

JVM 은 무엇이며 자바 코드는 어떻게 실행하는 것인가.

JVM 이란 무엇인가?

JVM (Java Virtual Machine)은 자바 애플리케이션을 실행하기 위한 가상 머신이다.

자바 바이트코드를 기계어로 변환해 실제 하드웨어에서 실행한다.

JVM 은 플랫폼에 독립적이며, 이를 통해 "한 번 작성하면, 어디에서나 실행"이라는 자바의 핵심 철학을 실현한다.

컴파일 하는 방법

자바 소스 코드는 `.java` 확장자로 저장된다. 이 코드를 바이트코드로 변환하기 위해서는 `javac` 라는 컴파일러를 사용한다.

```
$ javac HelloWorld.java
```

위의 명령어를 실행하면, `HelloWorld.class` 라는 바이트코드 파일이 생성된다.

실행하는 방법

바이트코드로 컴파일된 `.class` 파일은 `java` 명령어를 통해 실행할 수 있다.

```
$ java HelloWorld
```

바이트코드란 무엇인가?

바이트코드는 `javac` 에 의해 생성된 중간 코드로, JVM이 이해할 수 있는 언어이다.

바이트코드는 플랫폼이 독립적이므로, 다양한 환경에서 JVM만 있으면 실행될 수 있다.

컴파일러는 무엇이며 어떻게 동작하는가?

JIT (Just-In-Time) 컴파일러는 JVM 내에 내장된 컴파일러로, 자바 프로그램 실행 중에 바이트코드를 특정 플랫폼의 네이티브 코드로 동적으로 변환하며, 이를 통해 프로그램의 실행 속도가 향상된다.

JIT 컴파일러는 한 번 컴파일된 코드를 캐싱하여, 같은 코드가 다시 실행될 때 빠르게 처리할 수 있다.

JVM 구성 요소

클래스 로더(Class Loader): `.class` 파일을 메모리로 로드한다.

런타임 데이터 영역(Runtime Data Area): 프로그램 실행 중 필요한 데이터를 저장하는 영역이다.

실행 엔진(Execution Engine): 바이트코드를 실행하며, JIT 컴파일러도 포함하고 있다.

JDK와 JRE의 차이

JDK (Java Development Kit): 자바 개발을 위한 툴셋으로, 컴파일러(`javac`), 디버거 등의 개발 도구와 JRE를 포함하고 있다.

JRE (Java Runtime Environment): 자바 애플리케이션을 실행하기 위한 환경으로, JVM과 라이브러리 클래스들을 포함하고 있다.