**14주차\_실습과제(학번: 20157135 이름: 이상우 )**

* 강의 내용
* 정적 필드, 정적 메소드

**import** java.util.\*;

**class** Car {

**private** String owner; // 소유주

**private** **int** year; // 연식

**private** String type; // 차종

**private** **int** id; //일련번호

// 정적 변수(클래스 변수) - 객체 생성 없이 클래스로 접근, 생성된 객체 들이 서로 공유

**private** **static** **int** *cnt* = 0;

**public** Car(String owner, **int** year, String type) {

**this**.owner = owner;

**this**.year = year;

**this**.type = type;

**id = ++*cnt*;**

}

// 정적메소드 : 지역변수와 정적변수만 사용가능, 정적 메소드만 호출할 수 있다

**public** **static** **int** getCnt() { // 정적변수 cnt 값 반환

**return** *cnt*;

}

**public** String toString() { // 객체 내용을 문자열로 반환

**return** "Car[owner=" + owner + ", year=" + year + ", type=" + type + ", id= " + id +"]";

}

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

Scanner in = **new** Scanner(System.***in***);

// 필드 초기화에 필요한 값을 생성자 호출 시 생성자 매개변수로 전달

// 생성자는 객체 생성 시 딱 한번 호출

Car carObj1 = **new** Car("윤인성", 2014, "소형"); // 객체 선언 & 생성 & 초기화

Car carObj2; // 객체 선언

System.***out***.print("소유주, 년식, 차종을 순서대로 입력하세요");

// 표준입력장치로 입력한 값을 생성자 매개변수로 전달하여 필드 초기화

carObj2 = **new** Car(in.next(), in.nextInt(), in.next());

System.***out***.println("객체 1: " + carObj1);

System.***out***.println("객체 2: " + carObj2);

// 정적 메소드는 클래스 이름으로 호출

System.***out***.println("생성 된 객체 수: " + Car.*getCnt*());

}

}

|  |
| --- |
| **[실행결과]** |

* **정적 메소드**

**import** java.util.\*;

**class** Box{

**private** **int** ga, se, he; //가로, 세로, 높이 저장

**private** **int** volume; //부피 저장

**public** Box(**int** ga, **int** se, **int** he){

**this**.ga=ga; **this**.se=se; **this**.he=he;

volume = ga\*se\*he; //부피 계산

}

**public** **int** getVolume(){

**return** volume;

}

**public** String toString(){

String str="가로 : " + ga +"\t세로 : " + se + "\t높이 : " + he;

str += "\t부피 : " + volume;

**return** str;

}

}

**class** BoxMax{

//매개변수로 받은 객체의 필드 volume을 비교하여 큰 volume값을 갖는 객체를 반환

**public** **static** Box isMax(Box b1, Box b2){

**if**(b1.getVolume() > b2.getVolume())

**return** b1;

**else**

**return** b2;

}

}

**public** **class** BoxTest {

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

Scanner in=**new** Scanner(System.***in***);

Box obj1=**new** Box(5, 3, 6); //객체 선언 & 생성

Box obj2=**new** Box(10, 2, 5); //객체 선언 & 생성

Box max = BoxMax.*isMax*(obj1, obj2); //정적메소드는 클래스 이름으로 호출

System.***out***.println("부피가 큰 박스 => " + max);

}

}

|  |
| --- |
| **[실행결과]** |

* **객체 배열**

**import** java.util.\*;

**class** Box {

**private** **int** ga, se, he; // 가로, 세로, 높이 저장

**private** **int** volume; // 부피 저장

**public** Box(**int** ga, **int** se, **int** he) {

**this**.ga = ga;

**this**.se = se;

**this**.he = he;

volume = ga \* se \* he; // 부피 계산

}

// 매개변수로 받은 객체의 필드 volume을 비교하여 실행중인 객체가 크면 1을 매개변수로 받은 객체가 크면 -1을 반환

**public** **int** isMax(Box b1) {

**if** (**this**.volume > b1.volume)

**return** 1;

**else**

**return** -1;

}

**public** **int** getVolume() {

**return** volume;

}

**public** String toString() {

String str = "가로 : " + ga + "\t세로 : " + se + "\t높이 : " + he;

str += "\t부피 : " + volume;

