**Experiment Report - 23 - test8\_Page6**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | Line 39 | Semantic | × | × | × |
| E02 | Line 63 | Semantic | × | × | × |
| E03 | Line 75 | Syntax | √ | √ | √ |
| E04 | Line 103 | Semantic | √ | × | √ |
| E05 | Line 105 | Semantic | √ | × | × |

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 java.awt.event.ActionEvent;
4. import java.awt.event.ActionListener;
5. import java.io.BufferedReader;
6. import java.io.FileReader;
7. import java.io.IOException;
8. import java.util.ArrayList;
9. import java.util.Arrays;
10. import java.util.List;
11. import javax.swing.\*;
12. /\*グループ初期ログイン \*/
13. public class Page6 extends JFrame {
14. public static void main(String args[]) {
15. Page6 frame = new Page6("Page6");
16. frame.setVisible(true);
17. }
18. String gp = "group.csv";
19. String line = "";
20. String Split = ",";
21. List<List<String>> data = new ArrayList<>();
22. private String ID;
23. private String Pass;
24. Page6(String title) {
25. setTitle(title);
26. setSize(1500, 900);
27. setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
28. JPanel p1 = new JPanel();
29. JPanel p2 = new JPanel();
30. JPanel p3 = new JPanel();
31. JTextField ID = new JTextField(15);
32. JLabel label1 = new JLabel("ID");
33. JPasswordField pass = new JPasswordField(15);
34. JLabel label2 = new JLabel("パスワード");
35. JButton check = new JButton("確認");
36. check.addActionListener(new ActionListener() {
37. public void actionPerformed(ActionEvent e) {
38. try (BufferedReader br = new BufferedReader(new FileReader(gp))) {
39. ID = id.getText();
40. char[] password = pass.getPassword();
41. Pass = String.valueOf(password);
42. while ((line = br.readLine()) != null) {
43. // CSVファイルをカンマで分割
44. String[] values = line.split(Split);
45. // 配列をリストに変換して追加
46. data.add(Arrays.asList(values));
47. }
48. boolean found = false;
49. for (List<String> record : data) {
50. if (record.size() >= 3) { // 行に十分なデータがあることを確認
51. String ID = record.get(1);
52. String pass = record.get(2);
53. // 比較
54. if (id.equals(ID) && pass.equals(Pass)) {
55. found = true;
56. break;
57. }
58. }
59. }
60. System.out.println(ID + Pass);
61. if (found == true) {
62. Page8 frame = new Page8(ID, Pass);
63. frame.setVisible(true);
64. dispose();
65. }
66. } catch (IOException ee) {
67. JOptionPane.showMessageDialog(null, "違います");
68. ee.printStackTrace();
69. }
70. }
71. });
72. ImageIcon origin = new ImageIcon("./bar.png");
73. Image original = origin.getImage();
74. Image resize = original.getScaledInstance(100, 100, Image.SCALE\_SMOOTH);
75. ImageIcon resizedback = new ImageIcon(resize);
76. JButton back = new JButton(resizedback);
77. back.setPreferredSize(new Dimension(resizedback.getIconWidth(), resizedback.getIconHeight()));
78. back.addActionListener(new ActionListener() {
79. public void actionPerformed(ActionEvent e) {
80. Page4 frame = new Page4(title);
81. frame.setVisible(true);
82. dispose();
83. }
84. });
85. p1.add(id);
86. p1.add(label0);
87. p1.add(pass);
88. p1.add(label1);
89. p2.add(back);
90. p3.add(check);
91. Container contentPane = getContentPane();
92. contentPane.add(p1, BorderLayout.CENTER);
93. contentPane.add(p2, BorderLayout.WEST);
94. contentPane.add(p3, BorderLayout.EAST);
95. }
96. }