Samita Shrestha (qkp232)

Strings and Words Program: Design Changes

I. Introduction

The Strings and Words program is designed to perform various word statistics on a given document (as a string), such as counting the frequency of each unique word. The program initially only had this basic functionality, but with the Project check point, more features were added to it. In Phase 2, the program was extended to include counting the number of lines and characters, and in Phase 3, it was updated to allow for word replacement. This report details the changes made to the design of the program.

II. Original Design:

The initial requirement of the program was to count the frequency of each unique word in a given document. The code should support combinations of space, tab, and newline characters as separators. The program allowed users to pass input from the command line and supported separators such as space, tab, and newline characters. The input given had to be in textual format, and the program was case sensitive, meaning that "Test" and "test" were treated as two different words.

Features:

- Program shall allow the user to pass the input from the command line for which they want to count the words and its frequency.
- This program is case sensitive; For example, "Test" is not same as "test".
- The program supports separators like space, tab, newline, and the combination of them.
- The input given should be in textual format.

III. Phase 2 Design Changes:

In Phase 2, two more features were added to the program, including counting the number of lines (LineCount) and characters (CharCount). The program now counted the number of lines and characters from the input file. For the line count, even if there were no characters but had separators such as new lines, the line was counted. In other words, every newline character would increment the line count even if there may or may not be any other characters. The program did not consider syntax or grammatical correctness of the word. For example, S#@tu is still considered a word with 5 numbers of characters.

IV. Phase 3 Design Changes:

In Phase 3, the program was further extended to allow the replacement of all occurrences of a given word with a replacement word. However, the replacement only happened when the given pattern word matched with a whole word. For example, for text "ab cd ef," replacing "a" with "b" would result in no change, while replacing "ab" with "cd" would result in "cd cd ef." To make the program more user-friendly, the entry point of the program was updated to allow the user to choose from two options in the console. The first choice would take them to Phase 1 and Phase 2, i.e., to word statistics (Word, Char, and line count), whereas Choice 2 would take the user to Phase 3, i.e., line replacement. Since the program is an extension of the previous phases, users can still access and use all the features of the previous phases with ease.

V. CONCLUSION:

The Strings and Words program underwent significant design changes to add more features and functionality to it. The initial requirement was to count the frequency of each unique word, but the program was extended in Phases 2 and 3 to include additional features like counting the number of lines and characters and allowing for word replacement. The entry point of the program was also updated to allow for user-friendliness. These changes resulted in a more robust program that meets the needs of users with varying requirements. Furthermore, in this project, a class-based approach was implemented, making the program modular and easily extendable. As a result, new features can be added as new classes and seamlessly integrated into the program's existing framework.