**return** str;

}

}

**public** **class** BoxTest {

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

Scanner in = **new** Scanner(System.***in***);

Box[] obj = **new** Box[5]; // 객체 배열 선언 & 생성

Box max;

**int** cnt = 0;

System.***out***.println("가로 세로 높이를 입력하세요(공백으로 구분)");

// 입력받은 값으로 객체 배열 원소 초기화

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

System.***out***.print((i + 1) + "번째 : ");

// 객체 배열 초기화

obj[i] = **new** Box(in.nextInt(), in.nextInt(), in.nextInt());

}

**for** (Box b : obj) { // 객체 내용 출력

System.***out***.print((++cnt) + "번째 Box : ");

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

}

// 부피가 가장 큰 객체 찾기

max = obj[0];

**for** (**int** i = 1; i < obj.length; i++) {

**if** (max.isMax(obj[i]) == -1)

max = obj[i];

}

System.***out***.println("가장 큰 부피 => " + max.getVolume());

}

}

|  |
| --- |
| **[실행결과]** |

* **내용 점검**

1. 다음 코드에 있는 오류를 수정하시오.

(1) public class Point {

private int x, y;

public void Point(int x, int y) {

x = x;

y = y;

}}

|  |
| --- |
| 수정소스  **public** **class** Point {  **private** **int** x, y;  **public** **void** Point(**int** x, **int** y) {  **this**.x = x;  **this**.y = y;  }  } |
| 틀린부분  필드가 오버로딩 되어있음으로 생성자에서 x=x가 아닌  This.x=x; This.y=y; 로 해줘야 한다. |

(2) public class MyClass {

private String getName() {

return "MyClass";

}

public static String getClassName() {

return getName();

}

}

|  |
| --- |
| 수정소스  **public** **class** MyClass {  **private** String getName() {  **return** "MyClass";  }  **public** String getClassName() {  **return** getName();  }  }  ------------------------------------------------  **public** **class** MyClass {  **private** **static** String getName() {  **return** "MyClass";  }  **public** **static** String getClassName() {  **return** *getName*();  }  } |
| 정적 메소드,에는 정적 메소드,정적 변수만이 올수 있다. |

1. 정육면체를 나타내는 클래스 Cube가 다음과 같이 정의되어 있다.

(1) 매개 변수가 없는 생성자를 추가하시오. 이 생성자는 side를 0으로 할당한다.

(2) 또 하나의 생성자를 추가하시오. 이 생성자는 매개 변수를 통하여 전달된 값을 side에 할당한다.

public class Cube {

private double side; // 정육면체의 한변

public double getSide() {

return side;

}

public double getVolume() {

return side\*side\*side;

}

}

|  |
| --- |
| **소스**  **class** Cube {  **private** **double** side; // 정육면체의 한변  Cube(){  side=0;  }  Cube(**int** side){  **this**.side=side;  }  **public** **double** getSide() {  **return** side;  }  **public** **double** getVolume() {  **return** side\*side\*side;  }  }  **public** **class** aa5{  **public** **static** **void** main(String[] args){  Cube obj = **new** Cube( 3);  System.***out***.println(obj.getSide());  System.***out***.println(obj.getVolume());  }  } |
| **[실행결과]** |

