# Java程序设计实验报告

**学号： 1190201421**

**姓名： 张瑞**

**专业： 工科试验班（计算机与电子通信）**

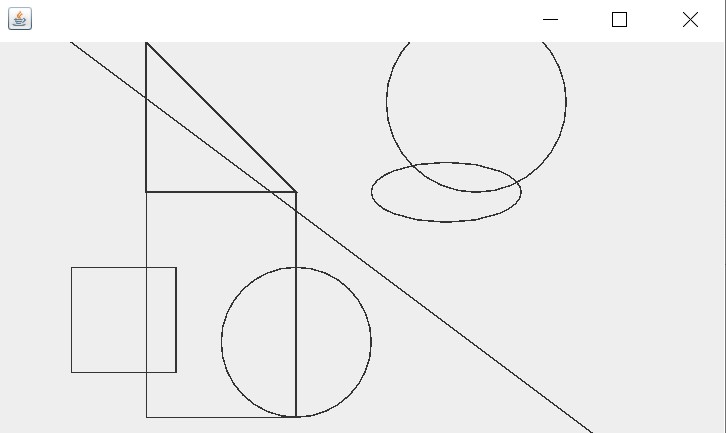
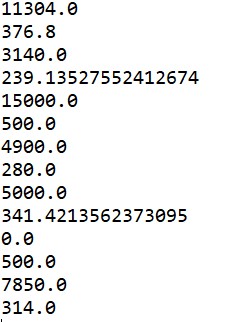
**班级： 19L0214**

**哈尔滨工业大学**

一、实验内容

1. **基于类继承和多态，在我们上午课程Shapes程序的基础上编写一个画图程序，在已有Shape、Line、Circle、Rectangle、Triangle的基础上增加Square和Ellipse两种新的类，并支持图形面积和周长的计算（Line面积设置为0）。**

二、实验运行结果



三、程序源代码

**package** shapes;

**public** **class** MyPic {//我的图像

**public** **static** **void** main(String[] args) {

Picture pic = **new** Picture(500,300);//初始化界面大小

Circle c1 = **new** Circle(320,40,60);//初始化各个图

Ellipse e1 = **new** Ellipse(300,100,50,20);

Rectangle r1 = **new** Rectangle(100, 100, 100, 150);

Square s1 = **new** Square(50,150,70);

Triangle t1 = **new** Triangle(100, 100, 200, 100, 100, 0);

Line l1 = **new** Line(50,0,450,300);

Circle c2 = **new** Circle(200,200,50);

pic.add(c1);//依次将各个图加入pic中，计算并输出各图面积与周长

System.***out***.println(c1.getArea());

System.***out***.println(c1.getPerimeter());

pic.add(e1);

System.***out***.println(e1.getArea());

System.***out***.println(e1.getPerimeter());

pic.add(r1);

System.***out***.println(r1.getArea());

System.***out***.println(r1.getPerimeter());

pic.add(s1);

System.***out***.println(s1.getArea());

System.***out***.println(s1.getPerimeter());

pic.add(t1);

System.***out***.println(t1.getArea());

System.***out***.println(t1.getPerimeter());

pic.add(l1);

System.***out***.println(l1.getArea());

System.***out***.println(l1.getPerimeter());

pic.add(c2);

System.***out***.println(c2.getArea());

System.***out***.println(c2.getPerimeter());

pic.draw(); //将所有图画出

}

}

**package** shapes;

**import** java.awt.Graphics;

**import** java.util.ArrayList;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**public** **class** Picture **extends** JFrame {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** **int** width;

**private** **int** height;

**private** ArrayList<Shape> listShape = **new** ArrayList<Shape>();

**private** **class** ShapesPanel **extends** JPanel {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

@Override

**protected** **void** paintComponent(Graphics g) {

**super**.paintComponent(g);

**for** ( Shape s : listShape ){//依次画出图像

s.draw(g);

}

}

}

**public** **void** add(Shape s){//添加图像

listShape.add(s);

}

**public** Picture(**int** width, **int** height){//设置界面大小并创建储存各图像的容器类

add(**new** ShapesPanel());

**this**.setDefaultCloseOperation(***EXIT\_ON\_CLOSE***);

**this**.width = width;

**this**.height = height;

}

**public** **void** draw(){//画图

setLocationRelativeTo(**null**);

setSize(width, height);

setVisible(**true**);

}

}

**package** shapes;

**import** java.awt.Graphics;

**public** **abstract** **class** Shape {//抽象类——形状

**public** **abstract** **void** draw(Graphics g);//抽象函数——画法、面积、周长

**public** **abstract** **double** getArea();

**public** **abstract** **double** getPerimeter();

}

**package** shapes;

**import** java.awt.Graphics;

**public** **class** Circle **extends** Shape {//圆继承形状

**private** **int** x;

**private** **int** y;

**private** **int** xradius;

**public** Circle(**int** x, **int** y, **int** xradius){//对圆心及半径的初始化

**this**.x = x;

**this**.y = y;

**this**.xradius = xradius;

}

**protected** **int** getx(){//得到圆心x坐标

**return** x;

}

**protected** **int** gety(){//得到圆心y坐标

**return** y;

}

**protected** **int** getxradius(){//得到圆半径

**return** xradius;

}

@Override

**public** **void** draw(Graphics g) {//圆的画法

g.drawOval(x-xradius, y-xradius, xradius\*2, xradius\*2);

