшний день задача. Во-первых, ошибка бухгалтера может стоить очень дорого, поэтому очевидна выгода использования возможностей автоматизации бухгалтерии. Во-вторых, задача бухгалтерского учета довольно легко формализуется, так что разработка систем автоматизации бухгалтерского учета не представляет технически сложной проблемы.

*Управление финансовыми потоками.* Внедрение информационных технологий в управление финансовыми потоками также обусловлено критичностью этой области управления предприятия к ошибкам.

*Управление складом, ассортиментом, закупками.* Можно автоматизировать процесс анализа движения товара.

*Управление производственным процессом* представляет собой очень трудоемкую задачу. Основными механизмами здесь являются планирование и оптимальное управление производственным процессом.

Автоматизированное решение подобной задачи дает возможность грамотно планировать, учитывать затраты, проводить техническую подготовку производства, оперативно управлять процессом выпуска продукции в соответствии с производственной программой и технологией.

*Управление маркетингом* подразумевает сбор и анализ данных о фирмах-конкурентах, их продукции и ценовой политике, а также моделирование параметров внешнего окружения для определения оптимального уровня цен, прогнозирования прибыли и планирования рекламных кампаний.

*Документооборот* является очень важным процессом деятельности любого предприятия. Хорошо отлаженная система учетного документооборота отражает реально происходящую на предприятии текущую производственную деятельность и дает управленцам возможность воздействовать на нее. Поэтому автоматизация документооборота позволяет повысить эффективность управления.

*Оперативное управление предприятием.* Информационная технология, решающая задачи оперативного управления предприятием строится на основе базы данных, в которой фиксируется вся возможная информация о предприятии. Информационная система оперативного управления включает в себя массу программных решений автоматизации бизнес-процессов, имеющих место на конкретном предприятии.

# Информационные технологии в обучении.

***Автоматизированная обучающая система (АОС) -*** комплекс программных, технических и учебно-методических средств, предназначенных для активного индивидуального обучения человека на основе программного управления этим обучением.

Благодаря своим конструктивным и функциональным особенностям современный персональный компьютер находит применение в обучении самым разнообразным дисциплинам и служит базой для создания большого числа новых информационных технологий обучения.

Компьютерная технология повышает интерес к обучению. В настоящее время существует огромное множество обучающих программ по самым разным предметам, ориентированных на самые различные категории учащихся, начиная с детских садов и заканчивая персоналом атомных электростанций.

*Типы обучающих программ*

* тренировочные и контролирующие;
* наставнические;
* имитационные и моделирующие;
* развивающие игры.

*Тренировочные программы* предназначены для закрепления умений и навыков.

*Наставнические программы* предлагают ученикам теоретический материал для изучения.

Существует и продолжает разрабатываться большое количество *инструментальных* программ такого вида. Общим их недостатком *является* высокая трудоемкость разработки, затруднения организационного и методического характера при использовании в реальном процессе обучения.

*Моделирующие программы* основаны на графических иллюстративных возможностях компьютера, с одной стороны, и вычислительных, с другой, и позволяют осуществлять *компьютерный* эксперимент. Такие программы предоставляют возможность наблюдать на экране дисплея некоторый процесс, влияя на его ход подачей команды с клавиатуры, меняющей значения параметров.

*Развивающие игры* предоставляют в распоряжение ученика некоторую воображаемую среду, существующий только в компьютере мир, набор каких-то возможностей и средств их реализации.

Наибольшее распространение получили обучающие программы первых двух типов в связи с их относительно невысокой сложностью, возможностью унификации при разработке многих блоков программ

Учебная мультимедиа и гипермедиа-технология представляет собой развитие технологии программированного обучения, хотя упор делается не на адаптивность обучения и его методическое обоснование, а на внешнюю иллюстративно-наглядную сторону. Современные графические и звуковые возможности компьютера обусловили появление средств гипер- и мультимедиа.

***Мультимедиа технология -*** представление информации в форме видеоизображения с применением мультипликации и звукового сопровождения.

***Гипермедиа технология -*** компьютерное представление данных различного типа, в котором автоматически поддерживаются смысловые связи между выделенными понятиями, объектами или разделами.

# Автоматизированные системы научных исследованиях

Автоматизированные системы научных исследований (АСНИ) - системы, в которых ряд процедур получения, анализа, передачи и накопления информации для повышения эффективности научных исследований формализован и выполняется автоматизированно. Задачи, которые можно решать с помощью АСНИ: 1) сократить сроки исследований, 2) повысить точность моделей и получить качественно новую информацию, 3) повысить эффективность использования оборудования, 4) сократить вспомогательный персонал изыскательских групп.

Применение автоматизированных систем научных исследований и комплексных испытаний образцов новой техники (АСНИ) наиболее эффективно в тех современных областях науки и техники, которые имеют дело с использованием больших объемов информации.

АСНИ отличаются от других типов автоматизированных систем характером информации, получаемой на выходе системы. Прежде всего - это обработанные или обобщенные экспериментальные данные, но главное - полученные на основе этих данных математических моделей исследуемых объектов, явлений или процессов. Адекватность и точность таких моделей обеспечивается всем комплексом методических, программных и других средств системы. В АСНИ могут использоваться также и готовые математические модели для изучения поведения тех или иных объектов и процессов, а также для уточнения самих этих моделей. АСНИ поэтому являются системами для получения, корректировки или исследования моделей, используемых затем в других типах автоматизированных систем для управления, прогнозирования или проектирования.

# Системы автоматизированного проектирования

Под САПР подразумеваются автоматизированные системы, которые призваны реализовывать ту или иную информационную технологию путем проектирования. На практике САПР представляют собой технические системы, которые позволяют таким образом автоматизировать, обеспечить функционирование процессов, которые составляют разработку проектов. Под САПР в зависимости от контекста может иметься в виду:

— программное обеспечение, применяемое в качестве основного элемента соответствующей инфраструктуры;

— совокупность технических и кадровых систем (в том числе и тех, что предполагают использование САПР в виде программного обеспечения), применяемых на предприятии с целью автоматизации процесса разработки проектов;

Таким образом, можно выделить широкую и более узкую трактовку термина, о котором идет речь.

Основной целью разработки САПР является повышение эффективности труда специалистов предприятия, которые решают различные производственные задачи, в том числе и те, которые связаны с инженерным проектированием.

К наиболее распространенным критериям классификации САПР относится отраслевое назначение. Выделяют следующие типы:

1. Автоматизированное проектирование инфраструктуры машиностроения;
2. САПР для электронного оборудования;
3. САПР в сфере строительства.

<http://computerologia.ru/sapr-eto-sistemy-avtomatizirovannogo-proektirovaniya/>

# Геоинформационные системы и технологии

Геоинформационная система (ГИС) - это многофункциональная информационная система, предназначенная для сбора, обработки, моделирования и анализа пространственных данных, их отображения и использования при решении расчетных задач, подготовке и принятии решений. Основное назначение ГИС заключается в формировании знаний о Земле, отдельных территориях, местности, а также своевременном доведении необходимых и достаточных пространственных данных до пользователей с целью достижения наибольшей эффективности их работы.  
Геоинформационные технологии (ГИТ) — это информационные технологии обработки географически организованной информации.  
Основной особенностью ГИС, определяющей ее преимущества в сравнении с другими АИС, является наличие геоинформационной основы, т.е. цифровых карт (ЦК), дающих необходимую информацию о земной поверхности. При этом ЦК должны обеспечивать:  
• точную привязку, систематизацию, отбор и интеграцию всей поступающей и хранимой информации (единое адресное пространство);  
• комплексность и наглядность информации для принятия решений;  
• возможность динамического моделирования процессов и явлений;  
• возможность автоматизированного решения задач, связанных с анализом особенностей территории;  
• возможность оперативного анализа ситуации в экстренных случаях.

В широком смысле ГИТ - это наборы данных и аналитические средства для работы с координатно-привязанной информацией. ГИТ - это не информационные технологии в географии, а информационные технологии обработки географически организованной информации.  
Существо ГИТ проявляется в ее способности связывать с картографическими (графическими) объектами некоторую описательную (атрибутивную) информацию (в первую очередь алфавитно-цифровую и иную графическую, звуковую и видеоинформацию). использование ГИТ - это инфор­мационно-поисковые, справочные системы.

ГИТ, так же как и любая другая технология, ориентирована на решение оп­ределенного круга задач. Поскольку области применения ГИС достаточно ши­роки (военное дело, картография, география, градостроительство, организация транспортных диспетчерских служб, и т.д.), то ввиду специфики проблем, ре­шаемых в каждой из них, и особенностей, связанных с конкретным классом ре­шаемых задач и с требованиями, предъявляемыми к исходным и выходным данным, точности, техническим средствам и прочее, говорить о какой-то еди­ной ГИС-технологии достаточно проблематично.

# Информационная система управления

Автоматизированная система управления или АСУ -- комплекс аппаратных и программных средств, предназначенный для управления различными процессами в рамках технологического процесса, производства, предприятия. АСУ применяются в различных отраслях промышленности, энергетике, транспорте и т. п. Термин автоматизированная, в отличие от термина автоматическая подчеркивает сохранение за человеком-оператором некоторых функций, либо наиболее общего, целеполагающего характера, либо не поддающихся автоматизации.

В зависимости от роли человека в процессе управления, форм связи и функционирования звена “человек--машина”, распределения информационных и управляющих функций между оператором и ЭВМ, между ЭВМ и средствами контроля и управления все системы можно разделить на два класса.

1. Информационные системы, обеспечивающие сбор и выдачу в удобном для обозрения виде измерительную информацию о ходе технологического или производственного процесса.

Цель таких систем -- получение оператором информации с высокой достоверностью для эффективного принятия решений.

Различают два вида информационных систем: информационно-справочные (пассивные), которые поставляют информацию оператору после его связи с системой по соответствующему запросу, информационно-советующие (активные), которые сами выдают абоненту предназначенную для него информацию периодически или через определенные промежутки времени.

2. Управляющие системы, которые обеспечивают наряду со сбором информации выдачу непосредственно, команд исполнителям или исполнительным механизмам. Управляющие системы работают обычно в реальном масштабе времени, т.е. в темпе технологических или производственных операций. В управляющих системах важнейшая роль принадлежит машине, а человек контролирует и решает наиболее сложные вопросы, которые по тем или иным причинам не могут решить вычислительные средства системы.

Принято рассматривать каждую АСУ одновременно в двух аспектах: с точки зрения ее функций (что и как она делает) и с точки зрения ее схемы, т. е. с помощью каких средств и методов эти функции реализуются. Соответственно АСУ подразделяют на две группы подсистем -- функциональные и обеспечивающие.

# Искусственный интеллект

**Искусственный интеллект** - это наука создания интеллектуальных технологий и компьютерных программ.

Искусственный интеллект тесно связан с задачей понять человеческий интеллект с помощью компьютерных технологий. На данный момент нельзя точно сказать, какие вычислительные методы можно называть интеллектуальными. Одни механизмы интеллекта открыты для понимания, остальные нет. На данный момент в программах используются методы, не встречающиеся у людей.

Искусственный интеллект имеет научное направление, которое изучает решение задач интеллектуальной деятельности человека. Искусственный интеллект направлен на выполнение творческих задач в области, знания о которой хранится в интеллектуальной системе программы - **базе знаний**.

С этими знаниями работает механизм программы - **решатель задач**. Затем человек получает представление о результате работы программы через интеллектуальный интерфейс. Результатом программы искусственного интеллекта, является воссоздание интеллектуального рассуждения или разумного действия.

Одним из главных свойств искусственного интеллекта является способность самообучаться. В первую очередь, это **эвристическое обучение** -  непрерывное обучение программы, формирование процесса обучения и собственных целей, анализ и осознание своего обучения.

Научное направление изучающее искусственный интеллект начало зарождаться еще давно:

* философы думали о познании внутреннего мира человека
* психологи изучали мышление человека
* математики занимались расчетами

Вскоре, были созданы первые компьютеры, которые позволили выполнять вычисления обгоняя по скорости человека.

