**자바 LAB**

**20184115 주정윤**

**컴퓨터공학과**

1. **Circle클래스 상속받은 Pizza클래스 작성하고 객체를 생성 후 피자의 면적을 비교하라**

**class** Circle { //부모클래스인 Circle 클래스

**private** **double** radius;

**private** String color;

**public** Circle() //Circle클래스의 생성자를 만들어준다

{

radius = 1.0;

color = "red";

}

**public** Circle(**double** radius) //Circle 클래스의 생성자를 만들어서 radius를 초기화해준다

**this**.radius = radius;

color = "red";

}

**public** Circle(**double** radius, String color) //Circle 클래스의 생성자하여 값을 초기화시켜준다

{

**this**.radius = radius;

**this**.color = color;

}

**public** **double** getRadius() //radius 접근자 생성

{

**return** radius;

}

**public** String getColor() //color 접근자생성

{

**return** color;

}

**public** **void** setRadius(**double** radius) //radius 설정자 생성

{

**this**.radius = radius;

}

**public** **void** setColor(String color) //color 설정자 생성

{

**this**.color = color;

}

**public** String toString() //toString 메소드를 만들어서 클래스의 정보를 반환해준다

{

**return** "Cirlce [radius="+radius+", color="+color+"]";

}

**public** **double** getArea() //면적을 구하는 메소드

{

**return** radius \* radius \* 3.14;

}

}

**class** Pizza **extends** Circle { //Circle 클래스를 상속받는 Pizza 클래스

**private** String type;

**private** **int** price;

**public** Pizza() //Pizza 클래스의 생성자를 만들어준다

{

type = “콤비네이션 피자";

price = 10;

}

**public** Pizza(String type, **int** price) //Pizza 클래스의 생성자(type, price 초기화)

{

**this**.type = type;

**this**.price = price;

}

**public** String getType() //type 접근자

{

**return** type;

}

**public** **int** getPrice() //price 접근자

{

**return** price;

}

}

**public** **class** Pizzatest {

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

Pizza obj1 = **new** Pizza("불고기피자", 15); //Pizza 클래스의 참조변수 obj1 선언, 객체 생성

obj1.setRadius(10); //obj1의 Radius를 10으로 초기화

Pizza obj2 = **new** Pizza("포테이토피자",20); //Pizza 클래스의 참조변수 obj2 선언, 객체 생성

obj2.setRadius(20); //obj2의 Radius를 20으로 초기화

System.***out***.println(obj1.getType() +", 가격: "+ obj1.getPrice()+ " 피자의 크기는: "+obj1.getRadius());

System.***out***.println(obj2.getType() +", 가격: "+ obj2.getPrice()+ " 피자의 크기는: "+obj2.getRadius());

**if**(obj1.getRadius() > obj2.getRadius()) //obj1과 obj2의 Radius를 비교 후 ob1이 더 큰 경우

System.***out***.println(obj1.getType()+" 가 더 큽니다.");

**else** //obj2가 더 큰 경우

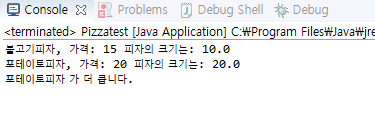
System.***out***.println(obj2.getType()+" 가 더 큽니다.");

}

}

**설명: 부모클래스인 Circle 클래스를 우선 만들어주고 Circle 클래스의 생성자를 만들어준다. 값을 초기화해주고 그리고 getRadius getColor와 같은 접근자와 set으로 하는 두 가지의 설정자를 생성해주어 값을 반환하고 정의하게끔 한다. toString 메소드를 만들어서 클래스의 정보를 반환해주도록 한다. 면적을 구하는 메소드를 만들어서 반지름\*반지름\*3.14의 값을 반환해준다. Circle 클래스를 상속받는 피자 클래스를 만들어주고 역시 생성자를 만들어주어서 값을 초기화시켜준다. 그리고 Type의 접근자와 price의 접근자 또한 만들어준다. 메인함수에는 피자 클래스에 대한 참조변수 객체를 2개 만들어주고 필드의 값도 넣어준다. 설정자를 불러서 값을 초기화시켜주고 가격과 함께 피자의 크기를 아까 면적을 구하는 함수를 이용해서 출력해주면 된다.**

**실행결과:**



**2. Person 클래스를 상속받아 Student, Employee 클래스 정의 후 steve jobs인 사람 찾기**

**class** Person{ //Person 부모클래스 정의

**private** String name;

**private** **int** id;

**void** printInfo() { //name과 id의 정보를 출력할 메소드

System.***out***.println("name = "+name+"id = "+id);

