5. 17. 프로그램과 자바 기초

학습목표

1. 프로그램과 프로그램언어란 무엇인가?
2. 프로그램 언어로서 자바가 가진 특징은 무엇인가?
3. 자바 언어를 사용하기 위해서는 어떻게 해야 할까?
4. 자바 언어는 어떤 구조로 동작하는가?
5. 자바의 기초 문법, 변수와 자료형이란 무엇인가?

1. 프로그램과 프로그램 언어

컴퓨터: 전자회로를 이용해 다양한 종류의 데이터를 처리하는 기기. 논리, 연산을 수행.

하드웨어: 컴퓨터에서 물리적인 기계장치, 몸체 그 자체

소프트웨어: 하드웨어에 대응하는 용어. 컴퓨터를 활용하기 위한 프로그램.

프로그램: 컴퓨터가 작업하기 위한 명령어의 리스트. 순차작동.

프로그래밍 언어: 프로그래밍을 하기 위해 사용하는 언어.

프로그래밍 언어의 두가지 종류:

저급언어: 기계수준에 가까움. 기계에게 직접 지시를 내림. 기계어, 어셈블리어 등.

고급언어: 인간친화적 언어. 인간이 사용하기 용이한 형태. C, 파이썬, JAVA 등.

고급 언어는 인간이 작성한 “소스 코드”를 번역하여 기계가 읽을 수 있는 “오브젝트 파일”로 변환하는 과정을 거침. 이것을 컴파일 이라고 함.

2. 자바의 작동원리와 자바의 특징

자바는 인간이 작성한 “소스코드”를 javac와 같은 컴파일러로 “바이트 코드”를 생산함. 자바가상머신(JVM)이 이를 사용하는 운영체제에 맞추어 해석, 실행함.

이것이 자바의 가장 큰 특징. JVM은 많은 운영체제에서 지원되기 때문에 한번 작성한 코드를 윈도우, 맥, 리눅스 등 여러 곳에서 실행할 수 있음.

이 JVM 덕에 자바는 컴퓨터구조에 중립적이라는 특징을 가짐. 컴퓨터 환경이 달라도 JVM이 동작하는 환경이라면 똑 같은 프로그램을 작동시킬 수 있음. 이식성이 좋음.

또한 객체지향적이다, 비교적 단순하고 강력하다, 등의 장점이 있음.

3. 자바 언어의 사용을 위한 세팅

JRE: Java Runtime Environment. 자바프로그램을 실행하기만을 원하는 일반인을 위한 환경 제공.

JDK: Java Development kit. 자바 프로그램을 개발하는데에 필요한 개발도구. JRE가 포함됨. 이 JDK를 다운받아 세팅해야 함.

4. 자바 언어의 동작 구조

(1) 인간이 에디터(메모장 등) 혹은 통합개발환경(이클립스 등)을 이용하여 소스코드 생성. .java의 확장자를 가짐.

(2) 작성된 소스파일을 자바 컴파일러(javac)로 컴파일하여 바이트코드 생성. 이는 .class 의 확장자를 가짐.

(3) 이후 JVM이 클래스를 적재하고, 바이트코드를 검증한 뒤 실행함.

5. 자바의 기본 문법, 변수, 자료형.

import java.util.Scanner;

public class Hello {

// Hello 라는 클래스 지정. 자바는 클래스의 모임으로 구성됨. 클래스의 첫글자는 대문자.

    public static void main(String args[]) {

// 메인 메소드(method) - 프로그램의 시작점. 메소드는 프로그램이 해야하는 작업을 받아 수행하고 결과를 내보내는 작은 기계.

        System.out.println("Hello World");

// 문장(statement) 사용자가 작업을 지시하는 단위. 문장의 끝은 세미콜론.

// 메소드 안의 문장은 순차진행함. 이것을 절차수행이라고 함.

    }

}

변수: 프로그램이 사용하는 데이터를 저장하기 위한 메모리공간. 변수는 “자료형”과 “변수이름”으로 선언. (Ex: int x; ) “초기화”해줘야 함.

자료형: 저장하는 데이터값의 타입을 나타냄. String(문자열), Int(정수), double(실수) 등이 있음.

변수 이름: 변수를 구분하기 위함. 식별자의 일종. 영소문자로 시작해야 하며 그 뒤에는 숫자도 올 수 있음. 공백을 사용할 수 없어 단어를 구분할 때 -, \_, 영대문자 등을 씀. 의미가 있는 키워드는 변수 이름으로 사용 불가.(ex: Int)

초기화: initialize. 변수에 제일 초기에 값을 집어넣는 것. 변수를 사용할 때엔 꼭 해야 함. 하지 않으면 에러가 남.