Nicholas Okerberg
Arizona State University
SER316 Spring 2018 Session B
Software Engineering Construction
Assignment 7

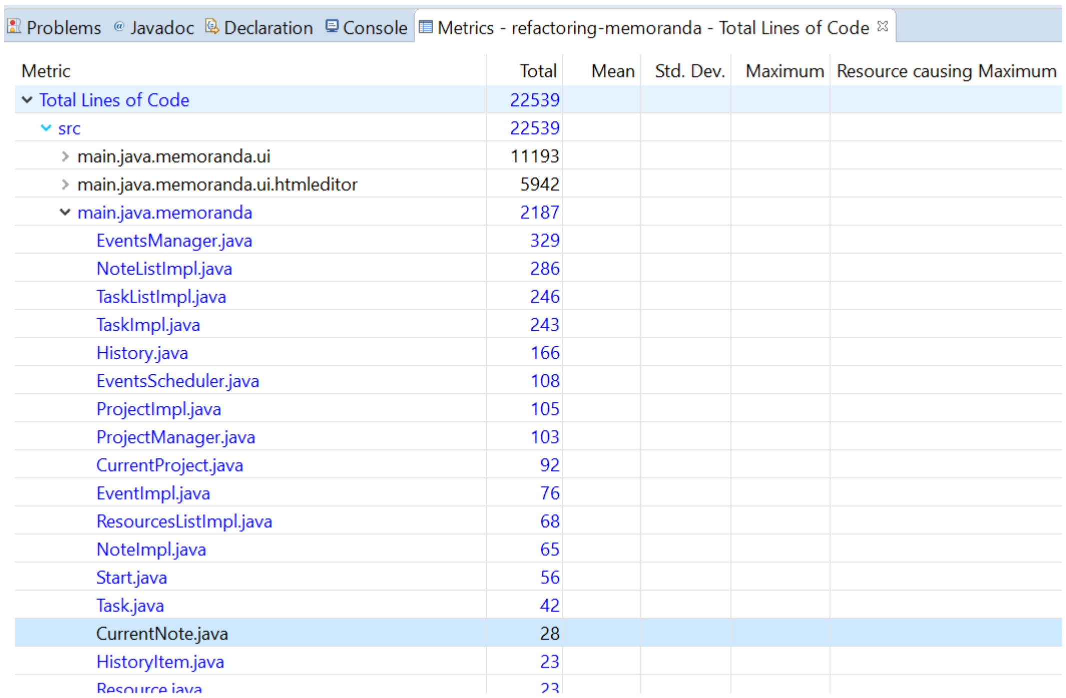
Task 1: Metric Evaluation

Size

1) The total LOC for the project is: 22,539.

2) The largest single code file in the project is HTMLEditor.java at 2,144 total LOC.

3) The CurrentNote.java file shows that it has a Total LOC of 28 as shown in the diagram below. The method used to determine this is Method #1 from the Lecture video/notes. Using this method, each of the statements is counted as 1 and everything else 0. The comments and whitespace is 0.



Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum
Total Lines of Code	22539				
src	22539				
main.java.memoranda.ui	11193				
main.java.memoranda.ui.htmleditor	5942				
main.java.memoranda	2187				
EventsManager.java	329				
NoteListImpl.java	286				
TaskListImpl.java	246				
TaskImpl.java	243				
History.java	166				
EventsScheduler.java	108				
ProjectImpl.java	105				
ProjectManager.java	103				
CurrentProject.java	92				
EventImpl.java	76				
ResourcesListImpl.java	68				
NoteImpl.java	65				
Start.java	56				
Task.java	42				
CurrentNote.java	28				
HistoryItem.java	23				
Resource.java	23				

Cohesion

1) LCOM2 is the Henderson-Sellers method of determining the extent to which a module has only a single responsibility. In other words, it's used to determine how cohesive a module is.

