Eclipse Variation

TASK 1

Size

1. 2187 Total LOC found is java.main.memoranda
2. EventsManager.java
3. Takes every non blank and non commented line as a line of code

Cohesion

1. It's a way to calculate cohesion of the class lower number being the better its calculated by
    - m being the number of methods in the code
    - a being the number of variables
    - mA being the number of methods that access a variable

    LOCOM2 = 1 – sum(mA)/(m*a)
2. The highest Cohesion of Methods is TaskListImpl I think its partially to do with the large amount of coding needed just to create tasks or calculate tasks and the amount of gets and sets used in the code while. This would make it look like that I'm not accessing the variable of the code when I actually am in a different way.

Complexity

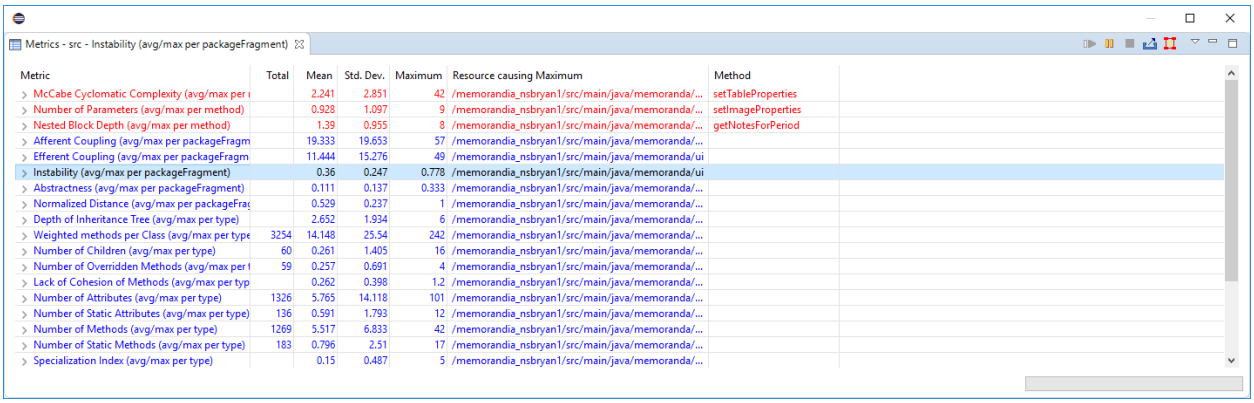
1. Mean 1.746 Std. Dev. 1.547 Max 16
2. EventsManager.java has the worst because in most of the functions you have 1 or more if statements that create new branches in the code to follow
3. Couldn't find any way to reduce compliexity think its an issue with how eclipse is seeing the metrics

Package-level Coupling

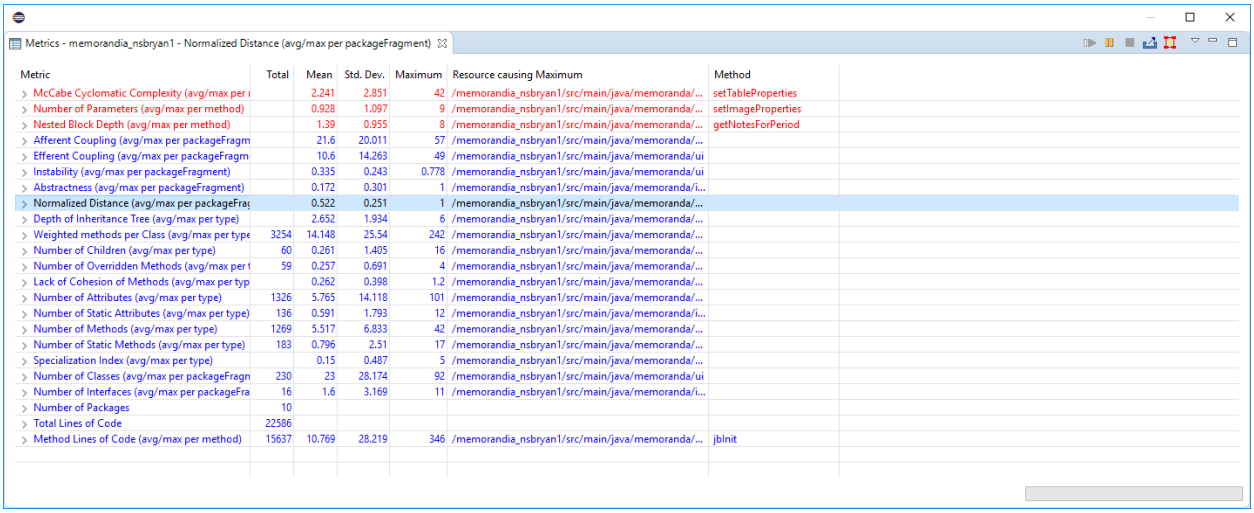
1. Deffintions
    a. Afferent coupling: The amount of other packages relying on the specific package
    b. Efferent coupling: The amount of packages the current package is reliant on
2. java.main.memoranda.util has the worst afferent Coupling at a value of 57
3. java.main.memoranda.ui has the worst efferent coupling at a value 49

