**[ 9 ] 접근제한과 static**

**오늘의 워밍업1)** 객체 두 개를 생성하여 다음과 같이 출력하는 프로그램을 구현하시오

/\* 결과 : 이름 = 홍길동 나이 = 20 성별 = m

\* 이름 = 홍길순 나이 = 19 성별 = f \*/

**package** com.oop.janmuri;

**class** PersonInfo {

**private** String name;

**private** **int** age;

**private** **char** gender;

**public** PersonInfo(){ }

**public** PersonInfo(String name){

**this**.name = name;

}

**public** PersonInfo(**int** age){

**this**.age = age;

}

**public** PersonInfo(**char** gender){

**this**.gender = gender;

}

**public** PersonInfo(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**public** PersonInfo(String name, **int** age, **char** gender) {

**this**.name = name;

**this**.age = age;

**this**.gender = gender;

}

**public** **void** print(){

System.***out***.println("이름:"+name+"\t나이="+age+"\t성별="+gender);

}

**public** **void** setName(String name) {**this**.name = name;}

**public** **void** setAge(**int** age) {**this**.age = age;}

**public** **void** setGender(**char** gender) {**this**.gender = gender;}

}

**public** **class** PersonInfoTest {

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

PersonInfo dong = **new** PersonInfo("홍길동",20,'m');

PersonInfo sun = **new** PersonInfo("홍길순");

sun.setAge(22);

sun.setGender('f');

dong.print();

sun.print();

}

}

**오늘의 워밍업2)** 다음과 같은 결과가 나오도록 클래스를 정의하고 객체를 생성하도록 프로그램을 구현하세요

주차장(ParkingSystem) 클래스 설계

주차장 들어올 때 결과 : 11라1111님 어서오세요

입차시간 : 12시

주차장 나갈 때 결과 : \*\*\* 11라1111님 \*\*\*

\* 입차시간 : 12시

\* 출차시간 : 14시

\* 주차요금 : 4000원

**package** com.oop.janmuri;

**class** Parking {

**private** String no;

**private** **int** inTime;

**private** **int** outTime;

**private** **int** fee;

**public** Parking(){ }

**public** Parking(String no, **int** inTime){

**this**.no = no; **this**.inTime = inTime;

System.***out***.println(no+"님 어서오세요");

System.***out***.println("입차시간 : "+inTime+"시");

}

**public** **void** outPrint(){

fee = (outTime - inTime) \* 2000;

System.***out***.println("\*\*\*\*\* "+no+" 님 안녕히 가세요 \*\*\*\*\*");

System.***out***.println("입차시간 : "+inTime);

System.***out***.println("출차시간 : "+outTime);

System.***out***.println("주차요금 : "+fee);

}

**public** **void** setNo(String no) {

**this**.no = no;

}

**public** **void** setInTime(**int** inTime) {

**this**.inTime = inTime;

}

**public** **void** setOutTime(**int** outTime) {

**this**.outTime = outTime;

}

}

**public** **class** ParkingTest {

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

Parking car1 = **new** Parking("11루1111",8);

Parking car2 = **new** Parking("22루2222",9);

car1.setOutTime(10);

car1.outPrint();

car2.setOutTime(11);

car2.outPrint();

Parking car3 = **new** Parking();

car3.setNo("33루3333");

car3.setInTime(11);

}

}

**오늘의 워밍업3)** 클래스를 이용하여 다음 값을 입력하고 다음과 같이 출력

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

성 적 표

------------------------------------------------------

이름 국어 영어 수학 총점 평균

--------------------------------------------------

정우성 90 90 90 270 90.0

김하늘 80 80 80 240 80.0

황정민 70 70 70 210 70.0

강동원 70 70 70 210 70.0

유아인 70 70 90 230 76.7

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

**public** **class** Student {

**private** String name;

**private** **int** kor;

**private** **int** eng;

**private** **int** mat;

**private** **int** tot;

**private** **double** avg;

**public** Student() { }

**public** Student(String name, **int** kor, **int** eng, **int** mat) {

**this**.name = name;

**this**.kor = kor;

**this**.eng = eng;

**this**.mat = mat;

tot = kor+eng+mat;

avg = tot / 3.0;

