





9주차: Java 프로그래밍

백석대학교 강윤희



과제

 [일반 과제] 7-8주차 과제 [진행중]

 수정  삭제  과제평가

참고 자료 :

0%

제출기간	성적반영	성적공개	성적공개일자	사용 여부	연장제출	제출자수 (제출 / 미제출 / 연장제출 / 수강인원)
2020-04-27 00:00 ~ 2020-05-08 23:59	반영	공개	2020-05-09 00:00	 사용 	미허용	0명 / 28명 / 0명 / 28명

1) 실습 목적
객체 배열을 사용하여 다수의 원을 다루는 과정의 제약점은 배열 크기를 해결하기 위해 ArrayList 를 이해하고 이를 프로그램으로 구성한다.

2) 실습내용

- 전체 프로그램은 최소 2개의 클래스로 구성하여야 하며 이중 1개는 Circle 클래스이며, 다른 1개는 Circle 객체를 처리하는 클래스로 구성한다. 생성되는 클래스는 1개의 public 클래스 만을 허용한다. (pp.128-129 클래스와 객체)
- 배열(고정크기)를 해결하기 위해 ArrayList(pp.179-181) 를 사용하여 원 객체를 5개 이상 생성한 후 원의 넓이를 구하시오.
반지름을 입력 받아 원을 생성하며, 원의 반지름이 0보다 작으면 더이상 원 객체를 생성하지 않도록 한다
- Circle 클래스는 setter와 getter 메소드를 사용하여 원의 반지름(radius)를 접근하도록 하여야 한다.(pp. 142)
필드는 외부에 접근을 허락하지 않도록 하여 캡슐화 한다. (pp. 129-130)
- Circle 클래스는 원의 둘레를 구하기 위한 메소드 getCircumference() 메소드를 추가하여야 한다.
- Circle 클래스는 원을 생성하기 위한 생성자 메소드를 color 를 지정할 수 있도록 추가하여야 한다. 이를 위해 color를 필드로 추가한다. (pp. 144-146)
- 원의 넓이 출력을 위해서는 for-each 문을 사용한다. (pp.182)
- 생성된 Circle 객체의 수는 static 변수를 사용하여 유지한다. (pp.152)

```
Input Radius: 5
Input Radius: 3
Input Radius: 4
Input Radius: 2
Input Radius: 7
Input Radius: 8
Input Radius: -1
원의 넓이 (반지름 : 5.0) = 78.50
원의 넓이 (반지름 : 3.0) = 28.26
원의 넓이 (반지름 : 4.0) = 50.24
원의 넓이 (반지름 : 2.0) = 12.56
원의 넓이 (반지름 : 7.0) = 153.86
원의 넓이 (반지름 : 8.0) = 200.96
원의 둘레 (반지름 : 5.0) = 31.40
원의 둘레 (반지름 : 3.0) = 18.84
원의 둘레 (반지름 : 4.0) = 25.12
원의 둘레 (반지름 : 2.0) = 12.56
원의 둘레 (반지름 : 7.0) = 43.96
원의 둘레 (반지름 : 8.0) = 50.24
```

수업내용

7주차 2020-04-25 ~ 2020-05-02	NOW 8주차 2020-05-02 ~ 2020-05-09
<p>1-실습- private setter getter 25명 1명 2명 강의 보기</p> <p>2-실습-setter getter overloading-2 25명 1명 2명 강의 보기</p> <p>3-실습- static variable(class variable)-3 26명 1명 1명 강의 보기</p> <p>4. 문자열-1 25명 2명 1명 강의 보기</p> <p>5. 문자열-2 26명 1명 1명 강의 보기</p> <p>6. 배열-1 25명 2명 1명 강의 보기</p> <p>7. 배열-1-실습 26명 1명 1명 강의 보기</p>	<p>8주 과제</p> <p>1. 배열 2 1명 0명 27명 강의 보기</p> <p>2. 배열 3 1명 0명 27명 강의 보기</p> <p>3. 배열-4 1명 0명 27명 강의 보기</p> <p>4. ArrayList 실습 1명 0명 27명 강의 보기</p> <p>5. 디버깅하기 1명 0명 27명 강의 보기</p> <p>월요일 5교시 0명 0명 28명 강의 보기</p>

