



Виртуальные машины и трансляторы

Виды виртуальных машин

```
graph TD; A[Виды виртуальных машин] --> B[Процессные]; A --> C[Системные]; B --> D["-Java Virtual Machine<br>-.NET Framework"]; C --> E["-Vmware workstation<br>-Virtualbox"]
```

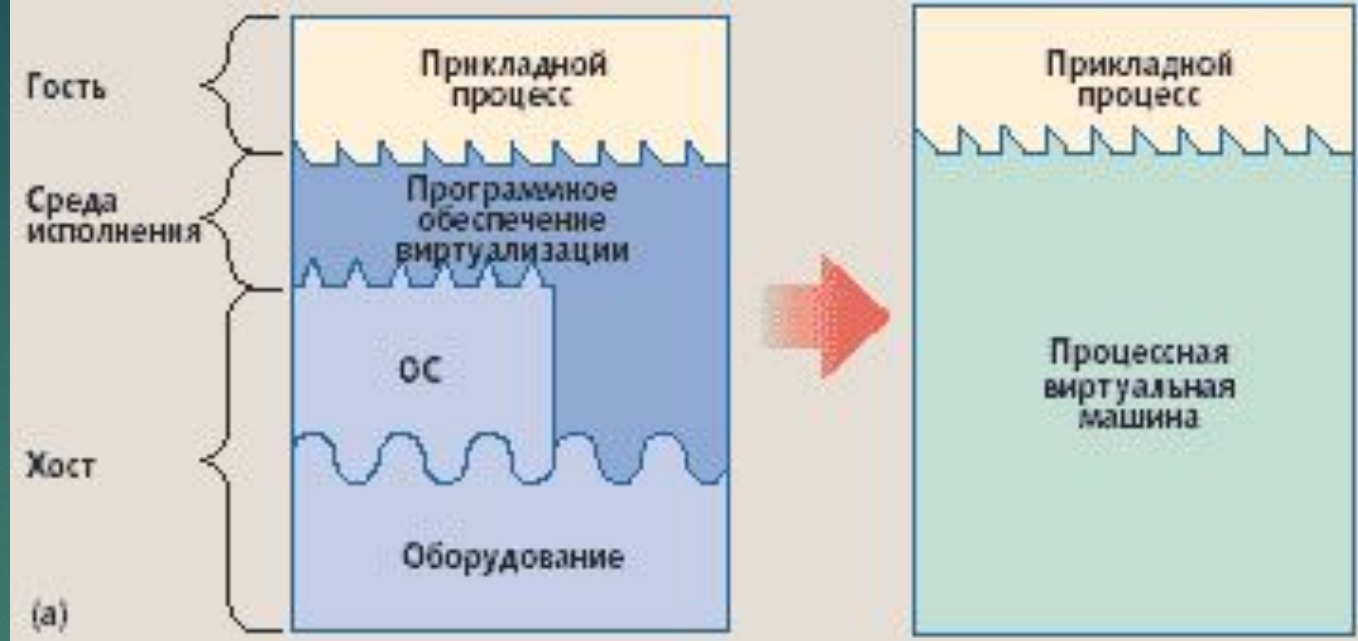
Процессные

- Java Virtual Machine
- .NET Framework

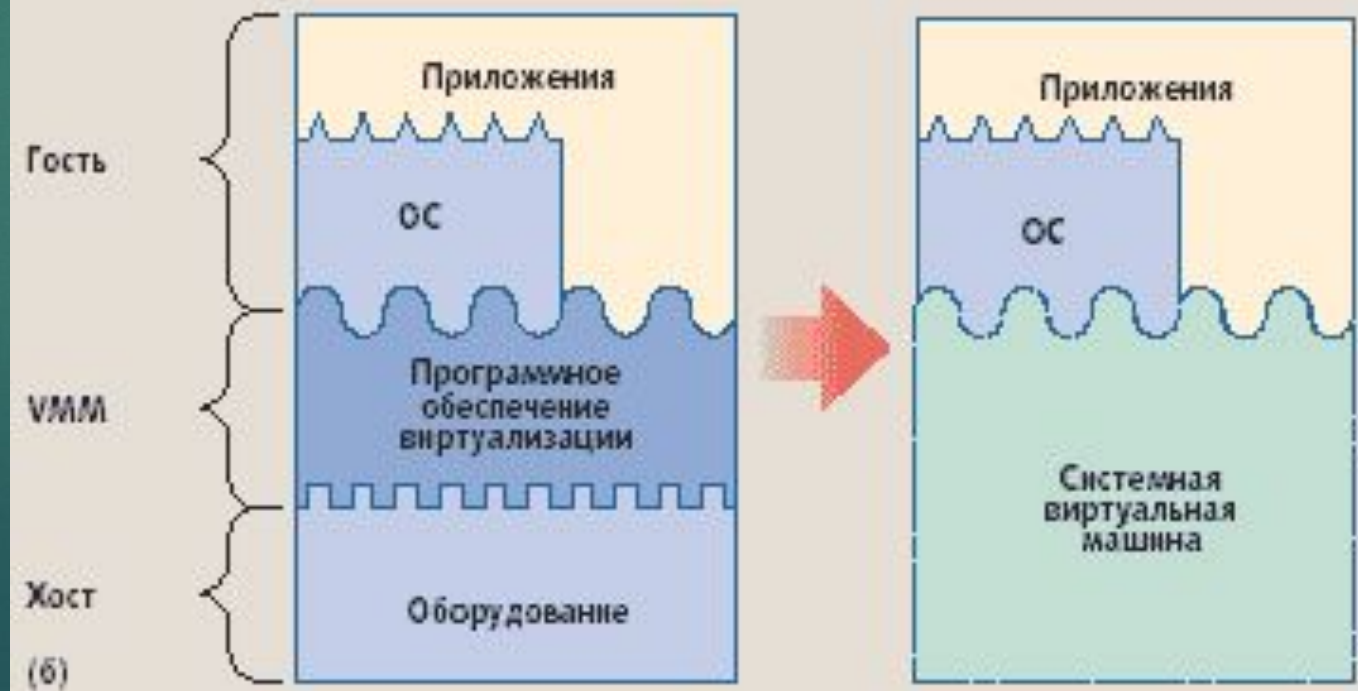
Системные

- Vmware workstation
- Virtualbox

Процесные виртуальные машины



Системные виртуальные машины



Среды языка
высокого
уровня



Трансляторы

- ▶ Транслятор – программа или техническое средство, выполняющее трансляцию программы.
- ▶ Трансляция программы – преобразование программы, представленной на одном из языков программирования, в файл определенного формата.



Виды трансляторов

- ▶ Диалоговый транслятор.
- ▶ Синтаксически-ориентированный (синтаксически-управляемый) транслятор.
- ▶ Однопроходной транслятор.
- ▶ Многопроходной транслятор.
- ▶ Оптимизирующий транслятор.
- ▶ Тестовый транслятор.
- ▶ Декомпилятор.

Виды трансляции

```
graph TD; A[Виды трансляции] --> B[Компиляторы]; A --> C[Интерпретация]; A --> D[Динамическая компиляция]; A --> E[Конвертация];
```

