

Nathan Spangler

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

https://github.com/njs55/Nathan_Spangler_CS1632_QA_Deliverables/tree/master/Deliverable6

SUMMARY

FindBugs

- Exception created and dropped rather than thrown
 - Fixed this bug by adding *“throw”* in front of *“new IllegalArgumentException();”*
- Nullcheck of value previously dereferenced
 - Changed *“if (results.length == 0 || results != null) {}”* to *“if (results != null) {}”*
- The class name sieve doesn't start with an upper case letter
 - Changed the class name and file name to *“Sieve”*
- The method name sieve.PrintSieve(int[]) doesn't start with a lower case letter
 - Change *“PrintSieve”* to *“printSieve”*
- Method invokes inefficient Number constructor; use static valueOf instead (convertResults)
 - In sieve.convertResults(), I changed *“actual.add(new Integer(j + 1));”* to *“actual.add(Integer.valueOf(j + 1));”*

CheckStyle

- Local variable name 'c' must match pattern *“^[a-z][a-z0-9][a-zA-Z0-9]*\$”*. [LocalVariableName]
 - Changed *‘c’* to *‘charHolder’* in *printsieve(int[], Boolean[])* even though the method is unused.
- First sentence of Javadoc is incomplete (period is missing) or not present. [SummaryJavadoc]
 - Added a sentence with a period to the Javadoc comment before the *getTrueArray()* method
- Parameter name 's' must match pattern *“^[a-z][a-z0-9][a-zA-Z0-9]*\$”*. [ParameterName]
 - Changed *‘s’* to *‘size’* in *getTrueArray()* method
- Array brackets at illegal position. [ArrayTypeStyle]
 - Changed *“boolean toReturn[] = new boolean[size];”* to *“Boolean[] toReturn = new boolean[size];”* in the *getTrueArray()* method.
- Array brackets at illegal position. [ArrayTypeStyle]
 - Changed *“convertResults(int[] results, boolean prime[])”* to *“convertResults(int[] results, Boolean[] prime)”*
- Empty else block. [EmptyBlock]
 - Deleted an extra *else{}* statement within the *calculateMax()* method.
- Catching 'Exception' is not allowed. [IllegalCatch]
 - Removed the *TryCatch* statement in main and only called the *calculateMax()* method. Then in the *calculateMax()* method I replaced the *“throw new IllegalArgumentException();”* statements with the print statements found in the *TryCatch* that I removed as well as set the max (return value) to 100.

I first ran the FindBugs GUI and fixed all the bugs accordingly. I then wrote Unit Tests for the `calculateMax()` method because the method caused a defect visible to the user. I then finished up with fixing the issues that CheckStyle found. After completing the fixes for CheckStyle I noticed that 2 of my unit tests were now failing. The reason they were failing was because I was checking to see if the *`IllegalArgumentException()`* was thrown or not. And after fixing the CheckStyle issues I ended up deleting the code that threw these exceptions. But I then realized that my unit tests should have never been testing to see if an exception was thrown. They should have been testing to make sure the max value was set to 100. However, my concern is that I still changed the behavior of the method, even though the functionality of the code has not changed.

Code can be found:

https://github.com/njs55/Nathan_Spangler_CS1632_QA_Deliverables/tree/master/Deliverable6

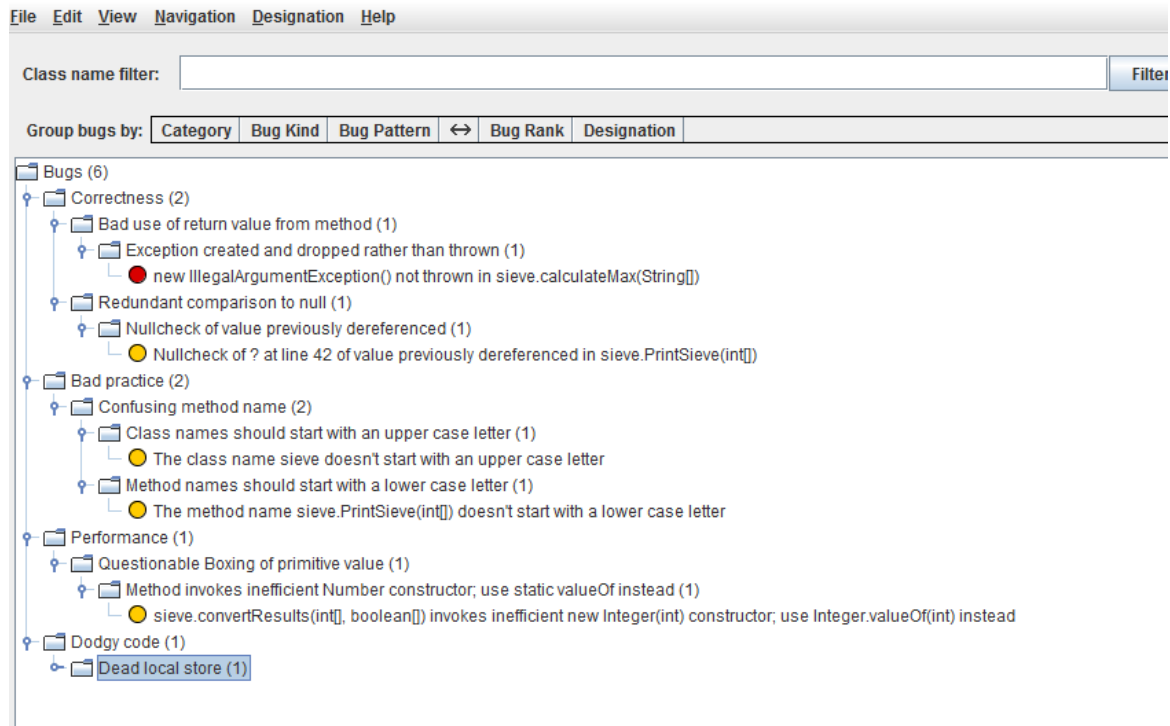
OR

<https://github.com/njs55/bug-fodder>

SCREENSHOTS

FindBugs untouched.