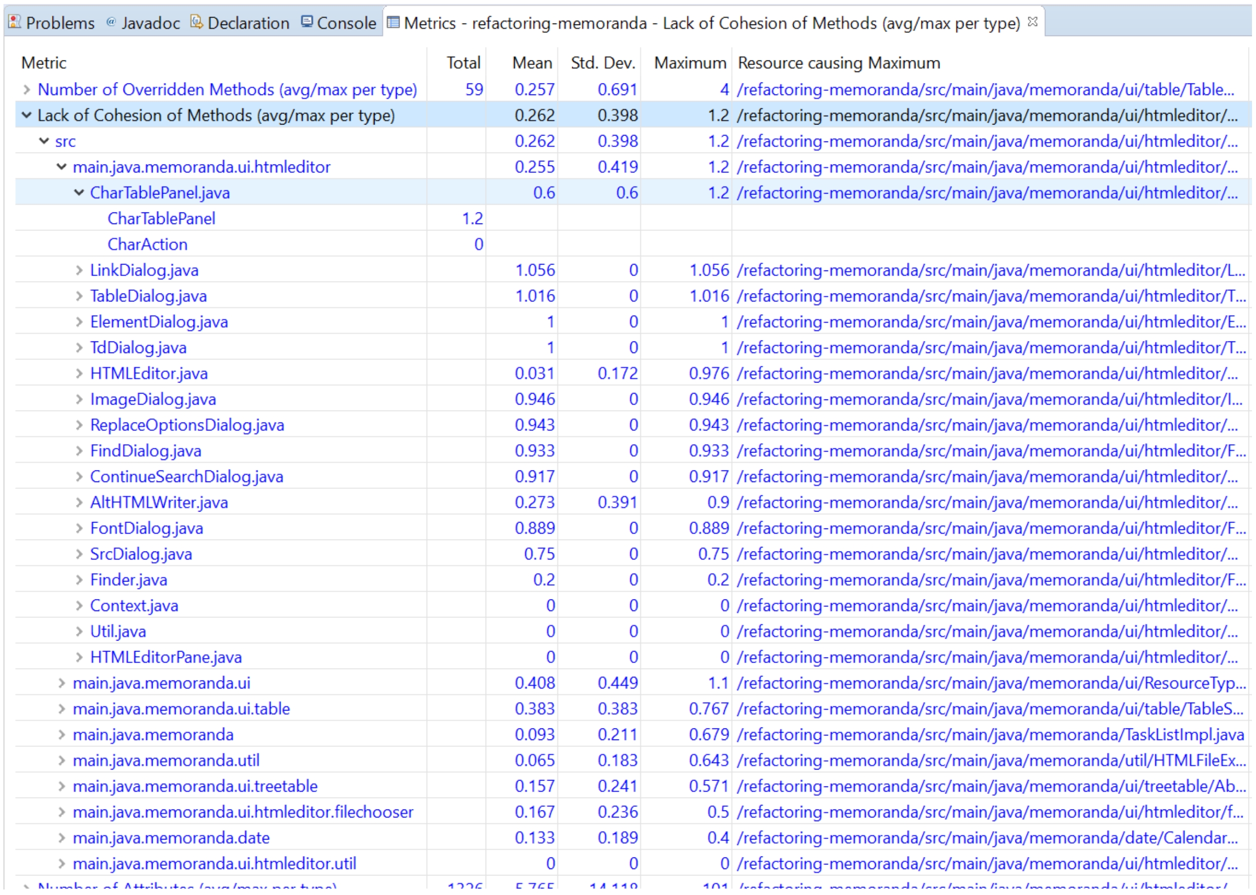
   LCOM2 is calculated by using the formula **(1 – sum(mA) / (m*a))** where:

   m = The number of methods in the class.
   a = The number of variables, or attributes, in the class, whether shared or not.
   mA = The number of methods that access a variable (or attribute).
   sum(mA) = The sum of methods that access a variable (or attribute).

   The formula gives the percentage of the methods that do not access specific attributes averaged over all attributes in the class. If the m or a values are 0, then LCOM2 is undefined and displayed as 0. A low value indicates higher cohesion and a class that is well-designed. A higher value indicates lower cohesion.

   Reference: http://www.aivosto.com/project/help/pm-oo-cohesion.html

2) The class that has the highest Cohesion is CharTablePanel.java as shown below. I believe this is because The CharTablePanel class doesn't rely on any other class in the project, only Swing various swing components. However one other class HTMLEditor.java, rely on CharTablePanel.

