

《Java 面向对象程序设计 SSD3》

实验报告

项目名称 实验 4 《面向对象——继承》

专业班级 软件工程 2005 班

学 号 8209200504

姓 名 李均浩

实验成绩:

批阅教师: 邝砾

2021 年 5 月 11 日

中南大学计算机学院实验报告

课程名称 Java 面向对象程序设计 SSD3

实验项目名称 实验 4 《面向对象——继承》

学生姓名 李均浩 专业班级 软件工程 2005 班 学号 8209200504

实验成绩 _____ 日期 2021 年 5 月 10 日

实验学时: 2

每组人数: 1

实验类型: 1 (1: 基础性 2: 综合性 3: 设计性 4: 研究性)

实验要求: 1 (1: 必修 2: 选修 3: 其它)

实验类别: 2 (1: 基础 2: 专业基础 3: 专业 4: 其它)

一、实验目的

1. 理解面向对象编程，尤其是继承的思想，学习通过继承，在父类的基础上创建子类
2. 使用关键字 super 调用父类的构造方法和方法
3. 在子类中覆盖方法

二、实验内容

1. (P380, 11.1) 【三角形类 Triangle】设计一个名为 Triangle 的类来扩展 GeometricObject 类。该类包括：

- ◆ (1) 三个名为 side1、side2 和 side3 的 double 数据域表示这个三角形的三条边，它们的默认值是 1.0。
- ◆ (2) 一个无参构造方法创建默认的三角形。
- ◆ (3) 一个能创建带指定 side1、side2 和 side3 的三角形的构造方法。

- ◆ (4) 所有三个数据域的访问器方法。
- ◆ (5) 一个名为 `getArea()` 的方法返回这个三角形的面积。计算三角形面积的公式参见实验 1 练习题 2.19。
- ◆ (6) 一个名为 `getPerimeter()` 的方法返回这个三角形的周长。
- ◆ (7) 一个名为 `toString()` 的方法返回这个三角形的字符串描述，显示三角形的三边长、周长及面积。

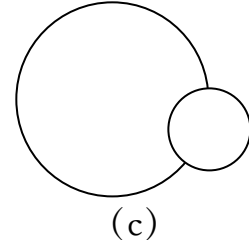
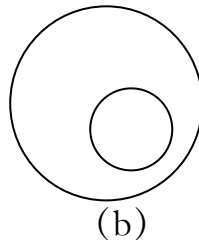
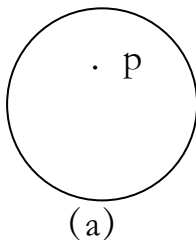
目标输出任务：

- ✓ 画出 `Triangle` 类和 `GeometricObject` 类的 UML 图。
- ✓ 实现这些类。
- ✓ 编写一个测试程序，创建边长为 1、1.5 和 1，颜色为 yellow，filled 为 true 的 `Triangle` 对象，然后显示它的三边长、周长、面积、颜色以及是否被填充。

2. (P342, 10.11) **【Circle2D 类】** 定义 `Circle2D` 类，包括：

- ◆ (1) 两个带有 `get` 方法的名为 `x` 和 `y` 的 `double` 型数据域，表明圆的中心点。
- ◆ (2) 一个带 `get` 方法的数据域 `radius`。
- ◆ (3) 一个无参构造方法，该方法创建一个 `(x, y)` 值为 `(0, 0)` 且 `radius` 为 1 的默认圆。
- ◆ (4) 一个构造方法，创建带指定的 `x`、`y` 和 `radius` 的圆。
- ◆ (5) 一个返回圆面积的方法 `getArea()`。

- ◆ (6) 一个返回圆周长的方法 `getPerimeter()`。
- ◆ (7) 如果给定的点 (x, y) 在圆内, 那么方法 `contains(double x, double y)` 返回 `true`。如图 a 所示。
- ◆ (8) 如果给定的圆在这个圆内, 那么方法 `contains(Circle2D circle)` 返回 `true`。如图 b 所示。
- ◆ (9) 如果给定的圆和这个圆重叠, 那么方法 `overlaps(Circle2D circle)` 返回 `true`。如图 c 所示。



