1강

JAVA_PROGRAMMING

프로그래밍 언어

※프로그래밍 언어는 사람과 컴퓨터간에 의사소통을 위한 도구 이다.

< 컴퓨터 >

대화(의사소통)

< 사람 >

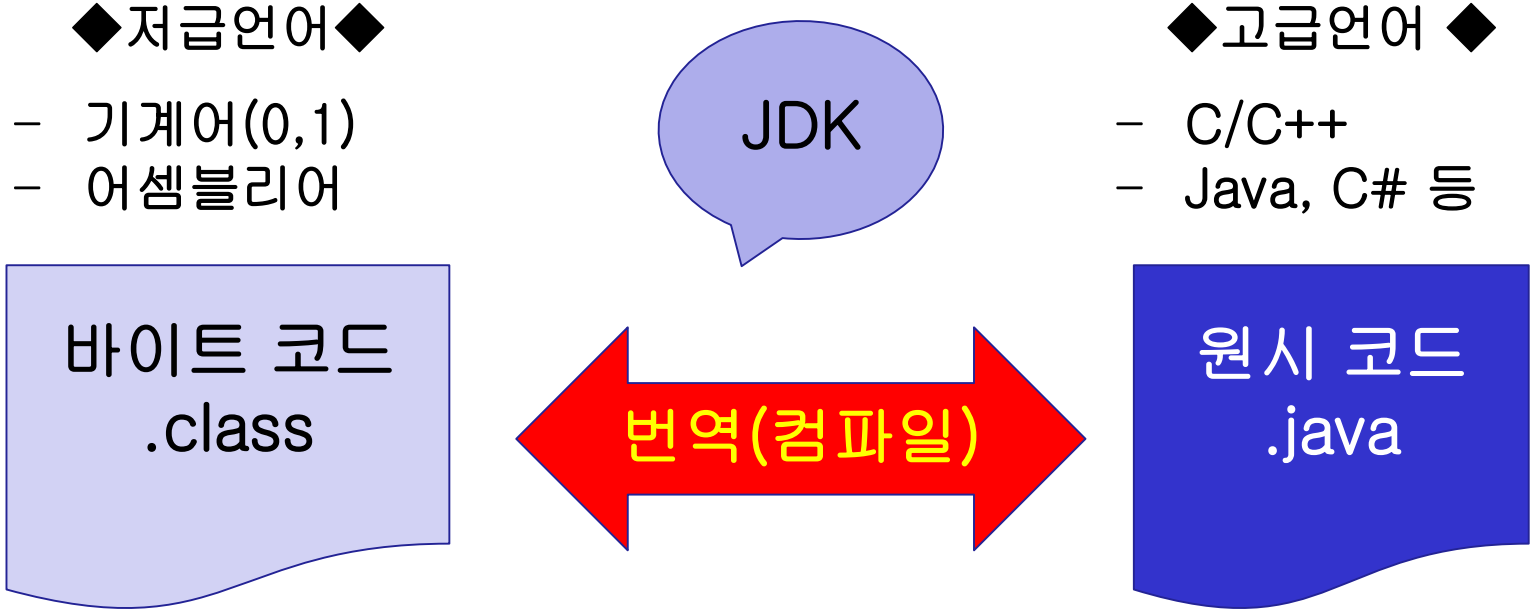
자바 프로그래밍 언어

◆저급언어◆

- 기계어(0,1)
- 어셈블리어

◆고급언어 ◆

- C/C++
- Java, C# 등



※실행 시 인터프리터방식으로 실행 (라인 별 실행)

언어의 역사

- ❖ 고급언어
  - 사람이 이해하기 쉬운 언어
- ❖ 저급언어
  - 컴퓨터가 이해하기 쉬운 언어
- ❖ 프로그래밍 언어의 발전 과정
  - ALGOL60 -> CPL(63) -> BCPL(70) -> B(70) -> C(73) -> C++(85) -> JAVA(95)
- ❖ 컴파일러
  - 원시 프로그램을 바이트 코드로 번역하는 도구
- ❖ 인터프리터
  - 라인 별로 한 줄씩 실행하는 방식

자바 개발 환경 설정

- JRE(Java Runtime Environment)
  - 사용자를 위한 실행 환경만을 제공
- JDK(Java Development Kit)
  - 실행환경과 반기계어로 변환하는 컴파일러 포함 환경제공(실습 환경)
- Download
  - 다음 사이트 접속 http://www.oracle.com/접속하여 OS에 맞는 jdk 다운로드
  - 다운로드 후 실행하여 설치 진행(중간에 jre 설치도 진행)
- **설치 후 환경변수 설정**
  - 시작 -> 설정 -> 제어판 -> 시스템 -> 고급 -> 환경변수 -> 시스템 변수 항목
  - 새로 만들기 -> JAVA_HOME = jdk설치 경로 지정
  - path에 추가 -> ;%JAVA_HOME%₩bin; 을 넣는다.(환경 변수 설정)
- **설치 후 테스트!**
  - cmd -> javac -version
- JavaDoc(도움말) 다운로드 및 Eclipse 준비