FindBugs - deliverable6

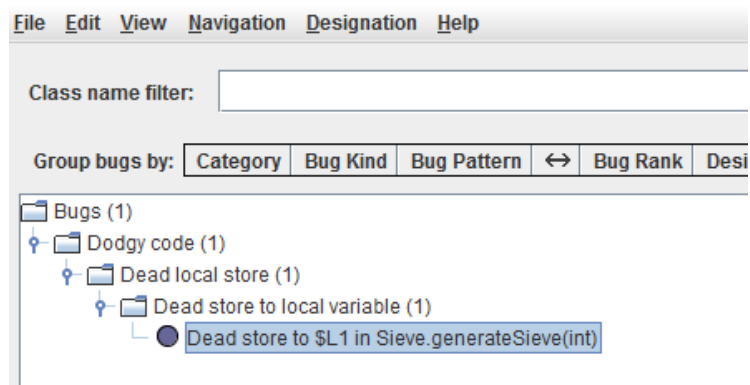


The screenshot shows the FindBugs IDE interface. At the top is a menu bar with File, Edit, View, Navigation, Designation, and Help. Below the menu is a 'Class name filter' text box with a 'Filter' button. A 'Group bugs by' section contains tabs for Category, Bug Kind, Bug Pattern, Bug Rank, and Designation. The main area displays a tree of bugs under the heading 'Bugs (6)'. The tree is expanded to show the following categories and bugs:

- Correctness (2)
 - Bad use of return value from method (1)
 - Exception created and dropped rather than thrown (1)
 - new IllegalArgumentException() not thrown in sieve.calculateMax(String[])
 - Redundant comparison to null (1)
 - Nullcheck of value previously dereferenced (1)
 - Nullcheck of ? at line 42 of value previously dereferenced in sieve.PrintSieve(int[])
- Bad practice (2)
 - Confusing method name (2)
 - Class names should start with an upper case letter (1)
 - The class name sieve doesn't start with an upper case letter
 - Method names should start with a lower case letter (1)
 - The method name sieve.PrintSieve(int[]) doesn't start with a lower case letter
- Performance (1)
 - Questionable Boxing of primitive value (1)
 - Method invokes inefficient Number constructor; use static valueOf instead (1)
 - sieve.convertResults(int[], boolean[]) invokes inefficient new Integer(int) constructor; use Integer.valueOf(int) instead
- Dodgy code (1)
 - Dead local store (1)

FindBugs after 5 Bugs were fixed.

FindBugs - deliverable6



The screenshot shows the FindBugs IDE interface after 5 bugs have been fixed. The menu bar and 'Class name filter' are the same. The 'Group bugs by' section shows the same tabs. The main area displays a tree of bugs under the heading 'Bugs (1)'. The tree is expanded to show the following categories and bugs:

- Dodgy code (1)
 - Dead local store (1)
 - Dead store to local variable (1)
 - Dead store to \$L1 in Sieve.generateSieve(int)

CheckStyle results after performing FindBugs fixes.

```
PS C:\Users\Nate\..findbugsworkspace>
PS C:\Users\Nate\..findbugsworkspace> java -jar ./checkstyle-7.0-all.jar -c ./google_checks_modified.xml ./Sieve.java
Starting audit...
[WARN] C:\Users\Nate\Documents\Classes\Summer 2016\Quality Assurance\deliverable6\findbugsworkspace\..Sieve.java:16:18: Local variable name 'c' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*$'. [LocalVariableName]
[WARN] C:\Users\Nate\Documents\Classes\Summer 2016\Quality Assurance\deliverable6\findbugsworkspace\..Sieve.java:53: First sentence of Javadoc is incomplete (period is missing) or not present. [SummaryJavadoc]
[WARN] C:\Users\Nate\Documents\Classes\Summer 2016\Quality Assurance\deliverable6\findbugsworkspace\..Sieve.java:57:46: Parameter name 's' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*$'. [ParameterName]
[WARN] C:\Users\Nate\Documents\Classes\Summer 2016\Quality Assurance\deliverable6\findbugsworkspace\..Sieve.java:58:25: Array brackets at illegal position. [ArrayTypeStyle]
[WARN] C:\Users\Nate\Documents\Classes\Summer 2016\Quality Assurance\deliverable6\findbugsworkspace\..Sieve.java:75:68: Array brackets at illegal position. [ArrayTypeStyle]
[WARN] C:\Users\Nate\Documents\Classes\Summer 2016\Quality Assurance\deliverable6\findbugsworkspace\..Sieve.java:191:20: Empty else block. [EmptyBlock]
[WARN] C:\Users\Nate\Documents\Classes\Summer 2016\Quality Assurance\deliverable6\findbugsworkspace\..Sieve.java:224:11: Catching 'Exception' is not allowed. [IllegalCatch]
Audit done.
```

CheckStyle results after fixing all issues CheckStyle found above.

```
PS C:\Users\Nate\..findbugsworkspace> java -jar ./checkstyle-7.0-all.jar -c ./google_checks_modified.xml ./Sieve.java
Starting audit...
Audit done.
```