2025.1.1.수

Chapter 1. 자바를 시작하기 전에

01. 자바(Java)란?

\* 자바는 썬 마이크로시스템즈(Sun Microsystems, Inc.)에서 개발하여 1996년 1월에 공식적으로 발표한 객체지향 프로그래밍 언어이다. 2010년에 썬이 오라클사에 인수되면서 이제는 오라클사의 제품이 되었다.

\* 자바의 특징

- 운영체제(Operating System, 플랫폼)에 독립적 -> 운영체제의 종류에 관계없이 실행 가능 -> 다양한 기종의 컴퓨터와 운영체제가 공존하는 인터넷 환경에 적합한 언어로써 인터넷의 발전과 함께 많은 사용자층을 확보할 수 있었다.

- 비교적 배우기 쉽고 이해하기 쉬운 간결한 표현이 가능하도록 했다.

- 풍부한 클래스 라이브러리(Java API)를 통해 프로그래밍에 필요한 요소들을 기본적으로 제공한다.

02. 자바의 역사

\* 1991년 썬의 엔지니어들에 의해서 고안된 오크(Oak)라는 언어에서부터 시작되었다.

\* 제임스 고슬링, 아서 밴 호프는 썬의 엔지니어들로 가전제품에 탑재될 소프트웨어를 만들고자 했음. C++의 장점을 도입, 단점을 보완한 새로운 언어를 개발함.

\* Oak를 자바로 변경하고, 자바로 개발한 웹브라우저인 ‘핫 자바’를 발표하고 그 다음 해인 1996년 1월에 자바의 정식 버전을 발표함

\* 자바로 작성된 애플릿이 많은 인기를 누렸으나 보안상의 이유로 현재는 서블릿, JSP가 더 많이 사용되고 있음.

03. 자바의 특징

\* 운영체제에 독립적이다.

- 자바가상머신(JVM) 통해 가능함. 자바 응용프로그램은 운영체제나 하드웨어가 아닌 JVM하고만 통신하고 JVM이 자바 응용프로그램으로부터 전달받은 명령을 해당 운영체제가 이해할 수 있도록 변환하여 전달한다. 자바로 작성된 프로그램은 운영체제에 독립적이지만 JVM은 운영체제에 종속적이어서 썬에서는 여러 운영체제에 설치할 수 있는 서로 다른 버전의 JVM을 제공하고 있다. ‘한번 작성하면, 어디서나 실행된다. (Write once, run anywhere)’고 표현하기도 한다.

\* 객체지향언어이다.

- 객체지향 프로그래밍언어(object-oriented programming language). 상속, 캡슐화, 다형성이 잘 적용되어 있다.

\* 비교적 배우기 쉽다.

- 자바의 연산자와 기본구문은 C++에서, 객체지향관련 구문은 스몰톡(small talk)이라는 객체지향언어에서 가져왔다.

\* 자동 메모리 관리(Garbage Collection)

- 자바로 작성된 프로그램이 실행되면, 가비지컬렉터(garbage collector)가 자동적으로 메모리를 관리해주기 때문에 프로그래머는 메모리를 따로 관리하지 않아도 된다. 가비지컬렉터가 없다면 프로그래머가 사용하지 않는 메모리를 체크하고 반환하는 일을 수동적으로 처리해야 할 것이다.

\* 네트워크와 분산처리를 지원한다.

- 인터넷과 대규모 분산환경을 염두에 둔 까닭인지 풍부하고 다양한 네트워크 프로그래밍 라이브러리(Java API)를 통해 비교적 짧은 시간에 네트워크 관련 프로그램을 쉽게 개발할 수 있도록 지원한다.

\* 멀티쓰레드를 지원한다.

- 일반적으로 멀티쓰레드(multi-thread)의 지원은 사용되는 운영체제에 따라 구현방법도 상이하며, 처리 방식도 다르다. 그러나 자바에서 개발되는 멀티쓰레드 프로그램은 시스템과는 관계없이 구현가능하며, 관련된 라이브러리(Java API)가 제공되므로 구현이 쉽다. 그리고 여러 쓰레드에 대한 스케줄링(scheduling)을 자바 인터프리터가 담당하게 된다.

\* 동적 로딩(Dynamic Loading)을 지원한다.

- 보통 자바로 작성된 애플리케이션은 여러 개의 클래스로 구성되어 있다. 자바는 동적 로딩을 지원하기 때문에 실행 시에 모든 클래스가 로딩되지 않고 필요한 시점에 클래스를 로딩하여 사용할 수 있다는 장점이 있다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

04. 자바 가장 머신(JVM)

\* JVM은 ‘Java virtual machine’을 줄인 것으로 직역하면 ‘자바를 실행하기 위한 가상 기계’라 할 수 있다. 또는 ‘자바를 실행하기 위한 가상 컴퓨터’

\* 가상 컴퓨터는 실제 컴퓨터(하드웨어)가 아닌 소프트웨어로 구현된 컴퓨터라는 뜻으로 컴퓨터 속의 컴퓨터라고 생각하면 된다.

|  |
| --- |
| Java 애플리케이션 |
| JVM |
| OS(Windows) |
| 컴퓨터(하드웨어) |

|  |
| --- |
| 일반 애플리케이션 |
| OS(Windows) |
| 컴퓨터(하드웨어) |

\* 일반 애플리케이션의 코드는 OS만 거치고 하드웨어로 전달되는데 Java애플리케이션은 JVM을 한 번 더 거치기 때문에, 그리고 하드웨어에 맞게 완전히 컴파일된 상태가 아니고 실행 시에 해석(interpret)되기 때문에 속도가 느리다는 단점을 가지고 있다.

