CS102A Introduction to Computer Programming
Fall 2020 Lab 7

Credit

The source code and document description are designed by ZHU Yueming.

Objective

1. Learn how to define a Java class.
2. Learn how to use instance variables.
3. Learn how to define and use instance methods.
4. Learn how to use get and set methods.
5. Learn how to use `ArrayList`.

1 Before Exercise

1.1 Step 1: How to define a circle on a 2D plane?

A circle has three attributes including the radius, the x coordinate and the y coordinate. We can define a class named `Circle`, in which there are three variables:

```
public class Circle {
    private double radius;
    private double x;
    private double y;
}
```

1.2 Step 2: Define the methods for printing the information of a circle.

Define three methods for computing the area, position and perimeter of the circle.

```
public class Circle {
    private double radius;
    private double x;
    private double y;

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

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

    public void position() {
        System.out.printf("Position of the circle is (%.1f,%.1f)\n",x,y);
    }
}
```

1.3 Step 3: How to use the class `Circle`?

Create another class named `CircleTest` in the same package, in which there is a `main` method to be used. In the `main` method, we can create an object of `Circle` by using the statement as follows:

```
Circle c1 = new Circle();
```

After that, we want to know the perimeter, area and position about the `c1`, so we need to invoke the method of `c1`:

```
public class CircleTest {

    public static void main(String[] args) {
        Circle c1 = new Circle();
        System.out.printf("The area of c1 is %.2f\n", c1.area());
        System.out.printf("The perimeter of c1 is %.2f\n", c1.perimeter())
            ;
        c1.position();
```

```
    }
}
```

When we run the program, the result would as follows:

```
The area of c1 is 0.00
The perimeter of c1 is 0.00
Position of the circle is (0.0,0.0)
```

1.4 Step 4: Set and get the values of the attributes

If we set or get the radius of a circle object in `main` method directly, it would lead to an error because of its **private** privilege. In addition, the radius of a circle should not contain a negative number, how can we set the restriction?

```
public static void main(String[] args) {
    Circle c1 = new Circle();
    System.out.printf("The area of c1 is %.2f\n", c1.area());
    System.out.printf("The perimeter of c1 is %.2f\n", c1.perimeter());
    c1.position();
    c1.radius = -1;
    System.out.println(c1.radius);
}
```

We can define several public methods in class `Circle` for getting or setting the class variables, and we can check the validity of input value in the set method.

```
public class Circle {
    private double radius;
    private double x;
    private double y;

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

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

```

```
    public void position() {
        System.out.printf("Position of the circle is (%.1f,%.1f)\n", x, y)
            ;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        if (radius > 0) {
            this.radius = radius;
        }
    }

    public double getX() {
        return x;
    }

    public void setX(double x) {
        this.x = x;
    }

    public double getY() {
        return y;
    }

    public void setY(double y) {
        this.y = y;
    }
}
```

After that, we can access the attributes by the get and set methods.

```
public static void main(String[] args) {
    Circle c1 = new Circle();

    c1.setRadius(5);
    System.out.println(c1.getRadius());
```

```
6
7       System.out.printf("The area of c1 is %.2f\n", c1.area());
8       System.out.printf("The perimeter of c1 is %.2f\n", c1.perimeter());
9       c1.position();
10  }
```

Sample input and output:

```
5.0
The area of c1 is 78.54
The perimeter of c1 is 31.42
Position of the circle is (0.0,0.0)
```

1.5 Step 5: How to manage multiple circle objects?

We can use an array or an `ArrayList` to manage them. In the `main` method, create an arrayList with a `Circle` type, to store many objects of `Circle`. Add the following code at the end of `main` method.

```
1  ArrayList<Circle> circleList = new ArrayList<Circle>();
2  circleList.add(c1);
3  System.out.printf("Radius of %d circle is %.2f: \n", 1, circleList.get(0).
       getRadius());
```

Sample input and output:

```
5.0
The area of c1 is 78.54
The perimeter of c1 is 31.42
Position of the circle is (0.0,0.0)
Radius of 1 circle is 5.00:
```

1.6 Step 6: Add more circles in the `ArrayList`.

Add the following code at the end of main method.

```
1      for(int i = 1; i < 5; i++) {
2          circleList.add(new Circle());
3          circleList.get(i).setRadius(i);
4          circleList.get(i).setX(Math.random() * 5);
5          circleList.get(i).setY(Math.random() * 5);
```

```
    }

    System.out.println("---Begin to print the circle list---");
    for(int i = 0; i < 5; i++) {
        System.out.printf("The area of %d circle is %.2f\n", i+1,
           circleList.get(i).area());
        System.out.printf("The perimeter is %.2f\n", circleList.get(i).
           perimeter());
    }
```

Sample input and output:

```
5.0
The area of c1 is 78.54
The perimeter of c1 is 31.42
Position of the circle is (0.0,0.0)
Radius of 1 circle is 5.00:
---Begin to print the circle list---
The area of 1 circle is 78.54
The perimeter is 31.42
The area of 2 circle is 3.14
The perimeter is 6.28
The area of 3 circle is 12.57
The perimeter is 12.57
The area of 4 circle is 28.27
The perimeter is 18.85
The area of 5 circle is 50.27
The perimeter is 25.13
```

2 Exercise

2.1 Exercise 1

Declare a class named `User`. The class contains:

- Private data fields:
  - `String name`;
  - `String password`;

  - `double money`;

- Implement a public method named `introduce()` to print the user name and his account balance.
- Implement a public method `expense(double value)`. It withdraws the money from the user account.
- Implement a public method `income(double value)`. It deposits the money to the user account.
- Implement the `getter` and `setter` methods for each private field of the class User.

In the same package, we create a class named `ClientTest`, which has a `main` method. Statements in `main` method:

```
public static void main(String[] args) {
    User user = new User();
    user.setName("Lucy");
    user.setPassword("123456");
    user.setMoney(1000);
    user.introduce();
    user.expense(2000);
    user.expense(500);
    user.income(1000);
    user.introduce();
}
```

Sample input and output:

```
My name is Lucy and I have 1000.00 dollar
no sufficient funds
You have expense 500.00 dollar and the remained amount is 500.00
The remained amount is 1500.00
My name is Lucy and I have 1500.00 dollar
```

2.2 Exercise 2

Design a class named Food. The class contains:

- Private data fields:
  - `String name`;
  - `String type`;
  - `int size`;

Object Name	name	type	size	price
pizza1	pizza	Seafood	11	120
pizza2	pizza	Beef	9	100
FriedRice	fried rice	Seafood	5	40
Noodles	noodles	Beef	6	35

  - `double price`;

- Implement a public method named `showInformation()` to print all the information of this food object.
- Implement the getter and setter method for each private field of `Food`.

In `ClientTest` class, create four objects of `Food` as follows:

Create an `ArrayList<Food>` to add those four `Food` objects, and then show the information of them as follows by iterating the `ArrayList<Food>` we created.

Sample input and output:

```
Seafood pizza: (11 Inches) 120.00 $
Beef pizza: (9 Inches) 100.00 $
Seafood fried rice: (5 Inches) 40.00 $
Beef noodle: (6 Inches) 35.00 $
```