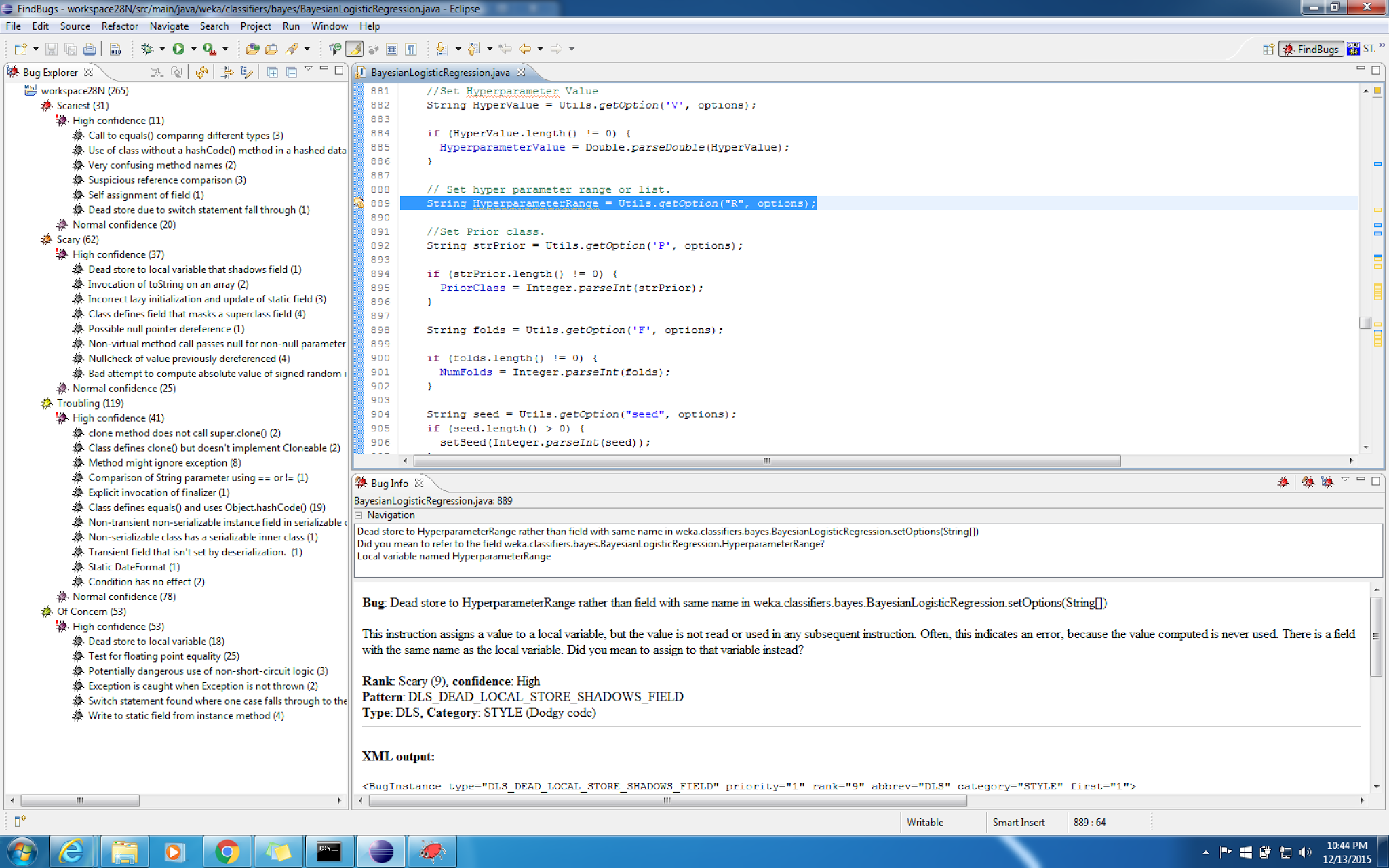
