

# 高级语言程序设计

## 上级报告2

### 习题3-2

#### 题目

创建一个NewRectangle类，该类包含：

1. double类型的成员变量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。

#### 思路

Just Code

### 习题3-3

#### 题目

在第2题的基础上，完成如下程序：

1. 定义Point类，该类包含：
  - double类型的成员变量x和y，代表点的坐标；
  - 默认构造方法，将x和y置于0.0；
  - 带两个参数的构造方法，用于初始化x和y；
  - 成员方法getPerimeter(Point p)返回当前点与p所代表的点之间的距离。
2. 修改第2题实现的NewRectangle类，加入一个Point类型的成员，代表矩形的左下角顶点的坐标。
3. 修改NewRectangle已有的两个构造方法，将构造出的矩形的左下角顶点坐标设为(0.0, 0.0)。再向NewRectangle中加入一个新的成员方法，方法传入4个参数，分别对width、height、左下角顶点的x坐标、左下角顶点的y坐标进行设置。
4. 向NewTriangle中加入一个新的成员方法：boolean bPointIn(Point p)，计算p所代表的点的坐标是否在矩形内部，如果在内部则返回true，否则返回false。

5. 考虑：能否为NewRectangle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。

## 思路

Just Code

题目2-2/2-3的源码位于 `{project}/src/main/java/com/frank/course/chapter_2/geometry` 中

## 程序输出

编写了测试

类 `{project}/src/test/java/com/frank/course/chapter_2/geometry/GeometryTest.java`，其内容如下：

```

package com.frank.course.chapter_2.geometry;
import static org.junit.Assert.*;
import java.util.Random;
import org.junit.Test;
public class GeometryTest {
    Random random = new Random();
    Point p = new Point();
    NewRectangle r = new NewRectangle();
    @Test
    public void initialization() {
        p = new Point();
        r = new NewRectangle();
        assertTrue(p.getX() == 0 && p.getY() == 0);
        assertTrue(r.height == r.width && r.width == 0);
        double w = random.nextDouble(), h = random.nextDouble();
        r.set(w, h);
        assertEquals(2 * (w + h), r.getPerimeter(), 0.0001);
        assertEquals(w * h, r.getArea(), 0.0001);
    }
    @Test
    public void setValue() {
        initialization();
        double x = random.nextDouble(), y = random.nextDouble();
        double a = random.nextDouble(), b = random.nextDouble();
        p.set(x, y);
        r.set(a, b);
        // throws no exception
    }
    @Test
    public void pointInRectangle() {
        initialization();
        setValue();
        r.set(p.x, p.y);
        assertTrue(r.bPointIn(p));
        p.set(p.x+1, p.y);
        assertFalse(r.bPointIn(p));
    }
}

```

## 习题3-9

### 题目

创建一个父类Cycle，再创建它的子类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，返回具体的车轮数量。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类中添加方法balance()方法，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。

## 思路

按照题目要求编写相应类

## 输出结果

对于测试类

```
package com.frank.course.chapter_2.transport;
import static org.junit.Assert.*;
import org.junit.Test;
public class TransportTest {
    final Cycle cycles[] = { new Cycle(), new Unicycle(), new Bicycle(), new Tricycle() }
    @Test
    public void ensureWheels() {
        for (int i = 0; i < 4; i++)
            assertTrue(cycles[i].wheel() == i);
    }
    @Test
    public void tryBalance() {
        for (int i = 0; i < 4; i++)
            if(cycles[i] instanceof Bicycle )
                assertEquals(((Bicycle)cycles[i]).balance(), "Bicycles are hard to ride")
            else if(cycles[i] instanceof Unicycle)
                assertEquals(((Unicycle)cycles[i]).balance(), "Unicycles are hard to ride")
    }
}
```

程序通过了测试

## 附录

```
[INFO] Running com.frank.course.chapter_2.transport.TransportTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in com.frank.course.chapter_2.transport.TransportTest
[INFO] Running com.frank.course.chapter_2.geometry.GeometryTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in com.frank.course.chapter_2.geometry.GeometryTest
```