

Danielle Panfili

Metrics and Measures

April 23<sup>rd</sup> 2018

## Homework: Metrics and Measures

### Task 1:

#### Size:

1. Total lines of code: 22,539
2. Largest Code File: HTMLEditor.java with 2144 Lines of code. But since the project refers to main.java.memoranda the largest file in that package is EventsManager.java with 329 lines of code.
3. It looks like it will count any line of code that is not a white space or comment. It will count packages, imports, brackets and code statements. It uses Method 1 of counting each statement as 1 and everything else as 0.

#### Cohesion

1. LCOM2 is the percentage of methods that do not access a attribute averaged over all attributes. The implementation is as follows  $LCOM2 = 1 - \frac{\sum(mA)}{(m * a)}$ .
2. If we are only looking at main.java.memoranda, then the highest LCOM is the TaskListImpl.java with .679 as the Mean. Looking else where in the memoranda program the highest LCOM would be ResourceTypeDialog.java with 1.1 as the mean.

#### Complexity

1. The Cyclomatic complexity of the main package is 1.746
2. In the main package, the worst complexity is EventsManager.java with 2.5 but overall the worst complexity is ALTHTMLWriter.java with 4.292
3. The complexity before any changes was 1.738. I edited EventsManager.java to reduce complexity. I removed some if statements and replaced with methods. This brought EventsManager.java to 2.444. and then I took out some nested for loops and replaced with methods in NoteListImpl. When you have nested if statements or nested for loops it causes the complexity to be higher. If you can replace with methods instead, it reduces complexity.

#### Package-level Coupling

1. Afferent coupling measures how many classes depend on a given class while Efferent coupling measures the number of classes on which a given class depends. Efferent coupling means that these classes will receive effects of changes or defects in other classes. Afferent Coupling classes affect other classes when changes are made. Summarized by source - <http://www.informit.com/articles/article.aspx?p=1561879&seqNum=3>
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. I believe the class with the worst quality is the AltHTMLWriter.java. The reason I believe this is because in McCabe Cyclomatic Complexity it is shown as the worst with high numbers. It is also the worst in Number of parameters. Since both of these are in red, this shows that these are the most violated. Furthermore, it has 9 classes and 1,460 lines of code. All in all, this I believe is one of the worst quality classes in the program.

## Task 2:

Before Refactoring:

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)		2.236	2.827	42	/SER316-Spring-2018-master/src/main/java/memora...	setTableProperties
> Number of Parameters (avg/max per method)		0.93	1.096	9	/SER316-Spring-2018-master/src/main/java/memora...	setImageProperties
> Nested Block Depth (avg/max per method)		1.391	0.949	8	/SER316-Spring-2018-master/src/main/java/memora...	getNotesForPeriod
> Afferent Coupling (avg/max per packageFragment)		19.333	19.653	57	/SER316-Spring-2018-master/src/main/java/memora...	
> Efferent Coupling (avg/max per packageFragment)		11.444	15.276	49	/SER316-Spring-2018-master/src/main/java/memora...	
> Instability (avg/max per packageFragment)		0.36	0.247	0.778	/SER316-Spring-2018-master/src/main/java/memora...	
> Abstractness (avg/max per packageFragment)		0.111	0.137	0.333	/SER316-Spring-2018-master/src/main/java/memora...	
> Normalized Distance (avg/max per packageFragment)		0.529	0.237	1	/SER316-Spring-2018-master/src/main/java/memora...	
> Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/SER316-Spring-2018-master/src/main/java/memora...	
> Weighted methods per Class (avg/max per type)	3265	14.196	25.622	242	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Children (avg/max per type)	60	0.261	1.405	16	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/SER316-Spring-2018-master/src/main/java/memora...	
> Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Methods (avg/max per type)	1273	5.535	6.857	42	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Static Methods (avg/max per type)	187	0.813	2.633	21	/SER316-Spring-2018-master/src/main/java/memora...	
> Specialization Index (avg/max per type)		0.15	0.487	5	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Classes (avg/max per packageFragment)	230	25.556	29.833	92	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Interfaces (avg/max per packageFragment)	16	1.778	3.292	11	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Packages	9					
> Total Lines of Code	22582					
> Method Lines of Code (avg/max per method)	15664	10.729	28.138	346	/SER316-Spring-2018-master/src/main/java/memora...	jblnit