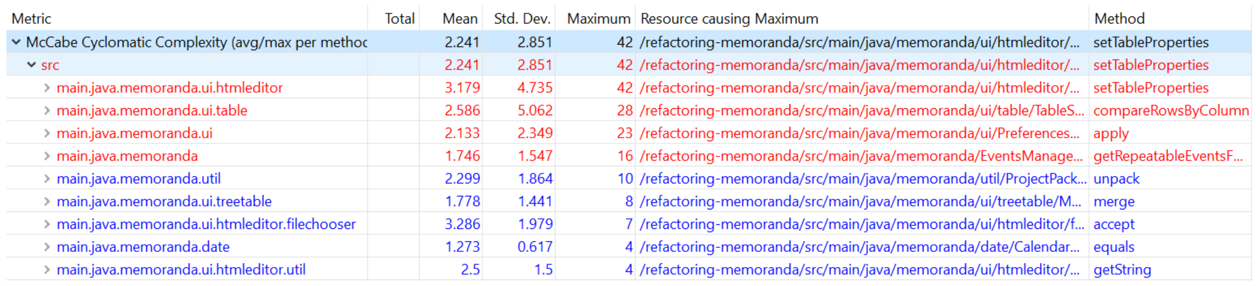


Problems Javadoc Declaration Console Metrics - refactoring-memoranda - Lack of Cohesion of Methods (avg/max per type)

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/refactoring-memoranda/src/main/java/memoranda/ui/table/Table...
Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
src		0.262	0.398	1.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
main.java.memoranda.ui.htmleditor		0.255	0.419	1.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
CharTablePanel.java		0.6	0.6	1.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
CharTablePanel	1.2				
CharAction	0				
LinkDialog.java		1.056	0	1.056	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/L...
TableDialog.java		1.016	0	1.016	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/T...
ElementDialog.java		1	0	1	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/E...
TdDialog.java		1	0	1	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/T...
HTMLEditor.java		0.031	0.172	0.976	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
ImageDialog.java		0.946	0	0.946	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/I...
ReplaceOptionsDialog.java		0.943	0	0.943	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
FindDialog.java		0.933	0	0.933	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/F...
ContinueSearchDialog.java		0.917	0	0.917	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
AltHTMLWriter.java		0.273	0.391	0.9	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
FontDialog.java		0.889	0	0.889	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/F...
SrcDialog.java		0.75	0	0.75	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
Finder.java		0.2	0	0.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/F...
Context.java		0	0	0	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
Util.java		0	0	0	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
HTMLEditorPane.java		0	0	0	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
main.java.memoranda.ui		0.408	0.449	1.1	/refactoring-memoranda/src/main/java/memoranda/ui/ResourceTyp...
main.java.memoranda.ui.table		0.383	0.383	0.767	/refactoring-memoranda/src/main/java/memoranda/ui/table/TableS...
main.java.memoranda		0.093	0.211	0.679	/refactoring-memoranda/src/main/java/memoranda/TaskListImpl.java
main.java.memoranda.util		0.065	0.183	0.643	/refactoring-memoranda/src/main/java/memoranda/util/HTMLFileEx...
main.java.memoranda.ui.treetable		0.157	0.241	0.571	/refactoring-memoranda/src/main/java/memoranda/ui/treetable/Ab...
main.java.memoranda.ui.htmleditor.filechooser		0.167	0.236	0.5	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/f...
main.java.memoranda.date		0.133	0.189	0.4	/refactoring-memoranda/src/main/java/memoranda/date/Calendar...
main.java.memoranda.ui.htmleditor.util		0	0	0	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/

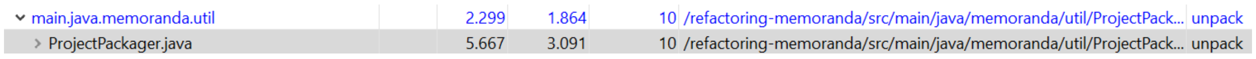
Complexity

1) The mean Cyclomatic Complexity in the main package is 2.241 with the maximum being 42 as shown in the snippet below:



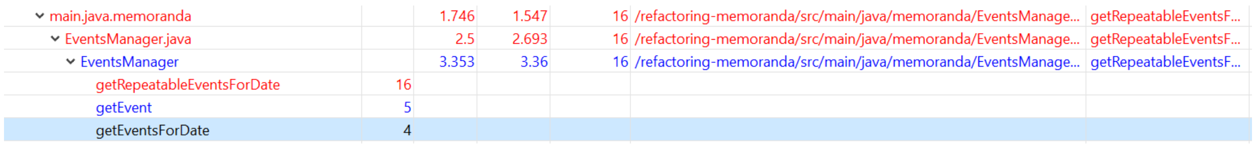
Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per methoc		2.241	2.851	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...	setTableProperties
src		2.241	2.851	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...	setTableProperties
main.java.memoranda.ui.htmleditor		3.179	4.735	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...	setTableProperties
main.java.memoranda.ui.table		2.586	5.062	28	/refactoring-memoranda/src/main/java/memoranda/ui/table/TableS...	compareRowsByColumn
main.java.memoranda.ui		2.133	2.349	23	/refactoring-memoranda/src/main/java/memoranda/ui/Preferences...	apply
main.java.memoranda		1.746	1.547	16	/refactoring-memoranda/src/main/java/memoranda/EventsManage...	getRepeatableEventsF...
main.java.memoranda.util		2.299	1.864	10	/refactoring-memoranda/src/main/java/memoranda/util/ProjectPack...	unpack
main.java.memoranda.ui.treetable		1.778	1.441	8	/refactoring-memoranda/src/main/java/memoranda/ui/treetable/M...	merge
main.java.memoranda.ui.htmleditor.filechooser		3.286	1.979	7	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/f...	accept
main.java.memoranda.date		1.273	0.617	4	/refactoring-memoranda/src/main/java/memoranda/date/Calendar...	equals
main.java.memoranda.ui.htmleditor.util		2.5	1.5	4	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/...	getString

2) The class that has the worst McCabe Cyclomatic Complexity (CC) is ProjectPackager.java at 5.667 as shown in the snippet below:



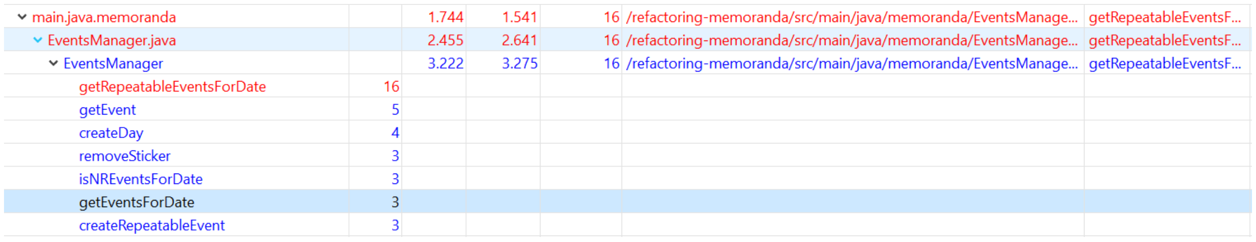
main.java.memoranda.util		2.299	1.864	10	/refactoring-memoranda/src/main/java/memoranda/util/ProjectPack...	unpack
ProjectPackager.java		5.667	3.091	10	/refactoring-memoranda/src/main/java/memoranda/util/ProjectPack...	unpack

3) Reduce the Cyclomatic Complexity for any Java class in the project. I chose the EventsManager.java class with a mean CC of 3.353 as shown below.



main.java.memoranda		1.746	1.547	16	/refactoring-memoranda/src/main/java/memoranda/EventsManage...	getRepeatableEventsF...
EventsManager.java		2.5	2.693	16	/refactoring-memoranda/src/main/java/memoranda/EventsManage...	getRepeatableEventsF...
EventsManager		3.353	3.36	16	/refactoring-memoranda/src/main/java/memoranda/EventsManage...	getRepeatableEventsF...
getRepeatableEventsForDate	16					
getEvent	5					
getEventsForDate	4					

I updated the getEventsForDate method and removed some for loop logic that builds a vector. Instead, I created a private method that takes in the date and builds out the vector to return it. I basically split up a larger method into two smaller ones to reduce complexity of the overall class.



