

Junit Readme

Avichal Chum

Import all files into the kwic project.

Run the AllTests.java. It will automatically run all other test files.

Sequence of events:

- Run the AllTests.java
- The first test automatically includes the 'abc.txt' filename, so ensure that it is present beforehand. You just need to press an enter after this.
- The code will run and give multiple java.IO.FileNotFoundException errors (which is what it should do).
- Again press enter to start the next test.
- Now enter 'cde.txt' without the quotes and press enter.
- Indexing will be done. Press enter.
- It runs another set of tests and for the next test asks for the filename again. Type 'cde.txt' and press enter.
- The previous was the program automatically ending test, so you did not see any results and did not need to press enter.
- For the final test, type 'cde.txt' again and press enter.
- It will mention that it did not find certain keywords which are hard coded to make the program fail.
- If everything goes right, you should get a coverage of 94.8%

Verifying tests via Emma:

A lot of conditions were noted that were not being executed. Junit tests were created to rectify this and convergence has increased to 94.8% from the initial 73%.

Almost all conditions are being executed in the code with some limitations. A convergence of 100% could not be achieved due to the following reasons:

Line 231-236 of Kwic.java; Line 51-52 of kwicSearch.java : No tests could be developed to cause IO.Exception to occur. My code handles the IO.Exception and java.IO.FileNotFoundException almost everywhere. The later is a subset of the former. While test cases could be easily developed to make the later fail, IO exceptions occur if the file opens but the read stream fails due to various reasons. Unfortunately, even after making multiple attempts with corrupt/encrypted and unknown files, this exception did not occur. Therefore, I have chosen not to include the Junit tests in which I attempt to cause the IO.Excpetion error. This also causes the code convergence rate to fall.

Line 179-183 of kwic.java ; Line 300-308 of kwicSearch.java: No tests have been developed that cause the program to retrieve a keyword before the index has been built. Since the program requires a user input before displaying the result (which tend of be slow), it ensures that the index is built in the meantime. However, if an extremely large file is supplied the user may press enter before the index is

built. Since I did not have such an extremely large file, the index not built error could not be tested via Junit.