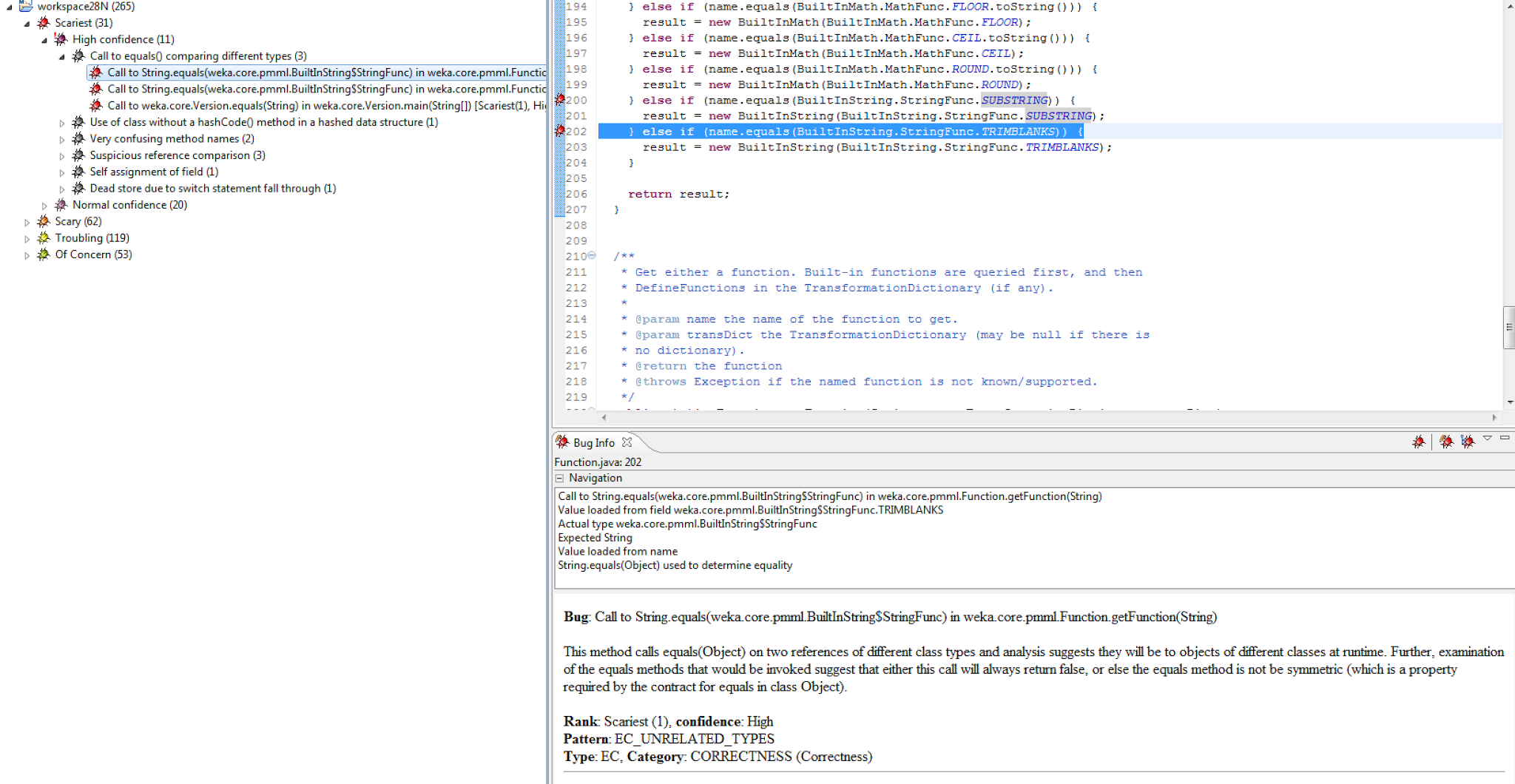
**FindBugs Analysis Report**



