|  |
| --- |
|  |
|  |
|  |
| **Simple Merge**  **Software Requirement Specification** |
| Team number **#5** Team member 김성재, 박준영, 이현재, 주현준, 전진우 |

Content

1. Analysis
2. Design
3. Implementation

IV. Test Report

1. Analysis
2. Design
3. MVC modeling

* We design MVC(Model, View, Control) model.

1. Implementation
2. Control

* EditPanelController

In editPanelController, we control EditPanelModel(Model), EditPanel(View). Main function is Load, Save, Edit and this function is interact with MainWindowController.

•Save File

If click ‘Save’ button, we can save new file as original file or another named file. We get the string on the ‘EditPanelModel’(M of MVC), and copy this to new file or original file.

•Load File

In Load button’s ‘LoadActionListener’, when we choose a file and click ‘open’, system read one line by one line until meet ‘EOF’. And set file name on panel, set readed string on panel and other buttons(save, edit) is activated. After two file is loaded successfully, ‘merge’ button is activated. We modeled button action in ‘Control’, and set the contents in ‘View’ using ‘Model’.

•Edit File

Edit button is activated when file is loaded successfully. When file is loaded first, panel is uneditable. When edit button is clicked, we can edit the text on panel. And click again, panel is uneditable. Edit button control is in ‘Control’ area and ‘Control’ manipulate ‘editable’ and ‘uneditable’ function which is in ‘Model’ area.

* MainWindowController

Main is in this controller. All Swing component is designed not ‘thread-safe’. So approaching to swing component is acted on single-thread. This is called as ‘event-dispatch thread’. When we don’t result value and don’t care when task is ended. So we use EventQueue.invokeLater(Runnable runnable) method. When program is started, thread call the MVC. Main function is control the merge panel’s function(compare button, merge button, search button) and interact with EditPanelController.

•Compare

Compare the two loaded text file using ‘LCSubsequence’ and highlight the different section.

•Select(Up, Down)

When click ‘Up’ or ‘Down’ button, we search the highlighted section and highlight another color again.

•Merge(To Left, To Right)

After select the section to change, when click ‘Copy to Left’, merge right section to left section another is acted in inverse.

* LCSubsequence

1. Model

1. Test Report
2. Compare(LCS algorithm) Junit test

|  |
| --- |
|  |
| Compare with assertEquals. |
|  |
| We compare Above two file. |

|  |
| --- |
|  |
| Which is different? we expect the above result. (answer 1 ~ 6) |

|  |
| --- |
|  |
| Result : success |

1. Load and Save Junit test

Check that the load and save functions are working properly.

|  |
| --- |
|  |
| Test text file. |

|  |
| --- |
|  |
| Test String |

* Load

|  |
| --- |
|  |
| Bring a test file |

|  |
| --- |
|  |
| File read and store the text in ‘str’ variable. |

|  |
| --- |
|  |
| Load test code |

|  |
| --- |
|  |
| Result : success |

* Save

|  |
| --- |
|  |
| Save the text to file |

|  |
| --- |
|  |
| save text to file as ‘compareStr’ variable which is stored test text. |

|  |
| --- |
|  |
| And bring that file. |

|  |
| --- |
|  |
| Save test code |

|  |
| --- |
|  |
| Result : success |