

Branden Turley

SER 316 Sping18 B

Assignment 7

4/27/18

## Task 1

### Size

1. Total Lines of Code = 2187 LOC
2. EventsManager = 329 LOC
3. Method 1, it counted each line that had something on it, except for line 21 which was a comment.

### Cohesion

1. The LCOM2 stands for Lack of Cohesion Among Method of Class, which measures the extent of intersections of individual method parameter types lists with the parameter type list of all methods in the class.

LCOM2 is in the range [0,2]. LCOM2  $\geq 1$  indicates a very problematic class.

2. TaskListImpl.java because it has the highest avg Lack of Cohesion of Methods. Not sure why, it could be because of dead variables and a lack of encapsulation.

### Complexity

1. Mean = 1.746, Std. Dev. 1.547, Max 16
2. EventsManager.java with a mean of 2.5
3. Removed some of the if statements in NoteListImpl, getNotesForPeriod, it reduced complexity by 1

### Package-level Coupling

1. **Afferent Couplings (Ca):** The number of classes in other packages that depend upon classes within the package is an indicator of the package's responsibility. Afferent = incoming.

**Efferent Couplings (Ce):** The number of classes in other packages that the classes in the package depend upon is an indicator of the package's dependence on externalities. Efferent = outgoing.

2. main.java.memoranda.util with 57 Afferent Coupling

### 3. main.java.memoranda.ui with 49 Efferent Coupling

#### Worst Quality

I think based on the metrics that the worst class is the TaskListImpl.java because it has the highest lack of cohesion of methods and has the highest number of methods. I would probably break this class up to increase its simplicity and to create more cohesive classes.