}

**public** String Getname(){ //name 접근자 정의

**return** name;

}

**public** **int** Getid() { //id 접근자 정의

**return** id;

}

**public** **void** Setname(String name) { //name 설정자 정의

**this**.name = name;

}

**public** **void** Setid(**int** id) { //id 설정자 정의

**this**.id = id;

}

}

**class** Student **extends** Person{ //Person을 상속받는 Student 클래스

**private** **double** gpa;

**private** **int** credits;

**public** Student(String name, **int** id, **double** gpa, **int** credits) { //Student 클래스의 생성자

Setname(name); //설정자를 통하여 name에 값을 넣어줌

Setid(id); //설정자를 통하여 id에 값을 넣어줌

**this**.gpa = gpa;

**this**.credits = credits;

}

**void** printInfo() { // name, id, gpa, credits에 있는 정보를 출력하는 메소드, 부모클래스의 printInfo() 메소드 오버라이딩

System.***out***.println("name = \""+Getname()+"\" id = "+Getid()+" gpa = "+gpa+" credits = "+credits);

}

}

**class** Employee **extends** Person{ // Person 클래스를 상속받는 Employee 클래스

**private** **int** salary;

**private** **int** years;

**public** Employee(String name, **int** id, **int** salary, **int** years) { //Employee 클래스의 생성자

Setname(name);

Setid(id);

**this**.salary = salary;

**this**.years = years;

}

**void** printInfo() {// 메소드 오버라이딩, name, id, salary, years 정보 출력

System.***out***.println("name =\""+Getname()+"\" id = "+Getid()+" salary = "+salary + " years = "+years);

}

}

**public** **class** PersonTest {

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

**int** i = 0, k = 0;

Person[] obj = **new** Person[10]; //부모클래스 Person의 객체배열을 생성

**for**(**int** j = 0; j < obj.length; j++) //객체배열의 요소를 null로 초기화

{

obj[j] = **null**;

}

obj[0] = **new** Student("steve jobs", 1, 3.9, 30); //obj[0]번째 요소에 Student 객체를 초기화

obj[1] = **new** Employee("Jeff Bezos", 2, 10000, 10); //obj[1]번째 요소에 Employee 객체를 초기화

obj[2] = **new** Employee("Bill Gates", 3, 20000, 15); //obj[2]번째 요소에 Employee 객체를 초기화

obj[3] = **new** Employee("steve jobs", 4, 30000, 20); //obj[3]번째 요소에 Employee 객체를 초기화

System.***out***.println("저장된 사람들");

**while**(obj[i] != **null**) { //obj[i]에 저장된 값이 null이 아니라면

obj[i].printInfo(); //obj 객체 배열의 printInfo() 메소드 호출

i++;

}

System.***out***.println("이름이 steve jobs인 사람");

**while**(obj[k] != **null**) {

**if**(obj[k].Getname().equals("steve jobs")) // obj[k]의 name이 steve jobs 일 경우

obj[k].printInfo(); //obj[k]의 printInfo() 메소드 호출

k++;

}

}

}

설명 : Person 부모 클래스를 정의해주고 private로 부모클래스를 정의해준다. 두 가지의 필드를 정의해주고 name과 id의 정보를 출력해줄 메소드를 생성한다. 메소드 안에서는 출력함수를 이용해서 출력하도록 한다. Name과 id의 접근자와 설정자를 정의해준다. Person을 상속받는 student 클래스를 만들어준다. Student클래스의 생성자를 만들어서 설정자를 통해서 name과 id에 값을 넣어주고 name, id. 헴, credits에 있는 정보를 출력하는 메소드를 만들어서 안에서 출력함수를 이용해서 출력해주도록 한다. Person 부모 클래스를 상속받는 employee 클래스를 만들어주고 필드를 정의해준다. Employee 클래스의 생성자를 만들어주고 printinfo 메소드를 만들어서 메소드 오버라이딩으로 정보를 더 출력하게끔 바꿔준다. 그리고 메인함수에서 부모클래스 person의 객체배열의 생성해주고 모든 요소를 null로 초기화 시켜준다. Obj 베열에 객체를 모두 초기화시켜주고 객체 안의 값이 null이 아니라면 printinfo 메소드를 호출해준다. 역시 아래도 똑같이 출력해준다.

**실행결과:**

