厦門大學

信息学院软件工程系

《JAVA 程序设计》实验报告

实验六

姓名：庾晓萍

学号：20420192201952

学院：信息学院

专业：软件工程

完成时间：2022/4/1

一、实验目的及要求

（一）实验目的

1、熟悉继承

2、第一次上机考模拟

（二）实验要求

1、按照题目要求写代码和实验报告，并上传到 FTP

二、实验题目及实现过程

一、基本题目：

题目 1：用多态改写实验 5 第二题

（一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

（二） 实现过程

（1）设计类

① DrawRandomShapes 类中，继承了 Application 抽象类并重写了 start()方法，在该方法中，Stage 就是 JavaFX 工具中用来表示整个图形工具界面窗口的类，

在该类中需要加入一个 Scene（场景）来进行填充，而所有的组件、元素都是构建在 Scene 中的。另外，在 JavaFX 8 中支持代码与布局和样式分离，所以在文件中通过 FXMLLoader 的 load()方法引入了一个外联的 DrawRandomShapes.fxml 文件，在此 fxml 文件中就可以专心编写图形界面布局和组件相关功能。

② DrawRandomShapes.fxml 文件中设置了画布的长度宽度为 300，控制文件为 DrawRandomShapesController.java。同时设置了四个 TextField，对应 id 是 x1，x2，y1，y2。

③ 在 DrawRandomShapesController 类中，将随机产生一个随机数（0,1,2），三个随机数分别对应直线、矩形和椭圆三种图形。根据随机数对应图形，提示用户输入图形所需初始化参数（x1，x2，y1，y2），提示信息包括参数的范围（这里的范围是画布的范围，也就是 0-300），用户输入后进行范围检查，若合法，则根据用户输入的信息在界面上绘制出相应的图形。

（2）多态方法

① 设计一个抽象类 MyShape，其中有私有成员 x1、x2、x3、x4 代表绘制图形的坐标数据，strokeColor 代表图像的填充颜色。初次之外还有无参构造函数、有参构造函数、各种 Setters 方法和各种 Getters 方法用于修改或获取类中的私有成员。同时有一个用于绘图的抽象方法 draw，只有声明没有实现。

```
//绘图，抽象方法，只有声明，没有实现
public abstract void draw(GraphicsContext gc);
```

② 设计抽象类 MyShape 的三个子类 MyLine、MyRectangle、MyOval 分别用于绘制直线、三角形、椭圆。

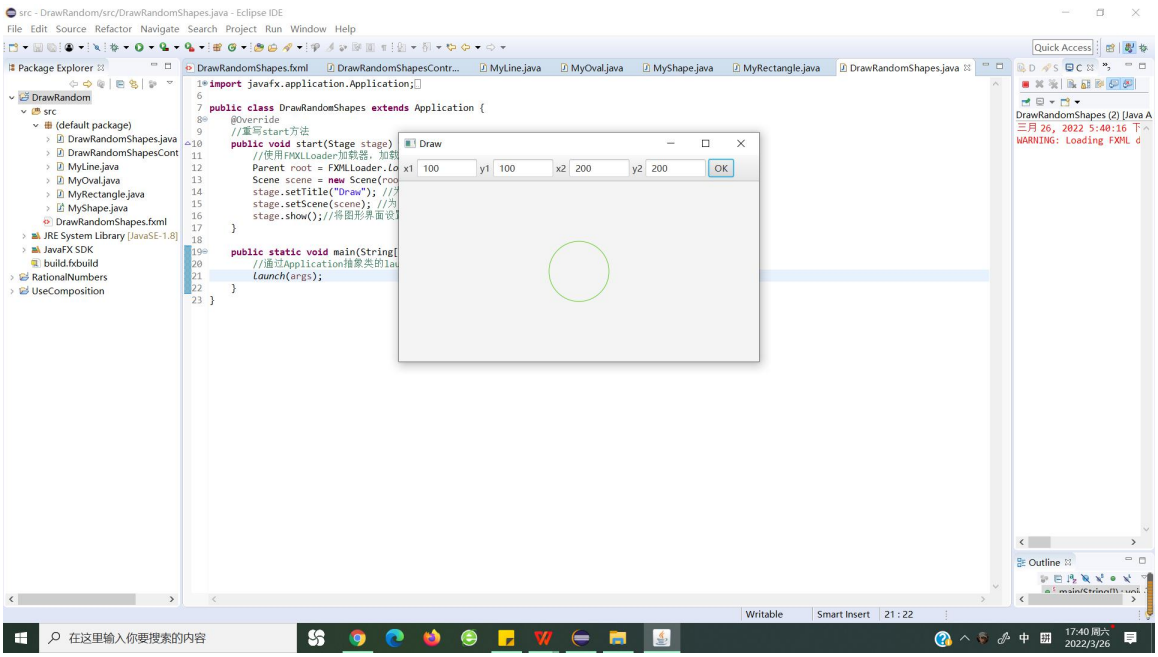
③ 在 DrawRandomShapesController 类中，temp1、temp2、temp3 都是 MyShape 类型，调用同一个 draw 方法，但是因为父类引用指向的是不同的子类对象

