Abdulhaq Zulfiqar

Shahmeer

Usman Asif

SE-G

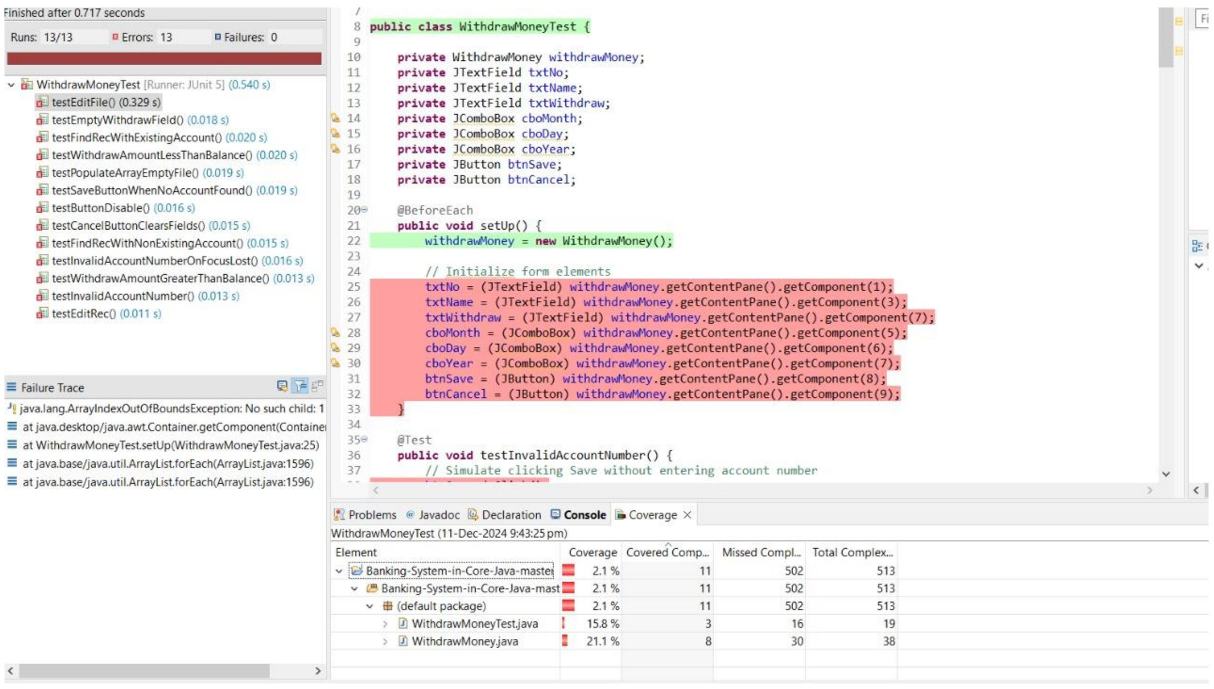
SQE

Saba Kanwal

WHITE BOX TESTING

Contents

WithdrawMoneyTest:



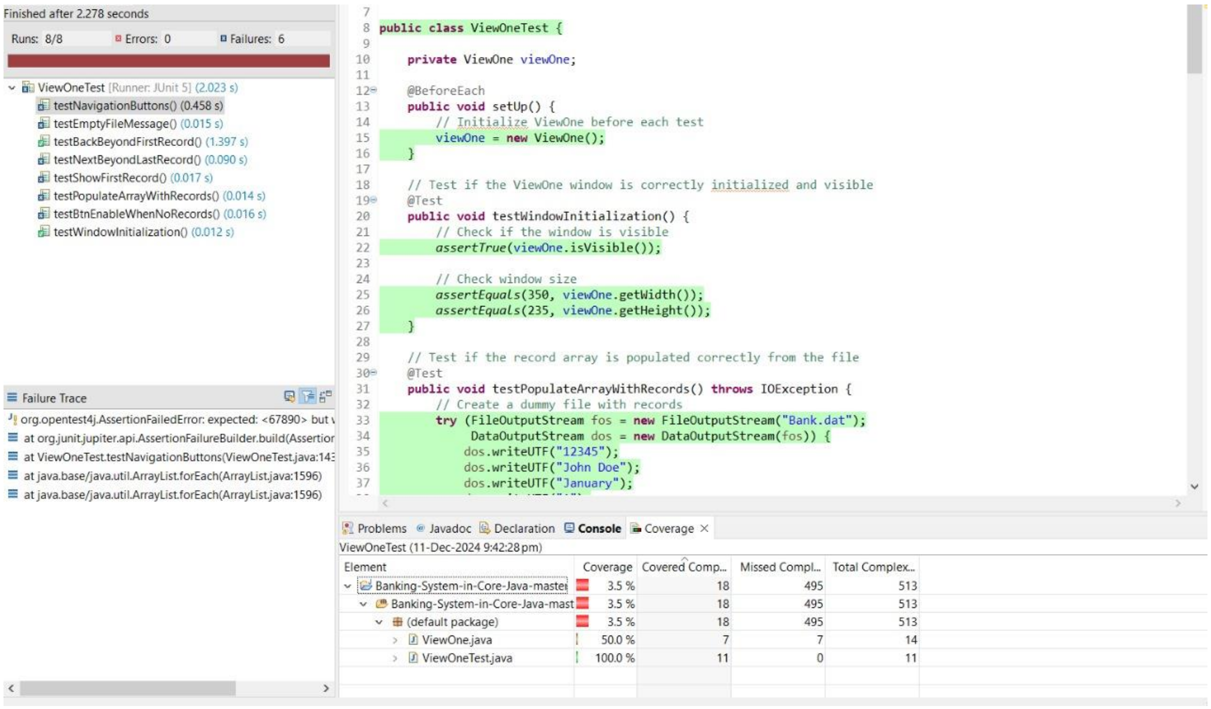
```
Finished after 0.717 seconds
Runs: 13/13     Errors: 13     Failures: 0

WithdrawMoneyTest [Runner: JUnit 5] (0.540 s)
    testEditFile() (0.329 s)
    testEmptyWithdrawField() (0.018 s)
    testFindRecWithExistingAccount() (0.020 s)
    testWithdrawAmountLessThanBalance() (0.020 s)
    testPopulateArrayEmptyFile() (0.019 s)
    testSaveButtonWhenNoAccountFound() (0.019 s)
    testButtonDisable() (0.016 s)
    testCancelButtonClearsFields() (0.015 s)
    testFindRecWithNonExistingAccount() (0.015 s)
    testInvalidAccountNumberOnFocusLost() (0.016 s)
    testWithdrawAmountGreaterThanBalance() (0.013 s)
    testInvalidAccountNumber() (0.013 s)
    testEditRec() (0.011 s)

Failure Trace
java.lang.ArrayIndexOutOfBoundsException: No such child: 1
at java.desktop/java.awt.Container.getComponent(Containe
at WithdrawMoneyTest.setUp(WithdrawMoneyTest.java:25)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```