* **프로그램 과제**

1. MyMetric이라는 클래스를 작성하고 여기에 킬로미터를 마일로 변환하는 정적 메소드인 kiloToMile()을 작성하라. 또 반대로 마일을 킬로미터로 변환하는 정적 메소드 mileToKilo()로 작성하라.

**public** **class** MyMetric {

//계산 결과를 저장하는 정적 필드 변수 선언 distance, double형 – 본인작성

//매개변수(double)로 받은 값을 mile로 변환하여

//distance에 저장한 후 반환 하는 정적메소드 – 본인작성

//매개변수(double)로 받은 값을 킬로미터로 변환하여

// distance에 저장한 후 반환하는 정적 메소드 – 본인작성

}

1. MyMetricTest 클래스에서 1번에서 정의한 정적 메소드를 사용하여 제시된 조건대로 처리하시오.

* 조건 : 1번이 입력되면 킬로미터를 마일로 변환, 2번이 입력되면 마일을 킬로미터로 변환한다

|  |
| --- |
| **[1번과2번 - 프로그램 소스]**  **import** java.util.\*;  **class** MyMetric {  **private** **static** **double** *distance*;  //계산 결과를 저장하는 정적 필드 변수 선언 distance, double형 – 본인작성  **public** **double** mile(**double** distance){  **return** distance\*0.6;  }  //매개변수(double)로 받은 값을 mile로 변환하여  //distance에 저장한 후 반환 하는 정적메소드 – 본인작성  **public** **double** km(**double** distance){  **return** distance\*1.6;  }  //매개변수(double)로 받은 값을 킬로미터로 변환하여  // distance에 저장한 후 반환하는 정적 메소드 – 본인작성    }  **public** **class** aa6 {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  MyMetric obj = **new** MyMetric();  System.***out***.println("변환할 매뉴 입력");  System.***out***.println("1.km->mile \n 2.mile->km");  **int** x=key.nextInt();  **switch**(x){  **case** 1:  System.***out***.println("변환할 km 입력");  System.***out***.println(obj.mile(key.nextDouble()));  **break**;  **case** 2:  System.***out***.println("변환할 mile 입력");  System.***out***.println(obj.mile(key.nextDouble()));  **break**;  }  }  } |
| **[실행 결과]** |

1. 성적 합을 비교하여 합이 큰 객체를 출력하는 프로그램을 작성하시오.

class Student{

//이름(String name), 세 개의 성적(int d1, d2, d3), 합(int hap)을 저장하기 위한 필드 선언 - 전용멤버

//매개변수로 받은 데이터를 필드값으로 초기화(이름과 세 개의 성적)하는 생성자. 단, 생성자에서 합을 계산하도록 한다

//객체 내용(이름, 세 과목 성적, 합)을 문자열로 반환하는 toString()

}

class StdComp{

//매개변수로 두개의 Student 객체를 받아 성적합이 큰 객체를 반환하는 정적메소드

}

class StudentTest{

public static void main(String[] args){

// Student 타입의 객체 2개를 생성한다. 단, 객체 생성시 키보드로 입력된 값을 생성자 매개변수로 전달

//객체 내용 출력

//성적이 큰 객체를 출력 – StdComp 클래스에서 정의된 정적메소드 사용

}

}

|  |
| --- |
| **[프로그램 소스]**  **import** java.util.\*;  **class** Student{  **private** String name;  **private** **int** d1,d2,d3;  **int** hap;  **public** Student gethap;  //이름(String name), 세 개의 성적(int d1, d2, d3), 합(int hap)을 저장하기 위한 필드 선언 - 전용멤버  Student(**int** d1, **int** d2,**int** d3 ,String name){  **this**.d1=d1;  **this**.d2=d2;  **this**.d3=d3;  **this**.hap=d1+d2+d3;  **this**.name=name;  }  //매개변수로 받은 데이터를 필드값으로 초기화(이름과 세 개의 성적)하는 생성자. 단, 생성자에서 합을 계산하도록 한다  **public** String toString(){  System.***out***.println("이름:"+name);  System.***out***.println("d1:"+d1);  System.***out***.println("d2:"+d2);  System.***out***.println("d3:"+d3);  System.***out***.println("합:"+hap);  **return** "";  }  **int** gethap(){  **return** hap;  }  //객체 내용(이름, 세 과목 성적, 합)을 문자열로 반환하는 toString()  }  **class** StdComp{  **public** **static** Student setmax(Student hap ,Student hap2){  Student temp=**null**;  **if**(hap.gethap() >hap2.gethap()){  temp=hap;  }  **else** **if**(hap.gethap() <hap2.gethap()){  temp=hap2;  }  **return** temp;  }  }  **public** **class** aa7 {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  System.***out***.println("정수 입력");  **int** x=key.nextInt();  **int** x1=key.nextInt();  **int** x2=key.nextInt();  System.***out***.println("2.정수 입력");  **int** x3=key.nextInt();  **int** x4=key.nextInt();  **int** x5=key.nextInt();  Student obj = **new** Student(x,x1,x2,"홍길동");  Student obj2 = **new** Student(x3,x4,x5,"홍길이");  StdComp obj3 = **new** StdComp();  System.***out***.println("객체1"+obj);  System.***out***.println("객체2"+obj2);  System.***out***.println("큰객체는");  System.***out***.println(StdComp.*setmax*(obj ,obj2));  System.***out***.println("입니다.");  }  } |
| **[실행 결과]** |

