

UIET MDU ROHTAK

**JAVA ASSIGNMENT - 03**

# SUBMITTED BY:-

HITESH KUMAR

Roll No. : 27566

Class : CSE-B (5TH SEM)

# SUBMITTED TO :-

DR. DHIRAJ KUMAR SAHNI

1. **Java Program to Print Pattern of Circle.**

**import java.applet.\*; import java.awt.\*;**

**public class DrawCircle extends Applet{ public void paint(Graphics G){**

**G.drawRoundRect(00, 00, 400, 400, 400, 400);**

**G.drawRoundRect(50, 50, 300, 300, 300, 300);**

**G.drawRoundRect(100, 100, 200, 200, 200, 200);**

**G.drawRoundRect(150, 150, 100, 100, 100, 100);**

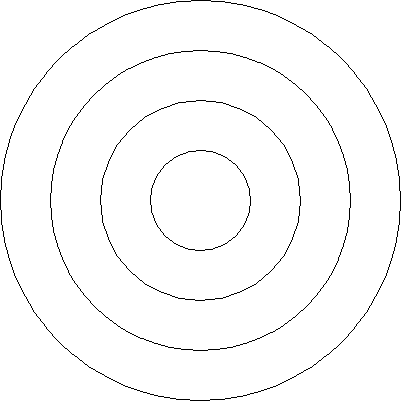
**G.setColor(Color.GREEN);**

**}**

**}**

**OUTPUT:**

**0 1 1 2 3 5 8 13 21 34**



**34**

# Java Program to Print Pattern of Triangle.

**import java.applet.\*; import java.awt.\*;**

**public class DrawTriangle extends Applet{ public void paint(Graphics g) {**

**g.drawLine(300, 0, 600, 600);**

**g.drawLine(0, 600, 600, 600);**

**g.drawLine(0, 600, 300, 0);**

**g.drawLine(300, 100, 500, 500);**

**g.drawLine(100, 500, 500, 500);**

**g.drawLine(100, 500, 300, 100);**

**g.drawLine(300, 200, 400, 400);**

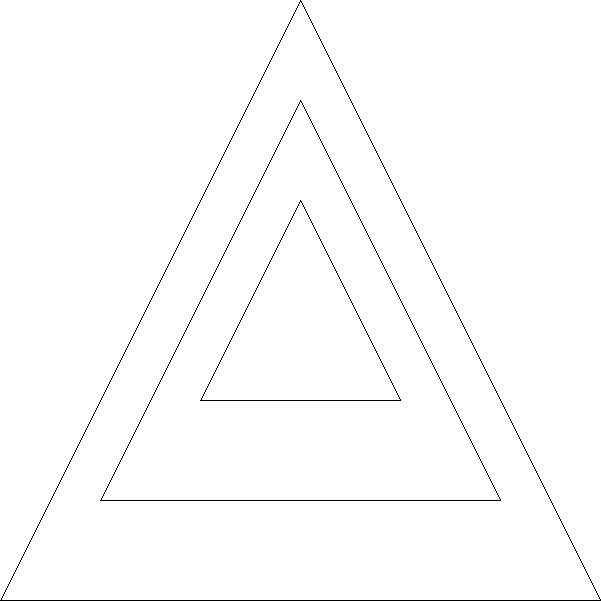
**g.drawLine(200, 400, 400, 400);**

**g.drawLine(200, 400, 300, 200);**

**}**

**}**

**Output:**



## Java Program to print Pattern of Square.

**import java.applet.\*; import java.awt.\*;**

**public class DrawSquare extends Applet { public void paint(Graphics G) {**

**G.drawRect(0, 0, 400, 400);**

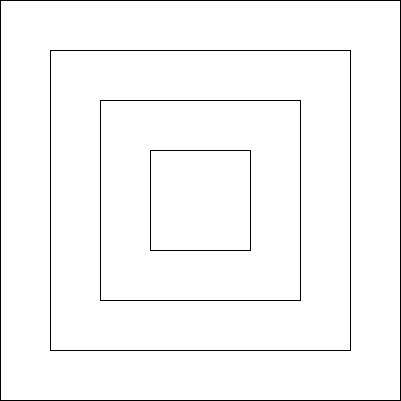