Существует множество разных подходов к изучению и пониманию искусственного интеллекта.

* Символьный подход
* Логический подход
* Агентно-ориентированный подход
* Гибридный подход
* Моделирование рассуждений
* Обработка естественного языка
* Представление и использование знаний
* Машинное обучение
* Биологическое моделирование ИИ
* Робототехника

# Информационная технология обработки данных

Информационная технология обработки данных предназначена для решения хорошо структурированных задач, по которым имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки.

Эта технология применяется на уровне исполнительской деятельности персонала невысокой квалификации в целях автоматизации некоторых рутинных постоянно повторяющихся операций управленческого труда. Поэтому внедрение информационных технологий и систем на этом уровне существенно повысит производительность труда персонала, освободит его от рутинных операций, возможно, даже приведет к необходимости сокращения численности работников.

На уровне операционной деятельности решаются следующие задачи:

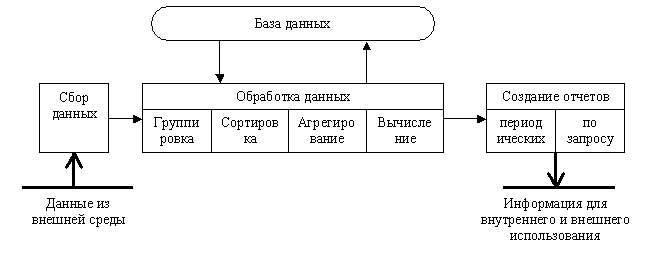
\* обработка данных об операциях, производимых фирмой;

\* создание периодических контрольных отчетов о состоянии дел в фирме;

\* получение ответов на всевозможные текущие запросы и оформление их в виде бумажных документов или отчетов.

Основные компоненты

Представим основные компоненты информационной технологии обработки данных (рис. 2.2.) и приведем их характеристики.



Сбор данных

Обработка данных.

Хранение данных. Многие данные на уровне операционной деятельности необходимо сохранять для последующего использования либо здесь же, либо на другом уровне. Для их хранения создаются базы данных.

Создание отчетов (документов). В информационной технологии обработки данных необходимо создавать документы для руководства и работников фирмы, а также для внешних партнеров. При этом документы могут создаваться как по запросу или в связи с проведенной фирмой операцией, так и периодически в конце каждого месяца, квартала или года.

Типичной информационной системой оперативного уровня является популярная программа "1C: Бухгалтерия" фирмы "1C" для Windows. Эта программа предоставляет широкие возможности манипулирования бухгалтерскими данными.

# Информационная технология управления

Цель информационной технологии управления - удовлетворение информационных потребностей всех без исключения сотрудников фирмы, имеющих дело с принятием решений. Она может быть полезна на любом уровне управления.

Для принятия решений на уровне управленческого контроля, информация должна быть представлена в агрегированном виде так, чтобы просматривались тенденции изменения данных, причины возникших отклонений и возможные решения.

Эта технология ориентирована на работу в среде информационной системы управления и используется при худшей структурированности решаемых задач, если их сравнивать с задачами, решаемыми с помощью информационной технологии обработки данных.

ИС управления идеально подходят для удовлетворения сходных информационных потребностей работников различных функциональных подсистем (подразделений) или уровней управления фирмой. Поставляемая ими информация содержит сведения о прошлом, настоящем и вероятном будущем фирмы. Эта информация имеет вид регулярных или специальных

Основные компоненты



Основные компоненты информационной технологии управления показаны на рис. 2.3.

Входная информация поступает из систем операционного уровня. Выходная информация формируется в виде управленческих отчетов в удобном для принятия решения виде.

Содержимое базы данных при помощи соответствующего программного обеспечения преобразуется в периодические и специальные отчеты, поступающие к специалистам, участвующим в принятии решений в организации. База данных, используемая для получения указанной информации, должна состоять из двух элементов:

1) данных, накапливаемых на основе оценки операций, проводимых фирмой;

2) планов, стандартов, бюджетов и других нормативных документов, определяющих планируемое состояние объекта управления (подразделения фирмы).

# Автоматизация офисной деятельности

Автоматизация офиса призвана не заменить существующую традиционную систему коммуникации персонала, а лишь дополнить ее. Совместное использование этих систем обеспечивает рациональную автоматизацию управленческого труда и наилучшее обеспечение управленцев информацией.

Автоматизированный офис привлекателен для менеджеров всех уровней управления в фирме не только потому, что поддерживает внутрифирменную связь персонала, но также потому, что предоставляет им новые средства коммуникации с внешним окружением.

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

Менеджеры благодаря этому быстрее делают решения.

В настоящее время известно несколько десятков программных продуктов для компьютеров и некомпьютерных технических средств, обеспечивающих технологию автоматизации офиса: текстовый процессор, табличный процессор, электронная почта, электронный календарь, аудиопочта, компьютерные и телеконференции, видеотекст, хранение изображений, а также специализированные программы управленческой деятельности: ведения документов, контроля за исполнением приказов и т.д.

Автоматизацию офиса дополняют некомпьютерные средства: аудио и видеоконференции, факсимильная связь, ксерокс и другие средства оргтехники.

Основные компоненты

База данных является обязательным компонентом любой информационной технологии.

Текстовый процессор - это вид прикладного программного обеспечения, предназначенный для создания и обработки текстовых документов.

Электронная почта (E-mail), основываясь на сетевом использовании компьютеров, дает возможность пользователю получать, хранить и отправлять сообщения своим партнерам по сети.

Аудиопочта - это почта для передачи сообщений голосом.

Табличный процессор так же, как и текстовый процессор, является базовой составляющей информационной культуры любого сотрудника и автоматизированной офисной технологии.

Электронный календарь предоставляет еще одну возможность использовать сетевой вариант компьютера для хранения и манипулирования рабочим расписанием управленцев и других работников организации.

Компьютерные конференции используют компьютерные сети для обмена информацией между участниками группы, решающей определенную проблему.

Наиболее популярным набором программ для офисной автоматизации является Microsoft Office. Продукты Microsoft Office тесно интегрированы между собой, они имеют более 50% общего программного кода. Это является основой однотипной работы со всеми приложениями.

# Информационная технология поддержки принятия решений

Главной особенностью информационной технологии поддержки принятия решений является качественно новый метод организации взаимодействия человека и компьютера. Выработка решения, что является основной целью этой технологии, происходит в результате итерационного процесса , в котором участвуют:

\* система поддержки принятия решений (СППР) в роли вычислительного звена и объекта управления;

\* лица, принимающего решение, оценивающего полученный результат вычислений на компьютере.

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

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

\* ориентация на решение плохо структурированных (формализованных) задач;

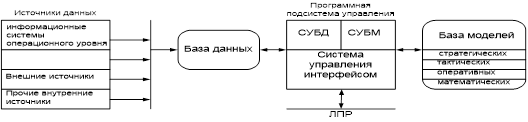
\* сочетание традиционных методов доступа и обработки компьютерных данных с возможностями математических моделей и методами решения задач на их основе;

\* направленность на непрофессионального пользователя компьютера;

\* высокая адаптивность

Информационная технология поддержки принятия решений может использоваться на любом уровне управления. Кроме того, решения, принимаемые на различных уровнях управления, часто должны координироваться. Поэтому важной функцией и систем, и технологий является координация лиц, принимающих решения, как на разных уровнях управления, так и на одном уровне.

Основные компоненты



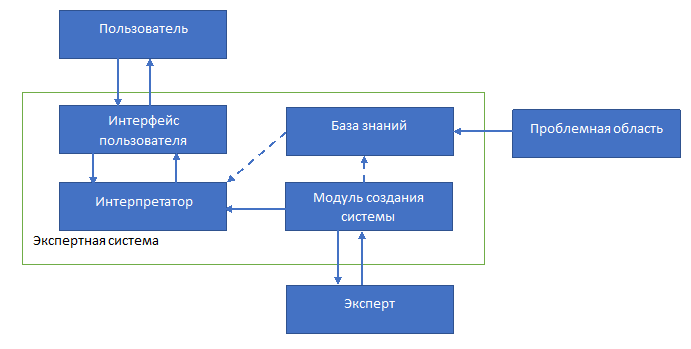
В состав системы поддержки принятия решений входят три главных компонента: база данных, база моделей и программная подсистема, которая состоит из системы управления базой данных (СУБД), системы управления базой моделей (СУБМ) и системы управления интерфейсом между пользователем и компьютером.

База данных играет в информационной технологии поддержки принятия решений важную роль. Данные могут использоваться непосредственно пользователем для расчетов при помощи математических моделей.

База моделей. Целью создания моделей являются описание и оптимизация некоторого объекта или процесса. Использование моделей обеспечивает проведение анализа в системах поддержки принятия решений. Модели, базируясь на математической интерпретации проблемы, при помощи определенных алгоритмов способствуют нахождению информации, полезной для принятия правильных решений.

# Экспертная система

Экспертная система — это программное средство, использующее знания экспертов, для высокоэффективного решения задач в интересующей пользователя предметной области. Она называется системой, а не просто программой, так как содержит базу знаний, решатель проблемы и компоненту поддержки. Последняя из них помогает пользователю взаимодействовать с основной программой. Экспертная система дает возможность специалисту получать консультацию экспертов по любым проблемам, на основе которых этими системами накоплены знания.



# Модель информационных процессов

При производстве информационного продукта исходный информационный ресурс в соответствии с поставленной задачей подвергается в определенной последовательности различным преобразованиям. Динамика этих преобразований отображается в протекающих при этом информационных процессах. Таким образом, информационный процесс - это процесс преобразования информации. В результате информация может изменить и содержание, и форму представления.

Управляющие воздействия формируются на основе накопленной и функционирующей в системе управления информации, а также поступающих по каналам прямой и обратной связи сведений из внешней среды.

Таким образом, важнейшая функция любой системы управления - получение информации, выполнение процедур по ее обработке с помощью заданных алгоритмов и программ, формирование на основе полученных сведений управленческих решений, определяющих дальнейшее поведение системы.

Поскольку информация фиксируется и передается на материальных носителях, необходимы действия человека и работа технических средств по восприятию, сбору информации, ее записи, передаче, преобразованию, обработке, хранению, поиску и выдаче. Эти действия обеспечивают нормальное протекание информационного процесса и входят в технологию управления. Они реализуются технологическими процессами обработки данных с использованием электронных вычислительных машин и других технических средств.

Фазы преобразования информации в информационной технологии достаточно многочисленны. Однако если провести структуризацию технологии, обобщенная схема технологического процесса обработки информации может быть представлена схемой, показанной на рис. 3.1.

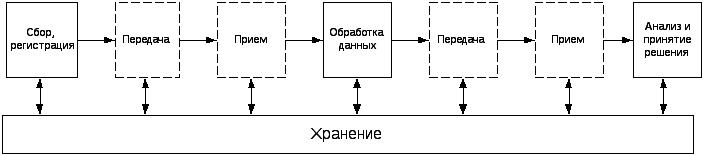


Рис.3.1 – Обобщенная схема технологического процесса обработки информации

При обработке данных формируются четыре основных информационных процесса: сбор и регистрация, обмен, обработка, накопление и хранение информации. Рассмотрим их модели.

# Сбор и регистрация информации

Сбор и регистрация информации происходят по-разному и в различных объектах.

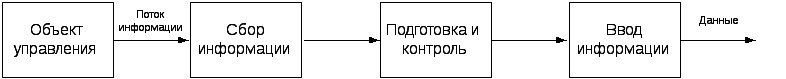


Рис.3.2 - Процесс преобразования информации в данные

*Сбор информации* состоит в том, что поток осведомляющей информации, поступающей от объекта управления, воспринимается человеком и переводится в документальную форму (записывается на бумажный носитель информации).

Передача осуществляется, как правило, традиционно, с помощью курьера, телефона.

Собранная информация для ввода должна быть предварительно подготовлена, поскольку модель предметной области, заложенная в компьютер, накладывает свои ограничения на состав и организацию вводимой информации.

*Контроль* подготовленной и вводимой информации направлен на предупреждение, выявление и устранение ошибок, которые неизбежны в первую очередь из-за так называемого "человеческого фактора".