（MyLine、MyRectangle、MyOval 类型），所以最后绘制出的对象可以呈现不同的状态。

```
switch (shapeNumber) {
    case 0://0代表绘制线
        MyShape temp1=new MyLine(x1, y1, x2, y2, strokeColor);
        //动态绑定，调用子类MyLine中的draw方法
        temp1.draw(gc);
        break;
    case 1://1代表绘制三角形
        MyShape temp2=new MyRectangle(x1, y1, x2, y2, strokeColor);
        //动态绑定，调用子类MyRectangle中的draw方法
        temp2.draw(gc);
        break;
    case 2://2代表绘制椭圆
        MyShape temp3=new MyOval(x1, y1, x2, y2, strokeColor);
        //动态绑定，调用子类MyOval中的draw方法
        temp3.draw(gc);
        break;
    }
```

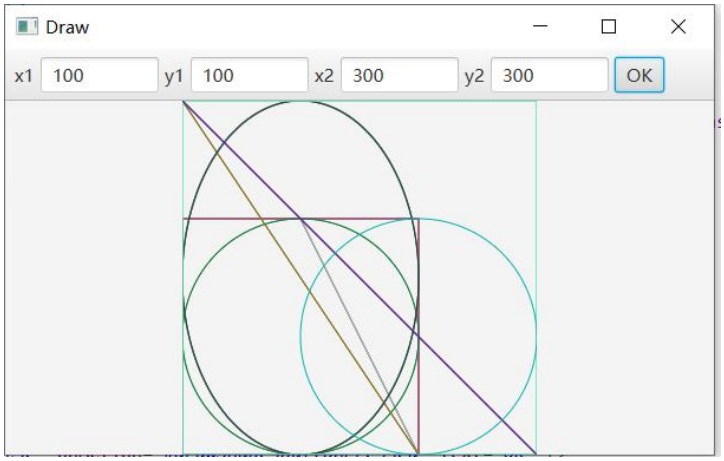
（三） 过程截图

（1）全屏截图

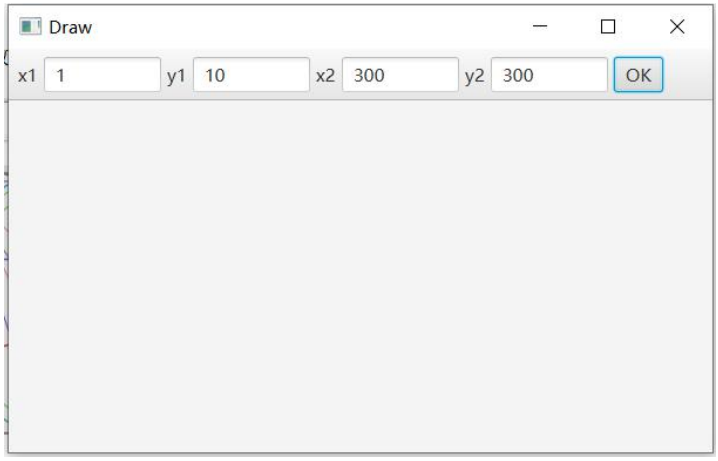


（2）运行结果

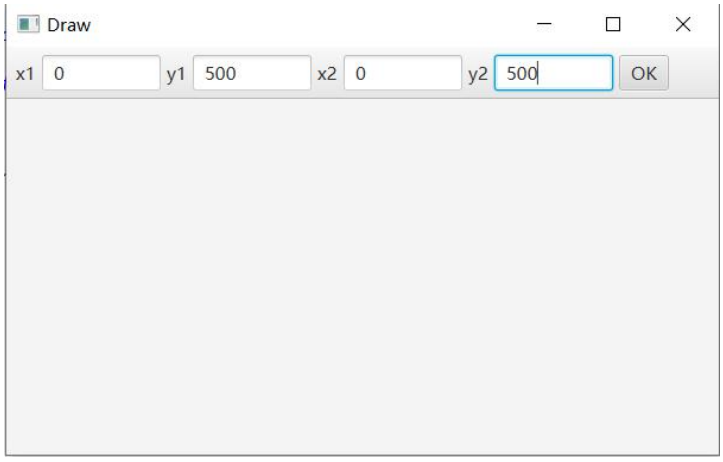
1. 根据用户的输入数据，随机绘制直线、椭圆和长方形。



2. 当绘制超过 20 个图形时，画布清空。



3. 当输入的数据不在画布范围内时，提示输入数据的范围。



题目 2：在 90 分钟内完成《第一次上机考试（模拟）.pdf》中的题目

