

절차지향(Procedural Programming)

절차지향(Procedural Programming)이란, 물이 위에서 아래로 흐르는 것처럼 순차적인 처리가 중요시되며 프로그램 전체가 유기적으로 연결되도록 만드는 프로그래밍 기법입니다.

절차지향(Procedural Programming) 특징

- 대표적인 예로는 C언어가 있습니다.
- 컴퓨터의 작업 처리 방식과 유사하기 때문에 객체지향 언어를 사용하는 것에 비해 더 빨리 처리 되어 시간적으로 유리합니다.

단점으로는

- 실행순서가 정해져 있어 코드의 순서가 바뀌면 동일한 결과를 보장하기 어렵습니다.
- 유지보수가 어렵습니다.

이러한 절차지향언어의 단점을 보완하기 위해 객체지향의 개념이 등장합니다.

객체지향(Object Oriented Programming)

객체지향 프로그래밍에서는 데이터와 절차를 하나의 덩어리로 묶어서 생각하게 됩니다. 이는 마치 컴퓨터 부품을 하나씩 사다가 컴퓨터를 조립하는 것과 같은 방법입니다.

우리가 익히 들어알고 있는 자바, c++, 파이썬 이외에도 최초의 객체지향언어인 시뮬라67, 비주얼 베이직 닷넷, 액션스크립트, 스위프트 등 그 종류는 다양합니다.

객체지향(Object Oriented Programming) 특성

- 캡슐화(Encapsulation)- 객체를 캡슐로 싸서 그 내부를 보호하고 볼 수 없게 하는 것으로 객체의 가장 본질적인 특징입니다.
- JAVA로 예를 들면, 클래스(Class)는 객체의 모양을 선언한 틀이며, 클래스 모양 그대로 생성된 실체(Instance)가 객체가 됩니다. 자바는 필드(Field)와 Method(Method)를 클래스 내에 모두 구현하고 캡슐화를 통해 객체 내 필드에 대한 외부로부터의 접근을 제한합니다.

(이해를 돕자면 붕어빵 틀이 클래스가 되고, 붕어빵이 객체가 된다고 생각하시면 됩니다!)

●상속(Inheritance)- 상위 개체의 속성이 하위 개체에 물려 져서, 하위 개체가 상위 개체의 속성을 모두 가지는 관계입니다.

●JAVA로 예를 들면, 자식 클래스가 부모 클래스의 속성을 물려받아 부모 클래스에 기능을 확장(Extends)하는 개념입니다. 이때 자바에서 부모 클래스를 슈퍼 클래스(Super Class)라고 부르며 자식 클래스를 서브 클래스(Sub Class)라고 부릅니다.

●다형성(Polymorphism)- 같은 이름의 메소드가 클래스 혹은 객체에 따라 다르게 구현되는 것을 말합니다.

●JAVA로 예를 들면, 슈퍼 클래스에 구현된 메소드를, 서브 클래스에서 자신의 특징에 맞게 동일한 이름으로 다시 구현하는 이른바 메소드 오버라이딩(Overriding)으로 부르고, 클래스 내에서 같은 이름의 메소드를 여러 개 만드는 메소드 오버로딩(Overloading)이 있습니다.

장점

- 코드의 재활용성이 높습니다.
- 코딩이 절차지향보다 간편합니다.
- 디버깅이 쉽습니다.

단점

- 처리속도가 절차지향보다 느립니다.
- 설계에 많은 시간소요가 들어갑니다.

자바 프로그래밍 구조

소스파일 - 클래스 - 메서드 - 실행문

1. 소스파일

확장명 .java

public class 이름과 같아야 함

자바는 중괄호({ }) 블록 형태

2. 클래스 (붕어빵 틀)

프로그램을 개발하는 단위

유사한 특징을 지닌 객체들의 속성을 묶어 놓은 집합체

적어도 하나의 클래스가 있어야함

클래스 이름은 대문자로 시작하는 것이 관례

3. 메소드 (붕어빵)

수행할 작업을 나열한 코드의 모음

자바 어플리케이션은 main으로 실행

3-1. 메소드 예시

- 클래스 텔레비전
- 메소드 음량 조절, 채널 변경, 전원 ON/OFF의 등 기능적인 것들의 모음

4. 실행문

작업을 지시하는 변수 선언, 값 저장, 메소드 호출 등을 실행하기 위해 적는 코드

4-1. 출력

① System.out.println(); ② System.out.print(); ③ System.out.printf();
모든 코드는 **;(세미콜론)으로 마무리!**

5. 주석문

프로그램에 덧붙이는 설명

프로그램 실행에 관여X

- ① 행 주석 //
- ② 범위 주석 /* */
- ③ 문서 주석 /** */ (자바의 문서를 만들 때 사용)

변수와 상수

변수

프로그램이 실행되면 메모리를 사용하여 정보를 기억시킨다.

메모리에 변수를 할당하고 필요할 때마다 가져와서 사용

컴퓨터는 모든 신호를 숫자로 인지(문자라도 모두 0과1로 나타냄)

-> 변수 = 변하는 수(덮어쓸 수 있다.)

상수

변할 수 없는 수

변수 앞에 final을 붙여주면 상수로 변함.

-> 값이 변하면 큰일나는 경우에 사용

여러가지 변수 타입

문자 변수

String 타입은 원래 변수가 아니고, class 변수이다.

<String 과 char의 차이점>

String: " " 사이에 문자열 적어줌 -> 한 글자, 여러 글자 모두 가능!

char: ' ' 사이에 문자 적어줌 -> 한 글자만 사용가능!

*string 도 문자 하나를 표현할 수 있지만, 가장 작은 단위로 해주는 것으로 설계되어 있어서 보통 한 글자를 표현할 때는 char을 사용해준다.

*붉은 색으로 표시된 타입(ex. char)은 예약어라서 변수이름으로 쓸 수 없다.

정수 변수

표현할 수 있는 범위 byte < short < int < long

*long은 int의 범위까지는 똑같이 나타낼 수있지만 더 커진다면 숫자뒤에 L을 붙여주어야함.

*메모리를 할당해주는 것이기 때문에 엄격하게 관리

실수 변수

표현 범위: float < double

*정수에 비해 용량을 많이 차지 하기 때문에 정수만 쓸 거면 정수 변수

를 사용하는 게 좋음

*float는 끝에 f를 써주어야 한다.

논리 변수

논리 값은 boolean 타입 사용

true / false 두가지로만 표현

true는 1 / false 는 0 표현