}

**public** **void** show() {

System.***out***.printf("%s\t%d\t%d\t%d\t%.1f\n",name, kor, eng, mat, tot, avg);

}

**public** String infoString() {

**return** String.*format*("%s\t%d\t%d\t%d\t%.1f\n",name, kor, eng, mat, tot, avg);

}

// 필요한 getter & setter

}

**package** com.oop.student;

**public** **class** StudentTestMain {

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

*line*('■',50);

System.***out***.println(" 성 적 표");

*line*('-',50);

System.***out***.println("이름\t국어\t영어\t수학\t총점\t평균");

*line*('-',50);

Student jung = **new** Student("정우성", 90, 90, 90);

Student kim = **new** Student("김하늘", 80, 80, 80);

Student hwang = **new** Student("황정민", 70, 70, 70);

Student kang = **new** Student("강동원", 70, 70, 70);

Student yu = **new** Student("유아인", 90, 90, 90);

jung.show();

kim.show();

hwang.show();

kang.show();

yu.show();

*line*('■',50);

}

**private** **static** **void** line(**char** c, **int** i) {

**for**(**int** cnt=0 ; cnt <i ; cnt++)

System.***out***.print(c);

System.***out***.println();

}

}

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

Student[] st = **new** Student[5];

st[0] = **new** Student("정우성",90,90,91);

st[1] = **new** Student("김하늘",99,80,81);

st[2] = **new** Student("황정민",90,95,95);

st[3] = **new** Student("강동원",99,99,99);

st[4] = **new** Student("유아인",90,90,92);

*line*('■');

System.***out***.println("\t\t성 적 표");

*line*('-');

System.***out***.println("이름\t국어\t영어\t수학\t총점\t평균");

*line*('-');

**for**(Student s : st){

s.calnShow();

}

*line*('■');

}

**private** **static** **void** line(**char** c) {

**for**(**int** i=0 ; i<45 ; i++)

System.***out***.print(c);

System.***out***.println();

}

* **멤버변수 선언**

[접근제한자] [static/final] 데이터형 변수명;

① [접근제한자] 접근제한이란 클래스의 데이터나 메소드에 대해서 다른 클래스로부터 접근을 제한하는 것.

public : 접근을 제한하지 않는다. 다른 모든 클래스에서 사용 가능

protected : 해당 클래스와 동일한 패키지에 있거나 상속받은 클래스일 경우에만 사용 가능

default : 접근제어자를 명시하지 않은 경우의 디폴드 접근제어자. 같은 패키지내의 클래스들은

public 권한을 갖고 접근가능

private : 해당 클래스만이 이 멤버를 사용할 수 있다. 외부객체에서는 절대로 접근할 수 없다.

**public**

**protected**

**default**

**private**

**package** com.tj.ex;

**public** **class** Woman {

**public** Woman(){

privateTest();

System.***out***.println("Woman 클래스 생성자");

}

**private** **void** privateTest(){

System.***out***.println("private 접근제한자");

}

**void** defaultTest(){

System.***out***.println("디폴트 접근제한자");

}

**public** **void** publicTest(){

System.***out***.println("public 접근제한자");

}

}

**package** com.tj.ex;

**public** **class** WomanTestMain {

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

Woman w = **new** Woman();

//w.privateTest();

w.defaultTest();

w.publicTest();

}

}

**package** com.tj.ex.another;

**import** com.tj.ex.Woman;

**public** **class** MainTestClass {

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

Woman w = **new** Woman();

//w.privateTest();

// w.defaultTest();

w.publicTest();

