Change request #1 log

1. Concept Location

Step #	Description	Rationale
1	Executed the code and ran the jEdit.	We had to initially get explore a little bit to understand the functionalities.
2	Typed a few lines in the jEdit.	Have done this step to get familiar with positions of the caret and to understand the count of the characters and the lines.
3	Searched for the keyword "caret" and identified the functions getCaretPosition and getCaretLine would probably be the functions returning the values of the caret on the jEdit.	Have done this step to locate the file where we could potentially make the changes.
4	Traced the getCaretPosition() and was redirected to StatusBar.java and Abbrevs.java files where the function was being called.	This step was done to identify where the function was defined.
5	Then created logs in Abbrevs.java and StatusBar.java.	To check for the values which were being returned, so that we could confirm in which files we could make the changes.
6	But this attempt wasn't very successful, both the logs in the in those files returned the values, so we had to go over the code in both the files individually.	At this point we were not sure where we could start making the changes.
7	Finally, we identified that StatusBar.java was to be changed where we found the updateCaretStatus(), here we confirmed where we had to make the changes.	We got the confirmation at this point because of the buf.append() functions which was used define the format for caret positions placing the "(" , "/" and ")"

Time spent (in minutes): 75

- org/gjt/sp/jedit/gui/StatusBar.java
  - updateCaretStatus()
- org/gjt/sp/jedit/Abbrevs.java
  - expandAbbrev()
- org/gjt/sp/jedit/textarea/TextArea.java
  - updateCaretPosition()

2. Impact Analysis

Step #	Description	Rationale
1	So in updateCaretStatus() we debugged the code using break points to inspect the values of getCaretPosition(), getCaretLine(), getCaretCount()	This step was done to analyze how the values for the caret position , caret line number and the number characters were being returned.
2	By doing the above step we figured out a way to calculate the total number of words and the number of words from start till the caret.	This step was done as we must return the output as required.

Time spent (in minutes): 15

- org/gjt/sp/jedit/gui/StatusBar.java
  - updateCaretStatus()

3. Actualization

Step #	Description	Rationale
1	Made the change by using the split() method and length() to calculate the total number of words.	The split() method separated the text using spaces and returned an array of words, while the length() method counted the number of words in the array.
2	Included an "if" statement right before the statement that returns the count of total words.	This ensured that the value was not calculated repeatedly, thereby reducing processing time.
3	In the second step, we followed a similar approach as before, but this time we provided the parameters 0 and caretPosition to obtain the word count from the beginning of the file to the current caret position.	We passed these parameters to the getText() method, where 0 represents the start of the file and the caretPosition indicates the current position of the caret.
4	Also included an "if" for the code in the previous step.	Included this step to reduce the executing time.
5	Finally at the end added a few more lines of code of "buf.append()" methods to return the count of total words and the count of words till the caret.	We did this step referring to the previous code on how the values for line number, column position of the caret, number of characters, etc

Time spent (in minutes): 60

Inspected:

- org/gjt/sp/jedit/gui/StatusBar.java
    - updateCaretStatus()

Changed:

- org/gjt/sp/jedit/gui/StatusBar.java
    - updateCaretStatus()

4. Validation

Step #	Description	Rationale
1	Tested the values were being returned properly. Inputs: A text file and placing the caret randomly a some position Expected output: line number, column number (number of characters from the start/ total number of characters) (number of words from the start till the caret offset/total number of words)	The test case was passed and the output was returned as expected with correct values.

Time spent (in minutes): 10

5. Summary of the change request

Phase	Time (minutes)	No. of classes inspected	No. of classes changed	No. of methods inspected	No. of methods changes
Concept location	75	3	3	3	3
Impact Analysis	15	1	1	1	1
Actualization	60	1	1	1	1
Verification	10	1	0	1	0
Total	160	6	5	6	5

6. Conclusions

For this change to concept location was a bit tricky but once after locating where the changes are to be made it was a smooth ride for the later part. The process involved locating the relevant functions, analyzing their impact, making the necessary changes, and validating the results. The change was successfully implemented and validated, resulting in the expected output. All the tasks were performed on IntelliJ itself.