1. 비행기를 나타내는 Plane라는 이름의 클래스를 설계하라. Plane 클래스는 제작사(예를 들어서 에어버스), 모델(A380), 최대 승객수(500)를 필드로 가지고 있다.

(1) 필드를 정의하라. 모든 필드는 전용 멤버로 하라.

(2) 모든 필드에 대한 접근자와 설정자 메소드를 작성한다.

(3) Plane 클래스의 생성자를 몇 개를 중복 정의하라. 생성자는 모든 데이터를 받을 수도 있고 아니면 하나도 받지 않을 수 있다.

(5) PlaneTest라는 이름의 테스트 클래스를 만드는데 main()에서 Plane 객체 여러 개를 생성하고 접근자와 설정자를 호출하여 보라.

(6) Plane 클래스에 지금까지 생성된 비행기의 개수를 나타내는 정적 변수인 planes를 추가하고 생성자에서 증가시켜 보자.

(7) Plane 클래스에 정적 변수 planes의 값을 반환하는 정적 메소드인 getPlanes()를 추가하고 main()에서 호출하여 보라.

|  |
| --- |
| **[프로그램 소스]**  **import** java.util.\*;  **class** plane{  **private** String name;  **private** **int** pass;  **private** **static** **int** *planes*=0;  plane(){  **this**.name="A380";  **this**.pass=500;  }  plane(String name, **int** pass){  *planes*++;  **this**.name=name;  **this**.pass=pass;  }  **void** setplane(String name, **int** pass){  **this**.name=name;  **this**.pass=pass;  }  String getname(){  **return** name;  }  **int** getpass(){  **return** pass;  }  **static** **int** getPlanes(){  **return** *planes*;  }    }  **public** **class** aa8 {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  plane obj =**new** plane("A340",250);  plane obj1 =**new** plane("A340",250);  plane obj2 =**new** plane("A340",250);  obj.setplane("A320",100);  obj1.setplane("A330",200);  obj2.setplane("A350",300);  System.***out***.println(obj.getname());  System.***out***.println(obj.getpass());  System.***out***.println(obj1.getname());  System.***out***.println(obj1.getpass());  System.***out***.println(obj2.getname());  System.***out***.println(obj2.getpass());  System.***out***.println(plane.*getPlanes*());  }  } |
| **[실행 결과]** |

1. 패스트푸드점 메뉴관리를 위한 클래스를 작성하고 테스트하는 프로그램을 작성하시오.

class FastFood {

//필드 선언 : 메뉴명(String), 단가(int), 구입가격(int)- 전용 멤버

// 생성자 – 매개변수로 받은 문자를 MenuProc 클래스에 있는 메소드 매개변수로 전달하고 반환 받은 값을 필드로

초기화

//costCal() 메소드 : 구입가격을 계산하여 필드에 저장(구입가격 = 단가 \* 개수). 단, 개수는 매개변수로 받는다. 반환값 없음

//구입가격 필드에 대한 접근자 메소드

//객체 내용(메뉴명, 단가)을 문자열로 반환하는 toString() 메소드

}

class MenuProc{

// menuName() 메소드 정의 : 문자를 매개변수로 받아 메뉴명을 반환하는 정적 메소드

//vaLue() 메소드 정의 : 문자를 매개변수로 받아 가격을 반환하는 정적 메소드

|  |  |  |
| --- | --- | --- |
| 문자 | 메뉴명 | 단가 |
| H 또는 h | 햄버거 | 2500 |
| I 또는 i | 아이스크림 | 1500 |
| P 또는 p | 감자튀김 | 3000 |
| C 또는 c | 치킨조각 | 1000 |