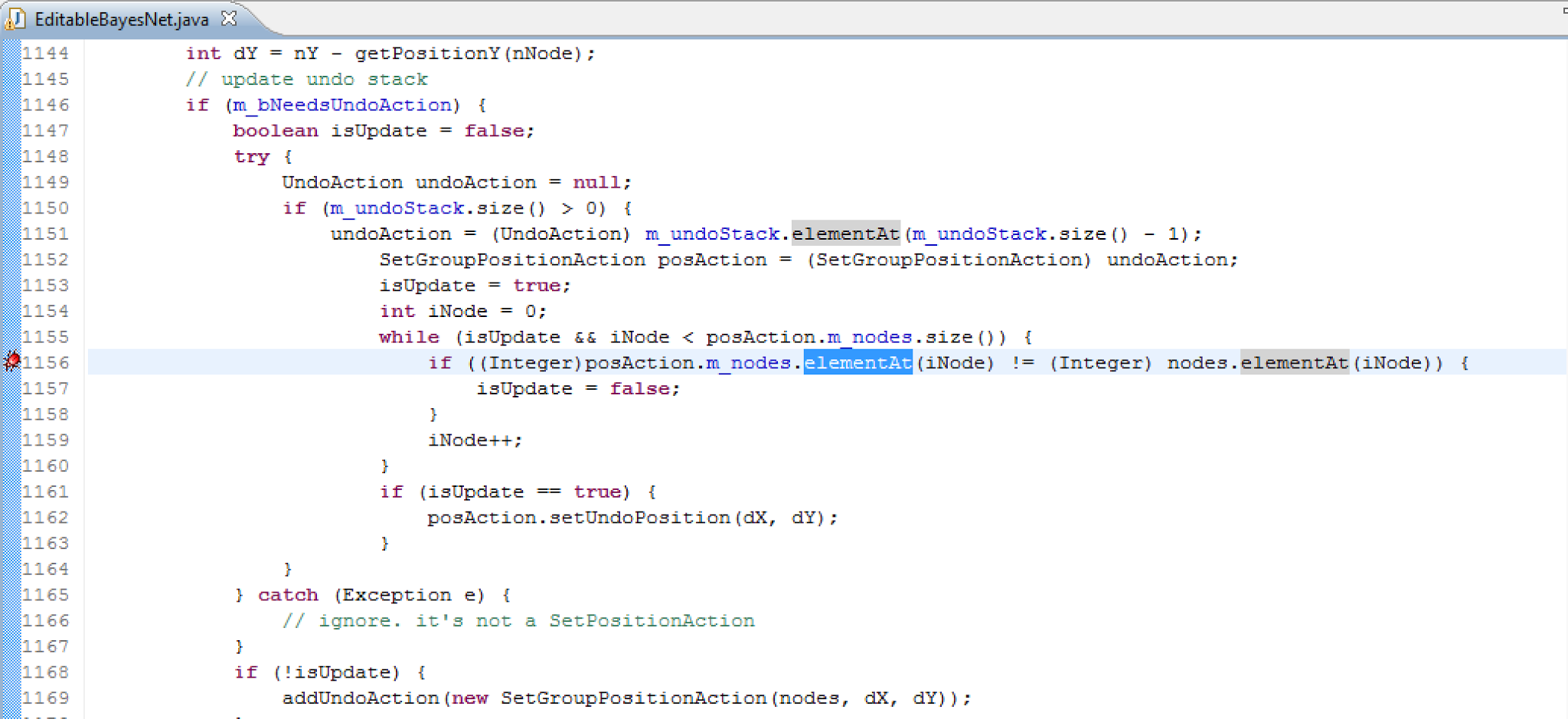
FindBugs reports a lot of warnings across the project with descending order of severity - Scariest, Scary, Troubling and Of Concern. Highest number of bugs reported are in Troubling section. Each is divided into High Confidence and Normal Confidence.

