

Лекция 2. Виды обеспечения САПР.

2.1 Техническое обеспечение САПР

2.2 Программное обеспечение САПР

2.3 Лингвистическое обеспечение САПР. Языки программирования

2.4 Лингвистическое обеспечение САПР. Языки управления и проектирования

2.5 Информационное обеспечение САПР. Определения и понятия

2.6 Информационное обеспечение САПР. Ведение и организация информационного фонда на ЭВМ

2.1 Техническое обеспечение САПР

Основу технического обеспечения САПР составляет, как правило, персональный компьютер. Конструктивно он представляет собой системный блок, состоящий из корпуса с блоком питания, в котором установлены: материнская плата с процессором, оперативной памятью, видеокартой (видеоплатой), при необходимости звуковой картой и сетевой картой; жесткий диск (винчестер); привод для компакт – дисков; привод для дискет. Кроме этого в состав компьютера обязательно входит монитор (дисплей), клавиатура, манипулятор «мышь». Это основные устройства компьютера. Кроме них применяются периферийные устройства: принтер, плоттер, звуковые колонки, микрофон, цифровой фотоаппарат, цифровая видеокамера.

На практике в настоящее время широко применяются локальные вычислительные сети (ЛВС). Это принадлежащая одной организации коммуникационная система, связывающая различные аппаратные средства: компьютеры, принтеры, плоттеры. Слово «локальная» указывает на близость расположения компьютеров. Диапазон действия ЛВС колеблется от нескольких метров до 8 – 10 км.

ЛВС предоставляет пользователям следующие возможности:

- **обмен информацией** (сообщениями электронной почтой, файлами текстовых документов, чертежей и программ);
- **разделение ресурсов компьютеров**, т.е. совместное использование баз данных и программ, хранящихся на любом из компьютеров сети (либо на удаленном мощном компьютере – сервере с жестким диском большой емкости);
- **вывод информации**, например, на лазерный принтер или плоттер, подключенный только к одному из компьютеров сети.

ЛВС состоит из следующих основных элементов:

- файлового сервера;
- рабочих станций;
- сетевой операционной системы;
- несущей среды (кабелей), сетевых карт и других аппаратных средств.

Сервер – это мощная ПЭВМ, на жестком диске которой хранятся прикладные программы, базы данных и т.д., необходимые для работы пользователей сети. Сервер, предназначенный только для обслуживания сетевых запросов, называется выделенным. При генерации (установке) сети можно сформировать и совмещенный сервер, на котором можно работать как на рабочей станции.

Рабочие станции – подключенные к сети ПЭВМ, на которых работают отдельные пользователи.

Каждая рабочая станция и сервер в ЛВС должны иметь специальное программное обеспечение: сетевую оболочку или операционную систему.

Кабель в ЛВС определяет физическую среду передачи информации. Существует три типа кабелей:

1. **Витая пара** (физически этот кабель состоит из четырех витых пар в оплетке, одна пара используется для передачи информации в одном направлении, вторая – в другом направлении, две оставшиеся пары предназначены для передачи служебных сообщений по сети, на практике иногда они не используются).
2. **Коаксиальный** (типа телевизионного).
3. **Волоконно – оптический** (физически это кварцевая нить в полимерной оплетке, оплетка предназначена для придания гибкости кабелю; передача информации по

кабелю производится световыми излучениями с **разной** длиной волны, за счет чего образуется ряд информационных каналов).

Скорость передачи информации – важнейший показатель эффективности сети, она измеряется в Мбит/с, Гбит/с. Скорость передачи информации по витой паре составляет от 10 до 100 Мбит/с, по коаксиальному кабелю – от 0,5 до 10 Мбит/с, по волоконно – оптическому теоретически – сотни Гбит/с, практически – около 2 Гбит/с (за счет более низкой пропускной способности приемных и передающих устройств).

Сетевые карты физически могут быть встроены в материнскую плату или устанавливаться в разъемы системного блока компьютера. Их тип определяется выбранной топологией сети.

В ЛВС компьютеры располагаются сравнительно недалеко друг от друга. Для связи на большом расстоянии можно использовать аппаратуру обычных телефонных линий, которая, правда, поддерживает относительно низкую скорость передачи информации. Дополнительным устройством при этом является модем. Когда с компьютера информация передается по телефонной линии, передаваемые сигналы подвергаются **модуляции**, а когда принимается – **демодуляции**. Отсюда название – модем. Назначение модема – замена двоичного сигнала компьютера (сочетания 0 и 1) аналоговым сигналом с частотой, соответствующей рабочему диапазону телефонной линии.

Конструктивно модем – это печатная плата, вставляемая в компьютер или присоединяемая к нему, связанная с кабелем, подключаемым к телефонной розетке.

Телефонные сети начинают переводиться на цифровые сигналы, совместимые с сигналами компьютеров. Поэтому необходимость в модемах в перспективе отпадет.

2.2 Программное обеспечение САПР

Программное обеспечение (ПО) САПР – совокупность машинных программ и сопутствующих им эксплуатационных документов, необходимых для выполнения автоматизированного проектирования.

ПО подразделяется на *общее* и *прикладное* (специальное). В свою очередь, общее ПО можно подразделить на *общесистемное* программное обеспечение и *языки* (среды, студии) *программирования*.

К программному обеспечению предъявляются следующие требования:

- **экономичность** (эффективность по быстродействию и затратам памяти);
- **удобство использования**, применение простых проблемно-ориентированных языков, наличие средств диагностики ошибок пользователя и т.п.;
- **надежность и правильность получения результатов проектирования**; универсальность по отношению к тем или иным ограничениям решаемых задач;
- **открытость** (адаптируемость) относительно внесения изменений в процессе эксплуатации программ;
- **сопровождаемость**, характеризующая работоспособность программ при внесении изменений в них;
- **мобильность** при перестройке программ с ЭВМ одного типа на ЭВМ другого типа.

Программное обеспечение целесообразно разрабатывать на основе принципов модульности и иерархичности. Принципы модульности и иерархичности позволяют организовать коллективную параллельную разработку различных частей программного обеспечения, создавать открытые программные системы, облегчают их комплексную отладку и информационное согласование.

2.2.1 Общесистемное программное обеспечение САПР

Общесистемное ПО служит для организации функционирования технических средств. **Операционная система** является основным компонентом системного программного обеспечения САПР.

Операционная система – это комплекс программ, который загружается при включении компьютера. Она производит диалог с пользователем, осуществляет управление компьютером, его ресурсами (оперативной памятью, местом на дисках и т.д.), запускает прикладные программы на выполнение и т.д.

