**Test Report**

**for**

**Auto Text Generator**

**Version 1.0 approved**

**Prepared by**

**George Suresh Varghese – 150911007**

**Kunal Gupta – 150911\_\_\_**

**Adil Maqusood – 150911\_\_\_**

**13th April 2017**

White Box Testing

Code Snippets and Corresponding CFGs (Control Flow Graph) for five important functions of the project.

1. **Add Placeholder function**

public void addPlaceholder() {

1. jTextArea1.replaceSelection("");
2. if (jTextArea1.getCaretPosition() == 0) {

try {

1. jTextArea1.setCaretPosition(jTextArea1.getLineEndOffset(jTextArea1.getLineCount()) - 1);

} catch (Exception e1) {

// e1.printStackTrace();

}

}

1. String placeholder = Constants.PLACEHOLDER\_TEXT + noOfPlaceHolders++;
2. jTextArea1.insert(placeholder, jTextArea1.getCaretPosition());
3. defaultListModel.addElement(placeholder);
4. END }

**Cyclomatic Complexity =** E - N + 2 = 12 – 15 + 2 = -1

E = the number of edges of the graph

N = the number of nodes of the graph.

**Independent Paths:**

Path 1: 1-2-

Path 2: 1-2-3-5-6-7-9-10-11-12

**Test Cases:**

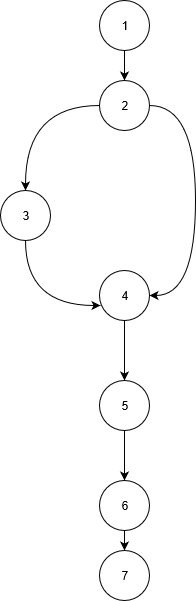
Test Case 1: JTextArea1.getCaretPosition() = 0

Test Case 2: JTextArea1.getCaretPosition() != 0

Test Case 1 covers Path 1

Test Case 2 covers Path 2

**Control Flow Graph (CFG):**

****

1. **Generate Text function**

private void generateFinal() {

1 System.out.println("ASASDASD");

2 String t1 = text

3 StringBuilder finalString = new StringBuilder();

4,5,12 for (int i = 0; i < tm.getRowCount(); i++) {

6,7,9 for (int j = 0; j < numberOfPlaceHolders; j++) {

8 text = text.replace(Constants.PLACEHOLDER\_TEXT + j, tm.getValueAt(i, j).toString());

}

10 finalString.append(text + "\n");

11 text = t1;

}

13 if (generateTextInterface != null) { //Check if started from plugin

14 generateTextInterface.replaceText(finalString.toString());

return;

}

15 JFileChooser jfc = new JFileChooser();

16 int result = jfc.showSaveDialog(this);

17 if (result == JFileChooser.CANCEL\_OPTION) {

return;

}

18 if (result == JFileChooser.APPROVE\_OPTION) {

try {

19 Files.write(Paths.get(jfc.getSelectedFile().getPath()),

finalString.toString().getBytes(), StandardOpenOption.CREATE);

20 JOptionPane.showMessageDialog(rootPane, jfc.getSelectedFile().getName().toString() + " saved successfully");

} catch (IOException ex) {

Logger.getLogger(GenerateTextForm.class.getName()).log(Level.SEVERE, null, ex);

}

}

21 END}

**Cyclomatic Complexity:** Predicate Nodes + 1

= 6 + 1

= 7

**Independent Paths:**

Path1: 1-2-3-4-5-13-15-16-17-18-19-20-21

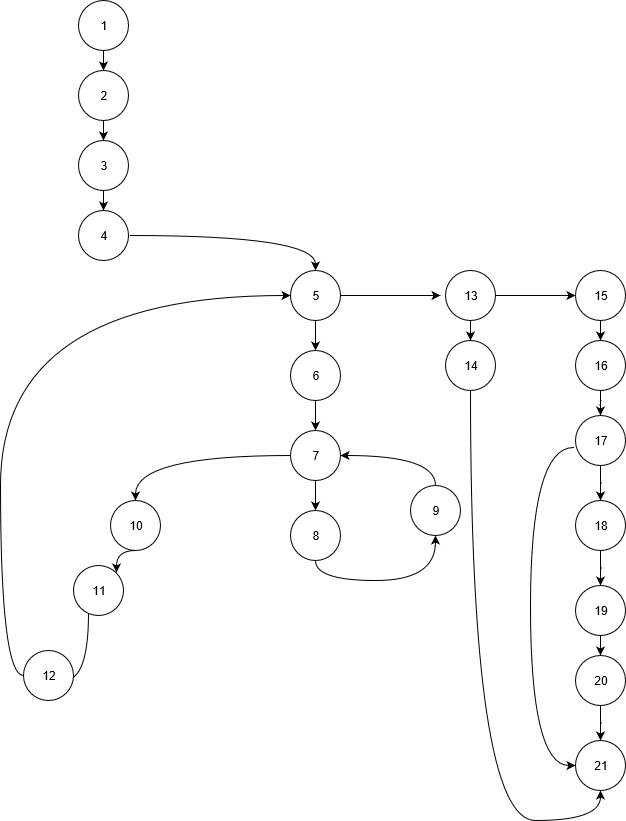
Path2 :1-2-3-4-5-6-7-8-9-7-10-11-12-5-13-15-16-17-18-19-20-21

