Size

Total LOC: 22539

Larget file: HTMLEditor.java at 2144 LOC

The metrics tool used method 1. Every line counts towards the total except for empty lines

Cohesion

LCOM2 definition: The percentage of methods that do not access a specific attribute averaged over all attributes in the class.

M = number of procedures (methods) in class

A = number of variables (attributes) in class

mA = number of methods that access a variable (attribute)

*Equation: LCOM2 = 1 - sum(mA)/(m*a)*

Every class where the LCOM = 0 was tied for the highest cohesion. We will use Local.java for example. This class has two methods: setMessages and getString. Both of these deal with the message attribute and thus work together to add to the functionality of the class. It would be impractical to separate these methods into two classes, thus making it by definition cohesive.

Complexity

1.) The mean cyclomatic complexity of the package is 2.241

2.) ProjectPackager has the worst average complexity at 5.667

3.) In Local.java, I removed an if statement in getString that returned the key if messages == null or was disabled. This if statement was entirely unnecessary as it returns the key in the else statement regardless. This removal changed the complexity ftom 2.241 to 1.867.

Package-level Coupling

1.) Afferent coupling is the number of classes outside the package that depend on classes inside the package. Efferent coupling is the number of classes outside the package that the classes inside the package depend on. In short. Afferent = incoming dependencies, efferent = outgoing dependencies.

2.) memoranda.util has the worst afferent coupling at 57.

3.) memoranda.ui has the worst efferent coupling at 49.

A case could be made for the util package because of its 57 afferent coupling, but I expect this number to be high since, as the name implies, its providing utiliy to other classes. A case could also be made for the ui package with its 49 efferent coupling and decently high cyclomatic complexity but I also expect

the ui to be relying on other packages for its functionality. In my opinion it's the htmleditor package. This is because its complexity, block depth and number of parameters are all above 1, with the complexity being exceptionally high at 3.17. The afferent and efferent coupling are at 6 and 12 respectively despite this being a package that should relatively be isolated.

Task 2

Before:

Metric	Total	Mean	Std. De...	Maxi...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity		2.24	2.85	42	/memoranda_jvredevo/src/main/java/me...	setTableProperties
Number of Parameters (avg/max		0.928	1.097	9	/memoranda_jvredevo/src/main/java/me...	setImageProperties
Nested Block Depth (avg/max pe		1.39	0.955	8	/memoranda_jvredevo/src/main/java/me...	getNotesForPeriod
Afferent Coupling (avg/max per		19.333	19.653	57	/memoranda_jvredevo/src/main/java/me...	
Efferent Coupling (avg/max per		11.444	15.276	49	/memoranda_jvredevo/src/main/java/me...	
Instability (avg/max per package		0.36	0.247	0.778	/memoranda_jvredevo/src/main/java/me...	
Abstractness (avg/max per packa		0.111	0.137	0.333	/memoranda_jvredevo/src/main/java/me...	
Normalized Distance (avg/max p		0.529	0.237	1	/memoranda_jvredevo/src/main/java/me...	
Depth of Inheritance Tree (avg/m		2.652	1.934	6	/memoranda_jvredevo/src/main/java/me...	
Weighted methods per Class (av	3252	14.139	25.535	242	/memoranda_jvredevo/src/main/java/me...	
Number of Children (avg/max pe	60	0.261	1.405	16	/memoranda_jvredevo/src/main/java/me...	
Number of Overridden Methods	59	0.257	0.691	4	/memoranda_jvredevo/src/main/java/me...	
Lack of Cohesion of Methods (av		0.262	0.398	1.2	/memoranda_jvredevo/src/main/java/me...	
Number of Attributes (avg/max p	1326	5.765	14.118	101	/memoranda_jvredevo/src/main/java/me...	
Number of Static Attributes (avg	136	0.591	1.793	12	/memoranda_jvredevo/src/main/java/me...	
Number of Methods (avg/max p	1269	5.517	6.833	42	/memoranda_jvredevo/src/main/java/me...	
Number of Static Methods (avg/	183	0.796	2.51	17	/memoranda_jvredevo/src/main/java/me...	
Specialization Index (avg/max pe		0.15	0.487	5	/memoranda_jvredevo/src/main/java/me...	
Number of Classes (avg/max per	230	25.556	29.833	92	/memoranda_jvredevo/src/main/java/me...	
Number of Interfaces (avg/max p	16	1.778	3.292	11	/memoranda_jvredevo/src/main/java/me...	
Number of Packages	9					
Total Lines of Code	22536					
Method Lines of Code (avg/max	15634	10.767	28.22	346	/memoranda_jvredevo/src/main/java/me...	jblnit

After:

Metric	Total	Mean	Std. De...	Maxi...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity		2.24	2.85	42	/memoranda_jvredevo/src/main/java/me...	setTableProperties
Number of Parameters (avg/max		0.928	1.097	9	/memoranda_jvredevo/src/main/java/me...	setImageProperties
Nested Block Depth (avg/max pe		1.39	0.955	8	/memoranda_jvredevo/src/main/java/me...	getNotesForPeriod
Afferent Coupling (avg/max per		21.6	20.011	57	/memoranda_jvredevo/src/main/java/me...	
Efferent Coupling (avg/max per		10.6	14.263	49	/memoranda_jvredevo/src/main/java/me...	
Instability (avg/max per package		0.335	0.243	0.778	/memoranda_jvredevo/src/main/java/me...	
Abstractness (avg/max per packa		0.172	0.301	1	/memoranda_jvredevo/src/main/java/me...	
Normalized Distance (avg/max p		0.522	0.251	1	/memoranda_jvredevo/src/main/java/me...	
Depth of Inheritance Tree (avg/m		2.652	1.934	6	/memoranda_jvredevo/src/main/java/me...	
Weighted methods per Class (av	3252	14.139	25.535	242	/memoranda_jvredevo/src/main/java/me...	
Number of Children (avg/max pe	60	0.261	1.405	16	/memoranda_jvredevo/src/main/java/me...	
Number of Overridden Methods	59	0.257	0.691	4	/memoranda_jvredevo/src/main/java/me...	
Lack of Cohesion of Methods (av		0.262	0.398	1.2	/memoranda_jvredevo/src/main/java/me...	
Number of Attributes (avg/max p	1326	5.765	14.118	101	/memoranda_jvredevo/src/main/java/me...	
Number of Static Attributes (avg	136	0.591	1.793	12	/memoranda_jvredevo/src/main/java/me...	
Number of Methods (avg/max p	1269	5.517	6.833	42	/memoranda_jvredevo/src/main/java/me...	
Number of Static Methods (avg/	183	0.796	2.51	17	/memoranda_jvredevo/src/main/java/me...	
Specialization Index (avg/max pe		0.15	0.487	5	/memoranda_jvredevo/src/main/java/me...	
Number of Classes (avg/max per	230	23	28.174	92	/memoranda_jvredevo/src/main/java/me...	
Number of Interfaces (avg/max p	16	1.6	3.169	11	/memoranda_jvredevo/src/main/java/me...	
Number of Packages	10					
Total Lines of Code	22577					
Method Lines of Code (avg/max	15634	10.767	28.22	346	/memoranda_jvredevo/src/main/java/me...	jbInit

Afferent coupling raised from 19.33 to 21.6 and efferent coupling lowered from 11.44 to 10.6. the instability also lowered from 0.36 to 0.33. I believe this change is for the better. The may be more afferent coupling, but it is a result of better organization among the code, as well as more stability keeping all the interfaces in the same place.

Task 3

Smell within code

I found a code smell of type speculative generality in the class CharTablePanel. This class contained another class called CharAction. CharAction seemed like it would do just fine on its own and would thus provide more distinction between the two classes, so I created a new java file for CharAction and removed it from CHarTablePanel.

Smell between code

I found a code smell of type Feature Envy in the class Context. This class did literally nothing except make a LoadableProperties object and manipulate it using LoadableProperties methods. Because of this, I decided to delete context altogether and move its functionality under the LoadableProperties class.

Metric	Total	Mean	Std. De...	Maxi...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity		2.237	2.848	42	/memoranda_jvredevo/src/main/java/me...	setTableProperties
Number of Parameters (avg/max		0.929	1.097	9	/memoranda_jvredevo/src/main/java/me...	setImageProperties
Nested Block Depth (avg/max pe		1.39	0.955	8	/memoranda_jvredevo/src/main/java/me...	getNotesForPeriod
Afferent Coupling (avg/max per		21.6	20.011	57	/memoranda_jvredevo/src/main/java/me...	
Efferent Coupling (avg/max per		10.7	14.311	49	/memoranda_jvredevo/src/main/java/me...	
Instability (avg/max per package		0.336	0.242	0.778	/memoranda_jvredevo/src/main/java/me...	
Abstractness (avg/max per packa		0.172	0.301	1	/memoranda_jvredevo/src/main/java/me...	
Normalized Distance (avg/max p		0.522	0.25	1	/memoranda_jvredevo/src/main/java/me...	
Depth of Inheritance Tree (avg/m		2.638	1.932	6	/memoranda_jvredevo/src/main/java/me...	
Weighted methods per Class (av	3255	14.03	25.452	242	/memoranda_jvredevo/src/main/java/me...	
Number of Children (avg/max pe	60	0.259	1.4	16	/memoranda_jvredevo/src/main/java/me...	
Number of Overridden Methods	59	0.254	0.689	4	/memoranda_jvredevo/src/main/java/me...	
Lack of Cohesion of Methods (av		0.26	0.397	1.2	/memoranda_jvredevo/src/main/java/me...	
Number of Attributes (avg/max p	1327	5.72	14.065	101	/memoranda_jvredevo/src/main/java/me...	
Number of Static Attributes (avg	137	0.591	1.786	12	/memoranda_jvredevo/src/main/java/me...	
Number of Methods (avg/max p	1270	5.474	6.819	42	/memoranda_jvredevo/src/main/java/me...	
Number of Static Methods (avg/	185	0.797	2.501	17	/memoranda_jvredevo/src/main/java/me...	
Specialization Index (avg/max p		0.149	0.485	5	/memoranda_jvredevo/src/main/java/me...	
Number of Classes (avg/max per	232	23.2	28.174	92	/memoranda_jvredevo/src/main/java/me...	
Number of Interfaces (avg/max p	16	1.6	3.169	11	/memoranda_jvredevo/src/main/java/me...	
Number of Packages	10					
Total Lines of Code	22603					
Method Lines of Code (avg/max	15637	10.747	28.194	346	/memoranda_jvredevo/src/main/java/me...	jbInit

Two stats changed for the better: Complexity from 2.24 to 2.237 and Weighted methods per class from 14.139 to 14.03. These are small improvements but I made small changes so it matches. I am sure if I refactored the entire project I would see major improvement based on the changes I've seen here.