Операционные системы включают в себя программы двух групп.

- **обрабатывающие** программы, составляющие подсистему подготовки программ пользователя (внешнее программное обеспечение);
- **управляющие** программы, образующие группу исполнения программ пользователя (внутреннее программное обеспечение).

К обрабатывающим программам относятся трансляторы с алгоритмических языков, библиотеки стандартных программ и системные обслуживающие программы.

Управляющая часть программного обеспечения имеет иерархическую организацию и в общем случае в ней можно выделить различные уровни: уровень операционных систем вычислительной сети, операционных систем отдельных ЭВМ, мониторных систем САПР и мониторов отдельных ППП.

Основные функции управляющей части, связь с пользователем в режиме диалога, планирование вычислительного процесса, распределение вычислительных ресурсов, динамическое распределение памяти и др.

Группа управляющих программ включает в себя программы управления задачами, заданиями и данными.

Программа управления задачами (супервизор, диспетчер, монитор, резидентная программа) находится в оперативной памяти и выполняет все необходимые диспетчерские функции — переключение с выполнения одной программы на другую, распределения ресурсов времени и оперативной памяти между программами. Именно супервизор реализует мультипрограммный режим работы ЭВМ или режим разделения времени.

Программы управления заданиями выполняют интерпретацию директив языка управления заданиями — ввод, трансляция, загрузка в память ЭВМ, решение, вывод информации.

Программы управления данными обеспечивают поиск, хранение, загрузку в оперативную память и обработку файлов.

Функции и структуры операционных систем имеют различия в одно- и многопроцессорных вычислительных системах, многомашинных комплексах и вычислительных сетях. Соответственно этому операционная система должна обеспечить одно- или мультипрограммный режим работы ЭВМ, режим мультиобработки задач, совместное функционирование уровней и подсистем САПР через специальную мониторную систему.

Операционная система типа Windows предоставляет следующие возможности для программистов:

1. **Независимость программ от внешних устройств.** DOS – программа может работать с аппаратными средствами компьютера (монитором, клавиатурой, принтером и т.д.) непосредственно, минуя DOS. Windows – программа может обращаться к внешним устройствам только через Windows. Это снимает с программиста проблему обеспечения совместимости с конкретными внешними устройствами, т.к. ее берет на себя Windows. Поэтому любая Windows – программа может работать с любым внешним устройством, если с ним может работать Windows. Программы (драйверы) для поддержки наиболее распространенных устройств входят в Windows, а для остальных устройств – поставляются вместе с этими устройствами.

2. **Наличие средств для построения пользовательского интерфейса.** В Windows входят все необходимые средства для построения пользовательского интерфейса: окон, меню, запросов, списков и т.д. При этом стиль пользовательского интерфейса практически стандартен и считается одним из лучших.

3. **Доступность всей оперативной памяти.** В отличие от MS DOS средства управления оперативной памятью Windows обеспечивают доступность всей оперативной памяти компьютера (а не только ее части), что облегчает создание больших программ.

Другие возможности: обмен данными между приложениями Windows, организация встроенных справочных программ и т.д.

Для пользователей Windows предоставляет следующие возможности:

1. **Единый пользовательский интерфейс.** Т.к. Windows предоставляет программисту все необходимые средства для создания пользовательского интерфейса (окон, меню и т.д.), то программисты пользуются ими, а не изобретают собственные средства. Вследствие этого пользовательский интерфейс Windows – программ в значительной степени унифицирован, и пользователям не требуется изучать для каждой программы новые принципы организации взаимодействия с этой программой.

2. **Многозадачность.** Windows обеспечивает возможность одновременного выполнения нескольких программ, переключения с одной задачи на другую, управления приоритетами выполняемых программ.

3. **Поддержка мультимедиа.** При подключении соответствующих устройств Windows может воспринимать звуки от микрофона, компакт – диска, изображения от цифрового фотоаппарата, цифровой видеокамеры или с компакт – диска, выводить звуки на колонки или в наушники, выводить на экран монитора движущиеся изображения.

Другие возможности: совместимость с DOS – приложениями, удобство поддержки устройств, поддержка масштабируемых шрифтов и т.д.

Для создания программ используются языки (среды, студии) программирования.

2.2.2 Прикладное программное обеспечение САПР

Прикладное программное обеспечение представляют пакеты прикладных программ (ППП) для выполнения различных проектных процедур. Оно разрабатывается на основе единого внутреннего представления графической и текстовой информации, единого входного языка, строится по модульному принципу и ориентировано на использование непрограммистом-проектировщиком.

Различают несколько типов ППП в зависимости от состава пакета. Пакеты прикладных программ **простой** структуры характеризуются наличием только обрабатывающей части — набора функциональных программ (модулей), каждая из которых предназначена для выполнения только одной проектной процедуры. Объединение нужных модулей осуществляется средствами операционной системы ЭВМ.

Пакеты прикладных программ **сложной** структуры и **программные** системы появились в результате развития прикладного программного обеспечения. В первых из них имеется собственная управляющая часть — **монитор**, во вторых, кроме того, **языковой процессор** с проблемно-ориентированным входным языком. Программные системы вместе с соответствующим лингвистическим и информационным обеспечением называют **программно-методическими комплексами САПР**.

Выделяют системный уровень разработки прикладного программного обеспечения, уровень прикладных программ и уровень подпрограмм (модулей).

Связи между отдельными программными модулями могут быть реализованы по управлению, информации, размещению и воздействию.

Связи модулей по управлению могут быть двух типов: последовательные связи между модулями без возврата в предыдущий модуль и иерархические связи с подчиненностью модулей различных уровней

Связи модулей по информации проявляются в передаче числовых массивов в несколько модулей пакета. Этот аспект взаимодействия модулей затрагивает проблемы построения информационного обеспечения САПР.

Связи модулей по размещению указывают группы модулей, одновременно размещаемых в оперативной памяти на различных этапах проектирования.

Связи модулей по воздействию отражают такие воздействия одних программ на другие, которые приводят к изменению самих программ, например воздействие языковых процессов на рабочие программы. Внутри рабочих программ связи модулей по воздействию стараются исключить.

К настоящему времени разработано большое количество пакетов прикладных программ САПР электрических и электронных средств. В качестве примеров можно привести ДИСП, САМРИС-2, СПАРС, АРОПС, КРОСС и др. Из зарубежных систем можно отметить пакеты MicroCAP, PSPICE, P-CAD, SPADE и др.