（一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

（二） 实现过程

（1）设计类

①设计一个抽象类 Vehicle，代表载具，设计抽象类 Vehicle 的两个子类 Car、Truck 分别代表小汽车类、卡车类。同时抽象类 Vehicle 有一个用于绘图的抽象方法 draw，只有声明没有实现，两个子类 Car、Truck 对父类的这个抽象方法进行重写。

② 设计了一个 Fleet 类代表车辆队，具有添加车辆、查询车辆、打印所有车辆的功能

（2）多态（动态绑定）

① 以创建 Car 对象为例。

```
//获得Class对象、获得构造、通过构造对象获得实例化对象
Car addCar=(Car) Class.forName("Car").getConstructor(String[].class).newInstance((Object)inputArray);
carList.add(addCar);
System.out.println("创建成功");
```

上面的代码获得 Car 对象、获得构造、通过构造对象获得实例化对象。在 new 这个 Car 对象时，调用了 Car 对象的构造方法，构造 Car 子类时，首先使用父类对象 Vehicle 的引用 super 调用父类的构造方法。把传过来的字符串 inputs 传递给父类对象。当 Car 对象的构造方法调用结束后，在堆内存里面 new 出了一个 Car 对象，除了拥有从 Vehicle 类中继承下来的 tradeMark、color、factoryYear、carType、capacity 属性外，还拥有自己的一个私有属性 carriagesNumber。

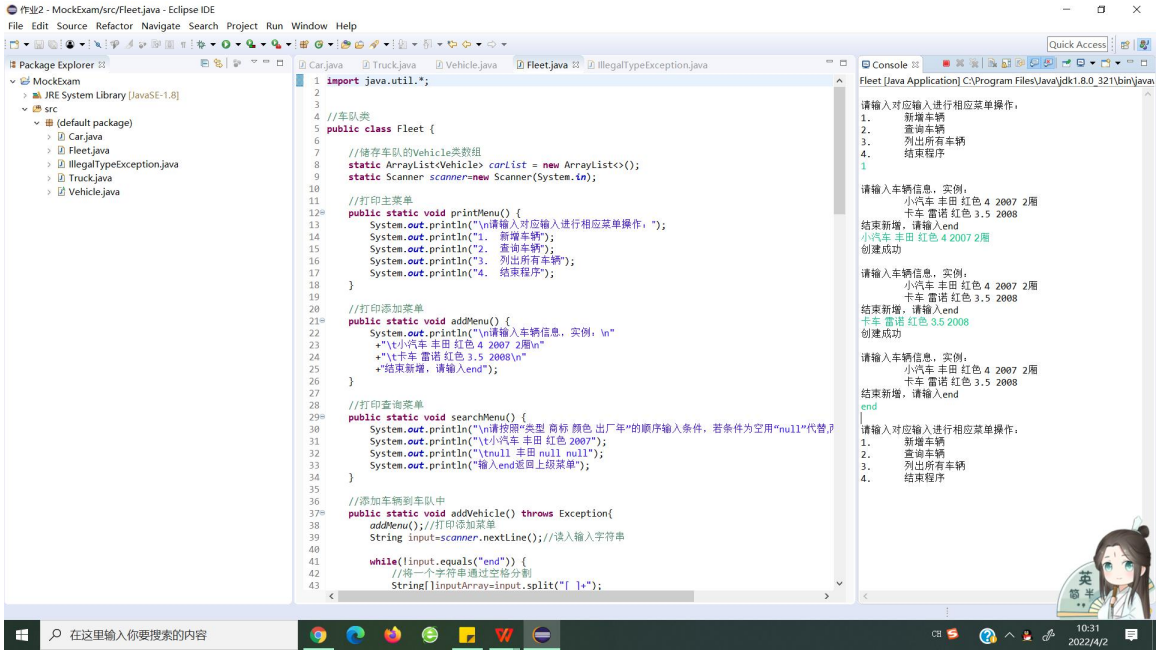
```
//构造函数
public Car(String[] inputs) {
    super(inputs);
    this.carriagesNumber = inputs[5];
}
```

