Noah Delano

Contents of Program/Progress

# of lines, including comments	163
# of for loops	0
# of while loops	1
# of if/else statements	5

In this program, the goal was to allow the user to upload a text file, which the software will then examine and list the number of lines, minus blank lines and comments, both of the form // and /* to */.

This time around, I wrote notes as I progressed, just to keep me updated on where I am in the program and what needs to get done:

Planning: Stage 1

On our last assignment, we did something similar in which we had to examine a file using JFileChooser, so that the user could get the mean and standard deviation of all the numbers within the selected file. With that in mind, I thought the method for which I traversed the lines within the text file was perfect for helping me find the total number of lines. Therefore, my first goal was to just create a simple JFrame with a JButton and JTextArea, such that the button activates JFileChooser, and therein examine a file for the number of lines. Before I started looking into blank lines or comments, I just wanted to generally find the number of lines just to make sure it works (just insert a counter for each new iteration of each line). So then I got to coding.

Testing: Stage 1

As I said before, I wanted to start off simple by a simple Jframe that will count the total number of lines in a file and display that value in the JTextArea. Luckily, I ran into no issues through this stage. I simply used the same method of traversing a text file as the last assignment, and simply created a counter that added 1 for every line traversed. I tested with 3 different text files, with various number of lines, and it worked each time. So now that I have a general program, I can start looking for special lines that can’t be included in the counter, such as empty lines or ones just containing white spaces.

Planning: Stage 2

The goal this time around is to not include lines that are empty or are only white spaces. So some pseudocode could be something like:

```
while (reading each line)
        Counter ++
    If (the line isEmpty() or isBlank())
        Counter -= 1
```

Now to code.

Testing: Stage 2

Once again, I didn't run into much issue. I just simply used the isBlank() function for each line as it accounted for both empty lines or just blank spaces, so that condition has been met. Now it is time for comments.

Planning: Stage 3

I saved this part for last as it appeared to be the hardest component of my program: comments. I went online and googled some example cases to help, and based on what I got, I thought a while loop would be best to use. So here is some pseudocode:

```
while(/* has been read)
    #ofCommentLines ++
    if(*/ is reached)
        Totalcount - #ofCommentLines
if(line contains //)
    counter-=1
```

Now to code

Testing: Stage 3

When I went back into my program, I realized I am already using a while loop to traverse the lines. Therefore a while loop within a while loop didn’t seem like the best idea. So instead I created a boolean variable for comment. Basically if a line contained /*, then it would turn to true, turning on a another counter, which would stop increasing once */ (turn comment to false) was reached. Therefore, subtracting the number of comment lines to the total count. Once again I had no errors. Honestly, this is the first time I had no issues in creating a program, so I am honestly quite proud of myself. All that was left after this was some additional coding in order to make my JFRame nicer looking, as well as inserting notes and JavaDocs throughout my program so that anyone can understand the program and in case I need to refer to it again.