Вообще процедуры контроля полноты и достоверности информации и данных используются при реализации информационных процессов повсеместно и могут быть подразделены *на визуальные, логические и арифметические.*

*Визуальный метод* широко используется на этапе сбора и подготовки начальной информации и является ручным. При визуальном методе производится зрительный просмотр документа в целях проверки полноты, актуальности, подписей ответственных лиц, юридической законности и т.д.

*Логический метод* контроля предполагает сопоставление фактических данных с нормативными или с данными предыдущих периодов обработки, проверку логической непротиворечивости функционально-зависимых показателей и их групп и т.д.

*Арифметический метод* контроля включает подсчет контрольных сумм по строкам и столбцам документов, имеющих табличную форму, контроль по формулам, признакам делимости или четности, балансовые методы, повторный ввод и т.п.

*Ввод* первоначальной информации при создании информационной технологии в организационно-экономической системе в конечном итоге является ручным - пользователь ЭВМ "набирает" данные (алфавитно-цифровые) на клавиатуре, визуально контролируя правильность вводимых символов по отображению на экране дисплея.

Таким образом, после сбора, подготовки, контроля и ввода исходная информация (документы, модели, программы) превращается в данные, представленные машинными кодами, которые хранятся на машинных носителях и обрабатываются техническими средствами информационной технологии.

# Передача информации

Информационный процесс обмена предполагает обмен данными между процессами информационной технологии. Из рисунка 1 видно, что процесс обмена связан взаимными потоками данных со всеми информационными процессами на уровне переработки данных.

*Передача информации осуществляется различными способами:* с помощью курьера, пересылка по почте, доставка транспортными средствами, дистанционная передача по каналам связи, с помощью других средств коммуникаций.

*При дистанционной передаче по каналам* связи (рис. 3.3) можно выделить два основных типа процедур. Это процедуры передачи данных по каналам связи и сетевые процедуры, позволяющие осуществить организацию вычислительной сети. Процедуры передачи данных реализуются с помощью операции кодирования - декодирования, модуляции - демодуляции, согласования и усиления сигналов. Процедуры организации сети включают в себя в качестве основных операции по коммутации и маршрутизации потоков данных (трафика) в вычислительной сети. Процесс обмена позволяет, с одной стороны, передавать данные между источником и получателем информации, а с другой - объединять информацию многих ее источников.

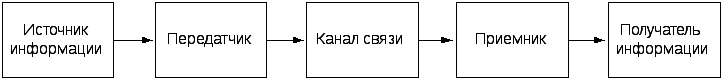


Рис. 3.3. Передача данных по каналу связи

Дистанционная передача по каналам связи сокращает время передачи данных, однако, для ее осуществления необходимы специальные технические средства, что удорожает процесс передачи. Предпочтительным является использование технических средств сбора и регистрации, которые, собирая автоматически информацию с установленных на рабочих местах датчиков, передают ее в ЭВМ для последующей обработки, что повышает ее достоверность и снижает трудоемкость.

Дистанционно может передаваться как первичная информация с мест ее возникновения, так и результатная в обратном направлении. В этом случае результатная информация фиксируется различными устройствами: дисплеями, табло, печатающими устройствами. Поступление информации по каналам связи в центр обработки в основном осуществляется двумя способами: на машинном носителе или непосредственно вводом в ЭВМ при помощи специальных программных и аппаратных средств.

Модель обмена данными включает в себя формальное описание процедур, выполняемых в вычислительной сети: передачи, маршрутизации, коммутации. Именно эти процедуры и составляют информационный процесс обмена. Для качественной работы сети необходимы формальные соглашения между ее пользователями, что реализуется в виде протоколов сетевого обмена. В свою очередь, передача данных основывается на моделях кодирования, модуляции, каналов связи.

# Обработка

На различных этапах информационного цикла данные преобразовываются из одного вида в другой с помощью различных методов. Общая схема процесса обработки информации выглядит следующим образом (рис. 1.15).

Схема процесса обработки информации

Рис.1.15. **Схема процесса обработки информации**

В процессе обработки информации решается некоторая информационная задача, для которой должны быть определены исходная (некоторый набор исходных данных) и итоговая (требуемые результаты) информация. Переход от исходных данных к результату и есть процесс обработки. Тот объект или субъект, который осуществляет обработку, называется исполнителем обработки. Это может быть человек или техническое устройство, в том числе компьютер.

Для успешного выполнения обработки информации исполнителю должен быть известен способ обработки, т. е. последовательность действий, которую нужно выполнить, чтобы достичь нужного результата. Описание такой последовательности действий в информатике принято называть алгоритмом обработки.

Можно выделить два типа обработки информации:

1. Обработка, связанная с получением новой информации, нового содержания знаний. К ней относится решение различных задач путем применения логических рассуждений.

2. Обработка, связанная с изменением формы, но не изменяющая содержания, например, перевод текста с одного языка на другой.

Обработка данных включает в себя множество разных операций, представляющих собой комплекс совершаемых технологических действий, в результате которых информация преобразуется. Основными операциями являются:

- формализация (приведение данных, поступающих из разных источников, к единой форме);

- фильтрация (устранение лишних данных, которые не нужны для принятия решений);

- сортировка (приведение в порядок данных по заданным признакам с целью удобства использования);

- архивация (сохранение данных в удобной и доступной форме);

- защита (комплекс мер, направленных на предотвращение потерь при воспроизведении и модификации данных);

- преобразование (преобразование данных из одной формы в другую или из одной структуры в другую или изменение типа носителя).

Обработка информации - это получение одних информационных объектов из других информационных объектов путем выполнения некоторых алгоритмов.

Обработка является одним из основных процессов, выполняемых над информацией, и главным средством увеличения объема и разнообразия информации.

# Хранение

Хранение и накопление информации вызвано многократным ее использованием, применением условно-постоянной, справочной и других видов информации, необходимостью комплектации первичных данных до их обработки.

Назначение технологического процесса накопления данных состоит в создании, хранении и поддержании в актуальном состоянии информационного фонда, необходимого для выполнения функциональных задач системы управления. Хранение и накопление информации осуществляется в информационных базах в виде информационных массивов, где данные располагаются по установленному в процессе проектирования порядку.

*Процесс накопления данных* состоит из ряда основных процедур, таких, как выбор хранимых данных, хранение данных, их актуализация и извлечение.

Информационный фонд систем управления должен формироваться на основе принципов необходимой полноты и минимальной избыточности хранимой информации. Эти принципы реализуются процедурой выбора хранимых данных, в процессе выполнения которой производится анализ циркулирующих в системе данных и на основе их группировки на входные, промежуточные и выходные определяется состав хранимых данных. Входные данные - это данные, получаемые из первичной информации и создающие информационный образ предметной области. Они подлежат хранению в первую очередь. Промежуточные данные - это данные, формирующиеся из других данных при алгоритмических преобразованиях. Как правило, они не хранятся, но накладывают ограничения на емкость оперативной памяти компьютера. Выходные данные являются результатом обработки первичных (входных) данных по соответствующей модели, они входят е состав управляющего информационного потока своего уровня и подлежат хранению в определенном временном интервале. Вообще, данные имеют свой жизненный цикл существования, который фактически и отображается в процедурах процесса накопления.

*Процедура хранения* состоит в том, чтобы сформировать и поддерживать структуру хранения данных в памяти ЭВМ. Современные структуры хранения данных должны быть независимы от программ, использующих эти данные, и реализовывать вышеуказанные принципы (полнота и минимальная избыточность). Такие структуры получили название баз данных. Осуществление процедур создания структуры хранения (базы данных), актуализации, извлечения и удаления данных производится с помощью специальных программ, называемых системами управления базами данных.

В процессе накопления данных важной *процедурой является их актуализация.* Под актуализацией понимается поддержание хранимых данных на уровне, соответствующем информационным потребностям решаемых задач в системе, где организована информационная технология. Актуализация данных осуществляется с помощью операций добавления новых данных к уже хранимым, корректировки (изменения значений или элементов структур) данных и их уничтожения, если данные устарели и уже не могут быть использованы при решении функциональных задач системы.

*Процедура извлечения данных из* базы необходима для пересылки требуемых данных либо для преобразования, либо для отображения, либо для передачи по вычислительной сети.

При выполнении процедур актуализации и извлечения обязательно выполняются операции поиска данных по заданным признакам и их сортировки, состоящие в изменении порядка расположения данных при хранении или извлечении.

# Основные понятия теории моделирования системы

Общие сведения о компьютерном математическом моделировании

Модель - материальный объект, система математических зависимостей или программа, имитирующая структуру или функционирование исследуемого объекта.

Моделирование - представление различных характеристик поведения физической или абстрактной системы с помощью другой системы.

Математическое моделирование - метод исследования процессов и явлений на их математических моделях.

Изучение компьютерного математического моделирования открывает широкие возможности для осознания связи информатики с математикой и другими науками - естественными и социальными. Компьютерное математическое моделирование в разных своих проявлениях использует практически весь аппарат современной математики.

Математическое моделирование не всегда требует компьютерной поддержки. Каждый специалист, профессионально занимающийся математическим моделированием, делает все возможное для аналитического исследования модели. Аналитические решения (т.е. представленные формулами, выражающими результаты исследования через исходные данные) обычно удобнее и информативнее численных. Возможности аналитических методов решения сложных математических задач, однако, очень ограничены и, как правило, эти методы гораздо сложнее численных. В компьютерном моделировании доминируют численные методы, реализуемые на компьютерах. Однако понятия "аналитическое решение" и "компьютерное решение" отнюдь не противостоят друг другу, так как:

а) все чаще компьютеры при математическом моделировании используются не только для численных расчетов, но и для аналитических преобразований:

б) результат аналитического исследования математической модели часто выражен столь сложной формулой, что при взгляде на нее не складывается восприятия описываемого ей процесса. Эту формулу нужно представить графически, проиллюстрировать в динамике, иногда даже озвучить, т.е. проделать то, что называется "визуализацией абстракций". При этом компьютер - незаменимое техническое средство.

6.1.2 Классификация математических моделей

К классификации математических моделей можно подходить по-разному, положив в основу классификации различные принципы.

1) Классификация моделей по отраслям наук (математические модели в физике, биологии, социологии и т.д.);

2) Классификация моделей по применяемому математическому аппарату (модели, основанные на применении обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, стохастических методов, дискретных алгебраических преобразований и т.д.);

3) Классификация моделей с точки зрения целей моделирования.

\* дескриптивные (описательные) модели;

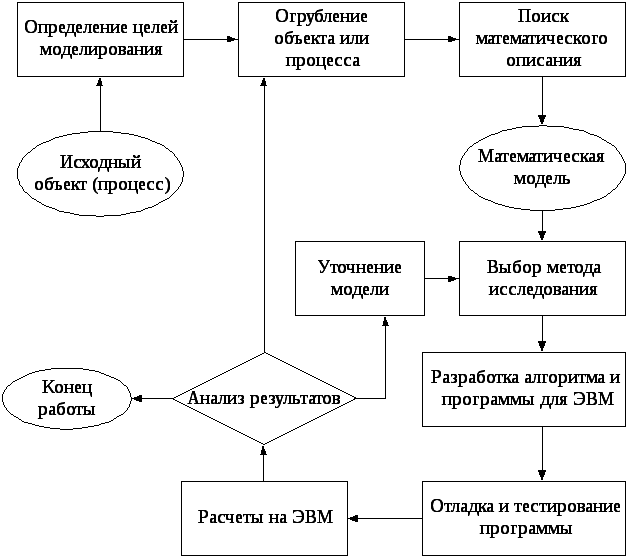
\* оптимизационные модели;

\* многокритериальные модели;

\* игровые модели;

\* имитационные модели.

# Этапы компьютерного математического моделирования



**Этапы компьютерного математического моделирования** изображены на рисунке. Первый этап — *определение целей моделирования.* Эти цели могут быть различными:

1. модель нужна для того, чтобы понять, как устроен конкретный объект, какова его структура, основные свойства, законы развития и взаимодействия с окружающим миром (понимание);
2. модель нужна для того, чтобы научиться управлять объектом (или процессом)  и определить наилучшие способы управления при заданных целях и критериях (управление);
3. модель нужна для того, чтобы прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействия на объект (прогнозирование).

