

Secrets

ABAP

Apex

C

C++

CloudFormation

COBOL

C#

CSS

Flex

Go

HTML

Java

JavaScript

Kotlin

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML

# Java static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

All rules632

Vulnerability53

Bug154

Security Hotspot36

Code Smell389

Quick Fix42

Tags ▾

Search by name... 🔍

Bug

Double Brace Initialization should not be used

Bug

Non-primitive fields should not be "volatile"

Bug

"toArray" should be passed an array of the proper type

Bug

Neither "Math.abs" nor negation should be used on numbers that could be "MIN\_VALUE"

Bug

The value returned from a stream read should be checked

Bug

"@NonNull" values should not be set to null

Bug

"Iterator.next()" methods should throw "NoSuchElementException"

Bug

"compareTo" results should not be checked for specific values

Bug

Math operands should be cast before assignment

Bug

Ints and longs should not be shifted by zero or more than their number of bits-1

Bug

"compareTo" should not return "Integer.MIN\_VALUE"

## A "for" loop update clause should move the counter in the right direction

Analyze your code

Bug

Major

cert

A `for` loop with a counter that moves in the wrong direction is not an infinite loop. Because of wraparound, the loop will eventually reach its stop condition, but in doing so, it will run many, many more times than anticipated, potentially causing unexpected behavior.

### Noncompliant Code Example

```
public void doSomething(String [] strings) {
    for (int i = 0; i < strings.length; i--) { // Noncompliant
        String string = strings[i]; // ArrayIndexOutOfBoundsException
        //...
    }
}
```

### Compliant Solution

```
public void doSomething(String [] strings) {
    for (int i = 0; i < strings.length; i++) {
        String string = strings[i];
        //...
    }
}
```

See

- [CERT, MSC54-J](#) - Avoid inadvertent wrapping of loop counters

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](#)

https://rules.sonarsource.com/java/RSPEC-2251

1/2

 Bug
<div>Boxing and unboxing should not be immediately reversed</div> <div> Bug</div>
<div>"equals(Object obj)" should test argument type</div> <div> Bug</div>
<div>"Serializable" inner classes of non-serializable classes should be "static"</div> <div> Bug</div>
<div>The non-serializable super class of a "Serializable" class should have a no-argument constructor</div>