



## Урок 1

# Как работает Java?

«Написано один раз — работает везде». JVM. Инструменты.

[Ведущий принцип Java](#)

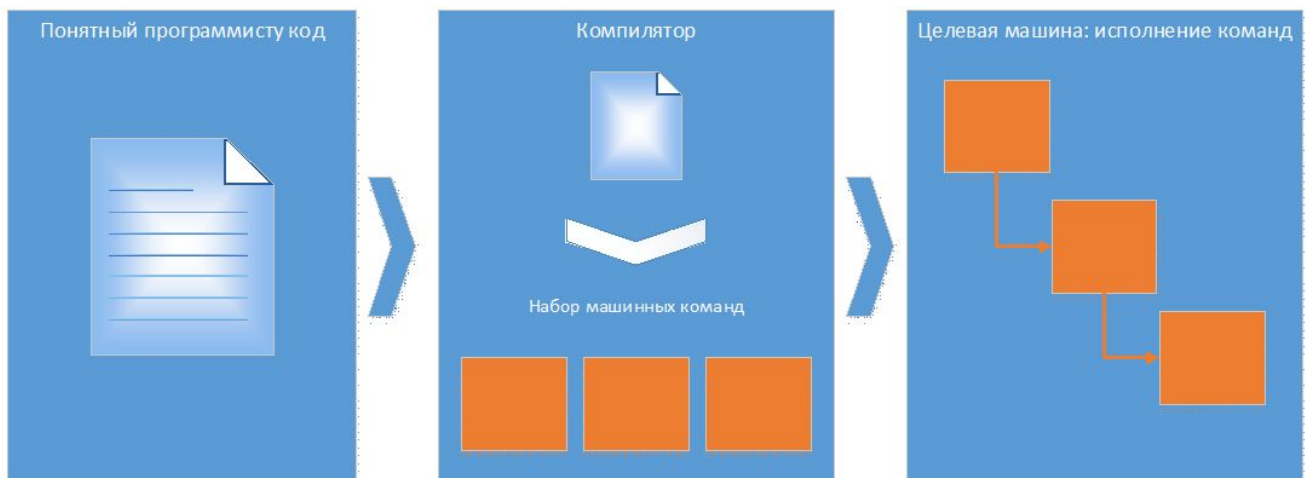
[JVM](#)

[Инструменты](#)

# Ведущий принцип Java

Основной принцип Java гласит: «Написано один раз — работает везде». Разумеется, он не мог быть воплощен без сложного, но изящного решения. Познакомимся с ним на этом уроке, но сначала рассмотрим общие подходы к созданию ПО, которые использует язык Java.

Java является компилируемым языком со строгой типизацией. Компилируемым называют тот язык, где программа после компиляции будет содержать только машинные инструкции, а не привычный код. Команды из понятных человеку преобразуются в читаемые компьютером.



Наряду с компилируемыми есть интерпретируемые языки. В них инструкции на понятном коде не выполняются непосредственно машиной, а считываются другой программой, которая направляет их целевой машине. К таким языкам относятся JavaScript, PHP.

Компиляция дает преимущество: между кодом и машиной, на которой исполняется программа, нет прослойки в виде программы-интерпретатора. Команды подаются в машинном коде, и работа выполняется быстрее.

## JVM

Чтобы выполнять принцип «Написано один раз — работает везде», создатели Java разработали решение — специальную виртуальную машину, на базе которой исполняются машинные команды, полученные в результате компиляции. Но если код скомпилирован, почему бы не запускать его сразу после компиляции — ведь полученная программа содержит машинный код?

Операционные системы (даже версии одного семейства) имеют разные машинные команды или подходы к их выполнению. Для мультиплатформенности разработчику пришлось бы компилировать свой код на всех возможных версиях компилятора или иметь на выходе множество версий программы для каждой ОС.

Разработчики языка Java создали виртуальные машины для всех популярных ОС. Они гарантируют, что программа будет вести себя одинаково в разных ОС.

Результат компиляции кода (байт-код) не похож на машинные команды, которые запускаются на реальном процессоре. Байт-код — это адаптация между исходным и машинным кодом. Его основная цель — быть эффективно исполненным внутри JVM.

Чтобы запустить на своем устройстве скомпилированную программу, написанную на Java, достаточно скачать и установить JRE – Java Runtime Environment. Это пакет, который включает JVM и запускает ее при вызове программы, написанной на Java.

