TUGAS PRATIKUM 7

(Sebagai pemenuhan salah satu tugas mata kuliah

Pemograman Berorientasi Objek(Pratikum) Program Studi D3 Teknik Informatika)

Disusun oleh :

Rani Indrianna Sembiring (201511025)

Kelas : 2A

POLITEKNIK NEGERI BANDUNG

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

PRODI D3 TEKNIK INFORMATIKA

2021 – 2022

Exercise 1 : The Circle and Cylinder Classes

- **[Task 1.1] Modify class** Circle
  1. Add variable color : string

     Menambahkan atribut color bertipe String (private)

```java
package ExerciseSatu;

/**
 * The Circle class models a circle with a radius and color.
 * */

public class Circle {  //save as "Circle.java"
    // private instance variable,  not accessible from outside this class
    private double radius;
    private String color;
```

  2. Add Constructor Circle (radius : double, color : string)

     Menambahkan Constructor Circle dengan r bertipe double dan c bertipe String

```java
/* Constructs a Circle instance with the given radius and default color*/
public Circle(double r, String c) {
    this.radius=r;
    this.color=c;
}
```

  3. Add Getter dan Setter for color

     Menambahkan getter dan setter pada color dan juga pada radius

```java
    /* Return the radius */
    /* GETTER and SETTER */
    public double getRadius() {
        return radius;
    }

    public String getColor() {
        return color;
    }

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

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

- **[Task 1.2] Overriding the getArea() method**
  - Menambahkan override dari metode getArea() di subclass Cylinder untuk menghitung luas permukaan Cylinder

```java
@Override
    public double getArea() {
        return 2*Math.PI*getRadius()*getHeight()+2*super.getArea();
    }
```

  - Memperbaiki syntax dari method getVolume() karena getArea() di subclass Sylinder telah di buat menjadi override Maka getVolume() sudah tidak berfungsi lagi Maka perlu memperbaiki syntax dengan cara memanggil super.getArea() * height.

```java
//use superclass method getArea() to get the base area
public double getVolume() {
    return super.getArea()*height;      //Use circle's get area
}
```

- **[Task 1.3] Provide a to String() method**

  - Memberikan method toString() pada class Cylinder dan memberikan override pada method tersebut karena superclass nya sendiri sudah ada method toString()

```java
@Override
    public String toString() {       //in Cylinder class
        return "Cylinder: subclass of " + super.toString()  //use circle's toString()
                    + " height=" + height;
    }
```

  - Kemudian melakukan testing pada method toString() yang ada pada class Cylinder.java dengan memanggil method tersebut pada class TestCylinder.java kemudian menghasilkan compilenya.

Start Page | Circle.java | Cylinder.java | TestCylinder.java

Source | History

```java
3   public class TestCylinder {     //save as "TestCylinder.java"
4       public static void main(String[]args) {
5           //Declare and allocate a new instance of cylinder
6           // with default color, radius and height
7           Cylinder c1 = new Cylinder();
8          // System.out.println("Cylinder:"
9          //  + " radius=" + c1.getRadius()
10         //  + " height="+ c1.getHeight()
11         //  + " base area=" + c1.getArea()
12         //  + " volume=" + c1.getVolume());
13          System.out.println(c1.toString());
14
15          //Declare and allocate a new instance of cylinder
16          // specifying height, with default color and radius
17          Cylinder c2 = new Cylinder(10.0);
18          System.out.println("Cylinder:"
19           + " radius=" + c2.getRadius()
20           + " height="+ c2.getHeight()
21           + " base area=" + c2.getArea()
22           + " volume=" + c2.getVolume());
```

Output - W7_PBO_PR (run-single)