}

}

② [static/final]

final : 종단변수

public class FinalEx {

public static void main(String[] args){

final double PI = 3.141592;

double r = 3;

double area = PI\* r \* r;

double round = 2 \* PI \* r;

System.out.println("반지름 " + r+"인 원의 넓이는 "+ area);

System.out.println("반지름 " + r+"인 원의 둘레는 "+ round);

}

}

static : 클래스변수(=static 변수)

; 객체변수(객체속성)는 객체가 생성될 때마다 각 객체안의 속성 변수들이 생성되지만,

클래스 변수는 클래스로부터 생성된 객체들의 수와 상관없이 하나만 생성된다.

한 클래스로부터 생성된 모든 객체들은 클래스 변수를 공유

클래스 변수를 이용하여 객체들 사이의 통신에 사용하거나 객체들의 공통속성을 나타낼 수 있다.

객체변수와 달리 클래스 변수(static 변수)는 클래스 이름을 통하여 접근한다.

1. static의 이해

; 객체들끼리 공유하고 싶은 정보가 있을 때 사용. 일명 깍두기.

필요에 의해 클래스를 만들고, 사용할 때는 생성자를 통해 객체로 만들어서 사용을 합니다.

이러한 방식으로 모든 것을 처리하다 보면, 문제가 생길 수 있습니다.

예를 들어 집안에 부모, 그리고 세 아이가 있습니다.

엄마 지갑에 100원짜리 2개가 있습니다.

이러한 상황에서 아이들이 엄마한테 100원을 달라고 합니다.

당연히 엄마지갑에는 100원 동전 2개가 있으니, 줄 수 있습니다.

첫째 아이가 먼저 엄마 지갑 객체를 생성하여 100원을 받았습니다.

그리고 둘째 아이가 엄마 지갑 객체를 생성하여 100원을 받았습니다.

마지막으로 셋째 아이가 엄마 지갑 객체를 생성하여 100원을 받았습니다.

여기까지 보면, 뭔가 이상 합니다. 분명 집안에 엄마도 한 명이고, 지갑도 한 개인데, 아이들 모두 엄마 지갑을 생성하여 100원씩을 받았습니다.

200원 밖에 없는데 아이들이 받은 돈은 모두 300원입니다. 버그입니다.

그럼 이런 상황에서 생각할 수 있는 방법은 무엇일까요?

아주 쉽게 생각해 보면 엄마 지갑 객체가 세 아이들 모두에게 공유되면 됩니다.

그러면 첫째 아이가 100원 받고, 둘째 아이가 100원을 받으면, 지갑에는 동전이 없고, 셋째 아이는 100원을 받을 수 없게 됩니다.

셋째 아이한테는 서운한 일이지만, 어째든 프로그램적으로는 이상이 없이 정상 동작 할 것입니다.

그러면 어떻게 엄마 지갑을 모든 아이들이 공유할 수 있을까요?

바로 ‘static’ 키워드를 사용하는 것입니다.

우선 ‘static’ 키워드를 ‘공유(깍두기)’ 라고 생각 합니다.

<잘못된 예>

**public** **class** MomPouch {

**public** **int** MONEY = 200;

}

**public** **class** FirstChild {

MomPouch momPouch = **new** MomPouch();

**public** FirstChild() {

// **TODO** Auto-generated constructor stub

}

**public** **void** takeMoney(**int** money){

**if**(momPouch.MONEY>=money){

//MomPouch.MONEY = MomPouch.MONEY - money;

momPouch.MONEY -= money;

System.***out***.println("첫째가 "+money+" 받음");

}**else**{

System.***out***.println("첫째는 돈을 못 받아요");

}

}

}

**public** **class** SecondChild {

MomPouch momPouch = **new** MomPouch();

**public** SecondChild() {

// **TODO** Auto-generated constructor stub

}

**public** **void** takeMoney(**int** money){

**if**(momPouch.MONEY>=money){

System.***out***.println("둘째 "+money+"받음");

momPouch.MONEY -= money;

