



자바의 역사와 JVM구조

자바의 역사

- **시작과 배경:** 1991년, 제임스 고슬링과 그의 팀은 가전제품에 적용될 프로그래밍 언어를 만들기 위해 '오크(Oak)'라는 언어를 개발했습니다. 처음에는 가전제품 제어를 목적으로 했습니다.
- **자바로 변경:** 1995년, 인터넷의 등장으로 오크는 더 넓은 활용을 위해 '자바(Java)'로 이름이 변경되었습니다. 이는 다양한 플랫폼에서 실행할 수 있는 장점이 있었기 때문입니다.
- **플랫폼 독립성:** "한 번 작성하면 어디서나 실행할 수 있다(Write Once, Run Anywhere)"라는 모토로, 자바는 다양한 운영체제(Windows, macOS, Linux 등)에서 작동할 수 있었습니다. 이는 기존 언어들과의 차별점이었으며, 자바의 큰 장점이 되었습니다.
- **오라클 인수:** 2009년, 오라클(Oracle)이 자바를 인수하여 현재까지 버전 업데이트와 관리가 이루어지고 있습니다.

2. 자바의 특징

- **플랫폼 독립성:** 다양한 운영체제에서 작동할 수 있는 장점.
- **JVM(Java Virtual Machine):** 자바의 플랫폼 독립성을 가능하게 해주는 핵심 요소로, 각 운영체제별로 JVM이 존재하여 바이트 코드를 기계어로 변환해 실행하게 해줍니다.

3. JVM(Java Virtual Machine) 구조

JVM은 자바 프로그램의 실행을 담당하는 가상 머신으로, 다음과 같은 주요 구성 요소로 이루어져 있습니다:

- **Execution Engine:** 바이트 코드를 기계어로 번역하여 실행하는 역할을 담당합니다.
- **Garbage Collector:** 사용하지 않는 객체를 자동으로 메모리에서 제거해주는 기능입니다.
- **Runtime Data Area:** 프로그램 실행 시 메모리를 관리하는 영역으로, 세부적으로는 Method Area, Heap Area, Stack Area가 있습니다.

4. 메모리 구조 (JVM의 메모리 영역)

- **Heap Area:** 동적 메모리 할당 공간으로, 객체와 배열이 저장되는 영역입니다.
- **Method Area:** 클래스 정보, 메서드, 상수, 스택틱 변수 등이 저장되는 영역입니다.
- **Stack Area:** 메서드 호출 시 사용되는 지역 변수와 매개변수, 메서드 호출 정보가 저장됩니다. 메서드 호출이 끝나면 메모리에서 자동으로 제거됩니다.

5. 자바 버전 현황

- **현재 버전:** 2024년 9월 기준 Java 17 이상이 가장 많이 사용되고 있습니다. 특히 Spring Boot 등의 최신 프레임워크는 Java 17 이상을 지원합니다.
- **과거 버전:** 이전에는 Java 7, 8, 11 등이 주로 사용되었습니다. 그러나 최근에는 더 이상 호환되지 않는 경우가 많으므로 최신 버전을 사용하는 것이 좋습니다.

6. 요약

자바는 가전제품에서 시작해 웹까지 확장된 범용적인 언어이며, 플랫폼 독립성과 JVM의 역할 덕분에 다양한 환경에서 사용될 수 있습니다. Java를 배울 때는 역사와 구조를 이해하면 이후 문법과 고급 문제 해결 시 큰 도움이 됩니다.