Chandler Boone
SER316
Assignment 7 – Refactoring/Metrics

Task 1

Size

1. There are a total of 22539 lines of code in the project.

2. The largest single file in the project is HTMLEditor.java in the ui.htmleditor package with 2144 lines of code.

3. The Metrics tool does not seem to name a specific method for determining total lines of code, but it appears to use a method that counts physical lines of code and excludes comments.

Cohesion

1. LCOM2 is determined by subtracting from one the quotient of all attribute-accessing methods over the product of all methods and attributes.

2. There are multiple classes with an LCOM rating of 0, suggesting perfect cohesion. In the main package, EventNotificationListener.java has a 0 rating, but this is because it is an interface that has no attributes. EventsScheduler.java also has a 0 rating, but this is because all of its methods and attributes are static, which are ignored in by the Metrics tool for LCOM rating. Lastly, ProjectImpl.java has a rating of 0, because it has one attribute accessed by all methods, giving a perfect cohesion rating.

Complexity

1. The main package has an average complexity of 1.746.

2. In the main package, the class Start.java has the worst complexity of 3.5.

3. In the main package, I reduced the complexity of CurrentProject.java by 0.083 by removing the if statement in the set() method, reducing the number of conditional numbers and therein the complexity.

Package-Level Coupling

1. Afferent and Efferent coupling refer to the direction of dependencies: Afferent coupling means that there are external classes that rely on the current package, and Efferent coupling means that the current package relies on external classes.

2. The util package has the worst Afferent coupling measure with a value of 57.

3. The ui package has the worst Efferent coupling measure with a value of 49.

Chandler Boone
SER316
Assignment 7 – Refactoring/Metrics

Quality

The worst class in the main package would likely be EventsManager.java. When including static methods and variables in the LCOM measure, it results in the worst rating with a value of 0.918. Additionally, it is the largest class in the package and the second most complex one, with a complexity measure of 2.5. All of this suggests that the class has a lot of separate functionalities pooled into one class and could possibly be separated to improve quality.

Task 2

Before

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.851	42	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setImageProperties
Nested Block Depth (avg/max per method)		1.39	0.955	8	/Assignment7/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		19.333	19.653	57	/Assignment7/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		11.444	15.276	49	/Assignment7/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.36	0.247	0.778	/Assignment7/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.111	0.137	0.333	/Assignment7/src/main/java/memoranda/date	
Normalized Distance (avg/max per packageFragment)		0.529	0.237	1	/Assignment7/src/main/java/memoranda/ui/htmleditor/util	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/Assignment7/src/main/java/memoranda/ui/FileExportDialog.java	
Weighted methods per Class (avg/max per type)	3253	14.143	25.54	242	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Children (avg/max per type)	60	0.261	1.405	16	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/Assignment7/src/main/java/memoranda/ui/table/TableMap.java	
Lack of Cohesion of Methods (avg/max per type)		0.326	0.416	1.333	/Assignment7/src/main/java/memoranda/ui/EventsTable.java	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/Assignment7/src/main/java/memoranda/Task.java	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Methods (avg/max per type)	183	0.796	2.51	17	/Assignment7/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.15	0.487	5	/Assignment7/src/main/java/memoranda/ui/ProjectsTablePanel.java	
Number of Classes (avg/max per packageFragment)	230	25.556	29.833	92	/Assignment7/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.778	3.292	11	/Assignment7/src/main/java/memoranda	
Number of Packages	9					
Total Lines of Code	22538					
Method Lines of Code (avg/max per method)	15636	10.769	28.219	346	/Assignment7/src/main/java/memoranda/ui/PreferencesDialog.java	jbInit

After

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.851	42	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setImageProperties
Nested Block Depth (avg/max per method)		1.39	0.955	8	/Assignment7/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/Assignment7/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		10.6	14.263	49	/Assignment7/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.335	0.243	0.778	/Assignment7/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/Assignment7/src/main/java/memoranda/interfaces	
Normalized Distance (avg/max per packageFragment)		0.522	0.251	1	/Assignment7/src/main/java/memoranda/ui/htmleditor/util	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/Assignment7/src/main/java/memoranda/ui/FileExportDialog.java	
Weighted methods per Class (avg/max per type)	3253	14.143	25.54	242	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Children (avg/max per type)	60	0.261	1.405	16	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/Assignment7/src/main/java/memoranda/ui/table/TableMap.java	
Lack of Cohesion of Methods (avg/max per type)		0.326	0.416	1.333	/Assignment7/src/main/java/memoranda/ui/EventsTable.java	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/Assignment7/src/main/java/memoranda/interfaces/ITask.java	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Methods (avg/max per type)	183	0.796	2.51	17	/Assignment7/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.15	0.487	5	/Assignment7/src/main/java/memoranda/ui/ProjectsTablePanel.java	
Number of Classes (avg/max per packageFragment)	230	23	28.174	92	/Assignment7/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/Assignment7/src/main/java/memoranda/interfaces	
Number of Packages	10					
Total Lines of Code	22585					
Method Lines of Code (avg/max per method)	15636	10.769	28.219	346	/Assignment7/src/main/java/memoranda/ui/PreferencesDialog.java	jbInit

Generally, refactoring did not make much of a difference other than improving the average Efferent coupling and increase Afferent coupling. The Afferent coupling increased because the new interfaces package has a significant number of classes that rely on it, and the Efferent coupling decreased because some of the couplings from the main package were removed with the interfaces, lowering the average but not entirely removing the couplings.

Chandler Boone
SER316
Assignment 7 – Refactoring/Metrics

Task 3

1. For the code smell within a class, I edited the method createRepeatableEvent() in EventManager.java in the main package to have less duplicate code, as portions of it were similar to the createEvent(). As such, the former method now uses the latter to initialize portions of the Element object for the Event, eliminating the repeated code.

2. For the code smell between classes, I split up the large class file of EventManager.java in the main package into separate class files, as it contained the code for the classes of Day.java, Month.java, and Year.java. Admittedly this is more of a "large file" code smell than a "large class" smell as the other classes were already separate and not a part of the EventManager class. However, it does provide better organization and a better visualization of the functionality of the formerly-embedded classes.

3. AFTER:

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.851	42	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	setImageProperties
Nested Block Depth (avg/max per method)		1.39	0.955	8	/Assignment7/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		21.7	20.13	57	/Assignment7/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		10.9	14.405	49	/Assignment7/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.339	0.243	0.778	/Assignment7/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/Assignment7/src/main/java/memoranda/interfaces	
Normalized Distance (avg/max per packageFragment)		0.518	0.249	1	/Assignment7/src/main/java/memoranda/ui/htmleditor/util	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/Assignment7/src/main/java/memoranda/ui/FileExportDialog.java	
Weighted methods per Class (avg/max per type)	3253	14.143	25.54	242	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Children (avg/max per type)	60	0.261	1.405	16	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/Assignment7/src/main/java/memoranda/ui/table/TableMap.java	
Lack of Cohesion of Methods (avg/max per type)		0.326	0.416	1.333	/Assignment7/src/main/java/memoranda/ui/EventsTable.java	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/Assignment7/src/main/java/memoranda/interfaces/ITask.java	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/Assignment7/src/main/java/memoranda/ui/htmleditor/HTMLEditor.java	
Number of Static Methods (avg/max per type)	183	0.796	2.51	17	/Assignment7/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.15	0.487	5	/Assignment7/src/main/java/memoranda/ui/ProjectsTablePanel.java	
Number of Classes (avg/max per packageFragment)	230	23	28.174	92	/Assignment7/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/Assignment7/src/main/java/memoranda/interfaces	
Number of Packages	10					
Total Lines of Code	22596					
Method Lines of Code (avg/max per method)	15634	10.767	28.219	346	/Assignment7/src/main/java/memoranda/ui/PreferencesDialog.java	jbInit

4. The only notable metrics changes are the minor increase in both Afferent and Efferent coupling (there might have been some more noticeable differences where this a cap of the metrics for just the main package, but alas), suggesting that refactoring made changes for the worse. The increase in coupling likely stems from the creation of three new classes, providing a minor numeric bump in the calculations, though it's questionable if this has any significant impact on performance seeing as there is no major changes in the code itself.