main.java.memoranda		1.744	1.541	16	/refactoring-memoranda/src/main/java/memoranda/EventsManage...	getRepeatableEventsF...
EventsManager.java		2.455	2.641	16	/refactoring-memoranda/src/main/java/memoranda/EventsManage...	getRepeatableEventsF...
EventsManager		3.222	3.275	16	/refactoring-memoranda/src/main/java/memoranda/EventsManage...	getRepeatableEventsF...
getRepeatableEventsForDate	16					
getEvent	5					
createDay	4					
removeSticker	3					
isNREventsForDate	3					
getEventsForDate	3					
createRepeatableEvent	3					

I noticed that the class Complexity score averages all of its methods' Complexity score. So by removing some iteration logic and adding it to a separate method, this helps reduce the score in different ways. For one, it will reduce the original method's Complexity because there's no more iteration logic. Then, it also reduces the overall class Complexity score because it creates a separate smaller method which has a low score and brings down the overall class Complexity score.

Package-level Coupling

1) Afferent coupling means that classes in other packages have dependence upon classes within the local package. The local package is responsible, similar to a server.

   Efferent coupling means that the local package has dependence upon other external packages. Similar to a client.

2) The package with the worst Afferent Coupling measure is main.java.memoranda.util with 57.

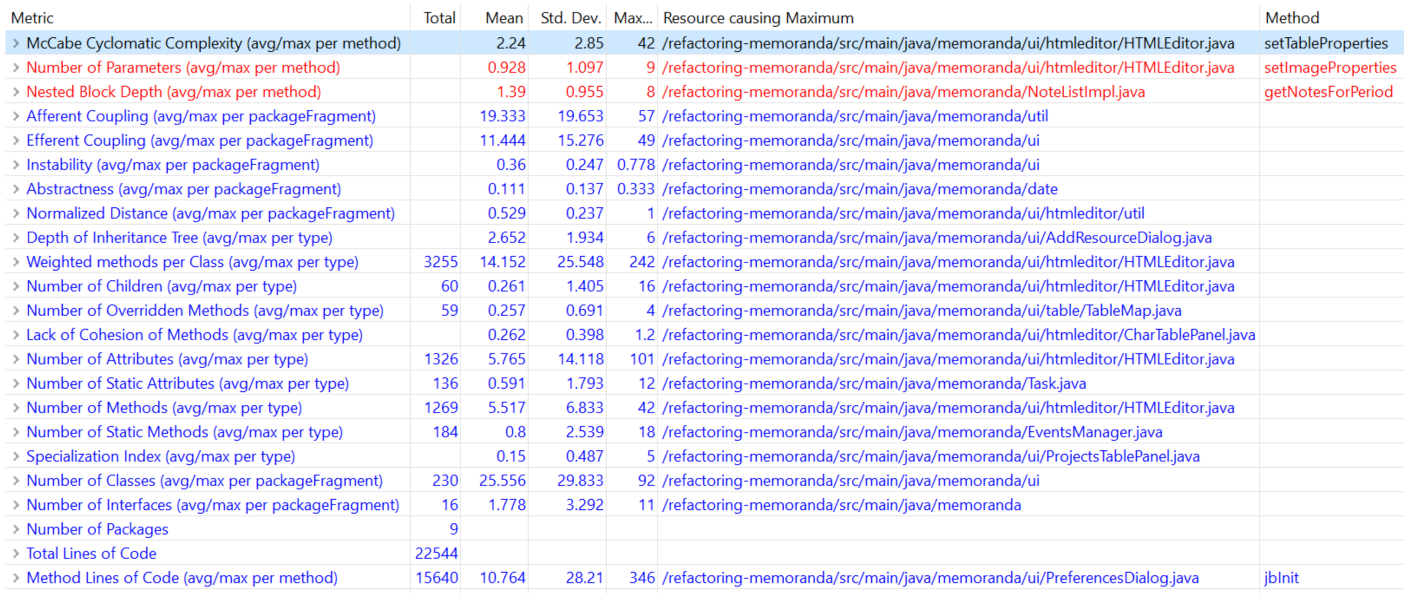
3) The package with the worst Efferent Coupling measure is main.java.memoranda.ui with 49.

Worst quality

In my opinion, the class with the worst quality based on metrics analysis is the HTMLEditor.java. I noticed that this class has a maximum of 242 methods. It has the most number of children at 16, the most number of attributes at 101, the most number of methods at 42. This class has a very large number of LOC at 2144. Based on these metrics, it might be beneficial to explore if the class can be broken up into multiple classes to make the overall solution less complex.

