Brian Coley – SER 316 – 2019 Spring B

Task 1 – Metrics

Size:

1. 22,539
2. HTMLEditor.java
3. “Method 1: Count each statement as 1, everything else 0”
   Basically each line of code that wasn’t blank or a comment was counted as 1.

Cohesion:

1. From what I could gather, LCOM2 is
   P = Number of pairs of methods that do not share attributes
   Q = Number of pairs of methods that share attributes

   $$\text{LCOM2} = \begin{cases} P-Q, & \text{if } P-Q >= 0 \\ 0, & \text{else} \end{cases}$$
2. TaskListImpl has the highest cohesion and this is likely due to the class being an implementation class and as such a lot of the methods work together to implement the business logic behind the scenes for making a task.

Complexity:

1. 1.746
2. EventsManager.java with 2.5
3. I extracted methods from the EventsManager.java class creating several smaller methods to take the place of the if else if statements of the getRepeatableEventsForDate method. Each of the if else if statements had several items they would take care of (some even had sub if statements embedded) and by moving these out into their own method it becomes easier to follow what is happening in the entire block of if else if section. This reduced the complexity by making the bigger method have fewer branches to follow as some of the branches now just call a sub method. The complexity for the entire class went from 3.353 to 2.333.

Package-level Coupling:

1. Afferent is the number of classes within other packages that depend on a given package. Efferent is the number of classes in other packages that a given package depend on. Basically it is characteristic of the classes dependent or independent nature in relation to other classes.
2. The worst afferent score is main.java.memoranda.util with 57 (a bunch of classes depend on it)
3. The worst efferent score is main.java.memoranda.ui with 49 (it depends on a bunch of classes)

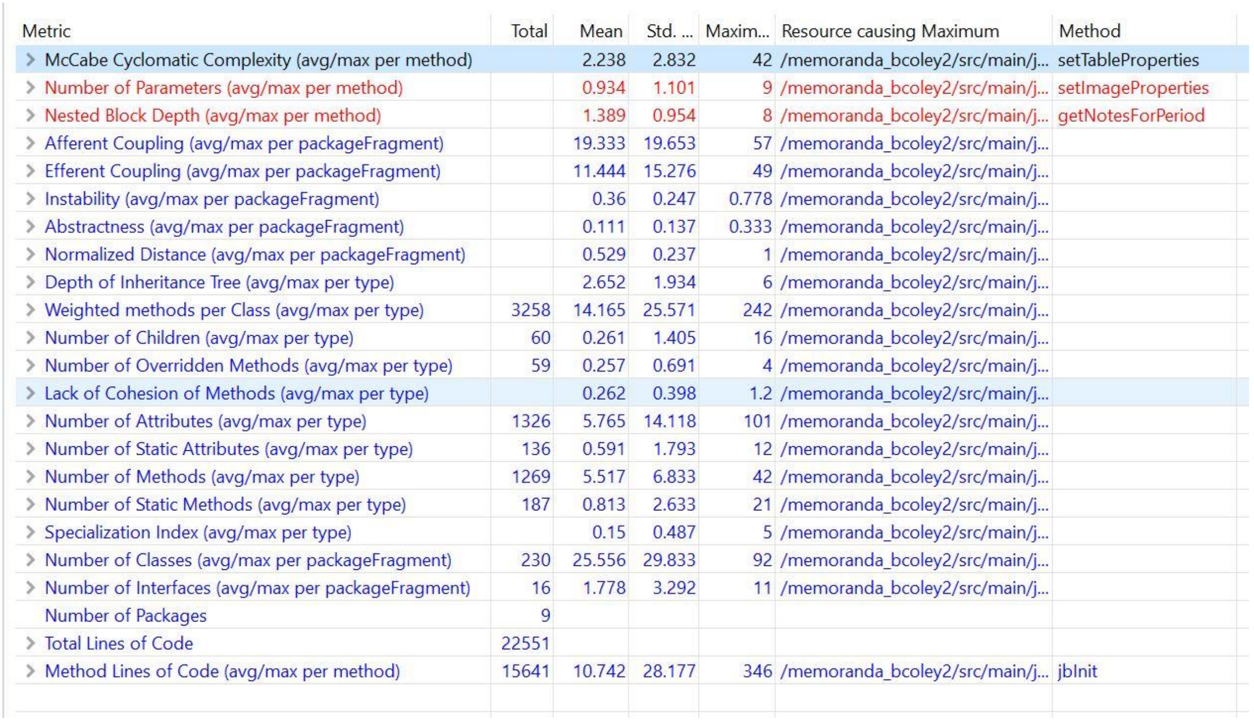
Worst Quality:

I believe AltHTMLWriter is one of the worst classes. It has a lower lack of cohesion score with 0.273, but the McCabe Cyclomatic Complexity score is a high mean of 4.292. This suggests to me that the code is a complex mess of code. It is also located in the main.java.memoranda.ui.htmleditor package which has an Efferent score of 12.