Второй этап: определение входных и выходных параметров модели; разделение входных параметров по степени важности влияния их изменений на выходные. Такой процесс называется ранжированием, или разделением по рангам (см. *"Формализация и моделирование"*).

Третий этап: построение математической модели. На этом этапе происходит переход от абстрактной формулировки модели к формулировке, имеющей конкретное математическое представление. Математическая модель — это уравнения, системы уравнений, системы неравенств, дифференциальные уравнения или системы таких уравнений и пр.

Четвертый этап: выбор метода исследования математической модели. Чаще всего здесь используются численные методы, которые хорошо поддаются программированию. Как правило, для решения одной и той же задачи подходит несколько методов, различающихся точностью, устойчивостью и т.д. От верного выбора метода часто зависит успех всего процесса моделирования.

Пятый этап: разработка алгоритма, составление и отладка программы для ЭВМ — трудно формализуемый процесс.

Шестой этап: тестирование программы. Работа программы проверяется на тестовой задаче с заранее известным ответом. Это — лишь начало процедуры тестирования, которую трудно описать формально исчерпывающим образом. Обычно тестирование заканчивается тогда, когда пользователь по своим профессиональным признакам сочтет программу верной.

Седьмой этап: собственно вычислительный эксперимент, в процессе которого выясняется, соответствует ли модель реальному объекту (процессу). Модель достаточно адекватна реальному процессу, если некоторые характеристики процесса, полученные на ЭВМ, совпадают с экспериментально полученными характеристиками с заданной степенью точности. В случае несоответствия модели реальному процессу возвращаемся к одному из предыдущих этапов.

# Системный подход компьютерного моделирования

Для построения моделей объектов используют системный подход, который представляет собой методологию решения сложных задач. В основе этой методологии лежит рассмотрение объекта как системы, которая функционирует в некоторой среде. Системный подход позволяет раскрыть целостность объекта, выявить и изучить его внутреннюю структуру, а также связи с внешней средой. При этом объект является частью реального мира, которую выделяют и исследуют в связи с решаемой задачей построения модели. Кроме того, при использовании системного подхода предполагается последовательный переход от общего к частному, в основе которого лежит рассмотрение цели проектирования, а объект рассматривается во взаимосвязи с окружающей средой. Сложный объект может разделяться на подсистемы, которые представляют собой части объекта и удовлетворяют таким требованиям: подсистема – функционально независимая часть объекта, которая связана с другими подсистемами и обменивается с ними информацией и энергией; каждая подсистема может иметь функции или свойства, которые не совпадают со свойствами всей системы; каждая из подсистем может делиться до уровня элементов. Под элементом здесь понимают подсистему нижнего уровня, которую далее делить не представляется целесообразным с позиции решаемой задачи. Таким образом, система представляется как объект, состоящий из набора подсистем, элементов и связей для его создания, исследования или усовершенствования. При этом укрупнение представления системы, которое включает основные подсистемы и связи между ними, называется макроструктурой, а детальное рассмотрение внутреннего строения системы до уровня элементов – микроструктурой. С понятием системы обычно связано понятие надсистемы – системы более высокого уровня, в состав которой входит рассматриваемый объект, причем функция любой системы может быть определена только через надсистему. Также немаловажно понятие среды – совокупности объектов внешнего мира, которые существенно влияют на эффективность функционирования системы, но не входят в состав системы и ее надсистемы. В системном подходе к построению моделей используют понятие инфраструктуры, которая описывает взаимосвязь системы с ее окружением (средой). Выделение, описание и исследование свойств объекта, которые являются существенными для конкретной задачи, называется стратификацией объекта. При системном подходе в моделировании важно определение структуры системы, которая определяется как совокупность связей между элементами системы, которые отражают их взаимодействие. Различают структурный и функциональный подход к моделированию. При структурном подходе определяется состав выделенных элементов системы и связи между ними. Совокупность элементов и связей составляет структуру системы. Реже применяется функциональное описание, при котором рассматриваются отдельные функции – алгоритмы поведения системы.

# Анализ и синтез в проектировании систем

*Проектирование* – это процесс создания описания, необходимого для построения в заданных условиях еще не существующего объекта, на основе его первоначального описания.

Выделяют две следующие основные задачи при проектировании в САПР.

1. Задача анализа - связана с определением функции объекта или системы по заданному описанию и оценкой проектных полученных решений.

2. Задача синтеза - связана с созданием самого объекта (проекта) и проектных документов.

Синтез бывает:

• структурным (его цель - получение структурной схемы объекта, формирование сведений о составе элементов и способах их соединения между собой);

• параметрическим (его цель - определение численных значений параметров элементов).

Синтез называется оптимизацией, если определяются наилучшие в заданном смысле структуры (структурная оптимизация) и параметры (параметрическая оптимизация).

Проведение оптимизации требует наличия спектра критериев оптимизации, которые бывают:

• частными

• аддитивными

• мультипликативными

• минимаксными.

Для проведения задач анализа применяются специальные языки моделирования, позволяющие создавать математические модели объектов, определять схемы их функционирования и производить оценку принимаемых решений.

# Характеристика моделей систем

Из лекции

# Виды моделирования (Детерминированное, стохастическое, статические, динамические, дискретное, непрерывное, дискретно-непрерывное)

Модель является детерминированной, если каждому входному набору параметров соответствует вполне определенный и однозначно определяемый набор выходных параметров.

Модель стохастическая, если некоторые из входных или выходных параметров имеют стохастическое природу (т.е. предполагают собой случайные величины).

Модель дискретная если она описывает поведение системы только в дискретные моменты времени.

Модель непрерывна если она описывает поведение системы для всех моментов промежутка времени.

Модель дискретно-непрерывная, если есть участки дискретно определяемые, и есть участки непрерывные

# Виды моделей (мысленное)

Наглядное делится еще на Гипотетическое, Аналоговое, макетированное.

1. При наглядном моделировании на базе представления человека о реальном объекте создаются различные наглядные модели отображающие явления и процессы, протекающие в объекте.

Пример. ()

* Наглядное гипотетическое в основу гипотетического моделирования закладывается некоторая гипотеза о закономерности протекания процесса в реальном объекте.
* Наглядное аналоговое моделирование основывается на применения мналогий различных уровней.(аналог Пензы-Карта Пензы)
* Наглядное макетированное может применяться в случае когда протекающие в реальном времени объекте процессы не подаются физическому моделированию. ( макет солнечной системы)

1. Символическое моделирование представляет собой искусственный процесс создания логического объекта, который замещает реальный и выражает основные свойства с помощью определенной системы символов.

Пример. («круг»)

* Символическое знаковое моделирование основывается на определенной системе знаков отображающих набор понятий (Дорожные знаки)
* Символическое языковое моделирование основывается на системе символов из специального словаря, отчищенного от неоднозначности(алфавит)

1. Математическое моделирование-это процесс установления соответствия данному реальному объекту некоторой математической модели и исследований этой модели для получения характеристик объекта.

* Математическое аналитическое при аналитическом моделировании свойства процесса описываются в виде функциональных соотношений или логических условий, которые решаются, либо в общем виде, либо при конкретных начальных данных(Система уравнений)
* Имитированное моделированное-это метод исследований при котором изучаемая система заменяется моделью с достаточной точностью описывающие реальную систему с которой приводятся эксперименты целью получения информации об этой системы(полет шарика)
* Математическое комбинированное моделирование-это где при анализе и синтезе систем объединяются преимущества имитационного и аналитического.

S= PR2

# Виды моделей(реальное)

Натурное. При натурном моделировании исследования проводят на реальном объекте с последующей обработкой результатов эксперимента на основе теории подобия.

* Научный эксперимент характеризуется использованием средств автоматизации, проведение активного эксперимента и обработки информации.
* Комплексное испытание. При комплексных испытаниях в следствии повторений испытания изделий выявления общие закономерности о надежности этих изделий о характеристиках качества и т.д.
* Производственный эксперимент связан с обобщением опыта накопленного в ходе производственного процесса

При физическом моделировании исследования проводят на установках, которые сохраняют природу явлений и обладают физ. Подобием.

# Структурный подход

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. При разработке системы «снизу-вверх» от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

Все наиболее распространенные методологии структурного подхода базируются на ряде общих принципов. В качестве двух базовых принципов используются следующие:

* + - принцип «разделяй и властвуй» – принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;
    - принцип иерархического упорядочивания – принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

Выделение двух базовых принципов не означает, что остальные принципы являются второстепенными, поскольку игнорирование любого из них может привести к непредсказуемым последствиям (в том числе и к провалу всего проекта). Основными из этих принципов являются следующие:

* + - *принцип абстрагирования* заключается в выделении существенных аспектов системы и отвлечении от несущественных;
    - *принцип формализации* заключается в необходимости строгого методического подхода к решению проблемы;
    - *принцип непротиворечивости* заключается в обоснованности и согласованности элементов;
    - *принцип структурирования данных* заключается в том, что данные должны быть структурированы и иерархически организованы.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой, и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых являются следующие:

* + - SADT (Structured Analysis and Design Technique) модели и соответствующие функциональные диаграммы;
    - DFD (Data Flow Diagrams) диаграммы потоков данных;
    - ERD (Entity-Relationship Diagrams) диаграммы «сущность-связь».

Перечисленные модели в совокупности дают полное описание ИС независимо от того, является ли она существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

# Case-технология

CASE-технология представляет собой совокупность методов анализа, проектирования и сопровождения сложных информационных систем, поддерживаемая комплексом взаимосвязанных средств автоматизации.

CASE - это инструментарий, позволяющий автоматизировать весь процесс проектирования экономической информационной системы. В большинстве современных CASE-технологий применяется методология структурного анализа и проектирования, основанная на наглядных диаграммах. Для описания проектируемой системы используются графы, диаграммы, таблицы, схемы, что обеспечивает наглядное описание проектируемой системы, которое начинается с ее общего обзора, а затем детализируется, приобретая иерархическую структуру со все большим числом уровней.

CASE-технология включает в себя следующие основные этапы:

· анализ требований;

· проектирование;

· программирование;

· тестирование и отладка;

· эксплуатация и сопровождение.

На этапе анализа требований уточняются требования заказчика, которые формализуются и документируются. В составе требований указываются условия, при которых предполагается эксплуатировать, описываются выполняемые системой функции и указываются ограничения в процессе разработки На этом этапе определяются:

· архитектура системы, ее функции, внешние условия, распределение функций между человеком и системой;

· требования к программным и информационным компонентам, требования к базе данных.

На этапе проектирования разрабатывается архитектура программного обеспечения и ЭИС, осуществляется согласование функций и технических требований к компонентам системы, а также детальное проектирование.

Другие этапы учитывают специфические особенности разработки и эксплуатации программного обеспечения. Инструментальными средствами CASE-технологии служат специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования. Основными положениями CASE-технологии являются:

· декомпозиция всей системы на некоторое множество компонентов;

· иерархия системы;

· представление всей информации в виде графических нотаций;

· диаграмма потока данных;

· диаграмма функций.

# Классификация CASE-технологий

В функции CASE входят средства анализа, проектирования и программирования программных средств, проектирования интерфейсов, документирования и производства структурированного кода на каком-либо языке программирования.

CASE-инструменты классифицируются по типам и категориям.

Классификация по типам отражает функциональную ориентацию средств на те или иные процессы жизненного цикла разработки программного обеспечения, и, в основном, совпадают с компонентным составом крупных интегрированных CASE-систем, и включает следующие типы:

* средства анализа — предназначены для построения и анализа модели предметной области;
* средства проектирования баз данных;
* средства разработки приложений;
* средства реинжиниринга процессов;
* средства планирования и управления проектом;
* средства тестирования;
* средства документирования.

