|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  OBJECT ORIENTED PROGRAMMING  *Lab Mid OOP* | **submitted by:**  **Shahzaneer Ahmed**  **registration number:**  **sp21-bcs-087**  **submitted to:**  **mA’M sANEEHA aMIR**  **date of submission:**  **mAy 16, 2022** |

# GeometricObject

## Source Code:

public abstract class GeometricObject{

    private boolean isFilled;

    private String color;

    private double thickness;

    public GeometricObject() {

    }

    public GeometricObject(boolean isFilled, String color, double thickness) {

*this*.isFilled = isFilled;

*this*.color = color;

*this*.thickness = thickness;

    }

    public boolean isIsFilled() {

        return *this*.isFilled;

    }

    public boolean getIsFilled() {

        return *this*.isFilled;

    }

    public void setIsFilled(boolean isFilled) {

*this*.isFilled = isFilled;

    }

    public String getColor() {

        return *this*.color;

    }

    public void setColor(String color) {

*this*.color = color;

    }

    public double getThickness() {

        return *this*.thickness;

    }

    public void setThickness(double thickness) {

*this*.thickness = thickness;

    }

    abstract double calculateArea();

    @Override

    public String toString() {

        return "{" +

            " isFilled='" + isIsFilled() + "'" +

            ", color='" + getColor() + "'" +

            ", thickness='" + getThickness() + "'" +

            "}";

    }

}

# Rectangle

## Source Code:

public class Rectangle extends GeometricObject{

    private double length;

    private double width;

    private double area;

    public Rectangle() {

    }

    public Rectangle(boolean isFilled, String color, double thickness, double length, double width) {

*super*(isFilled, color, thickness);

*this*.length = length;

*this*.width = width;

    }

    public double getLength() {

        return *this*.length;

    }

    public void setLength(double length) {

*this*.length = length;

    }

    public double getWidth() {

        return *this*.width;

    }

    public void setWidth(double width) {

*this*.width = width;

    }

    @Override

    public double calculateArea() {

        double area = *this*.length \* *this*.width;

        setArea(area);

        return area;

    }

    public double getArea() {

        return *this*.area;

    }

    public void setArea(double area) {

*this*.area = area;

    }

    @Override

    public String toString() {

        return "{" +

            " length='" + getLength() + "'" +

            ", width='" + getWidth() + "'" +

            ", area='" + getArea() + "'" +

            "}";

    }

}

# Circle

## Source Code:

public class Circle extends GeometricObject{

    private double radius;

    private double area;

    public Circle() {

    }

    public Circle(boolean isFilled, String color, double thickness, double radius) {

*super*(isFilled, color, thickness);

*this*.radius = radius;

    }

    public double getRadius() {

        return *this*.radius;

    }

    public void setRadius(double radius) {

*this*.radius = radius;

    }

    public double getArea() {

        return *this*.area;

    }

    public void setArea(double area) {

*this*.area = area;

    }

    @Override

    public double calculateArea() {

        double area = Math.PI \* radius \* radius;

        setArea(area);

        return area;

    }

    @Override

    public String toString() {

        return "{" +

            " radius='" + getRadius() + "'" +

            ", area='" + getArea() + "'" +

            "}";

    }

}

# Drawing

## Source Code:

public class Drawing {

*// Drawing will use the Concept of `has a relationship`*

*// Composition in classes*

    private GeometricObject geometricObject[] = new GeometricObject[5];

    public Drawing() {

    }

    public Drawing(GeometricObject [] geometricObject) {

*this*.geometricObject = geometricObject;

    }

    public void add(GeometricObject obj) {

        for (int i = 0; i < geometricObject.length; i++) {

            if (geometricObject[i] == null) {

                geometricObject[i] = obj;

                break;

            }

        }

    }

    public double totalArea() {

        double totalArea = 0;

        for (int i = 0; i < geometricObject.length; i++) {

            totalArea += geometricObject[i].calculateArea();

        }

        return totalArea;

    }

    public int countFilled() {

        int counter = 0;

        for (int i = 0; i < geometricObject.length; i++) {

            if (geometricObject[i].getIsFilled() == true) {

                counter++;

            }

        }

        return counter;

    }

}

# Runner

## Source Code:

public class Runner {

    public static void main(String[] args) {

        Circle c1 = new Circle (true,"Blue",2.5,5);

        Circle c2 = new Circle (false,"red",1.5,10);

        Rectangle r1 = new Rectangle(true,"Blue",2.5,4,5);

        Rectangle r2 = new Rectangle(false,"red",3.6,6,5);

        Rectangle r3 = new Rectangle(true,"green",4.5,1,5);

        Drawing d = new Drawing();

        d.add(c1);

        d.add(c2);

        d.add(r1);

        d.add(r2);

        d.add(r3);

        System.out.println("Total Area of Drawing is " + d.totalArea());

        System.out.println("Number of Filled Figures in the drawing is "+d.countFilled());

    }

}