

## Homework 4

### Group 1

#### Names:

- \* James Rodgers
- \* Yikchun Ng
- \* Napoleon de Mesa

### Index

Question & Answers	Page
List of the core functionality of the systems.	2
The relationship between major classes and interfaces.	2
Why you think those relations are good or not good.	2
Find instances of design patterns. Explain how they're used?	2
If Java Streams or Lambdas have been used. If so, how?	3

### Appendix

Number of commits with terms "bug", or "bugfix" in the commit message	4
A graph to show the frequency of code commits	5
A graph to show the code growth over time	6
External dependencies	7
Number of passing test cases	8
Major code contributors	9

1) List of the core functionality of the systems.

The core functionality of LeakCanary library is to provide memory leak detection for Android and Java.

2) The relationship between major classes and interfaces. Why you think those relations are good or not good.

All the classes in LeakCanary are closely associated. An example would be the relationship between two final classes, the core LeakCanary class and the RefWatcher class. The LeakCanary class consists the function that allows users to install and create an active RefWatcher into the system right out of the box. The function of RefWatcher is that it will constantly watch over references to the files in user's system. And as it detects any irregular activity of those references, it triggers the HeapDump class and then finds the file with the key associated to the reference out-of-ordinary, so as to figure out the source of memory leak. And to recover a file from the heap, it triggers the HeapDumper class, which basically reverts the memory in the heap back to a file.

From LeakCanary to RefWatcher to HeapDump and HeapDumper, it demonstrates some relationship among the classes in this project. This is a good example of software design because each of the classes is well-defined, it shows the interaction in between different classes and the fluidity that comes from the initiation of one major class leading to a second and a third class. Not a single class in this project was left out in this collectivity. Every class relates and supplies something to other classes. These relations not only had built up a structure for the project, they also set up a sample of good programming practice for the new programmers to follow.

3) Find instances of design patterns. Explain how they're used.

- One instance of a design pattern used is the Flyweight Pattern. The flyweight pattern is used when you need to create many objects of a class. What this can do is to apply it to reduce the load on memory by sharing objects.
- Another instance of design pattern used is the Mediator Pattern. "Mediator design pattern is used to provide a centralized communication medium between different objects in a system. Mediator design pattern is very helpful in an enterprise application where multiple objects are interacting with each other. If the objects interact with each other directly, the system components are tightly-coupled with each other that makes maintainability cost higher and not flexible to extend easily."
- <https://www.journaldev.com/1827/java-design-patterns-example-tutorial>

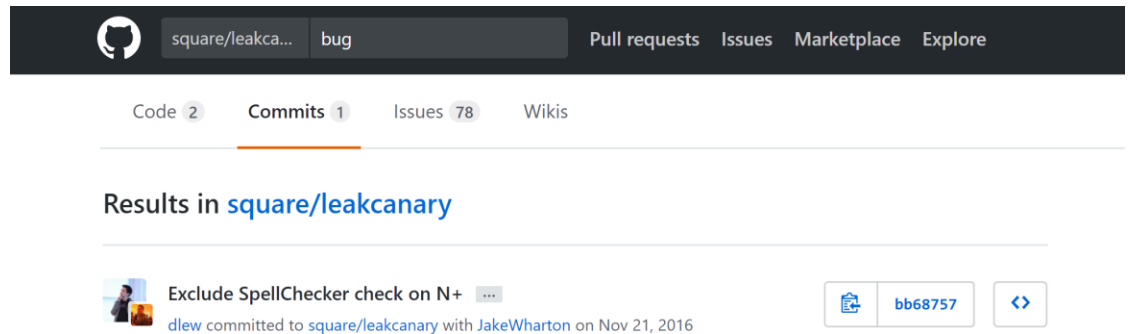
4) If Java Streams or Lambdas have been used. If so, how?

- Java 8 Streams were not used in the LeakCanary code.
- Java 8 SE Lambdas were not used in the LeakCanary code.

# Appendix

Number of commits with terms “bug”, or “bugfix” in the commit message

- Number of commits with term “bug”:



This screenshot shows the GitHub search interface for the repository 'square/leakcanary'. The search term 'bug' is entered in the search bar. The results are filtered to 'Commits', showing 1 result. The commit message is 'Exclude SpellChecker check on N+'. The commit was made by 'dlew' on Nov 21, 2016. The commit hash is 'bb68757'.

