# 实验三 面向对象2：类与对象继承

【实验目的】

1. 理解Java构造方法的含义及应用。
2. 进一步理解Java类的定义及对象的应用。

3. 掌握this关键字的应用。

4. 掌握属性、方法的继承与重载

5. 了解类的继承性和多态性的作用。

【实验要求】

1. 编写一个类，并为这个类编写多个构造方法，在构造方法中应用this关键字。

2. 编写体现类的继承性（成员变量，成员方法，成员变量隐藏）的程序。

3. 编写体现类多态性（成员方法重载，构造方法重载）的程序。

【实验内容】

1.编写一程序，完成以下工作：

（1）合理定义一个三角形类Triangle，成员属性包括3条边，能否构成三角形的标志；成员方法包括构造方法、修改3条边、计算面积。

（2）写一测试类，测试自定义三角形类Triangle是否正确。

public class Triangle {  
 private double side1;  
 private double side2;  
 private double side3;  
 private boolean isTriangle;  
  
 public Triangle(double side1, double side2, double side3) {  
 this.side1 = side1;  
 this.side2 = side2;  
 this.side3 = side3;  
 this.isTriangle = checkTriangleValidity();  
 }  
  
 public void setSides(double side1, double side2, double side3) {  
 this.side1 = side1;  
 this.side2 = side2;  
 this.side3 = side3;  
 this.isTriangle = checkTriangleValidity();  
 }  
  
 public double calculateArea() throws IllegalAccessException {  
 if (!isTriangle) {  
 throw new IllegalAccessException("Triangle is invalided.");  
 }  
  
 double semiPerimeter = (side1 + side2 + side3) / 2;  
 double area = Math.*sqrt*(semiPerimeter \* (semiPerimeter - side1) \*  
 (semiPerimeter - side2) \* (semiPerimeter - side3));  
 return area;  
 }  
  
 private boolean checkTriangleValidity() {  
 return (side1 + side2 > side3) && (side2 + side3 > side1) && (side3 + side1 > side2);  
 }  
}

public class TriangleTest {  
 public static void main(String[] args) {  
 Triangle triangle1 = new Triangle(3, 4, 5);  
 try {  
 double area = triangle1.calculateArea();  
 System.*out*.println("Triangle area is " + area);  
 } catch (IllegalAccessException e) {  
 System.*out*.println("Error occurred in " + e);  
 }  
  
 Triangle triangle2 = new Triangle(3, 4, 1);  
 try {  
 double area = triangle2.calculateArea();  
 System.*out*.println("Triangle area is " + area);  
 } catch (IllegalAccessException e) {  
 System.*out*.println("Error occurred in " + e);  
 }  
 }  
  
}

2.设计程序。

* 1. 定义一个点Point类。
     1. 声明两个属性x和y，表示点的两个坐标。
     2. 为该类创建两个构造方法：
        1. 第一个为有两个参数的构造方法，分别为x和y赋值。
        2. 第二个为无参数构造方法，用this调用第一个构造方法，为x和y赋初值为（1，1）。
     3. 定义一个show方法，来输出显示x和y的值。
  2. 定义一个矩形Rectangle类，使其继承Point类。
     1. 声明两个属性长（length）和宽（width）。
     2. 为该类创建三个构造方法：
        1. 第一个为有四个参数的构造方法：用super显式调用父类的第一个构造方法为x和y赋值，然后为属性length和width赋值。
        2. 第二个为有两个参数（length和width）的构造方法：用super调用父类的第二个构造方法，然后为属性length和width赋值。
        3. 第三个为无参数构造方法，用this调用本类的第一个构造方法，分布为x、y、length和width赋初值为0、0、10和10。
     3. 定义show方法，来输出显示x、y、length和width的值：先用super调用父类的show方法，然后输出显示length和width的值。
     4. 定义area方法，计算矩形的面积。
  3. 定义一个主类Test，分别用三种构造方法创建三个Rectangl对象。
     1. 显示x、y、length和width的值。
     2. 计算面积并显示。

1. public class Point {  
    private double x;  
    private double y;  
     
    public Point(double x, double y) {  
    this.x = x;  
    this.y = y;  
    }  
    public Point() {  
    this(1, 1);  
    }  
     
    void show() {  
    System.*out*.println("x: " + x + " y: " + y);  
    }  
     
     
   }

public class Rectangle extends Point {  
 private double length;  
 private double width;  
  
 public Rectangle(double x, double y, double length, double width) {  
 super(x, y);  
 this.length = length;  
 this.width = width;  
 }  
  
 public Rectangle(double length, double width) {  
 this.length = length;  
 this.width = width;  
 }  
  
 public Rectangle() {  
 this(0, 0, 10, 10);  
 }  
  
 @Override  
 void show() {  
 super.show();  
 System.*out*.println("length: " + length + " width: " + width);  
 }  
  
 double area() {  
 return length \* width;  
 }  
}

public class Test {  
 public static void main(String[] args) {  
 Rectangle rectangle1 = new Rectangle(1, 2, 3, 4);  
 Rectangle rectangle2 = new Rectangle(11.4, 5.14);  
 Rectangle rectangle3 = new Rectangle();  
  
 rectangle1.show();  
 rectangle2.show();  
 rectangle3.show();  
  
 System.*out*.println(rectangle1.area());  
 System.*out*.println(rectangle2.area());  
 System.*out*.println(rectangle3.area());  
  
 }  
}

【实验成绩】