*Scariest High Confidence warnings:*

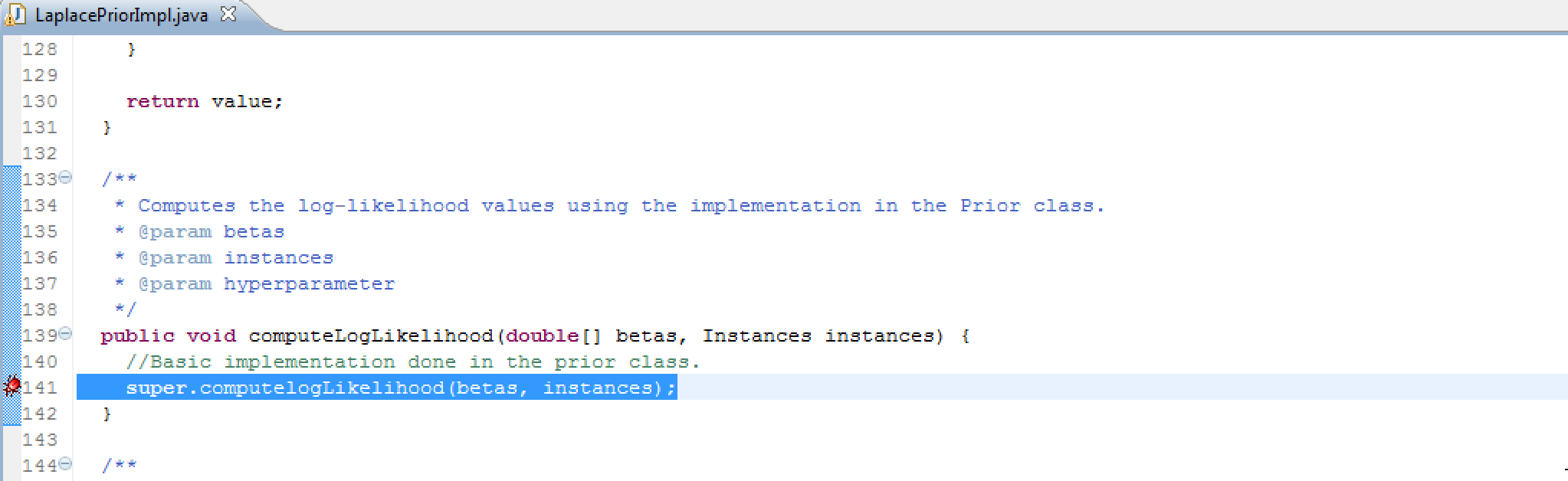
String equality comparison warning shown is not correct as it is comparing two same return types.



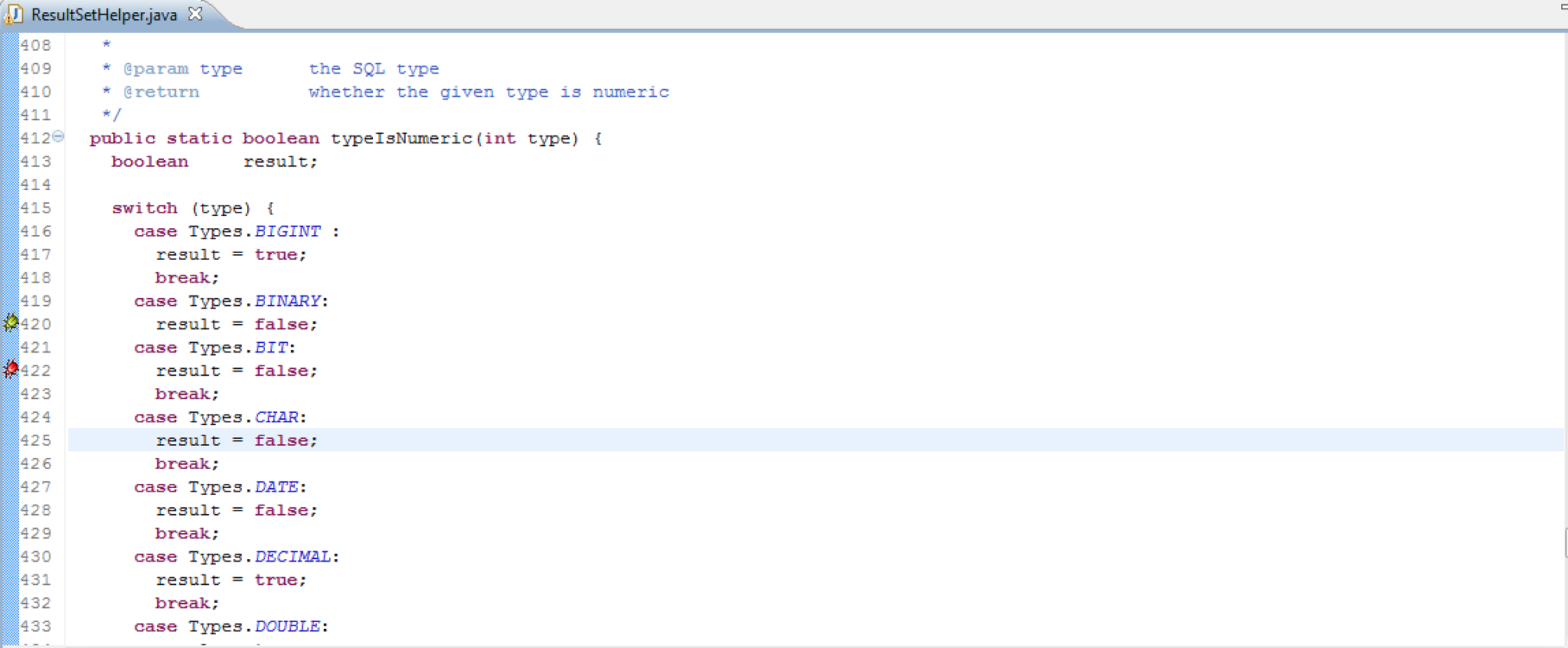
Methods names doesn’t explain the functionality of the method and most of them are confusing.