目标输出任务:

- ✓ 画出该类的 UML 图。
- ✓ 实现这个类。
- ✓ 编写测试程序, 创建一个 `Circle2D` 对象 `c1(new Circle2D(2, 2, 5.5))`, 显示它的面积和周长, 还要显示 `c1.contains(3, 3)`、`c1.contains(new Circle2D(4, 5, 10.5))` 和 `c1.overlaps(new Circle2D(3, 5, 2.3))`。

三、实验要求:

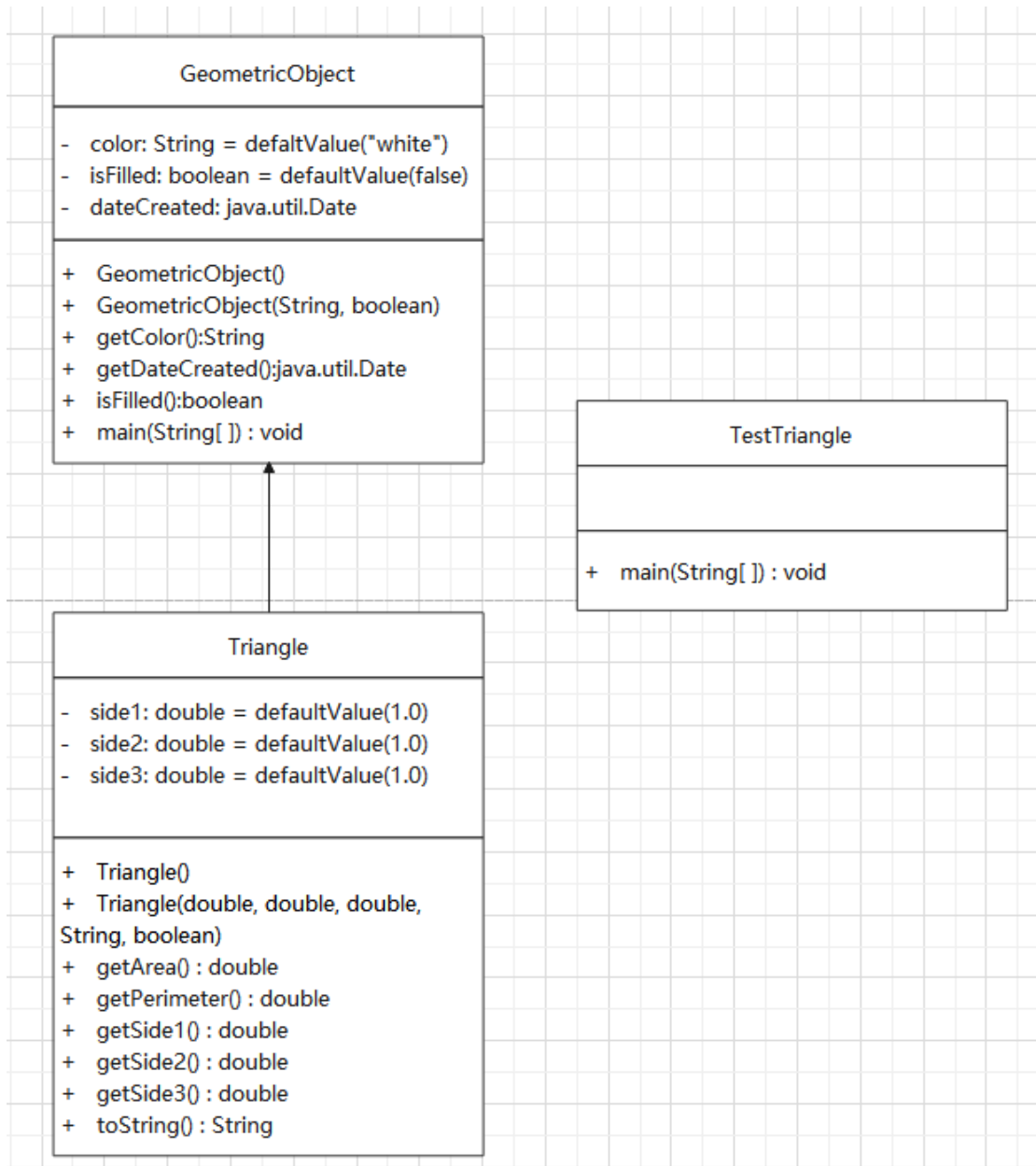
要求每个学生独立完成实验任务。

四、实验报告

1. 实验结果与分析

第一题：

UML 类图：



测试点(a)：使用题设提供值

(1)测试结果：

```
Triangle{color='yellow', isFilled=true, dateCreated=Tue May 11 16:43:30 CST 2021, side1=1.0, side2=1.5, side3=1.0}
面积为: 0.49607837082461076
周长为: 3.5
```

进程已结束 退出代码为 0

(2)结果分析

在 `main` 函数中构建了 `Triangle` 类的实例，向构造方法传递值，设定了父类 `GeometricObject` 以及 `Triangle` 类的所有属

性，经过 toString() 的字段拼接输出了所有的属性，getArea() 输出面积和 getPerimeter() 输出周长，结果符合预期。

测试点(b):

```
side1: 2.1, side2: 7.5, side3: 8, color: "Azure", isFilled: false;
```

(1)测试结果:

```
Triangle{color='Azure', isFilled=false, dateCreated=Tue May 11 16:47:23 CST 2021, side1=2.1, side2=7.5, side3=8.0}  
面积为: 7.83060661762549  
周长为: 17.6
```

进程已结束·退出代码为 0

(2)结果分析

在 main 函数中构建了 Triangle 类的实例，向构造方法传递值，设定了父类 GeometricObject 以及 Triangle 类的所有属性，经过 toString() 的字段拼接输出了所有的属性，getArea() 输出面积和 getPerimeter() 输出周长，结果符合预期。

测试点(c):

```
Triangle triangle = new Triangle();
```

(1)测试结果:

```
Triangle{color='white', isFilled=false, dateCreated=Tue May 11 16:51:51 CST 2021, side1=1.0, side2=1.0, side3=1.0}  
面积为: 0.4330127018922193  
周长为: 3.0
```

