

Инструментальное программное обеспечение

- 
- 1. Языки программирования**
 - 2. Методология разработки программных продуктов**
 - 3. Трансляторы и их виды**
 - 4. Системы программирования**
 - 5. Жизненный цикл программного обеспечения**

1. Языки программирования

Язык программирования - это система обозначений, служащая для точного описания программ или алгоритмов для ЭВМ.

По степени детализации предписаний:

- Языки низкого уровня:
 - ✓ машинные;
 - ✓ машинно-ориентированные (ассемблеры);
- Языки высокого уровня:
 - ✓ машинно-независимые .

Основные компоненты алгоритмического языка:

- алфавит,
- синтаксис,
- семантика.

Преимущества языков высокого уровня перед машинными:

- ❑ алфавит языка значительно шире алфавита машинного языка;
- ❑ набор операций, допустимых для использования, не зависит от набора машинных операций;
- ❑ формат предложений гибок и удобен для использования;
- ❑ требуемые операции задаются с помощью общепринятых математических обозначений;
- ❑ данным присваиваются индивидуальные имена, выбираемые программистом;
- ❑ более широкий набор типов данных.

Основные компоненты алгоритмического языка:

- ▣ **алфавит** - фиксированный набор основных СИМВОЛОВ;
- ▣ **синтаксис** - правила построения фраз;
- ▣ **семантика** - смысловое значение предложений.

Классификация языков высокого уровня:

1. *процедурные* - предназначены для однозначного описания алгоритмов :
 - ✓ *Fortran* - предназначен для решения задач;
 - ✓ *Basic* – для решения небольших вычислительных задач в диалоговом режиме;
 - ✓ *PL/1, Pascal, C, C+ , Modula, Ada* - универсальные языки

-
2. *логические* – ориентированы на формализованное описание задачи с тем, чтобы решение следовало из составленного описания. Указывается что дано и что требуется получить, и поиск решения задачи возлагается на ЭВМ

Lisp, Prolog

3. *объектно-ориентированные* – позволяют связать данные с обрабатывающими процедурами в единое целое - объект.

Object Pascal, C++, Java и др.

2. Методология разработки программных продуктов

Выделяют:

- *методы традиционного (неавтоматизированного) проектирования:*
 - ✓ алгоритмическое программирование;
 - ✓ структурное проектирование;
 - ✓ объектно-ориентированное проектирование ;
- *методы автоматизированного проектирования (CASE-технология и ее элементы).*

а. Структурное проектирование

Типичными методами структурного проектирования являются:

- ▣ *структурное программирование* - структура программы отражает структуру задачи, алгоритм решения ясно виден из текста программы, состоящей из подпрограмм — совокупностей операторов, выполняющих нужное действие и не зависящих от других частей исходного кода ;
- ▣ *нисходящее проектирование* - проектирование и разработка приложений сверху вниз ;

-
- ▣ *модульное программирование* - основано на понятии модуля — логически взаимосвязанной совокупности функциональных элементов, оформленных в виде отдельных программных модулей ;
 - ▣ *событийно-ориентированное программирование* - идеология системы Windows .

в. **Объектно-ориентированное программирование**

основано на :

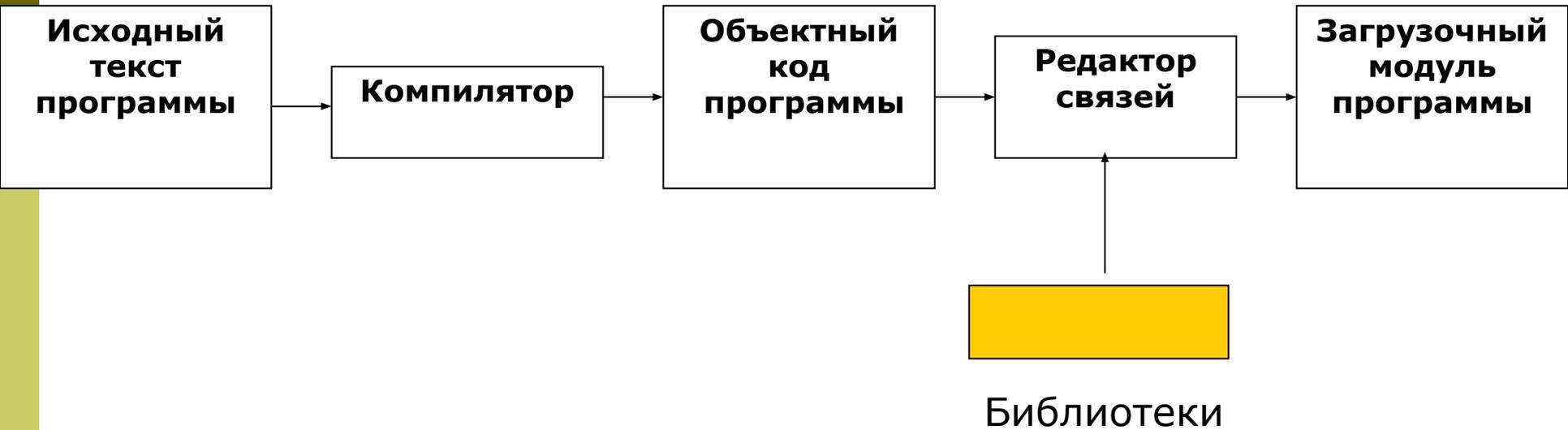
- ✓ выделении *классов объектов*;
- ✓ установлении характерных *свойств объектов и методов* их обработки;
- ✓ создании *иерархии классов, наследовании* свойств объектов и методов их обработки.

3. Трансляторы и их виды

Выделяют два типа трансляторов:

- ❑ *Компиляторы* - обрабатывают весь текст программы, выполняют синтаксический и смысловой анализ и затем автоматически генерируют машинный код ;
- ❑ *Интерпретаторы* - поочередно выполняют перевод и исполнение оператора языка из текста программы.

Схема процесса компиляции программы



3. Системы программирования

а. Средства создания программ :

- ✓ **текстовый редактор;**
- ✓ **транслятор;**
- ✓ **библиотеки программ** - готовые подпрограммы, реализующие стандартные функции, поставляются вместе с компилятором;
- ✓ **редактор связей** – связывает объектный модуль и машинного кода стандартных функций и формирует исполнимый код (файл с расширением .EXE или .COM)

в. Интегрированные системы программирования

✓ **Средства создания программ;**

- ✓ **отладчик** - позволяет анализировать работу программы во время ее выполнения

*Turbo Pascal, Turbo C++, Microsoft C,
Microsoft Basic*

с. Среды быстрого проектирования (RAD-среды)

– реализуют визуальный подход

*Microsoft Visual Basic; Borland Delphi;
Borland C++Builder; MATLAB; Maple;
Microsoft Visual C++; Java*

-
- d. **CASE-средства программирования** -
системы компьютерной разработки
программного обеспечения – программная
инженерия.

Жизненный цикл программного обеспечения

- Постановка задачи
- Анализ и исследование задачи, модели
- Разработка алгоритма
- Программирование
- Тестирование и отладка
- Анализ результатов решения задачи (и уточнение в случае необходимости математической модели с повторным выполнением этапов 2 — 5)
- Сопровождение программы