Значительное число этих пакетов ориентировано на автоматизацию проектирования печатных плат, цифровых и аналоговых интегральных схем, операционных усилителей, низкочастотных радиотехнических устройств. В то же время весьма недостаточно разработано пакетов программ проектирования радиочастотных, в том числе мощных устройств, радиоэлектронных средств СВЧ, пакетов, посвященных комплексному построению и интеграции радиочастотных средств, включающих в себя как усилители, так и пассивные радиочастотные устройства, вплоть до антенн и СВЧ-устройств.

Развитие программного обеспечения САПР требует все более значительных затрат высококвалифицированного труда. Стоимость многих промышленных САПР составляет миллионы долларов. Поэтому актуальной становится разработка САПР второго порядка, или САПР САПРов. Пока таких систем еще не существует, но прогресс в этом направлении имеется. В отличие от традиционных САПР в таких системах результат имеет нематериальный (информационный) характер. Различие результатов вызвано различными языками описания предметных областей: в одном случае — чертежи, схемы, устройства, а в другом — программа проектирования. Однако и в том, и в другом случае возможен единый системный методологический подход к проектированию; становится актуальным создание и развитие банка инженерных знаний, необходимых для проектирования.

2.3 Лингвистическое обеспечение САПР. Языки программирования

Лингвистическое обеспечение — совокупность языков, используемых в процессе разработки и эксплуатации САПР.

Под «языком» понимается любое средство общения, любая система символов и знаков для представления и обмена информацией.

Лингвистическое обеспечение образуется следующими языками:

- программирования;
- управления;
- проектирования.

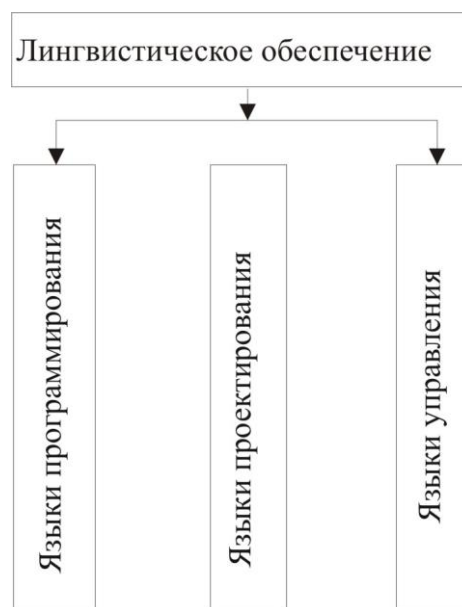


Рисунок 1 – Лингвистическое обеспечение

Языки программирования необходимы для создания программного обеспечения при разработке САПР. В принципе языки программирования относят и к программному обеспечению САПР. Здесь мы их подробно рассматривать не будем, информация о них приведена в специальной литературе. Напомним лишь, что к наиболее распространенным языкам программирования относятся Pascal, Fortran, Basic, Си (различных версий). В настоящее время на их базе разработаны и повсеместно используются среды программирования такие, как, соответственно, Delphi, Visual Fortran, Visual Basic, Visual Си (также различных версий).

Существуют различные уровни языков программирования: высокие, более удобные для пользователя, и более низкие, близкие к машинным языкам.

Программа, записанная на некотором языке программирования высокого уровня, называется исходной. Прежде чем исходная программа будет исполнена, она должна быть преобразована в машинную форму, соответствующую ЭВМ данного типа. Подобные преобразования осуществляются специальными программами, называемыми **языковыми процессорами**.

Основные типы языковых процессоров — трансляторы и интерпретаторы; соответственно преобразования программ называют трансляцией и интерпретацией.

Трансляцией называют перевод всего текста программы на исходном языке (исходной программы) в текст на объектном языке (объектную программу). Если исходный язык является языком высокого уровня, а объектный — машинным, то транслятор называют **компилятором**. Если исходный язык — машинно-ориентированный (в автокоде), а объектный — машинный, то транслятор называют **ассемблером**. Если исходный и объектный языки относятся к одному уровню, то транслятор называют **конвертером**.

По методу трансляции (компиляции) сначала исходная программа переводится на машинный язык, а затем скомпилированная рабочая программа исполняется.

При **интерпретации** перевод исходной программы в рабочую совмещены во времени; очередной оператор исходной программы анализируется и тут же исполняется.

В большинстве случаев применение трансляторов приводит к меньшим затратам машинного времени, но к большим затратам машинной памяти, чем при интерпретации.

Совокупность языка программирования и соответствующего ему языкового процессора называют **системой программирования**.

Класс машинно-зависимых языков представлен ассемблером (макроассемблером). Он относится к языкам низкого уровня и используется для написания программ, явно использующих специфику конкретной аппаратуры.

К машинно-ориентированным языкам относится язык **СИ** (разработан в 1972 г.). В нем объединяются достоинства низкоуровневых возможностей ассемблеров и мощных выразительных средств языков программирования высокого уровня. Язык СИ является одним из претендентов на роль основного языка программирования в САПР и ориентирован на разработку системных программ. Он, в частности, послужил главным инструментом для создания операционных систем для ПЭВМ UNIX и MS DOS.

Язык **Фортран** является первым универсальным языком высокого уровня (с 1954 г.). Фортран — самый примитивный из распространенных алгоритмических языков общего назначения. Наиболее эффективен при численных расчетах, прост по структуре и эффективен при выполнении программ. Несмотря на свои недостатки, этот язык получил большое распространение при разработке прикладных программ для решения научных задач. Наиболее популярная в настоящее время версия этого языка — Фортран-77.

Идеи Фортрана получили развитие в языке PL/I (создан в 1964 г.) В нем сделана попытка преодолеть некоторые недостатки, свойственные языкам для больших ЭВМ и использованы идеи структурного программирования. В настоящее время имеются различные версии этого языка: PL/M, PL/Z, PL/65 и др. Как язык программирования PL/M, в частности, значительно уступает конкурирующим с ним языкам Паскаль и Модула-2.

Язык **Паскаль** является одним из наиболее популярных языков программирования и используется для разработки системных и прикладных программ, в частности для персональных ЭВМ. Язык Паскаль создан вначале исключительно для учебных целей и изящно реализовал большинство идей структурного программирования. Достоинства языка оказались столь значительными, что он приобрел огромную популярность для самых различных приложений. В частности, компилятор Turbo Pascal, снабженный интерактивным редактором, позволяет создавать достаточно сложное программное обеспечение — системы управления базами данных, графические пакеты и др.