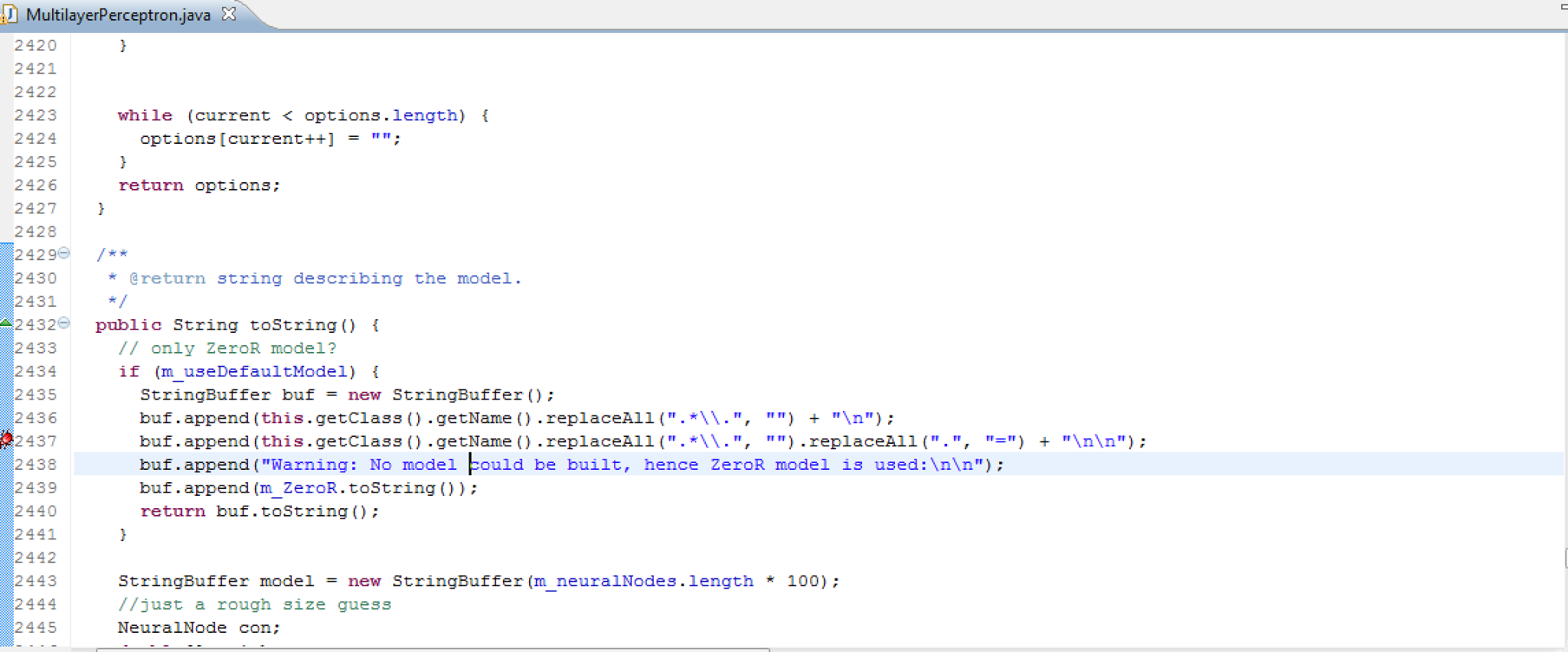
Object comparison is not correctly done, == operator is used instead of equals() method for the comparison.



Switch case condition has a missing break condition which will lead to unexpected results.