}

@Override

**public** **double** getArea() {//圆面积

**return** 3.14\*xradius\*xradius;

}

@Override

**public** **double** getPerimeter() {//圆周长

**return** 2\*3.14\*xradius;

}

}

**package** shapes;

**import** java.awt.Graphics;

**public** **class** Ellipse **extends** Circle {//椭圆继承圆

**private** **int** yradius;

**public** Ellipse(**int** x, **int** y, **int** xradius,**int** yradius){

**super**(x,y,xradius);//由父类对椭圆中心坐标及x轴半径初始化

**this**.yradius = yradius;//比父类多出的变量y轴半径的初始化

}

@Override

**public** **void** draw(Graphics g) {//椭圆的画法

g.drawOval(getx()-getxradius(), gety()-yradius, getxradius()\*2, yradius\*2);

}

@Override

**public** **double** getArea() {//椭圆面积

**return** 3.14\*getxradius()\*yradius;

}

@Override

**public** **double** getPerimeter() {//椭圆周长

**return** 2\*3.14\*Math.*sqrt*((getxradius()\*getxradius()+yradius\*yradius)/2);

}

}

**package** shapes;

**import** java.awt.Graphics;

**public** **class** Line **extends** Shape {//线继承形状

**private** **int** x1;

**private** **int** y1;

**private** **int** x2;

**private** **int** y2;

**public** Line(**int** x1, **int** y1, **int** x2, **int** y2){//将确定线的两顶点坐标初始化

**this**.x1 = x1; **this**.y1 = y1;

**this**.x2 = x2; **this**.y2 = y2;

}

@Override

**public** **void** draw(Graphics g) {//线的画法

g.drawLine(x1, y1, x2, y2);

}

@Override

**public** **double** getArea() {//线的面积（0）

**return** 0;

}

@Override

**public** **double** getPerimeter() {//线的长度

**return** Math.*sqrt*((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2));

}

}

**package** shapes;

**import** java.awt.Graphics;

**public** **class** Square **extends** Shape {//正方形继承形状

**private** **int** x;

**private** **int** y;

**private** **int** width;

**public** Square(**int** x, **int** y, **int** width) {//初始化顶点坐标与边长

**this**.x = x;

**this**.y = y;

**this**.width = width;

}

**protected** **int** getx(){//得到顶点x坐标

**return** x;

}

**protected** **int** gety(){//得到顶点y坐标

**return** y;

}

**protected** **int** getwidth(){//得到边长

**return** width;

}

@Override

**public** **void** draw(Graphics g) {//正方形画法

g.drawRect(x, y, width, width);

}

@Override

**public** **double** getArea() {//正方形面积

**return** width\*width;

}

@Override

**public** **double** getPerimeter() {//正方形周长

**return** 4\*width;

}

}

**package** shapes;

**import** java.awt.Graphics;

**public** **class** Rectangle **extends** Square {//矩形继承正方形

**private** **int** height;

**public** Rectangle(**int** x, **int** y, **int** width, **int** height) {

**super**(x,y,width);//由父类进行顶点坐标与宽的初始化

**this**.height = height;//比父类多出的变量高的初始化

}

@Override

**public** **void** draw(Graphics g) {//矩形画法

g.drawRect(getx(),gety(), getwidth(), height);

}

@Override

**public** **double** getArea() {//矩形面积

**return** getwidth()\*height;

}

@Override

**public** **double** getPerimeter() {//矩形周长

**return** 2\*(getwidth()+height);

}

}

**package** shapes;

**import** java.awt.Graphics;

**public** **class** Triangle **extends** Shape {//三角形继承形状

**private** **int**[] x = **new** **int**[3];

**private** **int**[] y = **new** **int**[3];

**private** **double** a;

**private** **double** b;

**private** **double** c;

**private** **double** s;

**public** Triangle(**int** x1, **int** y1, **int** x2, **int** y2, **int** x3, **int** y3){//初始化三个点的坐标，并计算三条边长

x[0] = x1; x[1] = x2; x[2] = x3;

y[0] = y1; y[1] = y2; y[2] = y3;

a = Math.*sqrt*((x[0]-x[1])\*(x[0]-x[1])+(y[0]-y[1])\*(y[0]-y[1]));

b = Math.*sqrt*((x[1]-x[2])\*(x[1]-x[2])+(y[1]-y[2])\*(y[1]-y[2]));

c = Math.*sqrt*((x[2]-x[0])\*(x[2]-x[0])+(y[2]-y[0])\*(y[2]-y[0]));

s = (a+b+c)/2;

}

@Override

**public** **void** draw(Graphics g) {//三角形的画法

g.drawPolygon(x, y, x.length);

}

@Override

**public** **double** getArea() {//三角形面积

**return** Math.*sqrt*(s\*(s-a)\*(s-b)\*(s-c));

}

@Override

**public** **double** getPerimeter() {//三角形周长

**return** a+b+c;

}

}