}**else**{

System.***out***.println("둘째 돈 못 받음");

}

}

}

**public** **class** ThirdChild {

MomPouch momPouch = **new** MomPouch();

**public** ThirdChild() {

// **TODO** Auto-generated constructor stub

}

**public** **void** takeMoney(**int** money){

**if**(momPouch.MONEY>=money){

System.***out***.println("셋째 "+money+"받음");

momPouch.MONEY -= money;

}**else**{

System.***out***.println("셋째 돈 못 받음");

}

}

}

**public** **class** MainTestClass {

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

FirstChild c1 = **new** FirstChild();

SecondChild c2 = **new** SecondChild();

ThirdChild c3 = **new** ThirdChild();

//System.out.println("엄마지갑:"+MomPouch.MONEY);

c1.takeMoney(100);

c2.takeMoney(100);

c3.takeMoney(100);

//System.out.println("엄마지갑 : "+MomPouch.MONEY);

}

}

<올바른 예>

**public** **class** MomPouch {

**public** **static** **int** *MONEY* = 200; // 객체가 생성되기전 클래스의 값 공유

}

**public** **class** FirstChild {

**public** FirstChild() { }

**public** **void** takeMoney(**int** money){

**if**(MomPouch.*MONEY* < money){

System.***out***.println("첫째 돈 못 받아요");

}**else**{

MomPouch.*MONEY* -= money;

System.***out***.println("첫째 "+money+"원 받았어요");

}

}

} **public** **class** SecondChild {

**public** SecondChild() { }

**public** **void** takeMoney(**int** money){

**if**(MomPouch.*MONEY* < money){

System.***out***.println("둘째 돈 못 받아요");

}**else{**

MomPouch.*MONEY* -= money;

System.***out***.println("둘째 "+money+"원 받았어요");

}

}

}

**public** **class** ThirdChild {

**public** ThirdChild() { }

**public** **void** takeMoney(**int** money){

**if**(MomPouch.*MONEY* < money){

System.***out***.println("셋째 돈 못 받아요");

}**else{**

MomPouch.*MONEY* -= money;

System.***out***.println("셋째 "+money+"원 받았어요");

}

}

}

**public** **class** MainTestClass {

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

FirstChild c1 = **new** FirstChild();

SecondChild c2 = **new** SecondChild();

ThirdChild c3 = **new** ThirdChild();

System.***out***.println("엄마지갑:"+MomPouch.*MONEY*);

c1.takeMoney(100);

c2.takeMoney(100);

c3.takeMoney(100);

System.***out***.println("엄마지갑 : "+MomPouch.*MONEY*);

}

}

* 또 다른 예제를 한번 살펴보자.

**class** Person{

**private** String name;

**private** **int** money;

**public** **static** **int** *comm\_money*;

**public** Person(String name) {

**this**.name = name;

}

**public** **void** currentMoney(){

System.***out***.println(name+"님 자산 : "+money);

System.***out***.println("공용 자산 : "+*comm\_money*);

}

**public** **void** saveMoney(**int** money) {

**this**.money += money;

}

}

**public** **class** A11\_Static {

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

Person p1 = **new** Person("홍길동");

Person p2 = **new** Person("이길동");

Person p3 = **new** Person("김길동");

p1.saveMoney(1000);p1.saveMoney(1000);p1.*comm\_money* +=1000;

p2.saveMoney(1000);p2.saveMoney(1000);p2.*comm\_money* +=1000;

p3.saveMoney(1000);p3.saveMoney(1000);p3.*comm\_money* +=1000;

p1.currentMoney();

p2.currentMoney();

p3.currentMoney();

}

}

* 또 다른 예제를 한번 살펴보자.