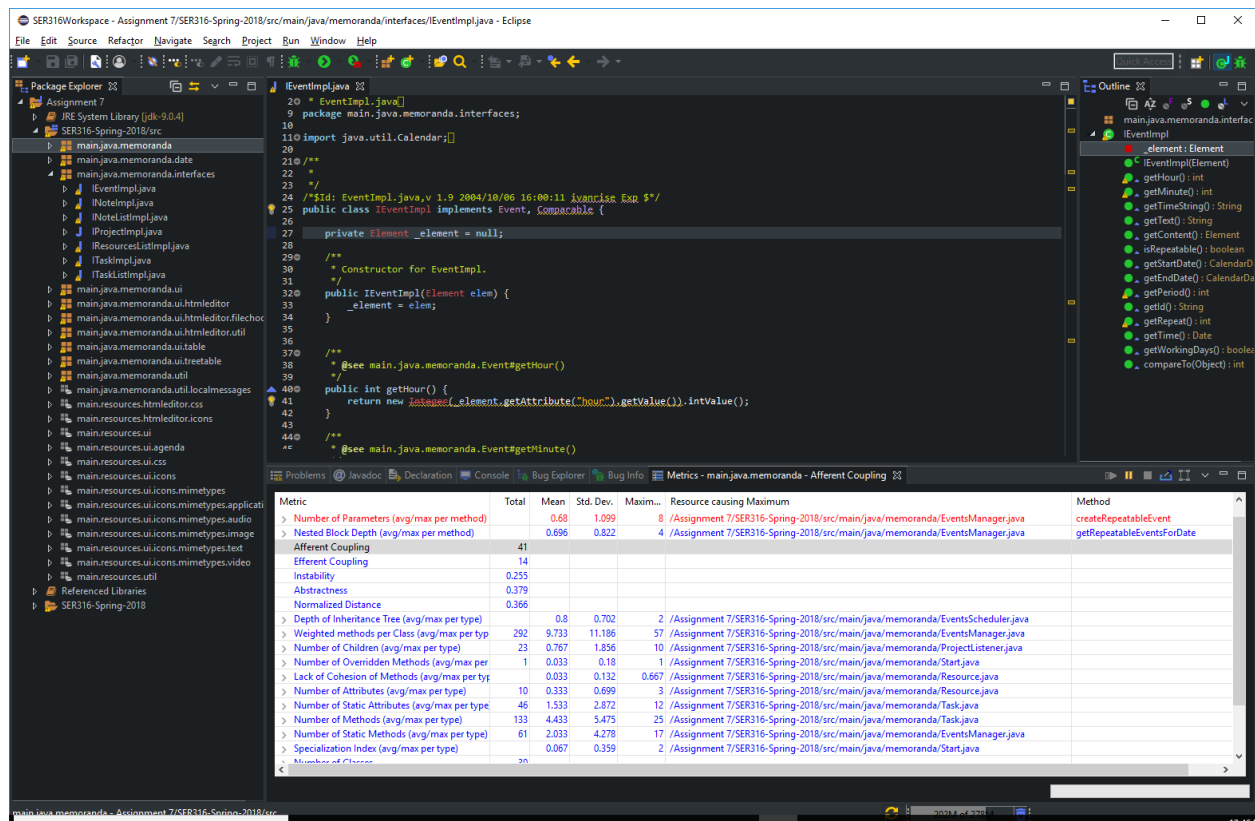
#### Task 2

##### 1.

The screenshot displays the Eclipse IDE interface with the 'Metrics - Assignment 7' view open. The Package Explorer on the left shows the project structure, including the 'main.java.memoranda' package. The Metrics table in the center provides a detailed breakdown of code quality metrics.

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> Number of Parameters (avg/max per method)	0.828	1.097		9	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/htmlEditor/HTMLEditor.java	setImageProperties
> Nested Block Depth (avg/max per method)	1.39	0.955		8	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/NoteListImpl.java	getNotesForPeriod
> Afferent Coupling (avg/max per package)	19.333	19.653		57	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui	
> Efferent Coupling (avg/max per package)	11.444	15.276		49	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui	
> Instability (avg/max per package)	0.36	0.247	0.778		/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui	
> Abstractness (avg/max per package)	0.111	0.137	0.333		/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/date	
> Normalized Distance (avg/max per package)	0.529	0.237		1	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/htmlEditor/util	
> Depth of Inheritance Tree (avg/max per type)	2.652	1.934		6	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/StickerDialog.java	
> Weighted methods per Class (avg/max per type)	3254	14.148	25.54	242	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/htmlEditor/HTMLEditor.java	
> Number of Children (avg/max per type)	60	0.261	1.405	16	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/htmlEditor/HTMLEditor.java	
> Number of Overridden Methods (avg/max per type)	59	0.257	0.691	4	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/table/TableModel.java	
> Lack of Cohesion of Methods (avg/max per type)	0.262	0.398	1.2		/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/htmlEditor/CharTablePanel.java	
> Number of Attributes (avg/max per type)	1326	5.765	14.118	101	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/htmlEditor/HTMLEditor.java	
> Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/Task.java	
> Number of Methods (avg/max per type)	1269	5.517	6.833	42	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/htmlEditor/HTMLEditor.java	
> Number of Static Methods (avg/max per type)	183	0.796	2.51	17	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/EventManager.java	
> Specialization Index (avg/max per type)	0.15	0.487		5	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui/ProjectTablePanel.java	
> Number of Classes (avg/max per package)	230	25.556	29.833	92	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ui	
> Number of Interfaces (avg/max per package)	16	1.778	3.292	11	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda	

After



8. The Weighted methods per Class changed as well as the Nested Block Depth (which is no longer red). These changed because the number of classes increased which lowered the average number of methods. This is an improvement because it makes the program more compartmental and organized.

