**Task 1**

**Size**

1) 2187 LOC

2) EventsManager.java with 329 LOC

3) It looks to be using something close to method 1. Specifically, it is counting all lines with anything on them even if it is just an empty brace. It also includes the package and import lines.

**Cohesion**

1) LCOM2, the Henderson-Sellers method, produces a value from 0 to 2 inclusive. It is found via the following formula:

$$LCOM\ HS = \frac{1}{M-1}\left(M - \frac{1}{F}\sum_{i=0}^{n} MF\right)$$

(Information and figure from https://codenforcer.com/metrics#lcom)

where M is the total number of methods in the class, F the number of instance fields, and MF the number of class methods with access to a given field. The lower the value, the better the cohesion with a value of 0 resulting from all methods using all class fields in the class.

2) There are a number of classes with an LCOM2 value of 0 indicating perfect cohesion. The class EventImpl has a value of zero because it only has one class member, _elem, that is used in most of its methods. There are also many more methods than fields.

**Complexity**

1) The mean McCabe Cyclomatic Complexity of main.java.memoranda is 1.746.

2) The class EventsManager has the worst complexity at 2.5.

3) So I ended up moving past this one forgetting that what I would do later would change what I have now. In any case coming back to this one and looking things over more closely I realized that what I wanted to do to reduce complexity here I did later in task 3 removing the code duplication code smell in NoteListImpl. This reduced cyclomatic complexity from 2.033 to 2.004.

**Package-level Coupling**

1) Afferent coupling is the number of classes in other packages that depend upon classes within this package while efferent coupling is a measure of how many classes in other packages classes in this package depend upon. The first provides information on how many classes might be affected by changes in this package while the second indicates this packages dependence on externalities. The first is fanning inward, the second fanning outward.

2) The package main.java.memoranda.util has the worst afferent coupling at 57.

3) The package mainjava.memoranda.ui has the worse efferent coupling at 49.

**Worst Quality**

I'd say that the NoteListImpl class is the worst. It has very high cyclomatic complexity, ugly, nested conditionals and loops up to seven and eight deep, and blocks of duplicated code. These factors together result in complex, hard to understand code that is duplicated so that any attempt to make changes to it in the future in its current state would be made more difficult by having to do so in multiple places to keep things working.

**Task 2**

| Metric | Total | Mean | Std. Dev. | Maxi... | Resource causing Maximum | Method |
|---|---|---|---|---|---|---|
| McCabe Cyclomatic Complexity (avg/max per method) | | 1.746 | 1.547 | 16 | /SER316-Spring-2018-master/src/... | getRepeatableEventsFor... |
| Number of Parameters (avg/max per method) | | 0.675 | 1.004 | 8 | /SER316-Spring-2018-master/src/... | createRepeatableEvent |
| Nested Block Depth (avg/max per method) | | 0.997 | 0.945 | 8 | /SER316-Spring-2018-master/src/... | 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/... | |
| Weighted methods per Class (avg/max per type) | 585 | 14.268 | 16.081 | 71 | /SER316-Spring-2018-master/src/... | |
| Number of Children (avg/max per type) | 23 | 0.561 | 1.624 | 10 | /SER316-Spring-2018-master/src/... | |
| Number of Overridden Methods (avg/max per type) | 3 | 0.073 | 0.341 | 2 | /SER316-Spring-2018-master/src/... | |
| Lack of Cohesion of Methods (avg/max per type) | | 0.093 | 0.211 | 0.679 | /SER316-Spring-2018-master/src/... | |
| Number of Attributes (avg/max per type) | 30 | 0.732 | 1.037 | 4 | /SER316-Spring-2018-master/src/... | |
| Number of Static Attributes (avg/max per type) | 46 | 1.122 | 2.549 | 12 | /SER316-Spring-2018-master/src/... | |
| Number of Methods (avg/max per type) | 274 | 6.683 | 7.687 | 37 | /SER316-Spring-2018-master/src/... | |
| Number of Static Methods (avg/max per type) | 61 | 1.488 | 3.768 | 17 | /SER316-Spring-2018-master/src/... | |
| Specialization Index (avg/max per type) | | 0.05 | 0.308 | 2 | /SER316-Spring-2018-master/src/... | |
| Number of Classes | 41 | | | | | |
| Number of Interfaces | 11 | | | | | |
| Total Lines of Code | 2187 | | | | | |
| Method Lines of Code (avg/max per method) | 1259 | 3.758 | 5.212 | 33 | /SER316-Spring-2018-master/src/... | getRepeatableEventsFor... |

AFTER

| Metric | Total | Mean | Std. Dev. | Maxi... | Resource causing Maximum | Method |
|---|---|---|---|---|---|---|
| McCabe Cyclomatic Complexity (avg/max per method) | | 2.033 | 1.737 | 16 | /SER316-Spring-2018-master/src/... | getRepeatableEventsFor... |
| Number of Parameters (avg/max per method) | | 0.707 | 1.009 | 8 | /SER316-Spring-2018-master/src/... | createRepeatableEvent |
| Nested Block Depth (avg/max per method) | | 1.38 | 0.841 | 8 | /SER316-Spring-2018-master/src/... | 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/... | |
| Weighted methods per Class (avg/max per type) | 492 | 16.4 | 17.85 | 71 | /SER316-Spring-2018-master/src/... | |
| Number of Children (avg/max per type) | 0 | 0 | 0 | 0 | /SER316-Spring-2018-master/src/... | |
| Number of Overridden Methods (avg/max per type) | 3 | 0.1 | 0.396 | 2 | /SER316-Spring-2018-master/src/... | |
| Lack of Cohesion of Methods (avg/max per type) | | 0.127 | 0.237 | 0.679 | /SER316-Spring-2018-master/src/... | |
| Number of Attributes (avg/max per type) | 30 | 1 | 1.095 | 4 | /SER316-Spring-2018-master/src/... | |
| Number of Static Attributes (avg/max per type) | 29 | 0.967 | 2.008 | 7 | /SER316-Spring-2018-master/src/... | |
| Number of Methods (avg/max per type) | 181 | 6.033 | 7.834 | 37 | /SER316-Spring-2018-master/src/... | |
| Number of Static Methods (avg/max per type) | 61 | 2.033 | 4.278 | 17 | /SER316-Spring-2018-master/src/... | |
| Specialization Index (avg/max per type) | | 0.068 | 0.359 | 2 | /SER316-Spring-2018-master/src/... | |
| Number of Classes | 30 | | | | | |
| Number of Interfaces | 0 | | | | | |
| Total Lines of Code | 2064 | | | | | |
| Method Lines of Code (avg/max per method) | 1259 | 5.202 | 5.486 | 33 | /SER316-Spring-2018-master/src/... | getRepeatableEventsFor... |

