Họ và tên : Phạm Văn Phúc

MSSV : 522H0068

Câu 2

Code :

public abstract class Shape {

    protected String color;

    protected boolean filled;

    public Shape() {

*this*.color = "red";

*this*.filled = true;

    }

    public Shape(String *color*, boolean *filled*) {

*this*.color = *color*;

*this*.filled = *filled*;

    }

    public void setColor(String *color*) {

*this*.color = *color*;

    }

    public String getColor() {

        return *this*.color;

    }

    public boolean isFilled() {

        return *this*.filled;

    }

    public void setFilled(boolean *filled*) {

*this*.filled = *filled*;

    }

    public abstract double getArea();

    public abstract double getPerimeter();

    public String toString() {

        return "Color : " + color + ", Filled : " + filled + "|";

    }

    public abstract boolean equals(Shape *s*);

}

public class Rectangle extends Shape {

    protected double width;

    protected double length;

    public Rectangle() {

*super*();

*this*.width = 1.0;

*this*.length = 1.0;

    }

    public Rectangle(double *width*, double *length*) {

*super*();

*this*.width = *width*;

*this*.length = *length*;

    }

    public Rectangle(double *width*, double *length*, String *color*, boolean *filled*) {

*super*(*color*, *filled*);

*this*.width = *width*;

*this*.length = *length*;

    }

    public double getWidth() {

        return *this*.width;

    }

    public void setWidth(double *width*) {

*this*.width = *width*;

    }

    public double getLength() {

        return *this*.length;

    }

    public void setLength(double *length*) {

*this*.length = *length*;

    }

    public double getArea() {

        return *this*.width \* *this*.length;

    }

    public double getPerimeter() {

        return (*this*.length + *this*.width) \* 2;

    }

    public boolean equals(Shape *s*) {

        if (*s* instanceof Rectangle) {

            Rectangle temp = (Rectangle) *s*;

            if (*s*.getArea() == temp.getArea() && *s*.getPerimeter() == temp.getPerimeter()) {

                return true;

            } else {

                return false;

            }

        } else {

            return false;

        }

    }

    public String toString() {

        return *super*.toString() + " Rectangle : width =  " + width + ", length =  " + length + ", Area = " + getArea()

                + "|";

    }

}

public class Circle extends Shape {

    private double radius;

    public Circle() {

*super*();

*this*.radius = 1.0;

    }

    public Circle(double *radius*, String *color*, boolean *filled*) {

*super*(*color*, *filled*);

*this*.radius = *radius*;

    }

    public double getRadius() {

        return *this*.radius;

    }

    public void setRadius(double *radius*) {

*this*.radius = *radius*;

    }

    public double getArea() {

        return *this*.radius \* *this*.radius \* 3.14;

    }

    public double getPerimeter() {

        return *this*.radius \* 2 \* 3.14;

    }

    public boolean equals(Shape *s*) {

        if (*s* instanceof Circle) {

            Circle temp = (Circle) *s*;

            if (*s*.getArea() == temp.getArea() && *s*.getPerimeter() == temp.getPerimeter()) {

                return true;

            } else {

                return false;

            }

        } else {

            return false;

        }

    }

    public String toString() {

        return *super*.toString() + " Circle : radius = " + radius + ", Area = " + getArea() + ", Perimeter = "

                + getPerimeter();

    }

}

public class Square extends Rectangle {

    private double side;

    public Square() {

*this*.side = 1.0;

    }

    public Square(double *side*) {

*super*(*side*, *side*);

*this*.side = *side*;

    }

    public Square(double *side*, String *color*, boolean *filled*) {

*this*.side = *side*;

*this*.color = *color*;

*this*.filled = *filled*;

*super*.setLength(*side*);

*super*.setWidth(*side*);

    }

    public double getSide() {

        return *this*.side;

    }

    public void setSide(double *side*) {

*this*.side = *side*;

*super*.setLength(*side*);

*super*.setWidth(*side*);

    }

    public double getLength() {

        return *this*.length;

    }

    public void setLength(double *length*) {

*super*.setLength(*length*);

*this*.side = *length*;

    }

    public void setWidth(double *length*) {

*super*.setWidth(*length*);

*this*.side = *length*;

    }

    public boolean equals(Shape *s*) {

        if (*s* instanceof Square) {

            Square temp = (Square) *s*;

            if (*s*.getArea() == temp.getArea() && *s*.getPerimeter() == temp.getPerimeter()) {

                return true;

            } else {

                return false;

            }

        } else {

            return false;

        }

    }

    public String toString() {

        return *super*.toString() + "Square: side = " + side;

    }

}

Class test

public class test {

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

        System.out.println("522H0068 - Pham Van Phuc");

        Shape arrShape[] = new Shape[5];

        arrShape[0] = new Circle(5, "blue", false);

        arrShape[1] = new Rectangle(4, 3, "red", true);

        arrShape[2] = new Square(7, "grey", false);

        arrShape[3] = new Rectangle(4, 3, "brown", true);

        arrShape[4] = new Circle(5, "blue", false);

        System.out.println("Shape 1 = Shape 5 : " + arrShape[0].equals(arrShape[4]));

        double max = arrShape[0].getArea();

        int index = 0;

        for (int i = 1; i < 5; i++) {

            if (max < arrShape[i].getArea()) {

                max = arrShape[i].getArea();

                index = i;

            }

        }

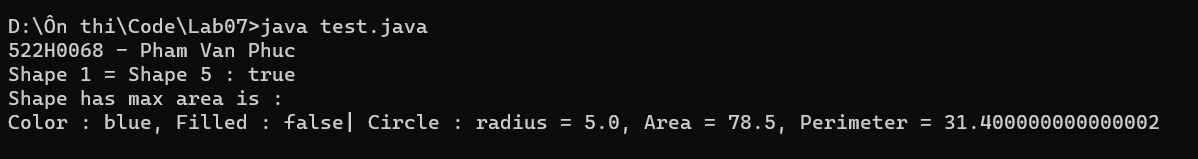
        System.out.println("Shape has max area is : ");

        System.out.println(arrShape[index]);

    }

}

Màn hình chạy



Câu 4:

Code :

public interface *GeometricObject* {

    public double getArea();

    public double getPerimeter();

}

public class Circle implements *GeometricObject* {

    public double radius;

    public Circle() {

*this*.radius = 1.0;

    }

    public void setRadius(double *radius*) {

*this*.radius = *radius*;

    }

    public double getRadius() {

        return *this*.radius;

    }

    public double getArea() {

        return *this*.radius \* *this*.radius \* 3.14;

    }

    public double getPerimeter() {

        return *this*.radius \* 2 \* 3.14;

    }

}

public interface *Resizable* {

    public void resize(int *percent*);

}

public class ResizableCircle extends Circle implements *Resizable* {

    public int percent;

    public ResizableCircle() {

*this*.percent = 100;

    }

    public void setPercent(int *percent*) {

*this*.percent = *percent*;

    }

    public int getPercent() {

        return *this*.percent;

    }

    public void resize(int *percent*) {

*this*.radius = (double) (*this*.radius \* *percent* / 100);

    }

}

public class test {

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

        ResizableCircle circle1 = new ResizableCircle();

        circle1.setRadius(5);

        System.out.println("Area before resizing: " + circle1.getArea());

        circle1.resize(50); // resize the circle by 50%

        System.out.println("Area after resizing: " + circle1.getArea());

    }

}

Màn hình chạy

