

TIC3001 Assignment 3: KWIC

| | |
|----------------------|---|
| Code Repository URL: | https://github.com/binbinhui/TIC3001_assignment.git |
| Names: | Lin Binhui & Wang Zhenquan |
| Student Numbers: | A0177820B & A0177740Y |

1. Introduction

Bin Hui - Search testing process & Software Patterns & Sonarqude.

Zhenquan - Search Facility & Architectural Design.

2. Architectural Design - Implicit Invocation with Search Facility

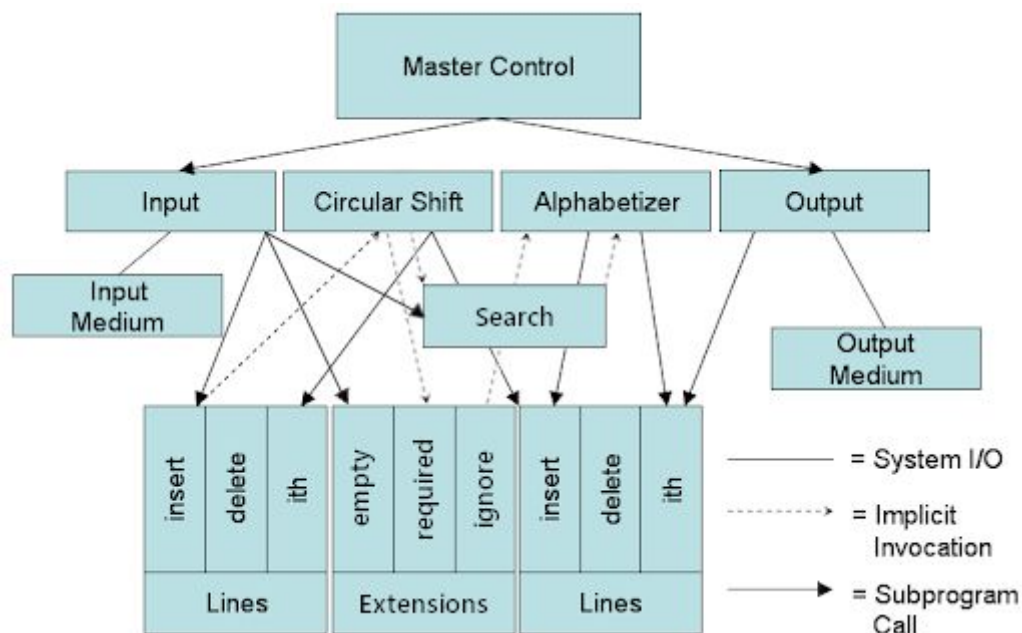


Figure 1. A new search class has been added.

In a very similar fashion to the required words component, instead of being fetched from a text file, the user inputs the search words which will be stored in accumulating ArrayList of strings that will search all text files listed in *ListOfFiles0.in* and compiles their combined output in a single text file.

The Search Facility is implemented in the Circular Shift module before invoking the Alphabetizer. The previous extensions of ignored and required words are toggled off during the search, else the expected output will not be achieved.

3. Applying Software Patterns

In order to implement a search function with the existing Extension Class, we have chosen to choose the Visitor Pattern under behaviour design pattern. It is used when we have to perform an operation on a group of similar kinds of objects. With the help of the Visitor Pattern, we can move the operational logic from the objects to another class. For example, the search function in this assignment. When a user gives a search input, it will search through the file by visiting the existing code.

In the previous design, we have implemented the Implicit Invocation System because new components can be easily added while not affecting the interfaces of other system components. The search function is added in the Circular Shift module without affecting other components, as shown below (Figure 2).

```
if (!requiredWordsIncluded || requiredWords.contains(firstWord) || requiredWords.size() == 0)
    if (!wordsToIgnoreIncluded || !ignoredWords.contains(firstWord))
        if (!isSearch || searchWords.contains(firstWord))
            result.add(arrToString(words));
```

Figure 2. One more if-statement added into the Circular Shift module.

We treat this input as a keyword and search the list of files which contains it. During the Circular Shift, once the search word is spotted as the first word, the same way as previously done with ignored and required words, it will then be put into the results and then sent to the Alphabetizer.

4. Additional Information

This search implementation has nearly no changes to the existing code of the KWIC except for the additional search class and corresponding UI changes.

Search Word Not Found Message

```
Enter a word to search: book
Current search word(s): book
OurOutput0.txt has been updated.
No matches found for the word 'book'.
Please try searching again.
WordsToIgnore: OFF   RequiredWords: OFF
Enter a word to search:
```

Figure 3. Notification when no matches are found for the search word.

When none of the text files in *ListOfFiles0.in* contains the search word that is inputted by the user, it will result in an empty output file and the message above (Figure 3) will be shown to notify the user.

Also a searchTest case has been implemented in order to build successfully, it has to pass the test case.

```

1177 OurOutput0.txt has been updated.
1178 Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 sec - in sg.edu.nus.comp.tic3001.kwic_assignment.SearchTest
1179 Running sg.edu.nus.comp.tic3001.kwic_assignment.ExtensionsTest
1180 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec - in sg.edu.nus.comp.tic3001.kwic_assignment.ExtensionsTest
1181
1182 Results :
1183
1184 Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
1185
1186 [INFO] -----
1187 [INFO] BUILD SUCCESS
1188 [INFO] -----
1189 [INFO] Total time: 10.493 s
1190 [INFO] Finished at: 2020-04-16T16:52:42Z
1191 [INFO] -----
1192 The command "mvn test -B" exited with 0.
1193

```

Sonarqube was implemented during the code development. We are able to achieve 0 bugs and 0 Vulnerabilities in our code. To improve the code security.