NOW 9주차 2020-05-09 ~ 2020-05-16
<p>1-상속개념이해하기-1 1명 0명 27명 강의 보기</p> <p>2-상속개념이해하기-2 1명 0명 27명 강의 보기</p> <p>3-상속-메소드 오버라이딩 1명 0명 27명 강의 보기</p> <p>4-상속 주사위 프로그램 1명 0명 27명 강의 보기</p> <p>5. Lab-Die-상속-1 1명 0명 27명 강의 보기</p> <p>6-Lab-Die-상속-2 1명 0명 27명 강의 보기</p>

```

public class test {

    public static void main(String[] args) {
        ArrayList<Circle> circles = new ArrayList<Circle>();

        Scanner in = new Scanner(System.in);

        double radius;

        System.out.print("Input Radius : ");
        while((radius = in.nextInt()) >= 0) {
            System.out.print("Input Color : ");
            Circle c = new Circle(radius);
            circles.add(c);
            System.out.print("Input Radius : ");
        }

        System.out.println("Total number of Circle : " + circles.size());

        for (int i = 0; i < circles.size(); i++) {
            Circle c = circles.get(i);
            System.out.printf("원의 넓이(반지름 : %.1f, 색깔 : %s) = %.2f\n", c.getRadius(), c.getcolor(), c.findArea());
        }

        for(Circle p : circles) {
            System.out.printf("원의 넓이(반지름 : %.1f, 색깔 : %s) = %.2f\n", p.getRadius(), p.getcolor(), p.findArea());
        }
    }
}

```

색깔 지정

```

43
44
45 class Circle{
46
47     private double PI = 3.14;
48     private double radius;
49     private double Circumference;
50
51     Scanner sc = new Scanner(System.in);
52     private String color = sc.nextLine();
53
54     public double getRadius() {
55         return radius;
56     }
57
58     public void setRadius(double r) {
59         this.radius = r;
60     }
61
62     public Circle(double r) {
63         radius = r;
64     }
65     public Circle(String color) {
66         this.color = color;
67     }
68     public void setcolor(String g) {
69         this.color = g;
70     }
71
72     public String getcolor() {
73         return color;
74     }
75     double findArea() {
76         return PI * radius * radius;
77     }
78
79     public double getCircumference() {
80         return Circumference = PI * radius * 2;
81     }
82
83     void show(double x, double y) {
84         System.out.printf("반지름 = %.1f, 넓이 = %.1f, 색깔 = %s", x, y);
85     }
86 }

```

```

34 public void setRadius(int radius) {
35     this.radius = radius;
36 }
37
38
39
40 }
41
42 public class ArrayListCircleDemo {
43     public static void main(String[] args) {
44         ArrayList<Circle> circles = new ArrayList<Circle>();
45         Scanner in = new Scanner(System.in);
46         String color;
47         int radius;
48         System.out.print("Input Radius:");
49         while ((radius=in.nextInt()) >= 0){
50             System.out.print("Input color:");
51             color=in.next();
52             Circle c = new Circle(radius,color);
53             circles.add(c);
54             System.out.print("Input Radus:");
55         }
56         System.out.printf("Total number of Circle:%d\n",(int)circles.size());
57         for(Circle c : circles)
58             System.out.printf("원의 넓이(반지름:%.1f,색깔:%s)=%.2f\n",c.getRadius(),c.getcolor(),c.findArea());
59         for(Circle c : circles)
60             System.out.printf("원의 둘레(반지름:%.1f)=%.2f\n",c.getRadius(),c.getCircumference());
61     }
62 }
63
64

