

Understand Java Application Execute

Code

코드(code) 혹은 소스(source)

프로그램이 어떤 모습이고, 어떻게 동작해야 하는가를 표현한 일종의 설계도라고 할 수 있다.

Pre - Compile (.java → .class)

자바의 문법으로 구성된 코드는 사람만이 이해할 수 있는 형식으로 되어 있는데

이 코드를 **JVM이 이해할 수 있는 자바바이트코드로 변환해주는 과정**이 필요한데 이것을 **컴파일(compile)**이라고 한다.

- **Javac.exe (Java Compiler)** : Compile을 수행
- Command Prompt > **javac** FristJavaProgram.java → FristJavaProgram.class 파일 생성

JIT vs AOT Compile (Java 런처 → JVM)

- **JIT (Just In Time)** : 프로그램 최초 실행시마다 코드를 변환한다.
 - Android Runtime : Dalvik
- **AOT (Ahead Of Time)** : 주로 설치시에 한번에 전체를 변환해 두고 저장한 뒤, 프로그램 실행시 마다 변환된 코드를 읽어들이게 된다.
 - Android Runtime : ART
- Android 7.0 : AOT 컴파일러만 들어있던 ART 엔진에 JIT 컴파일러를 추가, App에 따라 각 방식을 유연하게 적용한다.

Execute

코드를 컴파일 하면 JVM이 이해할 수 있는 .class라는 확장자의 파일이 생성된다.

컴파일된 파일은 .class 파일을 JVM에게 전달하는 런처(launcher)를 통해 실행이 된다.

- **Java.exe (Java launcher)**
- Command Prompt > **java** FristJavaProgram (.class 생략)

Launcher (런처)

발사대, JVM에게 .class의 실행을 유도하는 역할

Flow Chart