**G.drawRect(50, 50, 300, 300);**

**G.drawRect(100, 100, 200, 200);**

**G.drawRect(150, 150, 100, 100);**

**}**

**}**





**Output:**

# Java Program to Print the Graph Having X-axis and y-axis.

**import java.applet.\*; import java.awt.\*;**

**public class DrawGraph extends Applet{ public void paint(Graphics G){**

**G.drawLine(50, 50, 50, 450);**

**G.drawLine(50, 450, 450, 450);**

**G.drawString("Y", 40, 50);**

**G.drawString("X", 450, 465); for(int i=50, count=0;i<450;i+=40){**

**G.drawLine(i, 446, i, 454);**

**G.drawString(Integer.toString(count), i-5, 470); G.drawLine(46, 500-i, 54, 500-i); G.drawString(Integer.toString(count++), 35, 500-i);**

**}**

**//points ats (7,3) and (6,6) and (2,2) with red marker G.setColor(Color.RED);**

**G.fillRoundRect(327, 327, 6, 6, 6, 6);//Point(7,3)**

**G.drawString("(7,3)", 335, 330);**

**G.fillRoundRect(287, 207, 6, 6, 6, 6);//Point (6,6)**

**G.drawString("(6,6)", 295, 210);**

**G.fillRoundRect(127, 367, 6, 6, 6, 6);//Point (2,2)**

**G.drawString("(6,6)", 135, 375); G.setColor(Color.MAGENTA);**

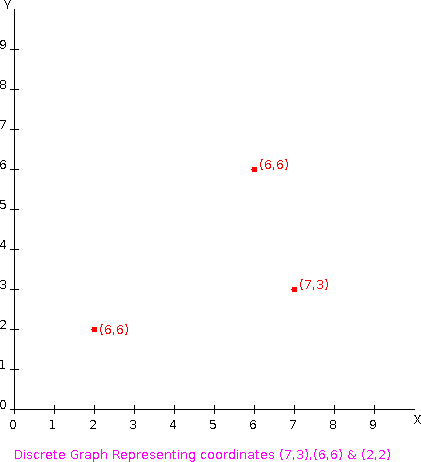
**G.drawString("Discrete Graph Representing coordinates (7,3),(6,6) & (2,2)", 50, 500);**

**}**

**}**



**Output:**



# 5-20) Java Program to Print Different Spirals.

**import java.applet.\*; import java.awt.\*;**

**public class DrawSpiral1 extends Applet { public void paint(Graphics G) {**

**drawSpiral1(G); drawSpiral2(G); drawSpiral3(G); drawSpiral4(G); drawSpiral5(G);**

**}**

**void drawSpiral1(Graphics G) { Graphics2D g2 = (Graphics2D) G; g2.translate(250, 250);**

**int x1 = 0; int y1 = 0; int x2;**

**int y2; g2.setColor(Color.GRAY);**

**g2.setStroke(new BasicStroke(20, BasicStroke.CAP\_ROUND, BasicStroke.CAP\_ROUND)); for (int i = 0; i < 2000; i++) {**

**double t = i \* Math.PI / 180;**

**x2 = (int) ((int) 5 \* t \* Math.sin(t));**

**y2 = (int) ((int) 5 \* t \* Math.cos(t)); g2.drawLine(x1, y1, x2, y2);**

**x1 = x2; y1 = y2;**

**}**

**}**

**void drawSpiral2(Graphics G){ Graphics2D g2 = (Graphics2D) G; g2.translate(400, 0);**

**int x1 = 0; int y1 = 0; int x2;**

**int y2;**

**g2.setStroke(new BasicStroke(5,BasicStroke.CAP\_ROUND,BasicStroke.CAP\_ROUND)); for (int i = 0; i < 3000; i++) {**

**double t = i \* Math.PI / 180;**

**x2 = (int) ((int) 2 \* t \* Math.sin(t));**

**y2 = (int) ((int) 2 \* t \* Math.cos(t)); g2.drawLine(x1, y1, x2, y2);**

**x1 = x2;**

**y1 = y2;**

**}**

**}**

**void drawSpiral3(Graphics G){ Graphics2D g2 = (Graphics2D) G; g2.translate(400, 0);**

**int x1 = 0; int y1 = 0; int x2;**

**int y2;**

**g2.setStroke(new BasicStroke(5,BasicStroke.CAP\_ROUND,BasicStroke.CAP\_ROUND)); for (int i = 0; i < 720; i++) {**

**double t = i \* Math.PI / 180;**

**x2 = (int) ((int) 15 \* t \* Math.cos(t));**

**y2 = (int) ((int) 15 \* t \* Math.sin(t)); g2.drawLine(x1, y1, x2, y2);**

**x1 = x2; y1 = y2;**

**}**

**}**

**void drawSpiral4(Graphics G){ Graphics2D g2 = (Graphics2D) G; g2.translate(500, 0);**

**int x1 = 0; int y1 = 0; int x2;**

**int y2;**

**g2.setStroke(new BasicStroke(20,BasicStroke.CAP\_ROUND,BasicStroke.CAP\_ROUND)); for (int i = 180; i < 1350; i++) {**

**double t = i \* Math.PI / 180;**

**x2 = (int) ((int) 10 \* t \* Math.cos(t));**

**y2 = (int) ((int) 10 \* t \* Math.sin(t));**

**x1 = (int) ((int) 10 \* t \* Math.cos(t));**

**y1 = (int) ((int) 10 \* t \* Math.sin(t)); g2.drawLine(x1, y1, x2, y2);**

**x1 = x2; y1 = y2;**

**}**

**}**

**void drawSpiral5(Graphics G){ Graphics2D g2 = (Graphics2D) G; g2.translate(-1300, 400);**

**int x1 = 0; int y1 = 0; int x2;**

**int y2;**

**g2.setStroke(new BasicStroke(5,BasicStroke.CAP\_ROUND,BasicStroke.CAP\_ROUND)); for (int i = 360; i < 2550; i++) {**

**double t = i \* Math.PI / 180;**

**x1 = (int) ((int) 3 \* t \* Math.cos(t));**

**y1 = (int) ((int) 3 \* t \* Math.sin(t));**

**x2 = (int) ((int) 3 \* t \* Math.cos(t));**

**y2 = (int) ((int) 3 \* t \* Math.sin(t)); g2.drawLine(x1, y1, x2, y2);**

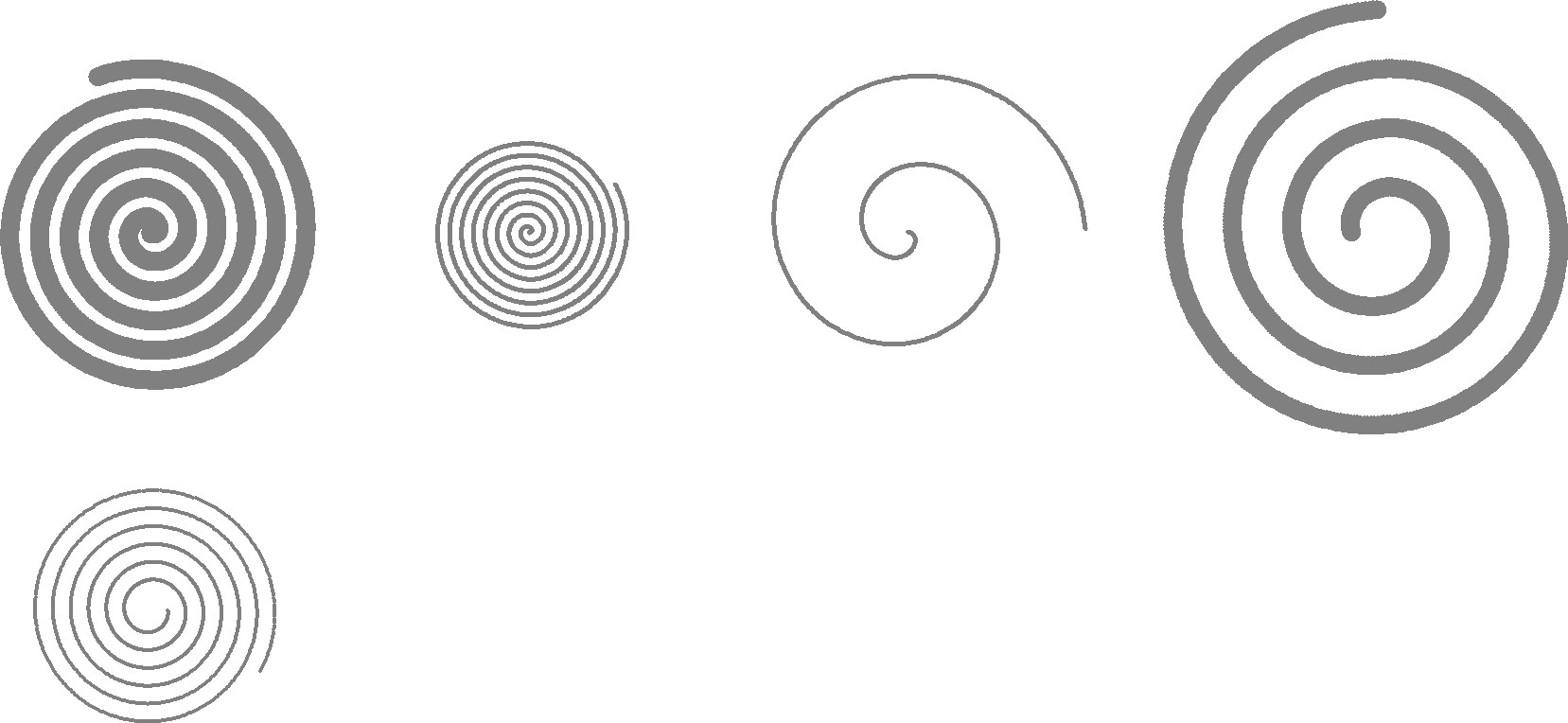
**x1 = x2; y1 = y2;**

**}**

**}**

**}**

## Output:



•Java program to print the pattern of the circle on the console

import java.awt.\*; import java.awt.event.\*; import java.awt.geom.\*;

public class DrawCircle extends Frame

{

// input the value for circle and square.

Shape circle=new Ellipse2D.Float(100.0f,100.0f,100.0f,100.0f);

// class paint to fill color in circle. public void paint(Graphics g)

{

Graphics2D ga=(Graphics2D)g; ga.draw(circle); ga.setPaint(Color.black); ga.fill(circle);

}

public static void main(String args[])

{

// create a frame object for circle. Frame frame=new DrawCircle();

frame.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

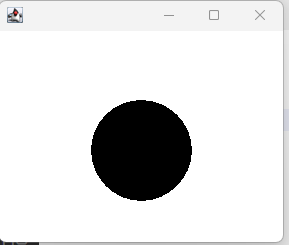
}

});

// circle coordinates. frame.setSize(300, 250); frame.setVisible(true);

}

}



•Java program to print the pattern of the triangle on the console

import javax.swing.\*; import java.awt.\*;

import java.awt.geom.Path2D; import java.awt.geom.Point2D;

public class Draw\_A\_Triangle extends JPanel { public void paintComponent(Graphics g) {

Triangle\_Shape triangleShape = new Triangle\_Shape(new Point2D.Double(50, 0),

new Point2D.Double(100, 100), new Point2D.Double(0, 100)); Graphics2D g2d = (Graphics2D) g.create(); g2d.draw(triangleShape);

}

public static void main (String [] args){ JFrame.setDefaultLookAndFeelDecorated(true); JFrame frame = new JFrame("");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setBackground(Color.white);

frame.setSize(200, 200);

Draw\_A\_Triangle panel = new Draw\_A\_Triangle(); frame.add(panel);

frame.setVisible(true);

}

}

class Triangle\_Shape extends Path2D.Double { public Triangle\_Shape(Point2D... points) {

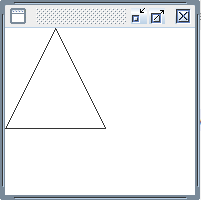
moveTo(points[0].getX(), points[0].getY());

lineTo(points[1].getX(), points[1].getY());

lineTo(points[2].getX(), points[2].getY()); closePath();

}

}



•Java program to print the pattern of the square on the console

//creating frame object

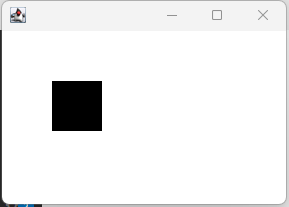
JFrame jFrame = new JFrame();

//adding graphics to the frame to display jFrame.add(graphicsDemo); jFrame.setSize(300, 300);

// f.setLayout(null); jFrame.setVisible(true);

}

}



•Java program to print the graph having an x-axis, and y-axis on the console

import java.awt.\*; import javax.swing.\*; import java.awt.geom.\*;

public class G extends JPanel{

int[] coordinates={100,20}; int mar=50;

protected void paintComponent(Graphics g){ super.paintComponent(g);

Graphics2D g1=(Graphics2D)g; g1.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,RenderingHints. VALUE\_ANTIALIAS\_ON);

int width=getWidth(); int height=getHeight();

g1.draw(new Line2D.Double(mar,mar,mar,height-mar));

g1.draw(new Line2D.Double(mar,height-mar,width-mar,height-mar)); double x=(double)(width-2\*mar)/(coordinates.length-1);

double scale=(double)(height-2\*mar)/getMax(); g1.setPaint(Color.BLUE);

for(int i=0;i<coordinates.length;i++){ double x1=mar+i\*x;

double y1=height-mar-scale\*coordinates[i]; g1.fill(new Ellipse2D.Double(x1-2,y1-2,4,4));

}

}

private int getMax(){

int max=-Integer.MAX\_VALUE; for(int i=0;i<coordinates.length;i++){

if(coordinates[i]>max) max=coordinates[i];

}return max;

}

public static void main(String args[]){ JFrame frame =new JFrame();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.add(new G());

frame.setSize(400,400); frame.setLocation(200,200); frame.setVisible(true);

}

}

•Java program to print the following pattern on the console

import java.awt.Graphics; import javax.swing.JPanel; import javax.swing.JFrame;

public class CircSpiral extends JPanel {

public void paintComponent(Graphics g) { int x = getSize().width / 2 - 10;

int y = getSize().height/ 2 - 10; int width = 20;

int height = 20; int startAngle = 0;

int arcAngle = 180; int depth = 10;

for (int i = 0; i < 17; i++) { if (i % 2 == 0) {

// g.drawArc(x + 10, y + 10, width, height, startAngle + 10, - arcAngle);

// x = x - 5; y = y - depth;

width = width + 2 \* depth; height = height + 2 \* depth;

g.drawArc(x, y, width, height, startAngle, -arcAngle);

} else {

// g.drawArc(x + 10, y + 10, width, height, startAngle + 10, arcAngle);

x = x - 2 \* depth; y = y - depth;

width = width + 2 \* depth; height = height + 2 \* depth;

g.drawArc(x, y, width, height, startAngle, arcAngle);

}

}

}

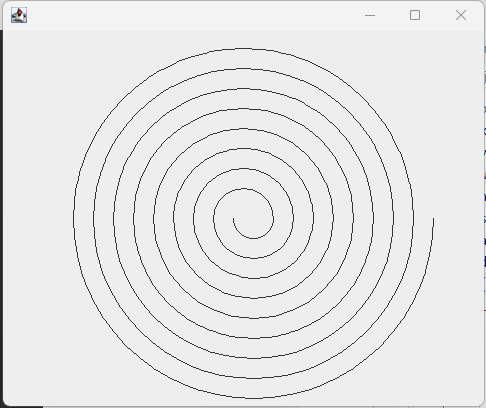
public static void main(String[] args) { CircSpiral panel = new CircSpiral(); JFrame application = new JFrame();

application.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); application.add(panel);

application.setSize(300, 300); application.setVisible(true);

}

}



•Java program to print the following pattern on the console



import java.awt.Dimension; import java.awt.Graphics; import java.awt.Graphics2D; import javax.swing.JFrame; import javax.swing.JPanel;

public class Main extends JFrame

{

private static final long serialVersionUID = 62; public static void main(String arg[])

{

JFrame frame = new Main(); frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setSize(new Dimension(400,400));

frame.pack(); frame.setVisible(true);

}

public Main()

{

super("");

JPanel panel = new DrawStuff(); panel.setPreferredSize(new Dimension(400,400)); add(panel);

}

class DrawStuff extends JPanel

{

private static final long serialVersionUID = -8; int nPoints = 720;

public DrawStuff()

{

}

public void paintComponent(Graphics g)

{

super.paintComponent(g); Graphics2D g2 = (Graphics2D)g; g2.translate(200,200);

int x1 = 0; int y1 = 0; int x2;

int y2;

for (int i = 0; i < nPoints; i++)

{

double t = i \* Math.PI / 180;

x2 = (int) ((int) 20\*t\*Math.cos(t));

y2 = (int) ((int) 20\*t\*Math.sin(t)); g2.drawLine(x1, y1, x2, y2);

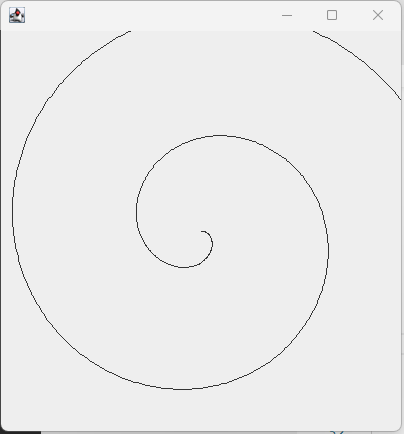
x1 = x2; y1 = y2;

}

}

}

}



•Java program to print the following pattern on the console



import java.awt.Graphics; import javax.swing.JPanel; import javax.swing.JFrame;

public class CircSpiral extends JPanel {

public void paintComponent(Graphics g) { int x = getSize().width / 2 - 10;

int y = getSize().height/ 2 - 10; int width = 20;

int height = 20; int startAngle = 0;

int arcAngle = 180; int depth = 10;

for (int i = 0; i < 11; i++) { if (i % 2 == 0) {

// g.drawArc(x + 10, y + 10, width, height, startAngle + 10, - arcAngle);

// x = x - 5; y = y - depth;

width = width + 2 \* depth; height = height + 2 \* depth;

g.drawArc(x, y, width, height, startAngle, -arcAngle);

} else {

// g.drawArc(x + 10, y + 10, width, height, startAngle + 10, arcAngle);

x = x - 2 \* depth; y = y - depth;

width = width + 2 \* depth; height = height + 2 \* depth;

g.drawArc(x, y, width, height, startAngle, arcAngle);

}

}

}

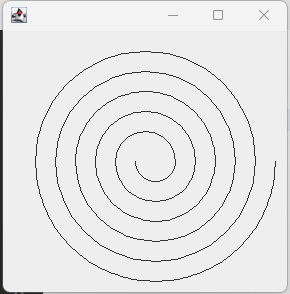
public static void main(String[] args) { CircSpiral panel = new CircSpiral(); JFrame application = new JFrame();

application.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); application.add(panel);

application.setSize(300, 300); application.setVisible(true);

}

}



•Java program to print the following pattern on the console



import java.awt.Graphics; import javax.swing.JPanel; import javax.swing.JFrame;

public class CircSpiral extends JPanel { public void paintComponent(Graphics g) {

int x = getSize().width / 2 - 10; int y = getSize().height/ 2 - 10; int width = 20;

int height = 20;

int startAngle = 0; int arcAngle = 180; int depth = 10;

for (int i = 0; i < 6; i++) { if (i % 2 == 0) {

// g.drawArc(x + 10, y + 10, width, height, startAngle + 10, - arcAngle);

// x = x - 5; y = y - depth;

width = width + 2 \* depth; height = height + 2 \* depth;

g.drawArc(x, y, width, height, startAngle, -arcAngle);

} else {

// g.drawArc(x + 10, y + 10, width, height, startAngle + 10, arcAngle);

x = x - 2 \* depth; y = y - depth;

width = width + 2 \* depth; height = height + 2 \* depth;

g.drawArc(x, y, width, height, startAngle, arcAngle);

}

}

}

public static void main(String[] args) { CircSpiral panel = new CircSpiral(); JFrame application = new JFrame();

application.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); application.add(panel);

application.setSize(300, 300); application.setVisible(true);

}

}