```

들여쓰기

Console Problems Debug Shell

<terminated> ArrayListCircleDemo [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020. 5. 6. 오전 1:04:09)

```

Input Radius:10
Input color:blue
Input Radus:20
Input color:    red
Input Radus:    -3
Total number of Circle:2
원의 넓이(반지름:10.0,색깔:blue)=314.00
원의 넓이(반지름:20.0,색깔:red)=1256.00
원의 둘레(반지름:10.0)=20.00
원의 둘레(반지름:20.0)=40.00

```

전체 circle 객체수는 ?

```
20
21 public double getRadius() {
22     return radius;
23 }
24
25 double findArea() {
26     return PI * getRadius() * getRadius();
27 }
28 double getCircumference() {
29     return 2 * PI * getRadius();
30 }
31
32 }
33
34 public class Circledemo {
35     public static void main(String[] args){
36         Scanner in = new Scanner(System.in);
37
38         ArrayList<Circle> circles = new ArrayList<Circle>();
39
40         int radius;
41         String colors;
42
43         System.out.print("input radius: ");
44
45         while(( radius = in.nextInt()) >= 0) {
46             System.out.print("input color: ");
47             colors = in.next();
48             Circle c = new Circle(radius, colors);
49             circles.add(c);
50             System.out.println("생성된 원의 개수: " + c.getNumOfCircles());
51             System.out.print("input radius: ");
52
53         }
54         in.close();
55         System.out.println("total number of circle: " + Circle.numOfCircles);
56         for(int i = 0 ; i < circles.size() ; i++) {
57             Circle c = circles.get(i);
58             System.out.printf("원의 넓이 (반지름: %.1f, 색깔: %s) = %.2f\n",
59                 c.getRadius(), c.color, c.findArea());
60         }
61         System.out.println("-----");
62         for(Circle c : circles) {
63             System.out.printf("원의 넓이 (반지름: %.1f, 색깔: %s) = %.2f\n",
64                 c.getRadius(), c.color, c.findArea());
65         }
66         System.out.println("-----");
67         for(Circle c : circles) {
68             System.out.printf("(반지름: %.1f)원의 둘레 = %.2f\n",
69                 c.getRadius(), c.getCircumference());
70         }
71     }
72
73 }
74
75 }
```

```
class Circle {
    private double radius;
    String color;

    public String getcolor() {
        return color;
    }
    public void setcolor(String color) {
        this.color=color;
    }

    public Circle(int radius,String color) {
        this.setRadius(radius);
        this.setcolor(color);
    }
    double getCircumference() {
        return 2 * getRadius();
    }

    double findArea () {
        return getRadius() * getRadius() * 3.14;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(int radius) {
        this.radius = radius;
    }
}
```

```

62 {
63     static int numofCircles=0;
64     final double PI=3.14;
65     private double radius;
66     String[] color= {"red","orange","yellow","green","blue","purple"};
67
68
69     public Circle()
70     {
71         String[] color;
72         numofCircles++;
73     }
74
75     void setRadius(int r)
76     {
77         this.radius=r;
78     }
79
80     double getRadius()
81     {
82         return radius;
83     }
84
85     double findArea()
86     {
87         return PI*radius*radius;
88     }
89
90     double getCircumference()
91     {

```

color는 객체 과정에서 정해짐

```

1
2 import java.util.Scanner;
3 import java.util.ArrayList;
4
5 public class Circle_0505
6 {
7     public static void main(String[] arg)
8     {
9         int z=0;
10
11         ArrayList<Circle> circles = new ArrayList<Circle>();
12         Scanner in = new Scanner(System.in);
13         System.out.print("radius:");
14
15         int radius=0;
16
17         while(radius>=0)
18         {
19             radius=in.nextInt();
20             Circle c = new Circle();
21             if(radius<0)
22             {
23                 Circle.numofCircles--;
24                 break;
25             }
26             c.setRadius(radius);
27             circles.add(c);
28             System.out.print("radius:");
29         }
30
31         System.out.println("생성된 원의 수 : " + Circle.numofCircles);

