

# OO 第七次作业-1 要求

2020.4

## 1. 作业目标

掌握类的继承、多态性使用方法以及接口的应用。

## 2. 作业要求

### 2.1 业务背景

#### (1) 卡片大小排序游戏规则

考虑一款适合于小学生的卡片 (Card) 排序游戏, 其规则为随机发放一些卡片给学生, 卡片分为四种形状: 圆形 (Circle)、矩形 (Rectangle)、三角形 (Triangle) 及梯形 (Trapezoid), 并给出各种卡片的相应参数, 要求学生能够迅速求出各卡片的面积大小然后将卡片按照其面积值**从大到小**进行排序, 同时求出所有卡片的面积之和。

#### (2) Comparable 接口的应用

图形类设计可参考作业 6-1 的类层次结构, 本次作业要求对卡片排序时使用 Comparable 接口, 即 Card 类需要实现 Comparable 接口中的 compareTo() 方法。

### 2.2 程序功能需求

#### (1) 实现功能

本程序仅用于为学生所玩的游戏提供正确答案的功能, 即根据学生得到的卡片种类与数量, 给出排序前和排序后的卡片顺序, 同时给出所有卡片的面积之和。

#### (2) 输入输出规则

##### ①输入规则

首先, 在一行上输入一串数字 (1~4, 整数), 其中, 1 代表圆形卡片, 2 代表矩形卡片, 3 代表三角形卡片, 4 代表梯形卡片。各数字之间以一个或多个空格分隔, 以“0”结束。例如:

1 3 4 2 1 3 4 2 1 3 0

然后根据第一行数字所代表的卡片图形类型, 依次输入各图形的相关参数, 例如: 圆形卡片需要输入圆的半径, 矩形卡片需要输入矩形的宽和长, 三角形卡片需要输入三角形的三条边长, 梯形需要输入梯形的上底、下底以及高。各数据之间用一个或多个空格分隔。

##### ②输出规则

- 如果图形数量非法 (小于 0) 或图形属性值非法 (数值小于 0 以及三角形三边不能组成三角形), 则输出“Wrong Format”。

- 如果输入合法，则正常输出，所有数值计算后均保留小数点后两位即可。输出内容如下：
  - 排序前的各图形类型及面积，格式为“图形名称 1:面积值 1 图形名称 2:面积值 2 ...图形名称 n:面积值 n”，注意，各图形输出之间用空格分开，且输出最后存在一个用于分隔的空格；
  - 排序后的各图形类型及面积，格式同排序前的输出；
  - 所有图形的面积总和，格式为“Sum of area:总面积值”。

输出格式见输入输出示例。

### 3. 作业内容和成果物

#### 3.1 作业内容

程序源码。

#### 3.2 提交内容

在 PTA 系统中提交程序源码进行测试。

### 4. 作业要求和限制

#### 4.1 输入输出示例

输入示例 1:

```
1 5 3 2 0
```

输出示例 1:

```
Wrong Format
```

输入示例 2:

```
4 2 1 3 0
3.2 2.5 0.4 2.3 1.4 5.6 2.3 4.2 3.5
```

输出示例 2:

```
The original list:
Trapezoid:1.14 Rectangle:3.22 Circle:98.52 Triangle:4.02
The sorted list:
Circle:98.52 Triangle:4.02 Rectangle:3.22 Trapezoid:1.14
Sum of area:106.91
```