Brian Coley – SER 316 – 2019 Spring B

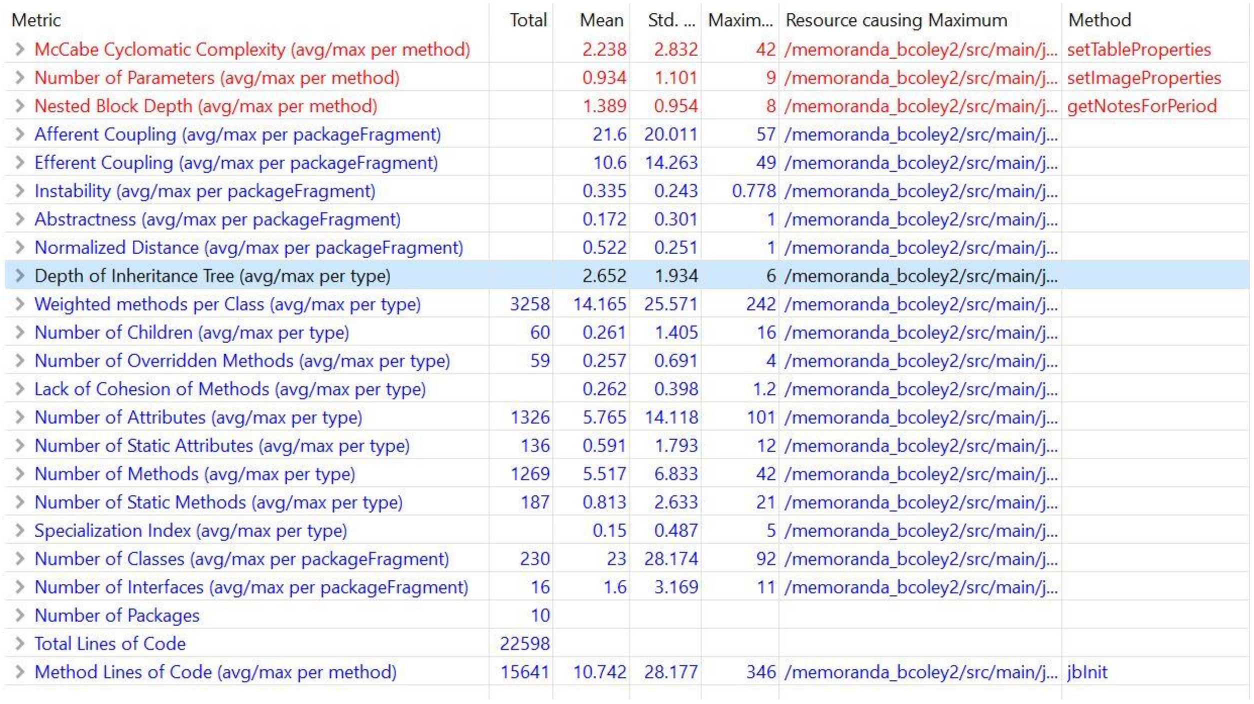
Task 2 – Refactoring

Step 2.1:



Metric	Total	Mean	Std. ...	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.238	2.832	42	/memoranda_bcoley2/src/main/j...	setTableProperties
Number of Parameters (avg/max per method)		0.934	1.101	9	/memoranda_bcoley2/src/main/j...	setImageProperties
Nested Block Depth (avg/max per method)		1.389	0.954	8	/memoranda_bcoley2/src/main/j...	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		19.333	19.653	57	/memoranda_bcoley2/src/main/j...	
Efferent Coupling (avg/max per packageFragment)		11.444	15.276	49	/memoranda_bcoley2/src/main/j...	
Instability (avg/max per packageFragment)		0.36	0.247	0.778	/memoranda_bcoley2/src/main/j...	
Abstractness (avg/max per packageFragment)		0.111	0.137	0.333	/memoranda_bcoley2/src/main/j...	
Normalized Distance (avg/max per packageFragment)		0.529	0.237	1	/memoranda_bcoley2/src/main/j...	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/memoranda_bcoley2/src/main/j...	
Weighted methods per Class (avg/max per type)	3258	14.165	25.571	242	/memoranda_bcoley2/src/main/j...	
Number of Children (avg/max per type)	60	0.261	1.405	16	/memoranda_bcoley2/src/main/j...	
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/memoranda_bcoley2/src/main/j...	
Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/memoranda_bcoley2/src/main/j...	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/memoranda_bcoley2/src/main/j...	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/memoranda_bcoley2/src/main/j...	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/memoranda_bcoley2/src/main/j...	
Number of Static Methods (avg/max per type)	187	0.813	2.633	21	/memoranda_bcoley2/src/main/j...	
Specialization Index (avg/max per type)		0.15	0.487	5	/memoranda_bcoley2/src/main/j...	
Number of Classes (avg/max per packageFragment)	230	25.556	29.833	92	/memoranda_bcoley2/src/main/j...	
Number of Interfaces (avg/max per packageFragment)	16	1.778	3.292	11	/memoranda_bcoley2/src/main/j...	
Number of Packages	9					
Total Lines of Code	22551					
Method Lines of Code (avg/max per method)	15641	10.742	28.177	346	/memoranda_bcoley2/src/main/j...	jbInit