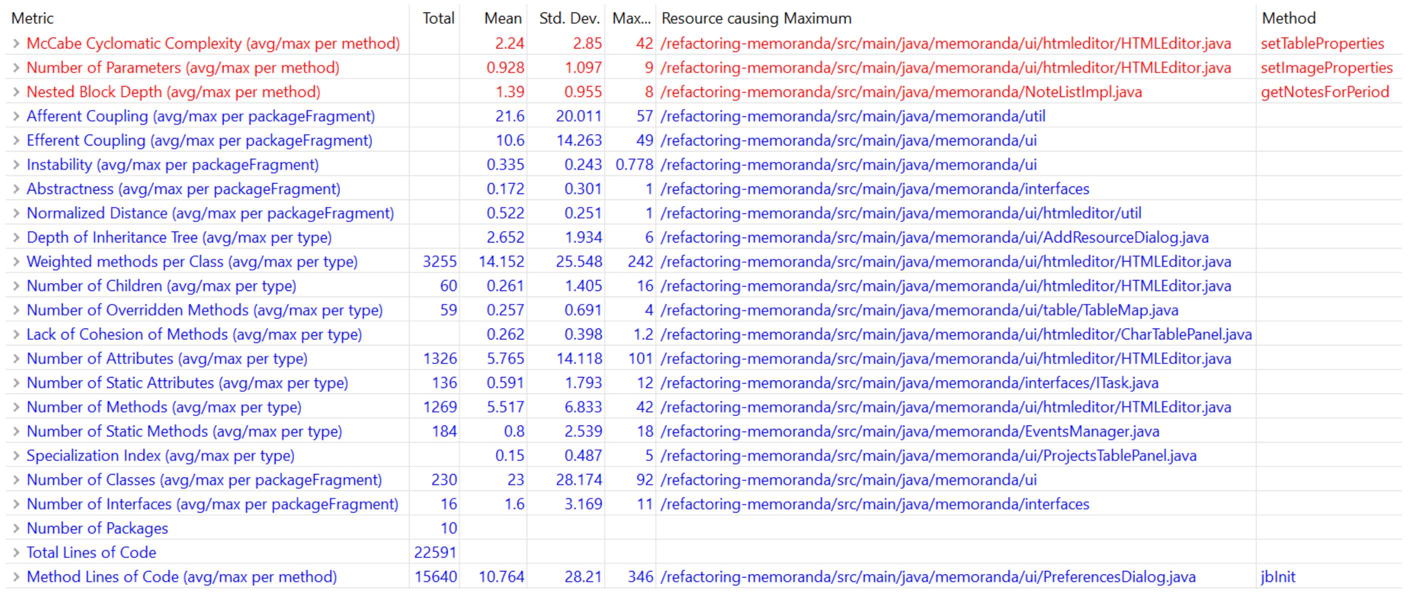
Task 2: Eclipse Refactoring

1) Metrics plugin results screenshot from rebuilding the project based on changes from Task 1.



Metric	Total	Mean	Std. Dev.	Max...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.85	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setImageProperties
Nested Block Depth (avg/max per method)		1.39	0.955	8	/refactoring-memoranda/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		19.333	19.653	57	/refactoring-memoranda/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		11.444	15.276	49	/refactoring-memoranda/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.36	0.247	0.778	/refactoring-memoranda/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.111	0.137	0.333	/refactoring-memoranda/src/main/java/memoranda/date	
Normalized Distance (avg/max per packageFragment)		0.529	0.237	1	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/util	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/refactoring-memoranda/src/main/java/memoranda/ui/AddResourceDialog.java	
Weighted methods per Class (avg/max per type)	3255	14.152	25.548	242	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Children (avg/max per type)	60	0.261	1.405	16	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/refactoring-memoranda/src/main/java/memoranda/ui/table/TableMap.java	
Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/CharTablePanel.java	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/refactoring-memoranda/src/main/java/memoranda/Task.java	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Methods (avg/max per type)	184	0.8	2.539	18	/refactoring-memoranda/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.15	0.487	5	/refactoring-memoranda/src/main/java/memoranda/ui/ProjectsTablePanel.java	
Number of Classes (avg/max per packageFragment)	230	25.556	29.833	92	/refactoring-memoranda/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.778	3.292	11	/refactoring-memoranda/src/main/java/memoranda	
Number of Packages	9					
Total Lines of Code	22544					
Method Lines of Code (avg/max per method)	15640	10.764	28.21	346	/refactoring-memoranda/src/main/java/memoranda/ui/PreferencesDialog.java	jbInit

7) Metrics summary since all refactoring has been completed.