After Refactoring

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)		2.236	2.827	42	/SER316-Spring-2018-master/src/main/java/memora...	setTableProperties
> Number of Parameters (avg/max per method)		0.93	1.096	9	/SER316-Spring-2018-master/src/main/java/memora...	setImageProperties
> Nested Block Depth (avg/max per method)		1.391	0.949	8	/SER316-Spring-2018-master/src/main/java/memora...	getNotesForPeriod
> Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/SER316-Spring-2018-master/src/main/java/memora...	
> Efferent Coupling (avg/max per packageFragment)		10.6	14.263	49	/SER316-Spring-2018-master/src/main/java/memora...	
> Instability (avg/max per packageFragment)		0.335	0.243	0.778	/SER316-Spring-2018-master/src/main/java/memora...	
> Abstractness (avg/max per packageFragment)		0.172	0.301	1	/SER316-Spring-2018-master/src/main/java/memora...	
> Normalized Distance (avg/max per packageFragment)		0.522	0.251	1	/SER316-Spring-2018-master/src/main/java/memora...	
> Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/SER316-Spring-2018-master/src/main/java/memora...	
> Weighted methods per Class (avg/max per type)	3265	14.196	25.622	242	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Children (avg/max per type)	60	0.261	1.405	16	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/SER316-Spring-2018-master/src/main/java/memora...	
> Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Methods (avg/max per type)	1273	5.535	6.857	42	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Static Methods (avg/max per type)	187	0.813	2.633	21	/SER316-Spring-2018-master/src/main/java/memora...	
> Specialization Index (avg/max per type)		0.15	0.487	5	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Classes (avg/max per packageFragment)	230	23	28.174	92	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Packages	10					
> Total Lines of Code	22629					
> Method Lines of Code (avg/max per method)	15664	10.729	28.138	346	/SER316-Spring-2018-master/src/main/java/memora...	jblnit

After Refactoring Changes:

The means that went down: Efferent Coupling – because of adding another package, Normalized Distance, Number of Classes, Number of Interfaces, Instability.

The means that went up: Afferent Coupling – because of adding another package, Abstractness

## Task 3:

1. Code Smell within class: main.java.memoranda.TaskListImpl.java: I saw a long method – way more than 10 lines, so I created 2 smaller methods to go into the large method. The method had some for loops in the code that could be reduced to a method. The one draw back is that the methods do have 4 parameters but it is still not large enough to be a different code smell.
2. Code Smell between class: main.java.memoranda.History.java. According to <https://blog.codinghorror.com/code-smells/> There is a code smell called Indecent Exposure. This is when items that are made public should be private. Usually class variables. So in this file, there were all static methods that did not have private in front of it. So to refactor, I made them private.
3. After Refactor:

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)		2.235	2.825	42	/SER316-Spring-2018-master/src/main/java/memora...	setTableProperties
> Number of Parameters (avg/max per method)		0.934	1.101	9	/SER316-Spring-2018-master/src/main/java/memora...	setImageProperties
> Nested Block Depth (avg/max per method)		1.391	0.949	8	/SER316-Spring-2018-master/src/main/java/memora...	getNotesForPeriod
> Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/SER316-Spring-2018-master/src/main/java/memora...	
> Efferent Coupling (avg/max per packageFragment)		10.6	14.263	49	/SER316-Spring-2018-master/src/main/java/memora...	
> Instability (avg/max per packageFragment)		0.335	0.243	0.778	/SER316-Spring-2018-master/src/main/java/memora...	
> Abstractness (avg/max per packageFragment)		0.172	0.301	1	/SER316-Spring-2018-master/src/main/java/memora...	
> Normalized Distance (avg/max per packageFragment)		0.522	0.251	1	/SER316-Spring-2018-master/src/main/java/memora...	
> Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/SER316-Spring-2018-master/src/main/java/memora...	
> Weighted methods per Class (avg/max per type)	3267	14.204	25.634	242	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Children (avg/max per type)	60	0.261	1.405	16	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/SER316-Spring-2018-master/src/main/java/memora...	
> Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Methods (avg/max per type)	1275	5.543	6.079	42	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Static Methods (avg/max per type)	187	0.813	2.633	21	/SER316-Spring-2018-master/src/main/java/memora...	
> Specialization Index (avg/max per type)		0.15	0.487	5	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Classes (avg/max per packageFragment)	230	23	28.174	92	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/SER316-Spring-2018-master/src/main/java/memora...	
> Number of Packages	10					
> Total Lines of Code	22636					
> Method Lines of Code (avg/max per method)	15667	10.716	28.118	346	/SER316-Spring-2018-master/src/main/java/memora...	jblnit

4. After refactoring the code smells, I noticed a few changes. My McCabe Cyclomatic Complexity went down barely but it went down. However, I noticed that Number of parameters, Weighted methods, Depth Inheritance and Attributes went up in value. I will assume this is more to do with the code smell within the class. I created more methods that relied on other methods resulting in heavier weighted methods. Furthermore, more parameters are needed in the methods that I split up causing the number of parameters to increase.