```
compile-single:
run-single:
Cylinder: subclass of Circle[radius=1.0 color=red] height=1.0
Cylinder: radius=1.0 height=10.0 base area=69.11503837897544 volume=31.41592653589793
Cylinder: radius=2.0 height=10.0 base area=150.79644737231007 volume=125.66370614359172
BUILD SUCCESSFUL (total time: 1 second)
```

Exercise 2 : Superclass Shape dan it's Subclass Circle, Rectangle and Square

- **[Task 2.1]**
  - Membuat superclass untuk memanggil **Shape** yang berisi tentang :
    1. Dua variabel color bertipe String and Filled bertipe Boolean
    2. Dua constructor : a no-arg (no-argument) yang menginisialisasi color = "green" dan diisi ke true, dan constructor yang menginisialisasi warna dan diisi ke nilai yang di berikan
    3. Getter and setter untuk semua variabel instan. Dari konvensi, getter and setter untuk variabel Boolean xxx di panggil dengan isXXX() (bukan menjadi getXxx() untuk semua type)
    4. Method tostring() yang mengembalikan " A Shape with color of xxx and Filled/Not Filled"

Shape.java

Source History

```java
package ExerciseDua;

public class Shape {
    private String color;
    private boolean filled;

    public Shape() {
    color = "green";
    filled = true;
}

    public Shape(String c, boolean f) {
        this.color=c;
        this.filled=f;
    }

    // return the color
    public String getColor() {
        return color;
    }

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

    // return the filled
    public boolean isFilled() {
        return filled;
    }
```

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

    public String toString() {
        if(isFilled()) {
            return "A shape with color of " + color + " is Filled";
        }else{
            return "A shape with color of " + color + " isn't Filled";
        }
    }
}
```

- Membuat test program untuk mengetest method yang di definisikan pada Shape

```java
package ExerciseDua;

public class Main {
    public static void main(String args[]) {
        Shape s1 = new Shape();
        System.out.println(s1.toString());
        System.out.println("************************************");
        Shape s2 = new Shape ("green", true);
        System.out.println(s2.toString());
        System.out.println("************************************");

    }
}
```

```
Compiling 1 source file to C:\Users\Asus\Documents\TUGAS PBO\W7_PBO_PR\build\classes
compile-single:
run-single:
A shape with color of red is Filled
************************************
A shape with color of green is Filled
************************************
BUILD SUCCESSFUL (total time: 1 second)
```

- Membuat subclass ke dua dari Shape yang memanggil Circle dan Rectangle

  Kelas Circle yang berisi tentang :

  1. Sebuah variabel radius bertipe double
  2. Tiga konstruktor seperti yang di tunjukkan. Konstruktor no-arg menginisialisasi radius ke 1.0
  3. Getter dan setter untuk variabel radius
  4. Method getArea() dan getPerimeter()
  5. Mengganti method toString() yang di warisi untuk mengembalikan "Circle with radius=xxx, which is a subclass of yyy" , dimana yyy merupakan output dari metode toString() dari superclass.

Shape.Java | Circle.Java

Source | History

```java
package ExerciseDua;

public class Circle extends Shape {
    private double radius;

    public Circle() {
        super();
        radius = 1.0;
    }

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

    public Circle(double r, String c, boolean f) {
        this.radius=r;
        super.setColor(c);
        super.setFilled(f);
    }

    //Getter and Setter
    public double getRadius() {
        return radius;
    }

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

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

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

    public String toString() {
        return "A circle with radius = " + getRadius() + ", which is subclass of\n"
                + super.toString();
    }
}
```

- **Class Rectangle berisi :**
    1. Variabel width dengan tipe data double dan variabel length tipe data double
    2. Tigas constructor seperti yang di tunjukkan. Constructor no-arg menginisialisasi width dan length menjadi 1.0
    3. Getter dan setter untuk semua variabel instan
    4. Metode getArea() dan getPerimeter()
    5. Mengganti method toString() yang di warisi, untuk mengembalikan " A Rectangle with width=xxx and length=zzz, which is a subclass of yyy", dimana yyy = output dari method toString() dari superclass.

```java
package ExerciseDua;

public class Rectangle extends Shape {
    private double width;
    private double length;

    public Rectangle() {
        super();
        this.length=1.0;
        this.width=1.0;
    }

    public Rectangle(double width, double length) {
        super();
        this.width=width;
        this.length=length;
    }

    public Rectangle(double width, double length, String c, boolean f) {
        this.length=length;
        this.width=width;
        super.setColor(c);
        super.setFilled(f);
    }

    //getter and settter
    public double getWidth() {
        return width;
    }

    public void setWidth(double w) {
        this.width=w;
    }

    public double getLength() {
        return length;
    }

    public void setLength(double l) {
        this.length=l;
    }

    public double getArea() {
        return width*length;
    }

    public double getPerimeter() {
        return 2*(width+length);
    }

    public String toString() {
        return "A Rectangle with width = " + width + " and length = " + length
                + ", which is a subclass of \n " + super.toString();
    }
}
```