TASK 2

Initial Screenshot



After Work Screen Shot



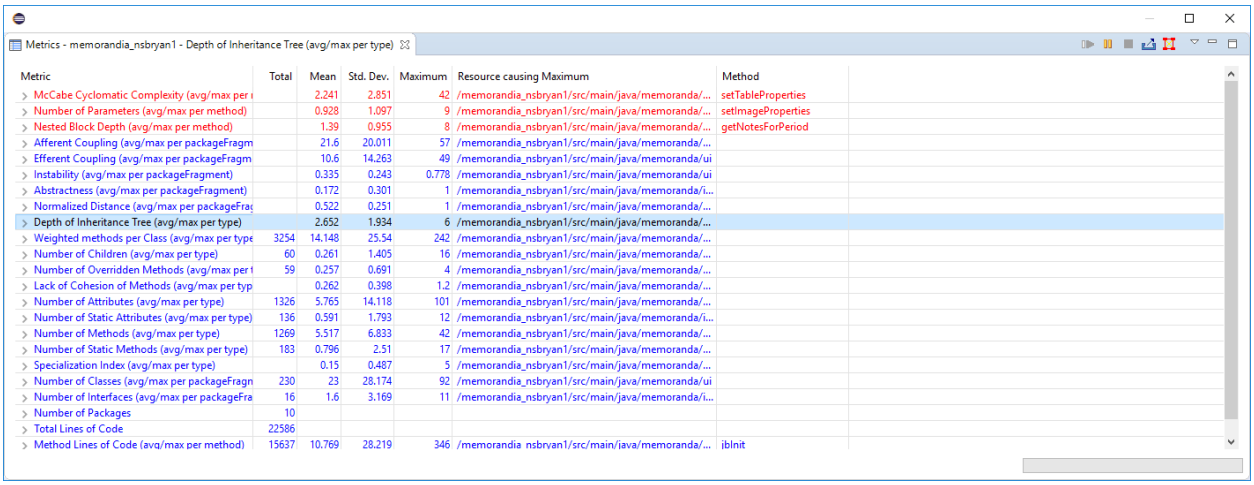
8. A few metrics changed Afferent and Efferent Coupling changed not surprise there as they specially deal with the issues of packages. Now if its good or not is an issue of choice. It increases the abstractness of the program from .111 to .172 which is a small increase but I'm not 100% sure what that means in an increase however. I will say it's a positive change has one of the advantages of OOP is that we can have a higher level of abstraction and an increase in that regard should be fine.

TASK 3

Code smell in class: TaskImpl.java line 35: Code smell was too short identifies. The identify of the variable was _tl which is confusing in coding anything as it doesn't let me know what the variable was or what it does.

Code smell in classes EventsManger line 91: Code Too long identifiers: The identifier which I forgot the name of what it was was too long and was confusing in what it did. Renamed it so it made it more clear and shorter.

After



Nothing changed as I didn't do enough to really fix anything major that would change any major issues