Metric	Total	Mean	Std. Dev.	Max...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.85	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setImageProperties
Nested Block Depth (avg/max per method)		1.39	0.955	8	/refactoring-memoranda/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/refactoring-memoranda/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		10.6	14.263	49	/refactoring-memoranda/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.335	0.243	0.778	/refactoring-memoranda/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/refactoring-memoranda/src/main/java/memoranda/interfaces	
Normalized Distance (avg/max per packageFragment)		0.522	0.251	1	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/util	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/refactoring-memoranda/src/main/java/memoranda/ui/AddResourceDialog.java	
Weighted methods per Class (avg/max per type)	3255	14.152	25.548	242	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Children (avg/max per type)	60	0.261	1.405	16	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/refactoring-memoranda/src/main/java/memoranda/ui/table/TableMap.java	
Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/CharTablePanel.java	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/refactoring-memoranda/src/main/java/memoranda/interfaces/ITask.java	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Methods (avg/max per type)	184	0.8	2.539	18	/refactoring-memoranda/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.15	0.487	5	/refactoring-memoranda/src/main/java/memoranda/ui/ProjectsTablePanel.java	
Number of Classes (avg/max per packageFragment)	230	23	28.174	92	/refactoring-memoranda/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/refactoring-memoranda/src/main/java/memoranda/interfaces	
Number of Packages	10					
Total Lines of Code	22591					
Method Lines of Code (avg/max per method)	15640	10.764	28.21	346	/refactoring-memoranda/src/main/java/memoranda/ui/PreferencesDialog.java	jbInit

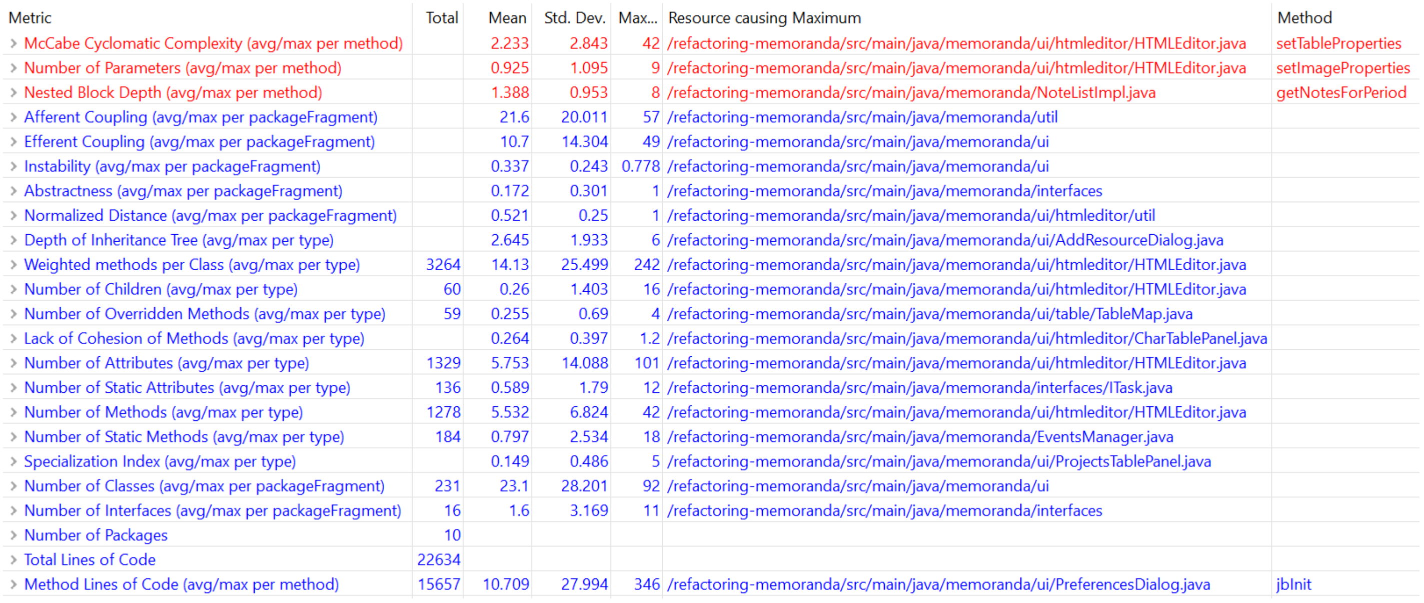
8) Some of the metrics have changed since refactoring. The Efferent Coupling metric was reduced from 11.444 to 10.6, so that one changed for the better. The Total LOC has increased since more imports were added because the interface files were moved to another package. Arguably the LOC increase has been a chance for the worst if someone strictly is focused on the LOC metric, but could be a change for the better for organizational and best practice purposes.

