IK22

Praktikum Abstract class dan Interface

DiSusun Oleh :

Iklima Mardiana, 2008765

PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER

FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PENDIDIKAN INDONESIA

26 APRIL 2022

Implementasi dan Hasil

Implementasi dan Hasil

1. Class GeometricObject

Soal:

(*Triangle class*) Design a new `Triangle` class that extends the abstract `GeometricObject` class. Draw the UML diagram for the classes `Triangle` and `GeometricObject` and then implement the `Triangle` class. Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a `Triangle` object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not.

- UML Diagram

GeometricObject
- color : String - filled : boolean
GeometricObject() GeometricObject(color: String, filled: boolean) + getColor : String + setColor (Color: String) : void + isFilled : boolean + setFilled (filled: boolean) : void + toString : String + getArea : double + getPerimeter : double

- Sourcecode GeometricObject

```
package Pertemuan9.Latihan1;

public abstract class GeometricObject {
    private String color = "white";
    private boolean filled;

    protected GeometricObject() {
    }

    protected GeometricObject(String color, boolean filled) {
        this.color = color;
```

```
        this.filled = filled;
    }

    public String getColor() {
        return color;
    }

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

    public boolean isFilled() {
        return filled;
    }

    public void setFilled(boolean filled) {
        this.filled = filled;
    }

    @Override
    public String toString() {
        return "\nWarna: " + color +
            "\n filled: " + filled;
    }

    public abstract double getArea();
    public abstract double getPerimeter();

}
```

- UML Triangle

Triangle
- side1 : double - side2 : double - side3 : double
+Triangle() +Triangle(side1: double, side2: double, side3: double) +Triangle(side1: double, side2: double, side3: double, color: Strig, filled: boolean)

+ getSide1 : double
+ getSide2 : double
+ getSide3 : double
+ setSide1(side1: double) : void
+ setSide2(side2: double) : void
+ setSide3(side3: double) : void
+ getArea : double
+ getPerimeter : double
+ toString : String

- Sourcecode Triangle

```
package pertemuan10.Latihan1;

public class Triangle extends GeometricObject{
    private double side1;
    private double side2;
    private double side3;

    public Triangle(){
    }

    public Triangle(double side1, double side2, double side3) {
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }

    public Triangle(double side1, double side2, double side3,
        String color, boolean filled) {
        this(side1, side2, side3);
        setColor(color);
        setFilled(filled);
    }

    public double getSide1() {
        return side1;
    }

    public void setSide1(double side1) {
        this.side1 = side1;
    }
```

```
    public double getSide2() {
        return side2;
    }

    public void setSide2(double side2) {
        this.side2 = side2;
    }

    public double getSide3() {
        return side3;
    }
    public void setSide3(double side3) {
        this.side3 = side3;
    }

    @Override
    public double getArea() {
        double s = (side1 + side2 + side3) / 2;
        return Math.sqrt(s * (s - side1) * (s - side2) * (s -
side3));
    }

    @Override
    public double getPerimeter() {
        return side1 + side2 + side3;
    }

    @Override
    public String toString() {
        return super.toString() + "\nArea: " + getArea() +
            "\nPerimeter: " + getPerimeter();
    }
}
```

- Sourcecode main.jav

```
package pertemuan10.Latihan1;
import java.util.Scanner;

public class main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Masukkan sisi-sisi Segitiga: ");
        double side1 = input.nextDouble();
```

```
        double side2 = input.nextDouble();
        double side3 = input.nextDouble();

        System.out.print("Berikan warna: ");
        String color = input.next();

        System.out.print("Apakah Segitiga terisi (true / false)?
");
        boolean filled = input.nextBoolean();

        // Create a Triangle
        Triangle triangle = new Triangle(side1, side2, side3,
color, filled);

        System.out.println(triangle);
    }
}
```

2. Class GeoemtricObject

Soal

(*Enable* **GeometricObject** *comparable*) Modify the **GeometricObject** class to implement the **Comparable** interface, and define a static **max** method in the **GeometricObject** class for finding the larger of two **GeometricObject** objects. Draw the UML diagram and implement the new **GeometricObject** class. Write a test program that uses the **max** method to find the larger of two circles and the larger of two rectangles.

