

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... 🔍

Public methods should not contain selector arguments

Code Smell

Java parser failure

Code Smell

Track uses of disallowed methods

Code Smell

Types should be used in lambdas

Code Smell

"java.time" classes should be used for dates and times

Code Smell

The names of methods with boolean return values should start with "is" or "has"

Code Smell

Files should contain only one top-level class or interface each

Code Smell

Classes should not have too many fields

Code Smell

The ternary operator should not be used

Code Smell

Standard functional interfaces should not be redefined

Code Smell

"NullPointerException" should not be caught

Code Smell

"NullPointerException" should not be explicitly thrown

Code Smell

"readObject" should not be "synchronized"

Analyze your code

Code Smell

Major

confusing

A readObject method is written when a Serializable object needs special handling to be rehydrated from file. It should be the case that the object being created by readObject is only visible to the thread that invoked the method, and the synchronized keyword is not needed, and using synchronized anyway is just confusing. If this is not the case, the method should be refactored to make it the case.

Noncompliant Code Example

```
private synchronized void readObject(java.io.ObjectInputStream
    throws IOException, ClassNotFoundException { // Noncomp
    //...
}
```






Compliant Solution

```
private void readObject(java.io.ObjectInputStream in)
    throws IOException, ClassNotFoundException { // Complia
    //...
}
```

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](#)

 Code Smell
<div>Classes should not have too many methods</div> <div> Code Smell</div>
<div>Methods should not have too many lines</div> <div> Code Smell</div>
<div>Track uses of "NOSONAR" comments</div> <div> Code Smell</div>
<div>Classes and enums with private members should have a constructor</div> <div> Code Smell</div>