```

```
4 public class findArea {  
5  
6     double radius;  
7  
8     public void circle(){  
9         this.radius = 0;  
10    }  
11    public double getRadius() {  
12        return radius;  
13    }  
14  
15    public void setRadius(double radius) {  
16        this.radius = radius;  
17    }  
18  
19    public double extent(){  
20        return radius*radius*3.14;  
21    }  
22  
23    public double circumference(){  
24        return 2*radius*3.14;  
25    }  
26 }  
27
```



```

class Circle {
    private double radius;
    private double area;
    private double round;
    private String color;

    public Circle(double radius, String color) {
        super();
        this.radius = radius;
        this.color = color;
        this.area = radius*radius*3.14;
        this.round = radius*2*3.14;
    }

    public void printArea() {
        System.out.println("원의 넓이(반지름 : " + radius + ", 색깔 : " + color
        + ") = " + String.format("%.2f", area));
    }

    public void printRound() {
        System.out.println("원의 둘레(반지름 : " + radius + ") = " +
        String.format("%.2f", round));
    }
}

```

getter()와
getter()

전체 circle 객체수는 ?

```

public class CircleDemo3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Circle> circleList = new ArrayList<Circle>();

        while(true) {
            System.out.print("Input Radius : ");
            int radius = sc.nextInt();
            if(radius == -1) {
                break;
            }
            System.out.print("Input Color : ");
            String color = sc.next();
            circleList.add(new Circle((double)radius, color));
        }

        System.out.println("Total number of Circle : " + circleList.size());

        for(int i=0; i<circleList.size(); i++) {
            circleList.get(i).printArea();
        }

        for (Circle circle : circleList) {
            circle.printArea();
        }

        for(int i=0; i<circleList.size(); i++) {
            circleList.get(i).printRound();
        }
    }
}

```

```

1 package lab2;
2 import java.util.ArrayList;
3 import java.util.Scanner;
4
5 public class ArrayListCircleDemo {
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8
9         ArrayList<Circle> circles = new ArrayList<Circle>();
10
11         Scanner in = new Scanner(System.in);
12
13         int radius;
14         int m = 0;
15         System.out.print("Input Radius: ");
16         while((radius = in.nextInt())>=0) {
17             Circle c = new Circle();
18             c.setRadius(radius);
19             circles.add(c);
20             System.out.print("Input Colors: ");
21             c.color = in.next();
22             System.out.print("Input Radius: ");
23         }
24         System.out.println("Total number of Circle : "+ Circle.count);
25         for(int i=0;i<circles.size();i++) {
26             Circle c = circles.get(i);
27             System.out.printf("원의 넓이(반지름: %.1f, 색깔: %s) = %.2f\n", c.getRadius(), c.color, c.findArea());
28         }
29         for(Circle a:circles) {
30             Circle c=circles.get(m);
31             m++;
32             System.out.printf("원의 넓이(반지름: %.1f, 색깔: %s) = %.2f\n", c.getRadius(), c.color, c.findArea());
33         }
34
35         for(int i=0;i<circles.size();i++) {
36             Circle c = circles.get(i);
37             System.out.printf("원의 둘레(반지름: %.1f) = %.2f\n", c.getRadius(), c.getCircumference());
38         }
39     }

```

캡슐화 문제 color

```

class Circle{
    private double radius;
    String color;
    static double count = 0.0;

    public double getRadius() {
        return radius;
    }
    public void setRadius(double c) {
        count = count + 1;
        radius =c;
    }
    double getCircumference() {
        return 3.14*2*radius;
    }
    double findArea() {
        return 3.14*radius*radius;
    }
}

