

Java Basic

Compilation, Constructor, this

Mentor: <....>

Разновидности JDK

- Oracle JDK
- OpenJDK (from oracle)
- AdoptOpenJDK (Eclipse)
- RedHat OpenJDK
- Amazon Corretto
- Dragonwell (Alibaba)
- Zulu (Azul, Azul provides an Alpine-comp. docker image)

Составные части JVM

- Загрузчик классов (Classloader)
- Сборщик мусора (Garbage Collector)
- интерпретатор
- JIT-компилятор
- компоненты управления потоками

Как работает Java

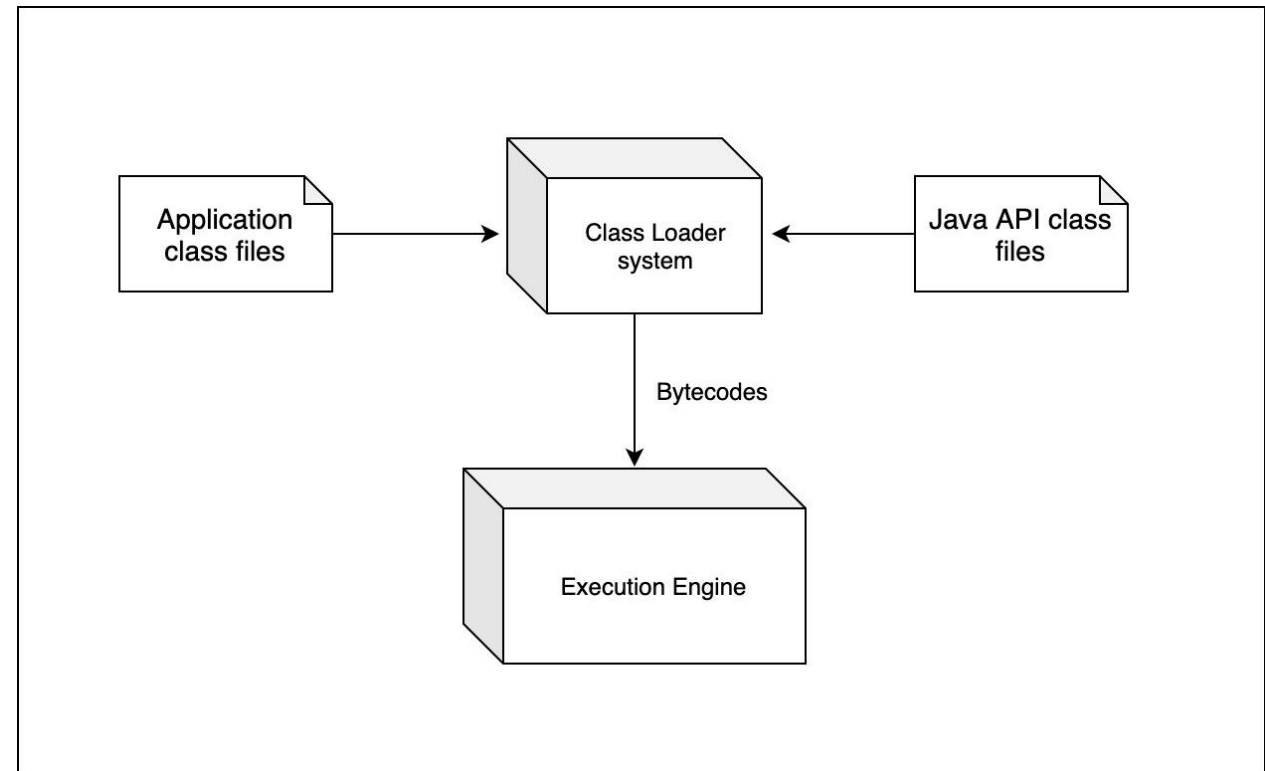
- Программист пишет код
- Компилятор создает байткод из написанных классов
- Classloader загружает байткод в память
- Verifier проверяет, что байткод ничего не нарушает
- Байткод запускается и если не первый раз, то из кэша

Загрузчик классов (Classloader)

[Bootstrap, Extension ClassLoader, AppClassLoader, собственный загрузчик]

Этапы загрузки классов (lazy loading + кеш)

- Loading (поиск и импорт бинарных данных для типа по его имени, создание класса или интерфейса из этого бинарного представления)
- Связывание / linking:
 - Верификация (проверка корректности импортируемого типа)
 - Подготовка / preparation (выделение памяти для статических переменных класса и инициализация памяти значениями по умолчанию)
 - Разрешение / resolution (преобразование символьных ссылок в прямые)
- Инициализация (вызов Java-кода, который инициализирует переменные класса их правильными начальными значениями)



