Leah Briggs

April 25th 2018

Assignment 7

Task 1:

Size:

1. Lines of Code: 22,539 LOC.
2. Largest File: HTMLEditor.java with 2144 LOC.
3. It looks like it counts any LOC that is not white space or a comment. However, it does count packages, imports, brackets and code statements (method 1 from notes)

Cohesion:

1. LCOM2 is the percentage of methods that don’t access an attribute, averaged over all attributes.
2. The highest LCOM would be ResourceTypeDialog.java with 1.1 as the mean. If we’re just looking in main.java.memoranda then the highest LCOM is TaskListImpl.java with 0.679 as the mean.

Complexity:

1. The Cyclomatic complexity of main is 1.746
2. The worst complexity is ALTHTMLWriter.java with 4.292. In the main package the worst complexity is EventsManager.java with 2.5.
3. The complexity before any changes in main is 1.746. Once I edited EventsManager.java it became 1.734. I did this by removing the unnecessary if statements which reduced the number of paths in the class.

Package-level Coupling:

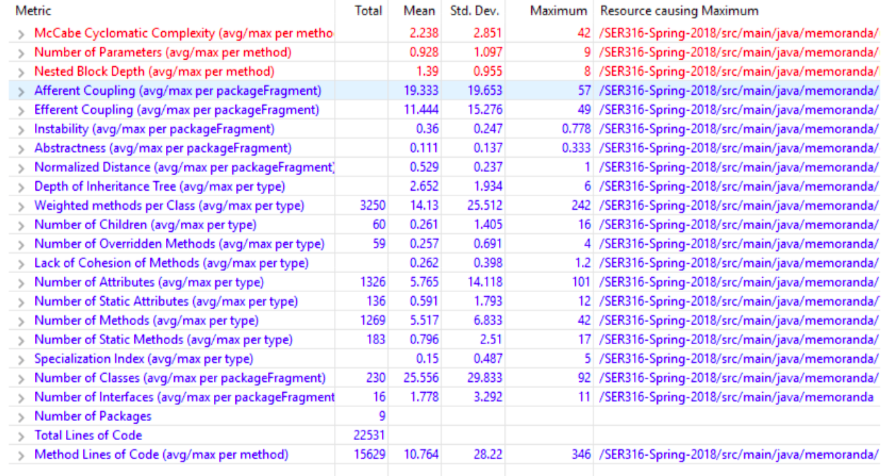
1. Afferent coupling measures how many classes outside of the reference package depend on those within the reference package. Efferent coupling measures how many classes outside of the reference package that classes in the reference package depends on.
2. Main.java.memoranda.util has worst Afferent Coupling with 57
3. Main.java.memoranda.ui has the worst Efferent Coupling with 49

Worst Quality:

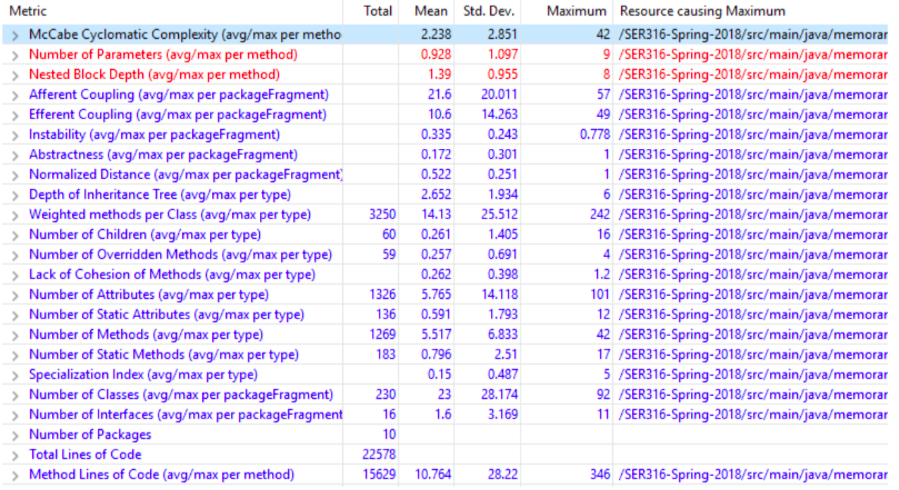
1. AltHTMLWriter.java is the worst quality class in the program. It has a lot of bad metrics numbers and violates them to the highest degree. It has a 4.292 complexity, 1.406 in parameters, 2.302 in Nested Block Depth.

Task 2:

Before Refactoring:



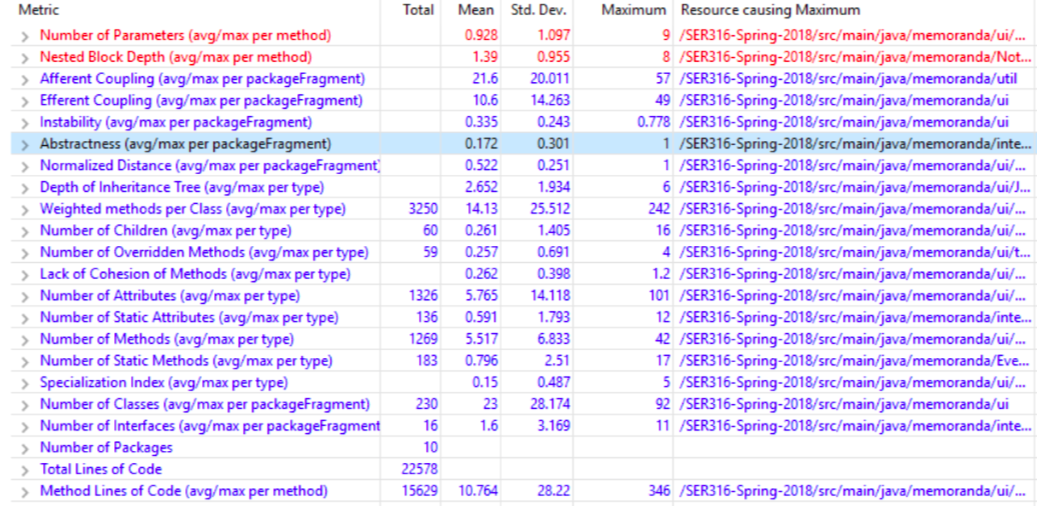
After refactoring: (Task 2 Step 8)



Afferent coupling is greater, efferent coupling is reduced, instability is lower and abstractness is higher.

Task 3:

1. The code smell was in main.java.memoranda.EventImpl.java, specifically in the getTime() method, there was commented out dead code (the comments and dead code smells). To fix this I removed the code.
2. The code smell between classes was in main.java.memoranda.ui.HTMLEditor.java. It’s a code smell known as Indecent Exposure, where variables are set as public. I fixed these and fixed the classes which accessed this data.



1. These fixes didn’t really change the metrics, the fixes I did weren’t very extensive.