

## Assignment No. 3

### Overview:

Creating a version control system where we are recording the commits made by users. Then we are adding said commits in our system and performing operations on the commits.

### Files:

The program contains 4 files:

- Commit.java - Commit class which contains all the variables needed with a commit..
- CommitManager.java - A class with implementation of our reporting functions.
- A3.java - Contains the main method where the commits are created and operations take place.
- GraphM.java - Contains the helper methods which our operations need.

### Flow of the program:

1. Create an instance of the CommitManager method.
2. Call the addCommit method to add commits.
3. Check whether the Task in the commit is a Task or feature.
4. If true add the commit else throw an IllegalArgumentException.
5. Set the threshold using the componentMinimum method.
6. Once the threshold is set, find the components using the softwareComponents method.
7. In the software components method create a matrix of all current occurrences of files in commits and as per that create component and add them.
8. Next in repititioninBugs add the number of times each bug was added with the number of its file occurrences.
9. Check if it is touching a file more than threshold times and return those files.
10. In the broadFeatures() for each commit store the no. of features with their commit files.
11. Iterate through components and for each component(set) use intersection with feature commitfiles(set) created before.
12. If intersection gives files more than 0 i.e. two sets have one or more files common increment the count.
13. When count is equal to threshold return the feature.
14. Using same concept for getexperts just storing experts instead of features.
15. In busyclasses store all filenames and their frequency.
16. Sort the map in descending order and the number of limit files.
17. If there are files with similar frequency as the last file added its frequency is compared and all the files are added.

### Logic behind methods:

Set> softwareComponents ( ):

Create a weighted matrix for all commits within the time frame.

1. For A[0][0] create a new set of strings with A[0] as the first string.
2. Check all A[0][j] elements for paths more than threshold.
3. Add the strings to the sets.
4. Next go to A[1][0] and check if A[1] exists in the set.
5. If yes -> Move ahead for other elements  
no-> Break, Create a new set until the graph ends.
6. Keep adding the sets created in a main set which will be returned.

Set repetitionInBugs ( int threshold ):

Create a frequency list associated with each bug which will have its files and frequency of each file stored in a map. Then a local variable is created which will store the highest frequency of each file in the bug. It will store the highest file frequency for the bugs and if that frequency is greater than or equal to the threshold the bug is added to the set.

Set broadFeatures ( int threshold ):

For each commit store the number of features with their commit files in a map. Iterate through components for each component use intersection with commit files in features. If after intersection the files has more than 0 then increment a count and move to the second component. When count is equal to threshold return the feature number meaning the file has touched the component atleast threshold times.

Set experts ( int threshold ):

Uses the same implementation just checks for experts instead of features.

List busyClasses ( int limit ):

In this method a new map is created which will have all files and its occurrences stored. Then we rearrange this map in descending order and save its top "limit" elements in our busyClasses.string to be returned. Lastly it checks whether there are other elements which have similar occurrences then add them.

### Test Cases

Input Validation:

Method	Test Case
addCommit()	Empty string for developer
	Developer is not string
	commitTime is 0.
	commitTime is greater than 0.
	commitTime is less than 0.

	Task is empty.
	Task is null
	Task is a string that is accepted
	commitFiles is empty.
	commitFiles is not a set.
setTimeWindow()	startTime is not integer.
	startTime is 0
	startTime is null
	startTime is an acceptable integer.
	endTime is not integer.
	endTime is 0.
componentMinimum(int threshold)	Threshold is not int.
	Threshold is null.
	Threshold is 0.
	Threshold is an acceptable integer.
repetitionInBugs ( int threshold )	Threshold is not int.
	Threshold is null.
	Threshold is 0.
	Threshold is an acceptable integer.
broadFeatures ( int threshold )	Threshold is not int.
	Threshold is null.
	Threshold is 0.
	Threshold is an acceptable integer.
experts ( int threshold )	Threshold is not int.
	Threshold is null.

	Threshold is 0.
	Threshold is an acceptable integer.
busyClasses ( int limit )	Limit is not int.
	Limit is null.
	Limit is 0.
	Limit is an acceptable integer.

## Boundary Cases

addCommit()	Developer is an empty string.
	Developer already exists.
	CommitTime is equal to endtime.
	Commitfiles is empty.
	Task already exists.
	Commitfiles contains a single file.
	Commitfiles contains all the files that exists.
setTimeWindow(int startTime, int endTime)	startTime is 0.
	endTime is 0.
	startTime is equal to endtime
componentMinimum(int threshold)	Threshold is 0.
	Threshold is 1.
	Threshold is more than the number of commits.
repetitionInBugs ( int threshold )	Threshold is 0.
	Threshold is 1.
	Threshold is more than the number of commits.
broadFeatures ( int	Threshold is 0.

threshold )	
	Threshold is 1.
	Threshold is more than the number of commits.
experts ( int threshold )	Threshold is 0.
	Threshold is 1.
	Threshold is more than the number of commits.
busyClasses ( int limit )	Limit is 0
	Limit is equal to the number of classes.

## Control Flow

addCommit	Task is not a valid Bug or Feature.
	CommitTime does not fall in range of startTime and endTime
setTimeWindow( int startTime, int endTime )	startTime is greater than endTime.
	startTime is equal to endTime..
	setTimeWindow is called without calling clearWindow.
repetitionInBugs ( int threshold )	There are no bugs in commits.
broadFeatures ( int threshold )	There are no features in commits.
experts ( int threshold )	There is only one developer.
busyClasses ( int limit )	All initial files have same frequency.
	All files have the same frequency.

## Data Flow

addCommit	Invoke before calling setTimeWindow.
-----------	--------------------------------------

boolean setTimeWindow( int startTime, int endTime )	Invoked before clearTimeWindow.
	Invoked after clearTimeWindow.
componentMinimum( int threshold )	Invoked after softwareComponents.
softwareComponents ( )	Invoke before componentMinimum.
repetitionInBugs ( int threshold )	Invoked before addcommits.
broadFeatures ( int threshold )	Invoked before addcommits
	Invoked before softwareComponents.
experts ( int threshold )	Invoked before addcommits
	Invoked before softwareComponents.
busyClasses ( int limit )	Invoked before addcommits
	Invoked before softwareComponents.