Rebecca Rodriguez

SER 316 Spring B

4/29/19

Assignment 7 (-E-)

Task 1:

Size

1.) The Total Lines of Code in the project is **2187**.
2.) The largest single code file in the project is **EventsManager.java** at **329** lines of code.
3.) The Metrics tool used **SLOC**, which is a method that counts the executable statements excluding the comment statements.

Cohesion

1.) **LCOM 2** is the percentage of methods that don't access a specific attribute averaged over all attributes within the class. This is calculated by the following formula: **LCOM2 = 1 – sum(mA)/(m*a).**
2.) Many of the classes that do not use any sort of decision making (if-else statements, etc.) have 0 cohesion, such as **Task.java** and **NoteList.java**. Of the classes that do contain some sort of branching or decision-making elements, **HistoryItem.java** has the highest Cohesion at 0.333; it has a Boolean method ***equals()*** that checks for a **HistoryItem**, and returns data with an AND operator. This is the simplest decision-making method and thus has the highest cohesion.

Complexity

1.) The cyclomatic complexity of the main package is **1.746**.
2.) The class with the worst McCabe Cyclomatic Complexity is **EventsManager.java** (**2.5**).
3.) By removing the try-catch statements checking for the program being open in **Start.java**, the CC was reduced; **Start.java** dropped from **3.5** to **2.5**, and the whole package dropped from **1.746** to **1.74**. This reduced the complexity because it removed both an if-statement and try-catch statements, which decreased the amount of branching that was happening in this class.

Package-Level Coupling

1.)
   a. **Afferent:** The number of classes in outside packages that depend on classes in the current package (out-in).
   b. **Efferent:** The number of classes in outside packages that classes in the current package depend upon (in-out).
2.) The package with the worst Afferent Coupling is **main.java.memoranda.util** at **57**.
3.) The package with the worst Efferent Coupling is **main.java.memoranda.ui** at **49**.

Worst Quality

1.) The class with the worst quality would be **TaskListImpl.java**. This class has the 3rd worst CC in the **main.java.memoranda** package (a difference of **0.227** from **EventsManager.java**, which has the worst CC). This class also has the highest Lack of Cohesion at **0.679**, and the 3rd highest Total Lines of Code at **286**.

Task 2:

1.) *Before*

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		1.74	1.538	16	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	getRepeatableEven...
Number of Parameters (avg/max per method)		0.675	1.004	8	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	createRepeatableEv...
Nested Block Depth (avg/max per method)		0.994	0.94	8	/memoranda_rrodri64/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling	34					
Efferent Coupling	21					
Instability	0.382					
Abstractness	0.275					
Normalized Distance	0.343					
Depth of Inheritance Tree (avg/max per type)		0.854	0.607	2	/memoranda_rrodri64/src/main/java/memoranda/EventsScheduler.java	
Weighted methods per Class (avg/max per type)	583	14.22	16.112	71	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Children (avg/max per type)	23	0.561	1.624	10	/memoranda_rrodri64/src/main/java/memoranda/ProjectListener.java	
Number of Overridden Methods (avg/max per type)	3	0.073	0.341	2	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Lack of Cohesion of Methods (avg/max per type)		0.093	0.211	0.679	/memoranda_rrodri64/src/main/java/memoranda/TaskListImpl.java	
Number of Attributes (avg/max per type)	30	0.732	1.037	4	/memoranda_rrodri64/src/main/java/memoranda/TaskListImpl.java	
Number of Static Attributes (avg/max per type)	46	1.122	2.549	12	/memoranda_rrodri64/src/main/java/memoranda/Task.java	
Number of Methods (avg/max per type)	274	6.683	7.687	37	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Static Methods (avg/max per type)	61	1.488	3.768	17	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.05	0.308	2	/memoranda_rrodri64/src/main/java/memoranda/Start.java	
Number of Classes	41					
Number of Interfaces	11					
Total Lines of Code	2179					
Method Lines of Code (avg/max per method)	1251	3.734	5.184	33	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	getRepeatableEven...

7.) *After*

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.025	1.727	16	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	getRepeatableEven...
Number of Parameters (avg/max per method)		0.707	1.009	8	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	createRepeatableEv...
Nested Block Depth (avg/max per method)		1.376	0.835	8	/memoranda_rrodri64/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling	31					
Efferent Coupling	16					
Instability	0.34					
Abstractness	0					
Normalized Distance	0.66					
Depth of Inheritance Tree (avg/max per type)		1.167	0.373	2	/memoranda_rrodri64/src/main/java/memoranda/EventsScheduler.java	
Weighted methods per Class (avg/max per type)	490	16.333	17.897	71	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Children (avg/max per type)	0	0	0	0	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Overridden Methods (avg/max per type)	3	0.1	0.396	2	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Lack of Cohesion of Methods (avg/max per type)		0.127	0.237	0.679	/memoranda_rrodri64/src/main/java/memoranda/TaskListImpl.java	
Number of Attributes (avg/max per type)	30	1	1.095	4	/memoranda_rrodri64/src/main/java/memoranda/TaskListImpl.java	
Number of Static Attributes (avg/max per type)	29	0.967	2.008	7	/memoranda_rrodri64/src/main/java/memoranda/History.java	
Number of Methods (avg/max per type)	181	6.033	7.834	37	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Static Methods (avg/max per type)	61	2.033	4.278	17	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.068	0.359	2	/memoranda_rrodri64/src/main/java/memoranda/Start.java	
Number of Classes	30					
Number of Interfaces	0					
Total Lines of Code	2056					
Method Lines of Code (avg/max per method)	1251	5.169	5.457	33	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	getRepeatableEven...

8.) After refactoring, it seems that both Afferent and Efferent Coupling decreased from **34** and **21** to **31** and **16** respectively. This was certainly attributed to moving the interface classes to a new package, since it decreased dependencies from outside and inside classes. Having an "interfaces" package allows classes in the "main" package, as well as other packages, to have a single location in which to access the necessary resources.

Task 3:

1.) The Code Smell found in **CurrentNote.java** was #11 – *Too short identifier*. This was found in the method ***addNoteListener()*** where an **INoteListener** object is passed as an argument. The argument variable name was "**nl**" and was changed to "**noteListen**" to be more clear and descriptive ("**nl**" could mean anything, it is too vague).

2.) The Code Smell found in **TaskListImpl.java** was one of several long method identifiers – **calculateCompletionFromSubTasks()** – which qualifies as #10 – *Too long identifier*. The method is also used in other classes, such as **ITaskList.java** and **TaskPanel.java**. The name is descriptive, but long; this was changed to '***getSubTaskCompletion()***' to be shorter while still retaining it's descriptiveness.

3.) *After Task 3*

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2.025	1.727	16	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	getRepeatableEven...
Number of Parameters (avg/max per method)		0.707	1.009	8	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	createRepeatableEv...
Nested Block Depth (avg/max per method)		1.376	0.835	8	/memoranda_rrodri64/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
Afferent Coupling	31					
Efferent Coupling	16					
Instability	0.34					
Abstractness	0					
Normalized Distance	0.66					
Depth of Inheritance Tree (avg/max per type)		1.167	0.373	2	/memoranda_rrodri64/src/main/java/memoranda/EventsScheduler.java	
Weighted methods per Class (avg/max per type)	490	16.333	17.897	71	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Children (avg/max per type)	0	0	0	0	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Overridden Methods (avg/max per type)	3	0.1	0.396	2	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Lack of Cohesion of Methods (avg/max per type)		0.127	0.237	0.679	/memoranda_rrodri64/src/main/java/memoranda/TaskListImpl.java	
Number of Attributes (avg/max per type)	30	1	1.095	4	/memoranda_rrodri64/src/main/java/memoranda/TaskListImpl.java	
Number of Static Attributes (avg/max per type)	29	0.967	2.008	7	/memoranda_rrodri64/src/main/java/memoranda/History.java	
Number of Methods (avg/max per type)	181	6.033	7.834	37	/memoranda_rrodri64/src/main/java/memoranda/TaskImpl.java	
Number of Static Methods (avg/max per type)	61	2.033	4.278	17	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	
Specialization Index (avg/max per type)		0.068	0.359	2	/memoranda_rrodri64/src/main/java/memoranda/Start.java	
Number of Classes	30					
Number of Interfaces	0					
Total Lines of Code	2056					
Method Lines of Code (avg/max per method)	1251	5.169	5.457	33	/memoranda_rrodri64/src/main/java/memoranda/EventsManager.java	getRepeatableEven...

4.) It seems as though the metrics did not change, which was due to tackling Code Smells violations that just had to do with code style (identifiers).