

```

1 import java.awt.Dimension;
2 import java.awt.Graphics;
3 import java.awt.Color;
4 import javax.swing.JPanel;
5 import javax.swing.JFrame;
6 import java.util.Random;
7
8 public class Stars extends JPanel {
9
10     // Unique version ID for this class to ensure saved objects can be loaded safely
11     private static final long serialVersionUID = 1L;
12
13     // Initial width of height of the window
14     private static int width = 1000;
15     private static int height = 650;
16
17     // main method to launch the program as a standalone application - no need to modify
18     public static void main(String[] args) {
19         Stars panel = new Stars();
20         panel.setPreferredSize(new Dimension(width, height)); // content size window dimensions
21
22         JFrame frame = new JFrame("Stars"); // Title of frame
23         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24         frame.add(panel);
25         frame.pack();
26         frame.setVisible(true);
27     }
28
29     /**
30      * Draws 10 stars (or more) of random sizes in random locations. The stars do not have to
31      * be fully contained within the panel.
32      * @param g the Graphics object used for drawing shapes, text, and images
33      */
34     public void drawStars(Graphics g) {
35         Random myRandom = new Random();
36         int numStars = 10;
37         int i = 0;
38         for (int n = 0; n < numStars; n++) {
39             int points = myRandom.nextInt(10, 18);
40             int r1 = myRandom.nextInt(200);
41             int r2 = r1 / 2;
42             int middleX = myRandom.nextInt(2 * r2, width - 2 * r2);
43             int middleY = myRandom.nextInt(2 * r2, height - 2 * r2);
44             int[] x = new int[points];
45             int[] y = new int[points];
46             if ((middleX + r1) > width || (middleY + r1) > height || (middleX - r1) < 0 || (middleY - r1) < 0) {
47                 n--;
48                 continue;
49             }
50             for (i = 0; i < points; i++) {
51                 int vertexX = 0;
52                 int vertexY = 0;
53                 int cornerX = 0;
54                 int cornerY = 0;
55                 if (i % 2 == 0) {
56                     x[i] = middleX + r2 + (int) (r2 / 2 * Math.cos(i * Math.PI / (points / 2)));

```

```
56         y[i] = middleY + r2 + (int) (r2 / 2 * Math.sin(i * Math.PI / (points/2)));
57     }
58     else {
59         x[i] = middleX + r2 + (int) (r1 * Math.cos(i * Math.PI / (points/2)));
60         y[i] = middleY + r2 + (int) (r1 * Math.sin(i * Math.PI / (points/2)));
61     }
62 }
63 i = 0;
64 Color starColor = new Color(myRandom.nextInt(255), myRandom.nextInt(255),
myRandom.nextInt(255));
65 g.setColor(starColor);
66 g.fillPolygon(x, y, points);
67 Color black = new Color(0,0,0);
68 g.setColor(black);
69 g.drawPolygon(x, y, points);
70 }
71 }
72
73 /**
74  * Overrides JPanel's paintComponent method to perform custom drawing.
75  * @param g the Graphics object used for drawing shapes, text, and images
76  */
77 @Override
78 protected void paintComponent(Graphics g) {
79     super.paintComponent(g); // Clears the panel before drawing
80     drawStars(g);
81 }
82
83 }
84
```