**Experiment Report - 10 - test4\_RadarChart**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | Line 8 | Semantic | √ | × | √ |
| E02 | Line 34 | Semantic | × | × | × |
| E03 | Line 55 | Syntax | √ | √ | √ |
| E04 | Line 69 | Logic | √ | × | × |
| E05 | Line 81 | Syntax | × | × | × |

Additional Errors Found by Self: 1

Self-Review Detection Rate: 60%

Peer 1 Detection Rate: 20%

Peer 2 Detection Rate: 40%

1. **Source Code**
2. import java.awt.\*;
3. import javax.swing.\*;
4. public class RadarChart extends JPanel {
5. private int[] values; // レーダーチャートの値
6. private String[] labels; // 各頂点のラベル
7. private int maxValue = 70; // 値の最大値（0からこの値までの範囲で指定）
8. public RadarChart(int[] values, String[] labels) {
9. this.values = values;
10. this.labels = labels;
11. setPreferredSize(new Dimension(400, 400)); // パネルのサイズを設定
12. }
13. @Override
14. protected void paintComponent(Graphics g) {
15. super.paintComponent(g);
16. Graphics2D g2d = (Graphics2D) g;
17. int centerX = getWidth() / 2; // 中心のX座標
18. int centerY = getHeight() / 2; // 中心のY座標
19. int numPoints = values.length; // 頂点の数
20. double angleStep = 2 \* Math.PI / numPoints; // 各頂点の角度間隔
21. // ポリゴンを描画するためのポイント配列
22. int[] xPoints = new int[numPoints];
23. int[] yPoints = new int[numPoints];
24. // 各頂点の位置を計算してポイント配列に格納
25. for (int i = 0; i < numPoints; i++) {
26. double angle = i \* angleStep - Math.PI / 2; // -90度（上方向）から開始
27. double radius = (double) values[i] / maxValue \* centerY; // 値に応じた半径
28. // 極座標を計算して、画面座標に変換
29. int x = centerX + (int) (Math.cos(angle) \* radius);
30. int y = centerY + (int) (Math.sin(angle) \* radius);
31. xPoints[i] = x;
32. yPoints[i] = y;
33. // 中心から各頂点に向けて線分を描画
34. g2d.setColor(Color.RED); // 線の色を赤に設定
35. g2d.drawLine(centerX, centerY, x, y);
36. }
37. // レーダーチャートの外接円を描画
38. g2d.setColor(Color.BLACK); // 外接円の色を黒に設定
39. int maxRadius = centerX; // 外接円の半径は中心からの距離
40. g2d.drawOval(centerX - maxRadius, centerY - maxRadius, 2 \* maxRadius, 2 \* maxRadius);
41. // ポリゴンを描画（塗りつぶし）
42. g2d.setColor(Color.BLUE); // ポリゴンの色を青に設定
43. g2d.fillPolygon(xPoints, yPoints);
44. // 各頂点にラベルを描画
45. g2d.setColor(Color.BLACK); // ラベルの色を黒に設定
46. FontMetrics fm = g2d.getFontMetrics();
47. for (int i = 0; i < numPoints; i++) {
48. double angle = i \* angleStep - Math.PI / 2;
49. int labelX = centerX + (int) ((Math.cos(angle) \* 1.1) \* centerX);
50. int labelY = centerY + (int) ((Math.sin(angle) \* 1.1) \* centerY);
51. String label = labels[i];
52. int labelWidth = fm.stringWidth(label);
53. int labelHeight = fm.getHeight();
54. g2d.drawString(label, labelX + labelWidth / 2, labelY + labelHeight / 2);
55. }
56. }
57. public static void main(String[] args) {
58. int[] values = {60, 80, 50, 70, 90}; // 各頂点の値（0から80までの値）
59. String[] labels = {"A", "B", "C", "D", "E"}; // 各頂点のラベル
60. JFrame frame = new JFrame("Radar Chart");
61. RadarChart radarChart = new RadarChart(values, labels);
62. frame.add(radarChart);
63. frame.pack();
64. frame.setDefaultCloseOperation(EXIT\_ON\_CLOSE);
65. frame.setVisible(true);
66. }
67. }