Change request log

1 Team

Team name:

Fari-Marylou

Team members and roles:

- Farzaneh Kadkhodaie

Role: Investigate and fix je-1. Implement the code and document the investigation and steps taken to solve the problem. Complete the change request log form and provide UML diagrams.

- Marylou Nash

Role: Provide support. Answer the questions and help evaluate ideas. Review and verify the fix.

2 Change Request

- Change request je-1:

The purpose of this change request is to modify the status bar to display:

- the word offset of the caret from the beginning of the file
- the number of words in the file.
- Additionally, for consistency, we should modify the code in a way that the status bar preference dialog displays the new caret display options as check boxes.

3 Concept Location

Step #	Description	Rationale
1	*I ran the code to execute the system.*	*To run the jEdit text editor in order to become familiar with jEdit software and its behavior and features.*
2	*I interacted with the system, in particular, I started to inspect the status bar displayed information related to caret/cursor information.*	*To observe how modifying editor's switch buffer (e.g. adding new words, changing cursor location) can change status bar's reported information.*
3	*I used IDE"s search tool and searched for "Status" (I used Status as the query for search).*	*To find any class/function/feature related to the status bar. Also, to locate where exactly the caret' information in source code is displaying on status bar.*
4	*The search tool listed several classes, including* • *StatusBar.java* • *StatusBarOptionPane.java* • *Status handler in EditPane.java*	*I found these three classes (among all 9 classes) more related to je-1 change request, hence I discard other classes from my investigation.*

5	*I clicked on "StatusBar.java" class, from search results as the first class to start my investigation.*	*This class seems the most related and promising class to start to investigate.*
6	*Start to observe/inspect the class and become familiar with its functionality and implementation.*	*Because some useful information (about caret) is already displayed on status bar. Therefore, I tried to locate them.*
7	*Read carefully updateCaretStatus function to understand its implementation and details.*	*Since this function is exactly where to provide the je-1 change request.*
8	*Start to look for other classes that "StatusBar.java" interact to, to located where caret/cursor info come from to be reported/displayed on the status bar.*	*"StatusBar.java" class already displays some info on the status bar, so I need to understand where this info is collected /calculated in a lower level.*
9	*I started to look for some functions that might be useful and related to caret's reported information. I found three functions:* • *getCaretPosition ()* • *getCaretLine ()* • *getLineCount ()*	*To find which class contains information related to caret's displayed on the status bar.*
10	*I searched these functions in my IDE's search tools.* *The first two functions (located in textArea.java class) did not provide any useful information, So I discarded their class (textArea.java) class from my inspection.*	
11	*Searching the last function (getLineCount ()) in the search bar leaded me to "jEditBuffer.java class, and I marked this class as the class, where new functions need to be implemented!*	*This class is exactly where some other useful functions (related to caret position and text buffer, such as the number of characters and the number of total lines in the buffer) are implemented.*
12	*Start to implement two new functions:* • *countWords () to calculate file word count* • *getWordOffset (int) to calculate word offset from the beginning of the file*	
13	*Run jEdit editor several times test/inspect the new implemented functions' behavior for different situations, i.e. writing single/multiple lines of words in the buffer switcher and changing caret's location.*	*To make sure the implemented feature for je-1 is working properly and shows desired output*

14	*Start to update status bar preference dialog to add two checkboxes corresponding to the two new implemented features.*	*For the sake of consistency, as the assignment asked for.*
15	*Using the IDE"s search tool, we started to search for the "status" keyword again.*	*To find other related classes (other than StautsBar.java" class.*
16	*From several options (mentioned in step 4), we clicked on "StatusBarOptionPane.java" class.*	*This class seems the most relevant class.*
17	*Inspect the class carefully and try to locate other implemented options provided in status bar preference dialog.* *To ease this step, I opened the Global Option menu (located in jEdit>utilities>Global Options>Status Bar) and tried to match the code with their visualize options.*	*To make sure that this class is the right location to provide modifications since other checkboxes are already implemented there.*
18	*I marked "StatusBarOptionPane.java" class, as the right class where the modifications need to be provided.*	*Concept location is done!*