```java
public class WithdrawMoneyTest {

    private WithdrawMoney withdrawMoney;
    private JTextField txtNo;
    private JTextField txtName;
    private JTextField txtWithdraw;
    private JComboBox cboMonth;
    private JComboBox cboDay;
    private JComboBox cboYear;
    private JButton btnSave;
    private JButton btnCancel;

    @BeforeEach
    public void setUp() {
        withdrawMoney = new WithdrawMoney();

        // Initialize form elements
        txtNo = (JTextField) withdrawMoney.getContentPane().getComponent(1);
        txtName = (JTextField) withdrawMoney.getContentPane().getComponent(3);
        txtWithdraw = (JTextField) withdrawMoney.getContentPane().getComponent(7);
        cboMonth = (JComboBox) withdrawMoney.getContentPane().getComponent(5);
        cboDay = (JComboBox) withdrawMoney.getContentPane().getComponent(6);
        cboYear = (JComboBox) withdrawMoney.getContentPane().getComponent(7);
        btnSave = (JButton) withdrawMoney.getContentPane().getComponent(8);
        btnCancel = (JButton) withdrawMoney.getContentPane().getComponent(9);
    }

    @Test
    public void testInvalidAccountNumber() {
        // Simulate clicking Save without entering account number
```

WithdrawMoneyTest (11-Dec-2024 9:43:25 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-maste	2.1 %	11	502	513
Banking-System-in-Core-Java-mast	2.1 %	11	502	513
(default package)	2.1 %	11	502	513
WithdrawMoneyTest.java	15.8 %	3	16	19
WithdrawMoney.java	21.1 %	8	30	38

ViewOneTest :



```
Finished after 2.278 seconds
Runs: 8/8    Errors: 0    Failures: 6

ViewOneTest [Runner: JUnit 5] (2.023 s)
    testNavigationButtons() (0.458 s)
    testEmptyFileMessage() (0.015 s)
    testBackBeyondFirstRecord() (1.397 s)
    testNextBeyondLastRecord() (0.090 s)
    testShowFirstRecord() (0.017 s)
    testPopulateArrayWithRecords() (0.014 s)
    testBtnEnableWhenNoRecords() (0.016 s)
    testWindowInitialization() (0.012 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <67890> but v
at org.junit.jupiter.api.AssertionFailureBuilder.build(Assertior
at ViewOneTest.testNavigationButtons(ViewOneTest.java:14
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```

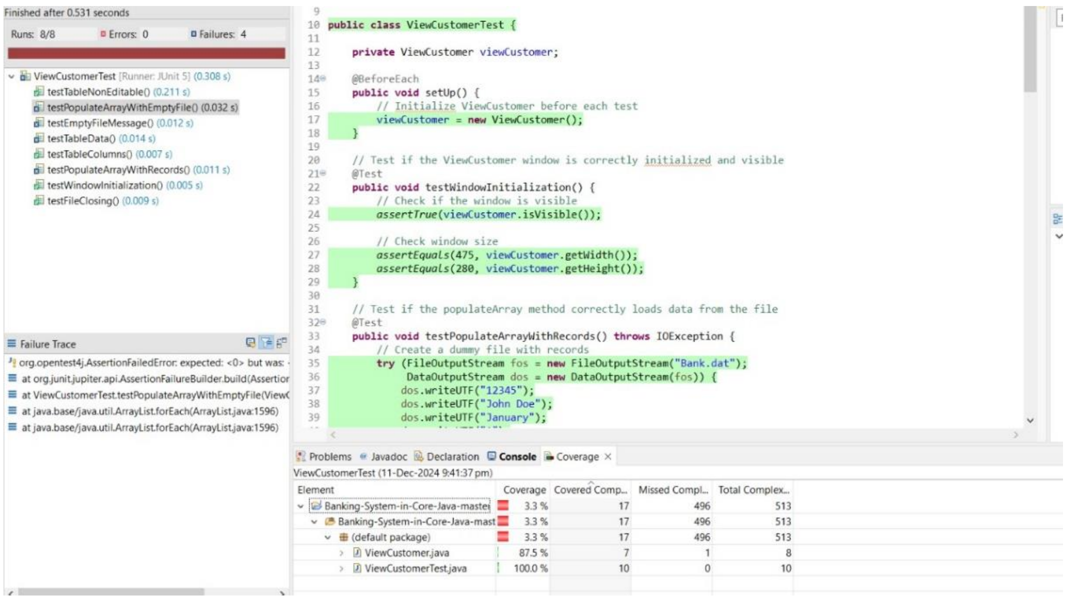
```java

public class ViewOneTest {

    private ViewOne viewOne;

    @BeforeEach
    public void setUp() {
        // Initialize ViewOne before each test
        viewOne = new ViewOne();
    }

    // Test if the ViewOne window is correctly initialized and visible
    @Test
    public void testWindowInitialization() {
        // Check if the window is visible
        assertTrue(viewOne.isVisible());

        // Check window size
        assertEquals(350, viewOne.getWidth());
        assertEquals(235, viewOne.getHeight());
    }

    // Test if the record array is populated correctly from the file
    @Test
    public void testPopulateArrayWithRecords() throws IOException {
        // Create a dummy file with records
        try (FileOutputStream fos = new FileOutputStream("Bank.dat");
             DataOutputStream dos = new DataOutputStream(fos)) {
            dos.writeUTF("12345");
            dos.writeUTF("John Doe");
            dos.writeUTF("January");
```

