**Experiment Report - 20 - test5\_Page8**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | Line 38 | Semantic | √ | × | × |
| E02 | Line 62 | Logic | × | × | √ |
| E03 | Line 65 | Semantic | √ | √ | √ |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 67%

Peer 1 Detection Rate: 33%

Peer 2 Detection Rate: 67%

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, 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. Container contentPane = getContentPane();
34. contentPane.add(p1, BorderLayout.SOUTH);
35. contentPane.add(p2, BorderLayout.CENTER);
36. contentPane.add(p3, BorderLayout.WEST);
37. contentPane.add(p4, BorderLayout.EAST);
38. }
39. private void addFavorite(String mangaName) {
40. // メモリ上にファイルの内容を読み込む
41. List<String> lines = new ArrayList<>();
42. try (BufferedReader reader = new BufferedReader(new FileReader(userFile))) {
43. String line;
44. while ((line = reader.readLine()) != null) {
45. lines.add(line);
46. }
47. } catch (IOException e) {
48. e.printStackTrace();
49. }
50. // ファイル内容の更新
51. try (PrintWriter writer = new PrintWriter(new FileWriter(userFile))) {
52. for (String line : lines) {
53. String[] columns = line.split(",");
54. if (columns[0].equals(Name) && columns[1].equals(Pass)) {
55. // favorite コラムが存在するか確認
56. if (columns.length <= 8) {
57. line += "," + mangaName;
58. } else {
59. columns[8] += "・" + mangaName;
60. line = String.join(",", columns);
61. }
62. }
63. writer.println(line);
64. }
65. } catch (IOException e) {
66. e.printStackTrace();
67. }
68. }
69. }