Time spent (in minutes): 190 min

4 Impact Analysis

Step #	Description	Rationale
1	*So far, three classes are involved in implementing je-1 change request:* • *StatusBar.java* • *jEditBuffer.java* • *StatusBarOptionPane.java*	
2	• *"StatusBar.java" class is changed because of extending/modifying updateCaretStatus() function.* • *"jEditBuffer.java" class is changed because of adding 2 new functions to implement change request" je-1" on it.* • *"StatusBarOptionPane.java" class also is changed, since the new codes to*	

	implement checkboxes located in global option tab are implemented in this class.	
3	*We labeled these three classes as "changed" classes, so we need to perform impact analysis for these three classes.*	*to make sure our modifications in these two class do not have any interfere with other classes that interact with them.*
4	• *For StatusBar.java class:* *Estimated inspected set for this class contains class "jEdit.java"class, because our new codes called getBooleanProperty(string,boolean) function from this class.* *Therefore, "jEdit.java" class labeled as "to be inspected".*	
5	• *For jEditBuffer.java class:* *Estimated inspected set for this class includes class "contentManager.java", as the two new implemented methods calls getLength() and getText(int, int) from this class.* *Therefore, "contentManager.java" class labeled as "to be inspected"*	
6	• *For "StatusBarOptionPane.java" class:* *Estimated inspected set for this class only contains "jEdit.java", as implementing new checkboxes for je-1 is calling method getProperty(name: string).* *Therefore, "jEdit.java" class labeled as "to be inspected"*	
7	*After inspecting estimated inspected sets carefully for all classes, I did not find any other classes which get impacted by the provided modifications.* *Hence, they are discarded from the estimated impact set, because the changes are internal, and our methods only call these functions (belong to other classes) without any further manipulating.*	
8	*We change their labels from "to be inspected" to "inspected and unchanged".*	

Time spent (in minutes): 60

5 Prefactoring (optional)

Step #	Description	Rationale
1	*No prefactoring was done.*	*Prefactoring phase would be easy, since countWords() and getWordOffset(position:int) methods have some common fragments to be extracted and make up a separated simple and short function.*

Time spent (in minutes): 0

6 Actualization

Step #	Description	Rationale
1	*I created two new functions, wordCounts() and getWordOffset(int) directly in "jEditBuffer.java" class, found in concept location phase .*	*I realized that the functions' body are short, and for simplicity and consistency, they can be implemented next to the other functions that are interacting with status bar contents.*
2	*I called these two functions within "StatusBar.java" class from updateCaretStatus() function.*	*Since "StatusBar.java" is responsible to report/display caret position information in the status bar.*
3	*In addition, in "StatusBar.java" class, I followed the same structure used to implement the other caret's information and give proper values to getBooleanProperty(name: string, def: boolean) in associate with the" number of words in the file" and "caret word offset from the beginning of a file".*	
4	*I followed the same structure used to implement other checkboxes for status bar preference dialog. So, in "StatusBarOptionPane.java" class, I created two new checkboxes (JcheckBox type) and gave them proper values in associate with the" number of words in the file" and "caret word offset from the beginning of a file".*	*For consistency and to keep implementation similar to prior options.*
5	*I implement codes in "StatusBarOptionPane.java" class, to add these two components (checkboxes) to the global option panel. So, they will be shown in global options as follow:* • *Offset of word from beginning of the file* • *Number of words in the file*	

6	*Push my codes to my branch in github and ask my teammate to pull them on her local machine and confirm if she agrees.*	

Time spent (in minutes): 120

7 Postfactoring (optional)

Step #	Description	Rationale
	No postfactoring was done.	

Time spent (in minutes): 0

8 Validation

Step #	Description	Rationale
1	*I provided manual testing for this part. To do that, I run the system, opened a file on jEdit editor and started to test different cases.*	*To make sure system satisfies the change request requirements and it works properly.*
2	***Test Case defined:*** *This test case is to verify the number of words in the file. So, we wrote the following input in jEdit buffer to see the result.* ***Input:*** *Hi! This is me ! . @@ #* ***Expected output:*** *8*	*To verify the number of words in file I used a MS word file to make sure the number of words counted in jEdit and MS word file is the same.* *The returned output is 8, which is correct. Therefore, the test passed!*
3	*I repeated step 3 multiple times for different inputs (including punctuations, multiple lines, blank line, empty text, multiple space, tab, etc.) to ensure our function work properly.*	*All tests passed.*
4	***Test Case defined:*** *Test case for verifying the offset of words in the file from the beginning of the file.* *(cursor is on "This" and sentence starts from beginning of the file, offset zero!)* ***Input:***	*The returned output is 2, and I expected to see 2, since cursor is located on the second word of the file.* *The test passed!*

	Hi! This is me ! . @@ # ***Expected output** 2*	
5	***Test Case defined:*** *Test case for verifying the offset of words in the file from the beginning of the file.* *(cursor is on "This" and text starts with multiple spaces!)* ***Input:*** *Hi! This is me ! . @@ #* ***Expected output (cursor is on "This")*** *2*	*The returned output is 3, however, I expected to see 2.* *The test failed, since I noticed is not working correctly when the text file started with some specific characters such as "\n", \t", space " ".*
6	*I came back through the code and found out that some characters (e.g. '\t' considered as a word). So, I fixed the bug!*	
7	***Test Case defined:*** *(cursor is on "This" and sentence starts with multiple spaces !)* ***Input:*** *Hi! This is me ! . @@ #* ***Expected output (cursor is on "This")*** *2*	*The returned value is 2, and I expected to see 2, as caret is on the second word. Therefore, the test passed!*
8	*I repeated step 7 multiple times for different inputs (in multiple lines, start with simple, multiple blank, '\t', characters, etc.) to ensure our function work properly.*	*All tests passed!*
9	*Also, for multiple times, I checked/unchecked the checkboxes to verify their implementation work properly.*	
10	***Test case defined:*** *Test case to verify if unchecking the checkboxes is working properly.*	*Since the property checkbox is unchecked, we expect not to see corresponding information on the status bar.*

