**[ 1 ] 자바프로그램이란**

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 컴파일러과정을 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(컴파일된 프로그램이 가상머신을 통해 여러 OS에서 실행 가능) ex. java

1. 처음으로 프로그램을 접하는 시점에서
   * 옆사람을 존중하는 자세를 가져야 한다
   * 100% 완벽한 프로그램은 없다 꾸준히 리팩토링하고 종료된 프로젝트의 경우 개인적으로 최적화된 구조에 대해 연구하자.
   * 초기 개발자는 토끼와 거북이 중 거북이가 되자
2. 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 Server 프로그램이나 웹 프로그래밍에 활용되고 있다.
     6. JAVA는 대표적으로 3가지의 플랫폼을 이루고 있습니다.
        1. JAVA **SE**(Stand Edition) : JAVA Application 개발 > 데스크탑에서 실행되는 응용PG 개발
        2. JAVA ME(Mobile Edition) : 휴대 가능한 소형 디바이스에 사용을 목적으로 개발 > 현재의 스마트폰 등작으로 쇠퇴
        3. JAVA EE(Enterprise Edition) : 기업 솔루션, 서버와 클라이언트 PG 개발 목적
   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

원도우

리눅스

맥OS

Virtual Machine

Virtual Machine

Virtual Machine

**컴파일된 java program**

Virtual Machine을 통한 단점으로는 속도저하(현재는 딜레이를 잘 못 느낌)

* + 1. Object Oriented Program(OOP) – 객체지향 언어로 개발과 유지보수를 효율적으로 진행 할 수 있다
    2. 분산 네트워크 기술을 지원 : 원격 메소드 호출(RMI : Romote Method Invocation), CORBA(Common Request Broker Architecture) : 네트워크 상 다른 computer에 있는 서버를 java 프로그래밍으로 연결해서 처리하는 것.
    3. 다중 쓰레드를 지원한다. (ex) 네트워크 상에 한번에 여러 클라이언트 접속. 하나의 프로세스 안에 여러 단위의 요청 내용을 처리
    4. 보안 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
   * JDK 설치
     1. Java.sun.com 또는 www.oracle.com 접속
     2. 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. 이는 jdk도 Eclipse도 마찬가지다. 바이너리 코드로 컴파일된 클래스 파일을 시스템에 맞는 JVM이 실행하게 될 때 플랫폼 독립적이라 할 수 있다. 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

맥킨토시

컴퓨터

윈도우용 JVM

리눅스용 JVM

맥킨토시용 JVM

컴파일된 파일

(.class)

컴파일안된 프로그램 파일

(.java)

개발영역

실행영역

[c:/mega-IT/source/1\_java]폴더에 Hello.java코딩

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.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. 컴퓨터 오른마우스-속성-고급시스템설정-환경변수
    2. 새로만들기-변수이름:JAVA\_HOME /변수값 : JDK설치된 폴더
    3. Path에 ;%JAVA\_HOME%\bin; 추가

1. 이클립스 다운로드(<http://www.eclipse.org>)
   * 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 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[] agrs) {

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

/\* System은 Java에서 제공하는 클래스 (JRE)

\* dot(.)는 접근연산자

\* out(출력관련한 부분 제공)

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

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

\* 매개변수란 메소드 괄호 내부에 입력된 값

\*/

} // 프로그램 종료 지점

}