

Языки программирования

Теория разработки трансляторов и
практическая применимость
технологий генерации
компиляторов

План лекции 1

- Классификация подходов при создании трансляторов
- Компиляция
- Интерпретация
- Ассемблер
- Компиляция JIT
- Виртуальная машина
- Фазы компиляции
- Компиляторы компиляторов
- Формулировка практического задания.
Обсуждение этапов работы.

Классификация подходов при создании трансляторов

Транслятор — программа, которая принимает на вход программу на одном языке, и преобразует её в программу, написанную на другом языке.

Компилятор — выдает результат в виде исполняемого файла (компоновка входит в компиляцию).

Интерпретатор — исполняет программу после разбора. Код исполняется построчно, программа транслируется (интерпретируется) при каждом запуске.

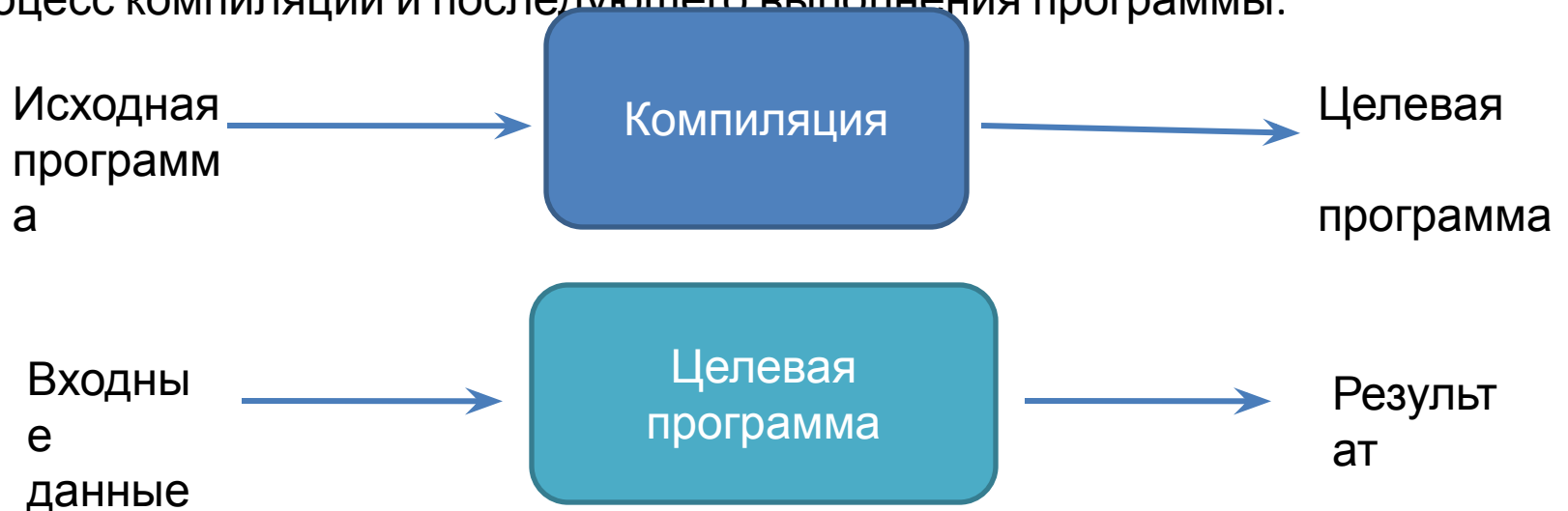
Ассемблер — компьютерная программа, компилятор исходного текста программы написанной на языке ассемблера, в программу на машинном коде.

Компиляция

Компилятор (compiler) переводит исходную программу в эквивалентную программу на языке, понятном компьютеру, то есть на машинном языке.

Программа, полученная в результате работы компилятора - целевая программа (target program).

