Edward Mendoza

IT 372

October 25th, 2017

Assignment 07 Technical Summary

My understanding of functional decomposition is somewhat shaky, but I will do my best in this technical summary to address as many of the questions asked on this assignment’s Canvas page. For this technical summary, I looked at two Java files: one I created for this assignment, and another that I downloaded from Canvas. I only had to create one functional decomposition diagram for both these files, as I found that they were essentially the same in function. Both these files interact with text files that are given as inputs, and the lines of text that are within the inputted text files. Both files also read, edit, alphabetize, and output/print these text files and lines of text after editing them. The only difference is that the Canvas version of KWIC calls two methods before commencing the editing of a line of text; once in main() where circularShift() is called, and again in circularShift() itself, where circularShiftLine() is called. My own version of KWIC does not have its functionality implemented, but my intention was to make it call only one method to begin line editing. Given this difference, I do not believe either of these files will be particularly resilient to being changed. If I had to answer though, I’d probably say my KWIC file would be more difficult to change. If I had given it functionality, my sendToEnd() method probably would have been larger than circularShiftLine(). I say this because I find it more difficult to break a method apart into smaller methods than to combine methods into a larger method, although the concept of breaking methods apart to make smaller methods is very new to me. If these files were to interact with databases, rather than printing the lines of text after the circular shifting, the files would probably send the lines to the databases. I am not sure how the files would interact with databases if they were to send the text files to them, though. Changing these files to utilize a GUI instead of the console is simple enough, though—only an additional view class needs to be created.