	(cursor is on "This" and sentence starts with multiple spaces !) ***Input:*** *Hi! This is me ! . @@ #* ***Expected output (cursor is on "This")*** *nothing, blank*	*The test passed!*
11	***Test case defined:*** *Split a file into two/more windows and try to see if changing one file will update other windows' status bar.*	*We expect when one file/window's contents is changed, the update will be shown on other file's status bars. Therefore, test passed!*

Time spent (in minutes): 150

9 Timing

Phase Name	Time (in minutes)
Concept location	190
Impact Analysis	60
Prefactoring	0
Actualization	120
Postfactoring	0
Verification	150
Total	520

10 Reverse engineering

UML sequence diagram:

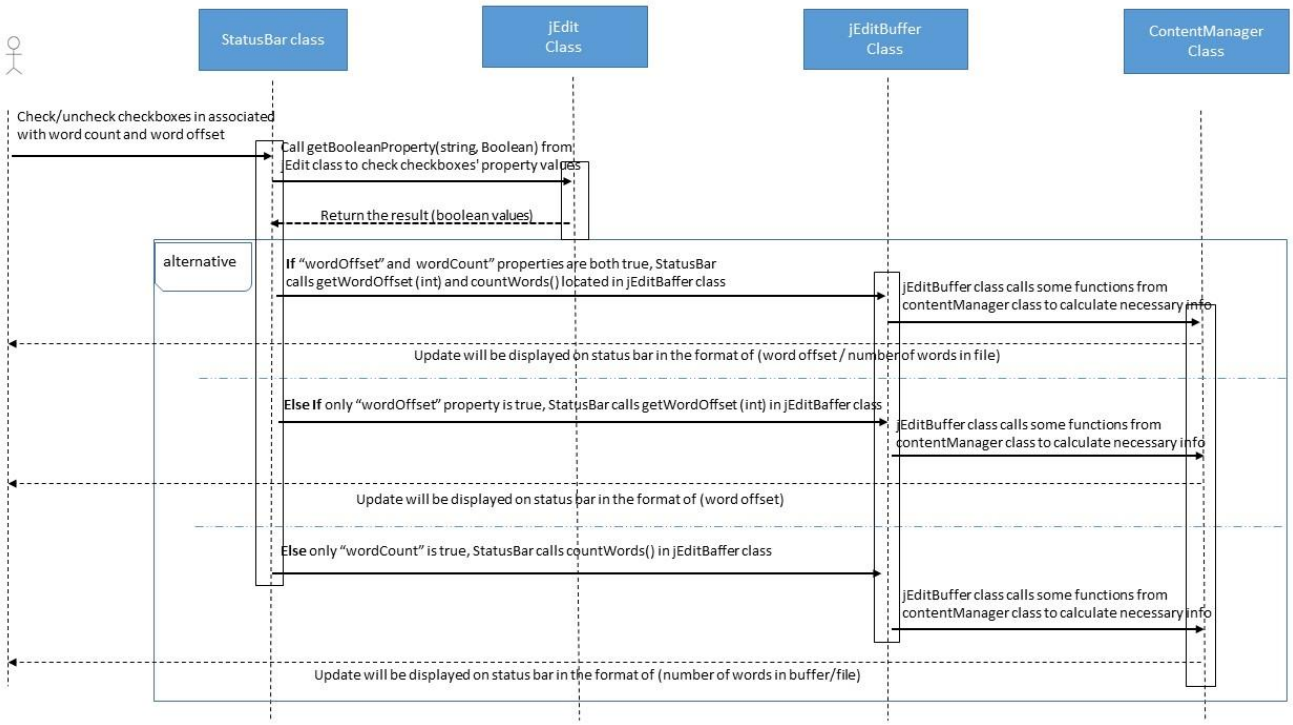

Figure1. UML Sequence diagram for je-1

Partial UML class diagram For Je-1:

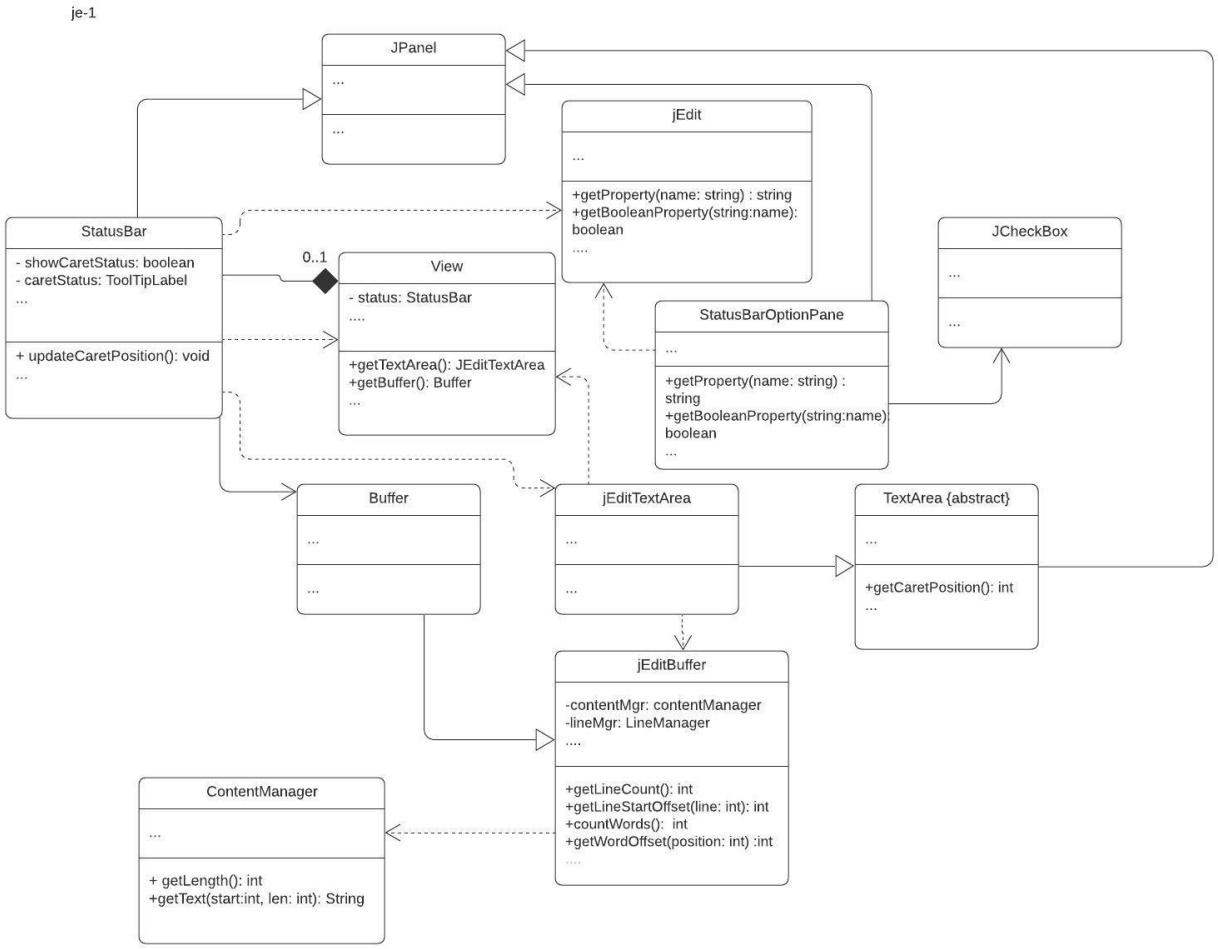

Figure2. UML Class diagram for je-1

11 Conclusions

For this change, concept location was quite easy and straight forward, because system architecture was well organized and designed. In addition, classes and functions' names were relevant to their functionality, which was very helpful for searching keywords to locate concepts for concept location phase. However, this phase took a bit longer time, compared to other phases, because it was the first change request to be implemented for jEdit and I was not very familiar with code structure.

Impact analysis phase was not complicated in my opinion, because most modifications are implemented as internal changes, therefore, they do not affect other external classes. The list of inspected classes for impact analysis is provided in impact analysis phase.

Moreover, actualization was quite simple and easy, as functions to be implemented were quite short and simple.

Additionally, validation and testing were performed manually for this change request. This made the process of bug tracking a bit challenging and time consuming, especially to fix the bug mentioned in validation section (phase8). However, running debugger could ease this step significantly.

Classes and methods changed:

- org/gjt/sp/jedit/buffer/jEditbuffer.java /** two new methods are added to this class */
    - public int WordCounts()
    - public int getWordOffset(position: int)
- org/gjt/sp/jedit/gui/StatusBar.java
    - public void updateCaretStatus() /** this function is modified */
- org/gjt/sp/jedit/options/StatusBarOptionPane.java
    - protected void _init() /** this function is modified/extended*/