

Что такое Bytecode?

Зачем разработчикам знать про Bytecode?

Очень частая задача

В прикладной код:

```
public void secureAccess(String param) {  
    {какие-то действия}  
}
```

надо автоматически добавить что-то «техническое»:

```
public void secureAccess(String param) {  
    <логирование вызова метода>  
    {какие-то действия}  
}
```

Aspect-oriented programming (AOP)

Aspect-oriented programming (AOP) is a **programming paradigm** that aims to increase modularity by allowing the separation of cross-cutting concerns. It does so by adding additional behavior to existing code (an advice) without modifying the code itself.

Википедия

Aspect-oriented programming (AOP)

Суть подхода:

Отделение «технического функционала» от «бизнес-логики».

Примеры «технического функционала»:

- логирование
- проверка прав доступа
- Управление транзакциями
- Метрики

```
accountDebit(account, sum); -- бизнес-логика
```



```
accountDebit(account, sum) { -- новый метод  
    logger(account, sum); -- технический функционал  
    accountDebitProxied(account, sum); -- старый метод  
}
```

Aspect-oriented programming (AOP)

Варианты реализации:

- динамический прокси ([Class Proxy](#))

пример: aop.proxy

Как работает JVM

Программа на java из исходного кода
компилируется в byte-code.

Byte-code выполняется в JVM.

Write once, run anywhere (WORE)

Как работает JVM

А оно работает так:

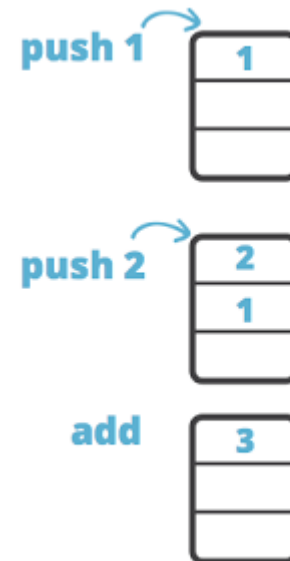
Да, JVM – это стек.

$x + y$ это в byte-коде:

iload_1 – загрузить в стек x

iload_2 – загрузить в стек y

iadd – сложить, два значения из стека



Что такое Byte-коды

Byte-коды – это инструкции для JVM, длиной в 1 байт.
Всего их может быть 256.

Примеры:

iload – загрузить в стек локальную переменную int

fstore – выгрузить из стека в локальную переменную float

ladd – сложить два long-а не стеке.

Операции для boolean, byte, char, short не определены, они приводятся к int.

Кстати, а какой размер boolean согласно спецификации JVM?

Byte-коды. Пример

Пример Byte-кодов:
class OneOne

ASM

ASM

ASM is an all purpose Java bytecode manipulation and analysis framework.

Asm используется в следующих проектах:

- OpenJDK
- Nashorn compiler
- Groovy compiler
- Kotlin compiler
- Cobertura и Jacoco
- CGLIB
- Gradle
- Spring

Aspect-oriented programming (AOP)

Варианты реализации:

- модификация класса

пример: aop.instrumentation

Еще парочка примеров

Добавляем метод set:

SetterDemo

Подменяем код:

SummatorDemo

Зачем разработчикам знать про Bytecode?

- Понимание фундаментальных основ JVM
- Понимание технических инструментов современных прикладных фреймворков.

Вопросы?

Остались вопросы?
petrelevich@yandex.ru

<https://github.com/petrelevich/jvm-digging/tree/master/bytecodes>

Спасибо
за внимание.