- UML Diagram GeometricObject

GeometricObject
- color : long - filled : boolean
GeometricObject() GeometricObject(color: String, filled: boolean) + getColor() : String + setColor(color: String) : void + isFilled() : boolean + setFilled(filled: boolean) : void

+getMax(GeometricObject: 01, GeometricObject: 02): GeomtericObject.
+ getArea : double + getPerimeter : double

- Sourcecode GeometricObject

```
package pertemuan10.Latihan2;


public abstract class GeometricObject implements
Comparable<GeometricObject>{
    private String color = "white";
    private boolean filled;

    protected GeometricObject() {
    }

    protected GeometricObject(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
    }

    public String getColor() {
        return color;
    }

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

    public boolean isFilled() {
        return filled;
    }

    public void setFilled(boolean filled) {
        this.filled = filled;
    }

    @Override
    public String toString() {
        return "\nWarna: " + color +
            "\nfilled: " + filled;
    }

    @Override
```

```
    public int compareTo(GeometricObject object){
        if(this.getArea()>object.getArea()){
            return 1;
        }
        else if (this.getArea() > object.getArea()){
            return -1;
        }
        else{
            return 0;
        }

    }

    public static GeometricObject max(GeometricObject object1,
GeometricObject object2){
        return object1.compareTo(object2)==1? object1:object2;
    }

    public abstract double getArea();

    public abstract double getPerimeter();

}
```

- UML Rectangle

Rectangle
- Lebar : double - Tinggi : double
+ Rectangle() + Rectangle(Lebar: double, Tinggi: double) + Rectangle(Lebar: double, Tinggi: double, color: String, filled: boolean) + getLebar : double + setLebar(Lebar: double) : void + getTinggi : double + setTinggi(Tinggi: double) : void

+ getArea : double + getPerimeter : double + toString : String

- Sourcecode Rectangle

```
package pertemuan10.Latihan2;

public class Rectangle extends GeometricObject{
    private double Lebar;
    private double Tinggi;

    public Rectangle() {
    }

    public Rectangle(
        double Lebar, double Tinggi) {
        this.Lebar = Lebar;
        this.Tinggi = Tinggi;
    }

    public Rectangle(
        double Lebar, double Tinggi, String color,
boolean filled) {
        this.Lebar = Lebar;
        this.Tinggi = Tinggi;
        setColor(color);
        setFilled(filled);
    }

    public double getLebar() {
        return Lebar;
    }

    public void setLebar(double Lebar) {
        this. Lebar = Lebar;
    }

    public double getTinggi() {
        return Tinggi;
    }

    public void setTinggi(double Tinggi) {
        this.Tinggi = Tinggi;
    }

    @Override
```

```
    public double getArea() {
        return Lebar * Tinggi;
    }

    @Override
    public double getPerimeter() {
        return 2 * (Lebar * Tinggi);
    }

    @Override
    public String toString() {
        return super.toString() + "\nLebar: " +
Lebar + "\nTinggi: " + Tinggi
            + "\nArea: " + getArea() +
"\nPerimeter: " + getPerimeter();
    }

}
```

- UML Circle

Circle
- radius : double - Tinggi : double
+ Circle() + getRadius : double + setRadius : void + getArea : double + getDiameter : double + getPerimater : double + toString : String

- Sourcecode Circle

```
package pertemuan10.Latihan2;

public class Circle extends GeometricObject{
    private double radius;
    public Circle(){
```

```
    }

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

    public Circle(double radius, String color, boolean filled){
        this.radius = radius;
        setColor(color);
        setFilled(filled);
    }

    public double getRadius() {
        return radius;
    }

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

    @Override
    public double getArea() {
        return radius * radius * Math.PI;
    }

    public double getDiameter() {
        return 2 * radius;
    }

    @Override
    public double getPerimeter() {
        return 2 * radius * Math.PI;
    }

    @Override
    public String toString() {
        return super.toString() + "\nRadius: " + radius +
"\nArea: " + getArea() +
            "\nDiameter: " + getDiameter() + "\nPerimeter: " +
getPerimeter();
    }
}
```

- Sourcecode main.java

```
package pertemuan10.Latihan2;
```

```
public class main {
    public static void main(String[] args) {
        Circle circle1 = new Circle(21, "grey", true);
        Circle circle2 = new Circle(22, "black", false);

        System.out.println("\nCircle 1: ");
        print(circle1);

        System.out.println("\nCircle 2: ");
        print(circle2);

        print("\nLingkaran terbesar dari kedua lingkaran
tersebut adalah: ");
        print(Circle.max(circle1, circle2));

        Rectangle rectangle1 = new Rectangle(4, 5, "green",
true);
        Rectangle rectangle2 = new Rectangle(4.2, 5, "orange",
true);

        System.out.println("\nRectangle 1: ");
        print(circle1);

        System.out.println("\nRectangle 2: ");
        print(circle2);

        print("\nPersegi terbesar dari kedua persegi tersebut
adalah: ");
        print(Rectangle.max(rectangle1, rectangle2));
    }

    public static void print(String s) {
        System.out.println(s);
    }

    public static void print(GeometricObject o) {
        System.out.println(o);
    }
}
```