6장 객체지향 프로그래밍

\*객체간의 관계

-집합관계 :

-사용관계(use-a) : 논리적인 관계로 멤버가 아닌 다른 클래스의 객체를 메소드의 인자로 받아 사용하거나 메소드 내에서 다른 클래스의 객체를 생성하여 사용하는 경우

-상속관계(is-a) : 학생클래스는 사람클래스를 상속받아 정의할수있음 클래스선언시 학생 extends 사람()

-포함관계(has-a) : 물리적인관계로 객체가 다른객체를 멤버로유지하는경우**부분전체**(part-whole) 관계와 **연관**(association) 관계로 나뉜다. 부분전체 관계는 다시 **집합**(aggregation)과 **복합**(composition)으로 나뉜다. **부분전체**(part-whole) 관계와 **연관**(association) 관계로 나뉜다. 부분전체 관계는 다시 **집합**(aggregation)과 **복합**(composition)으로 나뉜다.

\*다형성

같은타입이지만 다양한 객체를 대입할수있음

\*하나의 클래스안에 다른 클래스를 선언할수있으나 public으로 선언 불가

\*new()연산자를 통해 객체를 힙 메모리영역에 생성한후 객체 생성 번지를 리턴한다

\*생성자: 객체생성시 초기화 역할 담당 ClassName(){}

\*클래스의 용도

-라이브러리API Application Program Interface용

자체적으로 실해되지않으며

다르클래스에서 이용할 목적으로 만든 클래스

-실행용

main메소드를 가지고있는 클래스로 실행할 목적으로 만든 클래스

\*필드사용

-객체 내부에서는 변수명 = 으로 접근

-객체 외부에서는 객채명.변수명 = 으로 접근할수 있다

\*생성자 생성 보통 필드값을 설정해줄 때 필요로한다

->오버로딩을(매개변수타입, 개수, 순서를 달리하여 생성)하여 여려매개변수로 접근 가능하다

Class Car(){

Stirng name;

String ssn;

public Car(String name, String ssn){

this.name = name; (this.name은 필드의 name을 가리킨다)

this.ssn = ssn;

}

Public Car(String n, String s){

name = n;

ssn = s;

}

으로 생성자를 생성시킬수있다

}

-생성자 오버로딩시

Car(String model){

**this**(model, **null**, 0); //일일히 this.model = model;을 할필요없음

}

Car(String model, String color){

**this**(model, color,0);

}

Car(Stirng model, String color, **int** maxSpeed){

**this**.model = model;

**this**.color = color;

**this**.maxSpeed = maxSpeed;

}

-메소드에 매개변수 여러 개를 넣을수있게 오버로딩

**int** sum(**int** ... values) {

**int** sum = 0;

**for**(**int** i=0;,i<values.length();, i++) {

sum +=calues[i];

}

**return** sum;

}

-리턴값을 여러 개하는방법

Java bean을 만들어서 그 안에 a,b,c를 담아서 보내거나, 배열, Hash Map, Object(개체)에 담아서 보낼수있음

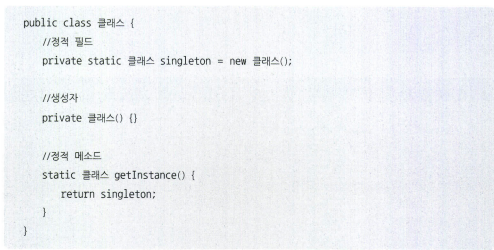
-싱글톤(Singleton)

프로그램에서 단하나의 객체만 만들도록 보장해야하는경우 ->클래스 외부에서 new 연산자로 생성자를 호출할수 없도록 막아야한다-> 클래스 내에서 객체 생성시 private 클래스() {}로 생성자를 표현

정적필드는 private static 클래스 singleton = new 클래스(); private접근제한자를 붙여

외부에서 필드값을 변경하지 못하도록 막고 대신 외부에서 호풀할수 있는 정적메소드인 getInstance()를 선언하고 정적필드에서 참조하고 있는 자신의 객체를 리턴

->클래스 변수1 = 클래스.getInstance();로 singleton객체를 불러올수있다



-static final 타입 상수(대문자로 표기하는것이관례)

불변의값을 설정할 때 static final로선언하여 상수를 정의

-접근제한자 모드 필드,생성자,메소드 에서 선언가능

Public 모든패키지에서 접근가능 (클래스선언가능)

Protected 같은 패키지내에서 접근가능

Default 패키지에 소속된 클래스에서만 사용할 수 있는 멤버를 만든다(클래스선언가능)

Private 외부에노출되지않고 내부 클래스에서만 접근가능

-String.valueOf(), toString() 차이

valueOf(변수명) : 대상값이 null이면 “null”을 만들어 반환한다

toString() : 대상값이 null이면 NPE를 발생시키고 Object에 담긴값이 String이 아니여도 출력한다