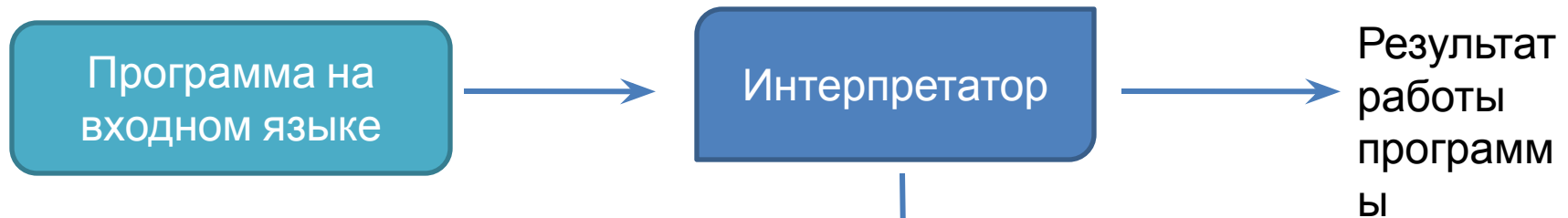
Процесс компиляции и последующего выполнения программы:



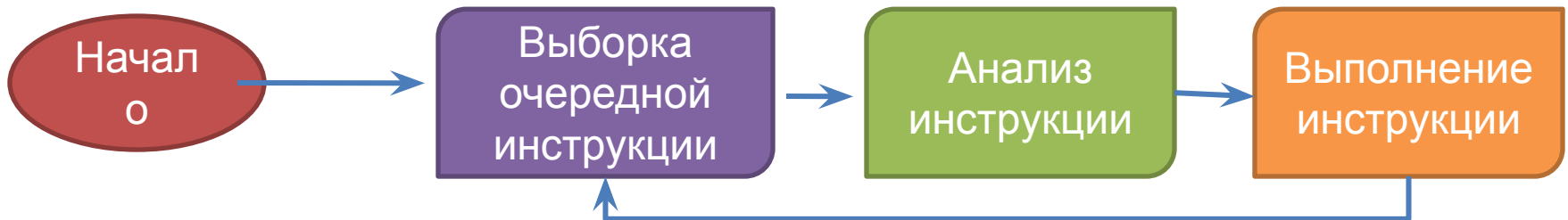
Интерпретация

Интерпретатор — исполняет программу после разбора (в этом случае в роли объектного кода выступает внутреннее представление программы интерпретатором).

Процесс интерпретации программы в общем случае:



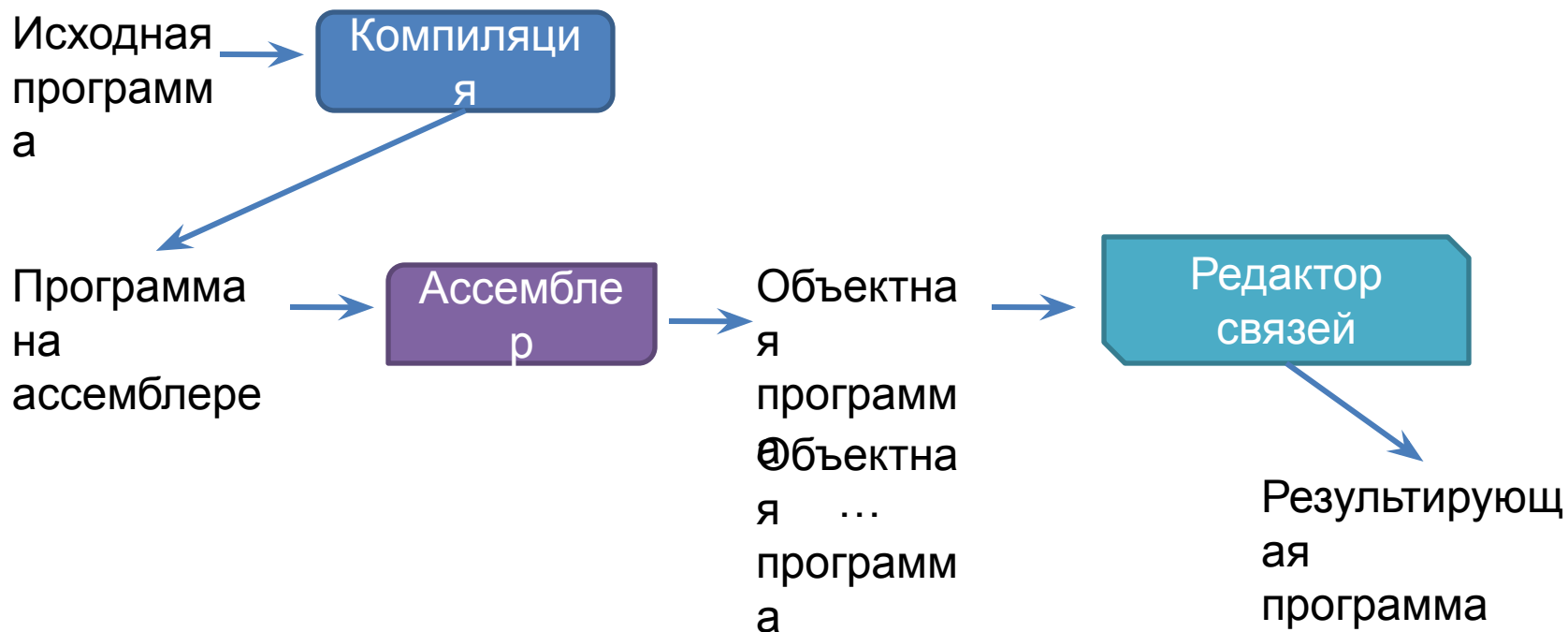
Программа исполняется построчно:



Ассемблер

Ассемблер – это вид транслятора, компилятор исходного текста программы написанной на языке ассемблера, в программу на машинном коде.

Процесс работы компилятора, ассемблера и линковщика:

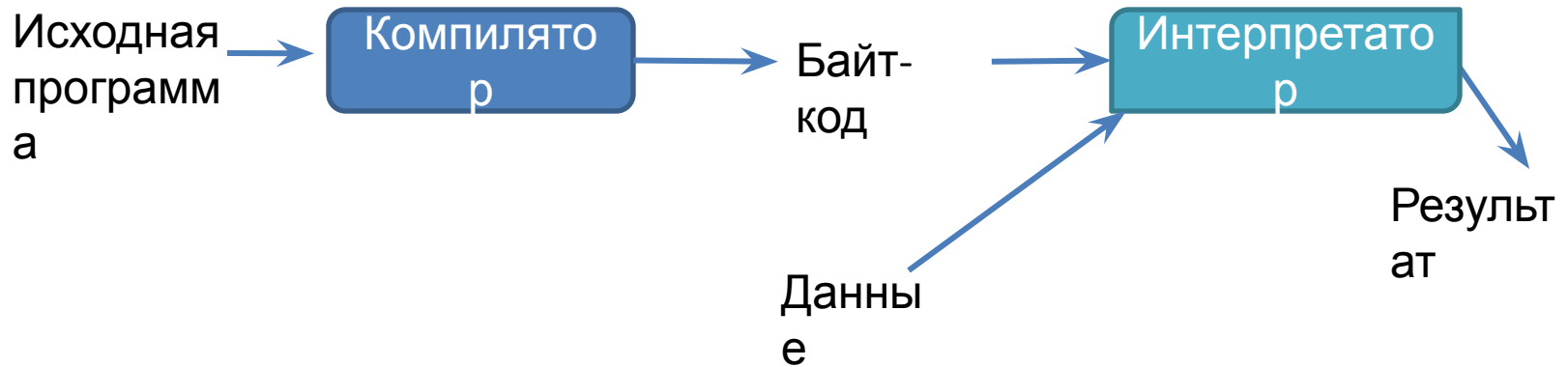


Виртуальная машина

Виртуальная машина– это разновидность интерпретатора, которая:

1. Перед исполнением переводят программу в байт-код
2. Предоставляет программам расширенное по функционалу окружение