Компиляторы

Интерпретация

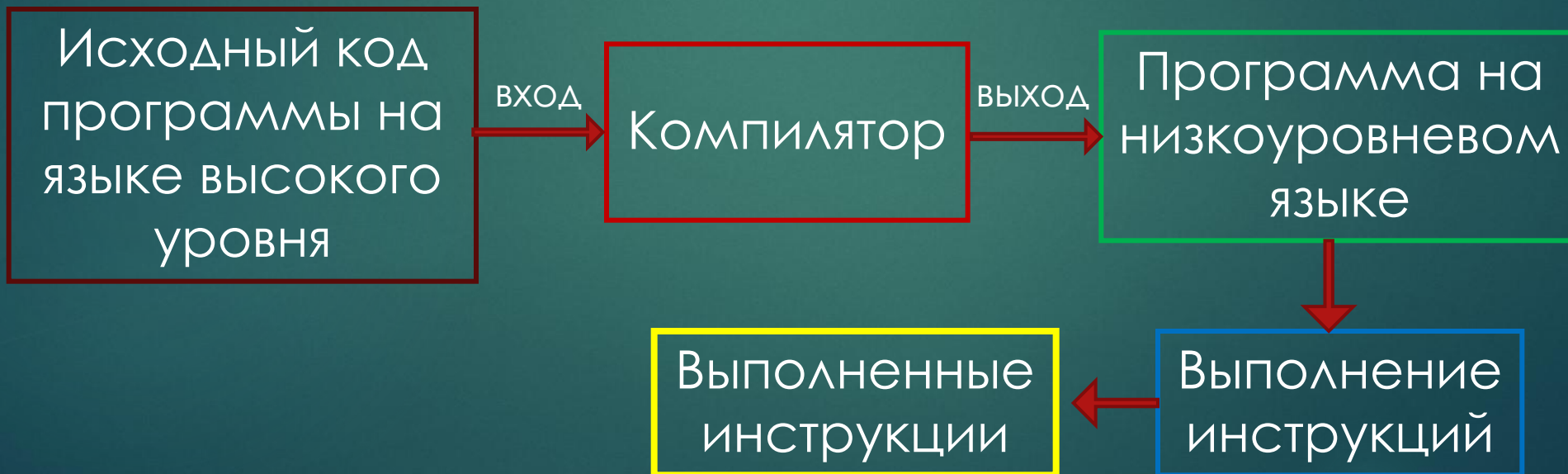
я

Динамическая
компиляция

Конвертация

КОМПИЛЯЦИЯ

- ▶ Компилятор – транслятор, преобразующий исходный код с какого-либо языка программирования на машинный язык.
- ▶ Компиляция – трансляция программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду (абсолютный код, объектный модуль, иногда на язык ассемблера).



Виды компиляции

- ▶ Пакетная.
- ▶ Построчная.
- ▶ Условная.

Этапы процесса компиляции

- ▶ Лексический анализ.
- ▶ Синтаксический (грамматический) анализ.
- ▶ Семантический анализ.
- ▶ Оптимизация.
- ▶ Генерация кода.

Достоинства компиляции

- ▶ компиляция программы выполняется один раз;
- ▶ наличие компилятора на устройстве, для которого компилируется программа, не требуется.

Недостатки компиляции

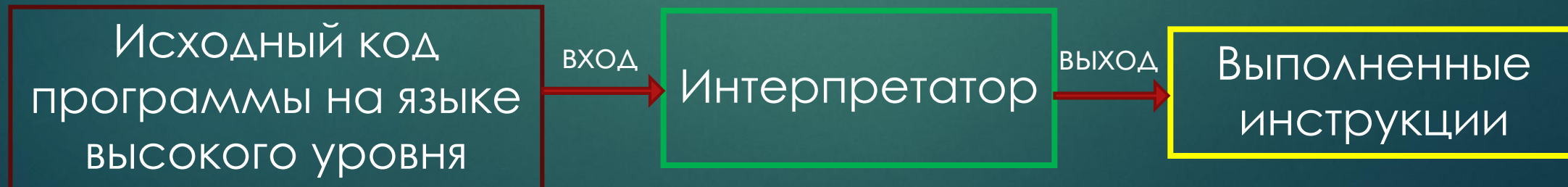
- ▶ компиляция – медленный процесс;
- ▶ при внесении изменений в исходный код, требуется повторная компиляция;
- ▶ при переносе программы на другую платформу требуется перекомпиляция исходного кода.

Интерпретация

- ▶ Интерпретация – процесс чтения и выполнения исходного кода. Реализуется программой – интерпретатором.

