

NCS - Java 001 Java 란?

1. Java 란

1. James Gosling이 개발을 시작해서 1995년에 공식 발표한 객체지향 개발언어 - OOP(Object Oriented Programming)

객체지향
(OOP)

현실 세계를 모델로 개체 사이에 메시지를 주고 받게 프로그래밍하는 기법입니다. 재 사용성이 높아지고 관리가 쉽다

캡슐화

여러 데이터와 기능을 하나로 만들어서 클래스 내부에 은닉한다
사용자는 인터페이스(Method 형태)를 통해 값을 주고 받는다

상속

이미 만들어진 객체의 기능을 다른 클래스에서 상속이라는 형태로 그대로 사용할 수 있게 해준다

다형성

상속 개념에서 자식 인스턴스를 생성할 때 부모의 타입에 할당할 수 있다. - 객체의 관리를 용이하게 한다

가비지
컬렉션

메모리 관리를 JVM(Java Virtual Machine) 에서 해준다
사용하지 않는 자원(메모리상의 객체)을 스스로 제거한다

플랫폼 독립

OS 의 종류와 상관없이 하나의 코드로 동작한다
Java 로 만들어진 코드는 window, iOS, Linux에서 모두 동일하게 동작한다

1. [부록1] James Gosling

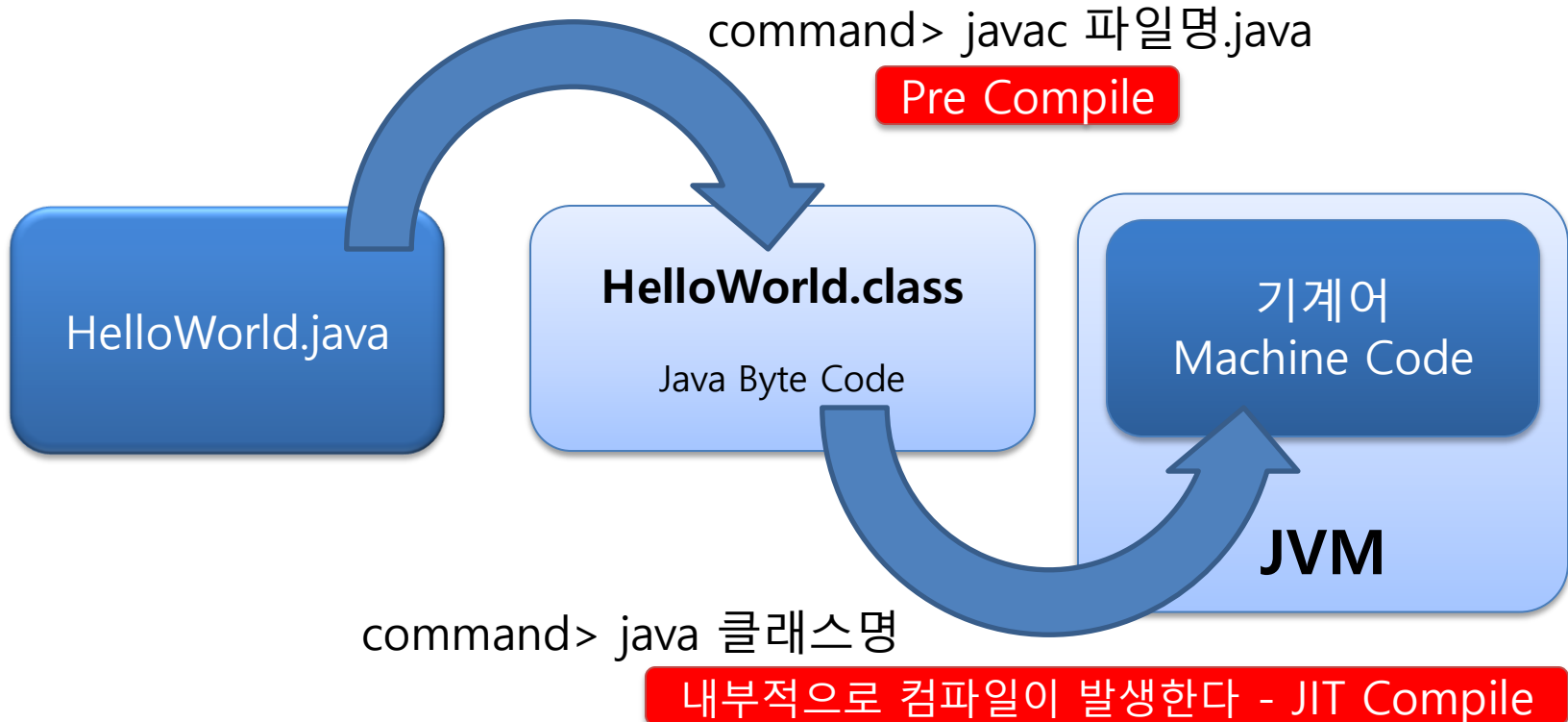
1. James Gosling 인터뷰

<https://www.youtube.com/watch?v=m2c-Tji4h-k>



2. Java 파일의 컴파일 순서

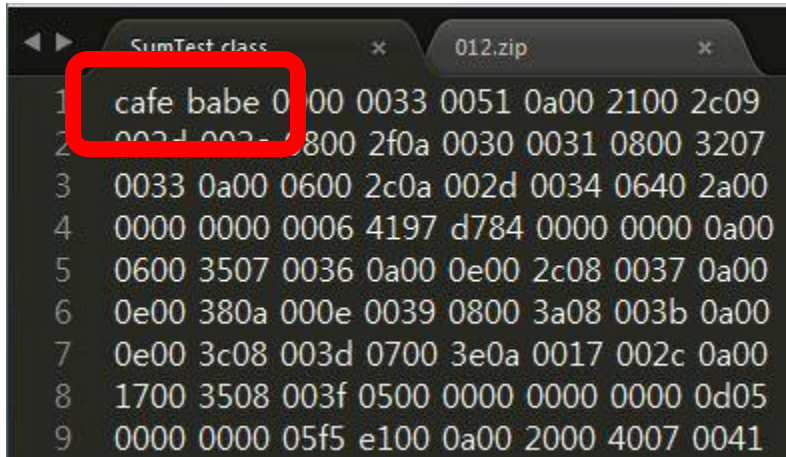
- 텍스트 형태의 [파일명.java 파일] 을
 - > Java Byte Code 형태인 [파일명.class] 로 Pre컴파일 한 후에
 - > 각각의 OS 에서 다시 [기계어] 로 컴파일 하여 실행한다



3. (.class) 파일의 Magic Number

- 모든 파일들은 해당 파일을 인식하기 위해 확장자 별로 Magic Number 를 가지고 있다. 실제로 OS 는 파일명과 함께 해당 Magic Number 로 파일의 종류를 구분한다 - (텍스트 에디터에서 Hex Code 로 확인할 수 있다)

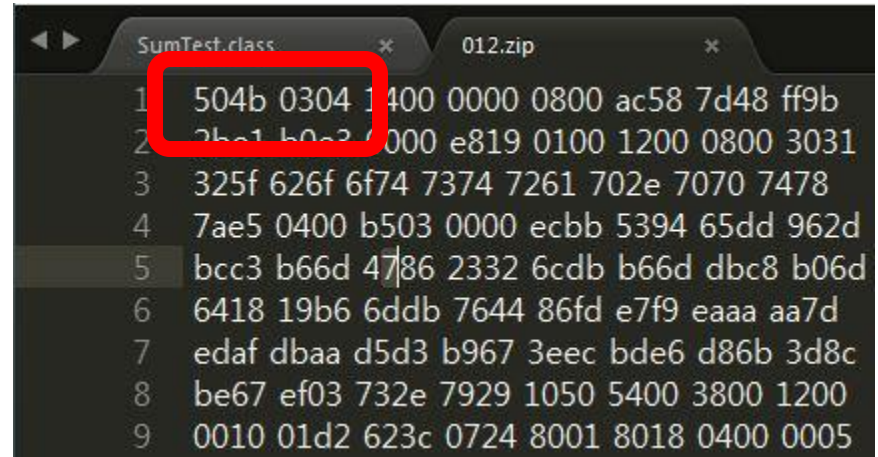
class 파일의 매직넘버



```
1 cafe babe 00 00 0033 0051 0a00 2100 2c09
2 002d 002e 0800 2f0a 0030 0031 0800 3207
3 0033 0a00 0600 2c0a 002d 0034 0640 2a00
4 0000 0000 0006 4197 d784 0000 0000 0a00
5 0600 3507 0036 0a00 0e00 2c08 0037 0a00
6 0e00 380a 000e 0039 0800 3a08 003b 0a00
7 0e00 3c08 003d 0700 3e0a 0017 002c 0a00
8 1700 3508 003f 0500 0000 0000 0000 0d05
9 0000 0000 05f5 e100 0a00 2000 4007 0041
```

café babe

zip 파일의 매직넘버



```
1 504b 0304 1400 0000 0800 ac58 7d48 ff9b
2 7b01 b0c2 0000 e819 0100 1200 0800 3031
3 325f 626f 6f74 7374 7261 702e 7070 7478
4 7ae5 0400 b503 0000 ecbb 5394 65dd 962d
5 bcc3 b66d 4786 2332 6cdb b66d dbc8 b06d
6 6418 19b6 6ddb 7644 86fd e7f9 eaaa aa7d
7 edaf dbaa d5d3 b967 3eec bde6 d86b 3d8c
8 be67 ef03 732e 7929 1050 5400 3800 1200
9 0010 01d2 623c 0724 8001 8018 0400 0005
```

504b 0304

4. (.class) Java Byte Code의 Layout

| byte offset | size | type or value | description |
|-------------|---------|---------------|---|
| 0 | 4 bytes | u1 = 0xCA hex | magic number (CAFEBABE) used to identify file as conforming to the class file format |
| 1 | | u1 = 0xFE hex | |
| 2 | | u1 = 0xBA hex | |
| 3 | | u1 = 0xBE hex | |
| 4 | 2 bytes | u2 | minor version number of the class file format being used |
| 5 | | | |
| 6 | 2 bytes | u2 | major version number of the class file format being used. Java SE 8 = 52 (0x34 hex), Java SE 7 = 51 (0x33 hex), Java SE 6.0 = 50 (0x32 hex), Java SE 5.0 = 49 (0x31 hex), JDK 1.4 = 48 (0x30 hex), JDK 1.3 = 47 (0x2F hex), JDK 1.2 = 46 (0x2E hex), JDK 1.1 = 45 (0x2D hex). |
| 7 | ... | ... | |

u1: an unsigned [8-bit](#) integer

u2: an unsigned [16-bit](#) integer in [big-endian](#) byte order

u4: an unsigned [32-bit](#) integer in big-endian byte order

5. Java 파일의 Runtime 컴파일

- Java 의 VM을 통한 실행 시 컴파일 방식에는 두 가지가 있다
JIT 방식으로 실행 시 최초 한번 컴파일 하는 방식과
AOT 방식으로 설치 시 최초 한번 컴파일 두 가지 방식이 있다

