# JVM은 무엇이며 자바 코드는 어떻게 실행하는 것인가.

## JVM 이란 무엇인가

- 기술적 정의 : 코드를 실행하고 해당 코드에 대해 런타임 환경을 제공하는 소프트웨어 프로그램에 대한 사양( Specification )이다.
- 일반적 정의 : 자바 프로그램을 실행하는 방법. JVM의 설정을 구성하고, 설정사항에 따라 실행 중에 프로그램 리소스를 관리한다.

## 컴파일 하는 방법

- 작성한 .java파일의 해당 경로를 찾아가서 javac 명령어에 java클래스 명을 작성한다. ex) javac HelloWorld.java

## 실행하는 방법

## 바이트코드란 무엇인가

- JVM이 이해할 수 있는 언어로 변환된 자바 소스 코드를 의미
- 자바 컴파일러에 의해 변환되는 코드의 명령어 크기가 1바이트라서 자바 바이트 코드라고 불리고 있습니다.
- 자바 바이트 코드의 확장자는 .class이다.
- 자바 바이트 코드는 자바 가상 머신만 설치되어 있으면, 어떤 운영체제에서도 실행될 수 있다.

## JIT 컴파일러란 무엇이며 어떻게 동작하는지

- JIT ( Just In Time ) 컴파일러는 바이트코드를 기계어로 변환하여 실행
- 인터프리터의 단점을 보완하는 역할, 반복되는 코드가 있다면 어느 시점에 전체 바이트 코드를 컴파일해 네이티브 코드로 변환한다.
- 이 네이티브 코드는 반복되는 메소드 호출에 사용되어 시스템 성능을 향상시킨다.

## JVM 구성 요소

JVM의 3가지 주요 구성 요소

1. 클래스 로더 ( Class Loader )
    - JAM 구성의 첫 번째 주요 요소이며, 클래스 파일을 Runtime Data Area에 적재하는 역할을 수행하고, 동적으로 클래스를 로딩한다.
    - 3 단계
        1. Loading
            1. 부트스트랩 클래스 로더(Bootstrap class loader) :

    2. 익스텐션 클래스 로더(Extension class loader) :
    3. 응용 프로그램 클래스 로더( Application class loader) :
2. Linking
    1. 확인( verify ) 단계 : 클래스 파일이 메모리에 로드될 때 확인 단계로 들어서며, 이 단계에서 표준에 적합한 경우 바이트코드가 확인된다.
    2. 준비( prepare ) 단계 : 메모리가 스태틱 변수에 할당되고, 디폴트 값이 할당된다.
    3. 해결( resolve ) 단계 : 모든 상징적 참조( symbolic references )가 실제 참조( actual references )로 교체된다.
3. 초기화( Initialization ) : 클래스 로더 요소의 초기화 단계에서, 모든 스태틱 변수들은 실제 값으로 할당되고, 이때 스태틱 이니셜라이저를 실행한다.

1. 런타임 데이터 에어리어 ( Runtime data area ) : 클래스 파일을 저장 및 실행
    1. METHOD AREA
    2. HEAP AREA
    3. STACK MEMORY
    4. PC REGISTER
    5. NATIVE METHOD STACK

1. JVM 엑씨큐션 엔진 ( JVM Execution Engine ) : 바이트 코드를 기계코드로 변환하고, 명령을 수행하는 실제 엔진이다.
    1. INTERPRETER : 클래스 파일 또는 바이트코드를 읽고, 하나 씩 실행한다. 인터프리터의 문제는 메서드가 여러 번 호출되면 반복해서 바이트 코드로 인식된다.
    2. JIT COMPILER
    3. GARBAGE COLLECTOR
    4. JAVA NATIVE METHOD INTERFACE

# JDK 와 JRE의 차이

| **JRE** | **JDK** |
|---|---|
| Java Runtime Environment | Java Development Kit |
| Java 프로그램을 실행하는 Java Virtual Machine( 자바 가상 머신 )을 구현하는 환경 | Java 기반 응용 프로그램을 개발하는데 사용할 수 있는 소프트웨어 번들 |
| Java 프로그램을 실행하는데 필요한 플러그인 | Java 프로그램 개발에 필요 |
| JRE는 JDK보다 크기가 작기 때문에 더 적은 디스크 공간이 필요 | JDK는 다양한 개발 도구와 JRE를 포함하므로 더 넓은 디스크 공간이 필요 |
| JRE는 JVM, 코어 라이브러리, 그리고 Java로 작성된 응용 프로그램과 애플릿을 실행하는 추가구성요소를 포함 | JDK는 JRE, API 클래스 집합, Java 컴파일러, Webstart 및 Java 애플릿과 응용 프로그램을 작성하기 위해 필요한 추가 파일을 포함 |

## Ref

Where can I get technical information about Java?
Java에 관한 기술 정보는 어디서 확인할 수 있읍니까?
자바 프로그램의 실행 과정
JVM ARCHITECTURE
JVM 동작원리 및 기본개념
자바 가상 머신 JVM(Java Virtual Machine

https://blog.naver.com/hsm622/222138523668
https://jeongjin984.github.io/posts/JVM/
https://github.com/league3236/startJava/blob/master/live_study/week1.md
https://sowhat4.tistory.com/61
https://github.com/yeo311/java-study-with-whiteship/tree/main/week1
https://gblee1987.tistory.com/173
https://github.com/kksb0831/Practice_project/blob/master/Java_Study_01.md
https://jul-liet.tistory.com/143
https://hooooooooooooop.tistory.com/entry/javahalle1
https://velog.io/@jaden_94/1%EC%A3%BC%EC%B0%A8-%ED%95%AD%ED%95%B4%EC%9D%BC%EC%A7%80
https://catch-me-java.tistory.com/9
https://catch-me-java.tistory.com/10
https://catch-me-java.tistory.com/11
https://catch-me-java.tistory.com/12?category=438116
https://catch-me-java.tistory.com/13
https://docs.google.com/presentation/d/1nTUGhSAQJnlcYTdyH4GXjHQq9w1vFByqAdG71Fbnesc/edit?usp=sharing
https://blog.naver.com/baekseongsa/222139174854
https://blog.naver.com/baekseongsa/222144525963
https://blog.naver.com/baekseongsa/222144528643
https://github.com/kyu9/WS_Study/blob/master/week1/day1.md
https://www.notion.so/1-01b08b753f0c4e93b16a96101c75d382
https://blog.naver.com/haena910/222150168303
https://github.com/Jason-time/java_study_with_whiteship/blob/main/week01.md
https://gintrie.tistory.com/61

jvm

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
**http://125.132.25.164/dokuwiki/doku.php?id=wiki:java:java-lecture:1week&rev=1615108320**

Last update: **2022/03/10 19:52**