

ТЕХНОЛОГИИ ВЫПОЛНЕНИЯ КОДА

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ЦЕЛЬ

Понять процессы компиляции и интерпретации

Получить представление о JVM

ПЛАН ЗАНЯТИЯ

Компиляция

Интерпретация

Особенности Java

Работа в IntelliJ IDEA

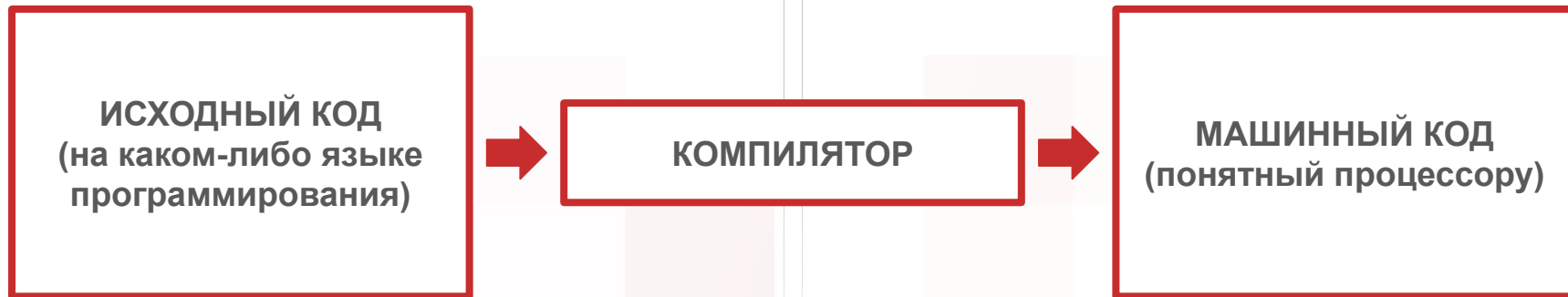
КОМПИЛЯЦИЯ

Какое бы Вы дали определение этому слову?

КОМПИЛЯЦИЯ

Компиляция преобразует файл исходного кода в программу **целиком**

Результатом компиляции является **исполняемый файл** с машинным (бинарным) кодом процессора

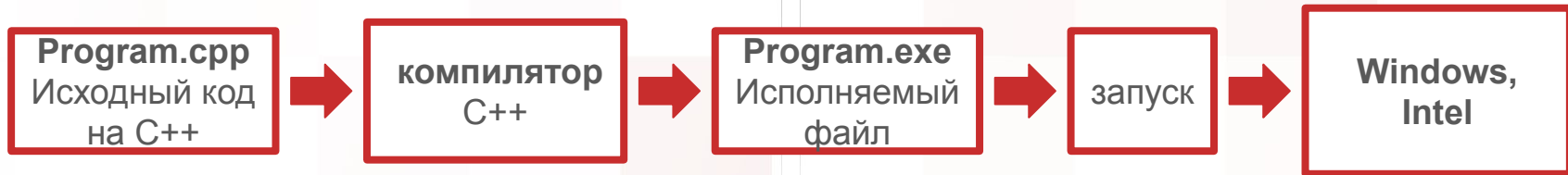


КОМПИЛЯЦИЯ

Исполняемый файл может быть запущен на компьютере **только с той архитектурой** (операционной системой и процессором), **для которой он был скомпилирован**

Скорость работы таких программ **очень высокая**

Программы, написанные на языках программирования компилируемого типа (C/C++), имеют **прямой доступ к “железу” и операционной системе**

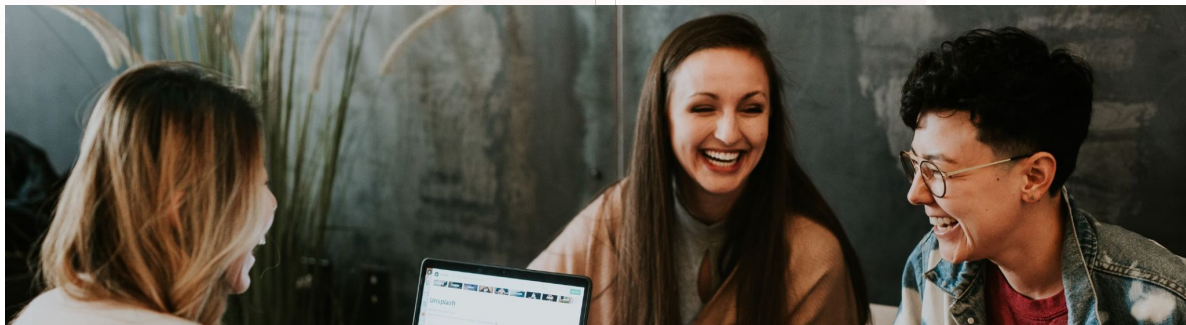


СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

Компилятор - тоже программа, которая преобразует исходный код в исполняемый файл с машинным кодом “целиком”