Several metrics changed for the better in main.java.memoranda. Afferent and efferent coupling improved for example. Cyclomatic complexity however got a bit worse. This makes sense as the interfaces would certainly lower this number and so by removing them from the package the average complexity increased.

**Task 3**

1) main.java.memoranda.NoteListImpl

The smell here was duplicate code. The methods getAllNotes() and getMarkedNotes() had identical code throughout most of the method. The only thing that distinguished them was a boolean check at the end where the getAllNotes method would simply add all notes to the returned collection while getMarkedNotes would check if a note was marked and only add them marked ones to the Collection. This was over a dozen lines of duplicate code and would be annoying to work with in the future should the year, month, day system change.

I fixed this by factoring out the identical looping code into a new method that takes a boolean parameter called marked. If marked is true then the check for marked notes is performed and the collection returned, if false all found notes are added to the returned collection. The parameter and check within the code is to avoid having to iterate over the returned collection again to check for marked notes rather than do it in place.

2) Changes were made in EventsManager in main.java.memoranda and EventsPanel in the UI package.

The smell between classes is primitive obsession. Methods associated with events were commonly passing the parameters “int hh, int mm” around to represent a particular time. A new class called Time has been added to encapsulate these primitives. All methods in EventManager that had these as parameters were changed to use a time object and access the data from that object. Other methods in

EventsManager and EventsPanel were modified to utilize this new time object as well (comments are in the code in the places changes were made).

While Time is only a data container at the moment one could envision adding more to it in the future such as the ability to convert between a 12 and 24 hour time for instance.

AFTER

| Metric | Total | Mean | Std. Dev. | Maxi... | Resource causing Maximum | Method |
|---|---|---|---|---|---|---|
| McCabe Cyclomatic Complexity (avg/max per method) | | 2.004 | 1.729 | 16 | /SER316-Spring-2018-master/src/... | getRepeatableEventsFor... |
| Number of Parameters (avg/max per method) | | 0.691 | 0.951 | 7 | /SER316-Spring-2018-master/src/... | createRepeatableEvent |
| Nested Block Depth (avg/max per method) | | 1.366 | 0.848 | 8 | /SER316-Spring-2018-master/src/... | getNotesForPeriod |
| Afferent Coupling | 31 | | | | | |
| Efferent Coupling | 16 | | | | | |
| Instability | 0.34 | | | | | |
| Abstractness | 0 | | | | | |
| Normalized Distance | 0.66 | | | | | |
| Depth of Inheritance Tree (avg/max per type) | | 1.161 | 0.368 | 2 | /SER316-Spring-2018-master/src/... | |
| Weighted methods per Class (avg/max per type) | 493 | 15.903 | 17.606 | 71 | /SER316-Spring-2018-master/src/... | |
| Number of Children (avg/max per type) | 0 | 0 | 0 | 0 | /SER316-Spring-2018-master/src/... | |
| Number of Overridden Methods (avg/max per type) | 3 | 0.097 | 0.39 | 2 | /SER316-Spring-2018-master/src/... | |
| Lack of Cohesion of Methods (avg/max per type) | | 0.14 | 0.244 | 0.679 | /SER316-Spring-2018-master/src/... | |
| Number of Attributes (avg/max per type) | 32 | 1.032 | 1.092 | 4 | /SER316-Spring-2018-master/src/... | |
| Number of Static Attributes (avg/max per type) | 29 | 0.935 | 1.983 | 7 | /SER316-Spring-2018-master/src/... | |
| Number of Methods (avg/max per type) | 185 | 5.968 | 7.756 | 37 | /SER316-Spring-2018-master/src/... | |
| Number of Static Methods (avg/max per type) | 61 | 1.968 | 4.223 | 17 | /SER316-Spring-2018-master/src/... | |
| Specialization Index (avg/max per type) | | 0.066 | 0.353 | 2 | /SER316-Spring-2018-master/src/... | |
| Number of Classes | 31 | | | | | |
| Number of Interfaces | 0 | | | | | |
| Total Lines of Code | 2068 | | | | | |
| Method Lines of Code (avg/max per method) | 1252 | 5.089 | 5.479 | 33 | /SER316-Spring-2018-master/src/... | getRepeatableEventsFor... |

4) Both cyclomatic complexity and parameters per a method decreased, an improvement. The first decreased due to refactoring duplicate code in NoteListImpl. This duplicated code has high complexity and so by factoring it out of two methods into one method the number of paths the program could go through decreased.

Parameters per a method decreasing is a result of dealing with the primitive obsession code smell. By combining hours and minutes into one object to pass around less parameters overall had to be passed.