

자바 기초

May. 14, 2007

객체와 클래스 III

류 대 우

davidryu@newtc.co.kr

학습 목표

■ 학습 목표

- Final을 사용하는 세 가지 경우
- Static에 대하여
- Abstract를 사용하는 이유와 사용 방법
- 인터페이스 사용 이유와 사용 방법
- 내부 클래스 사용 방법

final로 클래스 선언

- ShapeArea.java
 - final class ShapeArea{
 - public static double calRectArea(double w, double h){
 - double area = w*h;
 - return area;
 - }
 - public static double calCircleArea(double r){
 - double area = Math.PI*r*r;
 - return area;
 - }
 - }

Final로 선언된 클래스 사용불가

- FinalClassEx.java

- Public class FinalClassEx extends ShapeArea{
 - public static void main(String[] args){
 - double rect_area = calRectArea(4,5);
 - Double circle_area = calCircleArea(7);
 - System.out.println("가로가 4, 세로가 5인 사각형의 면적은"+rect_area+"입니다.");
 - System.out.println("반지름의 길이가 7인 원의 면적은"+circle_area+"입니다.");
 - }
- }

final로 메서드 선언

- ShapeArea2.java

- public class ShapeArea2{
 - public static final double calCircleArea(double r){
 - double area = Math.PI*r*r;
 - return area;
 - }
- }

Final로 선언된 메서드 오버라이딩 불가

- FinalClassEx.java

- public class FinalClassEx extends ShapeArea2{
 - public static double calCircleArea(double r){
 - double area = 3.14*r*r;
 - return area;
 - }
 - public static void main(String[] args){
 - double circle_area = calCircleArea(7);
 - System.out.println("반지름의 길이가 7인 원의 면적은"+circle_area+"입니다.");
 - }
- }

final로 변수 선언

- FinalVariable.java

- public class FinalVariable {
 - public static final static double PI = 3.14;
 - public static double calCircleArea(double r){
 - double area=PI*r*r;
 - PI=3.141592; ← PI를 고치려 하는 데, 에러가 발생하게 된다.
 - return area;
 - }
 - public static void main(String[] args){
 - double circle_area = calCircleArea(7);
 - System.out.println("반지름의 길이가 7인 원의 면적은" + circle_area + "입니다.");
 - }
- }

Static을 사용하지 않은 예

- NonStaticEx.java

- NonStaticEx{
 - String title = “자바”;
- }

- NonStaticUse.java

- Public class NonStaticUse{
 - public static void main(String[] args){
 - NonStaticEx x = new NonStaticEx();
 - x.title=“메모리 공간 2”;
 - NonStaticEx y = new NonStaticEx();
 - System.out.println(“x객체의 title의 값은 ”+x.title+”입니다.”);
 - System.out.println(“y객체의 title의 값은 ”+y.title+”입니다.”);
 - }
- }

Static을 사용한 예

- NonStaticEx.java

- NonStaticEx{
 - static String title = “자바”;
- }

- NonStaticUse.java

- Public class NonStaticUse{
 - public static void main(String[] args){
 - NonStaticEx x = new NonStaticEx();
 - x.title=“메모리 공간 2”;
 - NonStaticEx y = new NonStaticEx();
 - System.out.println(“x객체의 title의 값은 ”+x.title+”입니다.”);
 - System.out.println(“y객체의 title의 값은 ”+y.title+”입니다.”);
 - }
- }

Abstract 예제

- AbstractName.java
 - public abstract class AbstractName{
 - public abstract void printName();
 - public static void printDosCommand(){
 - ChulSu cs = new ChulSu();
 - cs.printName();
 - YouSun ys = new YouSun();
 - ys.printName();
 - }
 - public static void main(String args[]){
 - printDosCommand();
 - }
 - }

interface 사용법

- public class 클래스_이름 implements 인터페이스_이름 1, 인터페이스_이름2,.....{
 -
- }

interface 예제 (I)

- SpeedControl.java
 - public interface SpeedControl{
 - public void speedUp(int velocity);
 - public void speedDown(int velocity);
 - }

interface 예제(1)

- Airplane.java
 - public class Airplane implements SpeedControl{
 - int v;
 - public Airplane(){
 - v=0;
 - }
 - public void speedUp(int velocity){
 - System.out.println("비행기 속도를"+velocity+"올립니다.");
 - v = v + velocity;
 - }
 - public void speedDown(int velocity){
 - System.out.println("비행기 속도를"+velocity+"줄입니다.");
 - v = v - velocity;
 - }

interface 예제(I)

- Car.java
 - public class Car implements SpeedControl{
 - int v;
 - public Car(){
 - v=0;
 - }
 - public void speedUp(int velocity){
 - System.out.println("자동차 속도를"+velocity+"올립니다.");
 - v = v + velocity;
 - }
 - public void speedDown(int velocity){
 - System.out.println("자동차 속도를"+velocity+"줄입니다.");
 - v = v - velocity;
 - }

interface 응용 실행 구문

- InterfaceExcute.java
 - public class InterfaceExecute{
 - public static void main(String args[]){
 - Car c = new Car();
 - c.speedUp(5);
 - c.speedDown(3);
 - Airplane a = new Airplane();
 - a.speedUp(4);
 - a.speedDown(2);
 - }
 - }

interface 예제(II)

- SpeedControl2.java
 - public interface SpeedControl2{
 - int MAX_SPEED = 10; ➔ 상수형 변수가 된다.
 - int MIN_SPEED = 2; ➔ 상수형 변수가 된다.
 - public void speedUp(int velocity);
 - public void speedDown(int velocity);
 - }

interface 예제(II)

```
■ Airplane2.java
  ■ public class Airplane2 implements SpeedControl2{
    ■ int v;
    ■ public Airplane2(){
      ■ v=0;
    ■ }
    ■ public void speedUp(int velocity){
      ■ if(MAX_SPEED < velocity){
        ■ System.out.println("최고 SPEED를 초과하였습니다.");
        ■ return;
      ■ }
      ■ System.out.println("비행기 속도를"+velocity+"올립니다.");
      ■ v = v + velocity;
    ■ }
    ■ public void speedDown(int velocity){
      ■ if(MIN_SPEED >= velocity){
        ■ System.out.println("최저 SPEED를 초과하였습니다.");
        ■ return;
      ■ }
      ■ System.out.println("비행기 속도를"+velocity+"내립니다.");
      ■ v = v - velocity;
    ■ }
  ■ }
```

interface 예제(II)

```
■ Car2.java
  ■ public class Car2 implements SpeedControl2{
    ■ int v;
    ■ public Car2(){
      ■ v=0;
    ■ }
    ■ public void speedUp(int velocity){
      ■ if(MAX_SPEED < velocity){
        ■ System.out.println("최고 SPEED를 초과하였습니다.");
        ■ return;
      ■ }
      ■ System.out.println("자동차 속도를"+velocity+"올립니다.");
      ■ v = v + velocity;
    ■ }
    ■ public void speedDown(int velocity){
      ■ if(MIN_SPEED >= velocity){
        ■ System.out.println("최저 SPEED를 초과하였습니다.");
        ■ return;
      ■ }
      ■ System.out.println("자동차 속도를"+velocity+"내립니다.");
      ■ v = v - velocity;
    ■ }
  ■ }
```

interface 응용 실행 구문(II)

- InterfaceExcute2.java
 - public class InterfaceExecute2{
 - public static void main(String args[]){
 - Car2 c = new Car2();
 - c.speedUp(5);
 - c.speedDown(3);
 - Airplane2 a = new Airplane2();
 - a.speedUp(4);
 - a.speedDown(2);
 - }
 - }