**class** Product{

**static** **int** *count=100*; // 생성된 인스턴스의 수를 저장하기 위한 static 변수

Private **int** serialNo; // 인스턴스 고유의 번호

**public** Product() { serialNo= ++*count*;}

**getter들 추가**

}

**public** **class** ProductTest {

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

Product p1 = **new** Product();

System.***out***.print("p1의 SN : "+p1.getSerialNo()+"\t");

System.***out***.println("p1의 count:"+p1.*count*);

Product p2 = **new** Product();

System.***out***.print("p1의 SN : "+p1.getSerialNo()+"\t");

System.***out***.print("p2의 SN : "+p2.getSerialNo()+"\t");

System.***out***.println("p1의 count:"+p1.*count*);

Product p3 = **new** Product();

System.***out***.print("p1의 SN : "+p1.getSerialNo()+"\t");

System.***out***.print("p2의 SN : "+p2.getSerialNo()+"\t");

System.***out***.print("p3의 SN : "+p3.getSerialNo()+"\n");

System.***out***.println("p1의 count:"+p1.*count*);

System.***out***.println("p2의 count:"+p2.*count*);

System.***out***.println("p3의 count:"+p3.*count*);

System.***out***.println("Product의 count:"+Product.*count*);

}

}

* static변수도 있지만 static 메소드도 있다

1. static과 객체와의 관계

※ static 변수를 클래스 변수(정적변수, 공용변수)라 부릅니다.

객체는 클래스에서 생성(복사)되어 진다고 배웠습니다.

클래스 하나에서 무한대로 객체를 생성할 수 있고 그러한 객체는 모양은 동일할지 모르지만, 전혀 다른 덩어리 입니다.

하지만 static이 붙은 변수는 객체 변수가 아닌 클래스 변수로써 객체가 생성되기 전에 이미 존재합니다. (객체를 생성하지 않고 클래스 내 변수를 사용) 객체 생성과 상관없이 사용할 수 있는 필드와 메서드에 활용

메모리 관련 얘기를 좀 해보겠습니다.

┌──────────┓ ┌───────────┓

│ 클 래 스 들 이 │ │ 객체들이 모여있음 │

│ 모 여 있 음 │ │(new를 이용해 생성된) │

└──────────┘ └───────────┘

데이터 영역 heap 영역 → 가비지 콜렉션 관리영역

(가비지 콜렉션이 관리하지 않는 영역)

자바에서는 개발자가 메모리 관리를 하지 않는다. 다 사용한 메모리를 회수하고 사용할 수 있게 하는 것은 JVM이 가비지 콜렉션한다.

⑶ static의 장점과 단점

※static은 잘 사용하면 좋고, 그렇지 않으면 큰 재앙을 불러올 수 있습니다

static 키워드를 사용한 변수는 객체생성과는 상관없이 클래스와 함께 존재하게 됩니다. 즉 객체 생성을 하지 않고, 메모리에 상주하게 됩니다. 이렇게 객체를 생성하지 않고도 존재한다는 것은 Heap 영역은 사용하지 않지만, 데이터 영역의 메모리를 사용한다는 것이다.

지금은 간단한 예제를 살펴보고 있기 때문에 객체생성에 따른 메모리의 부하를 느낄 수 없지만, 프로젝트가 커지면 객체생성에 따른 메모리 부족현상이 있을 수 있습니다. 물론 가비지 콜렉터가 열심히 일을 하고는 있어도 부족현상이 생길 수 있습니다. ▼ 사용중인 객체는 메모리에 계속 상주하고 있고, 또 데이터 영역은 가비지 콜렉터의 관리영역이 아닙니다.