```

```

4
5 class Circle {
6     static int count = 0;
7
8     private double radius;
9     private String color;
10
11 public Circle(double radius, String color) {
12     this.radius = radius;
13     this.color = color;
14     this.count++;
15 }
16
17 public double getRadius() {
18     return this.radius;
19 }
20 public String getColor() {
21     return this.color;
22 }
23
24 public void setRadius(double radius) {
25     this.radius = radius;
26 }
27 public void setColor(String color) {
28     this.color = color;
29 }
30
31 public double getCircumference() {
32     return 3.14 * 2.0 * this.radius;
33 }
34 public double getArea() {
35     return 3.14 * this.radius * this.radius;
36 }
37 }
38

```

```

public class Week8 {
    public static void main(String[] args) {

        ArrayList<Circle> circleList = new ArrayList<Circle>();
        Scanner sc = new Scanner(System.in);

        String color;
        double radius;

        while (true) {
            System.out.print("Input Radius: ");
            radius = sc.nextDouble();
            if (radius < 0) { break; }

            System.out.print("Input Color: ");
            color = sc.next();

            Circle c = new Circle(radius, color);
            circleList.add(c);
        }

        System.out.println(String.format("Total number of Circle: %d", Circle.count));

        for(int i = 0; i < circleList.size(); i++) {
            Circle c = circleList.get(i);
            System.out.println(String.format(
                "원의 넓이(반지름: %.1f, 색깔: %s) = %.2f",
                c.getRadius(), c.getColor(), c.getArea()
            ));
        }

        for(Circle c : circleList) {
            System.out.println(String.format(
                "원의 넓이(반지름: %.1f, 색깔: %s) = %.2f",
                c.getRadius(), c.getColor(), c.getArea()
            ));
        }

        for(Circle c : circleList) {
            System.out.println(String.format(
                "원의 둘레(반지름: %.1f) = %.2f",
                c.getRadius(), c.getCircumference()
            ));
        }
    }
}

```

수업목표

- 상속을 통해 얻을 수 있는 장점을 이해한다 (복사하지 말고 상속하자)
 - 재사용 과 간결성
- 상속 클래스를 작성한다
- 오버로딩(overloading)과 오버라이딩(overriding)의 차이점을 이해한다

- 상속은 모듈의 재사용과 코드의 간결성을 제공함
- 상속의 절차는 일반적이고 개념적인 수퍼 클래스를 정의하고, 이 클래스를 닮은 새로운 서브 클래스를 선언
- 새로운 서브 클래스에는 수퍼 클래스에 수정이나 확장이 필요한 부분만을 재정의

클래스 상속

```
1 public class Animal {  
2     String eye;  
3     String mouth;  
4  
5     void eat() {  
6     }  
7  
8     void sleep() {  
9     }  
10 }  
11  
12 class Eagle extends Animal {  
13     String wing;  
14  
15     void fly() {  
16     }  
17 }  
18  
19 class Tiger extends Animal {  
20     String leg;  
21  
22     void run() {  
23     }  
24 }  
25  
26 class Goldfish extends Animal {  
27     String fin;  
28  
29     void swim() {  
30     }  
31 }
```



메서드 오버라이딩

```
1 public class Circle {
2     private void secret() {
3         System.out.println("Secrete");
4     }
5     protected void findRadius() {
6         System.out.println("Radius is 10.0");
7     }
8
9     public void findArea() {
10        System.out.println("Area is PI * radius * radius");
11    }
12 }

2 public class Ball extends Circle {
3     private String color;
4
5
6     public Ball(String color) {
7         this.color = color;
8     }
9
10    public void findColor() {
11        System.out.println(color + " Ball");
12    }
13
14
15    public void findVolume() {
16        System.out.println("Volume is 4/3*(PI* radius * radius* radius).");
17    }
18
19    // Ball 의 넓이를 구하기 위해 override 함
20    public void findArea() {
21        System.out.println("Area us 4*(PI* radius * radius)");
22    }
23 }
24 }
```



Die 클래스, EvenDie 와 OddDie 클래스