Развитием Паскаля являются языки Модула-2 (в Европе) и **Ада** (в США). Язык Модула-2 обладает лучшими средствами для обработки больших программных комплексов и позволяет более эффективно использовать особенности аппаратуры. Таким образом, этот язык призван заполнить ниши между Паскалем и СИ. По оценке специалистов, язык Модула-2 через несколько лет станет наиболее популярным среди всех языков программирования.

Язык Ада можно назвать наиболее универсальным среди созданных языков. Однако трансляторы с этого языка пока не получили достаточного распространения.

Язык *Алгол* — общепризнанный язык для публикации алгоритмов решения научных задач, построен на четких и полных определениях. Для Алгола характерны строгие, но негибкие структуры данных и программ. Алгол труден для реализации на большинстве ЭВМ, поэтому используются неполные варианты языка или его расширения.

Язык *Кобол*, разработанный для решения экономических задач, широко распространенный на больших и средних ЭВМ прошлых лет, на персональных ЭВМ почти не используется. На ПЭВМ имеются интегрированные системы, базы данных и другие типы прикладных систем, используемые в задачах экономического и управленческого характера.

Самыми распространенными на ПЭВМ являются различные версии языка Бейсик, простота которого делает его превосходным средством для начинающих программистов. В языке встроены удобные функции для работы с экраном дисплея, клавиатурой, внешними накопителями, принтером, каналами связи. Это позволяет относиться к Бейсику как к продолжению аппаратуры ПЭВМ. Системы Бейсика работают в режиме интерпретации, что способствует сокращению характерного цикла в работе программиста: составление программы — пробное исполнение — исправление ошибок — повторное исполнение.

Бейсик наряду с Паскалем принят во многих учебных заведениях как базовый язык для изучения программирования.

Язык *АПЛ* применяется для обработки структурных данных (векторов, матриц) и использует иероглифическую запись программных текстов. Из-за большого числа иероглифов (около 100) его иногда называют китайским Бейсиком.

К *классу проблемно-ориентированных* языков можно отнести Лого, CPSS, Форт и Смолток.

Язык *Лого* — диалоговый процедурный язык, реализованный на принципе интерпретации и работающий со списками, текстами, графическими средствами и др. Язык очень перспективен для обучения, создания электронных игрушек и т.д.

В системах программирования GPSS легко описывается и исследуется класс моделей массового обслуживания.

Развитием проблемно-ориентированных языков является объектно-ориентированный подход (языки Смолток, Форт, Модула и Ада). Отличительными особенностями таких языков являются модульность построения процедур, абстракция данных, динамическая связка программ, что позволяет отказаться от перекомпилирования всей программы при внесении изменений в отдельные модули, и использование механизма наследования иерархического типа.

К недостаткам таких языков относится некоторая замедленность выполнения программ из-за их динамической связи и сложность трансляторов.

Язык *Смолток* предназначен для решения нечисловых задач при построении систем искусственного интеллекта. В языке *Форт* применены структурное программирование и очень компактный машинный код.

Для разработки систем искусственного интеллекта также используются *функциональные языки Лисп, Пролог и Снобол*. Эти языки ориентированы на обработку символьной информации, требуют больших массивов данных и стали применяться в ПЭВМ в связи с появлением дешевой полупроводниковой памяти, позволяющей довести объем ОЗУ до нескольких Мбайт. Языки этого класса относятся к так называемым языкам представления знаний.

Язык *Лисп* применяется для программирования интеллектуальных задач — общение на естественном языке, доказательство теорем, принятие решений и т.п.

Язык *Пролог* приобрел в последние годы большую популярность в связи с японским проектом создания вычислительных систем пятого поколения. Он предназначен для создания широкого класса систем искусственного интеллекта, в том числе и персональных экспертных систем.

2.4 Лингвистическое обеспечение САПР. Языки управления и проектирования

Языки управления служат для управления ЭВМ, периферийными устройствами. Это операционная система Windows, драйверы принтеров и т.д. Эти языки также относят и к программному обеспечению САПР.

При использовании САПР приходится решать задачи не только вычислительного характера и обработки данных, но и автоматизировать описание объектов, процессы ввода, вывода и редактирования данных, ввода графических изображений, схем, чертежей и т.п. Для этой цели служат языки проектирования.

Языки проектирования ориентированы на пользователей – проектировщиков и предназначены для эксплуатации САПР. На них мы и остановимся более подробно. Эта группа языков делится на:

- входные;
- внутренние;
- сопровождения;
- промежуточные;
- выходные.

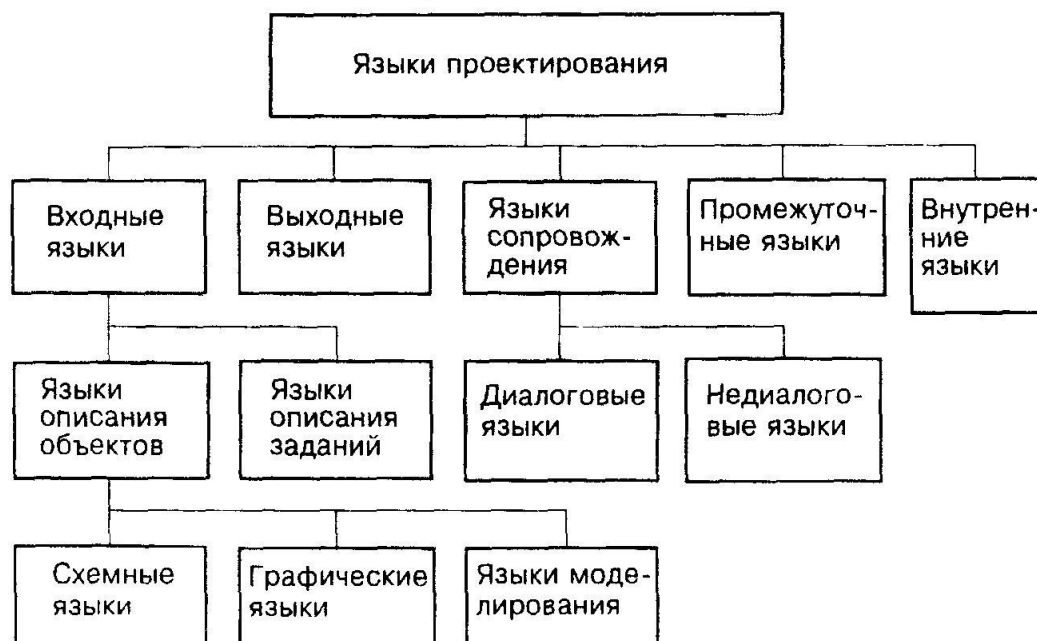


Рисунок 2 - Классификация языков проектирования

Входные языки являются средством взаимодействия конечного пользователя с САПР, например, в ходе подготовки и ввода исходных данных или формирования проблемы. С помощью их задается исходная информация об объектах и целях проектирования. Во входных языках можно выделить две части: непроцедурную, служащую для описания структур объектов, и процедурную, предназначенную для описания заданий на выполнение проектных операций и процедур. Этим частям соответствуют языки описания объектов и языки описания заданий. Разновидности первых; схемные, графические языки и языки моделирования.