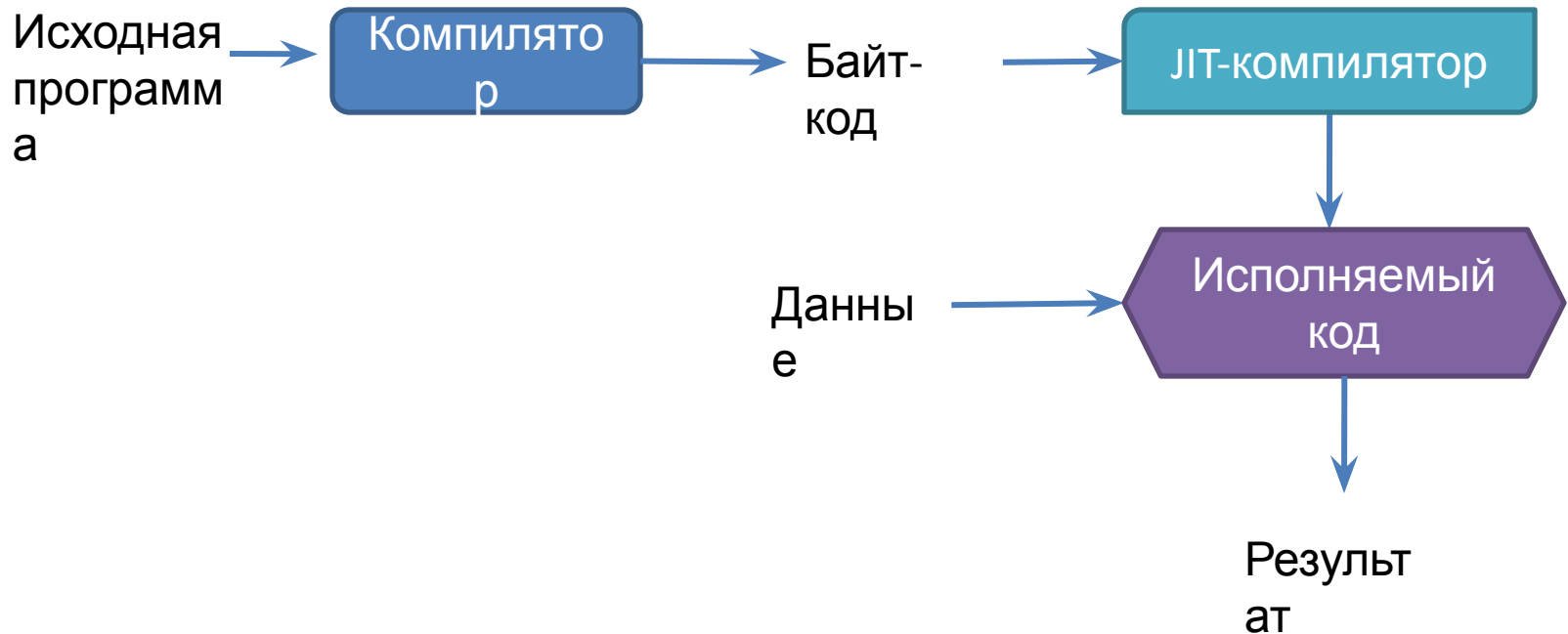
Процесс исполнения кода посредством виртуальной машины:



Компиляция Just In Time

JIT компилятор – это разновидность интерпретатора, которая перед исполнением переводит программу непосредственно в машинный код и хранит его в памяти не сохраняя в виде исполняемого файла

Процесс исполнения кода посредством компилятора JIT:



Фазы компиляции

- *Лексический анализ (сканирование, scanning)*
- *Препроцессор*
- *Синтаксический анализ (разбор, parsing)*
- *Семантический анализ*
- *Генерация промежуточного кода*
- *Оптимизация кода*
- *Генерация кода*

Роль фаз компиляции в наше время

Ранее:



Сейчас

с:

