Java 프로그래밍

7주차 과제

학 과	소프트웨어전공
학 번	2126097
이 름	이동민
제 출 일	2024. 04. 17.

문제 1번: 원을 표현하는 다음 Circle 클래스가 있다.

```
class Circle {
  private int radius;
  public Circle(int radius) { this.radius = radius; }
  public int getRadius() { return radius; }
}
```

Circle 클래스를 상속받은 NamedCircle 클래스를 작성하여, 다음 main()을 실행할 때 다음 실행 결과와 같이 출력되도록 하라.

```
public static void main(String[] args) {
    NamedCircle w = new NamedCircle(5, "Waffle");
    w.show();
}
```

```
Waffle, 반지름 = 5
```

Source Code

```
class Circle {
  private int radius;

  public Circle(int radius) {
    this.radius = radius;
  }

  public int getRadius() {
    return radius;
  }
}

public class NamedCircle extends Circle {
  private String name;

  public NamedCircle(int radius, String name) {
    super(radius);
    this.name = name;
  }

  public void show() {
    System.out.printf("%s, 반지름 = %d\n", name, getRadius());
  }

  public static void main(String[] args) {
    NamedCircle w = new NamedCircle(5, "Waffle");
    w.show();
  }
}
```

출력 결과

```
Waffle, 반지름 = 5
```

문제 2번: 인터페이스 AdderInterface의 코드는 다음과 같다.

```
interface AdderInterface {
  int add(int x, int y);
  int add(int n);
}
```

AdderInterface를 상속받은 클래스 MyAdder를 작성하여, 다음 main()을 실행할 때 아래 실행 결과와 같이 출력되도록 하라.

```
15
55
```

Source Code

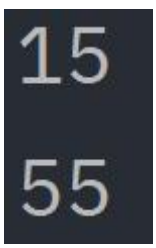
```
interface AdderInterface {
  int add(int x, int y);
  int add(int n);
}

public class MyAdder implements AdderInterface {
  private int sum = 0;

  public int add(int x, int y) {
    return x + y;
  }
  public int add(int n) {
    for (int i = 0; i <= n; i++) {
      sum += i;
    }
    return sum;
  }

  public static void main(String[] args) {
    MyAdder adder = new MyAdder();
    System.out.println(adder.add(5, 10));
    System.out.println(adder.add(10));
  }
}
```

출력 결과



문제 3번: 다음 코드와 실행 결과를 참고하여 추상 클래스 Calculator를 상속받는 Adder와 Subtracter 클래스를 작성하라.

```
import java.util.Scanner;

abstract class Calculator {
  protected int a, b;
  abstract protected int calc();
  protected void input() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("정수 2개를 입력하세요 >> ");
    a = scanner.nextInt();
    b = scanner.nextInt();
  }
  public void run() {
    input();
    int res = calc();
    System.out.println("계산된 값은 " + res);
  }
}

// Adder Subtracter 클래스 작성

public class App {

  public static void main(String[] args) {
    Adder adder = new Adder();
    Subtracter sub = new Subtracter();

    adder.run();
    sub.run();
  }

}
```

```
정수 2개를 입력하세요>>5 3
계산된 값은 8
정수 2개를 입력하세요>>3 5
계산된 값은 -2
```

Source Code는 다음 페이지에 있습니다.

```
import java.util.Scanner;

abstract class Calculator {
  protected int a, b;

  abstract protected int calc();

  protected void input() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("정수 2개를 입력하세요 >> ");
    a = scanner.nextInt();
    b = scanner.nextInt();
  }

  public void run() {
    input();
    int res = calc();
    System.out.println("계산된 값은 " + res);
  }
}


class Adder extends Calculator {
  @Override
  public int calc() {
    return a + b;
  }
}

class Subtracter extends Calculator {
  @Override
  public int calc() {
    return a - b;
  }
}


public class App {

  public static void main(String[] args) {
    Adder adder = new Adder();
    Subtracter sub = new Subtracter();

    adder.run();
    sub.run();
  }

}
```

출력 결과

```
정수 2개를 입력하세요 >> 7 5
계산된 값은 12
정수 2개를 입력하세요 >> 5 7
계산된 값은 -2
```

문제 4번: 2차원 상의 한 점을 표현하는 Point 클래스는 다음과 같다.

```
class Point {
  private int x, y;
  public Point(int x, int y) { this.x = x; this.y = y; }
  public int getX() { return x; }
  public int getY() { return y; }
  protected void move(int x, int y) { this.x = x; this.y = y; }
}
```

다음 main()과 실행 결과를 참고하여 Point를 상속받은 ColorPoint 클래스(main() 포함)를 작성하라.

```
public static void main(String[] args) {
  ColorPoint cp = new ColorPoint(5, 5, "YELLOW");
  cp.setPoint(10, 20);
  cp.setColor("GREEN");
  cp.show();
}
```

```
GREEN색으로(10,20)
```

Source Code는 다음 페이지에 있습니다.

```
class Point {
  private int x, y;
  public Point(int x, int y) {
    this.x = x;
    this.y = y;
  }
  public int getX() {
    return x;
  }
  public int getY() {
    return y;
  }
  protected void move(int x, int y) {
    this.x = x;
    this.y = y;
  }
}

public class ColorPoint extends Point {
  private String color;

  public ColorPoint(int x, int y, String color) {
    super(x, y);
    this.color = color;
  }

  public void setPoint(int x, int y) {
    move(x, y);
  }

  public void setColor(String color) {
    this.color = color;
  }

  public void show() {
    System.out.printf("%s색으로(%d,%d)", color, getX(), getY());
  }

  public static void main(String[] args) {
    ColorPoint cp = new ColorPoint(5, 5, "YELLOW");
    cp.setPoint(10, 20);
    cp.setColor("GREEN");
    cp.show();
  }
}
```

출력 결과

```
GREEN색으로(10,20)
```

문제 5번: 다음 StackInterface는 문자열을 푸시하고 팝할 수 있는 스택에 대한 스펙을 정의하고 있다. StackInterface를 상속받는 StringStack 클래스를 구현하라. 그리고 StackManager 클래스에 main() 메소드를 작성하여 StringStack 객체를 생성하고, 사용자로부터 문자열을 5개 읽어 스택 객체를 저장하고, 다시 팝하여 읽은 반대순으로 출력하라.

```
interface StackInterface {
  int length();
  String pop();
  boolean push(String ob);
}
```

```
>>I love sunny very much
much very sunny love I
```

Source Code는 다음 페이지에 있습니다.

```
import java.util.Scanner;

interface StackInterface {
  int length();
  String pop();
  boolean push(String ob);
}

class StringStack implements StackInterface {
  private String[] stackArray;
  private int length;
  private int top;

  public StringStack(int length) {
    this.length = length;
    stackArray = new String[length];
    this.top = -1;
  }

  @Override
  public int length() {
    return top + 1;
  }

  @Override
  public String pop() {
    if (top == -1) {
      System.out.println("Stack is empty");
      return null;
    }
    else {
      String popped = stackArray[top];
      top--;
      return popped;
    }
  }

  @Override
  public boolean push(String ob) {
    if (top == length - 1) {
      System.out.println("Stack is full");
      return false;
    }
    else {
      top++;
      stackArray[top] = ob;
      return true;
    }
  }
}

public class StackManager {

  public static void main(String[] args) {
    StringStack st = new StringStack(5);

    Scanner sc = new Scanner(System.in);
    System.out.print("문자열 5개 입력 >> ");

    for (int i = 0; i < 5; i++) {
      String input = sc.next();
      st.push(input);
    }

    while (st.length() > 0) {
      System.out.print(st.pop() + " ");
    }
  }
}
```

출력 결과

```
문자열 5개 입력 >> I love java very much
much very java love I
```

문제 6번: 간단한 그래픽 편집기를 만들어보자. 본문 5.6절의 메소드 오버라이딩과 5.7절의 추상 클래스의 설명 중에 Line, Rect, Circle 클래스 코드를 활용하여, 다음 실행 결과처럼 동작하는 프로그램을 작성하라.

Source Code

```
import java.util.Scanner;

abstract class Shape {
  abstract void draw();
}

class Line extends Shape {
  @Override
  public void draw() {
    System.out.println("Line");
  }
}

class Rectangle extends Shape {
  @Override
  public void draw() {
    System.out.println("Rectangle");
  }
}

class Circle extends Shape {
  @Override
  public void draw() {
    System.out.println("Circle");
  }
}



public class GraphicEditor {
  static void paint(Shape p) {
    p.draw();
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Shape[] shapes = new Shape[100];
    int shapeCount = 0;

    while (true) {
      System.out.print("삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> ");
      int action = sc.nextInt();
      sc.nextLine();
```

```
      switch (action) {
        case 1:
          if (shapeCount >= shapes.length) {
            System.out.println("더 이상 도형을 추가할 수 없습니다.");
            break;
          }
          System.out.print("도형 종류 Line(1), Rect(2), Circle(3) >> ");
          int shapeOption = sc.nextInt();
          Shape shape;
          switch (shapeOption) {
            case 1:
              shape = new Line();
              break;
            case 2:
              shape = new Rectangle();
              break;
            case 3:
              shape = new Circle();
              break;
            default:
              System.out.println("올바른 도형 옵션을 선택하세요.");
              continue;
          }
          shapes[shapeCount] = shape;
          shapeCount++;
          break;

        case 2:
          if (shapeCount == 0) {
            System.out.println("삭제할 도형이 없습니다.");
          }
          System.out.print("삭제할 도형의 위치 >> ");
          int shapeRemoveNum = sc.nextInt();
          if (shapeRemoveNum < 1 || shapeRemoveNum > shapeCount) {
            System.out.println("삭제할 수 없습니다.");
          }
          else {
            for (int i = shapeRemoveNum - 1; i < shapeCount - 1; i++) {
              shapes[i] = shapes[i + 1];
            }
          }
          shapeCount--;
          break;

        case 3:
          for (int i = 0; i < shapeCount; i++) {
            shapes[i].draw();
          }
```

```
          break;

        case 4:
          System.out.println("프로그램을 종료합니다...");
          System.exit(0);

        default:
          System.out.println("올바른 옵션을 입력해주세요.");
          break;
      }
    }
  }
}
```

출력 결과

```
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 1
도형 종류 Line(1), Rect(2), Circle(3) >> 1
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 1
도형 종류 Line(1), Rect(2), Circle(3) >> 3
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 3
Line
Circle
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 2
삭제할 도형의 위치 >> 3
삭제할 수 없습니다.
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 4
프로그램을 종료합니다...
```

보너스 문제: 다음은 도형을 묘사하는 인터페이스 Shape이다.

```
interface Shape {
  final double PI = 3.14;
  void draw();
  double getArea();
  default public void redraw() {
    System.out.println("--- 다시 그립니다. ---");
    draw();
  }
}
```

다음 main() 메소드와 실행 결과를 참고하여, 인터페이스 Shape를 구현한 클래스 Circle를 작성하고 전체 프로그램을 완성하라.

```
public class ShapeApp {
  public static void main(String[] args) {
    Shape coin = new Circle(10);
    coin.redraw();
    System.out.println("코인의 면적은 " + coin.getArea());
  }
}
```

Source Code는 다음 페이지에 있습니다.

```
interface Shape {
  final double PI = 3.14;
  void draw();
  double getArea();
  default public void redraw() {
    System.out.println("--- 다시 그립니다. ---");
    draw();
  }
}

class Circle implements Shape {
  private int radius;

  Circle(int radius) {
    this.radius = radius;
  }

  public void draw() {
    System.out.printf("반지름 %d ", radius);
  }
  public double getArea() {
    return PI * radius * radius;
  }
}

public class ShapeApp {
  public static void main(String[] args) {
    Shape coin = new Circle(10);
    coin.redraw();
    System.out.println("코인의 면적은 " + coin.getArea());
  }
}
```

출력 결과