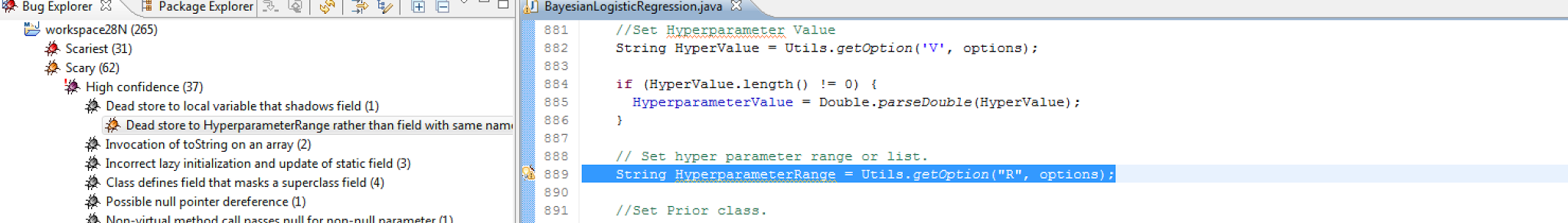


Suggested warning displayed for the regular expression usage is good but not valid in the scenario.

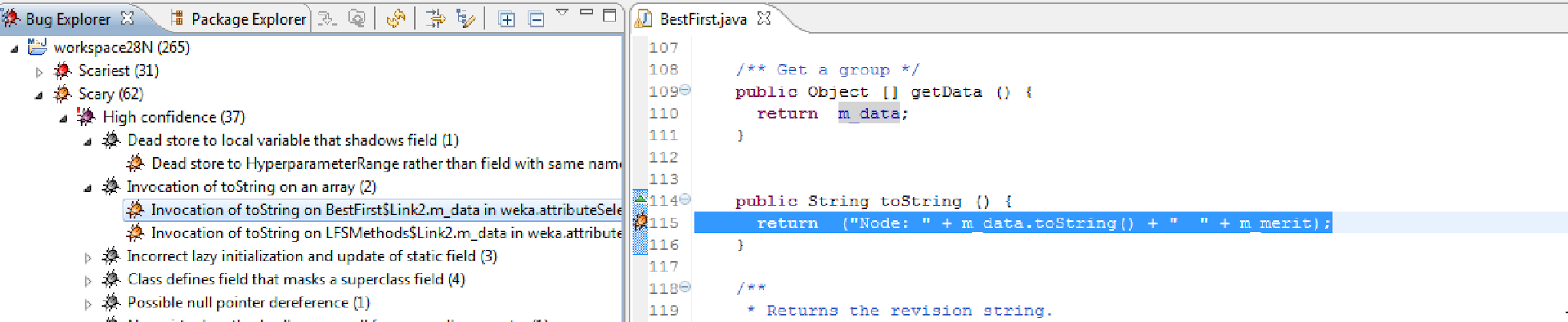


*Scary High Confidence Warnings:*

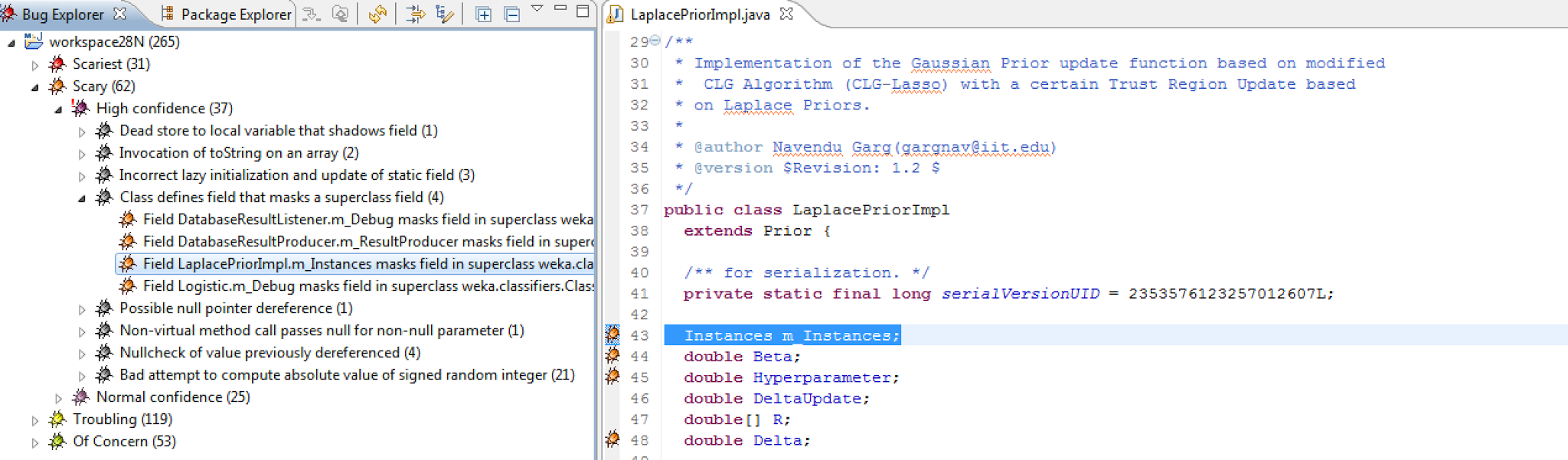
Local variables are defined but not used which can be safely removed