}

// FastFood 클래스에 대한 객체를 생성하고 테스트 하는 클래스

public class FastFoodTest {

public static void main(String[] args) {

// Fastfood 클래스 객체 선언

//메뉴 입력 – 문자 n을 입력하면 반복문 종료

//Fastfood 객체 생성, 메뉴로 입력 받은 문자를 생성자 매개변수로 전달

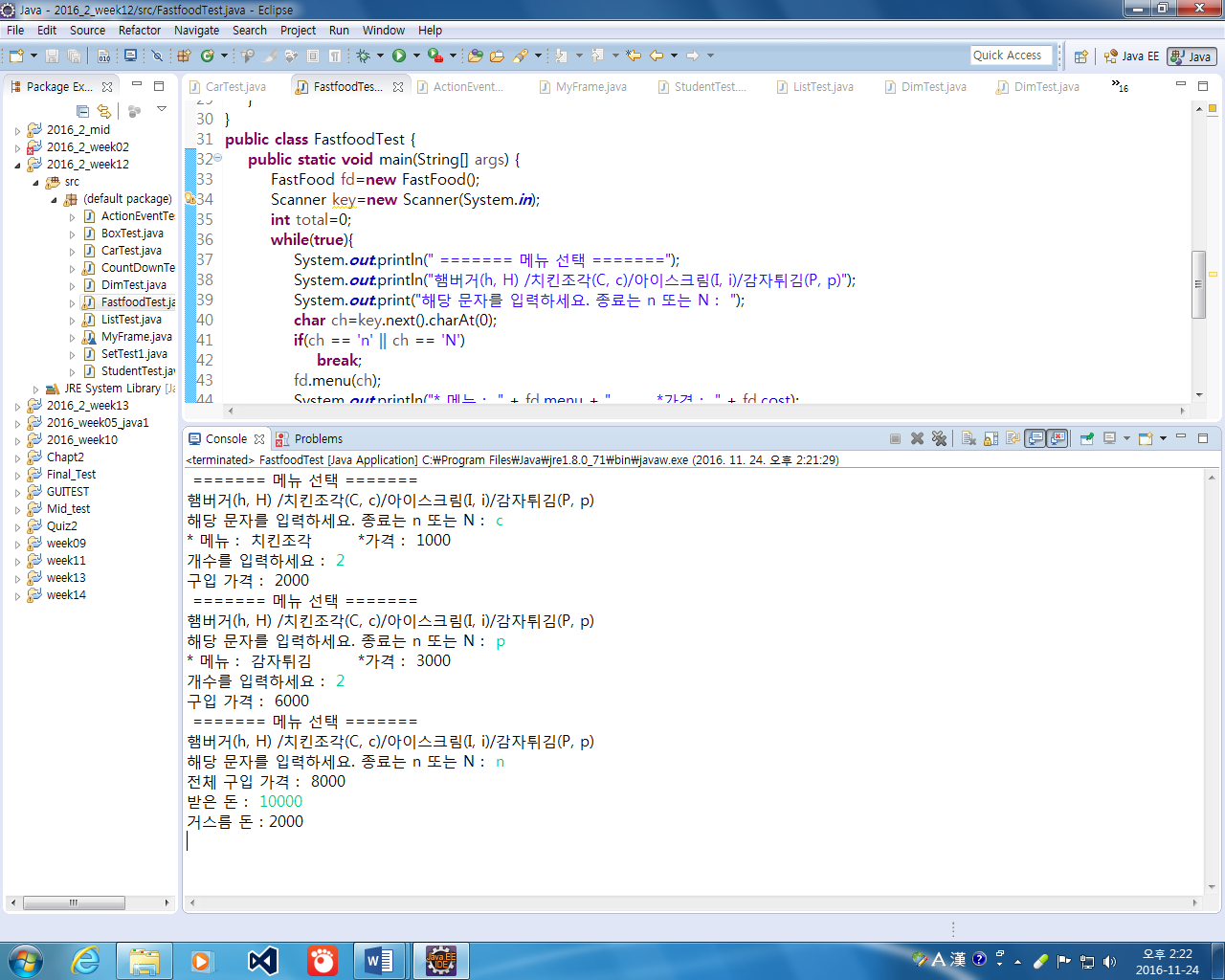
//사용자로부터 구입개수 입력 받아 costCal() 메소드 매개변수로 전달