Классификация по категориям определяет степень интегрированности по выполняемым функциям и включают — отдельные локальные средства, решающие небольшие автономные задачи, набор частично интегрированных средств, охватывающих большинство этапов жизненного цикла и полностью интегрированных средств, охватывающий весь жизненный цикл информационной системы и связанных общим репозиторием.

Типичными CASE-инструментами являются:

* инструменты управления конфигурацией;
* инструменты моделирования данных;
* инструменты анализа и проектирования;
* инструменты преобразования моделей;
* инструменты редактирования программного кода;
* инструменты рефакторинга кода;
* генераторы кода;
* инструменты для построения UML-диаграмм.

# Основные понятия БД

База данных– это совокупность структурированных и взаимосвязанных данных, организованная по определенным правилам, которые предусматривают общие принципы описания, хранения и обработки данных.

Существуют 4 основные модели данных – списки (плоские таблицы), реляционные базы данных, иерархические и сетевые структуры.

В настоящее время наибольшее распространение при разработке БД получили реляционные модели данных. Реляционная модель данных является совокупностью простейших двумерных таблиц – отношений т.е. простейшая двумерная таблица определяется как отношение (множество однотипных записей объединенных одной темой).

От термина relation (отношение) происходит название реляционная модель данных. В реляционных БД используется несколько двумерных таблиц, в которых строки называются записями, астолбцы полями, между записями которых устанавливаются связи. Этот способ организации данных позволяет данные (записи) в одной таблице связывать с данными (записями) в других таблицах через уникальные идентификаторы (ключи) или ключевые поля.

Основные понятия реляционных БД: нормализация, связи и ключи

1. Принципы нормализации:

В каждой таблице БД не должно быть повторяющихся полей;

В каждой таблице должен быть уникальный идентификатор

Каждому значению первичного ключа должна соответствовать достаточная информация о типе сущности или об объекте таблицы (например, информация об успеваемости, о группе или студентах);

Изменение значений в полях таблицы не должно влиять на информацию в других полях (кроме изменений в полях ключа).

2. Виды логической связи.

Связь устанавливается между двумя общими полями (столбцами) двух таблиц. Существуют связи с отношением «один-к-одному», «один-ко-многим» и «многие-ко-многим».

один - к - одному– каждой записи из одной таблицы соответствует одна запись в другой таблице;

один - ко - многим– каждой записи из одной таблицы соответствует несколько записей другой таблице;

многие - ко - многим– множеству записей из одной таблицы соответствует несколько записей в другой таблице.

3. Ключи. Ключ – это столбец добавляемый к таблице и позволяющий установить связь с записями в другой таблице. Существуют ключи двух типов: первичные и вторичные (внешние).

Первичный ключ– это одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ не допускает значений Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

Внешний (вторичный) ключ- это одно или несколько полей (столбцов) в таблице, содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ объединения таблиц.

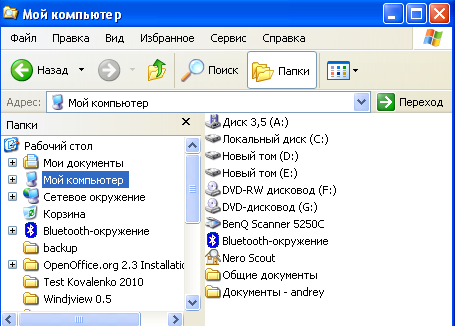
# Иерархическая модель баз данных

Иерархические базы данных — самая ранняя модель представления сложной структуры данных. Информация в иерархической базе организована по принципу древовидной структуры, в виде отношений «предок-потомок». Каждая запись может иметь не более одной родительской записи и несколько подчиненных. Связи записей реализуются в виде физических указателей с одной записи на другую. Основной недостаток иерархической структуры базы данных — невозможность реализовать отношения «много-ко-многим», а также ситуации, когда запись имеет несколько предков.

Иерархические базы данных. Иерархические базы данных графически могут быть представлены как перевернутое дерево, состоящее из объектов различных уровней. Верхний уровень (корень дерева) занимает один объект, второй - объекты второго уровня и так далее.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект, более близкий к корню) к потомку (объект более низкого уровня), при этом объект-предок может не иметь потомков или иметь их несколько, тогда как объект-потомок обязательно имеет только одного предка. Объекты, имеющие общего предка, называются близнецами.

Организация данных в СУБД иерархического типа определяется в терминах: элемент, агрегат, запись (группа), групповое отношение, база данных.

****

**Рисунок 4.1 Иерархическая база данных Каталог папок Windows**

* **Атрибут** (элемент данных) - наименьшая единица структуры данных. Обычно каждому элементу при описании базы данных присваивается уникальное имя. По этому имени к нему обращаются при обработке. Элемент данных также часто называют полем.
* **Запись -** именованная совокупность атрибутов. Использование записей позволяет за одно обращение к базе получить некоторую логически связанную совокупность данных. Именно записи изменяются, добавляются и удаляются. Тип записи определяется составом ее атрибутов. Экземпляр записи - конкретная запись с конкретным значением элементов
* **Групповое отношение** - иерархическое отношение между записями двух типов. Родительская запись (владелец группового отношения) называется исходной записью, а дочерние записи (члены группового отношения) - подчиненными. Иерархическая база данных может хранить только такие древовидные структуры.

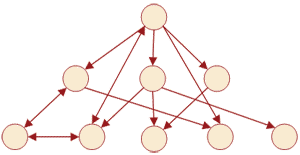
**Корневая запись** каждого дерева обязательно должна содержать ключ с уникальным значением. Ключи некорневых записей должны иметь уникальное значение только в рамках группового отношения. Каждая запись идентифицируется полным сцепленным ключом, под которым понимается совокупность ключей всех записей от корневой по иерархическому пути.

При графическом изображении групповые отношения изображают дугами ориентированного графа, а типы записей - вершинами (диаграмма Бахмана).

# Сетевая модель баз данных

**2. Сетевая модель данных** является более общей структурой по сравнению с иерархической.

В сетевых БД наряду с вертикальными реализованы и горизонтальные связи. Каждый отдельный сегмент (ячейка) может иметь произвольное число непосредственных исходных (старших) сегментов, а также и произвольное число порожденных (младших) ( рис. 8). Это обеспечивает представление отношения "многие к многим".



**Рис. 8.**Сетевая модель данных

Однако унаследованы многие недостатки иерархической и главный из них, необходимость четко определять на физическом уровне связи данных и столь же четко следовать этой структуре связей при запросах к базе. Поскольку логика процедуры выборки данных зависит от их физической организации, то сетевая модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно поменять и приложение.

# Реляционная модель баз данных

**Реляционной моделью** называется база данных, в которой все данные, доступные пользователю, организованны в виде таблиц, а все операции над данными сводятся к операциям над этими таблицами. Наглядной формой представления отношения является двумерная таблица. Таблица имеет строки (записи) и столбцы (колонки). Каждая строка имеет одинаковую структуру и состоит из полей. Строкам таблицы соответствуют кортежи, а столбцам – атрибуты отношения. Достоинство реляционной модели заключается в простоте, понятности и удобстве физической реализации на ЭВМ. Именно простота и понятность для пользователя явились для пользователя основной причиной ее использования. Проблема же эффективности обработки данных этого типа оказались технически вполне разрешимыми. Основными недостатками являются: отсутствие стандартных средств идентификации отдельных записей и сложность описания иерархических и сетевых связей. Примерами зарубежных реляционных СУБД являются: Visual FoxPro и Access (Microsoft).

Основные элементы реляционной модели: Отношение – важнейшее понятие и представляет собой двумерную таблицу, содержащую некоторые данные.

Сущность-объект любой природы, данные о котором хранятся в БД. Данные о сущности хранятся в отношении.

Атрибуты-свойства, характеризующие сущность. В таблице он именуется и ему соответствует заголовок столбца таблицы.

Домен-множество всех возможных значений определенного атрибута отношения.

Схема отношения (заголовок отношения) список имен атрибутов.

Первичный ключ (ключ отношения) атрибут отношения, однозначно идентифицирующий каждый из его картежей. Если кортежи идентифицируются только сцеплением значений нескольких атрибутов, то говорят, что отношение имеет составной ключ.

Достоинства реляционной модели:

-простота и доступность для понимания пользователем. Единственной используемой информационной конструкцией является "таблица";

-строгие правила проектирования, базирующиеся на математическом аппарате;

-полная независимость данных. Изменения в прикладной программе при изменении реляционной БД минимальны;

-для организации запросов и написания прикладного ПО нет необходимости знать конкретную организацию БД во внешней памяти.

Недостатки реляционной модели:

-далеко не всегда предметная область может быть представлена в виде "таблиц";

-в результате логического проектирования появляется множество "таблиц". Это приводит к трудности понимания структуры данных;

-БД занимает относительно много внешней памяти;

-относительно низкая скорость доступа к данным.

# Построение Баз данных

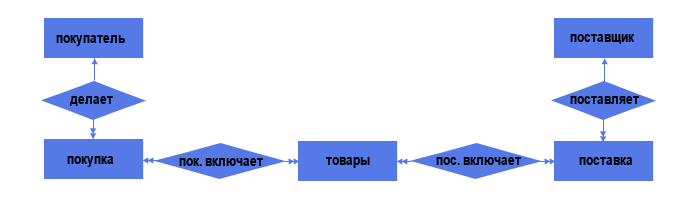
1. концептуальная модель
2. логическая модель
3. Физическая модель

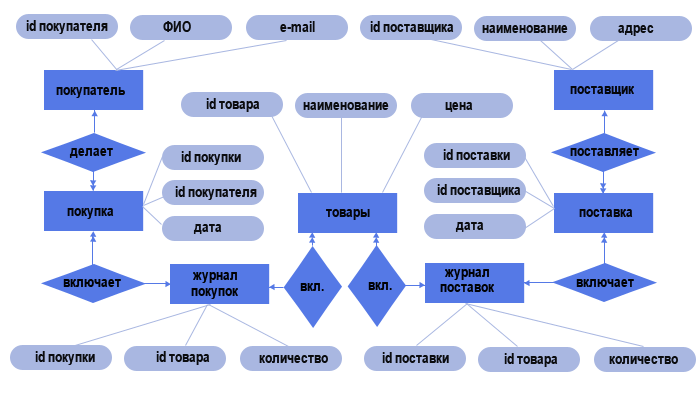
Концептуальная модель

**Концептуальная модель** - это отражение предметной области, для которой разрабатывается база данных.

Не вдаваясь в теорию, отметим, что это некая диаграмма с принятыми обозначениями элементов. Так, все объекты, обозначающие вещи, обозначаются в виде прямоугольника. Атрибуты, характеризующие объект - в виде овала, а связи между объектами - ромбами. Мощность связи обозначаются стрелками (в направлении, где мощность равна многим - двойная стрелка, а со стороны, где она равна единице - одинарная).





Каждый объект нашего магазина имеет свои атрибуты: 

Вообще, если предметная область обширная, то ее полезно разбить на несколько локальных предметных областей (наша концептуальная модель отражает именно локальную предметную область). Объем локальной области выбирается таким образом, чтобы в нее входило не более 6-7 объектов. После создания моделей каждой выделенной предметной области производится объединение локальных концептуальных моделей в одну общую, как правило, довольно сложную схему.

Логическая модель-это версия концептуальная модели, которая может быть обеспечена конкретной СУБД.

Характеристики

Студент

Номер зачетки

Рост

Пол

Весь

Имя

Фамилия

Атрибуты

Физическая модель БД- описывает способ хранения данных, структуру записей, а так же пути доступа к данным.

КАК В MYSQL WORKBENCH

# Функциональные возможности СУБД

Определение данных.

СУБД должна допускать определения данных в исходной форме и преобразовывать эти определения в форму соответствующих объектов. Иначе говоря, СУБД должна включать в себя компонент языкового процессора для различных языков определений данных. СУБД должно также «понимать» синтаксис языка определений данных.

Обработка данных.

СУБД должна уметь обрабатывать запросы пользователя на выборку, изменение или удаление существующих данных в базе данных или на добавление новых данных в базу данных.

Безопасность и целостность данных.

СУБД должна контролировать пользовательские запросы и пресекать попытки нарушения правил безопасности и целостности, определенные АБД.

