

Немного о байт-коде Java.

Байт-код Java – это промежуточное представление кода Java, предназначенное для выполнения Java Virtual Machine (JVM). Он генерируется компилятором Java и интерпретируется JVM. Байт-код Java состоит из последовательности инструкций, которые указывают JVM, какие операции необходимо выполнить.

Байт-код Java имеет ряд преимуществ перед исходным кодом Java. Во-первых, он более компактный, что позволяет быстрее передавать его по сети и хранить на диске. Во-вторых, он более независим от платформы, чем исходный код Java, что позволяет выполнять его на любой платформе, где установлена JVM.

Байт-код Java также имеет некоторые недостатки по сравнению с исходным кодом Java. Во-первых, он менее читаем для человека, что затрудняет его отладку и модификацию. Во-вторых, он более уязвим для вредоносного кода, поскольку злоумышленники могут использовать его для создания вирусов и других вредоносных программ.

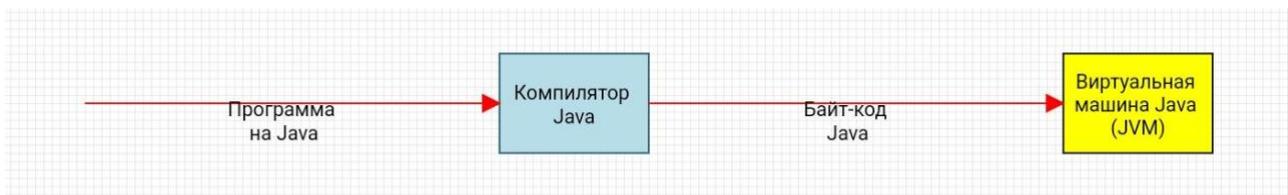
Здесь, исходным код Java – это код, написанный программистом на языке Java, например, при создании приложения.

Вот некоторые из наиболее важных инструкций байт-кода Java:

- `aload` - загрузить ссылку на объект из локальной переменной
- `astore` - сохранить ссылку на объект в локальную переменную
- `iload` - загрузить целочисленное значение из локальной переменной
- `istore` - сохранить целочисленное значение в локальную переменную
- `fload` - загрузить значение с плавающей точкой из локальной переменной
- `fstore` - сохранить значение с плавающей точкой в локальную переменную
- `dload` - загрузить двойное число из локальной переменной
- `dstore` - сохранить двойное число в локальную переменную
- `lconst_0` - загрузить значение 0 в локальную переменную
- `lconst_1` - загрузить значение 1 в локальную переменную
- `iconst_m1` - загрузить значение -1 в локальную переменную
- `fconst_0` - загрузить значение 0.0 в локальную переменную
- `fconst_1` - загрузить значение 1.0 в локальную переменную
- `dconst_0` - загрузить значение 0.0 в локальную переменную
- `dconst_1` - загрузить значение 1.0 в локальную переменную
- `bipush` - загрузить целочисленное значение в локальную переменную
- `sipush` - загрузить целочисленное значение в локальную переменную

- ldc - загрузить константу в локальную переменную
- getstatic - загрузить статическое поле из класса
- putstatic - сохранить статическое поле в класс
- getfield - загрузить поле из объекта
- putfield - сохранить поле в объект
- invokevirtual - вызвать виртуальный метод
- invokespecial - вызвать специальный метод
- invokestatic - вызвать статический метод
- invokeinterface - вызвать интерфейсный метод
- return - вернуть значение из метода
- athrow - бросить исключение

Если использовать язык программирования Java, то схема преобразования будет такой:



Если использовать язык программирования Kotlin, то схема преобразования будет такой:



Компилятор располагается обычно в среде разработки приложения. Например, входит в состав Android Studio, IntelliJ IDEA. JVM находится на планшете, смартфоне. Таким образом, преобразование в байт-код Java, это необходимый этап преобразования, который затем, используется в JVM.

Байт-код всегда используется, когда есть виртуальная машина Java (JVM). Это потому, что JVM является основой выполнения программ на Java и других языках, которые компилируются в байт-код Java. Байт-код представляет собой промежуточный код, который генерируется компилятором из исходного кода программы. JVM интерпретирует этот байт-код, преобразуя его в машинные инструкции (машинный код), которые могут быть выполнены процессором.

Использование байт-кода позволяет программам на Java и других языках работать на разных платформах, независимо от типа процессора или операционной системы. Это делает Java одной из самых популярных платформ для разработки программного обеспечения.

Чтобы работать в другой ОС, программе написанной на Java, в эту другую ОС необходимо установить JVM. Виртуальная машина обеспечивает выполнение Java-приложения, независимо от того, какая операционная система используется.