//구입 가격 출력

//반복문 종료 후 거스름돈 계산하여 출력

}

}



|  |
| --- |
| **[프로그램 소스]**  **import** java.util.\*;  **class** FastFood {  **static** String *food*;  **static** **int** *pay*;  **static** **int** *realpay*;  **static** **char** *x*;  FastFood(**char** x){  **this**.*x*=MenuProc.*y*;  }  **void** costCal(**int** num){  *realpay*=*pay*\*num;  }  **int** getrealpay(){  **return** *realpay*;  }  **public** String toString(){  System.***out***.println("==========매뉴 선택============");  System.***out***.println("햄버거(H,h)/치킨조각(C,c),아이스크림(I,i)/감자튀김(P,p)");  System.***out***.println("해당 문제를 입력하세요. 종료는 n 또는 N:");  **return** "";  }  }  **class** MenuProc{  **static** String *food*;  **static** **int** *pay*;  **static** **int** *realpay*;  **static** **char** *y*;    **static** String menuName(){  **switch**(*y*){  **case** 'H':  **case** 'h':  *food*="햄버거";  **break**;  **case** 'I':  **case** 'i':  *food*="아이스크림";  **break**;  **case** 'P':  **case** 'p':  *food*="감자튀김";  **break**;  **case** 'C':  **case** 'c':  *food*="치킨조각";  **break**;    }  **return** *food*;  }  **static** **int** vaLue(){  **switch**(*y*){  **case** 'H':  **case** 'h':  *pay*=2500;  **break**;  **case** 'I':  **case** 'i':  *pay*=1500;  **break**;  **case** 'P':  **case** 'p':  *pay*=3000;  **break**;  **case** 'C':  **case** 'c':  *pay*=1000;  **break**;    }  **return** *pay*;  }  }  **public** **class** aa9 {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  **char** x =0;  FastFood obj=**new** FastFood (x);  **while**(**true**){  System.***out***.print(obj);  x=key.next().charAt(0);  **if**(x=='n' || x=='N')  **break**;  System.***out***.println("\* 매뉴"+MenuProc.*food*+" "+"\* 가격"+MenuProc.*pay*);  System.***out***.print("개수를 입력하세요");  obj.costCal(key.nextInt());  }  System.***out***.println("받은 돈:");  **int** a=key.nextInt();  System.***out***.println("거스름 돈:"+(a- obj.getrealpay())+"입니다." );  }  } |
| **[실행 결과]** |

1. 제품 관리를 위한 Product 클래스를 작성하시오

* 필드 구성 : 제품번호(String, private), 제품명(String, private), 판매한 제품 갯수(int, private), 제품가격(int, private)
* 매개변수로 받은 값으로 필드를 초기화 하는 생성자와 모든 필드를 0으로 하는 디폴트 생성자
* 각 필드에 대한 접근자와 설정자 메소드
* getInvoice() 메소드 : 제품 판매액(제품가격 \* 판매한 제품 갯수)을 계산하여 int로 반환
* display() 메소드: 필드값과 제품 판매액을 출력, 반환값 없음
* int maxProduct(Product obj1) : 매개변수로 받은 객체와 비교하여 실행중인 객체가 크면 1을 매개변수로 받은 객체가 크면 -1을 반환

