

```
1import java.awt.Dimension;
2import java.awt.Graphics;
3import javax.swing.JPanel;
4import javax.swing.JFrame;
5
6public class LineArt extends JPanel {
7
8    // Unique version ID for this class to ensure saved
    // objects can be loaded safely
9    private static final long serialVersionUID = 1L;
10
11    // Initial width of height of the starting rectangle
12    private static int width = 980;
13    private static int height = 630;
14
15    // main method to launch the program as a standalone
    // application - no need to
16    // modify
17    public static void main(String[] args) {
18        LineArt panel = new LineArt();
19        panel.setPreferredSize(new Dimension(width + 20,
20        height + 20)); // content size window dimensions
21
22        JFrame frame = new JFrame("Line Art"); // Title
    // of frame
23
24        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25        frame.add(panel);
26        frame.pack();
27        frame.setVisible(true);
28    }
29
30    /**
31     * Draw the four corners of line art. Line art
    // displays straight lines inside a rectangle from one side
    // to
32     * a perpendicular side. The lines must be drawn in
    // such a way that both the starting points of the lines on
33     * one side and the ending points on the other side
```

```
are equi-distant along the sides. The size of the
rectangle
32     * is 980 pixels wide by 630 pixels high.
33     *
34     * @param g the Graphics object used for drawing
    shapes, text, and images
35     */
36     public void drawLineArt(Graphics g) {
37
38         // Draw the initial rectangle
39         g.drawRect(10, 10, width, height);
40
41         // Draw bottom-left corner
42
43         // Draw bottom-right corner
44
45         int x = 990;
46         int y = 20;
47
48         while (x <= 990 && x >= 20 && y <= 630 && y >= 20)
49         {
50             g.drawLine(x, 640, 990, y);
51
52             x = x - 20;
53             y = y + 13;
54
55
56         }
57
58
59         int w = 10;
60         int z = 20;
61
62         while (w <= 990 && w >= 10 && z <= 630 && z >= 20)
63         {
64             g.drawLine(w, 640, 10, z);
65
```

```
66         w = w + 20;
67         z = z + 13;
68
69
70     }
71
72     int k = 10;
73     int p = 630;
74
75     while (k <= 990 && k >= 10 && p <= 630 && p>=10)
76     {
77         g.drawLine(k, 10, 10, p);
78
79         k = k + 20;
80         p = p - 13;
81
82
83     }
84
85
86     int i = 990;
87     int m = 630;
88
89     while (i <= 990 && i >= 10 && m <= 630 && m>=10)
90     {
91         g.drawLine(i, 10, 990, m);
92
93         i = i - 20;
94         m = m - 13;
95
96
97     }
98
99     //Seperating between large and small
100
101
102     int a = 770;
```

```
103         int b = 180;
104
105         while (a <= 770 && a >= 230 && b <= 470 &&
    b>=180) {
106
107             g.drawLine(a, 470, 770, b);
108
109             a = a - 20;
110             b = b + 13;
111
112
113         }
114
115
116         int c = 230;
117         int d = 180;
118
119         while (c <= 770 && c >= 230 && d <= 470 && d >=
    180) {
120
121             g.drawLine(c, 470, 230, d);
122
123             c = c + 20;
124             d = d + 13;
125
126
127         }
128
129         int e = 230;
130         int f = 470;
131
132         while (e <= 770 && e >= 230 && f <= 470 &&
    f>=180) {
133
134             g.drawLine(e, 180, 230, f);
135
136             e = e + 20;
137             f = f - 13;
138
```

```
139
140     }
141
142
143     int j = 770;
144     int h = 470;
145
146     while (j <= 770 && j >= 230 && h <= 470 &&
h>=180) {
147
148         g.drawLine(j, 180, 770, h);
149
150         j = j - 20;
151         h = h - 13;
152
153
154     }
155
156
157
158     // Draw top-right corner
159
160     // Draw top-left corner
161
162 }
163
164 /**
165  * Overrides JPanel's paintComponent method to
perform custom drawing.
166  *
167  * @param g the Graphics object used for drawing
shapes, text, and images
168  */
169 @Override
170 protected void paintComponent(Graphics g) {
171     super.paintComponent(g); // Clears the panel
before drawing
172     drawLineArt(g);
173 }
```

LineArt.java

Monday, January 12, 2026, 8:28 AM

```
174  
175}  
176
```