1. **10.3. Лингвистическое обеспечение САПР**

Это совокупность языков, используемых в процессе разработки и эксплуатации *САПР* для обмена информацией между человеком и ЭВМ. Термином "язык" в широком смысле называют любое средство общения, любую систему символов или знаков для обмена информацией.

*Лингвистическое обеспечение* *САПР* состоит из *языков программирования*, проектирования и управления.

*Языки программирования* служат для разработки и редактирования системного и прикладного программного обеспечения *САПР*. Они базируются на *алгоритмических языках* - наборе символов и правил образования конструкций из этих символов для задания алгоритмов решения задач ( *языки программирования* подробно изучаются в курсе "*Информатика*").

*Языки проектирования* - это проблемно-ориентированные языки, служащие для обмена информацией об объектах и процессе проектирования между пользователем и ЭВМ.

*Языки управления* служат для формирования команд управления технологическим оборудованием, устройствами *документирования*, *периферийными устройствами* ЭВМ.

Существуют различные уровни *языков программирования*: высокие, более удобные для пользователя, и 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 совмещен 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*.

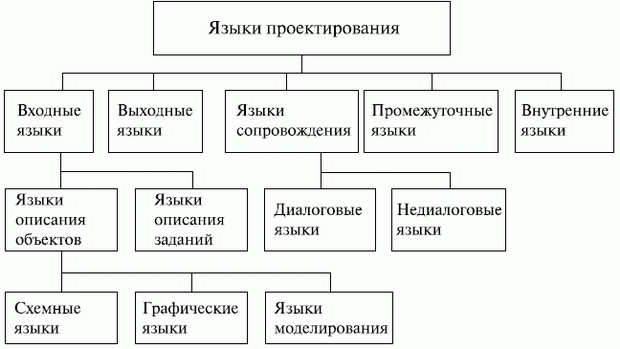
*Класс* машинно-зависимых языков представлен *Ассемблером* (макроассемблером). Он относится к *языкам низкого уровня* и применяется для написания программ, явно использующих специфику конкретной аппаратуры.

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 ЭВМ *UNIX* и MS *DOS*.

Классификация *языков проектирования* приведена на [рис. 10.1](http://www.intuit.ru/studies/courses/650/506/lecture/11505?page=1#image.10.1).

*Языки проектирования* делят на: входные, выходные, сопровождения, промежуточные и внутренние.

*Входные*языки служат для задания исходной информации 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*.



**Рис. 10.1.**Классификация языков проектирования

Этим частям соответствуют языки описания объектов и языки описания заданий. Разновидности первых: схемные языки, 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). Разработаны специальные системы графического программирования.

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

*Выходные языки*используются для представления 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.

*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*,когда 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*.

*Внутренние языки*устанавливают единую форму представления 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 в области цифровой электроники.