Восстановление данных и дублирование.

СУБД или другой связанный с ней программный компонент, обычно называемый администратором транзакций, должны осуществлять необходимый контроль над восстановлением данных и дублированием.

Словарь данных.

СУБД должна обеспечить функцию словаря данных. Сам словарь данных можно по праву считать БД (но не пользовательской, а системой). Словарь «содержит данные о данных» (иногда называемые метаданными), т.е. определения других объектов системы, а не просто «сырые данные». В частности, исходная и объектная формы различных схем (внешних, концептуальных и т.д.) и отображений будут сохранены в словаре. Расширенный словарь будет включать также перекрестные ссылки, показывающие, например, какие из программ какую часть БД используют, какие отчеты требуются тем или иным пользователям, какие терминалы подключены к системе и т.д. Словарь может быть (а на самом деле даже должен быть) интегрирован в определяемую им БД, а значит, должен содержать описание самого себя. Конечно, должно быть возможность обращения к словарю, как и к другой БД, например, для того узнать, какие программы и/или пользователи будут затронуты при предполагаемом внесении изменения в систему.

Производительность.

Очевидно, что СУБД должна выполнять все указанные функции с максимально возможной эффективностью.

# Информационная технология обработки данных (технологии централизованной обработки данных)

В эпоху централизованного использования ЭВМ с пакетной обработкой информации пользователи вычислительной техники предпочитали приобретать компьютеры, на которых можно было бы решать почти все классы задач. Однако, сложность решаемых задач обратно пропорциональна их количеству, и это приводило к неэффективному использованию вычислительной мощности ЭВМ при значительных материальных затратах. Кроме того, доступ к компьютерным ресурсам был затруднен из-за политики централизации вычислительных средств в одном месте.

Принцип **централизованной** обработки данных не отвечал высоким требованиям к надежности процесса обработки, затруднял развитие систем и не мог обеспечить необходимые временные параметры при диалоговой обработке данных в многопользовательском режиме.

Появление малых ЭВМ, микроЭВМ, и, наконец, ПК потребовало нового подхода к организации систем обработки данных, к созданию новых ИТ - произошел переход от использования отдельных ЭВМ в системах централизованной обработки данных к **распределенной** обработке данных.

**. Типы многомашинных ассоциаций для распределенной обработки данных.**

**Многомашинные вычислительные комплексы (МВК)** - группа установленных рядом вычислительных машин, объединенных с помощью специальных средств сопряжения и выполняющих совместно единый информационно-вычислительный процесс.

Многомашинные вычислительные комплексы могут быть:

**локальными** при условии установки компьютеров в одном помещении, не требующих для взаимосвязи специального оборудования и каналов связи;

**дистанционными**, если некоторые компьютеры комплекса установлены на значительном расстоянии от центральной ЭВМ и для передачи данных используются телефонные каналы связи.

# Топологии вычислительных сетей

При создании компьютерной сети передачи данных, когда соединяются все компьютеры сети и другие сетевые устройства, формируется топология компьютерной сети.

Сетевая топология(от греч. τоπος, - место) - способ описания конфигурации сети, схема расположения и соединения сетевых устройств.

«Общая шина»

Общая шина являлась до недавнего времени самой распространенной топологией для локальных сетей. В этом случае компьютеры подключаются к одному коаксиальному кабелю по схеме «монтажного ИЛИ». Передаваемая информация, в этом случае, распространяется в обе стороны.

Применение топологии «общая шина» снижает стоимость кабельной прокладки, унифицирует подключение различных модулей, обеспечивает возможность почти мгновенного широковещательного обращения ко всем станциям сети. Основными преимуществами такой схемы являются дешевизна и простота разводки кабеля по помещениям. Самый серьезный недостаток общей шины заключается в ее низкой надежности: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть.

Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть. Поэтому пропускная способность канала связи всегда делится здесь между всеми узлами сети.

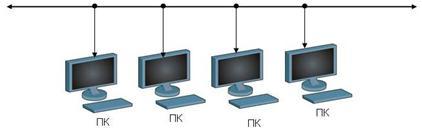


Рисунок 5. Схема подключения компьютеров по схеме «общая шина».

Топология «звезда»

В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому коммутатором (концентратором, хабом) который находится в центре сети. В функции коммутатора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. Главное преимущество этой топологии перед общей шиной - значительно большая надежность. Любые неприятности с кабелем касаются лишь того компьютера, к которому этот кабель присоединен, и только неисправность коммутатора может вывести из строя всю сеть. Кроме того, коммутатор может играть роль интеллектуального фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

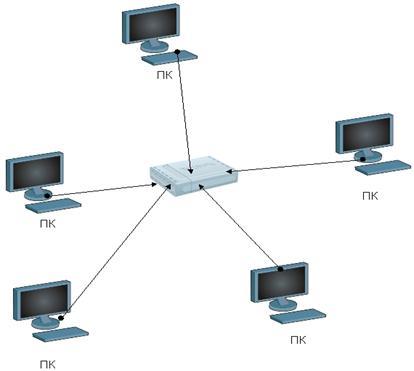


Рисунок 6. Схема подключения компьютеров по схеме «звезда»

Топология «кольцо»

В информационно вычислительных сетях с кольцевой конфигурациейданные передаются по кольцу от одного компьютера к другому, как правило, в одном направлении. Если компьютер распознает данные как «свои», то он копирует их себе во внутренний буфер. Кольцо представляет собой очень удобную конфигурацию для организации обратной связи - данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому этот узел может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. Для этого в сеть посылаются специальные тестовые сообщения.

В сети с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями.

Поскольку такое дублирование повышает надёжность системы, данный стандарт с успехом применяется в магистральных каналах связи.

Данная физическая топология с успехом реализуется в сетях, созданных с использованием технологии FDDI.

FDDI(англ. Fiber Distributed Data Interface — распределённый волоконный интерфейс данных)- стандарт передачи данных в локальной сети , протяжённостью до 200 километров . Стандарт основан на протоколеToken Bus. В качестве среды передачи данных вFDDIрекомендуется использовать волоконно-оптический кабель , однако можно использовать и медный кабель, в таком случае используется сокращениеCDDI(Copper Distributed Data Interface).В качестве топологии используется схемадвойного кольца, при этом данные в кольцах циркулируют в разных направлениях. Одно кольцо считается основным, по нему передаётся информация в обычном состоянии; второе — вспомогательным, по нему данные передаются в случае обрыва на первом кольце. Для контроля за состоянием кольца используется сетевой маркер , как и в технологииToken Ring.

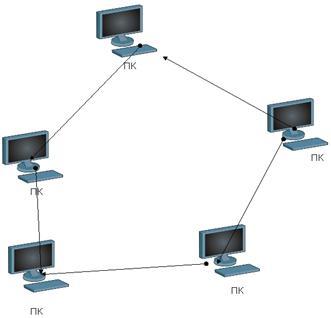


Рисунок 7. Схема подключения компьютеров по схеме «кольцо»

# Технология и модель клиент-сервер.

Клиент-сервер - это модель взаимодействия компьютеров в сети. Как правило, компьютеры не являются равноправными. Каждый из них имеет свое, отличное от других, назначение, играет свою роль. Некоторые компьютеры в сети владеют и распоряжается информационно-вычислительными ресурсами, такими как процессоры, файловая система, почтовая служба, служба печати, база данных. Другие же компьютеры имеют возможность обращаться к этим службам, пользуясь услугами первых. Компьютер, управляющий тем или иным ресурсом, принято называть **сервером** этого ресурса, а компьютер, желающий им воспользоваться - **клиентом**. Конкретный сервер определяется видом ресурса, которым он владеет. Так, если ресурсом являются базы данных, то речь идет о *сервере баз данных*, назначение которого - обслуживать запросы клиентов, связанные с обработкой данных; если ресурс - это файловая система, то говорят о *файловом сервере*, или файл-сервере, и т. д.

В сети один и тот же компьютер может выполнять роль как клиента, так и сервера.

Этот же принцип распространяется и на взаимодействие программ

Если предполагается, что проектируемая информационная система (ИС) будет иметь технологию клиент-сервер, то это означает, что прикладные программы, реализованные в ее рамках, будут иметь распределенный характер. Иными словами, часть функций прикладной программы (или, проще, приложения) будет реализована в программе-клиенте, другая - в программе-сервере, причем для их взаимодействия будет определен некоторый протокол.

Основной принцип технологии клиент-сервер заключается в разделении функций стандартного интерактивного приложения на четыре группы, имеющие различную природу. Первая группа - это *функции ввода и отображения данных*. Вторая группа объединяет чисто *прикладные функции*, характерные для данной предметной области (например, для банковской системы - открытие счета, перевод денег с одного счета на другой и т.д.). К третьей группе относятся фундаментальные *функции хранения и управления информационными ресурсами* (базами данных, файловыми системами и т.д.). Наконец, функции четвертой группы - это *служебные функции* (играющие роль связок между функциями первых трех групп.

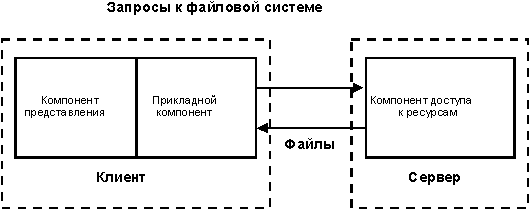
В соответствии с этим в любом приложении выделяются следующие логические компоненты:

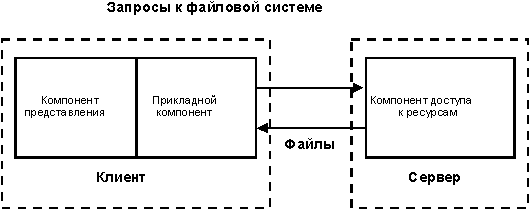
· **компонент представления**, реализующий функции первой группы;

· **прикладной компонент**, поддерживающий функции второй группы;

· **компонент доступа к информационным ресурсам**, поддерживающий функции третьей групп, а также вводятся и уточняются соглашения о способах их взаимодействия (протокол взаимодействия).

# модель файлового сервера



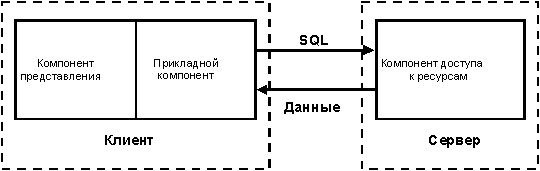
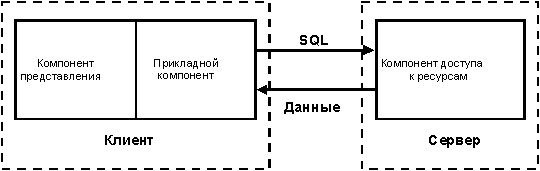
**Рисунок 2.** Модель файлового сервера.

**FS-модель** является базовой для локальных сетей персональных компьютеров. Не так давно она была исключительно популярной среди отечественных разработчиков, использовавших такие системы, как FoxPRO, Clipper, Clarion, Paradox и т.д. Суть модели проста и всем известна. Один из компьютеров в сети считается файловым сервером и предоставляет услуги по обработке файлов другим компьютерам. Файловый сервер работает под управлением сетевой операционной системы (например, Novell NetWare) и играет роль компонента доступа к информационным ресурсам (то есть к файлам). На других компьютерах в сети функционирует приложение, в кодах которого совмещены компонент представления и прикладной компонент (рис.2). Протокол обмена представляет собой набор низкоуровневых вызовов, обеспечивающих приложению доступ к файловой системе на файл-сервере.

FS-модель послужила фундаментом для расширения возможностей персональных СУБД в направлении поддержки многопользовательского режима. В таких системах на нескольких персональных компьютерах выполняется как прикладная программа, так и копия СУБД, а базы данных содержатся в разделяемых файлах, которые находятся на файловом сервере. Когда прикладная программа обращается к базе данных, СУБД направляет запрос на файловый сервер. В этом запросе указаны файлы, где находятся запрашиваемые данные. В ответ на запрос файловый сервер направляет по сети требуемый блок данных. СУБД, получив его, выполняет над данными действия, которые были декларированы в прикладной программе.

К технологическим недостаткам модели относят высокий сетевой трафик (передача множества файлов, необходимых приложению), узкий спектр операций манипуляции с данными (данные - это файлы), отсутствие адекватных средств безопасности доступа к данным (защита только на уровне файловой системы) и т.д. Собственно, перечисленное не есть недостатки, но - следствие внутренне присущих FS-модели ограничений, определяемых ее характером. Недоразумения возникают, когда FS-модель используют не по назначению, например, пытаются интерпретировать как модель сервера базы данных. Место FS-модели в иерархии моделей клиент-сервер - это место модели файлового сервера, и ничего более. Именно поэтому обречены на провал попытки создания на основе FS-модели крупных корпоративных систем - попытки, которые предпринимались в недавнем прошлом и нередко предпринимаются сейчас.

# Модель удаленного доступа к данным.



**Рисунок 3.** Модель доступа к удаленным данным.

Более технологичная **RDA-модель** существенно отличается от FS-модели характером компонента доступа к информационным ресурсам. Это, как правило, SQL-сервер. В RDA-модели коды компонента представления и прикладного компонента совмещены и выполняются на компьютере-клиенте. Последний поддерживает как функции ввода и отображения данных, так и чисто прикладные функции. Доступ к информационным ресурсам обеспечивается либо операторами специального языка (языка SQL, например, если речь идет о базах данных), либо вызовами функций специальной библиотеки (если имеется соответствующий интерфейс прикладного программирования - API).

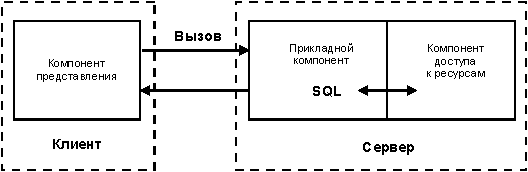
Клиент направляет запросы к информационным ресурсам (например, к базам данных) по сети удаленному компьютеру. На нем функционирует ядро СУБД, которое обрабатывает запросы, выполняя предписанные в них действия, и возвращает клиенту результат, оформленный как блок данных (рис.3). При этом инициатором манипуляций с данными выступают программы, выполняющиеся на компьютерах-клиентах, в то время как ядру СУБД отводится пассивная роль - обслуживание запросов и обработка данных.

RDA-модель избавляет от недостатков, присущих как системам с централизованной архитектурой, так и системам с файловым сервером.

Прежде всего перенос компонента представления и прикладного компонента на компьютеры-клиенты существенно разгружает сервер БД, сводя к минимуму общее число процессов операционной системы. Сервер БД освобождается от несвойственных ему функций; процессор или процессоры сервера целиком загружаются операциями обработки данных, запросов и транзакций. Это становится возможным благодаря отказу от терминалов и оснащению рабочих мест компьютерами, которые обладают собственными локальными вычислительными ресурсами, полностью используемыми программами переднего плана. С другой стороны, резко уменьшается загрузка сети, так как по ней передаются от клиента к серверу не запросы на ввод-вывод (как в системах с файловым сервером), а запросы на языке SQL, их объем существенно меньше.

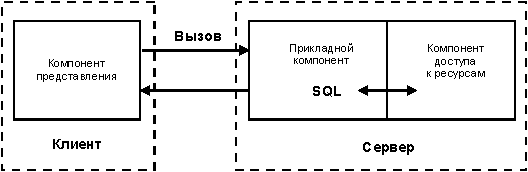
Основное достоинство RDA-модели - унификация интерфейса клиент-сервер в виде языка SQL. Действительно, взаимодействие прикладного компонента с ядром СУБД невозможно без стандартизованного средства общения. Запросы, направляемые программой ядру, должны быть понятны обоим. Для этого их следует сформулировать на специальном языке. Но в СУБД уже существует язык SQL, о котором уже шла речь . Поэтому целесообразно использовать его не только в качестве средства доступа к данным, но и стандарта общения клиента и сервера.

# Модель сервера базы данных



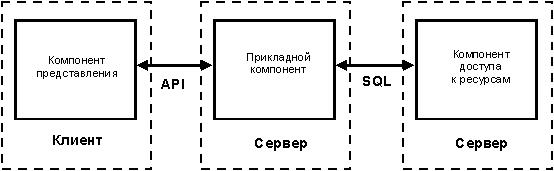
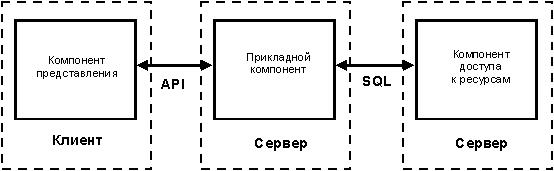
**Рисунок 4.** Модель сервера базы данных.

Наряду с RDA-моделью все большую популярность приобретает перспективная **DBS-модель** (рис. 4). Последняя реализована в некоторых реляционных СУБД (Informix, Ingres, Sybase, Oracle). Ее основу составляет механизм *хранимых процедур* - средство программирования SQL-сервера. Процедуры хранятся в словаре базы данных, разделяются между несколькими клиентами и выполняются на том же компьютере, где функционирует SQL-сервер. Язык, на котором разрабатываются хранимые процедуры, представляет собой процедурное расширение языка запросов SQL и уникален для каждой конкретной СУБД. Более подробно о хранимых процедурах рассказано в п. 2.3.3.

В DBS-модели компонент представления выполняется на компьютере-клиенте, в то время как прикладной компонент оформлен как набор хранимых процедур и функционирует на компьютере-сервере БД. Там же выполняется компонент доступа к данным, то есть ядро СУБД. Достоинства DBS-модели очевидны: это и возможность централизованного администрирования прикладных функций, и снижение трафика (вместо SQL-запросов по сети направляются вызовы хранимых процедур), и возможность разделения процедуры между несколькими приложениями, и экономия ресурсов компьютера за счет использования единожды созданного плана выполнения процедуры. К недостаткам модели можно отнести ограниченность средств, используемых для написания хранимых процедур, которые представляют собой разнообразные процедурные расширения SQL, не выдерживающие сравнения по изобразительным средствам и функциональным возможностям с языками третьего поколения, такими как C или Pascal. Сфера их использования ограничена конкретной СУБД, в большинстве СУБД отсутствуют возможности отладки и тестирования разработанных хранимых процедур.

На практике часто используется смешанные модели, когда поддержка целостности базы данных и некоторые простейшие прикладные функции поддерживаются хранимыми процедурами (DBS-модель), а более сложные функции реализуются непосредственно в прикладной программе, которая выполняется на компьютере-клиенте (RDA-модель). Так или иначе современные многопользовательские СУБД опираются на RDA- и DBS-модели и при создании ИС, предполагающем использование только СУБД, выбирают одну из этих двух моделей либо их разумное сочетание.

# Модель сервера приложений



**Рисунок 5.** Модель сервера приложений.

В **AS-модели** (рис.5) процесс, выполняющийся на компьютере-клиенте, отвечает, как обычно, за интерфейс с пользователем (то есть осуществляет функции первой группы). Обращаясь за выполнением услуг к прикладному компоненту, этот процесс играет роль *клиента приложения* (*Application Client - AC*). Прикладной компонент реализован как группа процессов, выполняющих прикладные функции, и называется сервером приложения (*Application Server - AS*). Все операции над информационными ресурсами выполняются соответствующим компонентом, по отношению к которому AS играет роль клиента. Из прикладных компонентов доступны ресурсы различных типов - базы данных, очереди, почтовые службы и др.

RDA- и DBS-модели опираются на двухзвенную схему разделения функций. В RDA-модели прикладные функции приданы программе-клиенту, в DBS-модели ответственность за их выполнение берет на себя ядро СУБД. В первом случае прикладной компонент сливается с компонентом представления, во-втором - интегрируется в компонент доступа к информационным ресурсам. В AS-модели реализована трехзвенная схема разделения функций, где прикладной компонент выделен как важнейший изолированный элемент приложения, для его определения используются универсальные механизмы многозадачной операционной системы, и стандартизованы интерфейсы с двумя другими компонентами. AS-модель является фундаментом для мониторов обработки транзакций (*Transaction Processing Monitors - TPM*), или, проще, мониторов транзакций, которые выделяются как особый вид программного обеспечения. Мониторы транзакций - предмет **Раздела 4**.

В заключение отметим, что, часто, говоря о сервере базы данных, подразумевают как компьютер, так и программное обеспечение - ядро СУБД. При описании архитектуры Клиент-сервер под сервером базы данных мы имели в виду компьютер. Далее сервер базы данных будет пониматься как программное обеспечение - ядро СУБД.

# Общая характеристика создания ПО

**Технология разработки программного обеспечения** (ПО) – это комплекс мер по созданию программных продуктов (ПП). Данная деятельность включает в себя несколько этапов, с которыми так или иначе придётся столкнуться при разработке достаточно крупного ПО.

Ключевым понятием в технологии разработки ПО является понятие жизненного цикла программного продукта. С его рассмотрения мы и начнём.

## Жизненный цикл программы

Перечислим основные **этапы жизненного цикла программы** и дадим краткую характеристику каждому из этапов. Всякая разработка включает в себя:

* **Процесс приобретения**. Данный процесс представляет собой действия заказчика разработки ПО, и обычно включает в себя такие мероприятия, как: формирование требований и ограничений к программному продукту (ограничения могут быть связаны с выбором программной архитектуры, а также с приемлемым быстродействием системы и т.д.); заключение договора на разработку; анализ и аудит работы исполнителя. В конце данного процесса заказчик осуществляет приёмку готового программного продукта.
* **Процесс поставки** включает в себя мероприятия, проводимые исполнителем по поставке ПО. Исполнитель анализирует требования заказчика, выполняет проектирование и анализ работ, решает, как будет происходить процесс конструирования (программирования): своими силами, либо же с привлечением сторонних команд разработки (подрядчика), также осуществляет оценку и контроль качества готового программного продукта и выполняет непосредственно поставку продукта и сопутствующие завершающие мероприятия.
* **Процесс разработки**. Его мы подробно рассмотрим в разделе [“Этапы создания программных продуктов”](https://vscode.ru/articles/tehnologiya-razrabotki-po.html#razdel1).
* **Процесс эксплуатации**. После того, как программное обеспечение будет готово, начинается процесс его эксплуатации организацией-заказчиком и её операторами.
* **Процесс сопровождения**. Фирма-разработчик осуществляет поддержку пользователей программного продукта в случае возникновения у них каких-либо вопросов или проблем. Если в процессе эксплуатации будет обнаружена ошибка в ПП, разработчики должны её устранить. Процесс эксплуатации и процесс сопровождения идут параллельно.

**Этапы создания ПО**

* **Составление требований заказчика**. На данном эта производится работа с заказчиком и документирование его видения и его требований к программе. В подавляющем большинстве случаев данный этап проходит трудно. Поскольку, слабо разбираясь в особенностях разработки ПО, заказчик плохо представляет себе, что нужно знать разработчикам и (самое главное!), что им нужно сообщить о продукте.  
  Выработка требований чрезвычайно важное мероприятие. Убедитесь, что все требования полностью понятны вам и вашей команде.
* **Проектирование программного продукта**. Разобравшись в предметной области, разработчики приступают к проектированию. На данном этапе создания программного продукта разрабатывается архитектура компонентов ПО, выбираются нужные [шаблоны проектирования (паттерны)](https://vscode.ru/prog-lessons/strategy-pattern.html) и составляется схема информационной базы данных системы.
* **Разработка**. Когда требования сформулированы и архитектура готова – команда начинает разработку ПП. На этапе разработки также выполняется документирование системы.
* **Тестирование**. После разработки необходимо произвести тестирование системы в целом, тем самым подтвердить её соответствие требованиям заказчика.  
  Здесь стоит сказать, что модульные тесты (unit-тесты; т.е. тесты отдельных частей программы) обычно выполняются на этапе разработки программистом, разрабатывавшем конкретный модуль.  
  Когда все тесты пройдены, программное обеспечение готово к выпуску.
* **Сопровождение ПП**. После выпуска фирма-разработчик отвечает за поддержку программного продукта и выпуска новых версий, которые исправляют ошибки и привносят новый функционал. Также необходимо осуществлять поддержку пользователей разработанного ПО.