② 动态绑定指的是实际中的对象是什么类型，就调用谁的方法。以下面的 printVehicle 方法的调用为例，虽然引用是 Vehicle 类型，但如果实际中的对象是 Car，那么调用的是 Car 子类里面重写后的 printVehicle 方法。这是因为在父类对象 Vehicle 的内部有一个 printVehicle 方法的指针，指针指向代码区里面父类的 Vehicle 的 printVehicle 方法，只不过当 new 对象的时候，这个指针随之改变，new 的是 Car 对象，这个指针就指向这个 Car 对象重写后的那个 printVehicle 方法。动态绑定的意思是，只有在程序运行期间，new 出了这个对象了后才确定到底要调用哪一个方法。我实际当中的地址才会绑定到相应的方法的地址上面，所以叫动态绑定。

```
for(Vehicle vehicles : output)
{
    vehicles.printVehicle();
}
```

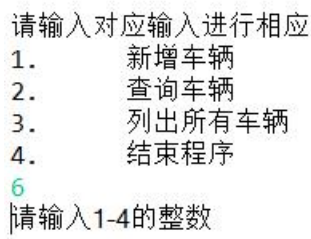
（三）过程截图

（1）全屏截图



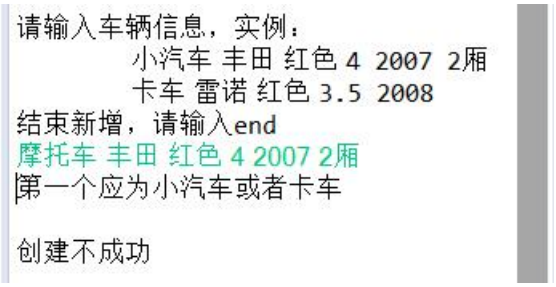
（2）运行结果

① 程序开始显示主菜单，若用户输入除了 1-4 之外的数，提示用户“请输入 1-4 的整数”



② 在主菜单下，若用户输入 1，则提示用户输入车辆信息，若用户输入符合要求，则提示用户“创建成功”。若用户输入“end”，则结束新增，重新显示主菜单。

③ 若用户输入的信息不符合要求，则提示用户相应的错误。



④ 在主菜单下，若用户输入 2，则提示用户按相应的格式输入。若用户输入的信息符合查询格式，则将查询结果返回给用户，若用户输入的信息不符合要求，则会给出相应的提示，若用户输入 end，则返回主菜单。

```
请按照“类型 商标 颜色 出厂年”的顺序输入条件，若条件为空用“null”代替,两个示例:
        小汽车 丰田 红色 2007
        null 丰田 null null
输入end返回上级菜单
小汽车 丰田 红色 2007
搜索到1辆车,信息如下：
小汽车，品牌：丰田 颜色：红色 出厂年份：2007 载客量：4人 厢数：2厢

请按照“类型 商标 颜色 出厂年”的顺序输入条件，若条件为空用“null”代替,两个示例:
        小汽车 丰田 红色 2007
        null 丰田 null null
输入end返回上级菜单
null 丰田 null null
搜索到1辆车,信息如下：
小汽车，品牌：丰田 颜色：红色 出厂年份：2007 载客量：4人 厢数：2厢

请按照“类型 商标 颜色 出厂年”的顺序输入条件，若条件为空用“null”代替,两个示例:
        小汽车 丰田 红色 2007
        null 丰田 null null
输入end返回上级菜单
摩托车 丰田 红色 2007
没有搜索到相应的交通工具

请按照“类型 商标 颜色 出厂年”的顺序输入条件，若条件为空用“null”代替,两个示例:
        小汽车 丰田 红色 2007
        null 丰田 null null
输入end返回上级菜单
end
```

⑤ 在主菜单下，若用户输入 3，则列出目前已有的所有车辆。

```
请输入对应输入进行相应菜单操作：
1.      新增车辆
2.      查询车辆
3.      列出所有车辆
4.      结束程序
3
目前有2辆车，信息如下：
小汽车，品牌：丰田 颜色：红色 出厂年份：2007 载客量：4人 厢数：2厢
卡车，品牌：雷诺 颜色：红色 出厂年份：2008 载货量：3.50吨
```

⑥ 在主菜单下，若用户输入 4，则退出整个程序。

```
请输入对应输入进行相应菜单操作：
1.      新增车辆
2.      查询车辆
3.      列出所有车辆
4.      结束程序
4
退出程序
```