Два вида интерпретаторов

- ▶ Простой интерпретатор
- ▶ Интерпретатор компилирующего вида



Этапы работы интерпретатора



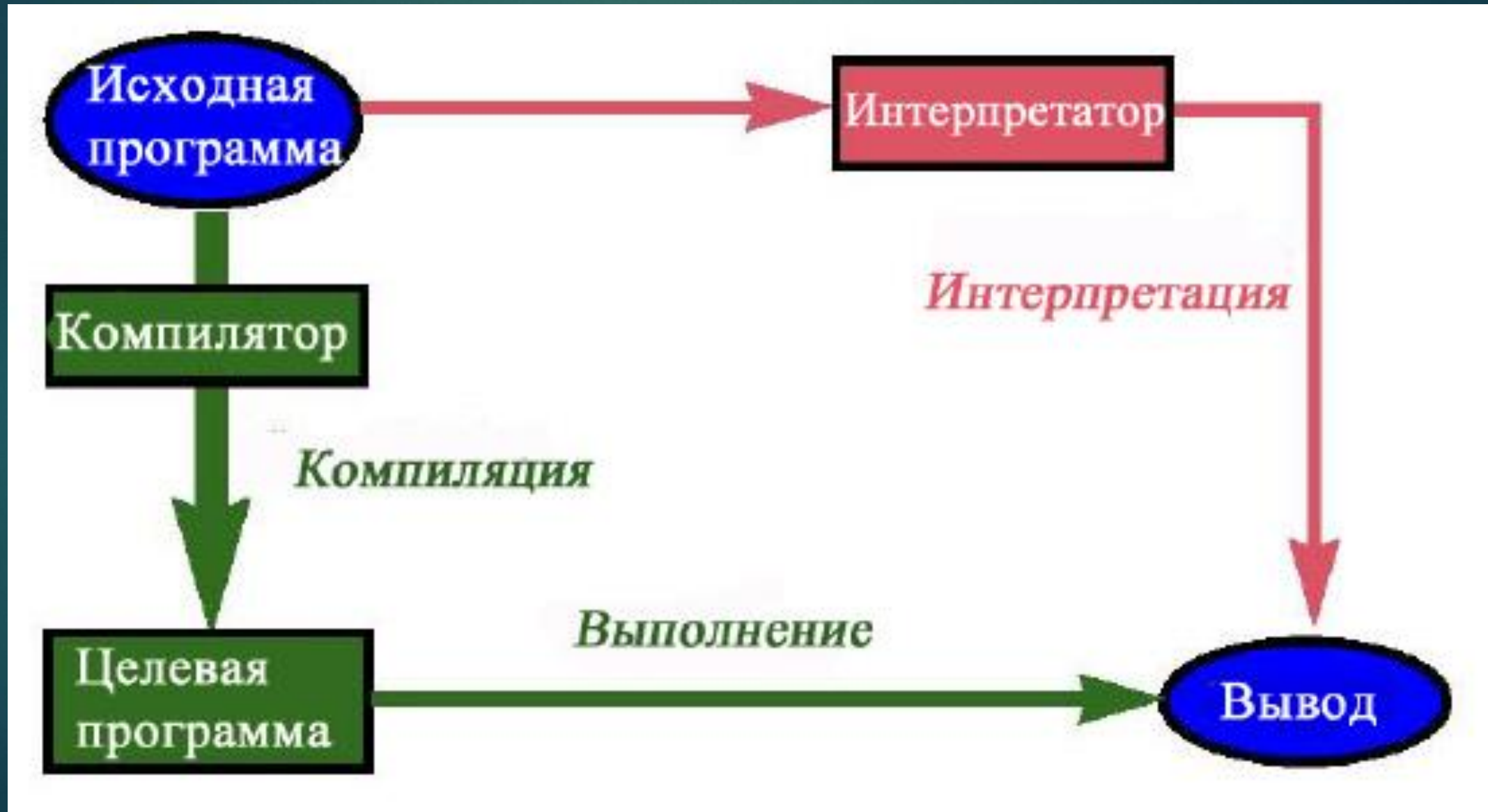
Достоинства интерпретации

- ▶ большая переносимость интерпретируемых программ – программа будет работать на любой платформе, на которой есть соответствующий интерпретатор;
- ▶ более совершенные и наглядные средства диагностики ошибок в исходных кодах;
- ▶ отсутствие необходимости перекомпиляции исходного кода после внесения изменений и при переносе кода на другую платформу;
- ▶ меньшие размеры кода по сравнению с машинным кодом, полученным после обычных компиляторов.

