TUGAS PRATIKUM 9

(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

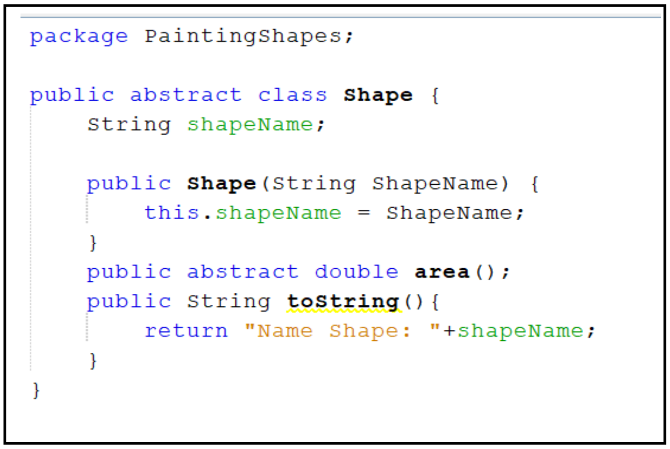
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

PRODI D3 TEKNIK INFORMATIKA

2021 – 2022

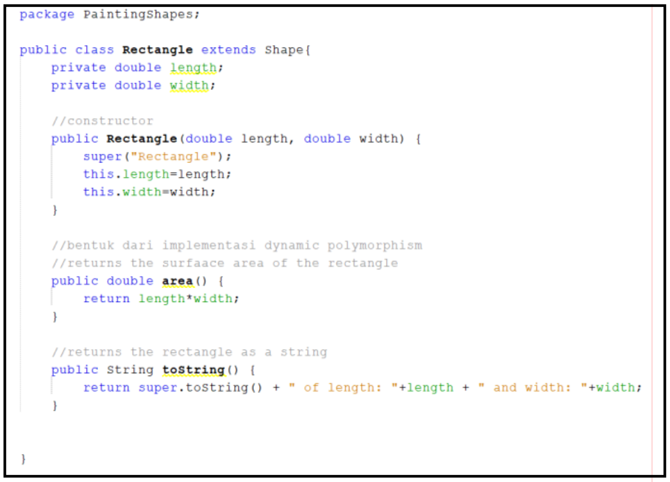
1. Studi kasus 1 – Painting Shapes

- Write an abstract class Shape with the following properties:
  - An instance variable shapeName of type String
  - An abstract method area()
  - A toString method that returns the name of the shape



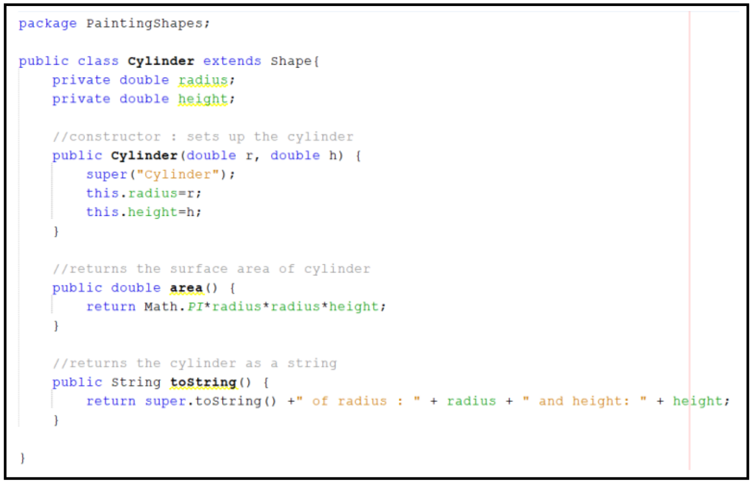
```java
package PaintingShapes;

public abstract class Shape {
    String shapeName;

    public Shape(String ShapeName) {
        this.shapeName = ShapeName;
    }
    public abstract double area();
    public String toString(){
        return "Name Shape: "+shapeName;
    }
}
```

- The file Sphere.java contains a class for a sphere which is a descendant of Shape. A sphere has a radius and its area (surface area) is given by the formula 4*PI*radius^2.
  - Define similar classes for a rectangle and a cylinder. Both the Rectangle class and the Cylinder class are descendants of the Shape class.
  - A rectangle is defined by its **length** and **width** and its area is **length times width.** Then define toString method (similar with Sphere class)



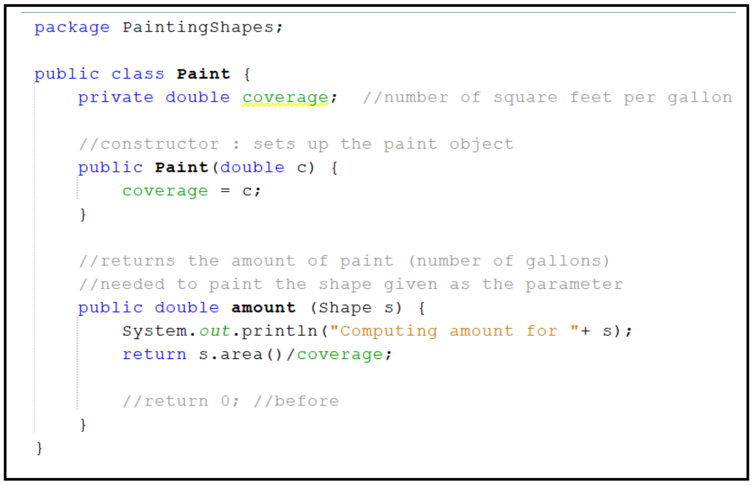
```java
package PaintingShapes;

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

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

    //bentuk dari implementasi dynamic polymorphism
    //returns the surfaace area of the rectangle
    public double area() {
        return length*width;
    }

    //returns the rectangle as a string
    public String toString() {
        return super.toString() + " of length: "+length + " and width: "+width;
    }


}
```

- A cylinder is defined by a radius and height and its area (surface area) **is PI*radius^2*height**. Define the toString method in a way similar to that for the Sphere class.



```java
package PaintingShapes;

public class Cylinder extends Shape{
    private double radius;
    private double height;

    //constructor : sets up the cylinder
    public Cylinder(double r, double h) {
        super("Cylinder");
        this.radius=r;
        this.height=h;
    }

    //returns the surface area of cylinder
    public double area() {
        return Math.PI*radius*radius*height;
    }

    //returns the cylinder as a string
    public String toString() {
        return super.toString() +" of radius : " + radius + " and height: " + height;
    }

}
```

- The file Paint.java contains a class for a type of paint (which has a "coverage" and a method to compute the amountof paint needed to paint a shape). Correct the return statement in the amount method so the correct amount will be returned. Use the fact that the amount of paint needed is the area of the shape divided by the coverage for the paint. (NOTE: Leave the print statement - it is there for illustration purposes, so you can see the method operating on different types of Shape objects.)

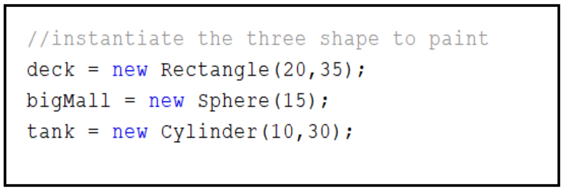


```java
package PaintingShapes;

public class Paint {
    private double coverage;  //number of square feet per gallon

    //constructor : sets up the paint object
    public Paint(double c) {
        coverage = c;
    }

    //returns the amount of paint (number of gallons)
    //needed to paint the shape given as the parameter
    public double amount (Shape s) {
        System.out.println("Computing amount for "+ s);
        return s.area()/coverage;

        //return 0; //before
    }
}
```

- The file PaintThings.java contains a program that computes the amount of paint needed to paint various shapes. A

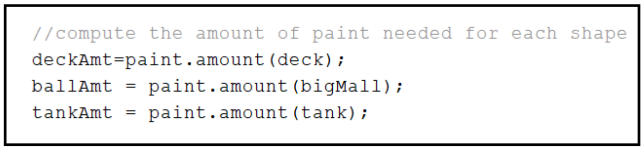
paint object has been instantiated. Add the following to complete the program:

- Instantiate the three shape objects:
  - deck to be a 20 by 35 foot rectangle,
  - bigBall to be a sphere of radius 15, and
  - tank to be a cylinder of radius 10 and height 30.



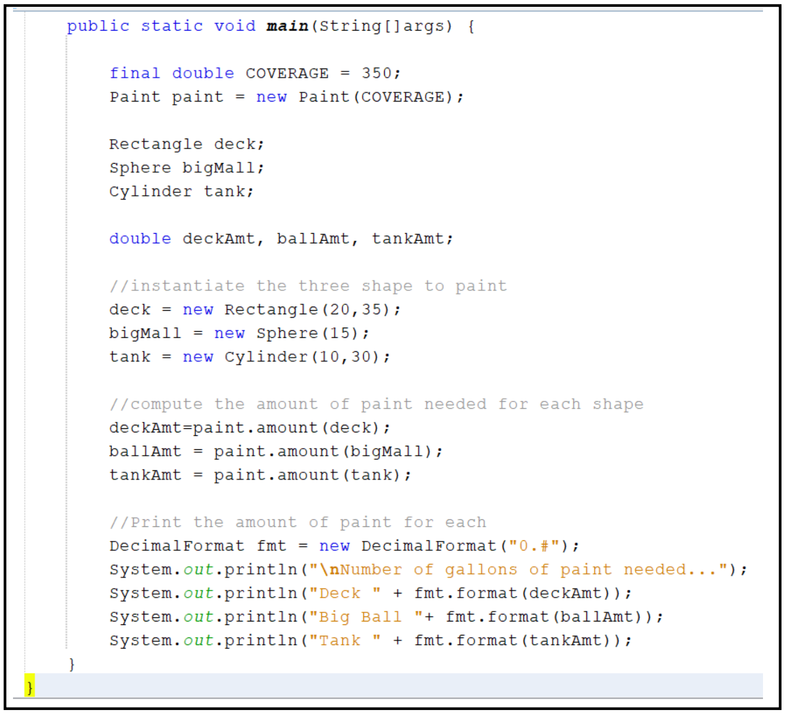
```java
//instantiate the three shape to paint
deck = new Rectangle(20,35);
bigMall = new Sphere(15);
tank = new Cylinder(10,30);
```

- Make the appropriate method calls to assign the correct values to the three amount variables.



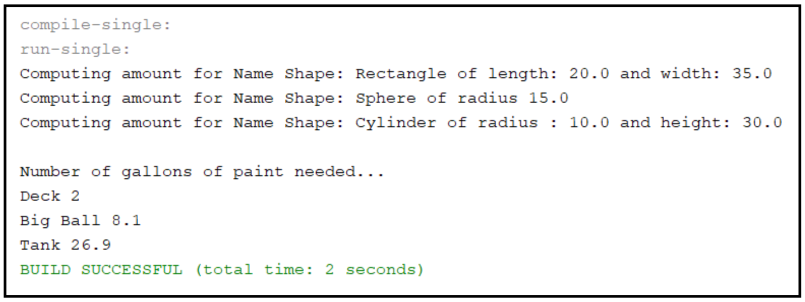
```
//compute the amount of paint needed for each shape
deckAmt=paint.amount(deck);
ballAmt = paint.amount(bigMall);
tankAmt = paint.amount(tank);
```

- Run the program and test it. You should see polymorphism in action as the amount method computes the amount of paint for various shapes.



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

        final double COVERAGE = 350;
        Paint paint = new Paint(COVERAGE);

        Rectangle deck;
        Sphere bigMall;
        Cylinder tank;

        double deckAmt, ballAmt, tankAmt;

        //instantiate the three shape to paint
        deck = new Rectangle(20,35);
        bigMall = new Sphere(15);
        tank = new Cylinder(10,30);

        //compute the amount of paint needed for each shape
        deckAmt=paint.amount(deck);
        ballAmt = paint.amount(bigMall);
        tankAmt = paint.amount(tank);

        //Print the amount of paint for each
        DecimalFormat fmt = new DecimalFormat("0.#");
        System.out.println("\nNumber of gallons of paint needed...");
        System.out.println("Deck " + fmt.format(deckAmt));
        System.out.println("Big Ball "+ fmt.format(ballAmt));
        System.out.println("Tank " + fmt.format(tankAmt));
    }
}
```

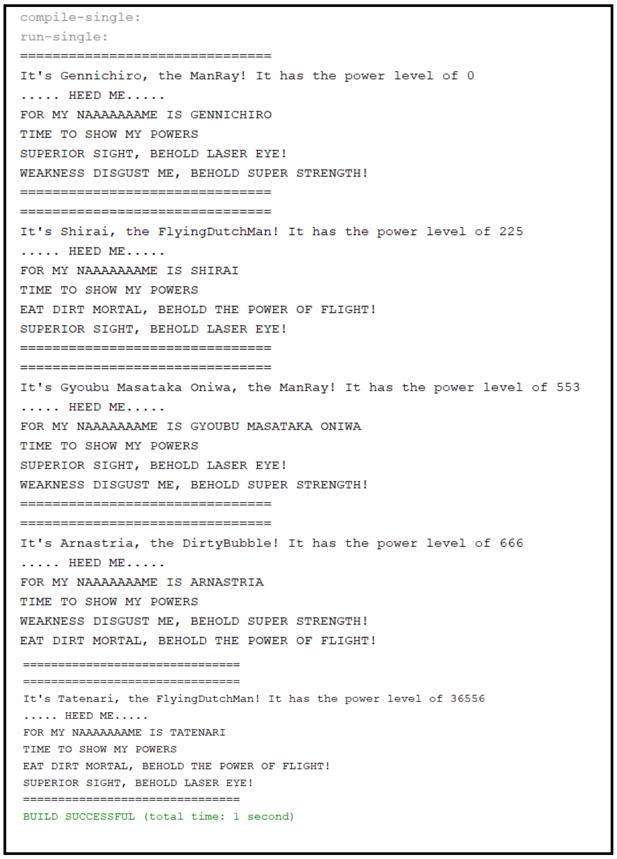
- Hasil Output



```
compile-single:
run-single:
Computing amount for Name Shape: Rectangle of length: 20.0 and width: 35.0
Computing amount for Name Shape: Sphere of radius 15.0
Computing amount for Name Shape: Cylinder of radius : 10.0 and height: 30.0

Number of gallons of paint needed...
Deck 2
Big Ball 8.1
Tank 26.9
BUILD SUCCESSFUL (total time: 2 seconds)
```

2. Studi Kasus 2 : TheAvenged

- Screenshoot hasil program



```
compile-single:
run-single:
===============================
It's Gennichiro, the ManRay! It has the power level of 0
..... HEED ME.....
FOR MY NAAAAAAAME IS GENNICHIRO
TIME TO SHOW MY POWERS
SUPERIOR SIGHT, BEHOLD LASER EYE!
WEAKNESS DISGUST ME, BEHOLD SUPER STRENGTH!
===============================
===============================
It's Shirai, the FlyingDutchMan! It has the power level of 225
..... HEED ME.....
FOR MY NAAAAAAAME IS SHIRAI
TIME TO SHOW MY POWERS
EAT DIRT MORTAL, BEHOLD THE POWER OF FLIGHT!
SUPERIOR SIGHT, BEHOLD LASER EYE!
===============================
===============================
It's Gyoubu Masataka Oniwa, the ManRay! It has the power level of 553
..... HEED ME.....
FOR MY NAAAAAAAME IS GYOUBU MASATAKA ONIWA
TIME TO SHOW MY POWERS
SUPERIOR SIGHT, BEHOLD LASER EYE!
WEAKNESS DISGUST ME, BEHOLD SUPER STRENGTH!
===============================
===============================
It's Arnastria, the DirtyBubble! It has the power level of 666
..... HEED ME.....
FOR MY NAAAAAAAME IS ARNASTRIA
TIME TO SHOW MY POWERS
WEAKNESS DISGUST ME, BEHOLD SUPER STRENGTH!
EAT DIRT MORTAL, BEHOLD THE POWER OF FLIGHT!
===============================
===============================
It's Tatenari, the FlyingDutchMan! It has the power level of 36556
..... HEED ME.....
FOR MY NAAAAAAAME IS TATENARI
TIME TO SHOW MY POWERS
EAT DIRT MORTAL, BEHOLD THE POWER OF FLIGHT!
SUPERIOR SIGHT, BEHOLD LASER EYE!
===============================
BUILD SUCCESSFUL (total time: 1 second)
```

- Permasalahan yang dihadapi : Dalam penggunaan Comparable pada class SuperHero.java
- Solusi Permasalahan
  - Menggunakan method compareTo() pada class SuperHero.java, kemudian ada perbandingan antara 2 objek yang ada pada class SuperHero dan menggunakan import java.lang.Comparable :
    - Bilangan 0 : Jika dua objek yang di bandingkan sama
    - Bilangan positif 1 : artinya object yang dibandingkan lebih besar dari object parameter
    - Bilangan negatif -1 : Artinya object yang dibandingkan lebih kecil dari object parameter
- Nama Teman yang membantu : Gina Anifah