Step 2.7:



Metric	Total	Mean	Std. ...	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.238	2.832	42	/memoranda_bcoley2/src/main/j...	setTableProperties
Number of Parameters (avg/max per method)		0.934	1.101	9	/memoranda_bcoley2/src/main/j...	setImageProperties
Nested Block Depth (avg/max per method)		1.389	0.954	8	/memoranda_bcoley2/src/main/j...	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/memoranda_bcoley2/src/main/j...	
Efferent Coupling (avg/max per packageFragment)		10.6	14.263	49	/memoranda_bcoley2/src/main/j...	
Instability (avg/max per packageFragment)		0.335	0.243	0.778	/memoranda_bcoley2/src/main/j...	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/memoranda_bcoley2/src/main/j...	
Normalized Distance (avg/max per packageFragment)		0.522	0.251	1	/memoranda_bcoley2/src/main/j...	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/memoranda_bcoley2/src/main/j...	
Weighted methods per Class (avg/max per type)	3258	14.165	25.571	242	/memoranda_bcoley2/src/main/j...	
Number of Children (avg/max per type)	60	0.261	1.405	16	/memoranda_bcoley2/src/main/j...	
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/memoranda_bcoley2/src/main/j...	
Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/memoranda_bcoley2/src/main/j...	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/memoranda_bcoley2/src/main/j...	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/memoranda_bcoley2/src/main/j...	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/memoranda_bcoley2/src/main/j...	
Number of Static Methods (avg/max per type)	187	0.813	2.633	21	/memoranda_bcoley2/src/main/j...	
Specialization Index (avg/max per type)		0.15	0.487	5	/memoranda_bcoley2/src/main/j...	
Number of Classes (avg/max per packageFragment)	230	23	28.174	92	/memoranda_bcoley2/src/main/j...	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/memoranda_bcoley2/src/main/j...	
Number of Packages	10					
Total Lines of Code	22598					
Method Lines of Code (avg/max per method)	15641	10.742	28.177	346	/memoranda_bcoley2/src/main/j...	jbInit

Step 2.8: The differences I noticed were the Afferent and Efferent numers. Afferent rose from 19.333 to 21.6 and Efferent dropped from 11.444 to 10.6. This changed due to pulling the interfaces that the

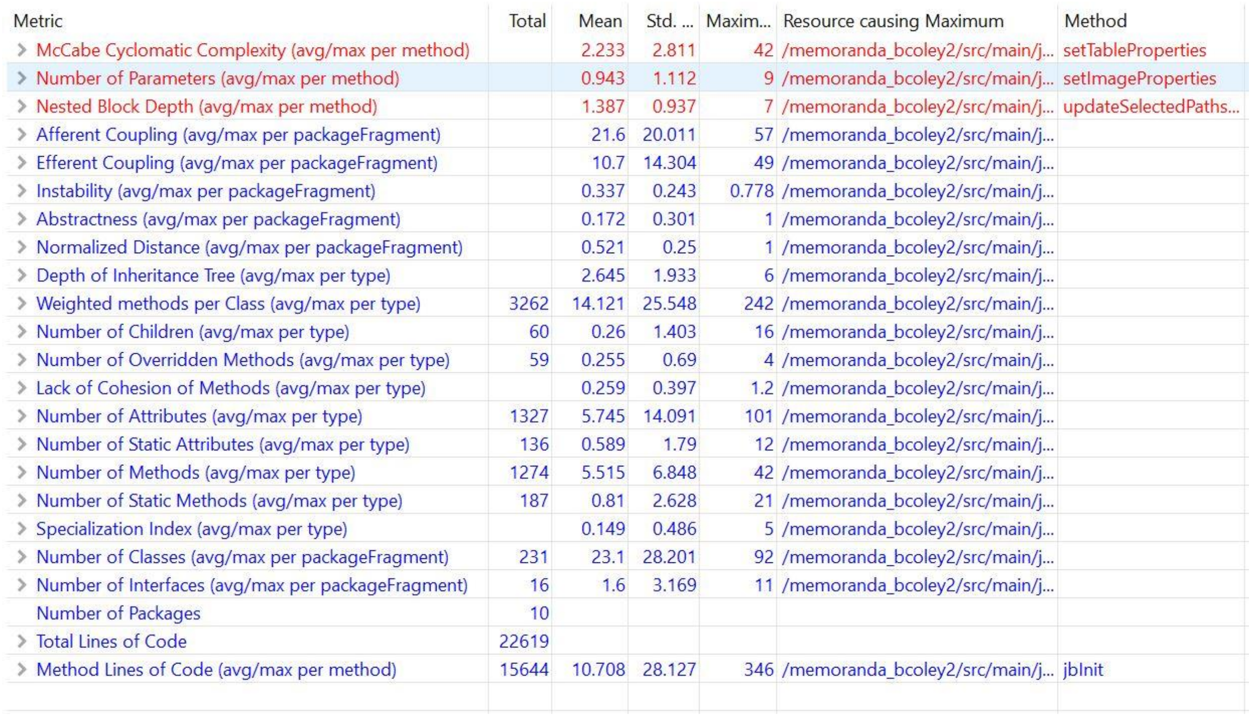
main.memoranda.main package directly relied on and moved it to its own class thus raising the Afferent score, but the same movement also resulted in fewer classes needing to call the main.memoranda.main package thus reducing the Efferent score. Overall I believe this is better because even though the scores may mostly even out there is a central location to find the interface classes.

Task 3 – Code Smells and Refactor

3.1: Large Method: NotesListImpl – Broke up getNotesForPeriod method (a several deep if statement tree) and made each sub if statement a separate function to be called. This code made following the logic hard to do, but by splitting out to smaller functions you can follow each if in isolation thus making it an easier path to follow.

3.2: Divergent Change: NotesListImpl – Factored out a class so that the NotesListImpl could call the extracted class as necessary. This should make the larger NotesListImpl class easier to manage/read overall and will keep extra features out of it.

3.3:



Metric	Total	Mean	Std. ...	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.233	2.811	42	/memoranda_bcoley2/src/main/j...	setTableProperties
Number of Parameters (avg/max per method)		0.943	1.112	9	/memoranda_bcoley2/src/main/j...	setImageProperties
Nested Block Depth (avg/max per method)		1.387	0.937	7	/memoranda_bcoley2/src/main/j...	updateSelectedPaths...
Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/memoranda_bcoley2/src/main/j...	
Efferent Coupling (avg/max per packageFragment)		10.7	14.304	49	/memoranda_bcoley2/src/main/j...	
Instability (avg/max per packageFragment)		0.337	0.243	0.778	/memoranda_bcoley2/src/main/j...	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/memoranda_bcoley2/src/main/j...	
Normalized Distance (avg/max per packageFragment)		0.521	0.25	1	/memoranda_bcoley2/src/main/j...	
Depth of Inheritance Tree (avg/max per type)		2.645	1.933	6	/memoranda_bcoley2/src/main/j...	
Weighted methods per Class (avg/max per type)	3262	14.121	25.548	242	/memoranda_bcoley2/src/main/j...	
Number of Children (avg/max per type)	60	0.26	1.403	16	/memoranda_bcoley2/src/main/j...	
Number of Overridden Methods (avg/max per type)	59	0.255	0.69	4	/memoranda_bcoley2/src/main/j...	
Lack of Cohesion of Methods (avg/max per type)		0.259	0.397	1.2	/memoranda_bcoley2/src/main/j...	
Number of Attributes (avg/max per type)	1327	5.745	14.091	101	/memoranda_bcoley2/src/main/j...	
Number of Static Attributes (avg/max per type)	136	0.589	1.79	12	/memoranda_bcoley2/src/main/j...	
Number of Methods (avg/max per type)	1274	5.515	6.848	42	/memoranda_bcoley2/src/main/j...	
Number of Static Methods (avg/max per type)	187	0.81	2.628	21	/memoranda_bcoley2/src/main/j...	
Specialization Index (avg/max per type)		0.149	0.486	5	/memoranda_bcoley2/src/main/j...	
Number of Classes (avg/max per packageFragment)	231	23.1	28.201	92	/memoranda_bcoley2/src/main/j...	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/memoranda_bcoley2/src/main/j...	
Number of Packages	10					
Total Lines of Code	22619					
Method Lines of Code (avg/max per method)	15644	10.708	28.127	346	/memoranda_bcoley2/src/main/j...	jbInit

3.4: The McCabe Cyclomatic Complexity changed from 2.238 to 2.233. I believe this change occurred because most of my changes involved factoring out methods and classes which ultimately reduced their overall complexity.

Code can be found at: https://github.com/briancoley/memoranda_bcoley2/tree/Task1