1. ProductTest 클래스를 작성하시오

* Prodect 객체 생성 시 필드 초기값을 생성자 매개변수로 전달. 단, 표준입력 장치로 입력받은 값을 생성자 매개변수로 전달. 단, 객체 배열을 사용하며 크기는 5로 할 것
* display()메소드를 호출하여 필드값과 제품 판매액 출력
* 제품 판매액이 가장 큰 객체 출력

|  |
| --- |
| **[6번과7번 - 프로그램 소스]**  **import** java.util.Scanner;  **class** Product{  **private** String namenum;  **private** String name;  **private** **int** num;  **private** **int** pay;  Product(){  namenum="0";  name="0";  num=0;  pay=0;  }  Product(String namenum,String name,**int** num,**int** pay){  **this**.namenum=namenum;  **this**.name=name;  **this**.num=num;  **this**.pay=pay;  }  **int** getInvoice() {  **return** pay\*num;  }  **void** display(){  System.***out***.println("제품번호"+namenum);  System.***out***.println("제품명"+name);  System.***out***.println("판매개수"+num);  System.***out***.println("제품가격"+pay);  System.***out***.println("판매금액"+getInvoice());  }  **int** maxProduct(Product obj1) {  **if**(getInvoice() > obj1.getInvoice()) {  **return** 1;  }  **else**{  **return** -1;  }    }  }  **public** **class** aa10 {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  Product[] obj = **new** Product[5];  **for**(**int** i=0;i<obj.length;i++){  System.***out***.println("제품 번호,이름,개수,가격 입력");  obj[i]=**new** Product(key.next(),key.next(),key.nextInt(),key.nextInt());  }  **for**(**int** i=0;i<obj.length;i++){  obj[i].display();  }  Product max=obj[0];;  **for**(**int** i=0;i<obj.length;i++){  **if**(max.maxProduct(obj[i]) == -1){  max= obj[i];  }  }System.***out***.println("=== 최대 ===");  max.display();  }  } |
| **[실행 결과]** |

1. 키보드로 영 단어를 입력하면 단어 뜻을 한글로 출력하는 프로그램을 작성하시오. 영단어와 한글은 객체배열 생성시 초기화 하도록 한다.

class Word{

//영어 단어 저장을 위한 필드, private

//영 단어에 대한 뜻을 저장하는 필드, private

//생성자 – 매개변수로 받은 값을 필드로 초기화

//모든 필드에 대한 접근자 메소드

}

class WordTest{

public static void main(String[] args){

//객체 배열을 선언 생성하면서 초기화 – 본인이 완성할 것

Word[] object={new Word(“apple”, “사과” ), ………}

//입력된 영 단어에 해당하는 한글 단어 출력 – equals() 사용

//입력된 문자로 시작하는 영단어와 뜻을 한글로 출력 – charAt() 사용

**}**

|  |
| --- |
| **[프로그램 소스]**  **import** java.util.Scanner;  **class** Word{  **private** String eng;  **private** String kor;  Word(String eng ,String kor){  **this**.eng=eng;  **this**.kor=kor;  }  **static** **void** engtokor(String c , Word[] obj){  **for**(**int** i=0;i<obj.length;i++){  **if**(obj[i].eng.equals(c)){  **int** x=i;  System.***out***.println("단어의 뜻은:"+obj[x].kor);  **break**;  }  }  }  **static** **void** kortoeng(**char** eng , Word[] obj){  **for**(**int** i=0;i<obj.length;i++){  **if**(eng == obj[i].eng.charAt(0)){  **int** x=i;  System.***out***.println("문자의 해당하는 단어는:"+obj[x].eng);  System.***out***.println("단어의 뜻은:"+obj[x].kor);  **break**;  }  }  }  }  **public** **class** aa11 {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  Word[] obj = **new** Word[5];  obj[0]=**new** Word("apple" ,"사과");  obj[1]=**new** Word("bear" ,"곰");  obj[2]=**new** Word("car" ,"자동차");  obj[3]=**new** Word("deliver" ,"배달");  obj[4]=**new** Word("effect" ,"효과");  System.***out***.println("단어를 입력 해 주세요.");  Word.*engtokor*(key.next() ,obj);  System.***out***.println("문자를 입력 해 주세요.");  Word.*kortoeng*(key.next().charAt(0) ,obj);    }  } |
| **[실행결과]** |