### Task 3

1. CurrentNote.java line 21 – 24, unused / dead code. Fixed by simply commenting it out for reference.

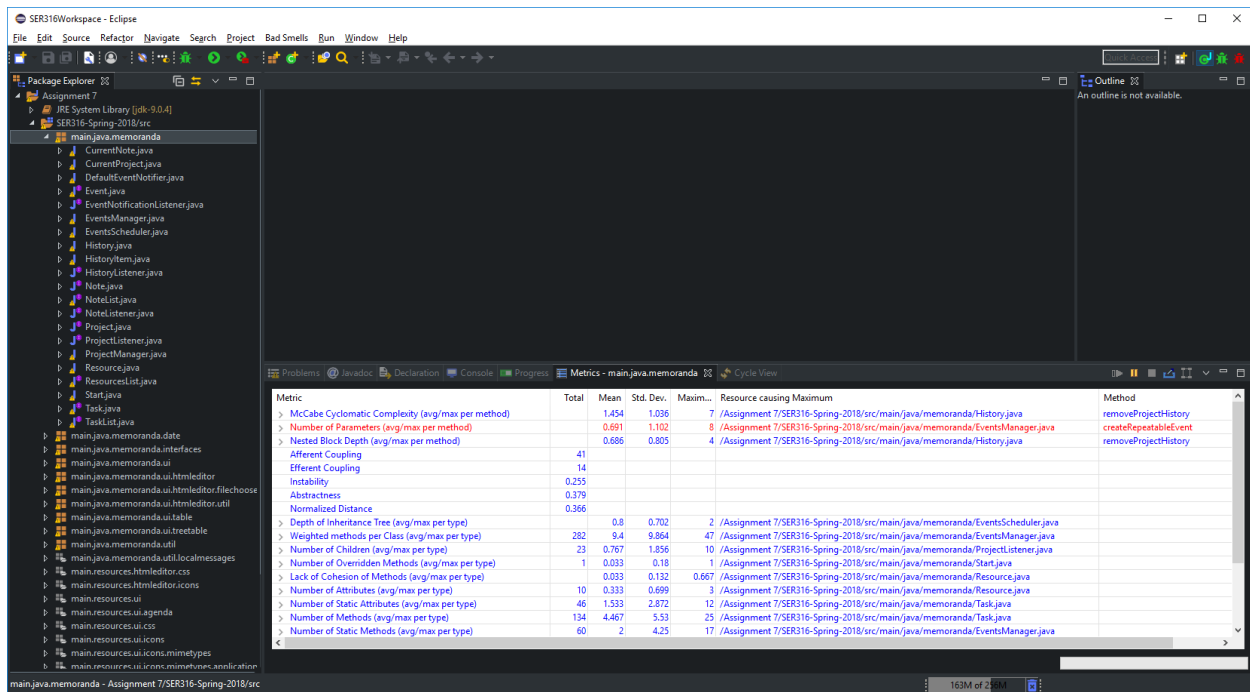
2. Large Class

Replace conditional structure with polymorphic method invocation in EventsManager.java.

Event.java add reference to getRepeatableEventsForDate...

IEventImpl.java, added method to check for repeatable events for date.

### 3. AFTER



Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)	1,454	1,026		7	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/History.java	removeProjectHistory
> Number of Parameters (avg/max per method)	0.691	1.102		8	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/EventManager.java	createRepeatableEvent
> Nested Block Depth (avg/max per method)		0.686	0.805	4	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/History.java	removeProjectHistory
Afferent Coupling	41					
Efferent Coupling	14					
Instability	0.255					
Abstractness	0.379					
Normalized Distance	0.366					
> Depth of Inheritance Tree (avg/max per type)		0.8	0.702	2	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/EventsScheduler.java	
> Weighted methods per Class (avg/max per type)	282	9.4	9.864	47	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/EventManager.java	
> Number of Children (avg/max per type)	23	0.767	1.856	10	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/ProjectListener.java	
> Number of Overridden Methods (avg/max per type)	1	0.033	0.18	1	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/Start.java	
> Lack of Cohesion of Methods (avg/max per type)		0.033	0.132	0.667	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/Resource.java	
> Number of Attributes (avg/max per type)	10	0.333	0.699	3	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/Resource.java	
> Number of Static Attributes (avg/max per type)	46	1.533	2.872	12	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/Task.java	
> Number of Methods (avg/max per type)	134	4.467	5.53	25	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/Task.java	
> Number of Static Methods (avg/max per type)	60	2	4.25	17	/Assignment 7/SER316-Spring-2018/src/main/java/memoranda/EventManager.java	

4. After refactoring the Cyclomatic Complexity improved. It changed for the better because by refactoring the class it made it more simple and straightforward by breaking up the complexity.