LAB 7

Exercise 1: Create a GeometricObject Class

- Create a GeometricObject class that contains:
  - attributes of color (String), and filled (boolean)
  - 2 constructors:
    - GeometricObject() which set the default to “blue”
    - GeometricObject(String color, boolean filled) which set the color and filled attributes according to the input
  - 5 method
    - void setColor(String color)
    - void setFilled(boolean filled)
    - String getColor()
    - boolean isFilled()
    - void printInfo() //print the color of the object and print if it is filled or not filled

The GeometricObject class (completed)

```java
public class GeometricObject {
  private String color;
  private boolean filled;

  public GeometricObject() {
	  this.color = "blue";
  }
  public 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;
  }
  public void printInfo(){
	  if(filled){
		  System.out.println("The Geometric is "+color+" and it is "+"filled.");
	  }
	  else{
		  System.out.println("The Geometric is "+color+" but it is not "+"filled.");
	  }
  }
}
```

Exercise 2: Create a Circle Class

- Create a Circle class that extends the GeometricObject and contains:
  - attributes of radius (double), and a constant pi (double) from Math class (Math.PI)
  - 3 constructors
    - Circle() \\ call the second constructor with radius 1.0
    - Circle(double radius) \\ call the third constructor with the input radius, color = “white”, and filled = true
    - Circle(double radius, String color, boolean filled) \\ set radius and call a constructor of GeometricObject with the input of color and filled

- The Circle class

```
public class Circle extends GeometricObject {
  private double radius;
  private final double PI = Math.PI;

  public Circle() {
        this(1.0);
  }

  public Circle(double radius) {
        this(radius, "white", true);
  }

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

- With an `extends` keyword, a class can inherit attributes, and methods of its super class
- In this case,
  - the Circle class is a sub class of the GeometricObject class
  - the GeometricObject is a super class of the Circle class

Exercise 2: Create a Circle Class

- Add the following methods in the Circle class
  - void setRadius(double radius)
  - double getRadius() // return radius
  - double getArea() // calculate area of a circle and return the area
  - double getPerimeter() // calculate area of a circle and return the perimeter

- Override the printInfo() method of the GeometricObject class by
  - call the printInfo of the GeometricObject by using keyword `super`
  - add another print statement to print “It is a circle with radius of ” + radius

- The Circle class

```
public class Circle extends GeometricObject {
  private double radius;
  private final double PI = Math.PI;

  …

  public double getRadius() {
    return radius;
  }
  public void setRadius(double radius) {
    this.radius = radius;
  }
  public double getArea() {
    return radius*radius*Math.PI;
  }
  public double getPerimeter() {
    return 2*radius*Math.PI;
  }
  public void printInfo() {
        super.printInfo();
        System.out.println( "It is a circle with radius of " + radius);
  }
}
```

Exercise 3: Create a Rectangle Class

- Create a Rectangle class that extends the GeometricObject and contains:
  - attributes of width (double) and height (double)
  - 3 constructors
    - Rectangle() \\ call the second constructor with width = 1.0 and height = 1.0
    - Rectangle(double width, double height) \\ call the third constructor with the input radius, color = “green”, and filled = true
    - Rectangle(double width, double height, String color, boolean filled) \\ set width, height and call a constructor of GeometricObject with the input of color and filled

- The Rectangle class

```
public class Rectangle extends GeometricObject {
    private double width;
    private double height;

    public Rectangle() {
        this(1.0, 1.0);
    }

    public Rectangle(double width, double height) {
        this(width, height,"green", true);
    }

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

Exercise 3: Create a Rectangle Class

- Add the following methods in the Rectangle class
  - void setWidthHeight(double width, double height)
  - double getWidth() // return width
  - double getHeight() // return height
  - double getArea() // calculate area of a rectangle and return the area
  - double getPerimeter() // calculate area of a rectangle and return the perimeter

- Override the printInfo() method of the GeometricObject class by
  - call the printInfo of the GeometricObject by using keyword `super`
  - add another print statement to print “It is a rectangle with width of ” + width + “ and height of ” + height

- The Rectangle class

```
public class Rectangle extends GeometricObject {
    …
        public double getWidth() {
                return width;
        }
        public double getHeight() {
                return height;
        }
        public void setWidthHeight(double width, double height) {
                this.width = width;
                this.height = height;
        }
        public double getArea() {
                return width*height;
        }
        public double getPerimeter() {
                return (2*width)+(2*height);
        }
        public void printInfo() {
                super.printInfo();
                System.out.println( "It is a rectangle with width of " + width
+ " and height of " + height);
        }
}
```

Exercise 4: Inheritance & Polymorphism

- Create an InheritanceTester class
- Add a main() method
- Add a printGeometricObjectInfo() method
  - void printGeometricObjectInfo(GeometricObject g)

- The InheritanceTester class

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

        }

        public void printGeometricObjectInfo(GeometricObject g){
                g.printInfo();
        }
}
```

- Create an object of GeometricObject, Circle, and Rectangle and name them g1, c1, r1 accordingly

- For each object call a method printInfo()

- The InheritanceTester class

```
public class InheritanceTester {
        public static void main(String[] args){
                GeometricObject g1 = new GeometricObject();
                g1.printInfo();

                Circle c1 = new Circle();
                c1.printInfo();

                Rectangle r1 = new Rectangle();
                r1.printInfo();
        }

        public void printGeometricObjectInfo(GeometricObject g){
                g.printInfo();
        }
}
```

- Try calling other methods from super class and sub class and then observe

- Delete statements of the printInfo() method
- Then call the method printGeometricObjectInfo(GeometricObject g)

```
public class InheritanceTester {
        public static void main(String[] args){
                InheritanceTester tester = new InheritanceTester();

                GeometricObject g1 = new GeometricObject();
                tester.printGeometricObjectInfo(g1);

                Circle c1 = new Circle();
                tester.printGeometricObjectInfo(c1);

                Rectangle r1 = new Rectangle();
                tester.printGeometricObjectInfo(r1);
        }

        public void printGeometricObjectInfo(GeometricObject g){
                g.printInfo();
        }
}
```

Exercise 4: Inheritance & Polymorphism

- Observe the result of the InheritanceTester

- What does it print? and why?

- Why does the method printGeometricObjectInfo(GeometricObject g) work with the input type of Circle and Rectangle?
  - Polymorphism allows objects of different classes related by inheritance to respond differently to the same method