Problems Javadoc Declaration Console Coverage

ViewOneTest (11-Dec-2024 9:42:28 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	3.5 %	18	495	513
Banking-System-in-Core-Java-mast	3.5 %	18	495	513
(default package)	3.5 %	18	495	513
ViewOne.java	50.0 %	7	7	14
ViewOneTest.java	100.0 %	11	0	11

ViewCustomerTest:



```
Finished after 0.531 seconds
Runs: 8/8    Errors: 0    Failures: 4

ViewCustomerTest [Runner: JUnit 5] (0.308 s)
    testTableNonEditable() (0.211 s)
    testPopulateArrayWithEmptyFile() (0.032 s)
    testEmptyFileMessage() (0.012 s)
    testTableData() (0.014 s)
    testTableColumns() (0.007 s)
    testPopulateArrayWithRecords() (0.011 s)
    testWindowInitialization() (0.005 s)
    testFileClosing() (0.009 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <0> but was:
at org.junit.jupiter.api.AssertionFailureBuilder.build(Assertior
at ViewCustomerTest.testPopulateArrayWithEmptyFile(View
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```

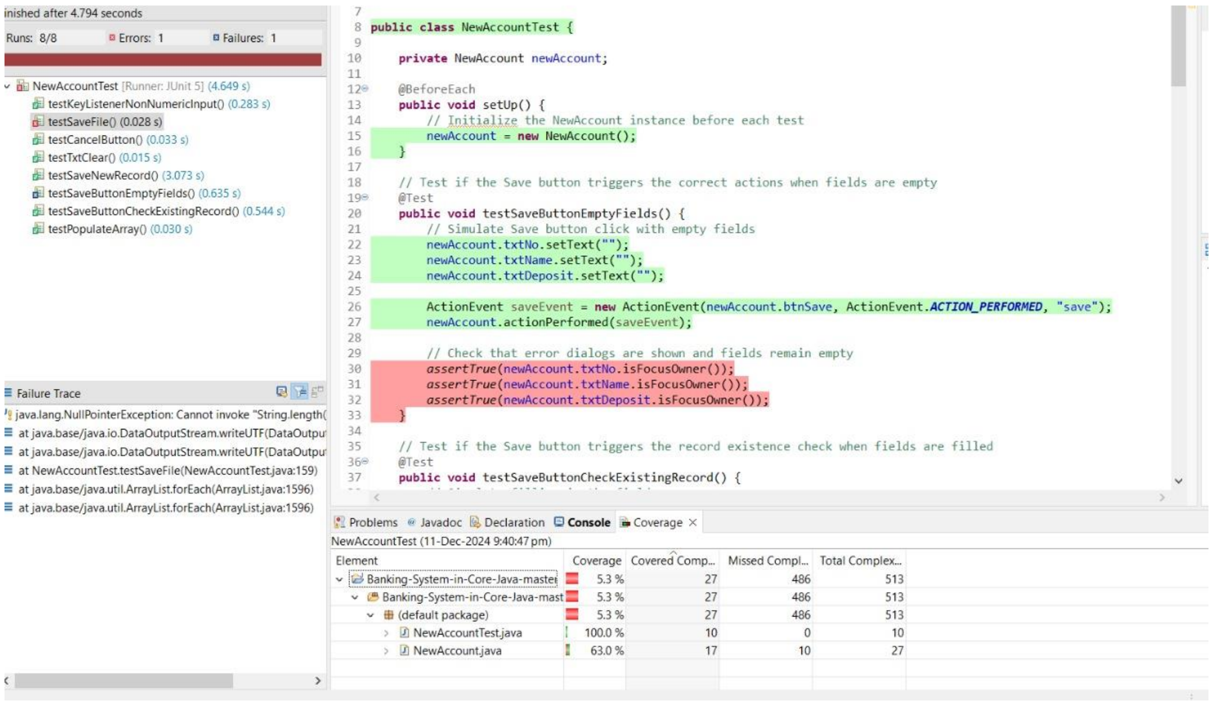
```java

public class ViewCustomerTest {

    private ViewCustomer viewCustomer;

    @BeforeEach
    public void setUp() {
        // Initialize ViewCustomer before each test
        viewCustomer = new ViewCustomer();
    }

    // Test if the ViewCustomer window is correctly initialized and visible
    @Test
    public void testWindowInitialization() {
        // Check if the window is visible
        assertTrue(viewCustomer.isVisible());

        // Check window size
        assertEquals(475, viewCustomer.getWidth());
        assertEquals(280, viewCustomer.getHeight());
    }

