Change request log

# Team

Manish Kanagala and Shravan Konduru

# Change Request

Change request 6.2. Change modifies the results of the search of recent files. Initially the search results highlighted files which started with the search phrase. The change request was to highlight search results which contained the search phrase anywhere.

# Concept Location

Use the table below to describe each step you follow when performing concept location for this change request. In your description, include the following information when appropriate:

* IDE Features used (e.g., searching tool, dependency navigator, debugging, etc.)
* Queries used when searching
* System executions and input to the system
* Interactions with the system (e.g., pages visited)
* Classes visited
* The first class found to be changed (this is when concept location ends)

When there is a major decision/step in the process, include its rationale, i.e., why that decision/step was taken.

Make sure you time yourselves when going through this process and provide the total time spent below.

The following is an example of a concept location process for the change request "Color student schedule":

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We looked through the file structure* | *To observe the different functional components of the program* |
| 2 | *We searched through the search folder* | *Since our change request was centered around modifying search results, we hoped to find the necessary classes to modify under the search folder* |
| 3 | *Read some classes in the search folder* | *To familiarize ourselves to the search process, to get a better understanding of what types of changes to make. We realized that the search class was not relevant to the search in the recent files tool* |
| 4 | *Ran the project* | *To use the search feature in the recent files tool to better understand how the function works* |
| 5 | *Use the regular expression search to search for “Recent files”* | *We noticed that the search result algorithm we wished to modify was specific to the recent files tool after running the program, so we implemented a search specific to that.* |
| 6 | *We inspected the files which were provided in our search* | *To find the file which contained the regex filtering of the search, and found the filter in the file “RecentFilesProvider”* |
| 7 | *We inspected the class “RecentFilesProvider”* | *We read the current regex matching to understand how to change the expression* |
| 8 | *Modified the regex* | *To make certain that the RecentFilesProvider class was correct, we removed the statements related regex matching. This confirmed that we were in the correct location, as the changes were reflected when we ran the project and no search results were highlighted.* |

**Time spent (in minutes):** 53

# Impact Analysis

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We made a list of methods called by “RecentFilesProvider”* | *To track the classes that could be impacted by the change.* |
| 2 | *We inspected the class which were called in the “RecentFilesProvider”class to check if they had any cascading impacts aswell* | *….* |
| 3 | *...* |  |
| 4 |

**Time spent (in minutes):** 30

# Actualization

Use the table below to describe each step you followed when changing the code. Include as many details as possible, including why classes/methods were modified, added, removed, renamed, etc.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *Changed the initialization of the String variable “regex”* | *Since the search phrase can be anywhere in the file name, we changed the regex to have “\*” on either side, signifying that any number of any characters could surround the search phrase in the file name.* |
| 2 | *Deleted the if statements which were necessary in the previous matching scheme* | *New regex expression did not need to perform the checks done by the previous one.* |
| 3 | *Ran tests by running the program and entering a pre-established suite of inputs.* | *To make sure everything works.* |
| 4 | *...* |  |

**Time spent (in minutes):** 35

# Validation

Use the table below to describe any validation activity (e.g., testing, code inspections, etc.) you performed for this change request. Include the description of each test case, the result (pass/fail) and its rationale.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *Test case defined: Input is the entire file name.*  *Inputs: “IntelliJ\_idea\_keys.props”*  *Expected output: “IntelliJ\_idea\_keys.props”* | *This is the regular expected behavior.*  *The test passed.* |
| 2 | *Test case defined: One letter*  *Inputs: “c”*  *Expected output: ”workspace.xml”, “vcs.xml”, “misc.xml”, “QuickNotepad.jar”* | *This is the regular expected behavior.*  *The test passed.* |
| 3 | *Test case defined: small phrase*  *Inputs: “dul”*  *Expected output: “modules.xml”* | *This is the regular expected behavior.*  *The test passed.* |
| 4 | *Test case defined: name of file plus an extra character*  *Inputs: “modules.xmls”*  *Expected output: nothing* | *This is the regular expected behavior.*  *The test passed.* |

**Time spent (in minutes):** 15

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 53 |
| Impact Analysis | 30 |
| Actualization | 35 |
| Verification | 15 |
| Total | 133 |

# Reverse engineering

# Diagram Description automatically generatedDiagram Description automatically generatedConclusions

The main challenge we faced during this change was having to locate the class which was to be changed. We begin the concept location process somewhat clumsily, randomly searching functions which we perceived to be relevant. Once we took the time to look at the actual functionality of the element, we had a much better idea what to look for, and were able to implement the IDE’s regular expression search with success. The next *.*

*Classes and methods changed:*

* *Jedit-f21-team05/org/gjt/sp/jedit/menu/RecentFilesProvider*
  + *Public void update(Jmenu menu)*
    - *Public void keyReleased(keyEvent e)*