Схемные языки применяются для описания электрических и электронных схем и содержат данные об элементах схем и их связях друг с другом.

Графические языки используются для ввода чертежей, геометрических изображений, деталей и т.п. Задание геометрии деталей осуществляется различными способами: координатным, структурно-символическим (методом типовых графических элементов), аналитическим (математическими уравнениями поверхностей и линий) и рецепторным (мозаичным). При этом используются графические примитивы и элементарные графические операции. Разработаны специальные системы графического программирования.

Языки моделирования близки к алгоритмическим языкам и применяются для описания процессов в моделируемом объекте.

Внутренние языки обычно скрыты от рядового пользователя и служат для представления информации, передаваемой между различными подсистемами САПР и ЭВМ. Они устанавливают единую форму предъявления данных (текстовой и графической информации) в памяти ЭВМ по подсистемам САПР. Принимаются определенные соглашения об интерфейсах отдельных

программ, что делает САПР открытой по отношению к новым элементам программного обеспечения.

В качестве примера современного языка проектирования можно указать язык VHDL (VHSIC — hardware description language) — язык описания аппаратуры на базе сверхвысокоскоростных интегральных схем. Этот язык принят в качестве стандарта как инструментальное средство автоматизации проектирования СБИС, ориентированное на методологию нисходящего проектирования. Он является достаточно универсальным, чтобы охватить все аспекты проектирования изделий области цифровой электроники.

Языки *сопровождения* служат для непосредственного общения пользователя с ЭВМ и применяются для корректировки и редактирования данных при выполнении проектных процедур. В диалоговых режимах работы с ЭВМ средства языков входного, выходного и сопровождения тесно связаны и объединяются под названием диалогового языка. Современные *диалоговые* языки широко используют средства машинной графики (графический диалог). Диалог с ЭВМ может быть пассивным, тогда инициатор диалога — система, и от пользователя требуются только простые ответы, и активным при двусторонней инициативе диалога. Наиболее распространенная форма пассивного диалога — это система встроенных, в том числе иерархических, директивных меню. *Недиалоговые* системы языков сопровождения ориентированы на пакетный режим работы ЭВМ.

Промежуточные языки используются для описания информации в системах поэтапной трансляции исходных программ. Введение таких языков облегчает адаптацию программных комплексов САПР к новым входным языкам, т.е. делает комплекс открытым по отношению к новым составляющим лингвистического обеспечения.

Выходные языки обеспечивают оформление результатов проектирования в удобном для разработчика виде. Возможные формы представления — таблицы, графики, чертежи, диаграммы, текстовые сообщения. При этом необходимо обеспечить эффективность понимания разработчиком проектных результатов (желательно в графической форме), соблюдение требований стандартов при формировании подлинников конструкторской, программной и технологической документации.

Такое деление языков проектирования можно назвать классическим. В различных САПР они могут применяться с различной степенью развернутости и в различном исполнении. В одних САПР, реализующих, например, принцип синтеза технологических процессов, информация о детали для автоматического проектирования варианта изделия вводится единовременно. В других подобных системах применяется диалоговое проектирование (диалоговый синтез), и в них информация о детали вводится постепенно по ходу проектирования технологического процесса.

Так или иначе место языков проектирования на различных этапах переработки информации в САПР ТП (один из вариантов) показано на рисунке 3.

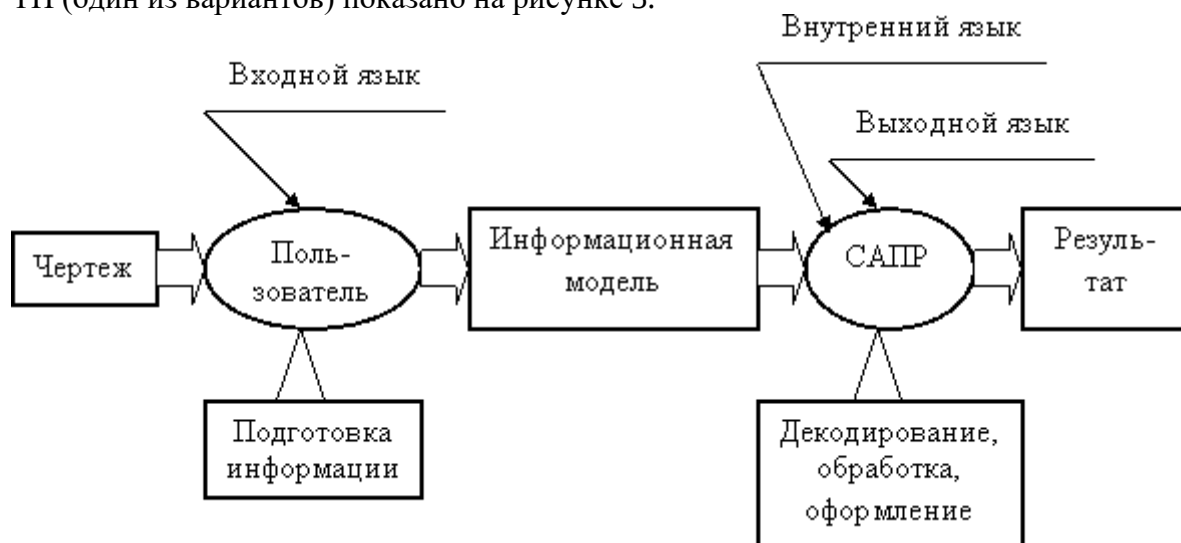


Рисунок 3 - Преобразование информации в САПР

2.5 Информационное обеспечение САПР. Определения и понятия

При автоматизированном проектировании для удовлетворения потребности прикладных программ и подсистем САПР, а также запросов пользователей в диалоговом режиме возникает необходимость в машинном представлении данных.