Такие программы очень быстро работают, потому что имеют прямой доступ к операционной системе и железу

Такие программы нельзя “переносить” с одного компьютера на другой, если на том отличается операционная система или процессор



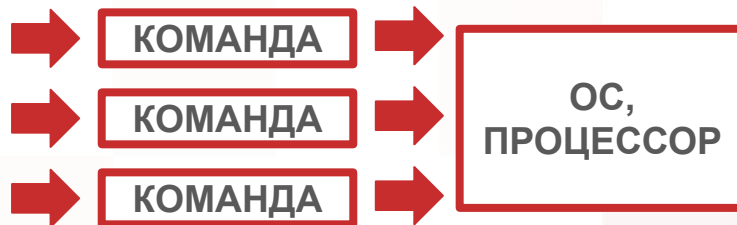
ИНТЕРПРЕТАЦИЯ

Интерпретация преобразует файл исходного кода в машинный код **строками или блоками**

Программа **не имеет прямого доступа к ОС и “железу”**



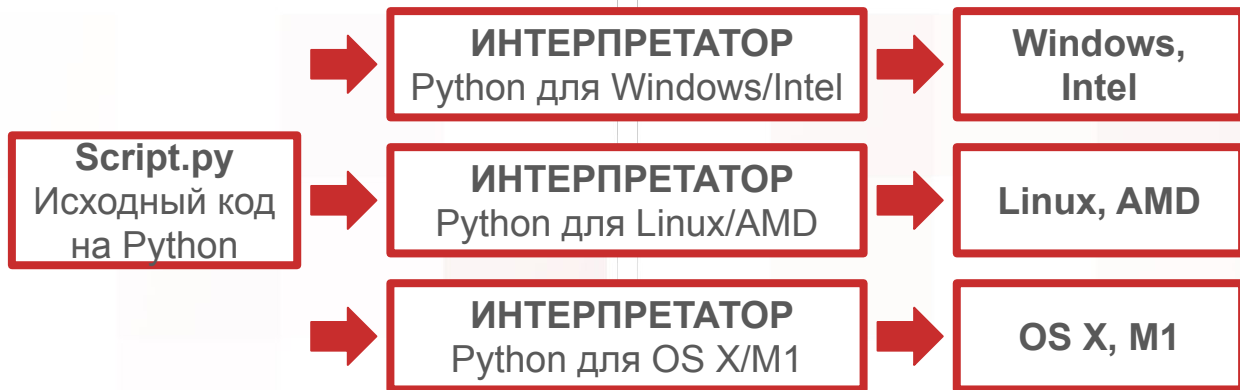
Интерпретация происходит **во время работы** приложения



ИНТЕРПРЕТАЦИЯ

Программа, написанная на языке интерпретируемого типа (JavaScript, Python), **может быть запущена на любом типе архитектуры (кроссплатформенность)**. Главное, чтобы для этого типа был соответствующий интерпретатор

Интерпретируемые программы, как правило, выполняются **медленнее компилируемых**.



СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■ Интерпретатор - тоже программа, которая преобразует исходный код в машинный “на лету”, в процессе работы, причем делает это фрагментами (строками, блоками)

■ Интерпретируемые программы работают медленнее

■ Интерпретируемые программы можно переносить на любой компьютер, где есть соответствующий интерпретатор



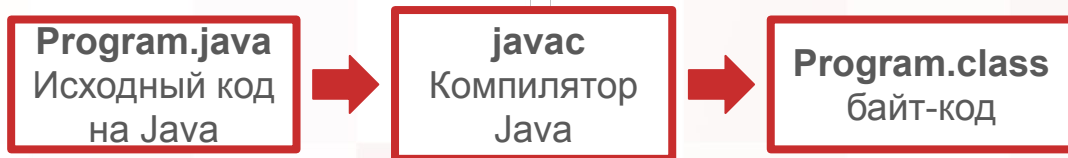
JAVA

Что Вы уже знаете о
Java?