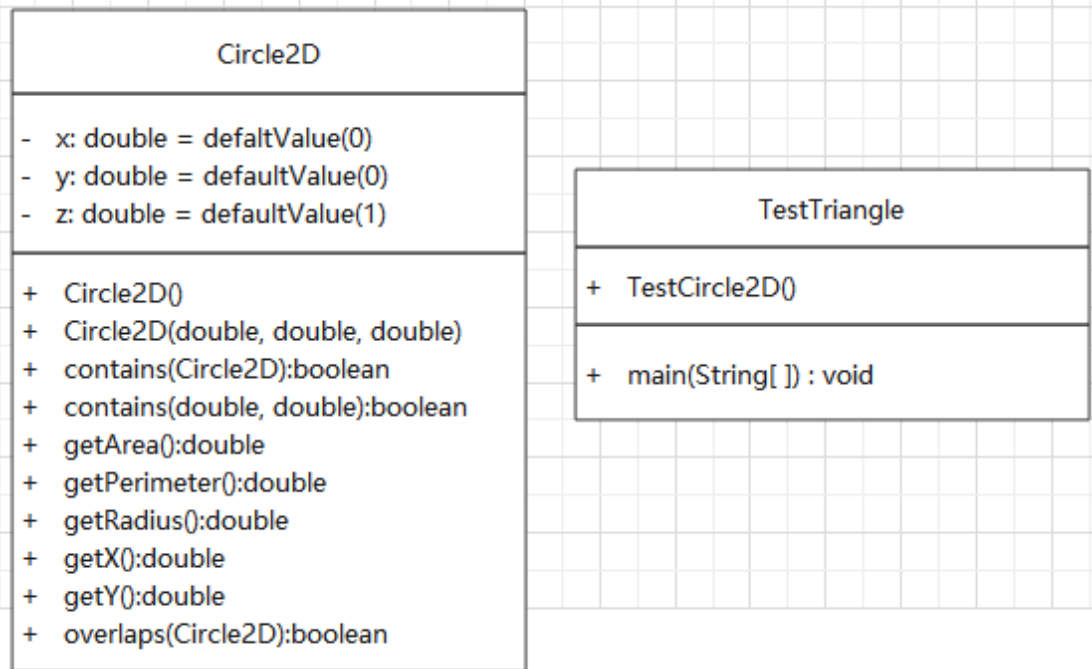
进程已结束·退出代码为 0

(2)结果分析

在 main 函数中构建了 Triangle 类的实例，使用了 Triangle 的无参构造函数，同时也调用了父类的无参构造函数，将父类 GeometricObject 以及 Triangle 类的所有属性设定为默认值，经过 toString() 的字段拼接输出了所有的属性，getArea() 输出面积和 getPerimeter() 输出周长，结果符合预期。

第二题：

UML 类图：



测试点(a)：使用题设提供值

(1)测试结果：

面积为： 95.03317777109125

周长为： 34.55751918948772

c1.contains(3,3) = true

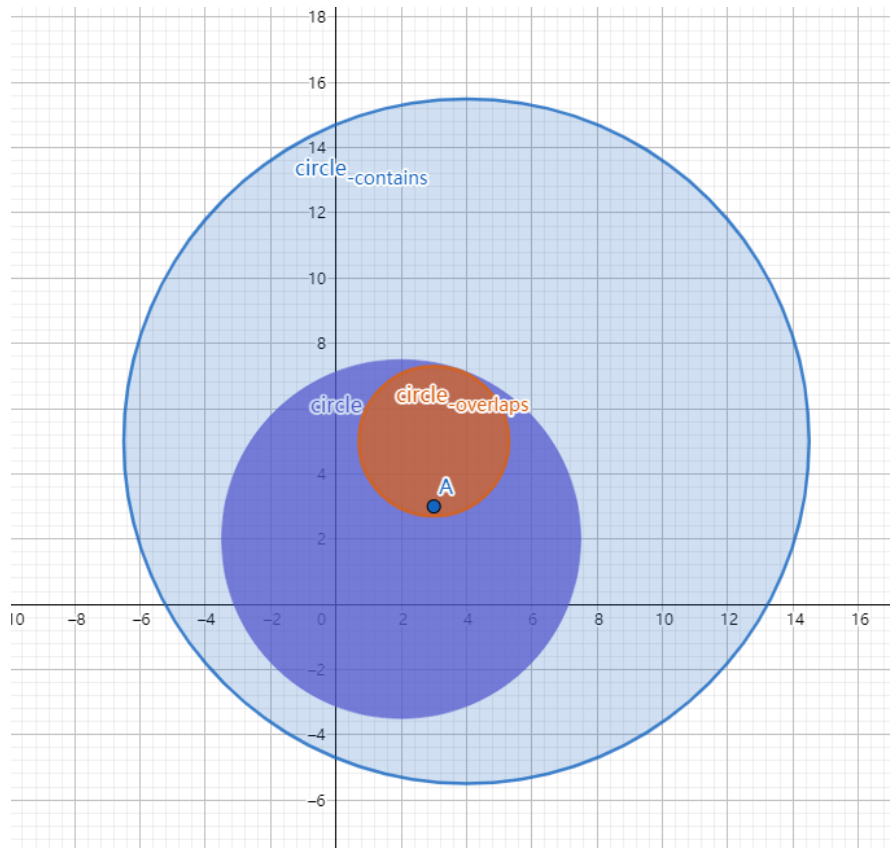
c1.contains(new Circle2D(4, 5, 10.5)) = false

c1.overlaps(new Circle2D(3, 5, 2.3)) = false

进程已结束：退出代码为 0

(2)结果分析

通过 getArea()的计算可得到 c1 的面积，通过 getPerimeter()可得到 c1 的周长，三个图形与 c1 的相对位置如下图所示，程序运行结果符合预期。



附图4.2.1 图形的相对位置(a)

测试点(b)： c1(1,5,9) 点 A(4,7) circle_contains(8,5,7)
circle_overlaps(2,6,4)

(1)测试结果：

面积为： 254.46900494077323

周长为： 56.548667764616276

c1.contains(3,3) = true

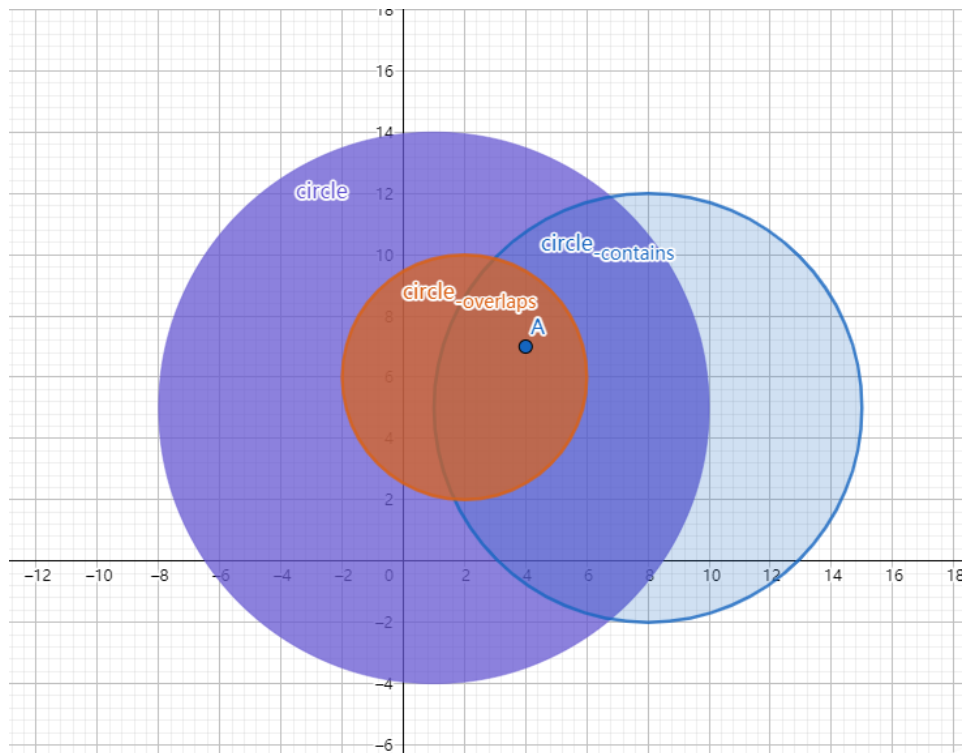
c1.contains(new Circle2D(4, 5, 10.5)) = false

c1.overlaps(new Circle2D(3, 5, 2.3)) = false

进程已结束：退出代码为 0

(2)结果分析

通过 getArea()的计算可得到 c1 的面积，通过 getPerimeter()可得到 c1 的周长，三个图形与 c1 的相对位置如下图所示，程序运行结果符合预期。



附图4.2.2 图形的相对位置(b)

测试点 (c) : c1(7,1,5) 点 A(1,3) circle_contains(2,2,2)
circle_overlaps(3,3,8)

(1)测试结果:

面积为: 78.53981633974483

周长为: 31.41592653589793

c1.contains(3,3) = false

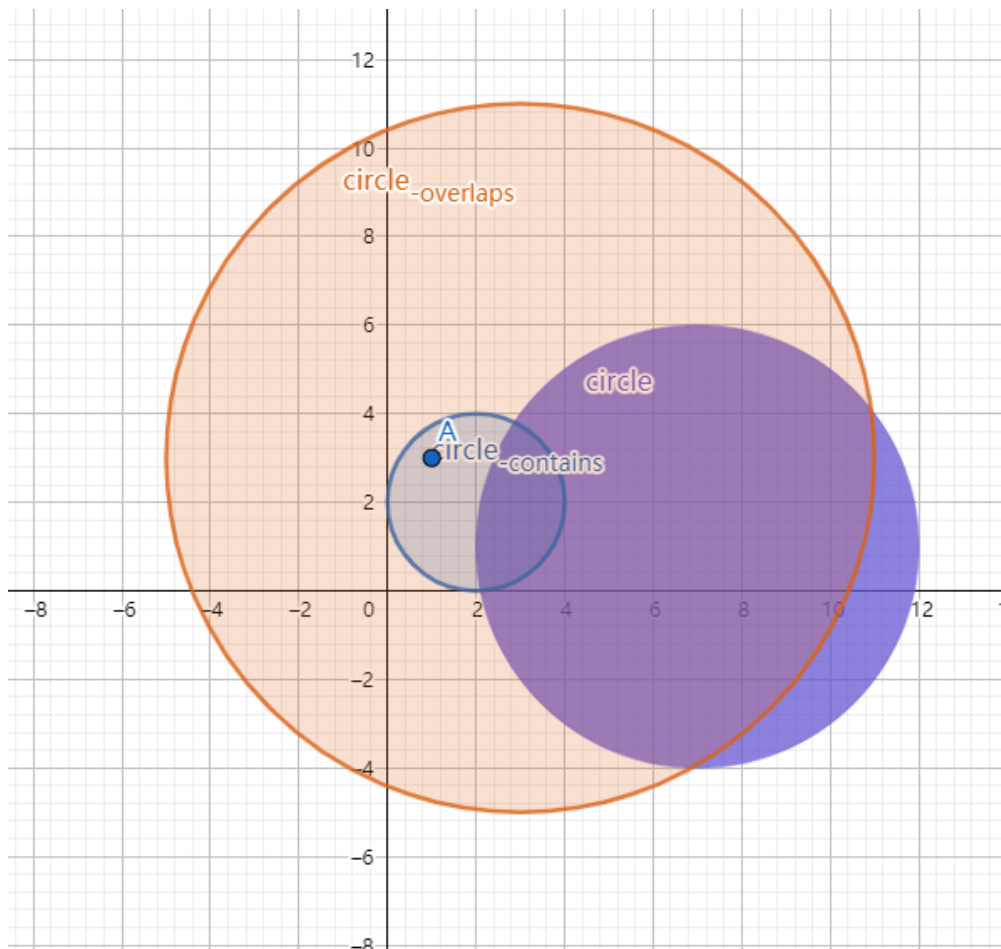
c1.contains(new Circle2D(4, 5, 10.5)) = false

c1.overlaps(new Circle2D(3, 5, 2.3)) = true

进程已结束: 退出代码为 0

(2)结果分析

通过 getArea()的计算可得到 c1 的面积, 通过 getPerimeter()可得到 c1 的周长, 三个图形与 c1 的相对位置如下图所示, 程序运行结果符合预期。



附图4.2.2 图形的相对位置(b)

2. 心得体会

第一题练习了类的继承，对父类构造方法的调用，类的继承可以减少代码的重复书写，使得代码变得更加整洁和容易辨识，增强了可重用性，与此同时也建立了类与类之间的一种联系，在各种项目的实现中极为重要。

第二题，构造了类 Circle2D 类，巩固了 getter 和 setter 的使用，方法的编写、调用、返回，数据可见性的设置等。本题的关键点在于对三种相对位置情况的判断，把握住条件即可。

【附源程序】

```
import java.util.Date;

public class GeometricObject {
    private final String color;
    private final boolean isFilled;
    private final Date dateCreated;

    GeometricObject() {
        color = "white";
        isFilled = false;
        dateCreated = new Date();
    }

    GeometricObject(String color, boolean isFilled) {
        this.color = color;
        this.isFilled = isFilled;
        dateCreated = new Date();
    }

    public String getColor() {
        return color;
    }

    public boolean isFilled() {
        return isFilled;
    }

    public Date getDateCreated() {
        return dateCreated;
    }
}

class Triangle extends GeometricObject {
    private final double side1;
    private final double side2;
    private final double side3;

    Triangle() {
        super();
        this.side1 = 1.0;
        this.side2 = 1.0;
        this.side3 = 1.0;
    }
}
```

```

    Triangle(double side1, double side2, double side3, String color,
boolean isFilled) {
    super(color, isFilled);
    this.side1 = side1;
    this.side2 = side2;
    this.side3 = side3;
}

public double getSide1() {
    return side1;
}

public double getSide2() {
    return side2;
}

public double getSide3() {
    return side3;
}

public double getArea() {
    double p = (this.side1 + this.side2 + this.side3) / 2;
    return Math.sqrt(p * (p - this.side1) * (p - this.side2) * (p
- this.side3));
}

public double getPerimeter() {
    return this.side1 + this.side2 + this.side3;
}

@Override
public String toString() {
    return "Triangle{" +
        "color='" + getColor() + '\'' +
        ", isFilled=" + isFilled() +
        ", dateCreated=" + getDateCreated() +
        ", side1=" + side1 +
        ", side2=" + side2 +
        ", side3=" + side3 +
        '}';
}
}

```

```

class TestTriangle{
    public static void main(String[] args) {
        Triangle triangle = new Triangle(1,1.5,1,"yellow",true);
        System.out.println(triangle.toString());
        System.out.println("面积为: "+triangle.getArea());
        System.out.println("周长为: "+triangle.getPerimeter());
    }
}

```

源代码4.1 GeometricObject.java

```

import javax.swing.plaf.metal.MetalSliderUI;

public class Circle2D {
    private double x;
    private double y;
    private double radius;

    Circle2D() {
        this.x = 0;
        this.y = 0;
        this.radius = 1;
    }

    Circle2D(double x, double y, double radius) {
        this.x = x;
        this.y = y;
        this.radius = radius;
    }

    public double getX() {
        return x;
    }

    public double getY() {
        return y;
    }

    public double getRadius() {
        return radius;
    }

    public double getArea() {
        return this.radius * this.radius * Math.PI;
    }
}

```

```

    }

    public double getPerimeter() {
        return 2 * Math.PI * this.radius;
    }

    public boolean contains(double x, double y) {
        return Math.sqrt((x - this.x) * (x - this.x) + (y - this.y) *
(y - this.y)) < this.radius;
    }

    public boolean contains(Circle2D circle) {
        return Math.sqrt((circle.x - this.x) * (circle.x - this.x) +
(circle.y - this.y) * (circle.y - this.y)) + circle.radius <=
this.radius;
    }

    public boolean overlaps(Circle2D circle) {
        return Math.sqrt((circle.x - this.x) * (circle.x - this.x) +
(circle.y - this.y) * (circle.y - this.y)) - circle.radius < this.radius
&& (Math.sqrt((circle.x - this.x) * (circle.x - this.x) + (circle.y
- this.y) * (circle.y - this.y)) + circle.radius > this.radius);
    }

}

class TestCircle2D {
    public static void main(String[] args) {
        Circle2D circle = new Circle2D(7, 1, 5);
        System.out.println("面积为: " + circle.getArea());
        System.out.println("周长为: " + circle.getPerimeter());
        System.out.println("c1.contains(3,3) = " + circle.contains(1,
3));
        System.out.println("c1.contains(new Circle2D(4, 5, 10.5)) = "
+ circle.contains(new Circle2D(2, 2, 2)));
        System.out.println("c1.overlaps(new Circle2D(3, 5, 2.3)) = "
+ circle.overlaps(new Circle2D(3, 3, 8)));
    }
}

```

源代码4.2 Circle2D.java