    // Test if the populateArray method correctly loads data from the file
    @Test
    public void testPopulateArrayWithRecords() throws IOException {
        // Create a dummy file with records
        try (FileOutputStream fos = new FileOutputStream("Bank.dat");
             DataOutputStream dos = new DataOutputStream(fos)) {
            dos.writeUTF("12345");
            dos.writeUTF("John Doe");
            dos.writeUTF("January");
```

Problems Javadoc Declaration Console Coverage

ViewCustomerTest (11-Dec-2024 9:41:37 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	3.3 %	17	496	513
Banking-System-in-Core-Java-mast	3.3 %	17	496	513
(default package)	3.3 %	17	496	513
ViewCustomer.java	87.5 %	7	1	8
ViewCustomerTest.java	100.0 %	10	0	10

NewAccountTest:



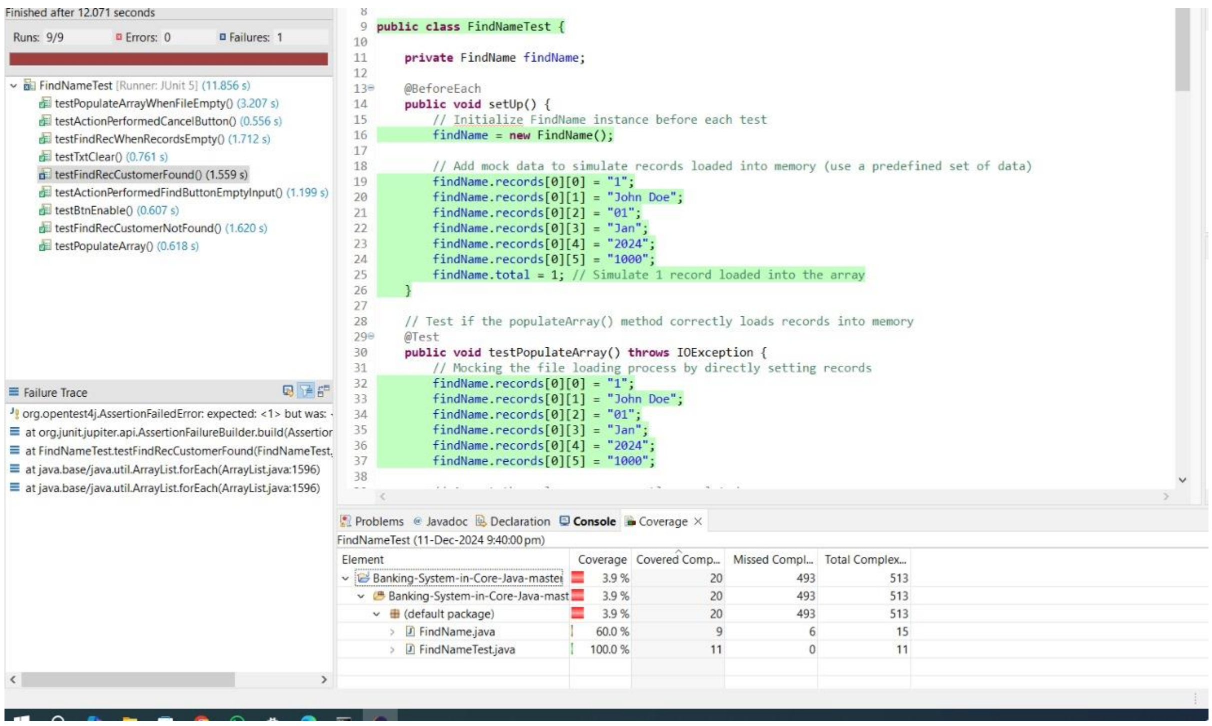
```
inished after 4.794 seconds
Runs: 8/8    Errors: 1    Failures: 1

NewAccountTest [Runner: JUnit 5] (4.649 s)
    testKeyListenerNonNumericInput() (0.283 s)
    testSaveFile() (0.028 s)
    testCancelButton() (0.033 s)
    testTxtClear() (0.015 s)
    testSaveNewRecord() (3.073 s)
    testSaveButtonEmptyFields() (0.635 s)
    testSaveButtonCheckExistingRecord() (0.544 s)
    testPopulateArray() (0.030 s)

Failure Trace
java.lang.NullPointerException: Cannot invoke "String.length(
at java.base/java.io.DataOutputStream.writeUTF(DataOutpu
at java.base/java.io.DataOutputStream.writeUTF(DataOutpu
at NewAccountTest.testSaveFile(NewAccountTest.java:159)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```

```java

public class NewAccountTest {

    private NewAccount newAccount;

    @BeforeEach
    public void setUp() {
        // Initialize the NewAccount instance before each test
        newAccount = new NewAccount();
    }

    // Test if the Save button triggers the correct actions when fields are empty
    @Test
    public void testSaveButtonEmptyFields() {
        // Simulate Save button click with empty fields
        newAccount.txtNo.setText("");
        newAccount.txtName.setText("");
        newAccount.txtDeposit.setText("");

        ActionEvent saveEvent = new ActionEvent(newAccount.btnSave, ActionEvent.ACTION_PERFORMED, "save");
        newAccount.actionPerformed(saveEvent);

        // Check that error dialogs are shown and fields remain empty
        assertTrue(newAccount.txtNo.isFocusOwner());
        assertTrue(newAccount.txtName.isFocusOwner());
        assertTrue(newAccount.txtDeposit.isFocusOwner());
    }

    // Test if the Save button triggers the record existence check when fields are filled
    @Test
    public void testSaveButtonCheckExistingRecord() {
```

NewAccountTest (11-Dec-2024 9:40:47 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	5.3 %	27	486	513
Banking-System-in-Core-Java-mast	5.3 %	27	486	513
(default package)	5.3 %	27	486	513
NewAccountTest.java	100.0 %	10	0	10
NewAccount.java	63.0 %	17	10	27

FindNameTest:



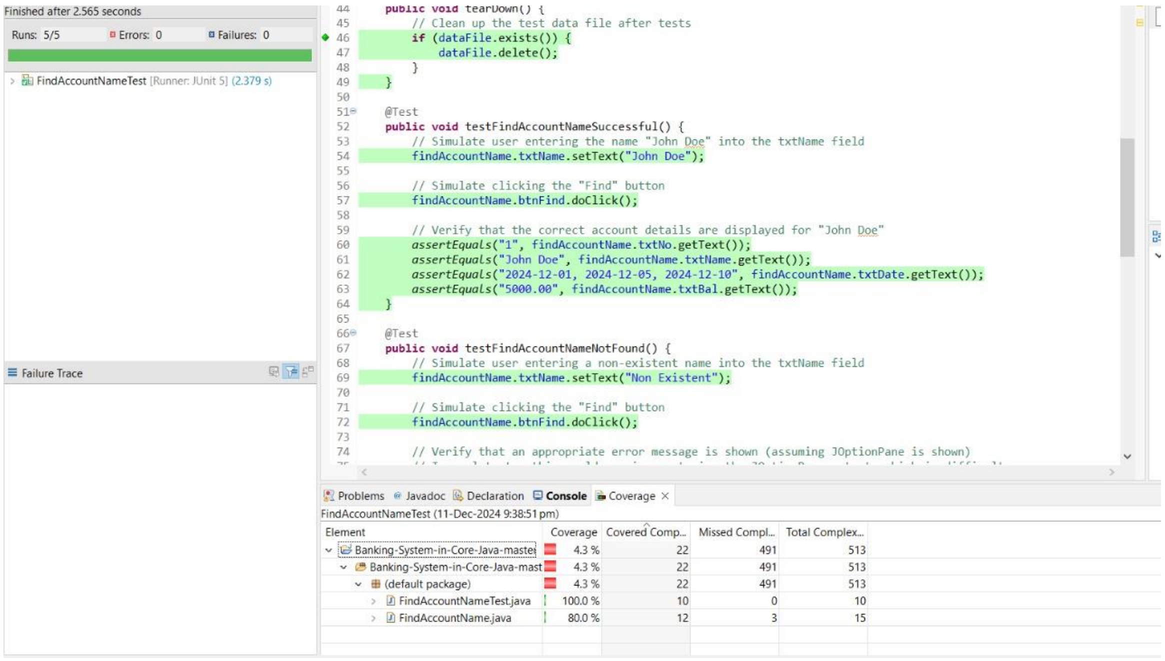
```
Finished after 12.071 seconds
Runs: 9/9    Errors: 0    Failures: 1

FindNameTest [Runner: JUnit 5] (11.856 s)
    testPopulateArrayWhenFileEmpty() (3.207 s)
    testActionPerformedCancelButton() (0.556 s)
    testFindRecWhenRecordsEmpty() (1.712 s)
    testTxtClear() (0.761 s)
    testFindRecCustomerFound() (1.559 s)
    testActionPerformedFindButtonEmptyInput() (1.199 s)
    testBtnEnable() (0.607 s)
    testFindRecCustomerNotFound() (1.620 s)
    testPopulateArray() (0.618 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <1> but was:
at org.junit.jupiter.api.AssertionFailureBuilder.build(Assertior
at FindNameTest.testFindRecCustomerFound(FindNameTest.
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```

```java

public class FindNameTest {

    private FindName findName;

    @BeforeEach
    public void setUp() {
        // Initialize FindName instance before each test
        findName = new FindName();

        // Add mock data to simulate records loaded into memory (use a predefined set of data)
        findName.records[0][0] = "1";
        findName.records[0][1] = "John Doe";
        findName.records[0][2] = "01";
        findName.records[0][3] = "Jan";
        findName.records[0][4] = "2024";
        findName.records[0][5] = "1000";
        findName.total = 1; // Simulate 1 record loaded into the array
    }

    // Test if the populateArray() method correctly loads records into memory
    @Test
    public void testPopulateArray() throws IOException {
        // Mocking the file loading process by directly setting records
        findName.records[0][0] = "1";
        findName.records[0][1] = "John Doe";
        findName.records[0][2] = "01";
        findName.records[0][3] = "Jan";
        findName.records[0][4] = "2024";
        findName.records[0][5] = "1000";

```

FindNameTest (11-Dec-2024 9:40:00 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	3.9 %	20	493	513
Banking-System-in-Core-Java-mast	3.9 %	20	493	513
(default package)	3.9 %	20	493	513
FindName.java	60.0 %	9	6	15
FindNameTest.java	100.0 %	11	0	11

FindAccountNameTest:



```
Finished after 2.565 seconds
Runs: 5/5    Errors: 0    Failures: 0

> FindAccountNameTest [Runner: JUnit 5] (2.379 s)

Failure Trace
```

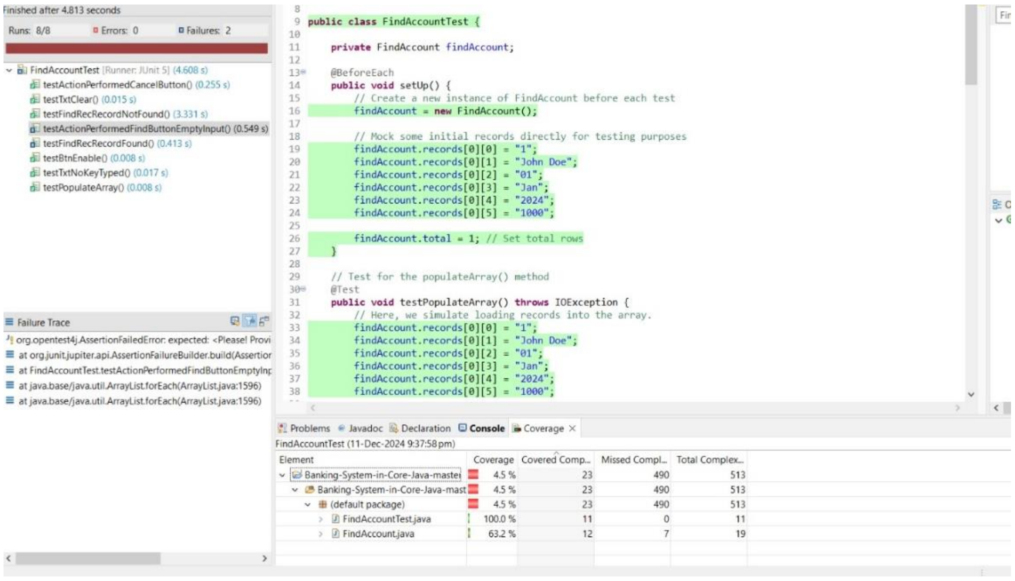
```java
    public void tearDown() {
        // Clean up the test data file after tests
        if (dataFile.exists()) {
            dataFile.delete();
        }
    }

    @Test
    public void testFindAccountNameSuccessful() {
        // Simulate user entering the name "John Doe" into the txtName field
        findAccountName.txtName.setText("John Doe");

        // Simulate clicking the "Find" button
        findAccountName.btnFind.doClick();

