Daniel Torzala
SER316 Assignment 7
4/29/2019
asurite = dtorzala
github username = dtorzala90
https://github.com/dtorzala90/memoranda-dtorzala.git

Assignment7 PDF

Task 1

-E- Size

1. 1. What is the Total Lines of Code (LOC) in the project?
   22539
2. What is the largest single code file in the project and its Total LOC?
   HTMLEdistor.java with 2144 LOC
3. Inspect CurrentNote.java - what method did the Metrics tool use to determine Total LOC? Describe the method.
   The total LOC of CurrentNote.java was calculated by counting the non-blank and non-comment lines of the compilation code.

-E- Cohesion

1. What is the definition of LCOM2 and how is it calculated?
   Definition from SourceForge - Eclipse Metrics plugin 1.3.8 - Getting started (http://metrics2.sourceforge.net/)
   If (m(A) is the number of methods accessing an attribute A, calculate the average of m(A) for all attributes, subtract the number of methods m and divide the result by (1-m). A low value indicates a cohesive class and a value close to 1 indicates a lack of cohesion and suggests the class might better be split into a number of (sub)classes.
   In other words, it calculates whether the class has become so large that there are far too many methods compared to attributes and therefore it would be better to break up the class to delegate behavioral methods.
2. Which class has the highest Cohesion and do you have an idea why?
   HTMLEditor.java has the highest cohesion with a mean of 0.031. I believe this is because many of the methods are Action or ActionListeners which refer to and use the attributes of the class.

-E- Complexity

1. What is the cyclomatic complexity in the main package?
   The main package McCabe Cyclomatic Complexity has a mean of 2.241
2. What class has, on average, the worst McCabe Cyclomatic Complexity (CC) and what is it?
   The class HTMLEditor.java has the worst complexity with a mean 5.762
3. Go back to your code and reduce the Cyclomatic Complexity. You can choose any class but the Cyclomatic Complexity needs to be reduced at least by a small amount somewhere. Explain what you changed and why, and why it reduced the complexity and how much you were able to reduce the complexity.
   ImagePreview.java complexity mean 3 was reduced to complexity mean 2.75 by simplifying an if statement to an else in the paintComponent method.

-E- Package-level Coupling

1. What do Afferent and Efferent coupling mean? Look these terms up on Wikipedia and summarize the distinction.
   Efferent Coupling is measured by the number of classes in other packages which the package in question depends on/knows about.
   Afferent Coupling is measured by the number of classes that depend on/know about the package in question.
2. What package has the worse Afferent Coupling measure and what is the value?
   The main.java.memoranda.util has the worst afferent coupling with a total of 57
3. What package has the worse Efferent Coupling measure and what is the value?
   The main.java.memoranda.ui has the worst efferent coupling with a total of 49

