Java 第二次上机报告

17180210027 李欣

2.创建一个 NewRectangle 类，该类包含：
(1)double 类型成员变量 width 和 height。
(2)默认构造方法（能够将 width 和 height 均置为 0.0）
(3)带两个参数的构造方法（分别对 width 和 height 用参数进行初始化）
(4)成员方法 getArea()，返回矩形面积
(5)成员方法 getPerimeter()，返回矩形周长

具体构建的类和方法参见代码，此处只展示运行结果。

```
E:\大学课程学习\Java程序设计\第二次上机\代码\2
λ javac NewRectangle.java

E:\大学课程学习\Java程序设计\第二次上机\代码\2
λ java NewRectangle
3.0; 3.0
Area of Rectangle: 9.0
Perimeter of Rectangle: 12.0
```

3.在第二题的基础上完成如下程序：
(1)定义 Point 类，该类包含：
1) double 类型的成员变量 x 和 y，代表点的坐标
2) 默认的构造方法，将 x 和 y 置为 0.0
3) 带两个参数的构造方法，用于初始化 x 和 y
4) 成员方法 distance(Point p)，返回当前点与 p 所代表点之间的距离
(2)修改第二题实现的 NewRectangle 类，加入一个 Point 类型的成员，代表矩形左下角顶点的坐标
(3)修改 NewRectangle 已有的两个构造方法，将构造出矩形的左下角顶点坐标为(0.0,0.0)。再向 NewRectangle 中加入一个新的构造方法，方法传入 4 个参数，分别对 width、height、左下角顶点的 x 与 y 坐标进行设置
(4)向 NewRectangle 类中加入一个新的成员方法：bool bPointIn(Point p)，计算 p 所代表的点是否在矩形内部，是返回 true，否返回 false
(5)考虑，能否为 NewRectangle 类实现成员方法，计算当前矩形是否包含了另一个矩形，以及当前矩形是否与另一个矩形有重叠部分，如果可以，请实现这些方法。

第 5 问的方法实现，利用数学上的方法判断

```
boolean bRectangleIn(NewRectangle r) {
    if(r.p.x>this.p.x&&r.p.y>this.p.y&&(r.width+r.p.x)<(this.width+this.p.x)&&(r.height+r.p.y)<(this.height+this.p.y))
        return true;
    else
        return false;
}
boolean bRectangleLap(NewRectangle r) {
    if(r.p.x<(this.p.x+this.width)&&(r.p.x+r.width)>this.p.x&&r.p.y<(this.p.y+this.height)&&(r.p.y+r.height)>this.p.y)
        return true;
    else
        return false;
}
```

还有另一种实现方案，没在代码中体现，那就是判断矩形的四个点是不是都在该矩形内部，如果都在，就是包含，部分在，就是重叠，都不在，就是没有重叠部分。

具体的各个类和方法的实现均在代码中体现，这里只放一下测试代码运行结果：

```
    public static void main(String[] args) {
        NewRectangle rgl=new NewRectangle();
        System.out.println(rgl.getArea());
        System.out.println(rgl.getPerimeter());
        NewRectangle rgl1=new NewRectangle(5.0,6.0);
        System.out.println(rgl1.getArea());
        System.out.println(rgl1.getPerimeter());
        Point p=new Point(4.9,5.8);       //Test p in r2
        NewRectangle rgl2=new NewRectangle(5.0,6.0,0,0);
        System.out.println(rgl2.bPointIn(p));
        NewRectangle rgl3=new NewRectangle(5,4,1,1);    //Test r2 r3
        System.out.println(rgl2.bRectangleIn(rgl3));
        NewRectangle rgl4=new NewRectangle(2.1,4,-2,-1);    //Test r2 r4
        System.out.println(rgl2.bRectangleLap(rgl4));
    }
}
```

```
E:\大学课程学习\Java程序设计\第二次上机\代码\3
λ java NewRectangle
0.0
0.0
30.0
22.0
true
false
true
```

9.创建一个父类 Cycle，再创建三个它的子类 Unicycle 、Bicycle 和 Tricycle:
(1)在 Cycle 类中定义 ride()方法，使得三个子类的实例都能通过方法参数向上转型为 Cycle 类型
(2)向 Cycle 类中加入 wheel()方法，用于返回车轮数量。在 Unicycle、Bicycle、Tricycle 中重写 wheel()方法，返回具体车轮数量。在 ride()方法中调用 wheel()方法，输出参数所指向的对象的车轮数量，并说明多态性的存在
(3)在 Unicycle 和 Bicycle 中添加方法 balance()，在 Tricycle 中不添加。在 ride()方法中，尝试通过向下转型调用 balance()方法，并使用 instanceof 保证向下转型不会产生异常。

多态性验证：

```java
public static void main(String[] args){
    Unicycle u=new Unicycle();
    Bicycle b=new Bicycle();
    Tricycle t=new Tricycle();
    Cycle cyc=new Cycle();
    cyc.ride(u);    //test 2 duotai
    cyc.ride(b);
    cyc.ride(t);
    if (u instanceof Unicycle)
        ((Unicycle)u).balance();
    if (b instanceof Bicycle)
        ((Bicycle)b).balance();
}
```

输出结果：

```
E:\大学课程学习\Java程序设计\第二次上机\代码\9
λ java Cycles
1
2
3
balance1
balance2
```

ride 的三种输出证明了多态性的存在！

balance1 和 balance2 的输出证明成功向下转型！