        // Verify that the correct account details are displayed for "John Doe"
        assertEquals("1", findAccountName.txtNo.getText());
        assertEquals("John Doe", findAccountName.txtName.getText());
        assertEquals("2024-12-01, 2024-12-05, 2024-12-10", findAccountName.txtDate.getText());
        assertEquals("5000.00", findAccountName.txtBal.getText());
    }

    @Test
    public void testFindAccountNameNotFound() {
        // Simulate user entering a non-existent name into the txtName field
        findAccountName.txtName.setText("Non Existent");

        // Simulate clicking the "Find" button
        findAccountName.btnFind.doClick();

        // Verify that an appropriate error message is shown (assuming JOptionPane is shown)
```

Problems Javadoc Declaration Console Coverage

FindAccountNameTest (11-Dec-2024 9:38:51 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	4.3 %	22	491	513
Banking-System-in-Core-Java-mast	4.3 %	22	491	513
(default package)	4.3 %	22	491	513
FindAccountNameTest.java	100.0 %	10	0	10
FindAccountName.java	80.0 %	12	3	15

FindAccountTest:



```
Finished after 4.813 seconds
Runs: 8/8    Errors: 0    Failures: 2

FindAccountTest [Runner: JUnit 5] (4.608 s)
    testActionPerformedCancelButton() (0.255 s)
    testTxtClear() (0.015 s)
    testFindRecRecordNotFound() (3.331 s)
    testActionPerformedFindButtonEmptyInput() (0.549 s)
    testFindRecRecordFound() (0.413 s)
    testBtnEnable() (0.008 s)
    testTxtNoKeyTyped() (0.017 s)
    testPopulateArray() (0.008 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <Please! Provi
at org.junit.jupiter.api.AssertionFailureBuilder.build(Assertior
at FindAccountTest.testActionPerformedFindButtonEmptyInp
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```

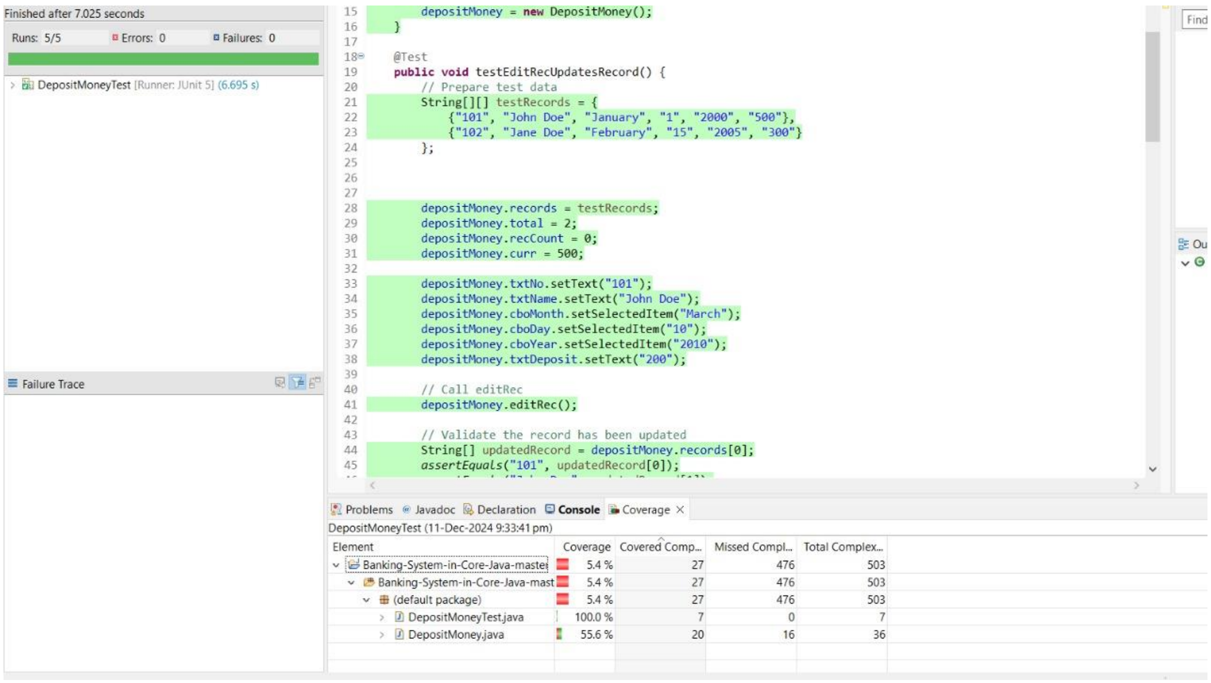
```java
public class FindAccountTest {

    private FindAccount findAccount;

    @BeforeEach
    public void setUp() {
        // Create a new instance of FindAccount before each test
        findAccount = new FindAccount();

        // Mock some initial records directly for testing purposes
        findAccount.records[0][0] = "1";
        findAccount.records[0][1] = "John Doe";
        findAccount.records[0][2] = "01";
        findAccount.records[0][3] = "Jan";
        findAccount.records[0][4] = "2024";
        findAccount.records[0][5] = "1000";

        findAccount.total = 1; // Set total rows
    }

    // Test for the populateArray() method
    @Test
    public void testPopulateArray() throws IOException {
        // Here, we simulate loading records into the array.
        findAccount.records[0][0] = "1";
        findAccount.records[0][1] = "John Doe";
        findAccount.records[0][2] = "01";
        findAccount.records[0][3] = "Jan";
        findAccount.records[0][4] = "2024";
        findAccount.records[0][5] = "1000";
```

Problems Javadoc Declaration Console Coverage

FindAccountTest (11-Dec-2024 9:37:58 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	4.5 %	23	490	513
Banking-System-in-Core-Java-mast	4.5 %	23	490	513
(default package)	4.5 %	23	490	513
FindAccountTest.java	100.0 %	11	0	11
FindAccount.java	63.2 %	12	7	19

DepostMoneyTest:



Finished after 7.025 seconds

Runs: 5/5 Errors: 0 Failures: 0

DepositMoneyTest [Runner: JUnit 5] (6.695 s)