二、拓展题目：

题目 1：用多态改写实验 5 扩展题目 1

（一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

（二） 实现过程

（1）设计类

① 设计一个基础图形类 Graph，拥有抽象方法 area()用于计算面积，只有声明没有实现。然后实现三角形类 Triangle 和矩形类 Rectangle，继承自 Graph。在 Test 测试类中，可以根据输入的边数实现不同的对象，并计算面积。

（2）多态

① 抽象类 Graph 有一个用于计算面积的抽象方法 draw，两个子类 Triangle 和 Rectangle 分别用于计算三角形和矩形面积。

Graph.java

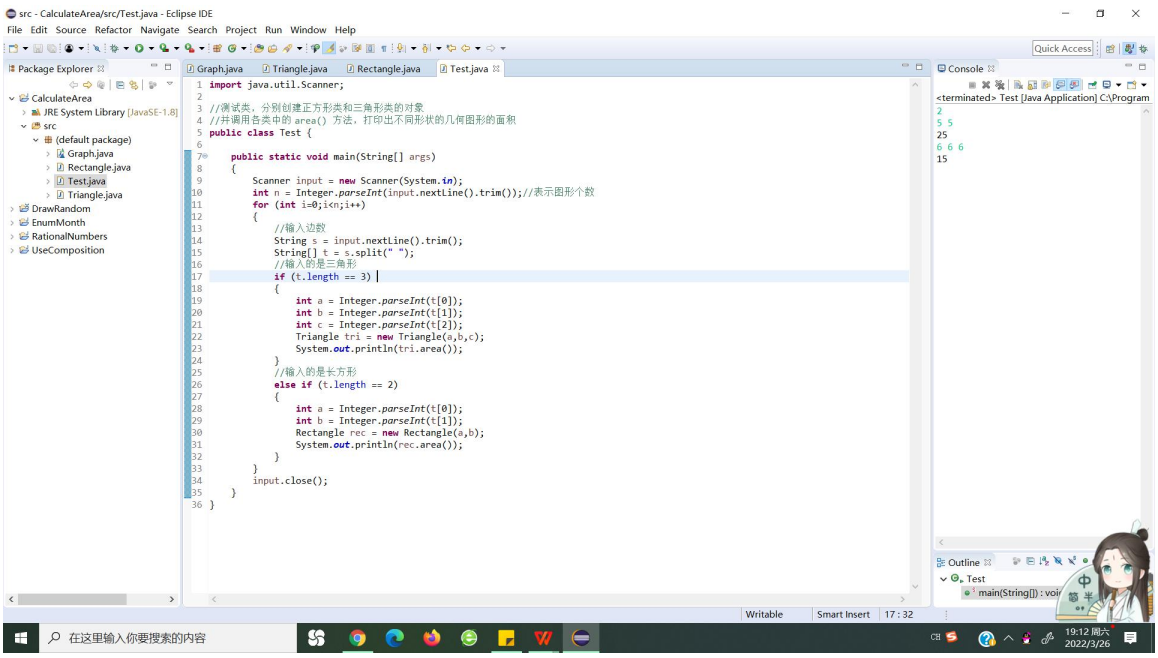
```
1 //表示图形的抽象类Graph
2 abstract class Graph
3 {
4     //抽象方法，计算面积
5     abstract int area();
6 }
```

② 在 test 类中，调用同一个 draw 方法，但是因为父类引用指向的是不同的子类对象（Triangle 和 Rectangle 类型），所以最后可以根据不同对象进行不同的面积公式的计算。

```
if (t.length == 3)
{
    int a = Integer.parseInt(t[0]);
    int b = Integer.parseInt(t[1]);
    int c = Integer.parseInt(t[2]);
    Triangle tri = new Triangle(a,b,c);
    System.out.println(tri.area());
}
//输入的是长方形
else if (t.length == 2)
{
    int a = Integer.parseInt(t[0]);
    int b = Integer.parseInt(t[1]);
    Rectangle rec = new Rectangle(a,b);
    System.out.println(rec.area());
}
```

（三） 过程截图

（1）全屏截图



（2）运行结果

（1）根据输入的边数实现了不同的对象，并正确计算面积。

三、实验总结与心得记录

在本次实验过程中，我练习了多态，熟悉了 java 的语法，熟悉了 java 类的定义，实例化和调用。我也熟悉了简单的 JavaFX 图形界面，体会到了 JAVA 语言的优点。