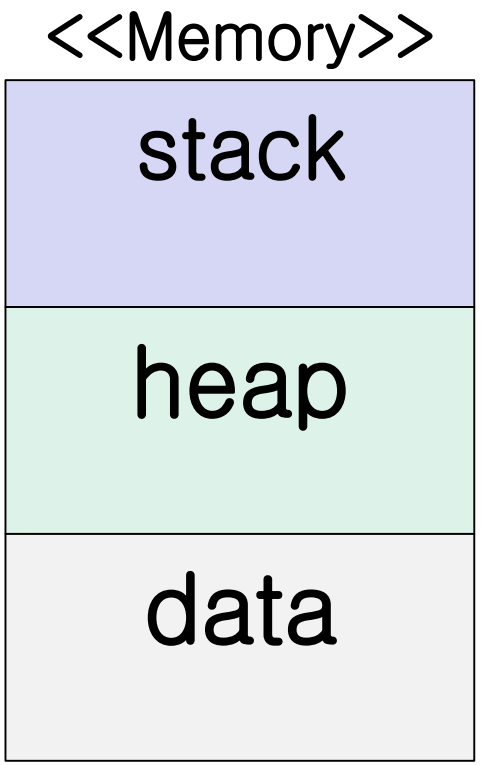
Java API 문서

❖ Document API 사용법 및 활용

- C:₩java₩jdk8₩docs₩api 폴더 안에 index.html 을 더블 클릭한다.
- 왼쪽 위가 package 영역
- 아래가 package안의 클래스 영역
- 오른쪽 가운데가 class에 대한 documentation 영역이다.
- class 구역 안에서 Ctrl + F키로 찾고자 하는 것을 찾으면 된다.
- java.lang 안의 String을 찾아서 확인해 본다.

메모리 구조

- ❖ 스택 영역(Runtime Stack)
  - 로컬변수를 할당, 선언된 블럭을 벗어나는 소멸됨.
  - 자동으로 초기화 되지 않으므로 초기화 해주어야 한다.
- ❖ 동적 할당 메모리영역
  - 배열로 만들어진 연속된 공간들이나 클래스에 포함된 멤버를 할당하는 영역.
  - Garbage Collector 에 의해 관리.(초기화가 디폴트)
- ❖ 상수와 Method 영역
  - 프로그램의 시작 시 생성, 종료 시 소멸.
  - 고정데이터를 가진 상수나 추상화 형식을 지닌 Method의 경우 해당 내용을 기억해 두어야 할 필요성 때문에 만들어진 영역.(실행 시 위치 값을 가지게 된다.)
- ❖ Register 영역
  - JVM을 실행하면서 필요한 부분의 주소를 기억해 두는 곳.
  - 각 영역에 연결되어 있는 프로세서가 작동할 때 여기에 등록된 내용을 기초로 실행



Hello Java!!

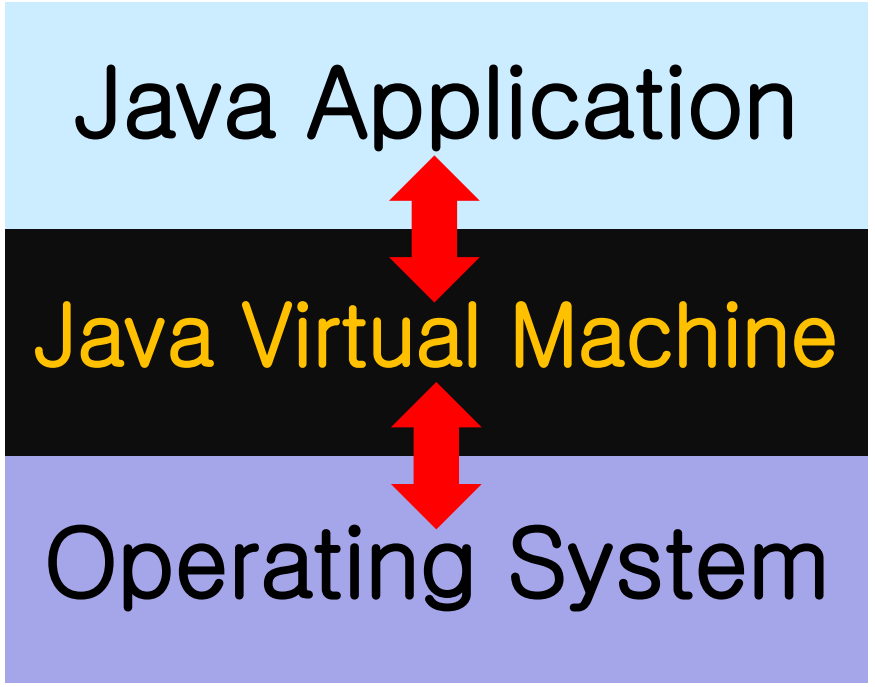
Hello Java 코딩 실습

기본 자바 코드의 모양을 이해 하자.

```
import java.lang.*;        //c,c++의 #include 같은 개념
public class HelloJava{
        public static void main(String []args)
        {
                System.out.println("Hello Java!!");
        }
}
```

자바의 특징

- 이식성이 높은 언어
- 외부의 포인터를 제거하고 내부 포인터를 사용
- 완벽한 객체지향 언어
- JVM(Java Virtual Machine)
- 바이트코드를 읽어서 실행 할 수 있도록 해주는 도구



주석문

❖ 주석이란

- 프로그램에 영향을 주지 않고 코드의 설명을 달아 놓을 수 있는 기능
- 다른 사람이 코드를 보고 이해하기 쉽도록 해주는 방법

- // 내용 ~
  - 한 줄 주석 : // 해당하는 라인의 끝까지만 주석
- /* ~ 내용 ~ */
  - 범위 주석 : /* 로 시작해서 */끝나는 곳까지가 주석
- /** ~ 내용 ~ */
  - 문서 주석 : 문서에 주석을 달아줄 때 사용