**[ 1 ] 자바 프로그램이란**

목표 : 자바 프로그래밍을 위한 설치 프로그램을 알고, 개발환경을 만들 수 있다.

자바 프로그램의 특징을 이해하고, 환경에 따른 프로그램 설치의 장∙단점을 안다.

1. 컴퓨터 프로그래밍 : 컴퓨터를 제어할 수 있는 명령어들의 집합

컴퓨터 + C or Java or C++ or Python = 언어를 이용해 P/G을 만든다

CPU(연산-일) 명령어(이 명령어는 컴퓨터가 이해하기 위해 **컴파일러**과정을 거친다)

메모리(연습장)

컴퓨터 프로그래밍 언어는 컴퓨터 프로그램을 만드는 것. 그럼 왜 Java인가?

* CUP 종속적 언어(컴파일된 프로그램이 특정 CPU에서만 실행 가능) ex. 어셈블리
* OS 종속적 언어(컴파일된 프로그램이 특정 OS에서만 실행 가능) ex. C, C++
* OS 독립적 언어(=플랫폼독립적언어, 컴파일된 프로그램이 가상머신을 통해 여러 OS에서 실행 가능) ex. java

(Java 인기 배경)

1. CPU 종속적 프로그래밍

Intel사 CPU AMD사 CPU

어셈블리P/G 어셈블리P/G **(탑재불가)**

1. CPU 독립적 프로그래밍 / OS 종속적 프로그래밍

Intel사 CPU AMD사 CPU Intel사 CPU

OS(Windows) OS(Windows) OS(MacOS)

a.exe a.exe**(탑재가능)** a.exe**(탑재불가)**

(a.c를 컴파일)

1. **OS 독립적 프로그래밍**

CPU CPU CPU CPU

OS(Windows) OS(Linux) OS(MacOS) OS(Unix)

가상머신 가상머신 가상머신 가상머신

a.class a.class**(탑재가능)** a.class**(탑재가능)** a.class**(탑재가능)**

(a.java를 컴파일)

가상머신(Java Virtual Machine)

(장점) OS 독립적. 객체지향 언어로 개발과 유지보수가 효율적으로 진행. 2005년부터 HW 퍼프먼스 증가로 각광받음

(단점) 속도 저하(현재는 딜레이를 거의 못 느낌)

* 컴퓨터 웹 프로그래밍 언어들의 예시

C -> PHP / Visual Basic -> ASP

Java -> JSP -> Spring Framework

Python -> dJango Framework, 플라스크, Fast API

R -> Shiny

1. Java 언어의 탄생과 특징
   * Java 언어의 탄생
     1. 1990년대 초에 Java가 탄생하였으나, 일반적으로 사람들이 알게 된 시기는 1990년대 말입니다.
     2. 제임스고슬링(James Gosling)이라는 사람이 가전제품의 효율적인 사용을 위한 객체지향 프로그램 언어로 JAVA를 만들었습니다.
     3. 하지만, 기존의 C/C++에 비해서 속도가 많이 느려 사람들로부터 외면을 받았습니다.
     4. 1990년대 말에서 2000년대에 들어오면서 인터넷의 등장으로 Java를 외면했던 사람들로부터 관심을 얻기 시작합니다.
     5. 현재 많은 웹 프로그래밍에 활용되고 있다.
     6. JAVA는 대표적으로 다음의 플랫폼을 이루고 있습니다.

JAVA **SE**(Stand Edition) : JAVA Application 개발 > 데스크탑에서 실행되는 응용PG 및 웹 프로그램 개발

* 1. Java 언어의 특징
     1. Virtual Machine – 운영체제(윈도우, 리눅스, 맥OS)의 영향을 받지 않는다.
     2. Garbage Collection – 개발자는 메모리 관리를 하지 않는다. JVM이 메모리 자동 관리
     3. Object Oriented Program(OOP) – 객체지향 언어로 개발과 유지보수를 효율적으로 진행 할 수 있다
     4. 다중 쓰레드를 지원한다.
     5. 기능이 지원 – 접근에 대한 권한을 변수로 처리 – Access Modifier(접근 제어자)

1. Java 프로그래밍 환경 설정과 실행방법
   * Java 개발 환경 설정
     1. JDK(Java Development Kit) 자바 개발 키트
     2. IDE(Integrated Development Environment); 통합 개발 툴(편집기) ex. 메모장, eclipse, IntelliJ
   * JDK 설치
     1. java.sun.com 또는 www.oracle.com 접속

(https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html)

* + 1. Downloads 에서 JDK 최신버전 다운로드 : **운영체제에 맞는 jdk**
       - 프로그램 설치에서 32/64는 무엇을 의미하고 jdk와 elipse 버전에 어떤 영향을 미치는가?

