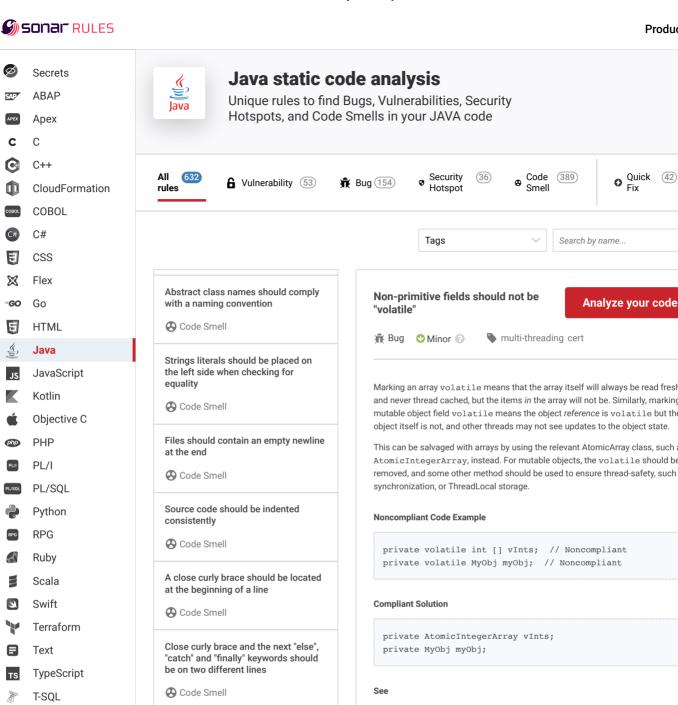


Products >

(42)

O Quick Fix



Marking an array volatile means that the array itself will always be read fresh and never thread cached, but the items in the array will not be. Similarly, marking a mutable object field volatile means the object reference is volatile but the object itself is not, and other threads may not see updates to the object state. This can be salvaged with arrays by using the relevant AtomicArray class, such as AtomicIntegerArray, instead. For mutable objects, the volatile should be removed, and some other method should be used to ensure thread-safety, such as private volatile int [] vInts; // Noncompliant private volatile MyObj myObj; // Noncompliant • CERT, CON50-J. - Do not assume that declaring a reference volatile guarantees safe publication of the members of the referenced object Available In: sonarlint ⊕ | sonarcloud ↔ | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. Privacy Policy

Tabulation characters should not be used

Close curly brace and the next "else", "catch" and "finally" keywords should

An open curly brace should be located

An open curly brace should be located

be located on the same line

at the beginning of a line

Code Smell

Code Smell

at the end of a line Code Smell

Code Smell

Functions should not be defined with a variable number of arguments

Code Smell

VB.NET

VB6

XML

Local-Variable Type Inference should be used

Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

Code Smell

Track uses of disallowed classes

Code Smell

Track uses of "@SuppressWarnings" annotations

Code Smell