## Инструменты

Если для запуска программы достаточно легковесного пакета JRE, то для разработки потребуется более объемный JDK – Java Development Kit. Этот комплект свободно предоставляется компанией Oracle. Мы будем применять JDK версии 8. Хотя Java 9 недавно вышла в релиз, в работе вам наверняка придется столкнуться именно с 8 версией платформы. В фундаментальных вопросах Java не претерпела изменений. JDK можно загрузить по этой [ссылке](#).

После установки можно проверить работоспособность при помощи команды **java -version**:

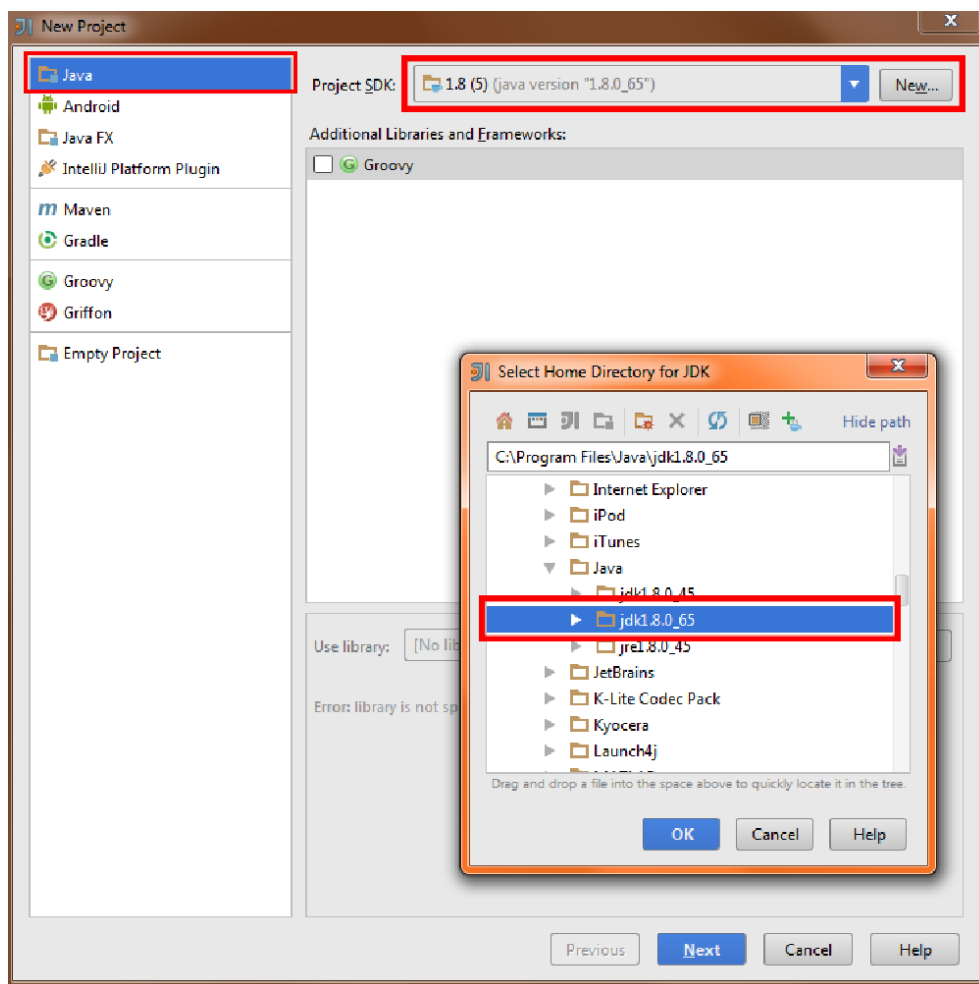
```
C:\>java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

Чтобы удобно вызывать команды компилятора, стоит прописать путь к JDK в переменной **PATH**. Как правило, это требуется только в Windows-средах. Настройка производится следующим образом:

1. Панель управления.
2. Система.
3. Дополнительные параметры системы.
4. Дополнительно.
5. Переменные среды.
6. Создать.
7. Выбрать переменной **PATH** (если ее не существует, создать).
8. Прописать в переменной путь до директории **bin** внутри директории установки JDK, не включая ее.

Пригодится IDE (Integrated Development Environment) — среда разработки, которая серьезно облегчает жизнь разработчикам, предоставляя функционал автозамен, подсказок, подсветки кода, запуска смежного ПО. Наиболее распространена в кругу Java-разработчиков IDE IntelliJ IDEA. Для обучения будет достаточно [бесплатной версии Community Edition](#).

При создании первого проекта в IntelliJ IDEA необходимо указать путь к установленному JDK, как показано на рисунке: **Project SDK -> New -> JDK ->** путь к папке **jdk1.x.x\_xxx**.



Теперь мы готовы писать код, чем и займемся в следующем уроке.