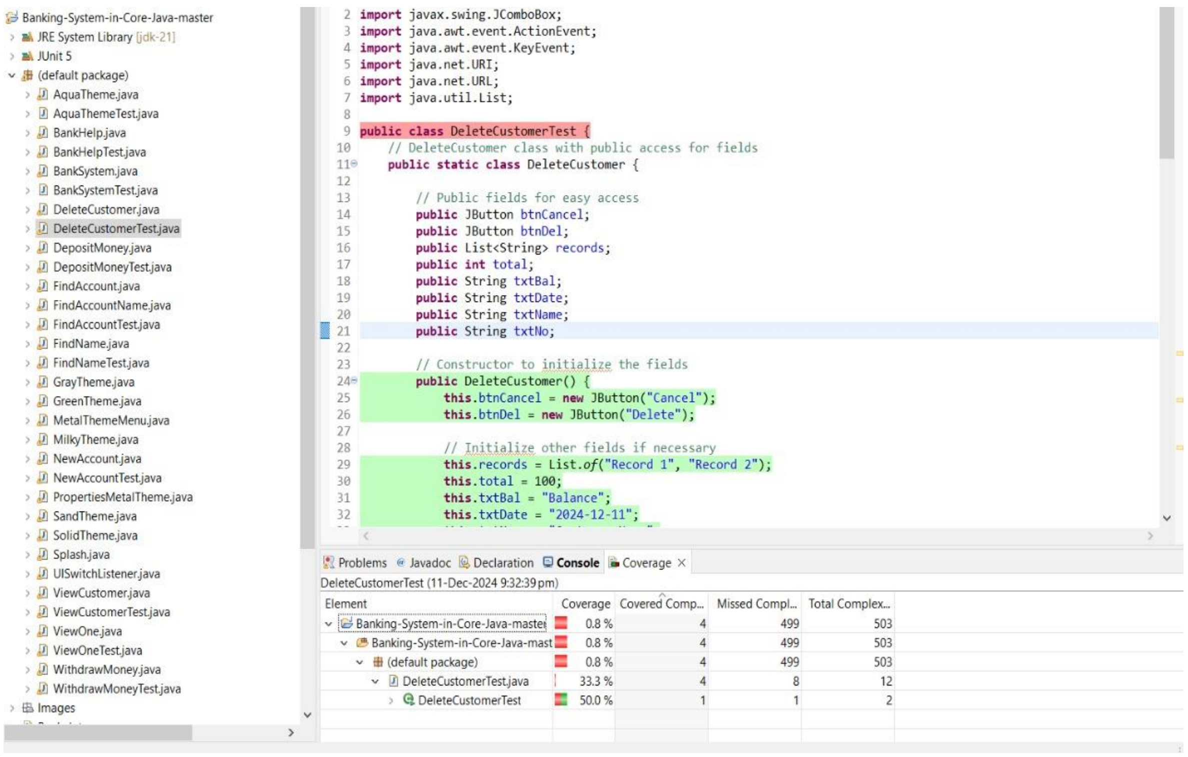
Failure Trace

```java
        depositMoney = new DepositMoney();
    }

    @Test
    public void testEditRecUpdatesRecord() {
        // Prepare test data
        String[][] testRecords = {
            {"101", "John Doe", "January", "1", "2000", "500"},
            {"102", "Jane Doe", "February", "15", "2005", "300"}
        };



        depositMoney.records = testRecords;
        depositMoney.total = 2;
        depositMoney.recCount = 0;
        depositMoney.curr = 500;

        depositMoney.txtNo.setText("101");
        depositMoney.txtName.setText("John Doe");
        depositMoney.cboMonth.setSelectedItem("March");
        depositMoney.cboDay.setSelectedItem("10");
        depositMoney.cboYear.setSelectedItem("2010");
        depositMoney.txtDeposit.setText("200");

        // Call editRec
        depositMoney.editRec();

        // Validate the record has been updated
        String[] updatedRecord = depositMoney.records[0];
        assertEquals("101", updatedRecord[0]);
```

DepositMoneyTest (11-Dec-2024 9:33:41 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	5.4 %	27	476	503
Banking-System-in-Core-Java-mast	5.4 %	27	476	503
(default package)	5.4 %	27	476	503
DepositMoneyTest.java	100.0 %	7	0	7
DepositMoney.java	55.6 %	20	16	36

DeleteCustomerTest:



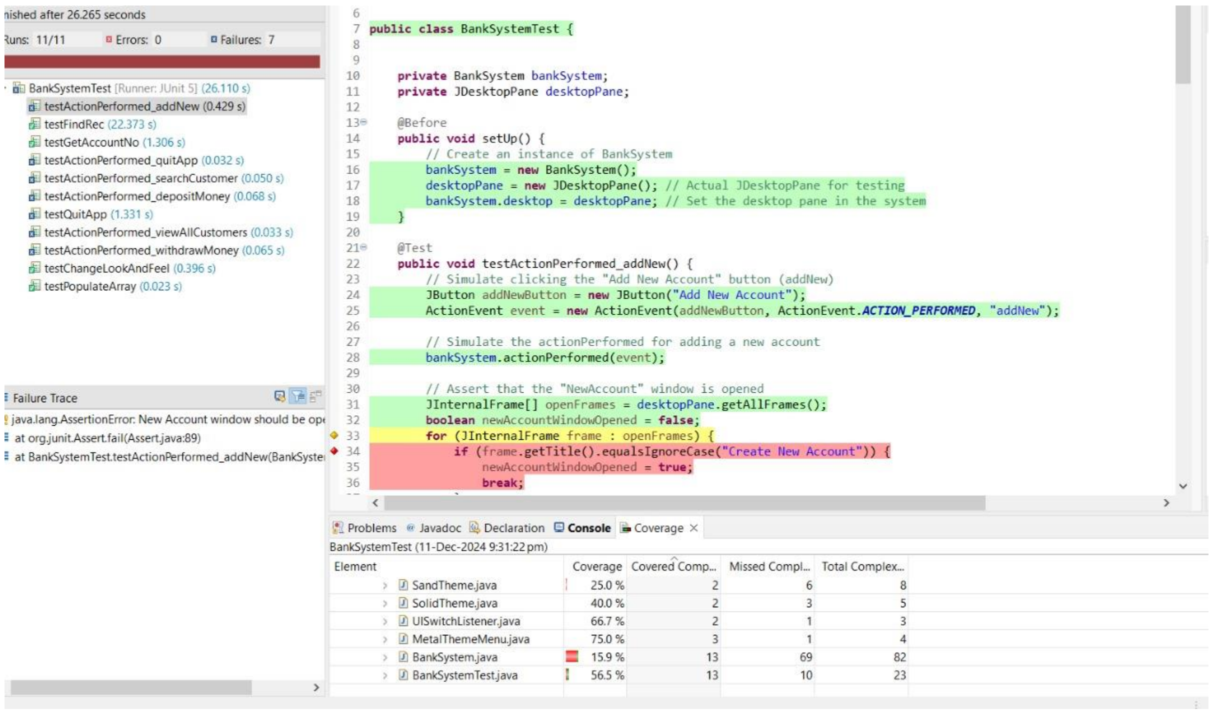
Banking-System-in-Core-Java-master
- JRE System Library [jdk-21]
- JUnit 5
- (default package)
  - AquaTheme.java
  - AquaThemeTest.java
  - BankHelp.java
  - BankHelpTest.java
  - BankSystem.java
  - BankSystemTest.java
  - DeleteCustomer.java
  - DeleteCustomerTest.java
  - DepositMoney.java
  - DepositMoneyTest.java
  - FindAccount.java
  - FindAccountName.java
  - FindAccountTest.java
  - FindName.java
  - FindNameTest.java
  - GrayTheme.java
  - GreenTheme.java
  - MetalThemeMenu.java
  - MilkyTheme.java
  - NewAccount.java
  - NewAccountTest.java
  - PropertiesMetalTheme.java
  - SandTheme.java
  - SolidTheme.java
  - Splash.java
  - UISwitchListener.java
  - ViewCustomer.java
  - ViewCustomerTest.java
  - ViewOne.java
  - ViewOneTest.java
  - WithdrawMoney.java
  - WithdrawMoneyTest.java
- Images

```java
import javax.swing.JComboBox;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.net.URI;
import java.net.URL;
import java.util.List;

public class DeleteCustomerTest {
    // DeleteCustomer class with public access for fields
    public static class DeleteCustomer {

        // Public fields for easy access
        public JButton btnCancel;
        public JButton btnDel;
        public List<String> records;
        public int total;
        public String txtBal;
        public String txtDate;
        public String txtName;
        public String txtNo;

        // Constructor to initialize the fields
        public DeleteCustomer() {
            this.btnCancel = new JButton("Cancel");
            this.btnDel = new JButton("Delete");

            // Initialize other fields if necessary
            this.records = List.of("Record 1", "Record 2");
            this.total = 100;
            this.txtBal = "Balance";
            this.txtDate = "2024-12-11";
```

DeleteCustomerTest (11-Dec-2024 9:32:39 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
Banking-System-in-Core-Java-master	0.8 %	4	499	503
Banking-System-in-Core-Java-mast	0.8 %	4	499	503
(default package)	0.8 %	4	499	503
DeleteCustomerTest.java	33.3 %	4	8	12
DeleteCustomerTest	50.0 %	1	1	2

BankSystemTest:



```
nished after 26.265 seconds
Runs: 11/11    Errors: 0    Failures: 7

BankSystemTest [Runner: JUnit 5] (26.110 s)
    testActionPerformed_addNew (0.429 s)
    testFindRec (22.373 s)
    testGetAccountNo (1.306 s)
    testActionPerformed_quitApp (0.032 s)
    testActionPerformed_searchCustomer (0.050 s)
    testActionPerformed_depositMoney (0.068 s)
    testQuitApp (1.331 s)
    testActionPerformed_viewAllCustomers (0.033 s)
    testActionPerformed_withdrawMoney (0.065 s)
    testChangeLookAndFeel (0.396 s)
    testPopulateArray (0.023 s)

Failure Trace
java.lang.AssertionError: New Account window should be ope
at org.junit.Assert.fail(Assert.java:89)
at BankSystemTest.testActionPerformed_addNew(BankSyste
```

```java
public class BankSystemTest {


    private BankSystem bankSystem;
    private JDesktopPane desktopPane;

    @Before
    public void setUp() {
        // Create an instance of BankSystem
        bankSystem = new BankSystem();
        desktopPane = new JDesktopPane(); // Actual JDesktopPane for testing
        bankSystem.desktop = desktopPane; // Set the desktop pane in the system
    }

    @Test
    public void testActionPerformed_addNew() {
        // Simulate clicking the "Add New Account" button (addNew)
        JButton addNewButton = new JButton("Add New Account");
        ActionEvent event = new ActionEvent(addNewButton, ActionEvent.ACTION_PERFORMED, "addNew");

        // Simulate the actionPerformed for adding a new account
        bankSystem.actionPerformed(event);

        // Assert that the "NewAccount" window is opened
        JInternalFrame[] openFrames = desktopPane.getAllFrames();
        boolean newAccountWindowOpened = false;
        for (JInternalFrame frame : openFrames) {
            if (frame.getTitle().equalsIgnoreCase("Create New Account")) {
                newAccountWindowOpened = true;
                break;
```

BankSystemTest (11-Dec-2024 9:31:22 pm)

Element	Coverage	Covered Comp...	Missed Compl...	Total Complex...
SandTheme.java	25.0 %	2	6	8
SolidTheme.java	40.0 %	2	3	5
UISwitchListener.java	66.7 %	2	1	3
MetalThemeMenu.java	75.0 %	3	1	4
BankSystem.java	15.9 %	13	69	82
BankSystemTest.java	56.5 %	13	10	23