Wrong conversion of an Array to String, Array.toString should be used instead of just toString() on array object.



Instance with same type and name is redeclared in the child class which is misleading and can be safely removed.

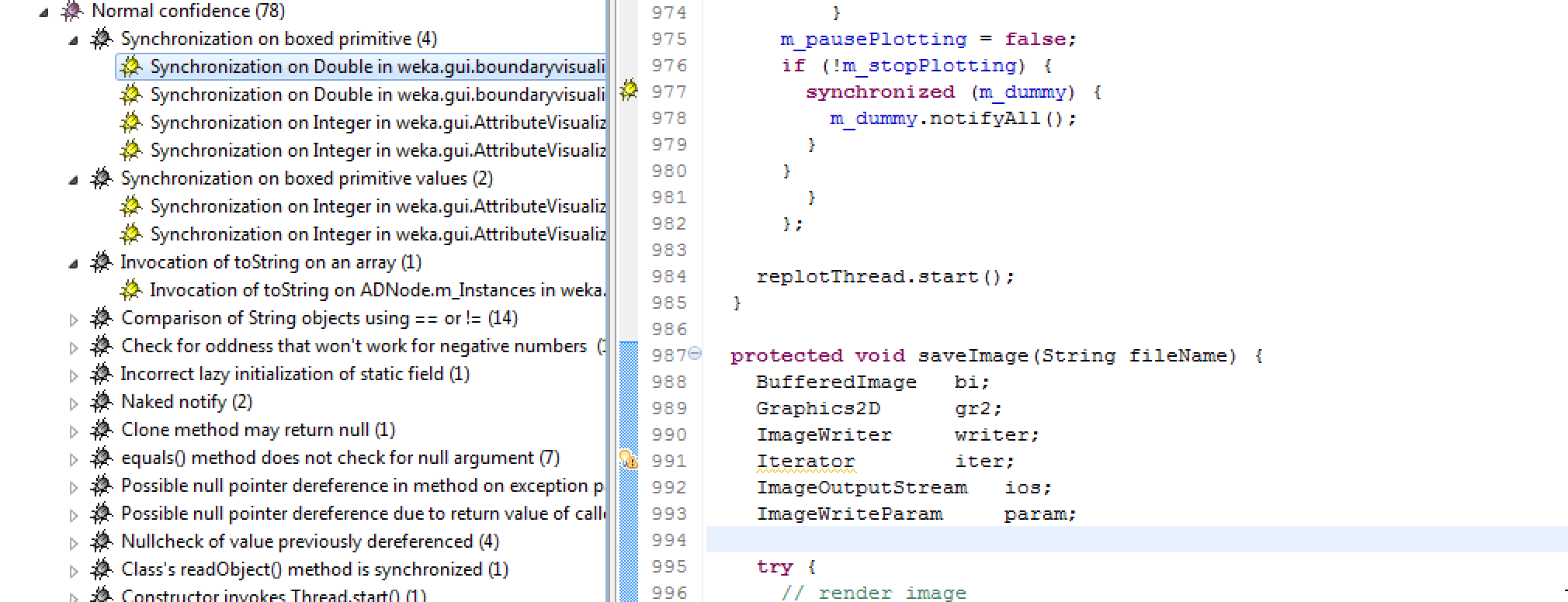


Other detected bugs are -

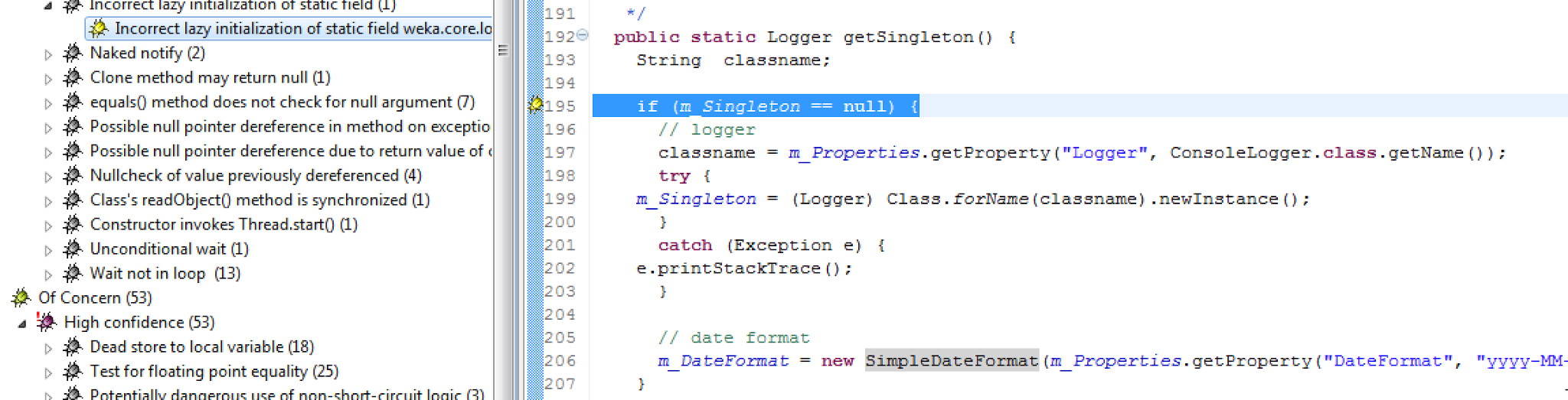
null dereference in method for which the null conditional check is already handled

*Troubling High Confidence Warnings:*

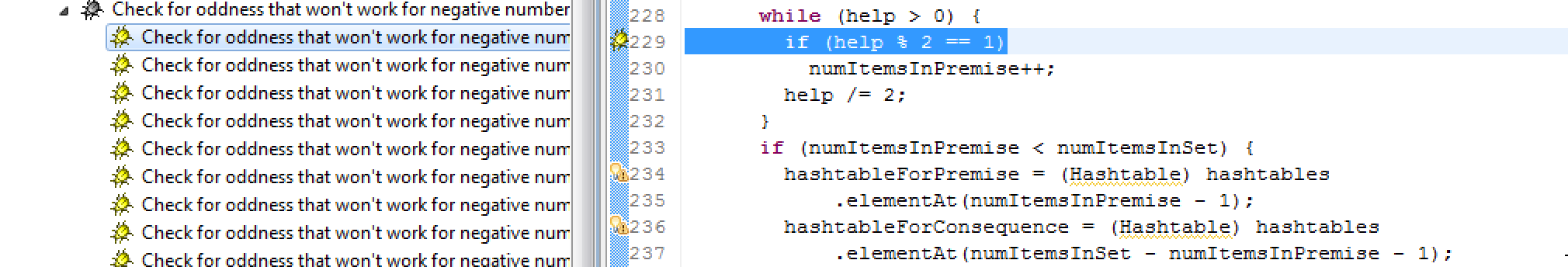
Synchronizing on primitive data types with other objects may lead to unresponsiveness and deadlock conditions in the code



Incorrect singleton initialization is done for the logger which may lead to incorrect logging.



Other bugs detected are not valid as FindBugs doesn’t evaluate the scenarios such as for checking whether the number is odd, initially the number is checked if the number is positive but findbugs fails to check this and shows that negative numbers may not be handled correctly



Most of the other bugs detected are repeated again for different data type and object types at different severity.