**Experiment Report - 12 – test3\_Page8**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | CSCR? | Self-Detected? | LLM? |
| E01 | Line 18 | Syntax | √ |  | √ |
| E02 | Line 54 | Logic | √ |  | √ |
| E03 | Line 91 | Semantic | √ |  | × |
| E04 | Line 38 (131) | Semantic |  | √ | √ |
| E05 | Line 62 (155) | Logic |  | × | × |
| E06 | Line 65 (158) | Semantic |  | √ | × |

Additional Errors Found by Self: 0

CSCR Rate: 100%

Self-Review Detection Rate: 67%

LLM Rate: 50%

1. **Source Code**
2. import javax.swing.\*;
3. import java.awt.event.ActionEvent;
4. import java.awt.event.ActionListener;
5. import java.io.\*;
6. import java.nio.file.\*;
7. import java.util.\*;
8. import java.awt.\*;
9. import java.util.List;
10. public class Page8 extends JFrame {
11. private List<Map.Entry<String[], Integer>> topMatches;
12. private Map.Entry<String[], Integer> entry;
13. private String Name;
14. private String Pass;
15. private String userFile = "udata.csv";
16. public static void main(String args[]) {
17. Page8 frame = new Page8("Page8", null, null, null);
18. frame.setVisible(true);
19. }
20. Page8(String title, Map.Entry<String[], Integer> entry, List<Map.Entry<String[], Integer>> topMatches, String Name,
21. String Pass) {
22. setTitle(title);
23. setSize(1500, 900);
24. setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
25. this.entry = entry;
26. this.topMatches = topMatches;
27. this.Name = Name;
28. this.Pass = Pass;
29. JPanel p1 = new JPanel();
30. JPanel p2 = new JPanel();
31. JPanel p3 = new JPanel();
32. JPanel p4 = new JPanel();
33. if (entry != null) {
34. String name = entry.getKey()[0];
35. String auther = entry.getKey()[1];
36. String volumes = entry.getKey()[2];
37. String company = entry.getKey()[3];
38. String imagePath = entry.getKey()[4];
39. String summary = entry.getKey()[5];
40. String genre = entry.getKey()[6];
41. JPanel mangaPanel = new JPanel();
42. mangaPanel.setLayout(new GridBagLayout());
43. GridBagConstraints c = new GridBagConstraints();
44. JLabel nameLabel = new JLabel(name);
45. nameLabel.setFont(new Font(Font.DIALOG\_INPUT, Font.PLAIN, 50));
46. p1.add(nameLabel);
47. ImageIcon imageIcon = new ImageIcon(image);
48. Image image = imageIcon.getImage().getScaledInstance(400, 600, Image.SCALE\_SMOOTH);
49. JLabel imageLabel = new JLabel(new ImageIcon(image));
50. c.gridx = 0;
51. c.gridy = 0;
52. c.gridheight = 3;
53. c.gridwidth = 1;
54. c.weightx = 1.0d;
55. c.weighty = 1.0d;
56. mangaPanel.add(imageLabel, c);
57. JLabel genreLabel = new JLabel("ジャンル：" + genre);
58. genreLabel.setFont(new Font(Font.DIALOG\_INPUT, Font.PLAIN, 20));
59. c.gridx = 1;
60. c.gridy = 0;
61. c.gridheight = 1;
62. c.gridwidth = 1;
63. c.weightx = 1.0d;
64. c.weighty = 1.0d;
65. mangaPanel.add(genreLabel, c);
66. JLabel summaryLabel = new JLabel("<html><body style='width: 300px'>" + summary + "</body></html>");
67. summaryLabel.setFont(new Font("Serif", Font.PLAIN, 20));
68. c.gridx = 1;
69. c.gridy = 1;
70. c.gridheight = 2;
71. c.gridwidth = 1;
72. c.weightx = 1.0d;
73. c.weighty = 1.0d;
74. c.anchor = GridBagConstraints.NORTH;
75. mangaPanel.add(summaryLabel, c);
76. JLabel infoLabel = new JLabel("<html>" + volumes + "巻<br/>" +
77. "作者：" + auther + "<br/>" + "連載：" + company + "</html>");
78. infoLabel.setFont(new Font(Font.DIALOG\_INPUT, Font.PLAIN, 20));
79. c.gridx = 1;
80. c.gridy = 2;
81. c.gridheight = 1;
82. c.gridwidth = 1;
83. c.weightx = 1.0d;
84. c.weighty = 1.0d;
85. mangaPanel.add(infoLabel, c);
86. p2.add(mangaPanel);
87. } else {
88. JLabel noDataLabel = new JLabel("表示するマンガがありません。");
89. noDataLabel.setFont(new Font(Font.DIALOG\_INPUT, Font.PLAIN, 20));
90. p2.add(noDataLabel);
91. }
92. ImageIcon origin = new ImageIcon("./bar.png");
93. Image original = origin.getImage();
94. Image resize = original.getScaledInstance(100, 100, Image.SCALE\_SMOOTH);
95. ImageIcon resizedback = new ImageIcon(resize);
96. JButton back = new JButton(resizedback);
97. back.setPreferredSize(new Dimension(resizedback.getIconWidth(), resizedback.getIconHeight()));
98. back.addActionListener(new ActionListener() {
99. public void actionPerformed(ActionEvent e) {
100. Page7 frame = new Page7("Page7", topMatches, Name, Pass);
101. frame.setVisible(true);
102. }
103. });
104. p3.add(back);
105. JButton home = new JButton("ホーム");
106. home.addActionListener(new ActionListener() {
107. public void actionPerformed(ActionEvent e) {
108. Page4 frame = new Page4("Page4", Name, Pass);
109. frame.setVisible(true);
110. }
111. });
112. p4.add(home);
113. Container contentPane = getContentPane();
114. contentPane.add(p1, BorderLayout.SOUTH);
115. contentPane.add(p2, BorderLayout.CENTER);
116. contentPane.add(p3, BorderLayout.WEST);
117. contentPane.add(p4, BorderLayout.EAST);
118. }
119. private void addFavorite(String mangaName) {
120. *// メモリ上にファイルの内容を読み込む*
121. List<String> lines = new ArrayList<>();
122. try (BufferedReader reader = new BufferedReader(new FileReader(userFile))) {
123. String line;
124. while ((line = reader.readLine()) != null) {
125. lines.add(line);
126. }
127. } catch (IOException e) {
128. e.printStackTrace();
129. }
130. *// ファイル内容の更新*
131. try (PrintWriter writer = new PrintWriter(new FileWriter(userFile))) {
132. for (String line : lines) {
133. String[] columns = line.split(",");
134. if (columns[0].equals(Name) && columns[1].equals(Pass)) {
135. *// favorite コラムが存在するか確認*
136. if (columns.length <= 8) {
137. line += "," + mangaName;
138. } else {
139. columns[8] += "・" + mangaName;
140. line = String.join(",", columns);
141. }
142. }
143. writer.println(line);
144. }
145. } catch (IOException e) {
146. e.printStackTrace();
147. }
148. }
149. }