**Experiment Report - 01 - test1\_page2**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | Line 23 | Semantic | √ | × | × |
| E02 | Line 32 | Semantic | √ | × | × |
| E03 | Line 42 | Semantic | √ | √ | × |
| E04 | Line 50 | Syntax | √ | √ | √ |
| E05 | Line 61 | Logic | √ | × | √ |

Additional Errors Found by Self: 1

Self-Review Detection Rate: 100%

Peer 1 Detection Rate: 40%

Peer 2 Detection Rate: 40%

1. **Source Code**
2. import javax.swing.\*;
3. import java.awt.\*;
4. import java.awt.event.\*;
5. import java.time.LocalDateTime;
6. public class page2 extends JPanel {
7. private JTextField amountTextField;
8. private JRadioButton pointUseButton, pointNotUseButton;
9. private ButtonGroup buttonGroup;
10. private JButton paymentButton;
11. private boolean usePoint;
12. public static final Integer[] COUPON\_RATES = {0, 10, 20, 30, 40, 50};
13. private JComboBox<Integer> couponComboBox;
14. private JLabel discountedAmountLabel;
15. public page2() {
16. // レイアウトの設定
17. setLayout(new FlowLayout()); // レイアウトをFlowLayoutに変更
19. // 支払金額入力欄
20. JLabel amountLabel = new JLabel("支払金額:");
21. add(amountLabel);
22. amountTextField = new JTextField(5); // 幅を指定
23. add(amountTextField);
24. add(new JLabel("円"));
25. // ポイント使用ラジオボタン
26. pointUseButton = new JRadioButton("ポイントを使用する");
27. buttonGroup = new ButtonGroup();
28. buttonGroup.add(pointUseButton);
29. pointNotUseButton = new JRadioButton("使用しない");
30. buttonGroup.add(pointUseButton);
31. JPanel radioPanel = new JPanel();
32. radioPanel.add(pointUseButton);
33. radioPanel.add(pointNotUseButton);
34. add(radioPanel);
35. // クーポン選択
36. JLabel couponLabel = new JLabel("クーポンを選ぶ");
37. add(couponLabel);
38. couponComboBox = new JComboBox<>(COUPON\_RATES);
39. couponComboBox.setSelectedIndex(1);
40. add(couponComboBox);
41. add(new JLabel("%"));
42. // 決済ボタン
43. paymentButton = new JButton("決済");
44. paymentButton.addActionListener(new ActionListener() {
45. @Override
46. public void actionPerformed(ActionEvent) {
47. int paymentAmount = Integer.parseInt(amountTextField.getText());
48. usePoint = pointUseButton.isSelected();
49. // クーポン割引率を取得
50. int couponRate = (int) couponComboBox.getSelectedItem();
51. // 決済金額を計算 (クーポン割引を適用)
52. int discountedAmount = paymentAmount - (paymentAmount \* couponRate / 100);
53. // 残高チェック (割引後の金額でチェック)
54. if (discountedAmount >= UserData.getBalance()) {
55. JOptionPane.showMessageDialog(null, "残高が不足しています。");
56. return;
57. }
59. // 割引後の金額を表示
60. discountedAmountLabel.setText("割引後金額: " + discountedAmount + "円");
61. }
62. });
63. add(paymentButton);
64. // 割引後の金額を表示するラベル
65. discountedAmountLabel = new JLabel();
66. add(discountedAmountLabel);
67. }
68. }