현재 설치되는 시스템의 운영체제 종류가 64비트 운영체제인지 32비트 운영체제인지에 따라 설치되는 파일이 다르다. 내 시스템의 종류에 맞지 않는 jdk나 eclipse를 설치하면 JVM이 제대로 구동되지 못할 것이다.

* + 1. 다운받은 파일을 install
       - JVM, API, JRE, JDK란?

윈도우 컴퓨터

리눅스 컴퓨터

Mac

컴퓨터

윈도우용 JVM

리눅스용 JVM

Mac용 JVM

컴파일된 파일

(.class)

프로그램 파일

(.java)

개발영역

실행영역

[d:/ai/source/01\_java]폴더에 Hello.java코딩

public class Hello {

public static void main(String[] args){

System.out.println("Hello, Java");

System.out.println("Welcome~^^");

}

}

/\* Java 프로그램의 실행단계별 내용

* + - * 1. Java 소스 파일(ex.d:/ai/source/1\_Java/Hello.java) 작성
        2. javac Hello.java (컴파일러로 바이너리 코드 생성)
        3. java Hello (JVM 구동명령어로 실행) \*/
  + **Java 환경 변수 설정**
    1. bin/javac.exe(컴파일러) : \*.java 파일을 JVM이 받아들일 수 있는 .class 파일로 변환시켜주는 프로그램. 매번 해당 경로에 들어가서 실행하기가 번거로우므로, 환경 변수 설정을 통해 어디서나 실행하도록 한다.

bin/java.exe(JVM) : \*.class 파일을 실행하도록 한다

위의 bin폴더에 있는 파일을 어디서나 샐행하기 위해 아래의 환경변수 설정을 한다

* + 1. 내PC 오른쪽 마우스 – 속성 - 고급시스템설정 - 환경변수
    2. 시스템변수 새로만들기-변수이름: JAVA\_HOME

변수값 : JDK설치된 폴더(C:\Program Files\Java\jdk17)

* + 1. Path에 추가할 내용 ; %JAVA\_HOME%\bin

1. 이클립스 다운로드 ; java 11을 설치한 경우

cf) https://soif0180.tistory.com/entry/Eclipse-이클립스-버전별-정보-및-JDK-최소-요구-버전

설치된 jdk 버전에 맞는 이클립스를 다운받아 압출을에서 다운로드 후 압축 푼다

(<https://www.eclipse.org/downloads/packages/release/2023-06/r> )

선택 사항 : ini파일 맨 마지막에 -Dfile.encoding=utf-8 추가

* + eclipse에서 workspace는 어떤 역할을 하는가?

작업환경이 저정된다. // 한줄 주석

/\* 여

\* 러

\* 줄

\* 주

\* 석

\*/

**public** **class** Hello { /\* 자바 파일은 반드시 클래스로 구성되어 있습니다 첫문자는 영문대문자 \*/

// public class Hello

// 파일의 이름과 동일한 클래스를 선언하는 룰

// public class 는 반드시 파일명과 동일하게 선언해야만 합니다.

// 하나의 소스 파일에는 하나의 public class를 선언할 수 있습니다.

// 특정 범위는 중괄호를 사용해서 명시적으로 선언합니다.

**public** **static** **void** main(String[] args) {

System.***out***.println("Hello, Java!!");

System.***out***.println("Welcome~~^^");

/\* System은 Java에서 제공하는 클래스 (JRE)

\* dot(.)는 접근연산자 = “System이라는 JRE안에” 라는 뜻

\* out(출력관련한 부분 제공)

\* println() printf() print() 등의 메소드를 제공한다

\* println() : 매개변수를 화면에 출력하고 자동 개행하는 기능

\* 매개변수란 메소드 괄호 내부에 입력된 값

\*/

} // 프로그램 종료 지점

}