Что внутри файла .class файла

- Signature (4 bytes) - 0xCAFEBAFE
 - Версия файла (4 bytes), major + minor versions
 - Пул констант. Таблица структур, представляющих строковые константы, имена классов, интерфейсов, полей, методов и другие константы, которые есть в структуре ClassFile и ее подструктурах.
 - Флаги доступа. Список флагов, которые указывают класс это или интерфейс, public или private, финальный класс или нет (ACC_ANNOTATION объявлен аннотацией, ACC_PUBLIC объявлен публичным)
 - This class. Ссылка на запись в пуле констант
 - Super class. Ссылка на запись в пуле констант
 - Интерфейсы. Количество реализаций в классе
 - Количество полей + сами поля.
 - Количество методов + сами методы (дескриптор, таблица исключений и т.д.)
 - Количество атрибутов + атрибуты
- Java 1.2 uses major version 46
 - Java 1.3 uses major version 47
 - Java 1.4 uses major version 48
 - Java 5 uses major version 49
 - Java 6 uses major version 50
 - Java 7 uses major version 51
 - Java 8 uses major version 52
 - Java 9 uses major version 53
 - Java 10 uses major version 54
 - Java 11 uses major version 55
 - Java 12 uses major version 56
 - Java 13 uses major version 57
 - Java 14 uses major version 58
 - Java 15 uses major version 59
 - Java 16 uses major version 60
 - Java 17 uses major version 61
 - Java 18 uses major version 62
 - Java 19 uses major version 63

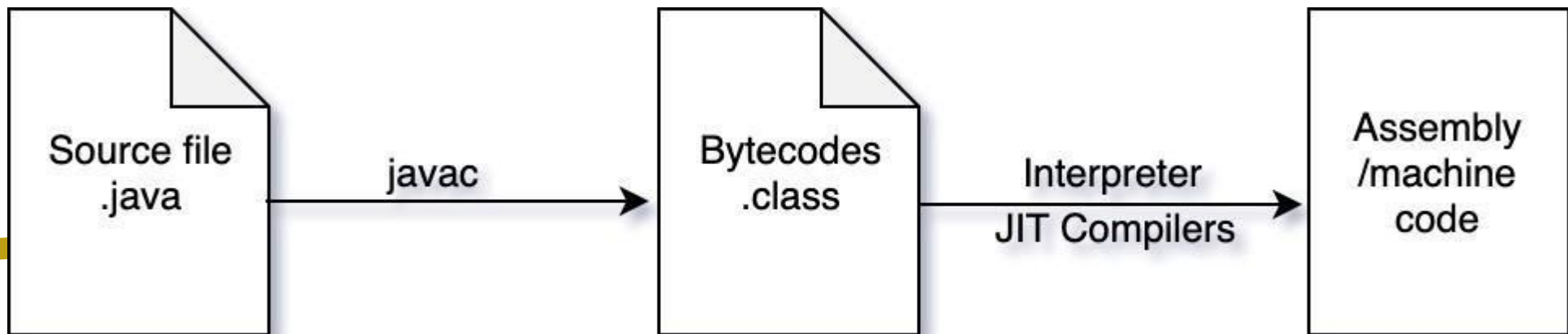
Компиляция

.java -> javac (Compiler) -> .class -> JVM (JIT, Interpreter, Runtime) -> output

JIT - Just-in-time (JIT) компиляция.

Interpreter - в роли интерпретатора выступает виртуальная машина, которая построчно преобразует байт-код в машинный код и тут же его исполняет.

Runtime and program output



Конструктор в Java

В Java конструктор — это блок кода, аналогичный методу. Он вызывается при создании экземпляра класса.

1. Конструкторы должны иметь то же имя.
2. Конструкторы не возвращают никакого типа.
3. Конструкторы вызываются только один раз во время создания объекта.

Ключевое слово "this"

Ключевое this слово относится к текущему объекту в методе или конструкторе.

"this требуется для того, чтобы метод мог сослаться на вызвавший его объект" (с)

Использование:

1. в случае, когда у переменной экземпляра класса и переменной метода одинаковые имена;
2. когда необходимо вызвать конструктор одного типа из другого (явный вызов конструктора)

Int age;

String name;

Cat(int age, String name) {

this.age = age;

this.name = name;

}

this

Использование ключевого слова this() в конструкторе для текущего объекта

```
public class Cat{  
    String name;  
    int age;  
    public Cat(String name, int age) {  
        this(name);  
        this.age = age;  
        System.out.println("В конструкторе с двумя параметрами");  
    }  
    public Cat(String name) {  
        this.name = name;  
        System.out.println("В конструкторе с одним параметрами");  
    }  
    public Cat() {  
        System.out.println("В конструкторе по умолчанию");  
    }  
}
```