JAVA

Java сочетает **высокую скорость** компилируемых языков и **кроссплатформенность** интерпретируемых

Перед тем как запустить программу на Java, ее нужно скомпилировать в **байт-код виртуальной машины Java (JVM)**



JAVA

Байт-код - инструкции, которые выполняет не процессор, а **JVM**

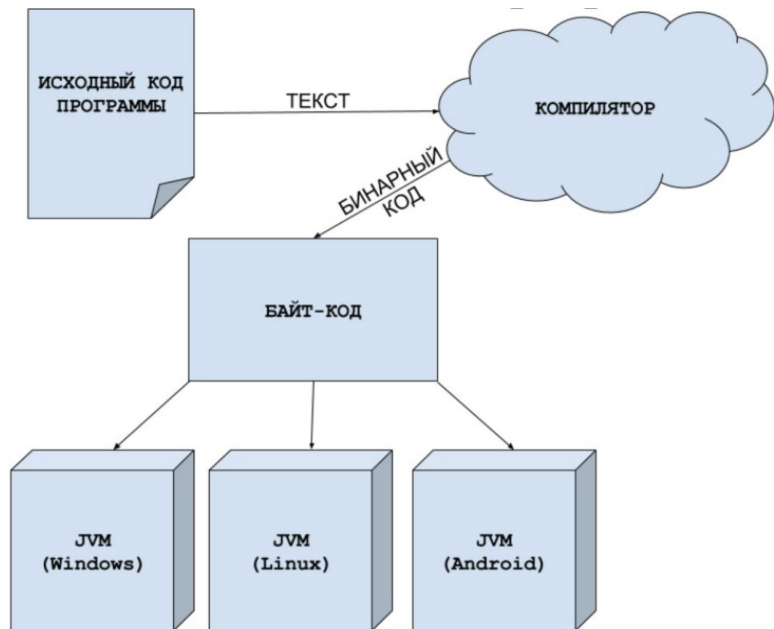
Внутри JVM находится **интерпретатор**, преобразующий байт-код в **команды конкретного процессора**

Программа на Java работает **везде**, где установлена соответствующая Java-машина



JAVA

Жизненный цикл программы на Java выглядит так:



Преобразование java-кода в байт-код для запуска на JVM:

```
public class Alp
    public static
        x = y * 2;
    }
}
```

```
...
load_const 2
load y
mul y, 2
save $0
...
```

Выполнение байт-кода на процессоре:

```
...
load_const 2
load y
mul y, 2
save $0
...
```

Hardware (CPU)

OS (Linux, Windows, etc.)

JAVA

JVM - это не просто интерпретатор, интерпретация байт-кода лишь одна из ее функций

JVM **эмулирует** для вашего приложения программно-аппаратную среду внутри основной среды компьютера

Дополнительно JVM оптимизирует байт-код во время работы приложения (**JIT-компиляция**), удаляет ненужные объекты (**сборка мусора**), позволяет программе работать в “многопоточном режиме” (**multithreading**).



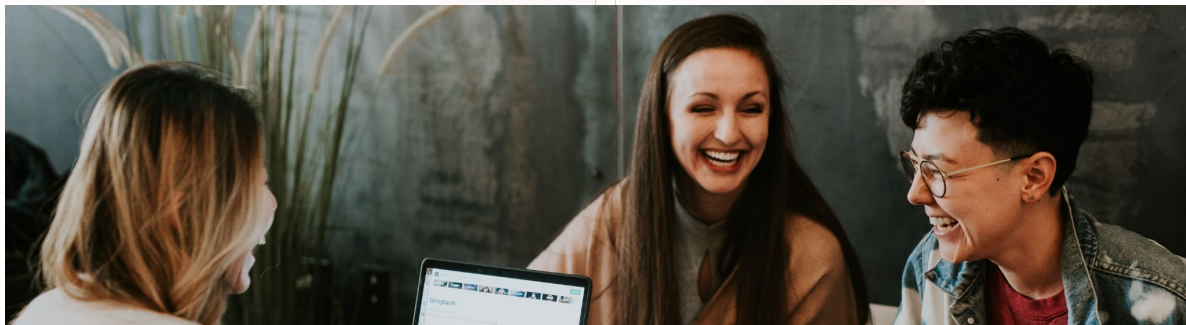
СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

Java - язык программирования, сочетающий особенности компиляции и интерпретации

Сначала программу на Java следует скомпилировать, но не в обычный машинный код, а в код виртуальной машины JVM

JVM включает в себя огромный набор механизмов, один из которых - интерпретация программ из байт-кода

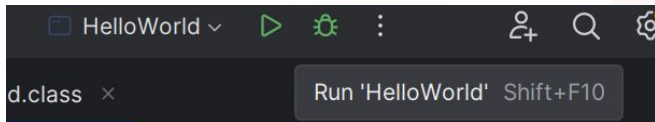
Программы на Java одновременно быстрые и кроссплатформенные