Поэтому в комплекс средств автоматизированного проектирования входит информационное обеспечение.

Главной целью создания информационного обеспечения САПР является разработка информационной системы, позволяющей правильно и быстро решать проектные задачи. Это может быть достигнуто своевременной выдачей источнику запроса полной и достоверной информации для выполнения определенной части проектно-конструкторского процесса.

Информационный фонд САПР – это совокупность всех необходимых для функционирования САПР данных.

Информационное обеспечение САПР – это совокупность информационного фонда и средств его ведения, т.е. средств создания, реорганизации данных и обеспечения доступа к ним с использованием ЭВМ.

В состав информационного фонда входят:

- нормативно – справочная информация (сведения о заготовках, типовых маршрутах обработки, станках, инструментах и т.д.);
- записываемые временно данные, которые являются результатом функционирования одной подсистемы САПР и которые затем вводятся в другую подсистему;
- программные модули отдельных подсистем, подпрограммы для разработки управляющих программ для станков с ЧПУ;
- чертежи инструментов и приспособлений, операционные эскизы;
- шаблоны для ввода информации и оформления документов, например, технологических карт и т.п.

Основные требования к информационному обеспечению САПР следующие:

1. **Наличие необходимой информации** для обеспечения как автоматизированных, так и ручных процессов проектирования.

2. **Возможность хранения и поиска информации**, представляющей результат ручных и автоматизированных процессов проектирования.

3. **Достаточный объем хранилищ информации.** Структура системы должна допускать возможность наращивания емкости памяти вместе с ростом объема информации, подлежащей хранению. Одновременно необходимо обеспечить компактность хранимой информации и минимальное изнашивание носителей информации.

4. **Достаточное быстродействие** системы информационного обеспечения.

5. **Возможность быстрого внесения изменений** и корректировки информации, доведения этих изменений до потребителя, а также получение твердой копии документа.

При создании информационного обеспечения САПР основная проблема заключается в преобразовании информации, необходимой для выполнения проектно-конструкторских работ над определенным классом объектов, в форму, приемлемую и наиболее рациональную для машинной обработки, и выводе информации на ЭВМ в виде, удобном для восприятия человеком. Множество данных, которые потенциально могут использоваться при функционировании САПР или служить запоминаемым результатом ее работы, образуют информационную базу данных (БД) системы. Типовыми группами данных информационного обеспечения автоматизированного проектирования являются классификаторы и таблицы соответствия для них, научно-техническая и расчетно-проектная (оперативная) информация

Информационное обеспечение САПР взаимодействует с проектными модулями через специально организуемый интерфейс, который защищает проектные программные модули от влияния специфики программной реализации информационной системы, поддерживая тем самым независимость проектных операций от вида представления информации в базе данных. В функции этого интерфейса входит также согласование и сопряжение информационной системы и проектных модулей:

- по форматам записей (*информационный* аспект),
- по обозначениям данных (*содержательный* аспект),
- по программным средствам, языкам программирования и т. п. (*программный* аспект).

Сложность разработки базы данных обусловлена тем, что формирование ее структуры возможно только после разработки алгоритмов проектирования. Степень разработки алгоритмов должна быть доведена до машинной реализации, так как структура базы данных должна учитывать специфику процесса автоматизированного проектирования. Но для разработки пакета прикладных программ (ППП) необходимы сведения о структуре базы данных. Следовательно, информационное обеспечение и специальное программное обеспечение САПР должны создаваться параллельно. Информация, используемая при проектировании, может быть разделена на *статическую* и *динамическую*.

Статическая информация характеризуется сравнительно редкими изменениями. К этой информации следует отнести данные ТЗ на проектирование и справочные данные, имеющие большой объем. Формирование, загрузка и корректировка справочных данных осуществляется исключительно администратором базы данных, т. е. системным программистом, формирующим базу данных. Администратор базы данных поддерживает непосредственный контакт со службой нормализации и стандартизации проектной организации. Объем данных ТЗ на проектируемый объект значительно меньше объема справочных данных, но круг лиц, имеющих право вносить изменения в ТЗ, должен быть еще более ограничен, чем круг лиц, имеющих право корректировать справочные данные.

Динамическая информация состоит из данных, накапливаемых для выполнения определенных операции проектирования (промежуточные данные), и данных, представляющих собой результат проектирования при выполнении данных операций. Промежуточные данные постоянно меняются при функционировании САПР. Вносить изменения в варианты проектных решений имеет право только конструктор-исполнитель и его руководитель.

Информация, используемая при проектировании, по виду ее представления может быть подразделена на:

- документальную,
- иконографическую,
- фактографическую.

Документальная информация — это метаинформация. Она представляет собой поисковый образ документа, находящегося в базе данных. При необходимости может быть выдана совокупность документов, удовлетворяющих поисковому образу. В САПР информация такого вида широко используется для нахождения сведений об аналогах объекта проектирования, о патентах и авторских свидетельствах, методике проектирования и расчетов, результатах испытания и т. п.

Информация, которая содержится в изображениях документов (чертежи, фотографии и т. д.), в идентичной форме представления называется *иконаграфической*. Для ее хранения используют специальные носители (микрофиши, рулонные микрофильмы и т. д.). В современных САПР этот вид информации служит для хранения больших объемов графической информации, поиск которой может осуществляться с помощью сопровождающей ее документальной информации.

Основу базы данных САПР составляет *фактографическая* информация. Она представляет собой числовые и буквенные справочные данные о материалах, ценах, комплектующих изделиях, о спроектированных в САПР объектах и т. п. Сюда же относятся данные, необходимые для выполнения расчетов: коэффициенты, таблицы, аппроксимированные графические зависимости и т. д.

В настоящее время различают два вида автоматизированных информационных систем САПР — *банки данных* и *информационно-поисковые системы* (ИПС). Эти системы различаются видом хранимой и обрабатываемой информации и информационным языком, с помощью которого осуществляется описание данных и манипуляции с ними. Эти различия накладывают определенные ограничения на организацию информации в системе (структуры данных, форматы, связи, доступ и т. д.) и на программную реализацию.

Функционирование информационной системы обеспечивается *программно-техническими средствами* (машинная организация) и *средствами внемашинной организации*. Программно-технические средства информационных систем — это, как правило, специальные ППП, которые обеспечивают накопление (ввод, изменения, модификацию), хранение и поиск информации. К средствам внемашинной организации данных в информационных системах относятся: система классификации и кодирования информации; система ведения информационных массивов (входные формы и таблицы, оперативные документы на изменение информации и т. д.); методические инструментальные материалы для системного персонала (службы администрации).

Проектирование, организацию функционирования и развитие информационной системы обеспечивает системный персонал. В информационно-поисковых системах САПР хранится и обрабатывается, как правило, документальная информация. Информационный язык в ИПС — это ограниченный (нормированный) естественный язык, с помощью которого описывают содержание документальных источников информации (статей, книг, стандартов и т. д.) в виде набора понятий, отражающих основное содержание документов. В информационно-поисковых системах ППП не имеет специального названия, и говорят о ППП для ИПС. Единицей хранения информации в ИПС является описание конкретного документа. Прообразами накапливаемых в системе описаний документов служат некоторые внешние первичные документы, содержащие информацию, используемую в процессе автоматизированного проектирования. Такими первичными документами могут быть отчеты по научным и конструкторским работам, патенты, справочники, статьи, каталоги и т. д. С точки зрения пользователя, каждое описание документа представляет собой краткую библиографию источника информации (автор, заглавие, название источника, год выпуска, издательство, аннотация или реферат). Для обеспечения взаимодействия пользователей и ИПС служит нормативный (фиксированный) словарь понятий, с помощью которого можно описывать содержание, как документов, так и запросов. Такой словарь называется тезаурусом. Тезаурус является моделью системы понятий предметной области. Поэтому документ, записанный в память ЭВМ, кроме библиографии, имеет поисковые признаки или поисковый образ, который составляется по определенным правилам с помощью понятий тезауруса. Запросы к системе формулируются также с помощью тезауруса по определенным правилам. Совокупность правил перевода с естественного языка на язык системы, и тезаурус образуют информационно-поисковый язык системы. Совокупность документов в памяти ЭВМ образует последовательный массив (файл). Поиск информации в системе осуществляется путем сравнения понятий поискового образа документа и понятий запроса. При их полном или частичном совпадении (в зависимости от критерия выдачи) документ считается релевантным, т. е. соответствующим запросу. Но при такой последовательной организации информации поиск и сравнение со всеми поисковыми образами заняли бы много времени. Для более эффективной организации информации в систему вводят инверсный (поисковый) массив, в котором каждому понятию тезауруса поставлен в соответствие набор номеров документов, в которых это понятие встречается. К функциям ППП для ИПС относятся: ведение и использование информационно-поискового языка; ввод, накопление и изменение информации; поддержка инверсного массива; поиск и выдача информации по запросам. ИПС описанного выше типа называются документальными ИПС.

Существует ИПС фактографического типа. Они отличаются тем, что в них хранение и поиск осуществляется не по набору понятий, а по набору признаков каких-либо объектов, т. е. кроме тезауруса в системе предусмотрен еще и специальный классификатор признаков объектов. ИПС фактографического типа более близки по своей организации к банкам данных.

Наиболее высокой формой организации информационного обеспечения больших САПР являются банки данных. Они представляют собой **проблемно-ориентированные информационно-справочные системы, которые обеспечивают ввод необходимой информации, автономное от конкретных задач ведение и сохранение информационных массивов, и выдачу необходимой информации по запросу пользователя или программы**. В банках данных используется информация фактографического вида. Информационный язык — совокупность двух языков: языка описания структуры данных и языка манипулирования данными. Пакетом прикладных программ этих информационных систем является система управления базами данных (СУБД), которая обеспечивает работу с информационной базой, организованной в виде структуры данных. По этой заранее сформированной структуре (модели) данных

производится их описание, хранение и поиск. В СУБД описание структуры информации принято называть схемой.

В зависимости от уровня представления информации различают следующие типы схем:

- **концептуальный** (общее представление об информационной базе предметной области);
- **внешний** (представление информации со стороны пользователей или задач; при большом числе задач их представления могут пересекаться); внешних схем бывает несколько;
- **внутренний** (представление информации в базе данных, т. е. на физических носителях — магнитных дисках). Среди всех перечисленных уровней представления информации концептуальный уровень занимает особое место. Он связывает внешний уровень с внутренним и обеспечивает их относительную независимость, т. е. возможность изменения внешней схемы при неизменной внутренней и наоборот.

Роль концептуального уровня состоит, прежде всего, в том, что на нем отображается та часть общей информационной базы, которая должна быть представлена в виде базы данных. Концептуальный уровень обеспечивает независимость СУБД от конкретного вида ЭВМ. Формализованное описание информационной базы на концептуальном уровне, как правило, осуществляется в терминах конкретной СУБД. Но на начальном этапе проектирования информационной базы еще неизвестно, какая СУБД удовлетворяет требованиям создаваемого банка данных. Поэтому вводится дополнительный уровень, на котором можно было бы задать описание предметной области, не касаясь вопросов реализации, т. е. использования конкретной СУБД. Его называют информационно-логическим (инфологическим).

Информационно-логическая модель определяет информационные потребности проектируемой системы и характеристики информационной базы. СУБД выполняет следующие основные функции: определение баз данных (т. е. описание концептуального, внешнего и внутреннего уровней схем); запись данных в базу; организацию хранения данных (изменение, дополнение, реорганизация данных); представление доступа к данным (поиск и выдача данных). Дополнительные функции (диалог, многопользовательский режим и т. д.) могут быть реализованы в виде пакетов программ окружения СУБД. Для определения данных и доступа к ним в СУБД имеются языковые средства (специальные языки). Так, определение данных (описание концептуальной, внутренней и внешней структур) обеспечивается с помощью языка определения данных. Функции доступа к данным реализуются с помощью языка манипулирования данными и языка запросов. По типу поддерживаемых структур различают следующие виды СУБД: иерархический, сетевой и реляционный.

2.6 Информационное обеспечение САПР. Ведение и организация информационного фонда на ЭВМ

2.6.1 Ведение информационного фонда на ЭВМ

Известны три подхода к организации информационного фонда:

Размещение данных непосредственно в теле программы – см. рисунок 4.

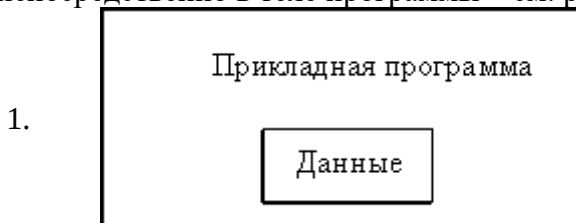


Рисунок 4 - Организация информационного фонда с размещением данных в теле программы

Запись данных в файлы – см. рисунок 5.

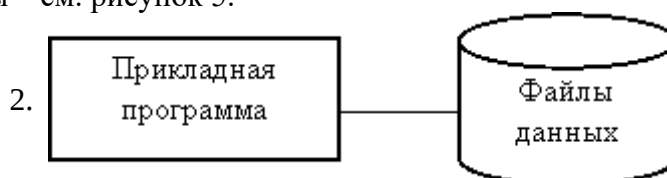


Рисунок 5 - Организация информационного фонда с записью данных в файлы

Использование баз данных – см. рисунок 6.



БнД – банк данных; БД – база данных; СУБД – система управления базой данных

Рисунок 6 - Организация информационного фонда с использованием базы данных

В принципе все три подхода имеют право на существование при обоснованном и квалифицированном их использовании в каждом конкретном случае.

При файловой организации информация записывается на винчестер отдельно от прикладной программы. Это обеспечивает относительную независимость прикладной программы от данных, т.е. исключает изменение программы в случае обновления данных. Если данные используются только конкретной прикладной программой, то такой подход вполне приемлем. Если нет, то очевиден следующий недостаток такого подхода. Часто одни и те же данные используются различными прикладными программами, в которых они имеют различную структуру и представлены по – разному. Это приводит к их необоснованному дублированию (избыточности) на диске.

Имеется еще один недостаток, который относится в целом ко второму подходу. К данным, рассредоточенным по десяткам файлов и организованным так, чтобы удовлетворять только запросам конкретных прикладных программ, нельзя обращаться пользователю, например, в диалоговом режиме.

2.6.2 Организация информационного фонда на ЭВМ с использованием баз данных

Организация информационного фонда на ЭВМ с использованием баз данных (БД) применяется во многих современных САПР ТП.

База данных – совокупность структурированных данных, используемых многими прикладными программами и хранящихся с минимальной избыточностью.

Система управления базой данных (СУБД) – программный комплекс, обеспечивающий создание структуры, ввод, модификацию, удаление и поиск данных.

Иногда используется понятие **банка данных (БнД)**, под которым понимается совокупность БД и СУБД.

Самой распространенной в настоящее время является СУБД Microsoft Access, которая является одним из продуктов пакета Microsoft Office XP.

2.6.2.1 Основные требования, предъявляемые к базам данных

К базам данных предъявляется ряд требований, среди которых можно выделить следующие основные требования:

1. **Минимальная избыточность.** Каждый элемент данных вводится в БД один раз и хранится в единственном экземпляре. При вводе данных СУБД выполняет проверку на дублирование. Этим достигается экономия внешней памяти и надежность информации.

2. **Независимость.** Модификация данных и изменения, вносимые в их структуру в связи с появлением новых пользователей и новых запросов, не должны отражаться на программах пользователей.

3. **Целостность данных:**

- **логическая** (СУБД должна защищать БД от некорректных действий пользователей путем восстановления состояния БД на момент, предшествующий ошибочной операции);
- **физическая** (защита носителей информации – дисков – от сбоев путем дублирования, например, двумя параллельно работающими зеркальными дисками).

4. **Секретность.** Пользователи должны работать только с теми данными (фрагментами данных), к которым им разрешен доступ.

2.6.2.2 Основные понятия и основы проектирования баз данных

Начнем с определения понятия «данные». Итак, **данные** – это информация, представленная в определенной форме, пригодной для хранения и обработки на ЭВМ. Можно дать и другое определение: **данные** – это представленные в цифровом виде сведения о некоторых объектах окружающего нас мира (об объектах интересующей нас предметной области).

При создании любой БД разрабатывается **модель данных**. При этом интересующая пользователей БД информация существует в двух представлениях:

1. Логическое представление данных.
2. Физическое представление данных на носителе информации (диске).

Логическое представление отражает структуру данных. Модель не содержит конкретных значений. Она только описывает их структуру. В дальнейшем структура остается неизменной, а данные могут меняться при вводе и редактировании информации в БД.

Для определения модели используются следующие понятия:

- объект;
- атрибут;
- экземпляр;
- ключ.

В дальнейшем мы укажем на соответствующие этим понятиям понятия, используемые при описании физического представления данных и понятия, принятые в СУБД Microsoft Access.

Объект представляет собой то, о чем накапливается информация в БД, например «диод», «резистор» и т.д.

Атрибуты – это интересующие пользователя характеристики объекта. Например, для объекта «диод» - это «обозначение», «параметр 1», «параметр 2» и т.д.

Экземпляр объекта – совокупность значений атрибутов, описывающих конкретную его реализацию. В нашем случае это строка таблицы.

Ключ – это атрибут, значение которого однозначно определяет экземпляр. Так в БД по диодам ключом может служить атрибут «обозначение», т.к. значение этого атрибута не дублируется ни в одной строке (экземпляре). Другие атрибуты не могут быть ключом, потому что могут принимать одинаковые значения для разных экземпляров. Например, вполне возможны два диода с одинаковым значением «параметра 1», хотя и разного исполнения.

При описании физического представления данных, а также в терминологии СУБД Microsoft Access понятию «атрибут» соответствует понятие «поле» (столбец таблицы). Понятию «экземпляр» соответствует понятие «запись» (строка таблицы). Объекту соответствует фрагмент файла данных или файл данных целиком.

База данных, состоящая из набора связанных между собой двумерных (плоских) таблиц, называется **реляционной базой данных**. Данные в этих таблицах организованы таким образом, чтобы обеспечить объединение разнородной информации, исключить ее дублирование, а также предоставить оперативный доступ к имеющимся сведениям и эффективное сопровождение базы данных в целом.

Реляционные СУБД используют реляционную модель данных, предложенную в 1970 году Э.Ф.Коддом. Если говорить упрощенно, то Кодд показал, что набор двумерных таблиц при соблюдении определенных ограничений может быть использован для хранения данных об объектах реального мира и моделирования связей между ними. В терминологии Кодда такие таблицы называются **отношениями** (англ. **relation**), вот почему подобная база данных называется реляционной.

В реляционной базе данных, в частности, реализованной в СУБД Microsoft Access, для однозначного распознавания экземпляра объекта подобно приведенному выше понятию «ключ» вводится уникальный идентификатор – **первичный ключ**. Первичный ключ – это уникальная характеристика для каждой записи в пределах таблицы. Первичный ключ таблицы помимо однозначной идентификации записей позволяет реализовать и связи между таблицами. Благодаря связям информация из одной таблицы становится доступной для другой. Связи устанавливаются за счет того, что в разных таблицах присутствуют поля с одинаковыми значениями.