COMP2396B Object-oriented Programming and Java

Tutorial 3 – State and Behaviour of Objects (1)

Outline:

1. Concept Overview
2. Exercises on Class and Object

Part 1: Concept Overview

In this tutorial, we will do some practice to revise the concepts in the following lecture slides:

Topic	Readings
Basic Concepts in Classes and Objects	**Course Materials:** Lecture 2 – P.15-35
Essential Elements in Class Declaration (e.g. parameters, return value, getters, setters, constructors)	**Course Materials:** Lecture 4 – P.8-19
State and Behaviour of Objects	**Course Materials:** Lecture 4 – P.1-7

Part 2: Exercises on Class and Object

1. Study **Tester.java** and the sample test case below. Create the **Circle** class to complete the program.

Tester.java

```
import java.io.*;

public class Tester {
    public static void main(String[] args) throws Exception {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader inData = new BufferedReader(isr);

        int c2_diameter = Integer.parseInt(inData.readLine());
        String c2_color = inData.readLine();
        int c3_diameter = Integer.parseInt(inData.readLine());
        String c3_color = inData.readLine();

        System.out.println("Creating circles...");
        Circle c1 = new Circle();
        Circle c2 = new Circle(c2_diameter);
        Circle c3 = new Circle(c3_diameter, c3_color, true);

        System.out.println("Getting the diameter of the circles...");
        System.out.println("C1: " + c1.getDiameter());
        System.out.println("C2: " + c2.getDiameter());
        System.out.println("C3: " + c3.getDiameter());

        System.out.println("Setting a new diameter of C2...");
        c2.setDiameter(16);
        System.out.println("The new perimeter of C2...");
        System.out.println("C2: " + c2.getPerimeter());

        System.out.println("Setting new color of c2...");
        c2.setColor(c2_color);
        c2.setFilled(true);

        System.out.println("Getting the information of c2...");
        System.out.println("C2 Color: " + c2.getColor() + " Filled: " + c2.getFilled());
        System.out.println("Getting the information of c3...");
        System.out.println("C3 Color: " + c3.getColor() + " Filled: " + c3.getFilled());
    }
}
```

Remarks:

- The diameter of the circle is in decimal format. It should have a default value of 1.
- The color of a circle is in string format. It should have a default value of “red”.
- A Circle object should be filled (with color) by default.
- You cannot use System.out.println() or other print functions in the **Circle** class.

Sample Test Cases:

Test case 1:
Input:
3
green
4
grey

Output:
Creating circles...
Getting the diameter of the circles...
C1: 1.0
C2: 3.0
C3: 4.0
Setting a new diameter of C2...
The new perimeter of C2...
C2: 50.26548245743669
Setting new color of c2...
Getting the information of c2...
C2 Color: green Filled: true
Getting the information of c3...
C3 Color: grey Filled: true

Hints:
3 constructors should be implemented:

- The no-argument constructor should construct a circle object with the default values of diameter, color and filled.
- The constructor with the argument diameter should construct a circle object with the specified diameter.
- The constructor with the argument diameter, color and filled should create a circle object with the specified diameter, color and filled values.

7 methods should be implemented:

- The method setDiameter should set a new diameter for this circle.
- The method getDiameter should return the diameter of this circle.
- The method getPerimeter should return the perimeter of this circle. Note that you should use the pi value declared in Math.PI library when calculating the perimeter of the circle.
- The method setColor should set a new color of this circle.
- The method getColor should return the color of this circle.
- The method getFilled should return the filled value (either true or false) of this circle.
- The method setFilled should set a new filled value (either true or false) of this circle.

2. A complex number is a number that can be expressed in the form a + bi, where a and bi are denoted as real part and imaginary part, respectively.

   The addition of two complex numbers can be obtained by summing their real parts and imaginary parts, respectively. Similarly, for subtracting, we consider the real and imaginary parts separately and subtract the real and imaginary parts.

   Study **Tester.java** and the the sample test case below. Create the **Complex** class to complete the program.

Tester.java

```
import java.io.*;

public class Tester {
    public static void main(String[] args) throws Exception {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader inData = new BufferedReader(isr);

        int c1_real = Integer.parseInt(inData.readLine());
        int c1_imaginary = Integer.parseInt(inData.readLine());
        int c2_real = Integer.parseInt(inData.readLine());
        int c2_imaginary = Integer.parseInt(inData.readLine());

        Complex c1 = new Complex(c1_real, c1_imaginary);
        Complex c2 = new Complex(c2_real, c2_imaginary);

        c1.add(c2);
        System.out.println(c1.toString());
        c1.subtract(c2);
        c1.subtract(c2);
        System.out.println(c1.toString());
    }
}
```

Remarks:

- You cannot use System.out.println() or other print functions in the Complex class.
- You may define the toString() method to return a String for any self-defined class.

Sample Test Cases:

Test case 1:
Input:
1
2
3
4

Output:
4 + 6i
-2 - 2i

Sample Solution:

1. Circle.java

```
public class Circle {

    private double diameter = 1;
    private String color = "red";
    private boolean filled = true;

    public Circle() {

    }

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

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

    public void setDiameter(double diameter) {
        this.diameter = diameter;
    }

    public double getDiameter() {
        return diameter;
    }

    public double getPerimeter() {
        final double pi = Math.PI;
        return diameter * pi;
    }

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

    public String getColor() {
        return color;
    }

    public boolean getFilled() {
        return filled;
    }

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

}
```

2. Complex.java

```java
public class Complex {

    private int real = 0;
    private int img = 0;

    public Complex(int real, int img) {
        this.real = real;
        this.img = img;
    }

    public void add (Complex c2) {
        this.real = this.real + c2.real;
        this.img = this.img + c2.img;
    }

    public void subtract (Complex c2) {
        this.real = this.real - c2.real;
        this.img = this.img - c2.img;
    }

    public String toString() {
        if (this.img >= 0) {
            return (this.real + " + " + this.img + "i");
        }
        else {
            return (this.real + " - " + Math.abs(this.img) + "i");
        }
    }

}
```