

## Code:

```
// LineArt.java
// Student version of the Lab06 Line Art Graphics Program assignment.

import java.awt.*;

import javax.swing.plaf.ColorUIResource;

import java.applet.*;

public class LineArt extends Applet
{

    public void paint(Graphics g)
    {
        int width = 980;
        int height = 630;
        g.drawRect(10,10,width,height);
        //Color graphColor = new Color(34, 139, 34);
        //g.setColor(graphColor);
        int spacing = 51;
        int portions_x = width / spacing;
        int portions_y = height / spacing;
        // Draw bottom-left corner
        for (int i = 0; i <= spacing; i = i + 1)
        {
            g.drawLine(10, 10 + portions_y*i, 10 + portions_x*i, 640);
        }

        // Draw bottom-right corner
        for (int i = 0; i <= spacing; i = i + 1)
        {
            g.drawLine(10 + portions_x*i, 640, 990, 640 - portions_y*i);
        }
        // Draw top-right corner
        for (int i = 0; i <= spacing; i = i + 1)
        {
            g.drawLine(10 + portions_x*i, 10, 990, 10 + portions_y*i);
        }
    }
}
```

```

// Draw top-left corner
for (int i = 0; i <= spacing; i = i + 1)
{
    g.drawLine(10, 640 - portions_y*i, 10 + portions_x*i, 10);
}

//I noticed that the corners of the inscribed rectangle lay on the
diagonals of the original rectangle, so I drew in the diagonals of the
original rectangle. Then I tested where the corners of the inscribed
rectangle should be by drawing little ovals as "points". I adjusted the
points until they were at the intersection of the "curve" and the
diagonals.
    //g.setColor(Color.RED); so that the diagonals look different
    //Drawing the main diagonal of the original rectangle

// g.drawLine(10, 10, 990, 640); (bottom left to top right)

//Drawing the alternate diagonal of the original rectangle
//g.drawLine(10, 640, 990, 10); (bottom right to top left)

//THE INSIDE CIRCLE
int r = 5; //size of the points
//finding top left point coordinates
int x_a = 257;
int y_a = 169;
//g.fillOval(x_a - r, y_a - r, r, r); the top left coordinate is (258,
172)

//finding top right point coordinates
int x_b = 749;
//g.fillOval(x_b - r, y_a - r, r, r); the top left coordinate is (749,
172)

//finding bottom right point coordinates
int y_c = 481;
//g.fillOval(x_b - r, y_c - r, r, r); the top left coordinate is (749,
482)

//Since we have 3 coordinates, the fourth should be (258, 482) because we
know it's a rectangle.
//g.fillOval(x_a, y_c, r, r);

```

```
int iter_x = (x_b - x_a) / spacing;
int iter_y = (y_c - y_a) / spacing;
g.drawRect(x_a, y_a, x_b - x_a, y_c - y_a);

//bottom left pattern
for (int i = 0; i <= spacing; i += 1)
{
    g.drawLine(x_a, y_a + iter_y*i, x_a + iter_x*i, y_c);
}

//bottom right pattern
for (int i = 0; i <= spacing; i += 1)
{
    g.drawLine(x_b, y_a + iter_y*i, x_b - iter_x*i, y_c);
}

//top left pattern
for (int i = 0; i <= spacing; i += 1)
{
    g.drawLine(x_a + iter_x*i, y_a, x_a, y_c - iter_y*i);
}

for (int i = 0; i <= spacing; i += 1)
{
    g.drawLine(x_b - iter_x*i, y_a, x_b, y_c - iter_y*i);
}
}
```

Output:

