CS 1632 - DELIVERABLE 6: Static Analysis of the Sieve of Eratosthenes

Michael Bowen

Summary

For fixing the code using static analysis, I first ran both FindBugs and Checkstyle on the code presented. From there, I examined the list of problems to determine what changes might need to be made. It was pretty straightforward, actually. The ease of use of the two modules allowed for quick and simple code changes.

I started with FindBugs and simply ran down the list, correcting each in turn, then rerunning FindBugs after each change and recompile. First I fixed the calculateMax() function to actually throw the new IllegalArgumentException() instead of just creating it. Then, I updated the if statement in printSieve() to correct the logic and placed the null check before the length check. While I was there, I also updated the function’s name to match accepted casing standards. Incidentally, this fix also fixed one of Checkstyle’s complaints. Finally, I looked up the functions from the Integer class based on FindBugs’s suggestion to make sure that the suggested change would lead to the same results, then I changed the convertResults function to no longer create new Integer objects for each result.

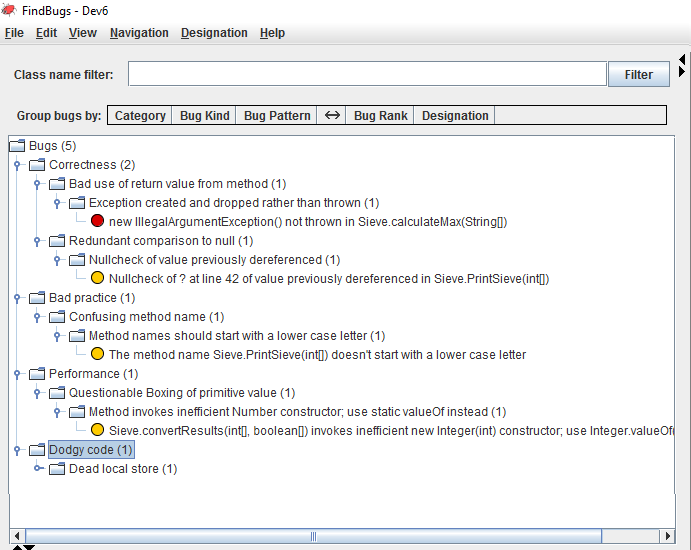
After changes were made from FindBugs, I moved one to fixing Checkstyle’s complaints, once again rerunning the module after each change to ensure it dropped off the list. There’s not really much of interest to talk about here as most of the changes were just stylistic. The two that may have been more practical were getting rid of an empty else block and updating the catch block to use the specific IllegalArgumentException instead of the generic Exception.

One thing I had a bit of trouble with was the unit tests. At first, I simply had forgotten to write any, instead just focusing on fixing the problems that FindBugs and Checkstyle found. This actually made it a bit more difficult to write the tests since I had to redownload the buggy version beforehand. In the end, I decided to write test for the convertResults() and calculateMax() functions, as in both cases the refactors could have led to the program’s results changing. In the case of convertResults(), I included three tests to make sure that the results returned did not change due to the use of Integer.valueOf() instead of creating a new Integer object. As for calculateMax(), I wrote two tests to make sure that exceptions were actually thrown on the two branches that should have thrown exceptions. All tests were passing after the refactors were made.

Code for the deliverable: github.com/mjb236/CS1632/tree/master/Deliverable%206

Screenshots

FindBugs before:



FindBugs after:



Checkstyle before:

