Maigan Sedate
SER 316
Assignment 7
Task 1

Size

1. Total Lines of Code in main.java.memoranda: 2187

2. Largest single code file and its LOC: EventsManager.java with 329 LOC

3. CurrentNote.java used the method noteChanged to determine Total LOC. The method takes in the parameters of a notes and a Boolean of whether the current note should be saved or not. Inside the method there is a for loop that goes through the entire noteListener vector and gets each object of the vector and uses recursive properties to change the note.

Cohesion

1. Henderson-Sellers' LCOM normalizes the number of methods and variables that are shown in the class. It is a simplified and normalized metric in comparison to LCOM 1. It is calculated by the following equation:

$$\text{LCOM2} = (((1/a)\textstyle\sum_{j-1}^{a} \mu(Aj)) - m\ )\ /\ (1/m)$$

In this formula, a stands for the number of attributes on instance variables, u(Aj) stands for the number of methods accessing attribute Aj, m is the number of methods and the summation u (Aj) is summed over the attributes over all of the attributes of j.

2. The class with the Highest Cohesion is TaskListImpl.java. This class has the highest cohesion because there is a lot of duplicate ode. And not much reuse. It is definitely possible to make a new method that does the same code that is repeated throughout the other methods.

Complexity

1. The mean of the cyclomatic complexity of the main package is 1.746

2. The class that has the worst average McCabe Cyclomatic Complexity is EventsManager.java at 2.5

3. I changed the setMark method in NotImpl.java. The original Total for the Cyclomatic Complexity was 4. It declared an unnecessary element, which it then removed at the end of the method, so I was able to get rid of it. I was also able to get rid of an unnecessary else statement reducing the method to a cyclomatic complexity of 2. The total complexity number went from 1.746 to 1.74.

Package-level Coupling

1. Afferent coupling is different than efferent coupling because afferent coupling is an inward signal while efferent is an outward signal. Afferent coupling is an indicator of the package's responsibility by showing the number of classes in other packages that depend on the classes within the current package. Efferent coupling is the current packages dependency on external things and shows the number of classes that the current package's classes depend on.

2.The package with the worst Afferent coupling measure is main.java.memoranda.util with a total of 57.

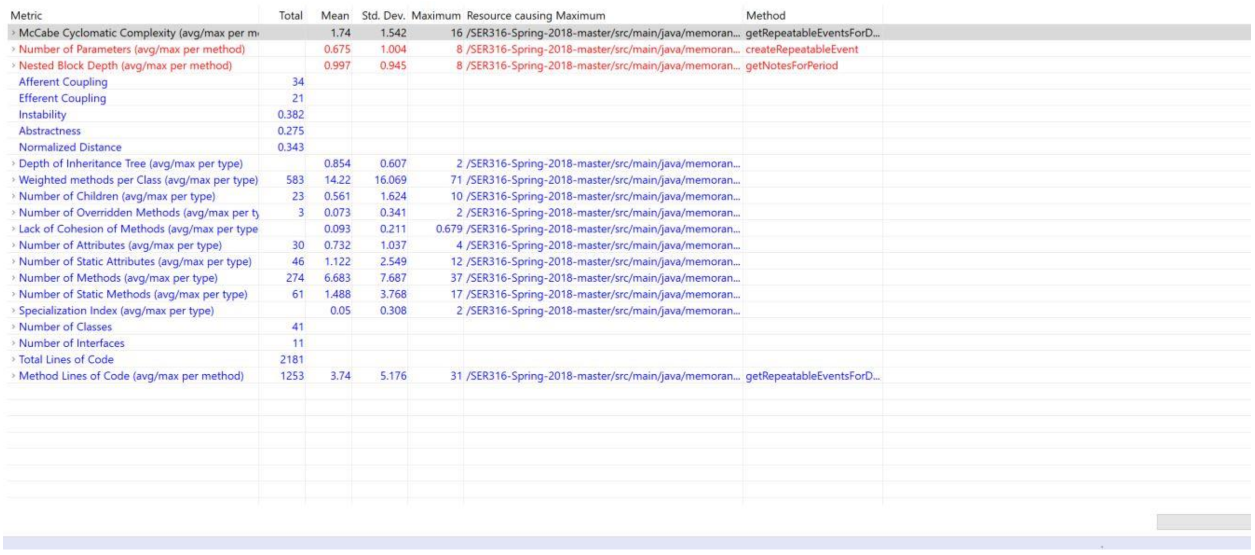
Maigan Sedate
SER 316
Assignment 7
3. The package with the worst Efferent Coupling measure is main.java.memoranda.ui with a total of 49.

Worst quality

EventsManager.java class has the worst quality. There are a few other classes that are close to the worst; however, in almost every metric area EventsManager.java consistently shows faults. The following are specific instances: EventsManager.java has the highest McCabe Cyclomatic Complexity at 2.5. It also has one of the largest number of parameters at 1.125. EventsManager.java has one of the higher numbers in the Nested Block Depth. It also has the largest total lines of Code at 329 and the largest method lines of code at 210.

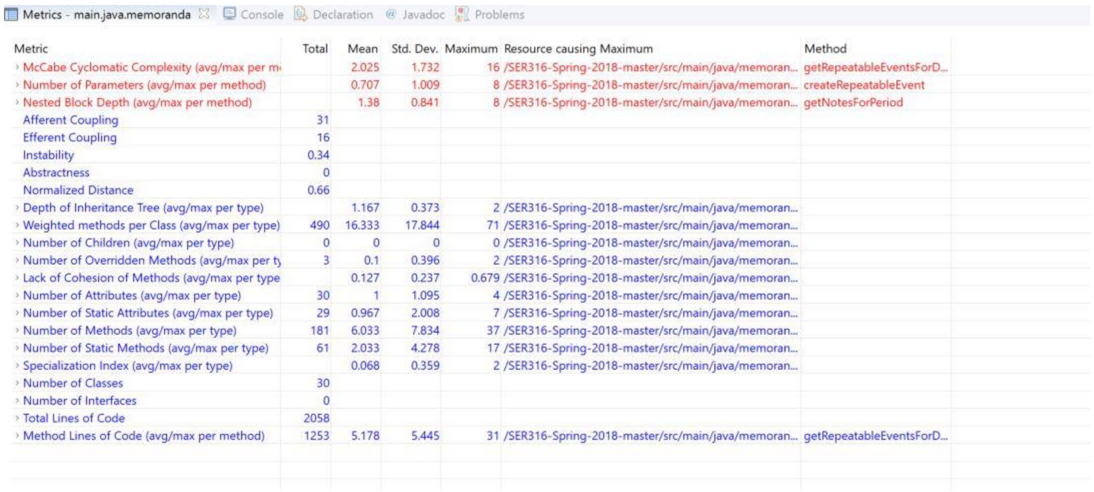
Task 2

1.



Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per m		1.74	1.542	16	/SER316-Spring-2018-master/src/main/java/memoran...	getRepeatableEventsForD...
Number of Parameters (avg/max per method)		0.675	1.004	8	/SER316-Spring-2018-master/src/main/java/memoran...	createRepeatableEvent
Nested Block Depth (avg/max per method)		0.997	0.945	8	/SER316-Spring-2018-master/src/main/java/memoran...	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	/SER316-Spring-2018-master/src/main/java/memoran...	
Weighted methods per Class (avg/max per type)	583	14.22	16.069	71	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Children (avg/max per type)	23	0.561	1.624	10	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Overridden Methods (avg/max per ty	3	0.073	0.341	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Lack of Cohesion of Methods (avg/max per type		0.093	0.211	0.679	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Attributes (avg/max per type)	30	0.732	1.037	4	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Static Attributes (avg/max per type)	46	1.122	2.549	12	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Methods (avg/max per type)	274	6.683	7.687	37	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Static Methods (avg/max per type)	61	1.488	3.768	17	/SER316-Spring-2018-master/src/main/java/memoran...	
Specialization Index (avg/max per type)		0.05	0.308	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Classes	41					
Number of Interfaces	11					
Total Lines of Code	2181					
Method Lines of Code (avg/max per method)	1253	3.74	5.176	31	/SER316-Spring-2018-master/src/main/java/memoran...	getRepeatableEventsForD...

8.



Metrics - main.java.memoranda | Console | Declaration | Javadoc | Problems

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per m		2.025	1.732	16	/SER316-Spring-2018-master/src/main/java/memoran...	getRepeatableEventsForD...
Number of Parameters (avg/max per method)		0.707	1.009	8	/SER316-Spring-2018-master/src/main/java/memoran...	createRepeatableEvent
Nested Block Depth (avg/max per method)		1.38	0.841	8	/SER316-Spring-2018-master/src/main/java/memoran...	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	/SER316-Spring-2018-master/src/main/java/memoran...	
Weighted methods per Class (avg/max per type)	490	16.333	17.844	71	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Children (avg/max per type)	0	0	0	0	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Overridden Methods (avg/max per ty	3	0.1	0.396	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Lack of Cohesion of Methods (avg/max per type		0.127	0.237	0.679	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Attributes (avg/max per type)	30	1	1.095	4	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Static Attributes (avg/max per type)	29	0.967	2.008	7	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Methods (avg/max per type)	181	6.033	7.834	37	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Static Methods (avg/max per type)	61	2.033	4.278	17	/SER316-Spring-2018-master/src/main/java/memoran...	
Specialization Index (avg/max per type)		0.068	0.359	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Classes	30					
Number of Interfaces	0					
Total Lines of Code	2058					
Method Lines of Code (avg/max per method)	1253	5.178	5.445	31	/SER316-Spring-2018-master/src/main/java/memoran...	getRepeatableEventsForD...

Maigan Sedate
SER 316
Assignment 7
Most of the metric values improved. For example, both Efferent and Afferent coupling improved by having their numbers go down after refactoring. This makes sense because Afferent coupling is the amount of dependency the package has between its own classes while efferent is the packages dependency on other package's classes. Removing the interfaces changes those dependencies and changed for the better.

Task 3

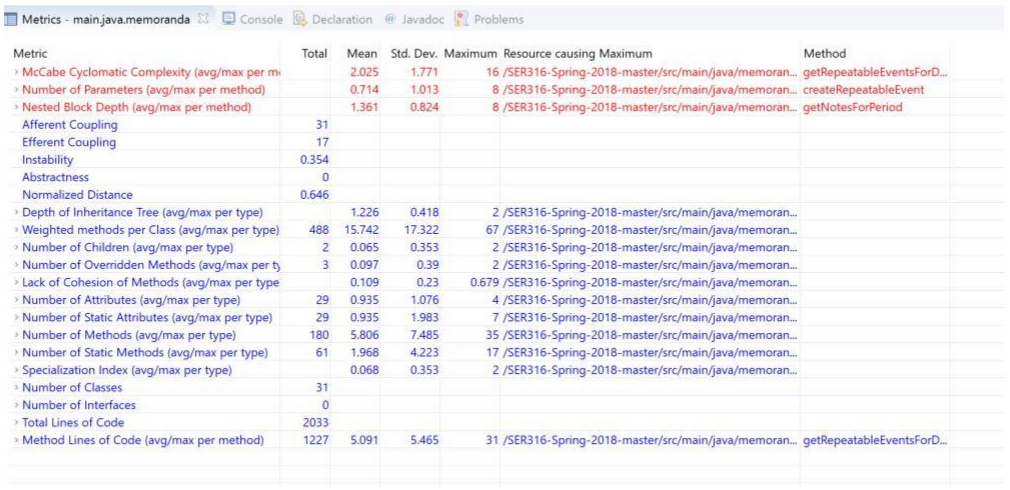
1. In class TaskListImpl.java, I noticed the methods getEarliestStartDateFromDubTasks(ITask t) and getLatestEndDateFromDubTasks(ITask t) had duplicate code. They were almost the exact same method, other than one line stating dd.before(d) and the other stating dd.after(d). The other different was the setter and getter in each method getting either the StartDate or Enddate. I made one new method called getDateFromSubTasks that each of the original methods now calls.

2. I will only mention two code smells between classes here.

First there is duplicate code in the ProjectImpl.java and TaskImpl.java. They both use the exact same method getDescription() and setDescription(String s), with absolutely no differences. I created a new parent class called BaseImpl, which created the _root variable. And then it has the getDescription() and setDescription() methods. The ProjectImpl and TaskImpl now extends BaseImpl.

Second there is an inconsistency with the Element name in TaskImpl.java versus all of the other *impl.java files. In all of the other files the Element is declared and initialized as "**private** Element _root = **null**;" However in TaskImpl.java and EventImpl.java it is "**private** Element _element = **null**;" I will not change this because we were told to change the variable name earlier in this assignment.

3.



Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per m		2.025	1.771	16	/SER316-Spring-2018-master/src/main/java/memoran...	getRepeatableEventsForD...
Number of Parameters (avg/max per method)		0.714	1.013	8	/SER316-Spring-2018-master/src/main/java/memoran...	createRepeatableEvent
Nested Block Depth (avg/max per method)		1.361	0.824	8	/SER316-Spring-2018-master/src/main/java/memoran...	getNotesForPeriod
Afferent Coupling	31					
Efferent Coupling	17					
Instability	0.354					
Abstractness	0					
Normalized Distance	0.646					
Depth of Inheritance Tree (avg/max per type)		1.226	0.418	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Weighted methods per Class (avg/max per type)	488	15.742	17.322	67	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Children (avg/max per type)	2	0.065	0.353	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Overridden Methods (avg/max per ty	3	0.097	0.39	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Lack of Cohesion of Methods (avg/max per type		0.109	0.23	0.679	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Attributes (avg/max per type)	29	0.935	1.076	4	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Static Attributes (avg/max per type)	29	0.935	1.983	7	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Methods (avg/max per type)	180	5.806	7.485	35	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Static Methods (avg/max per type)	61	1.968	4.223	17	/SER316-Spring-2018-master/src/main/java/memoran...	
Specialization Index (avg/max per type)		0.068	0.353	2	/SER316-Spring-2018-master/src/main/java/memoran...	
Number of Classes	31					
Number of Interfaces	0					
Total Lines of Code	2033					
Method Lines of Code (avg/max per method)	1227	5.091	5.465	31	/SER316-Spring-2018-master/src/main/java/memoran...	getRepeatableEventsForD...

One metric that changed is the "Number of Children" from 0 at the end of task 2 to 2 at the end of task 3. This is because I made TaskImpl and Project Impl classes of BaseImpl when I was refactoring the code smell between classes. Another metric that changed was the weighted methods per class. This number decreased from the previous screenshot, which is a good thing. This is because doing the refactoring allowed the code to lose two unnecessary methods since they were a duplicate.