**Test Cases :**  Test Case 1: tm.getRowCount() = 0, generateTextInterface = null, result == JFileChooser.APPROVE\_OPTION

Test Case 2: tm.getRowCount() != 0

TestCase 1 covers Path 1

TestCase 2 covers Path 2

**Control Flow Graph:**

1. **Remove placeholders**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

1 if (jList1.getSelectedIndex() == -1) {

return;

}

2 int index = jList1.getSelectedIndex();

3 jTextArea1.setText(jTextArea1.getText().replace(String.valueOf(defaultListModel.get(index)), ""));

4 defaultListModel.remove(index);

5,6,10 for (int i = index; i < defaultListModel.size(); i++) {

7 String cur = String.valueOf(defaultListModel.get(i));

8 renameInList(cur, Constants.PLACEHOLDER\_TEXT + (extractPlaceholdeNumber(cur) - 1));

9 jTextArea1.setText(jTextArea1.getText().replace(cur, Constants.PLACEHOLDER\_TEXT + (extractPlaceholdeNumber(cur) + 1)));

}

11 END }

**Cyclomatic Complexity :** Predicate node + 1

= 2 + 1

= 3

**Independent Paths:**  Path1: 1-11

Path2: 1-2-3-4-5-6-7-8-9-10-6-11

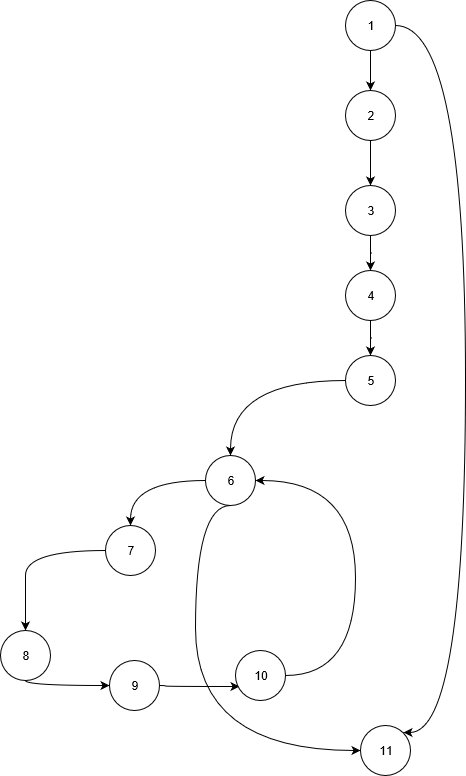
**Test Cases:** TestCase 1 : jList1.getSelectedIndex() == -1

TestCase 2 : jList1.getSelectedIndex() != -1, i = index; i <= defaultListModel.size()

TestCase 1 covers path3

TestCase 2 covers path1

**Control Flow Graph:**

****

1. **Add Placeholder rows**

private void menuAddRowsActionPerformed(java.awt.event.ActionEvent evt) {

1. Object result = JOptionPane.showInputDialog(this, "Enter Number of Rows to be added?");

try {

2 int no = Integer.parseInt(result.toString());

3,4,5 for (int i = 0; i < no; i++) {

6 tm.addRow();

}

} catch (Exception e1) {

showMessageDialog(this, e1.toString());

}

7 END }

**Cyclomatic Complexity :** Predicate node + 1

= 2 + 1

= 3

**Independent Paths:**  Path1: 1-11

Path2: 1-2-3-4-5-6-7-8-9-10-6-11

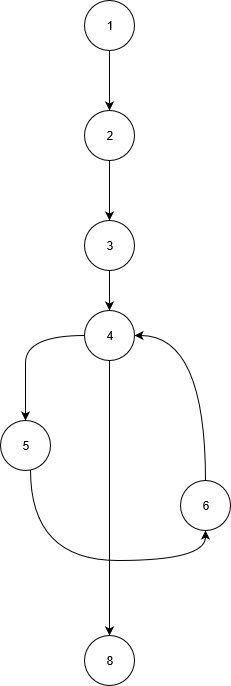
**Test Cases:** TestCase 1 : jList1.getSelectedIndex() == -1

TestCase 2 : jList1.getSelectedIndex() != -1, i = index; i <= defaultListModel.size()

TestCase 1 covers path3

TestCase 2 covers path1

**Control Flow Graph:**

****

1. **Generate numbers for placeholders**

private void generateNumbersActionPerformed(java.awt.event.ActionEvent evt) {

1 if (jTable1.getSelectedColumnCount() == 0) {

2 showMessageDialog(null, "No selected Column");