Недостатки интерпретации

- ▶ интерпретируемая программа не может выполняться отдельно без программы-интерпретатора. Сам интерпретатор при этом может быть очень компактным;
- ▶ интерпретируемая программа выполняется медленнее, поскольку промежуточный анализ исходного кода и планирование его выполнения требуют дополнительного времени по сравнению с непосредственным исполнением машинного кода, в который мог бы быть скомпилирован исходный код;
- ▶ практически отсутствует оптимизация кода, что приводит к дополнительным потерям в скорости работы интерпретируемых программ.

Компиляция против интерпретации без ввода



Компиляция против интерпретации С ВВОДОМ



Динамическая или JIT компиляция

- ▶ Динамическая или JIT (Just In Time) компиляция – трансляция, при которой исходный или промежуточный код преобразуется (компилируется) в машинный код непосредственно во время исполнения.



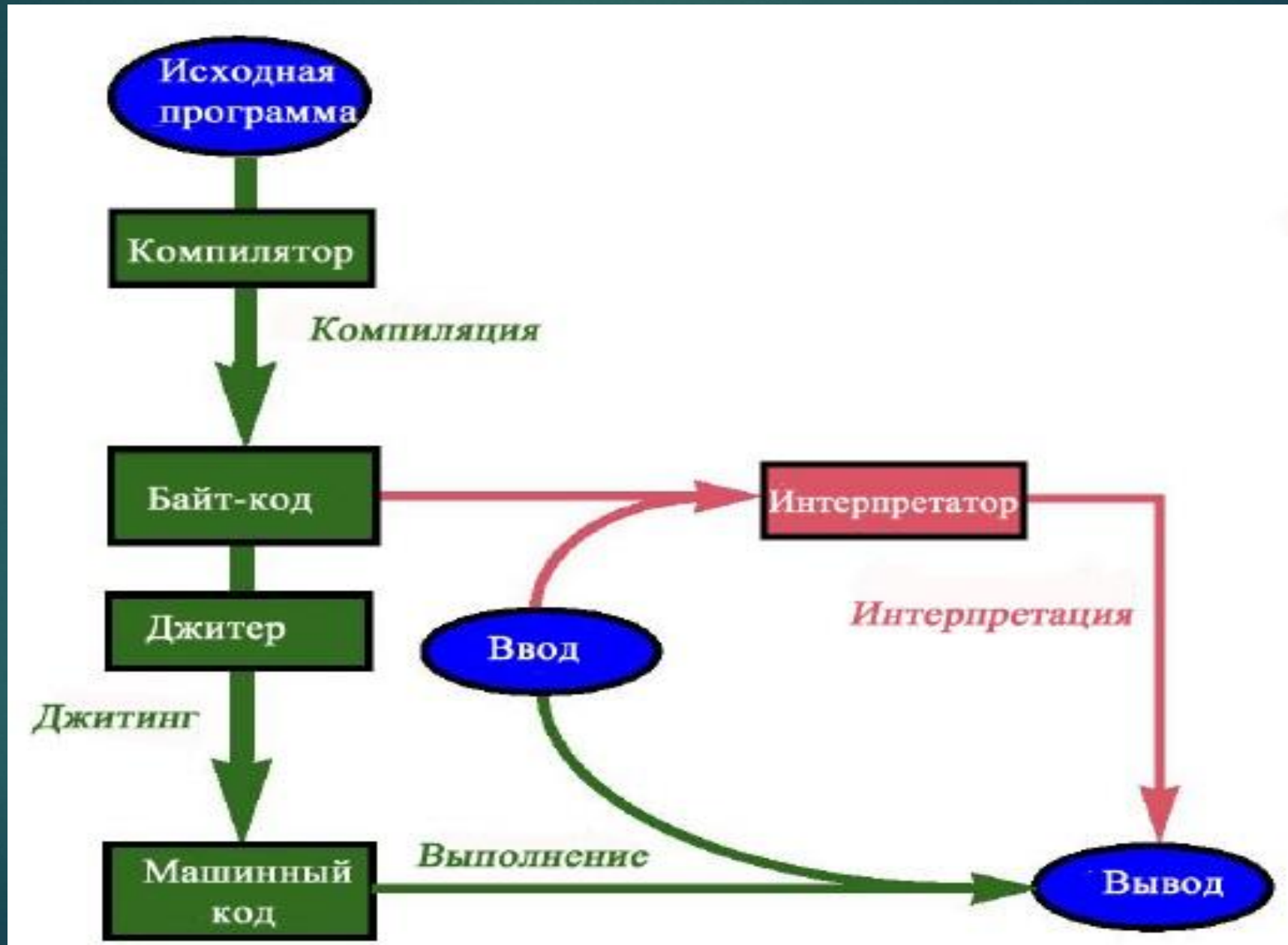
Достоинства JIT компиляция

- ▶ скорость работы динамически компилируемых программ близка к скорости работы компилируемых программ;
- ▶ отсутствие необходимости перекомпиляции программы при переносе на другую платформу.

Недостатки JIT компиляция

- ▶ большая сложность реализации;
- ▶ большие требования к ресурсам.

Компиляция + интерпретация + JIT



Конвертация

- ▶ Конвертор – переводит программу с одного языка на другой.



Pascal

```
Var
  a,i:integer;
Begin
  a = ReadLn();
  if a>5
    for I := 5 to 10 do
      WriteLn(i);
  else
    WriteLn("It's time for magic!");
END.
```

Конвертор

Пуф! Магия!

C++

```
#include <iostream>
Using namespace std;
int a;
int main() {
  cin >> a;
  if (a > 5)
    for (int i=5; i<=10)
      cout << i << endl;
  else
    cout << "It's time for magic!" << endl;
  return 0;
}
```

Достоинства конвертации

- ▶ быстрый перевод больших блоков программы из одного языка в другой;
- ▶ небольшие требования к ресурсам.

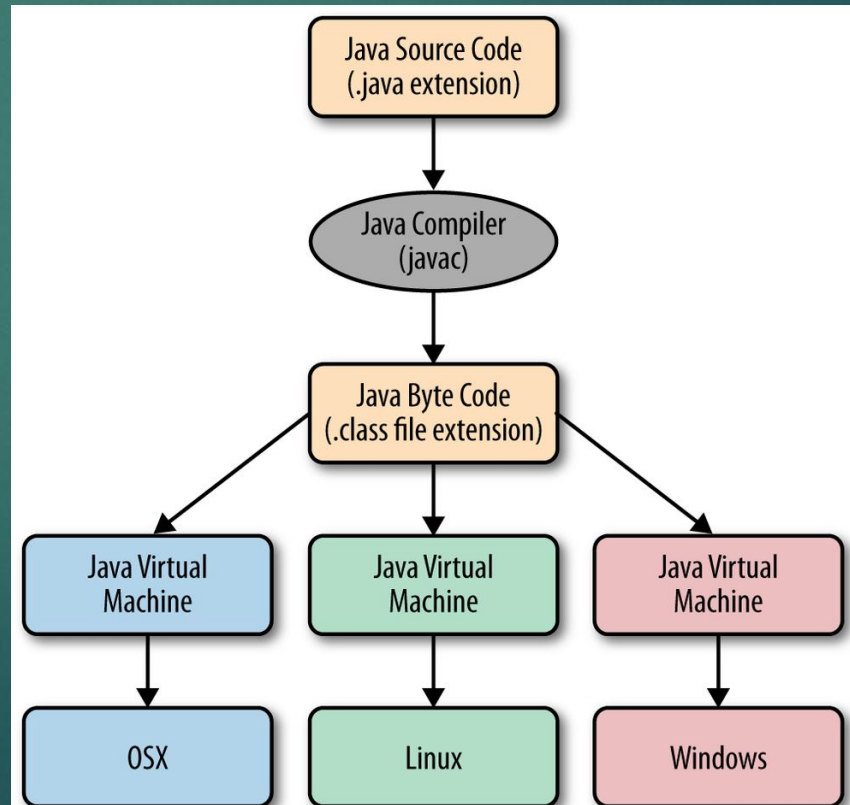
Недостатки конвертации

- ▶ отсутствие определенных конструкций в языке, в который переводится программа, в следствии чего возможны ошибки;
- ▶ перевод производится в основном в простые конструкции без использования особенностей языка, поскольку есть вероятность, что в переводимом языке нет аналога.

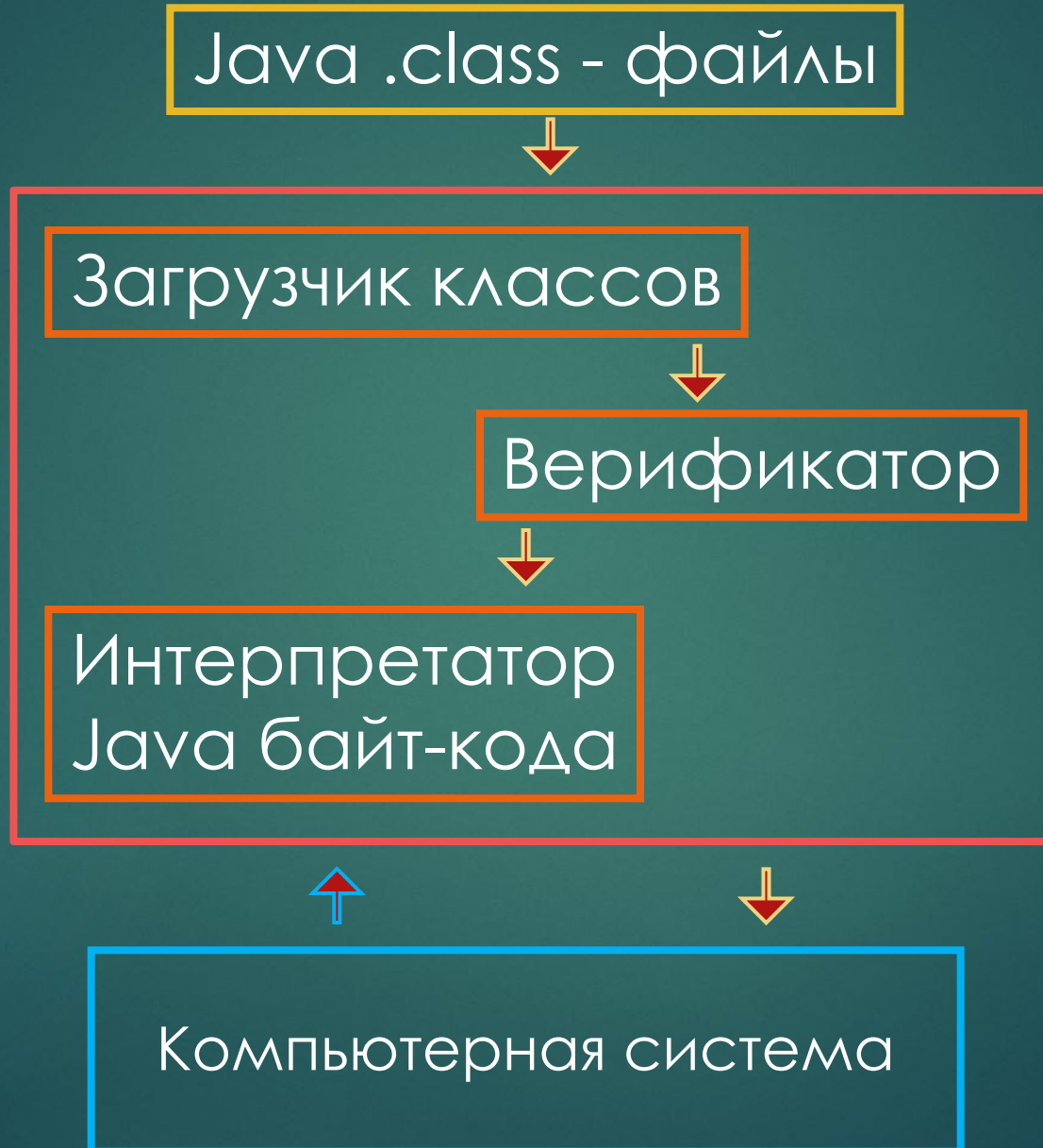


Java Virtual Machine

- ▶ JRE (Java Runtime Environment) – минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений, без компилятора и других средств разработки.
- ▶ JVM (Java Virtual Machine) – виртуальная машина Java – основная часть исполняющей системы Java.



Архитектура JVM





Конец истории