IntelliJ IDEA

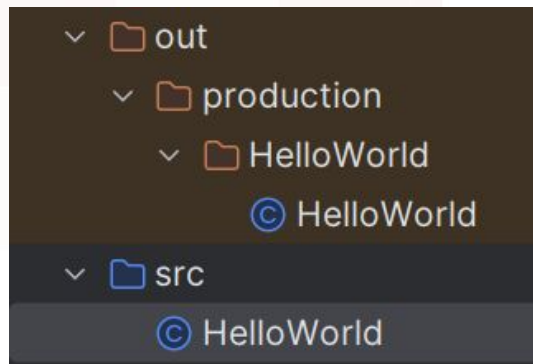
IntelliJ IDEA - это интегрированная среда разработки, включает огромное количество инструментов (отладчик, подсветка синтаксиса, автозаполнение и т. д.)

Важно помнить: IDEA - это не Java, JVM, интерпретатор или компилятор



IDEA делает разработку приятным и интересным процессом, анализирует код и дает подсказки.

IDEA вызывает эти инструменты автоматически “под капотом” при нажатии кнопки **Run**



Запуск в терминале/шелле/командной строке

Терминология: [краткая справка](#) (читайте до “Virtual Console”)

Некоторые команды для работы с интерфейсом командной строки:

- pwd - напечатать текущую директорию
- ls - отобразить содержание текущей директории или указанной в аргументах директорий
- mkdir - создать директорию
- cd - перейти в указанную директорию
- cp - скопировать файл или директорию указанную в качестве первого аргумента в расположение указанное во втором аргументе
- rm - удалить указанный в аргументе файл или директорию
- mv - переместить файл/директорию в другое место, либо переименовать файл/директорию
- echo - напечатать текст указанный в аргументах
- cat - напечатать содержимое файла

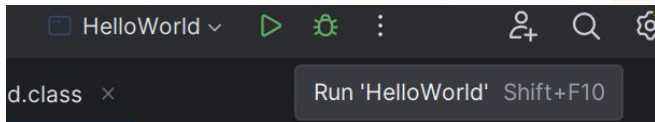
Примеры использования:

```
# pwd
/Users/username/projects/ait/l1/HelloWorld
# mkdir myDirName
# ls
HelloWorld.iml examples out src myDirName
# cp src/Main.java myDirName
# cd myDirName
# ls
Main.java
# mv Main.java Main2.java
# ls
Main2.java
# cat Main2.java
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
# rm Main2.java
```

IntelliJ IDEA

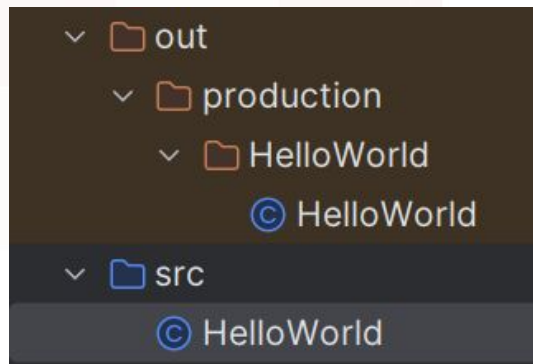
IntelliJ IDEA - это интегрированная среда разработки, включает огромное количество инструментов (отладчик, подсветка синтаксиса, автозаполнение и т. д.)

Важно помнить: IDEA - это не Java, JVM, интерпретатор или компилятор



IDEA делает разработку приятным и интересным процессом, анализирует код и дает подсказки.

IDEA вызывает эти инструменты автоматически “под капотом” при нажатии кнопки **Run**



ПОИГРАЕМ ;)

Компиляция

Интерпретация

Java

JVM

IntelliJ IDEA

Дополнительные материалы

Про версии java: [статья](#)

Запуск java программы в терминале: [читать до “используем пакеты”](#)

Подробнее про байт-кода java:

- [подробное объяснение](#)
- [разбор байт-кода HelloWorld](#)

ДОМАШНЕЕ ЗАДАНИЕ

1. Создайте новый проект в IntelliJ IDEA. Назовите его **CurrencyConverter**.
2. В Main-классе создайте main-метод, который выводит сообщения следующего вида:

Приветствуем в
CurrencyConverter!

Выберите исходную валюту:

1. EURO
2. USD
3. TL

3. Запустите программу

4. Попробуйте скомпилировать и запустить программу в терминале (командной строке)

СТАВИМ +, ЕСЛИ ВАМ ПОНЯТНО ДОМАШНЕЕ ЗАДАНИЕ



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH