Basis of Computer Programming (java A)
Tutorial 8

[Experimental Objective]

- Learn how to define constructor of Java class
- Lean what is the "this" key word
- Learn constructor overload
- Learn how to use toString()
- Learn how to use split
- Review how to use ArrayList<T>

[Before Exercises]

Step1: Try to define a constructor

Last time we build a class named circle, in which there are three attributes, radius, x and y. This time we define a constructor, which has three double arguments correspond the three attributes.

You should write the following code at first:

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

	Circle (double radius, double x, double y){
	    radius = radius;
	    x      = x;
	    y      = y;
	}
}
```

Step2: You may find some warning, what can we do if we want to assign the value of an argument to an attribute with the same name.

We need to use *this*.

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

	Circle (double radius, double x, double y){
	    this.radius = radius;
	    this.x      = x;
	    this.y      = y;
	}
}
```

Step3: Thinking about how can we initial the circle in different way?

Sometimes we need to initial a class in different way, we should use constructor overload.

```
public class Circle {
```

```
        private double radius;
        private double x;
        private double y;
        static final int DEFAULT_RADIUS = 5;

        Circle(double radius, double x, double y) {
               this.radius = radius;
               this.x = x;
               this.y = y;
        }

        Circle(double radius) {
               this.radius = radius;
               this.x = 0;
               this.y = 0;
        }

        Circle(double x, double y) {
               this.radius = DEFAULT_RADIUS;
               this.x = x;
               this.y = y;
        }

}
```

Step4: Thinking about how can we get all the values of the attributes?

We can generate a lot of Circle objects. But how can we get the statuses of all objects?
Can we print the objects out using the following code?

```
public class Circle_Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ArrayList<Circle> circleList = new ArrayList<Circle>();
        int n = 5;
        for (int i = 0; i < n; i++) {
            circleList.add(new Circle(i, i, i));
            circleList.add(new Circle(i));
            circleList.add(new Circle(i, i));
        }
        for (int i = 0; i < circleList.size(); i++) {
            System.out.print(circleList.get(i));
        }
    }
}
```

No, if we run above code, we will find that it just prints out the following information:

```
<terminated> Circle_Test [J
Circle@7852e922
Circle@4e25154f
Circle@70dea4e
Circle@5c647e05
Circle@33909752
Circle@55f96302
Circle@3d4eac69
Circle@42a57993
Circle@75b84c92
Circle@6bc7c054
Circle@232204a1
Circle@4aa298b7
Circle@7d4991ad
Circle@28d93b30
Circle@1b6d3586
```

Maybe someone remember getter methods which we learned last week. You are excellent!
We can use the following code:

```
for (int i = 0; i < circleList.size(); i++) {
    System.out.printf("The postion of the circle is (%.2f, %.2f), the radius is %.2f\n",
            circleList.get(i).getX(), circleList.get(i).getY(), circleList.get(i).getRadius());
}
```

Yes, this is what we want:

```
The postion of the circle is (0.00, 0.00), the radius is 0.00
The postion of the circle is (0.00, 0.00), the radius is 0.00
The postion of the circle is (0.00, 0.00), the radius is 5.00
The postion of the circle is (1.00, 1.00), the radius is 1.00
The postion of the circle is (0.00, 0.00), the radius is 1.00
The postion of the circle is (1.00, 1.00), the radius is 5.00
The postion of the circle is (2.00, 2.00), the radius is 2.00
The postion of the circle is (0.00, 0.00), the radius is 2.00
The postion of the circle is (2.00, 2.00), the radius is 5.00
The postion of the circle is (3.00, 3.00), the radius is 3.00
The postion of the circle is (0.00, 0.00), the radius is 3.00
The postion of the circle is (3.00, 3.00), the radius is 5.00
The postion of the circle is (4.00, 4.00), the radius is 4.00
The postion of the circle is (0.00, 0.00), the radius is 4.00
The postion of the circle is (4.00, 4.00), the radius is 5.00
```

But this way is a little boring, just image if we have hundreds of attributes.

Let's introduce another way to print out an object's status.
Every object has a method named *toString()* , but the default one only return the reference of the object, for example, Circle@78521922. If we want it to return a string what we want, we should override it.

We can add the following code to the class Circle.

```
@Override
public String toString(){
    return String.format("The postion of the circle is (%.2f, %.2f), the radius is %.2f\n",
            x, y, radius);
}
```

Then we use the following code again in the class Circle_Test:

```
System.out.print(circleList.get(i));
```

Now, the result is also what we want:

```
The postion of the circle is (0.00, 0.00), the radius is 0.00
The postion of the circle is (0.00, 0.00), the radius is 0.00
The postion of the circle is (0.00, 0.00), the radius is 5.00
The postion of the circle is (1.00, 1.00), the radius is 1.00
The postion of the circle is (0.00, 0.00), the radius is 1.00
The postion of the circle is (1.00, 1.00), the radius is 5.00
The postion of the circle is (2.00, 2.00), the radius is 2.00
The postion of the circle is (0.00, 0.00), the radius is 2.00
The postion of the circle is (2.00, 2.00), the radius is 5.00
The postion of the circle is (3.00, 3.00), the radius is 3.00
The postion of the circle is (0.00, 0.00), the radius is 3.00
The postion of the circle is (3.00, 3.00), the radius is 5.00
The postion of the circle is (4.00, 4.00), the radius is 4.00
The postion of the circle is (0.00, 0.00), the radius is 4.00
The postion of the circle is (4.00, 4.00), the radius is 5.00
```

Step5: How to create objects according different requirements?

Usually we will need to create objects according different requirements. These attributes usually store in database or a file or other formats. How to do? Here we just learn how to create objects from a file.

Firstly, we can store every attribute's value in a row separated by a space character (or other character, for example, ',' or '_') in a file;

Secondly, scan the file and repeat to read a line, we can parse the values from a line string use following code:

```
String circleInfo = scan.nextLine();
String[] values = circleInfo.split(" ");
double radius = Double.parseDouble(values[0]);
double x = Double.parseDouble(values[1]);
double y = Double.parseDouble(values[2]);
```

Finally, create an object according the values we parsed.

The whole process is like this:

```
Scanner scan = new Scanner(new File("circle_info.txt"));
ArrayList<Circle> circleList = new ArrayList<Circle>();
while (scan.hasNextLine()) {
    String circleInfo = scan.nextLine();
    String[] values = circleInfo.split(" ");
    double radius = Double.parseDouble(values[0]);
    double x = Double.parseDouble(values[1]);
    double y = Double.parseDouble(values[2]);

    circleList.add(new Circle(radius, x, y));
}
```

For example, we can store the following circle information in a file named "circle_info.txt".

```
1.2 3.0 4.7
2.4 6.0 7.8
9.6 9.0 9.2
6.7 7.0 8.3
4.8 5.0 6.4
```

Here is a sample run:

```
The postion of the circle is (3.00, 4.70), the radius is 1.20
The postion of the circle is (6.00, 7.80), the radius is 2.40
The postion of the circle is (9.00, 9.20), the radius is 9.60
The postion of the circle is (7.00, 8.30), the radius is 6.70
The postion of the circle is (5.00, 6.40), the radius is 4.80
```

[Exercises]

1. Modify the class **User** which you designed last week:

a. Design a constructor which argument list contains **name** (String), **password** (String), **money** (double).
b. Design a constructor which argument list contains **name** (String). In this constructor, **password** would set to a default value:"123456" and **money** would set to a default value 1000.0.
c. Override **toString()**, which return a string follow this format: **name** has **money** dollars.

We should modify the class **ClientTest** in which we scan a file containing the following information:

```
ZhangSan
LiSi
WangWu
WuLiu
HongQi 55555 0
ChenShiyi 666 1000000
```

We should create objects according above file, then print these objects.
Here is a sample run:

```
ZhangSan have 1000.00 dollars.
LiSi have 1000.00 dollars.
WangWu have 1000.00 dollars.
WuLiu have 1000.00 dollars.
HongQi have 0.00 dollars.
ChenShiyi have 1000000.00 dollars.
```

2. Modify the class **Food** which you designed last week:
   a. Design a constructor which argument list contains **name** (String), **type** (String), **size** (int), **price** (double).
   b. Design a constructor which argument list contains **name** (String), **type** (String), **size** (int). In this constructor, **price** would set to a default value according **type**. If **type** is "seafood", **price** would set to 40. If **type** is "beaf", **price** would set to 35.
   c. Design a constructor which argument list contains **name** (String), **size** (int). In this constructor, **price** would set to a default type: "seafood". **price** would set to a default value according **type**. Because the type set to "seafood", so the **price** set to 35.
   d. Override **toString()**, which return a string follow this format: **type name** : (**size** inch) **price**.

We should modify the class **ClientTest** in which we scan a file containing the following information:

```
pizza seafood 10 120
pizza beef 8 100
noodle 5
noodle beef 6
```

We should create objects according above file, then print these objects.

Here is a sample run:

```
seafood pizza: (10 Inches) 120.00 $
beef pizza: (8 Inches) 100.00 $
seafood noodle: (5 Inches) 40.00 $
beef noodle: (6 Inches) 35.00 $
```