

LAPORAN PRAKTIKUM PBO
Inheritance, Absrtact and Interface



Disusun oleh :
Muhamad Rafli Nur Ikhsan
201511048
D-3 Teknik Informatika 2B

Jurusan Teknik Komputer dan Informatika
Program studi D3 Teknik Informatika
Politeknik Negeri Bandung

1. Task 1

SS akhir program

```
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar=5000:C:\Program Files\JetBrains\IntelliJ IDEA\bin\jre\lib\dtplugin" -jar C:\Program Files\JetBrains\IntelliJ IDEA\bin\classes.jar
Cylinder: radius=1.0 height=1.0 base area=12.566370614359172 volume=3.141592653589793
Cylinder: radius=10.0 height=5.0 base area=12.566370614359172 volume=-753.9822368615503
Cylinder: radius=2.0 height=10.0 base area=150.79644737231007 volume=125.66370614359172

Process finished with exit code 0
```

- 1.1 Modify class Circle, add :

1. variable color : string

```
private double radius;  
private String color;
```

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

```
public Circle() { // 1st (default) constructor
    radius = 1.0;
    color = "red";
}

/** Constructs a Circle instance with the given radius and color */
public Circle(double h, String red) { // 2nd constructor
    height = h;
    color = "red";
}
```

- ### 3. Getter and setter for color

```
public String getColor(){
    return color;
}

public void setColor(String newColor){
    color = newColor;
}
```

- 1.2 :

1. Try overriding the `getArea()` method in the subclass `Cylinder` to compute the surface area ($=2\pi \times \text{radius} \times \text{height} + 2 \times \text{base-area}$) of the cylinder instead of base area.

- ## 2. Fix the getVolume()

- 1.3 :

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

2. Task 2

SS akhir program

```
Square side = 1.0 base area = 1.0 perimeter = 4.0
Square side = 10.0 base area = 100.0 perimeter = 40.0
Rectangle width = 1.0 length = 1.0 base area = 1.0 perimeter = 4.0
Rectangle width = 10.0 length = 10.0 base area = 100.0 perimeter = 40.0
Cylinder: radius = 1.0 base area = 6.283185307179586 volume = 6.283185307179586
Cylinder: radius = 10.0 base area = 628.3185307179587 perimeter = 62.83185307179586
```

- The Shape class contains
 - Two instance variables color (String) and filled (boolean).

```
private String color;
private boolean filled;
```

- Two constructors: a no-arg (no-argument) constructor that initializes the color to "green" and filled to true, and a constructor that initializes the color and filled to the given values.

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

public Shape(String g, boolean t){
    color = g;
    filled = t;
}
```

- Getter and setter for all the instance variables. By convention, the getter for a boolean variable xxx is called isXXX() (instead of getXxx() for all the other types).

```

public String getColor(){
    return color;
}

public boolean getFilled(){
    return filled;
}

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

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

```

- A toString() method that returns "A Shape with color of xxx and filled/Not filled".

```

public String toString(){
    return "A shape with color of green and filled";
}

```

- The Circle class contains
 - An instance variable radius (double).

```
private double radius;
```

- Three constructors as shown. The no-arg constructor initializes the radius to 1.0.

```

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

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

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

```

- Getter and setter for the instance variable radius.

```
public double getRadius() {
    return radius;
}

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

- Methods getArea() and getPerimeter().

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

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

- Override the toString() method inherited, to return "A Circle with radius=xxx, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.

```
public String toString(){
    return "A Circle with radius="+radius+", which is a subclass of"+super.toString();
}
```

- The Rectangle class contains

- Two instance variables width (double) and length (double).

```
private double width;
private double length;
```

- Three constructors as shown. The no-arg constructor initializes the width and length to 1.0.

```

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

public Rectangle(double w, double l){
    super();
    width = w;
    length = l;
}

public Rectangle(String color, boolean filled, double width, double length){
    super(color, filled);
    this.width = width;
    this.length = length;
}

```

- Getter and setter for all the instance variables.

```

public double getWidth() {
    return width;
}

public double getLength() {
    return length;
}

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

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

```

- Methods getArea() and getPerimeter().

```

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

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

```

- Override the toString() method inherited, to return "A Rectangle with width=xxx and length=zzz, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.

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

- The Square class contains

- Provide the appropriate constructors (as shown in the class diagram).

```
public Square (){
    super();
    double side = 1.0;
}

public Square(double side) {
    super(side, side);
}

public Square(double side, String color, boolean filled){
    super(color, filled, side, side);
}
```

- Override the toString() method to return "A Square with side=xxx, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.

```
public String toString(){
    return "A Square with side="+getSide()+" , which is a subclass of "+super.toString();
}
```

- Do you need to override the getArea() and getPerimeter()? Try them out.

```
public double getArea()
{
    return getSide()*getSide();
}

public double getPerimeter()
{
    return 4*getSide();
}
```

- Override the setLength() and setWidth() to change both the width and length, so as to maintain the square geometry.

```
public double getSide(){  
    return super.getLength();  
}  
  
public void setSide(double side){  
    setLength(side);  
    setWidth(side);  
}
```