객체 생성을 하지 않고도 사용할 수 있는다는 것은 분명 장점이지만, 한편으로 생각해 보면 가비지콜렉터의 관리 밖에 있기 때문에 항상 메모리에 상주해 있습니다. 즉 프로젝트가 커지고, 시스템이 오랜 시간 동안 돌아가게 되면 시스템 운영속도가 점차 느려지다가 급기야 큰 재앙이 올 수도 있습니다. 따라서 static의 사용은 신중해야 됩니다.

static 변수로 사용되는 대표적인 예를 아마도 상수(변하지 않는 수) 일 것입니다. 만약 프로그래밍을 하다가 원주율이 빈번이 사용되어야 한다면 이러한 값은 상수(final 변수)로 정의해 놓고 사용하는 편이 좋습니다.

원주율은 3.141592653589793238462643383279… 끝이 없습니다.

이러한 경우 원주율을 3.141592653589793238462643383279까지로 정의해 놓고 사용하자고 팀원들 사이 약속을 합니다.

**package** com.ch.sf;

**public** **class** PiClass {

**public** **static** **final** **double** ***PI*** = 3.141592653589793238462643383279;

}

**public** **class** MainClass {

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

System.***out***.println("원주율:"+PiClass.***PI***);

//PiClass.PI = 3.14; // final 클래스 변수(상수)이므로 변경 불가

}

}

③ 멤버변수의 초기화 시기와 순서

※클래스 변수의 초기화 시점 : 클래스가 처음 로딩될 때 단 한번

※인스턴스변수의 초기화 시점 : 인스턴스(객체)가 생성될 때 마다

**public** **class** Test1 {

**private** **int** i=1; // ② 코드 직접 초기화

{ // ③ 초기화 블럭을 통한 초기화

i=1;

}

**private** **int** si// ③ 초기화 블럭을 통한 초기화

static{

si=1;

}

**public** Test1() {i=1; } // ①생성자를 통한 초기화

**public** **int** getI() {

**return** i;

}

**public** **void** setI(**int** i) {

**this**.i = i;

}

}

**(앞선 소스를 다음과 같이 변환해 구현해보자-yisy0703@naver.com**

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

성 적 표

-------------------------------------------------------------

번호 이름 국어 영어 수학 총점 평균

-------------------------------------------------------------

1 정우성 90 80 95 265 88.33

2 김하늘 100 80 95 275 91.67

3 황정민 95 80 90 265 88.33

4 강동원 95 90 99 284 94.67

5 유아인 90 90 90 270 90.00

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

**package** com.oop.staticstudent;

**public** **class** Student {

**static** **int** *count* = 0;

**private** String name;

**private** **int** no;

**private** **int** kor;

**private** **int** eng;

**private** **int** math;

**private** **int** sum;

**private** **double** avg;

**public** Student(){}

**public** Student(String name, **int** kor, **int** eng, **int** math){

++*count*;

no = *count*;

**this**.name=name; **this**.kor=kor;

**this**.eng=eng; **this**.math=math;

}

**public** **void** calnShow(){

sum = kor + eng + math;

avg = sum / 3.0;

System.***out***.printf("%d\t%s\t%d\t%d\t%d\t%d\t%4.1f\n",no, name ,kor,eng,math,sum,avg);

}

}

**package** com.oop.staticstudent;

**public** **class** StudentTestMain {

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

*line*('■',50);

System.***out***.println(" 성 적 표");

*line*('-',50);

System.***out***.println("이름\t번호\t국어\t영어\t수학\t총점\t평균");

*line*('-',50);

Student jung = **new** Student("정우성", 90, 90, 90);

Student kim = **new** Student("김하늘", 80, 80, 80);

Student hwang = **new** Student("황정민", 70, 70, 70);

Student kang = **new** Student("강동원", 70, 70, 70);

Student yu = **new** Student("유아인", 90, 90, 90);

jung.calnShow();

kim.calnShow();

hwang.calnShow();

kang.calnShow();

yu.calnShow();

*line*('■',50);

}

**private** **static** **void** line(**char** c, **int** i) {

**for**(**int** cnt=0 ; cnt <i ; cnt++)

System.***out***.print(c);

System.***out***.println();

}

}