JIT

Just In Time

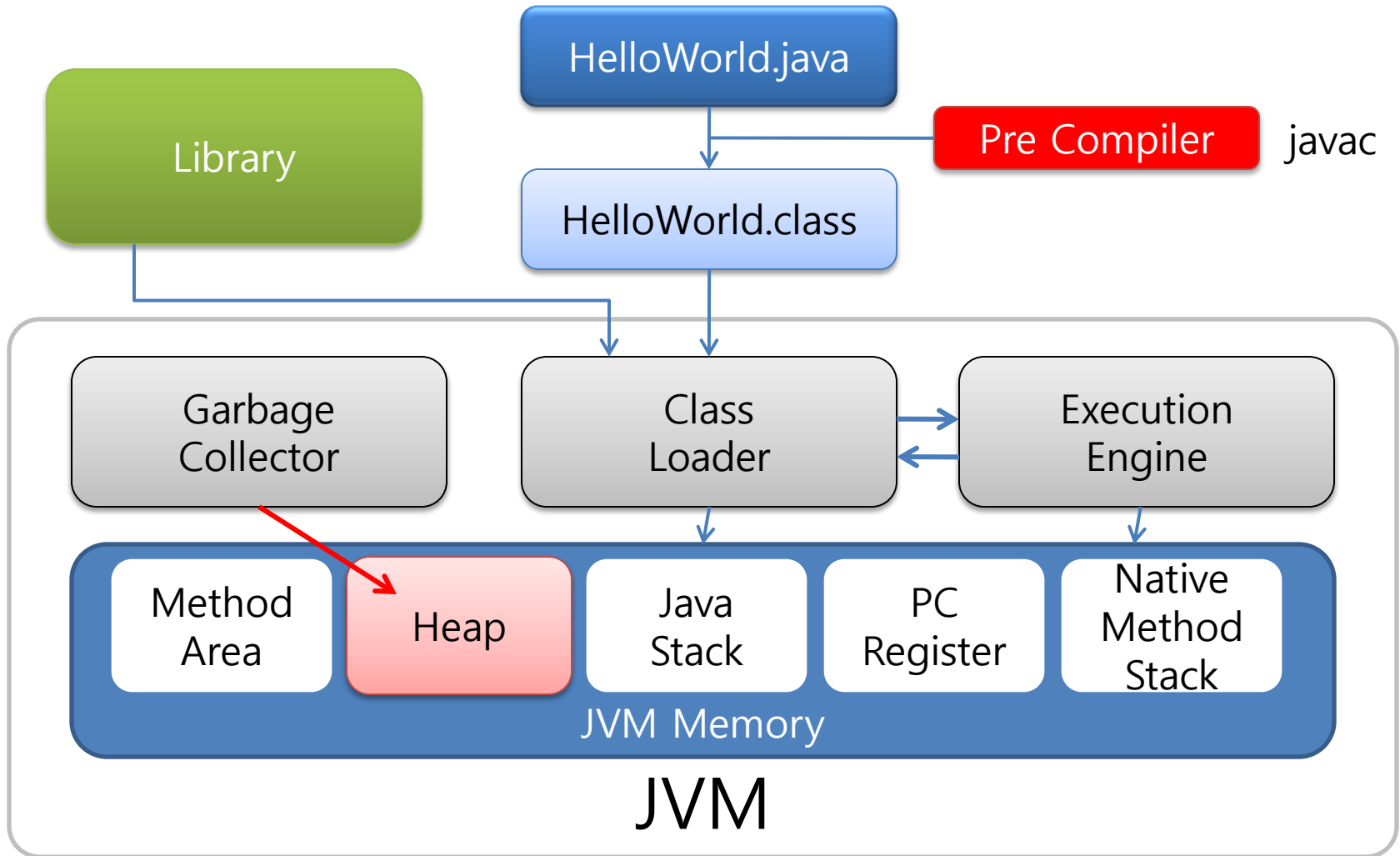
Class 파일을 실행 시 최초 한번 기계어로 컴파일 한다. 따라서 최초 한번은 속도의 저하가 발생할 수 있다

AOT

Ahead Of Time

Class 파일을 설치 시 최초 한번 기계어로 컴파일 한다. 속도 저하는 없지만 안드로이드 처럼 설치가 명확한 구조에서만 가능하다

6. Java 의 실행과정



7. JVM 의 메모리 관리 (Garbage Collection)