\* 일반 애플리케이션은 OS종속적. 반면 Java애플리케이션은 JVM하고만 상호작용을 하기 때문에 OS와 하드웨어에 독립적이라 다른 OS에서도 프로그램의 변경없이 실행이 가능한 것이다. 단 JVM은 OS에 종속적이기 때문에 해당 OS에서 실행가능한 JVM이 필요하다.

05. 자바 개발도구(JDK) 설치하기

\* JDK(Java Development Kit)를 설치하면 자바가상머신과 자바클래스 라이브러리 외에 자바를 개발하는데 필요한 프로그램들이 설치됨.

\* 오라클 계정 생성 필요함

06. 자바 개발도구(JDK) 설정하기

\* JDK의 bin폴더를 환경변수 path에 등록하는 것이 좋다. 이 폴더에는 자바로 프로그램을 개발하는데 필요한 실행파일들이 들어 있는데, 이 폴더를 path에 등록해 놓으면 실행파일을 실행할 때 일일이 경로를 입력하지 않아도 되어서 편리하다.

\* 환경 변수는 윈도우에서 사용하는 설정정보가 담겨있는 변수이다.

\* 제어판 -> 시스템 환경 변수 편집 -> 환경 변수 -> 시스템 변수 -> JAVA\_HOME, CLASSPATH, Path 다 추가 -> cmd창에서 java -version, javac -version으로 확인

<https://minstar0410.tistory.com/3>

\* 명령 프롬프트(Command Prompt, cmd.exe)는 Microsoft Windows의 기본 유틸리티로, Windows NT 계열 운영 체제의 명령줄 인터페이스(CLI)이다. 윈도우 명령어 처리기(Windows Command Processor)라고도 불린다.

07. 자바 API문서 설치하기

\* 자주 사용되는 것만을 공부한 다음 나머지는 영어사전처럼 필요할 때 찾아서 사용하면 된다.

\* url

java8 API문서 : <https://docs.oracle.com/javase/8/docs/api/>

자바와 관련된 유용한 내용 : <https://docs.oracle.com/javase/8/docs/>

Java8 API문서 다운로드: <https://www.oracle.com/java/technologies/javase-jdk8-doc-downloads.html>

08. 첫 번째 자바 프로그램 작성하기

\* java 파일이 존재하는 경로에서 javac, java 실행하기

텍스트, 친필, 문서, 종이이(가) 표시된 사진

자동 생성된 설명

**차이점: java Hello와 java Hello.java**

1. **java Hello**
   * 이 명령어는 컴파일된 클래스 파일(Hello.class)을 실행합니다.
   * 이때, **클래스 이름**만 입력해야 하며, 확장자 .java를 포함하면 안 됩니다.
   * 예를 들어, Hello.class가 현재 디렉토리에 있어야만 실행됩니다.
2. **java Hello.java (Java 11 이상에서 지원)**
   * 이 명령어는 \*\*직접 소스 파일(.java)\*\*을 실행합니다.
   * Java 11 이상에서는 컴파일과 실행을 한 번에 처리할 수 있는 기능이 추가되었습니다. 하지만 이 기능은 .class 파일을 생성하지 않으며, 임시로 실행합니다.

 java Hello는 컴파일된 Hello.class 파일을 실행하며, 클래스 이름만 입력해야 합니다.

 java Hello.java는 Java 11 이상에서 소스 파일을 직접 실행하는 기능으로, .class 파일 없이도 동작합니다.

 하지만, 권장 방식은 **javac Hello.java로 컴파일 → java Hello로 실행**하는 것입니다.

\* 자바에서 모든 코드는 반드시 클래스 안에 존재해야 하며, 서로 관련된 코드들을 그룹으로 나누어 별도의 클래스를 구성하게 된다. 그리고 이 클래스들이 모여 하나의 Java 애플리케이션을 이룬다.

\* public static void main(String[] args)는 main메서드의 선언부인데, 프로그램을 실행할 때 java.exe에 의해 호출될 수 있도록 미리 약속된 부분이므로 항상 똑같이 적어주어야 한다.

\* Java 애플리케이션은 main메서드의 호출로 시작해서 main메서드의 첫 문장부터 마지막 문장까지 수행을 마치면 종료된다.

텍스트, 책, 종이, 친필이(가) 표시된 사진

자동 생성된 설명09. 자바 프로그램의 실행과정

2025.1.4.토

10. 이클립스 설치하기

\* 이클립스는 자바 프로그램을 편리하면서도 빠르게 개발할 수 있는 통합 개발 환경(IDE, Integrated Development Environment)을 제공한다.

11. 이클립스로 자바 프로그램 개발하기

프로젝트 생성 -> 클래스 생성. 클래스명은 첫 글자 대문자로.

12. 이클립스의 뷰, 퍼스펙티브, 워크스페이스