- **Membuat class Square sebagai subclass dari Rectangle.** Square tidak memiliki variabel instan, tetapi mewarisi width dan length dari superclass Rectangle
    1. Berikan constructor yang sesuai:

    ```java
    public Square(double side) {
        super(side, side);  // Call superclass Rectangle(double, double)
    }
    ```

    2. Mengganti method toString() untuk mengembalikan “ A Square with side=xxx, which is a subclass of yyy” dimana yyy = output dari metode toString() dari superclass
    3. Mengganti setLength() dan setWidth() untuk mengubah width dan length untuk mempertahankan geometri persegi

Shape.java | Circle.java | Rectangle.java | Main.java | Square.java

Source | History

```java
package ExerciseDua;

public class Square extends Rectangle {

    public Square() {
        super();
    }

    public Square(double side) {
        super(side,side);   //call superclass rectangle (double, double)
    }

    public Square(double side, String c, boolean f) {
        super(side,side, c,f);
    }

    public double getSide() {
        return super.getLength();
    }

    public void setside(double side) {
        setWidth(side);
        setLength(side);
    }

    @Override
    public String toString() {
        return "A Square width side= " + getSide() +
                " which is a subclass of \n" + super.toString();
    }
}
```

- **TEST PROGRAM**

Main.java

Source | History

```java
package ExerciseDua;

public class Main {

    public static void main(String args[]) {

        System.out.println("---SHAPE---");
        Shape s1 = new Shape();
        Shape s2 = new Shape("blue", false);
        System.out.println(s1.toString());
        System.out.println(s2.toString());

        System.out.println("---CIRCLE---");
        Circle c1 = new Circle();
        Circle c2 = new Circle(10.0, "white", false);
        System.out.println(c1.toString());
        System.out.println("Luas = " + c1.getArea());
        System.out.println(c2.toString());

        System.out.println("---RECTANGLE---");
        Rectangle r1 = new Rectangle();
        Rectangle r2 = new Rectangle(12.0, 6.0, "black", true);
        System.out.println(r1.toString());
        System.out.println(r2.toString());
        System.out.println("Luas = "+ r2.getArea()+ " Keliling = " + r2.getPerimeter());

        System.out.println("---SQUARE---");
        Square sq1 = new Square();
        Square sq2 = new Square(4.5,"yellow", false);
        System.out.println(sq1.toString());
        System.out.println(sq2.toString());
        System.out.println("Luas = " + sq1.getArea()+" Keliling = "+sq1.getPerimeter());

    }
}
```

Output - W7_PBO_PR (run-single)

```
compile-single:
run-single:
---SHAPE---
A shape with color of green is Filled
A shape with color of blue isn't Filled
---CIRCLE---
A circle [A shape with color of green is Filled], radius= 1.0]
Luas = 3.141592653589793
A circle [A shape with color of white isn't Filled], radius= 10.0]
---RECTANGLE---
Square[Rectangle[A shape with color of green is Filled, width= 1.0, length= 1.0]
Square[Rectangle[A shape with color of black is Filled, width= 12.0, length= 6.0]
Luas = 72.0 Keliling = 36.0
---SQUARE---
Square[Square[Rectangle[A shape with color of green is Filled, width= 1.0, length= 1.0]]
Square[Square[Rectangle[A shape with color of yellow isn't Filled, width= 4.5, length= 4.5]]
Luas = 1.0 Keliling = 4.0
BUILD SUCCESSFUL (total time: 0 seconds)
```