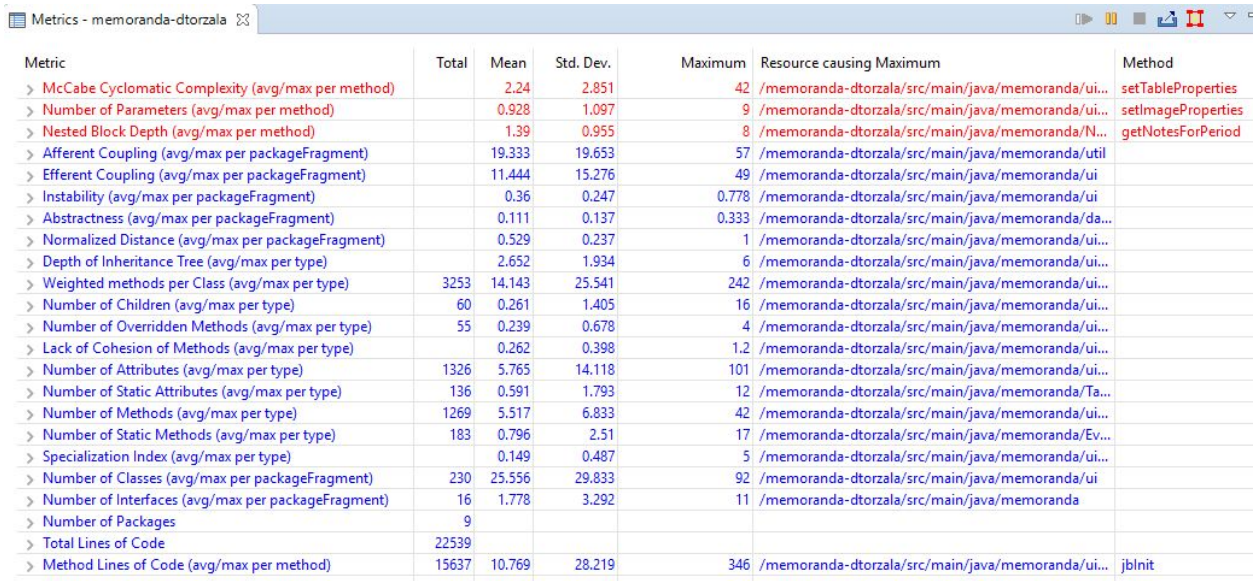
Worst Quality

Which class has the worst quality and why?
The class with the worst quality is the HTMLEditor. The reason for this lies in it having the highest complexity. This level of complexity makes troubleshooting this classes methods near impossible and changing anything in it extremely troublesome. The coupling of this class, both afferent and efferent, is mid-range, which is also not ideal.

Task 2

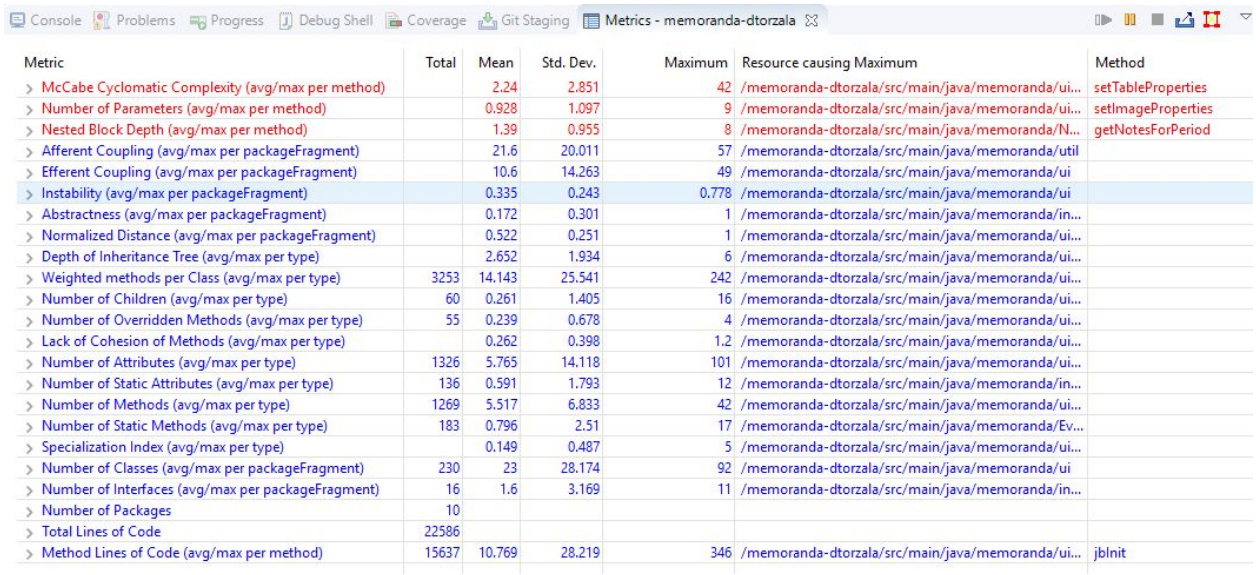
1. Run your Metrics tool again (Checkstyle or in Eclipse) before doing any of the steps in Task 2. Take a screenshot (-E-) and paste it in a document file and put it at the beginning of Task 2 in your document or save the HTML file (-G-) to later submit it.

BEFORE:


Metrics - memoranda-dtorzala

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.851	42	/memoranda-dtorzala/src/main/java/memoranda/ui...	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/memoranda-dtorzala/src/main/java/memoranda/ui...	setImageProperties
Nested Block Depth (avg/max per method)		1.39	0.955	8	/memoranda-dtorzala/src/main/java/memoranda/N...	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		19.333	19.653	57	/memoranda-dtorzala/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		11.444	15.276	49	/memoranda-dtorzala/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.36	0.247	0.778	/memoranda-dtorzala/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.111	0.137	0.333	/memoranda-dtorzala/src/main/java/memoranda/da...	
Normalized Distance (avg/max per packageFragment)		0.529	0.237	1	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Weighted methods per Class (avg/max per type)	3253	14.143	25.541	242	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Children (avg/max per type)	60	0.261	1.405	16	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Overridden Methods (avg/max per type)	55	0.239	0.678	4	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/memoranda-dtorzala/src/main/java/memoranda/Ta...	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Static Methods (avg/max per type)	183	0.796	2.51	17	/memoranda-dtorzala/src/main/java/memoranda/Ev...	
Specialization Index (avg/max per type)		0.149	0.487	5	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Classes (avg/max per packageFragment)	230	25.556	29.833	92	/memoranda-dtorzala/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.778	3.292	11	/memoranda-dtorzala/src/main/java/memoranda	
Number of Packages	9					
Total Lines of Code	22539					
Method Lines of Code (avg/max per method)	15637	10.769	28.219	346	/memoranda-dtorzala/src/main/java/memoranda/ui...	jbInit

AFTER


Console Problems Progress Debug Shell Coverage Git Staging Metrics - memoranda-dtorzala

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.851	42	/memoranda-dtorzala/src/main/java/memoranda/ui...	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/memoranda-dtorzala/src/main/java/memoranda/ui...	setImageProperties
Nested Block Depth (avg/max per method)		1.39	0.955	8	/memoranda-dtorzala/src/main/java/memoranda/N...	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		21.6	20.011	57	/memoranda-dtorzala/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		10.6	14.263	49	/memoranda-dtorzala/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.335	0.243	0.778	/memoranda-dtorzala/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/memoranda-dtorzala/src/main/java/memoranda/in...	
Normalized Distance (avg/max per packageFragment)		0.522	0.251	1	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Depth of Inheritance Tree (avg/max per type)		2.652	1.934	6	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Weighted methods per Class (avg/max per type)	3253	14.143	25.541	242	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Children (avg/max per type)	60	0.261	1.405	16	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Overridden Methods (avg/max per type)	55	0.239	0.678	4	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Lack of Cohesion of Methods (avg/max per type)		0.262	0.398	1.2	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/memoranda-dtorzala/src/main/java/memoranda/in...	
Number of Methods (avg/max per type)	1269	5.517	6.833	42	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Static Methods (avg/max per type)	183	0.796	2.51	17	/memoranda-dtorzala/src/main/java/memoranda/Ev...	
Specialization Index (avg/max per type)		0.149	0.487	5	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Classes (avg/max per packageFragment)	230	23	28.174	92	/memoranda-dtorzala/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/memoranda-dtorzala/src/main/java/memoranda/in...	
Number of Packages	10					
Total Lines of Code	22586					
Method Lines of Code (avg/max per method)	15637	10.769	28.219	346	/memoranda-dtorzala/src/main/java/memoranda/ui...	jbInit

8. Compare the results of your 2 metrics exports. Did any of the metrics change for the better after your refactoring. Pick and state a metric (or metrics) whose value changed, and indicated why it changed and whether it changed for the better (or worse) because of the refactoring.
The coupling metrics changed because now the dependencies between packages are different. The Afferent Coupling went from 19.333 to 21.6 meaning now on average there are more classes that have to know about each package. This is not a good change. The Efferent coupling went from 11.444 to 10.6 meaning now there are fewer classes that each class has to know about. This is a good change.

Task 3

1. Summarize your refactoring changes in your document and describe the smell (see below). Please state the class-name /class-path of the class being altered. State the main part of your change in the document justifying why you made the change
I changed the accept(File f) method in AllFilesFilter.java of the main.java.memoranda.ui package. I shortened the method a bit. It had a really long if-else statement that I saw could be reasonably shortened by using an ignoreCase method.

2. Find one code smell between classes. Identify the smell by making comments in ALL RELEVANT CLASSES prefixed by 'TASK 3-2 SMELL BETWEEN CLASSES < describe the smell(see below)>'. Then proceed to refactor the code to remove the smell. Summarize your refactoring changes in your code comments (see below).
I moved the class HistoryItem to be an inner class of History.java. This helps keep the class contained by the class that almost exclusively uses the object.

3. Re-run the Metrics plug-in and again capture an export. Paste a screen-shot of the new metric as 'after' in the document file.

AFTER

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.24	2.851	42	/memoranda-dtorzala/src/main/java/memoranda/ui...	setTableProperties
Number of Parameters (avg/max per method)		0.928	1.097	9	/memoranda-dtorzala/src/main/java/memoranda/ui...	setImageProperties
Nested Block Depth (avg/max per method)		1.391	0.956	8	/memoranda-dtorzala/src/main/java/memoranda/N...	getNotesForPeriod
Afferent Coupling (avg/max per packageFragment)		21.5	19.896	57	/memoranda-dtorzala/src/main/java/memoranda/util	
Efferent Coupling (avg/max per packageFragment)		10.5	14.228	49	/memoranda-dtorzala/src/main/java/memoranda/ui	
Instability (avg/max per packageFragment)		0.334	0.242	0.778	/memoranda-dtorzala/src/main/java/memoranda/ui	
Abstractness (avg/max per packageFragment)		0.172	0.301	1	/memoranda-dtorzala/src/main/java/memoranda/in...	
Normalized Distance (avg/max per packageFragment)		0.524	0.251	1	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Depth of Inheritance Tree (avg/max per type)		2.649	1.93	6	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Weighted methods per Class (avg/max per type)	3253	14.082	25.502	242	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Children (avg/max per type)	60	0.26	1.403	16	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Overridden Methods (avg/max per type)	55	0.238	0.677	4	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Lack of Cohesion of Methods (avg/max per type)		0.261	0.397	1.2	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Attributes (avg/max per type)	1327	5.745	14.091	101	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Static Attributes (avg/max per type)	136	0.589	1.79	12	/memoranda-dtorzala/src/main/java/memoranda/in...	
Number of Methods (avg/max per type)	1269	5.494	6.827	42	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Static Methods (avg/max per type)	183	0.792	2.505	17	/memoranda-dtorzala/src/main/java/memoranda/Ev...	
Specialization Index (avg/max per type)		0.149	0.486	5	/memoranda-dtorzala/src/main/java/memoranda/ui...	
Number of Classes (avg/max per packageFragment)	231	23.1	28.42	93	/memoranda-dtorzala/src/main/java/memoranda/ui	
Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/memoranda-dtorzala/src/main/java/memoranda/in...	
Number of Packages	10					
Total Lines of Code	22592					
Method Lines of Code (avg/max per method)	15640	10.771	28.227	346	/memoranda-dtorzala/src/main/java/memoranda/ui...	jbInit

4. Compare the results of your metrics at the end of task 2 and what you have now. Did any of the metrics change for the better after your 2 refactoring. Pick and state a metric (or metrics) whose value changed, and indicated why it changed and whether it changed for the better (or worse) because of the refactoring. Put the document in the root of your source tree for submission.
Both the Afferent and Efferent Coupling metrics went down from 21.6 and 10.6 to 21.5 and 10.5 respectively. This is due to the class History not relying on the class HistoryItem. Now it has the class as an inner class. Also, DailyItemsPanel now only has to import the History class and it comes with the HistoryItem inner class.

Include your GitHub link
https://github.com/dtorzala90/memoranda-dtorzala.git