输入示例 3:

```
4 2 1 3 0
3.2 2.5 0.4 2.3 1.4 5.6 2.3 4.2 8.4
```

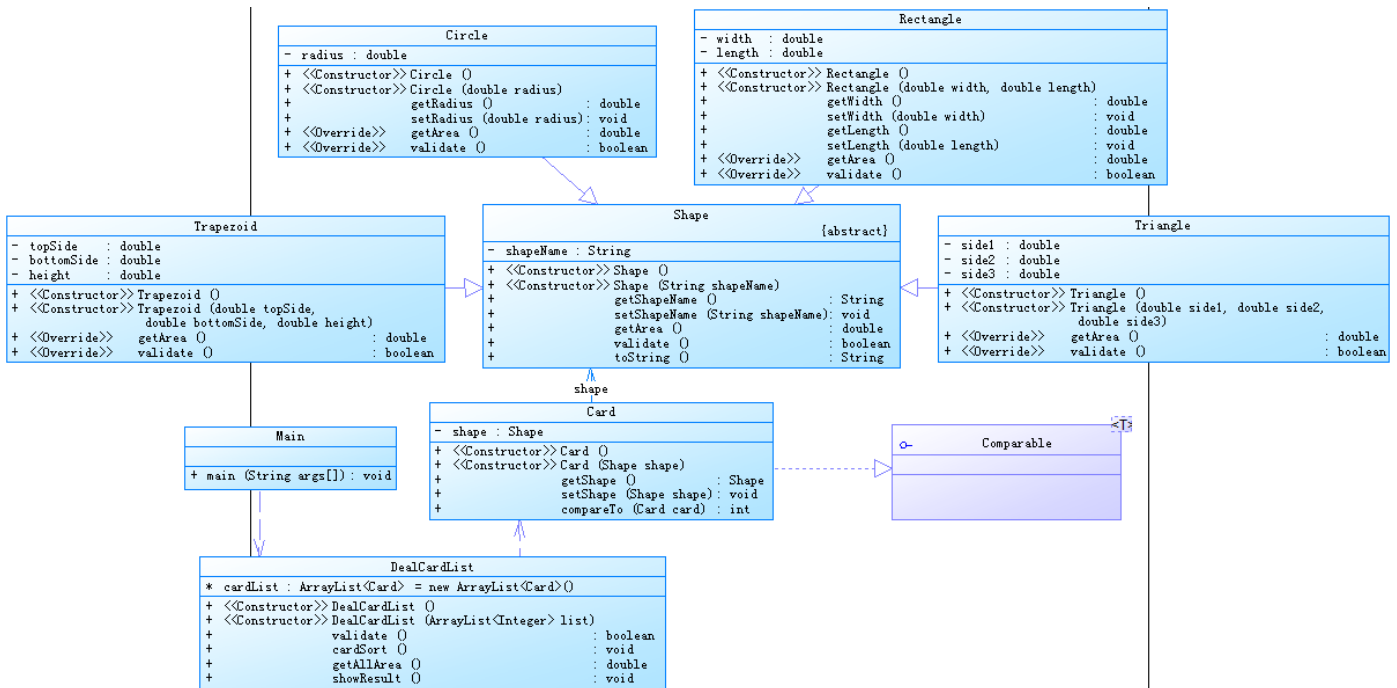
输出示例 3:

```
Wrong Format
```

输入、输出字符编码采用 utf-8 格式，例如，逗号为英文半角字符“，”，而非中文全角字符“，”。

## 4.2 设计要求

### (1) 参考类图



### (2) 排序要求

#### 主方法源码:

```
public class Main {
    //在Main类中定义一个静态Scanner对象，这样在其它类中如果想要使用该对象进行输入，则直接
    //使用Main.input.next...即可（避免采坑）
    public static Scanner input = new Scanner(System.in);

    public static void main(String[] args){
        ArrayList<Integer> list = new ArrayList<Integer>();

        int num = input.nextInt();

        while(num != 0){
            if(num < 0 || num > 4){
                System.out.println("Wrong Format");
                System.exit(0);
            }
            list.add(num);
            num = input.nextInt();
        }

        DealCardList dealCardList = new DealCardList(list);

        if(!dealCardList.validate()){
            System.out.println("Wrong Format");
            System.exit(0);
        }
    }
}
```

```
    }  
  
    dealCardList.showResult();  
  
    input.close();  
  }  
}
```

## 4.4 测试准则

PTA

## 5. 其它说明事项

### 5.1 设计建议

(1) 将本次作业程序与作业 6-1 进行比较，尤其是排序算法的运用；

(2) 在图形类的设计中，体会面向对象设计原则中的“开-闭”原则的实现方法及其作用，思考假如游戏中再增加一种图形，本次设计是否能实现“开-闭”原则。

### 5.2 Tips

类结构如何完善才能使得系统具有较好的可复用性。

通过此例，进一步深入理解面向对象设计原则中的“单一职责”原则和“开-闭”原则。

## 6. 其他规定

(1) 文档中**粗体**字体部分为强制要求。

(2) 无效作业，以下三种情况视为无效作业。

- 1) 程序不能编译和运行；
- 2) 无法通过任何一个可以输出正常结果的公共测试案例；
- 3) 测试程序无法识别。