}

else {

3 Object result = JOptionPane.showInputDialog(null, "Enter Numbers seperated by - (Eg. 4-11)");

4 String[] sd = String.valueOf(result).split("-");

5 int n1 = Integer.parseInt(sd[0]);

6 int n2 = Integer.parseInt(sd[1]);

7,8,15 for (int i = 0; i <= n2 - n1; i++) {

9 if (jTable1.getSelectedRow() + i >= jTable1.getRowCount()) {

10 tm.addRow();

}

11,12,14 for (int j = 0; j < jTable1.getSelectedColumnCount(); j++) {

13 tm.setValueAt(n1 + i, jTable1.getSelectedRow() + i, jTable1.getSelectedColumn() + j);

}

}

}

16 END}

**Cyclomatic Complexity :** Predicate node + 1

= 6 + 1

= 7

**Independent Paths:**  Path1: 1-2-16

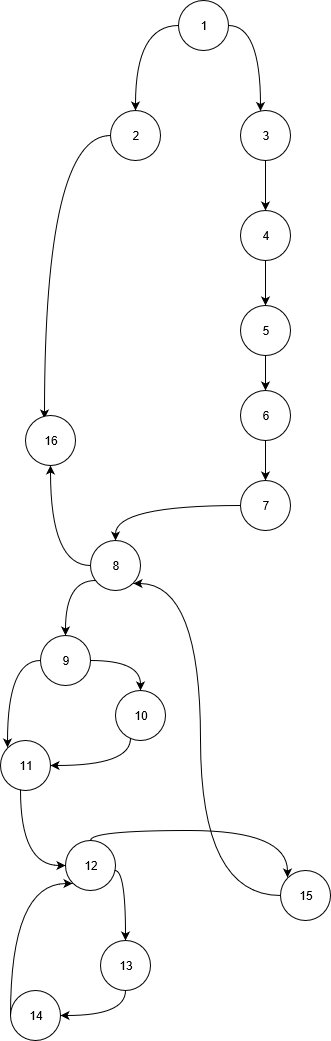
Path2: 1-3-4-5-6-7-8-9-10-11-12-13-14-12-15-8-16

**Test Cases:** TestCase 1 : jTable1.getSelectedColumnCount() == 0

TestCase 2 : jTable1.getSelectedColumnCount() != 0, i <= n2 - n1, j < jTable1.getSelectedColumnCount()

TestCase 1 covers path1

TestCase 2 covers path2

**Control Flow Graph:**

**Test Case 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Name**: Add Placeholder  **Short Description**: Insert a placeholder for generated text | | | |
| **Preconditions:**  The user has entered text into the Text Area | | | |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** |
| 1 | Click the “Add Placeholder” button | The system adds a placeholder at the end of the entered text, and adds it to the placeholder list. | Pass |
| **Post Condition:**   1. A new placeholder is added in the generating text | | | |

**Test Case 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Name**: Remove Placeholder  **Short Description**: Delete a placeholder for generated text | | | |
| **Preconditions:**  The user has entered text into the Text Area | | | |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** |
| 1 | Click the “REMOVE” button | The system adds a placeholder at the end of the entered text, and removes it from the placeholder list. | Pass |
| **Post Condition:**   1. A placeholder is removed from the generating text | | | |

**Test Case 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Name**: Generate Text Form  **Short Description**: Generate a table for placeholder insertion | | | |
| **Preconditions:**  The user has entered text into the Text Area | | | |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** |
| 1 | Click the “Generate” button | The system displays a table having number of columns equal to the number of placeholders in the entered text | Pass |
| 1 | Click the “Generate” button | The system displays the message "No placeholders. Add placeholders to generate" | Fail |
| **Post Condition:**   1. A table for placeholder insertion is generated | | | |

**Test Case 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Name**: Generate save file  **Short Description**: Generate save file for export | | | |
| **Preconditions:**  The user has entered text entries into the placeholder table | | | |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** |
| 1 | Click the “Generate” button | The system displays a file chooser for the user to choose the location of the file on the external storage. | Pass |
| 2 | Click the “Save” button | The system displays the message "<filename> saved successfully" | Pass |
| 2 | Click the “Cancel” button | The system cancels the operation of generated the save file | Fail |
| **Post Condition:**   1. A save file with generated text is created | | | |

**Test Case 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Name**: Generate numbers  **Short Description**: Generate consecutive numbers for a column in the placeholder table | | | |
| **Preconditions:**  The user has generated the placeholder table | | | |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** |
| 1 | Click the “Fill With” menu item | The system displays a list of options for modifying the placeholder table. | Pass |
| 2 | Click the “Generate numbers” menu item | The system displays the message “No column selected” | Fail |
| 2 | Click the “Generate numbers” menu item | The system displays the message asking the user to enter a range of numbers for the system to generate for each row in the selected column of the placeholder table | Pass |
| 3 | Click the “OK” button | The system generates a range of consecutive numbers for each row in the selected column of the placeholder table | Pass |
| **Post Condition:**   1. The rows of a selected column have consecutive numbers entered. | | | |