Angelo Blanchard

4 October 2017

IT372 – Software Maintenance

Assignment 02: A “Simple” Program

Beginning with downloading the zip files from Canvas, I opened up the zip file and see 7 different Java class sitting together. Initially, I wanted to create the file in IntelliJ IDE, but I am very new to it and wasn’t sure on how to utilize Junit. I was about to do the project in Eclipse instead, when I had a classmate help me begin on IntelliJ. Needing a refresher on my Junit implementation, I refer to some old homework assignments from my Data Structures class. After running the test, a 3 of the 6 initial tests work, with the other 3 failing. After inspecting the error info and a few questions posed to classmates, I realize the file input file in.txt had not been created to read from. I created the file and threw in a few words. On the following test based on method writeToStream, I created a new output file, test3.out.

Afterwards, I worked on writing a test for each method, influenced by the tests already implemented in the code. I realized quickly that my tests will be similar to the pre-written tests, since they all rely on methods created in the Java files. I did create my own setup method so I would build one of the test ArrayLists using the @Before I learned in Josh’s Data Structures class. I got stuck on the sorting tests for awhile, because my personal ArrayList used “four, five, six” as it’s elements opposed to the pre-written “one, two, three”. Now, the pre-written had a second ArrayList of “one, three, two”, and I had assumed sort would sort in numerical order(which, when you remember that you’re working with Strings, is clearly incorrect). After a lot more confusion, I realize it’s based off the String value, and also that four, five, six would have a different order than one, three two(the correct order by String value would be five, four, six). After figuring out that, it was fairly smooth sailing in implementing the rest of the test methods.

The hardest part was trying to figure out what was going on in the methods. Also, I realized that many of the classes didn’t apply to the assignment, so were mostly treated as filler. The coding style was also a lot different, with odd spacing. To some extent, I tried to follow the style of the pre-written, but the spacing style stayed as my own. It was tempting to make the code match my own style completely, but I figure I would have to follow someone else’s style in the real world. My code constantly followed their style, as mentioned, since their tests worked in tandem with the methods.