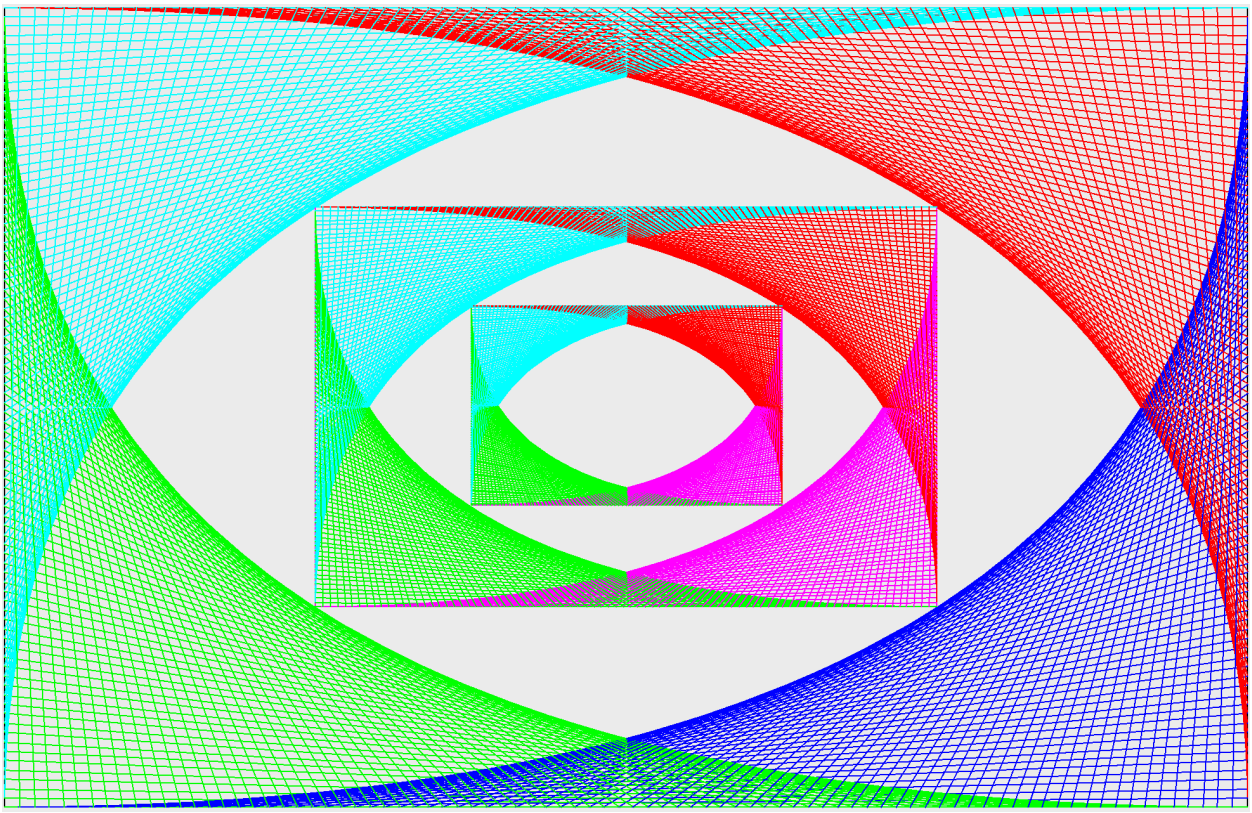

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.util.Random;
public class LineArt extends JPanel {
	// Unique version ID for this class to ensure saved objects can be loaded safely
	private static final long serialVersionUID = 1L;
	// Initial width of height of the starting rectangle
	private static int width = 980;
	private static int height = 630;
	private static int startX = 10;
	private static int startY = 10;
	private static double ratio = 980 / 255;
	// main method to launch the program as a standalone application - no need to
	// modify
	public static void main(String[] args) {
		LineArt panel = new LineArt();
		panel.setPreferredSize(new Dimension(width + 20, height + 20)); // content size window
dimensions
		JFrame frame = new JFrame("Line Art"); // Title of frame
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.add(panel);
```

```
                frame.pack();
                frame.setVisible(true);
        }
        /**
         * Draw the four corners of line art.  Line art displays straight lines inside a rectangle from one
side to
         * a perpendicular side. The lines must be drawn in such a way that both the starting points of the
lines on
         * one side and the ending points on the other side are equi-distant along the sides.  The size of
the rectangle
         * is 980 pixels wide by 630 pixels high.
         *
         * @param g the Graphics object used for drawing shapes, text, and images
         *
         */
        public void drawLineArt(Graphics g, int numLines) {

                Random random = new Random();
                int randNum = random.nextInt(4);

                // Draw the initial rectangle
                g.drawRect(startX, startY, width, height);

                for (int i = 0; i < numLines; i++) {

                        g.setColor(matchColor(randNum));
                        g.drawLine(startX + i * width / numLines, startY + height, width + startX, startY +
height - i * height / numLines);
                        g.setColor(matchColor(randNum+1));
                        g.drawLine(startX + width - i * width / numLines, startY + height, startX, startY +
height - i * height / numLines);
                        g.setColor(matchColor(randNum+2));
                        g.drawLine(startX + i * width / numLines, startY, startX + width, startY + i * height
/ numLines);
                        g.setColor(matchColor(randNum+3));
                        g.drawLine(startX + width - i * width / numLines, startY, startX, startY + i * height /
numLines);
                        g.setColor(matchColor(randNum+4));

                        }

                int newStartX = 255;
                int newStartY = 167;

                int newWidth = width/2;
                int newHeight = height/2;

                g.drawRect(newStartX, newStartY, newWidth, newHeight);
```

```
                for (int i = 0; i < numLines; i++) {

                        g.drawLine(newStartX + i * newWidth / numLines, newStartY + newHeight,
newWidth + newStartX, newStartY + newHeight - i * newHeight / numLines);
                        g.setColor(matchColor(randNum + 1));
                        g.drawLine(newStartX + newWidth - i * newWidth / numLines, newStartY +
newHeight, newStartX, newStartY + newHeight - i * newHeight / numLines);
                        g.setColor(matchColor(randNum + 2));
                        g.drawLine(newStartX + i * newWidth / numLines, newStartY, newStartX +
newWidth, newStartY + i * newHeight / numLines);
                        g.setColor(matchColor(randNum + 3));
                        g.drawLine(newStartX + newWidth - i * newWidth / numLines, newStartY,
newStartX, newStartY + i * newHeight / numLines);
                        g.setColor(matchColor(randNum + 4));

                        }
                int newStartX1 = 378;
                int newStartY1 = 245;

                int newNewWidth = newWidth / 2;
                int newNewHeight = newHeight / 2;

                for (int i = 0; i < numLines; i++) {

                        g.drawLine(newStartX1 + i * newNewWidth / numLines, newStartY1 +
newNewHeight, newNewWidth + newStartX1, newStartY1 + newNewHeight - i * newNewHeight /
numLines);
                        g.setColor(matchColor(randNum + 1));
                        g.drawLine(newStartX1 + newNewWidth - i * newNewWidth / numLines,
newStartY1 + newNewHeight, newStartX1, newStartY1 + newNewHeight - i * newNewHeight /
numLines);
                        g.setColor(matchColor(randNum + 2));
                        g.drawLine(newStartX1 + i * newNewWidth / numLines, newStartY1, newStartX1
+ newNewWidth, newStartY1 + i * newNewHeight / numLines);
                        g.setColor(matchColor(randNum + 3));
                        g.drawLine(newStartX1 + newNewWidth - i * newNewWidth / numLines,
newStartY1, newStartX1, newStartY1 + i * newNewHeight / numLines);
                        g.setColor(matchColor(randNum + 4));

                        }
        }

        public static Color matchColor(int num) {

                Color color = null;

                switch (num % 5) {
                case 0:
                        color = Color.blue;
```

```
                        break;
                case 1:
                        color = Color.green;
                        break;
                case 2:
                        color = Color.red;
                        break;
                case 3:
                        color = Color.cyan;
                        break;
                case 4:
                        color = Color.magenta;
                        break;
                }

                return color;
        }
        /**
         * Overrides JPanel's paintComponent method to perform custom drawing.
         *
         * @param g the Graphics object used for drawing shapes, text, and images
         */
        @Override
        protected void paintComponent(Graphics g) {
                super.paintComponent(g); // Clears the panel before drawing
                drawLineArt(g, 85);

        }
}
```