square/leakca... bug Pull requests Issues Marketplace Explore

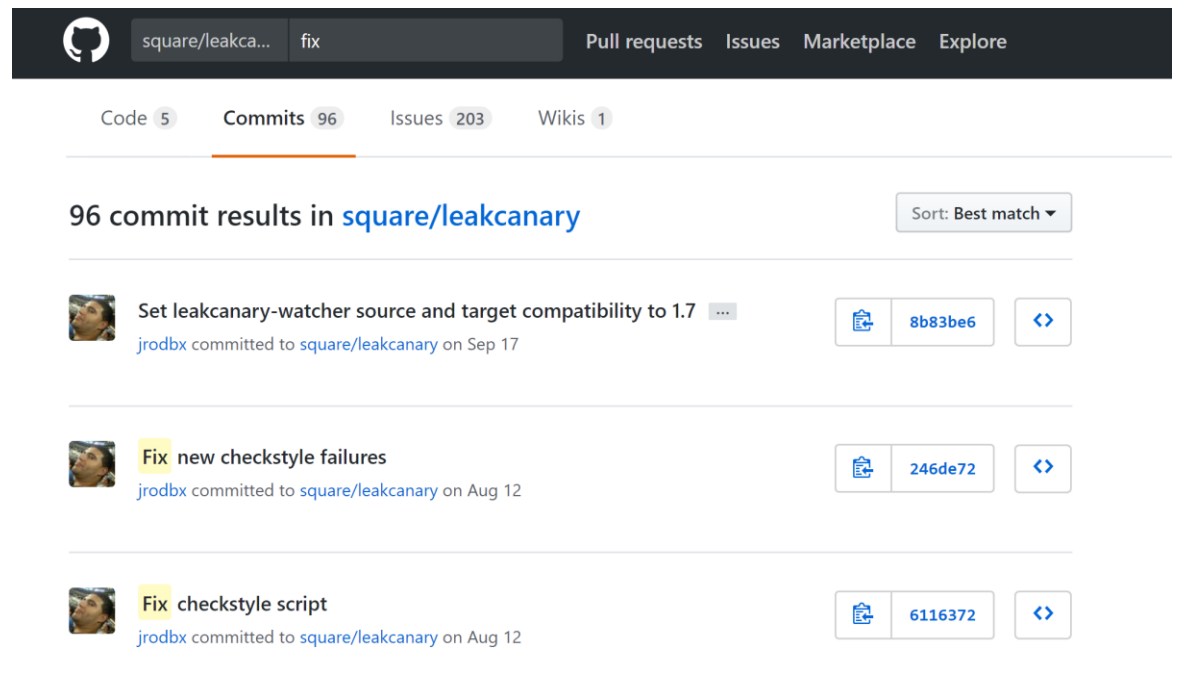
Code 2 Commits 1 Issues 78 Wikis

Results in square/leakcanary

Exclude SpellChecker check on N+ ...  
dlew committed to square/leakcanary with JakeWharton on Nov 21, 2016

bb68757

- Number of commits with term “fix”:



This screenshot shows the GitHub search interface for the repository 'square/leakcanary'. The search term 'fix' is entered in the search bar. The results are filtered to 'Commits', showing 96 results. The first three results are displayed, all with the prefix 'Fix'. The commit messages are 'Set leakcanary-watcher source and target compatibility to 1.7', 'new checkstyle failures', and 'checkstyle script'. The commits were made by 'jrodbx' on Sep 17, Aug 12, and Aug 12, respectively. The commit hashes are '8b83be6', '246de72', and '6116372'.

square/leakca... fix Pull requests Issues Marketplace Explore

Code 5 Commits 96 Issues 203 Wikis 1

96 commit results in square/leakcanary Sort: Best match

Set leakcanary-watcher source and target compatibility to 1.7 ...  
jrodbx committed to square/leakcanary on Sep 17

8b83be6

Fix new checkstyle failures  
jrodbx committed to square/leakcanary on Aug 12

246de72

Fix checkstyle script  
jrodbx committed to square/leakcanary on Aug 12

6116372

A graph to show the frequency of code commits.



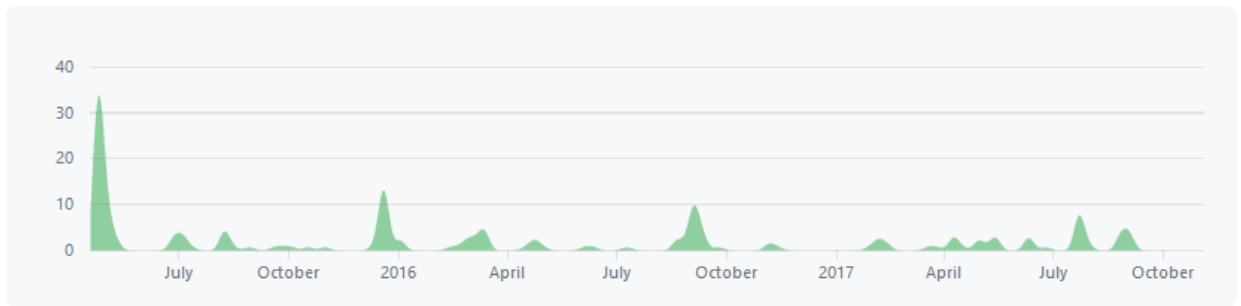
- The top graph shows the number of additional lines and deleted lines in the code in the past two years.
- The middle graph shows the number of commits done in the past year to the repository.
- The bottom graph shows the average number of commits a day that happened since the creation of the repository.

## Graph to show the code growth over time

May 3, 2015 – Nov 18, 2017

Contributions: **Commits** ▼

Contributions to master, excluding merge commits



## External dependencies

```
java.io;  
java.util;  
java.lang;  
android.app;  
android.os;  
android.util;  
android.view;  
android.text;  
android.graphics;  
com.squareup.leakcanary;
```

## Number of passing test cases

The number of passing test cases that were 17, they are located in the following locations:

- leakcanary/leakcanary-analyzer/src/test/java/com/squareup/leakcanary/
- leakcanary/leakcanary-watcher/src/test/java/com/squareup/leakcanary/
- leakcanary/leakcanary-sample/src/test/java/com/example/leakcanary/



## Major code contributors

- 1) Pierre-Yves Ricau is the main contributor who goes by Pyricau on GitHub.



- 2) John Rodriguez is the second major code contributor who goes by Jrodbx on GitHub.



Both Pierre-Yves and John work at Square Inc. who has the copyright license for this software.