 **sonar** RULES

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... 🔍

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

Classes should not have too many methods

Code Smell

Methods should not have too many lines

Code Smell

Track uses of "NOSONAR" comments

Code Smell

Classes and enums with private members should have a constructor

Code Smell

Track comments matching a regular expression

Code Smell

Statements should be on separate lines

Code Smell

Classes should not be coupled to too many other classes (Single

"Lock" objects should not be "synchronized"

Analyze your code

Code Smell

Major ?

multi-threading cert clumsy

java.util.concurrent.locks.Lock offers far more powerful and flexible locking operations than are available with synchronized blocks. So synchronizing on a Lock throws away the power of the object, and is just silly. Instead, such objects should be locked and unlocked using tryLock() and unlock().

Noncompliant Code Example

```
Lock lock = new MyLockImpl();
synchronized(lock) { // Noncompliant
    //...
}
```

Compliant Solution

```
Lock lock = new MyLockImpl();
lock.tryLock();
//...
```

See

- [CERT, LCK03-J](#). - Do not synchronize on the intrinsic locks of high-level concurrency objects

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-2442

1/2

Responsibility Principle)  Code Smell
"java.lang.Error" should not be extended  Code Smell
Anonymous classes should not have too many lines  Code Smell
Public types, methods and fields (API) should be documented with Javadoc  Code Smell
Exception handlers should preserve