# Современные методы разработки ПО

***Метод нисходящего проектирования***

Суть метода заключается в определении спецификаций компонентов системы путем последовательного выделения в ее составе отдельных составляющих и их постепенной детализации до уровня, обеспечивающего однозначное понимание того, что и как необходимо разрабатывать и реализовывать.

**Модульное программирование**

Реализация метода *нисходящего проектирования* тесно связана с другим понятием программирования - *модульным проектированием*, так как на практике при декомпозиции сложной программы возникает вопрос о разумном пределе ее дробления на составные части. Вместе с тем понятие модульности нельзя сводить только к представлению сложных программных комплексов в виде набора отдельных функциональных блоков.

*Модуль* - это последовательность логически взаимосвязанных фрагментов задачи, оформленных как отдельная часть программы. При этом программные модули должны обладать следующими свойствами:

* на модуль можно ссылаться (т.е. обращаться к нему) по имени, в том числе и из других модулей;
* по завершении работы модуль должен возвращать управление тому модулю, который его вызывал;
* модуль должен иметь один вход и выход;
* модуль должен иметь небольшой размер, обеспечивающий его обозримость.

При разработке сложных программ в них выделяют головной управляющий модуль, подчиненные ему модули, обеспечивающие реализацию отдельных функций управления, функциональную обработку (т.е. непосредственную реализацию основного назначения программного комплекса), а также вспомогательные модули, обеспечивающие сервисное обслуживание пакета (например, сбор и анализ статистики работы программы, обработка различного рода ошибочных ситуаций, обучение и выдача подсказок и т.п.).

Модульный принцип разработки программ обладает следующими преимуществами:

* большую программу могут разрабатывать одновременно несколько исполнителей, и это позволяет сократить сроки ее разработки;
* появляется возможность создавать и многократно использовать в дальнейшем библиотеки наиболее употребимых программ;
* упрощается процедура загрузки больших программ в оперативную память, когда требуется ее сегментация;
* возникает много естественных контрольных точек для наблюдения за осуществлением хода разработки программ, а в последующем для контроля за ходом исполнения программ;
* обеспечивается более эффективное тестирование программ, проще осуществляются проектирование и последующая отладка.

***Структурное программирование***

Актуальная для начального периода развития и использования ЭВМ проблема разработки программ, занимающих минимум основной памяти и выполняющихся за кратчайшее время, в последующем в связи резким падением стоимости аппаратной части ЭВМ, значительным возрастанием их быстродействия и объемов памяти сменилась необходимостью разработки и применения принципиально новых методов составления программ. Все это нашло свое воплощение в разработке принципа *структурного программирования.* Одной из целей структурного программирования было стремление облегчить разработку и отладку программных модулей, а главное - их последующее сопровождение и модификацию.

***CASE-технологии***

*CASE-технология* представляет собой совокупность средств системного анализа, проектирования, разработки и сопровождения сложных программных систем, поддерживаемых комплексом взаимоувязанных инструментальных средств автоматизации всех этапов разработки программ. Благодаря структурным методам CASE-технология на стадиях анализа и проектирования обеспечивает разработчиков широкими возможностями для различного рода моделирования, а централизованное хранение всей необходимой для проектирования информации и контроль за целостностью данных гарантируют согласованность взаимодействия всех специалистов, занятых в разработке ПО.

***Технологии RAD***

Это дало пользователю возможность, получая уже с первых шагов конкретное представление о характере реализации задачи, уточнять ее постановку. Тем самым облегчался процесс экспериментального поиска нужного решения автоматизации задачи. Благодаря тесному взаимодействию разработчика с заказчиком (пользователем) на самом ответственном этапе создания прикладных программ между ними достигалось быстрое взаимопонимание цели поставленной задачи и возможности ее автоматизации в данных конкретных условиях. Это повышало скорость разработки программ и послужило основанием для названия такой технологии *RAD (Rapid Application Development* - быстрая разработка программ), которая получила широкое распространение.

# Инструментальные системы технологии программирования.

При разработке программных средств используется в той или иной мере компьютерная поддержка процессов разработки и сопровождения ПС [16.1]. Это достигается путем представления хотя бы некоторых программных документов ПС (прежде всего, программ) на компьютерных носителях данных (например, на дискетах) и предоставлению в распоряжение разработчика ПС специальных ПС или включенных в состав компьютера специальных устройств, созданных для какой-либо обработки таких документов. В качестве такого специального ПС можно указать компилятор с какого-либо языка программирования. Компилятор избавляет разработчика ПС от необходимости писать программы на языке компьютера, который для разработчика ПС был бы крайне неудобен, - вместо этого он составляет программы на удобном ему языке программирования, которые соответствующий компилятор автоматически переводит на язык компьютера.

ПС, предназначенное для поддержки разработки других ПС, будем называть *программным инструментом* разработки ПС, а устройство компьютера, специально предназначенное для поддержки разработки ПС, будем называть *аппаратным инструментом* разработки ПС.

# Инструменты разработки пс

Инструменты разработки ПС могут использоваться в течение всего жизненного цикла ПС [16.2] для работы с разными программными документами. Так текстовый редактор может использоваться для разработки практически любого программного документа. С точки зрения функций, которые инструменты выполняют при разработке ПС, их можно разбить на следующие четыре группы:

* редакторы,
* анализаторы,
* преобразователи,
* инструменты, поддерживающие процесс выполнения программ.

*Редакторы* поддерживают конструирование (формирование) тех или иных программных документов на различных этапах жизненного цикла.

*Анализаторы* производят либо *статическую* обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных (например, проверку соответствия документов указанным стандартам), либо *динамический* анализ программ (например, с целью выявление распределения времени работы программы по программным модулям).

*Преобразователи* позволяют автоматически приводить документы к другой форме представления (например, форматеры) или переводить документ одного вида к документу другого вида (например, конверторы или компиляторы), синтезировать какой-либо документ из отдельных частей и т.п.

*Инструменты, поддерживающие процесс выполнения программ,* позволяют выполнять на компьютере описания процессов или отдельных их частей, представленных в виде, отличном от машинного кода, или машинный код с дополнительными возможностями его интерпретации.

# ПРИНЦИП ВИЗУАЛЬНОГО ПРОГРАММИРОВАННИЯ

# Основные понятия языком программирования в HiASM

HiAsm – среда визуального программирования, программы в которой не пишутся, как это делается в классических языках, а проектируются из кубиков и линий связей между ними. Каждый такой кубик представляет собой элемент, который выполняет какое-то достаточно простое и узкоспециализированное действие (например, копирует файл, проигрывает звук, складывает два числа, качает файл из Internet и т.д. и т.п.). Горизонтальные линии между элементами (или просто «связи») определяют логику будущей программы (то есть последовательность вызова событий и методов, если выражаться терминами классических языков программирования). Вертикальные же связи указывают элементам на то, какие данные откуда брать. Также у каждого элемента есть набор уникальных свойств, которые определяют особенности его функционирования (к примеру, у элемента «Кнопка» есть свойства, определяющие его положение на форме, заголовок, используемый шрифт, текст выводимой подсказки и прочие). Поэтому весь процесс конструирования сводится к расстановке элементов, протягиванию связей между ними и настройке (если необходимо) их свойств.

# Системы программирования

**Системы программирования** ­­­– это комплекс инструментальных программных средств, предназначенных для работы с программами на одном из языков программирования. Системы программирования представляют сервисные возможности программистам для разработки их собственных компьютерных программ.

В настоящее время разработка любого системного и прикладного программного обеспечения осуществляется с помощью систем программирования, в состав которых входят:

* Трансляторы с языков высокого уровня;
* Средства редактирования, компоновки и загрузки программ;
* Макроассемблеры (машинно-ориентированные языки);
* Отладчики машинных программ.

Системы программирования, как правило, включают в себя:

* Текстовый редактор (Edit), осуществляющий функции записи и редактирования исходного текста программы;
* Загрузчик программ(Load), позволяющий выбрать из директория нужный текстовый файл программы;
* Запускатель программ (Run), осуществляющий процесс выполнения программы;
* Компилятор (Compile), предназначенный для компиляции или интерпретации исходного текста программы в машинный код с диагностикой синтаксических и семантических (логических) ошибок;
* Отладчик (Debug), выполняющий сервисные функции по отладке и тестированию программы;
* Диспетчер файлов (File), предоставляющий возможность выполнять операции с файлами: сохранение, поиск, уничтожение и т.п.

# Развитие языков программирования

Первые языки программирования возникли в 20-е, 30-е годы ХХ столетия и были довольно примитивны и ориентированы на численные расчеты (математические и физические) и прикладные задачи, в частности, в области военного дела.

Программы, написанные на ранних языках программирования, были оптимизированы под аппаратную архитектуру конкретного компьютера, для которого предназначались. Они обеспечивали высокую эффективность вычислений, но до стандартизации было еще далеко. Программа, которая была вполне работоспособной на одной вычислительной машине, зачастую не могла выполняться на другой. Таким образом, ранние языки программирования зависели от того, что принято называть средой вычислений и приблизительно соответствовали современным машинным кодам или языкам ассемблер. Это так называемые **низкоуровневые языки** программирования.

Язык программирования считается языком низкого уровня, если он ориентирован на конкретный тип процессора и учитывает его особенности (операторы языка близки к машинному коду и ориентированы на конкретные команды процессора). Подобные языки обычно применяются для написания небольших системных приложений, драйверов устройств. Примером такого языка может служить Ассемблер (assembler – сборщик), который появился в конце 50-х – начале 60-х г.г.

Следующее поколение языков программирования – **языки высокого уровня**. Различие состоит в повышении эффективности труда разработчиков за счет абстрагирования от конкретных деталей аппаратного обеспечения. Одна инструкция (оператор) языка высокого уровня соответствовала последовательности из нескольких низкоуровневых инструкций, или команд. Исходя из того, что программа, по сути, представляла собой набор директив, обращенных к компьютеру, такой подход к программированию получил название **императивного**.

Еще одной особенностью языков высокого уровня была возможность использования ранее написанных программных блоков, которые получили название **функций** и **процедур**.

Кроме того, с появлением языков высокого уровня зависимость реализации от аппаратного обеспечения существенно уменьшилась. Платой за это стало появление специальных **программ-трансляторов**.

Следующим этапом в развитии технологии программирования стала концепция **структурного программирования**. Структурный подход предполагает использование только нескольких основных структур (следование, разветвление, цикл), комбинация которых дает все многообразие алгоритмов и программ.

С течением времени программные проекты становились сложнее и больше по размеру. Развитие технологии программирования требовало подхода, который бы позволил выделять алгоритмы, данные и описания новых типов в независимые блоки – модули.

**Модуль** – логически взаимосвязанная совокупность функциональных элементов, оформленных в виде отдельных программ.

**Модульное программирование** основано на понятии модуля.Принципы модульного программирования программных продуктов во многом сходны с принципами нисходящего проектирования.

**Метод нисходящего проектирования** предполагает последовательное разложение общей функции обработки данных на простые функциональные элементы методом «сверху - вниз», от целого к части.

**Структурное программирование** основано на модульной структуре программного продукта и типовых управляющих структурах алгоритмов обработки данных различных программных модулей. В любой типовой структуре блок, кроме условного (который имеет два выхода), имеет только один вход и один выход. Виды основных управляющих структур – последовательность, альтернатива (условие выбора), цикл.

Важным шагом по пути к совершенствованию языков программирования стало появление **объектно-ориентированного подхода** к программированию и соответствующего класса языков. В рамках данного подхода программа представляет собой описание объектов, их свойств, совокупностей, отношений между ними, способов их взаимодействия и операций над объектами (методов). Важным свойством объектно-ориентированного подхода является поддержка механизма обработки событий, которые изменяют атрибуты объектов и моделируют их взаимодействие.