Task 3: Code Smells and Refactor

1) I found one Code Smell within the class main.java.memoranda.ui.EditorPanel.java. The jbInit() method was very large, so the Code Smell was identified as "method too large". I split the method up by taking all of the JButton initialization statements from jbInit() and moved them to a new private method called jbInitButtons(). That way, jbInit() is no longer too large, and easier to understand. jbInitButtons() could even be broken up further if needed.

2) The inter-class Code smell found was when main.java.memoranda.EventsManager.java has a public method, getEvent, which returns an IEvent object. However, it takes in too many primitive parameters. So I created a new class in that package, DateWithTime.java, which is composed of those parameters.

3) The metric data "after". Also uploaded to the GitHub repo as "ser316_nokerber_metrics_Task3_Question3.xml".



Metric	Total	Mean	Std. Dev.	Max...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.233	2.843	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setTableProperties
Number of Parameters (avg/max per method)		0.925	1.095	9	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setImageProperties
Nested Block Depth (avg/max per method)		1.388	0.953	8	/refactoring-memoranda/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/refactoring-memoranda/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		10.7	14.304	49	/refactoring-memoranda/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.337	0.243	0.778	/refactoring-memoranda/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/refactoring-memoranda/src/main/java/memoranda/interfaces	
Normalized Distance (avg/max per packageFragment)		0.521	0.25	1	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/util	
Depth of Inheritance Tree (avg/max per type)		2.645	1.933	6	/refactoring-memoranda/src/main/java/memoranda/ui/AddResourceDialog.java	
Weighted methods per Class (avg/max per type)	3264	14.13	25.499	242	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Children (avg/max per type)	60	0.26	1.403	16	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Overridden Methods (avg/max per type)	59	0.255	0.69	4	/refactoring-memoranda/src/main/java/memoranda/ui/table/TableMap.java	
Lack of Cohesion of Methods (avg/max per type)		0.264	0.397	1.2	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/CharTablePanel.java	
Number of Attributes (avg/max per type)	1329	5.753	14.088	101	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Attributes (avg/max per type)	136	0.589	1.79	12	/refactoring-memoranda/src/main/java/memoranda/interfaces/ITask.java	
Number of Methods (avg/max per type)	1278	5.532	6.824	42	/refactoring-memoranda/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Methods (avg/max per type)	184	0.797	2.534	18	/refactoring-memoranda/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.149	0.486	5	/refactoring-memoranda/src/main/java/memoranda/ui/ProjectsTablePanel.java	
Number of Classes (avg/max per packageFragment)	231	23.1	28.201	92	/refactoring-memoranda/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/refactoring-memoranda/src/main/java/memoranda/interfaces	
Number of Packages	10					
Total Lines of Code	22634					
Method Lines of Code (avg/max per method)	15657	10.709	27.994	346	/refactoring-memoranda/src/main/java/memoranda/ui/PreferencesDialog.java	jbInit

4) Comparison of the results of metrics now vs. at the end of task 2. The metric that stuck out to me the most was the LOC value. It is higher now. But, I believe this changed for the better because of the Refactoring that was done on Task 3 Question 2 where I had to add a new class. Also, because of the new class, the mean Cyclomatic Complexity was reduced.