

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

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... 🔍

Magic numbers should not be used

Code Smell

Files should not have too many lines of code

Code Smell

Lines should not be too long

Code Smell

JEE applications should not "getClassLoader"

Bug

Math should not be performed on floats

Bug

"equals" methods should be symmetric and work for subclasses

Bug

Literal suffixes should be upper case

Code Smell

Unicode-aware versions of character classes should be preferred

Code Smell

Use Java 12 "switch" expression

Code Smell

"serialVersionUID" should not be declared blindly

Code Smell

"Stream.collect()" calls should not be redundant

Code Smell

Local constants should follow naming conventions for constants

Code Smell

Objects should not be created only to "getClass"

Analyze your code

Code Smell

Major ?

performance

Creating an object for the sole purpose of calling getClass on it is a waste of memory and cycles. Instead, simply use the class' .class property.

Noncompliant Code Example

MyObject myOb = new MyObject(); // Noncompliant
Class c = myOb.getClass();

Compliant Solution

Class c = MyObject.class;

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-2133

1/2

Unit tests should throw exceptions  Code Smell
Test methods should comply with a naming convention  Code Smell
Value-based objects should not be serialized  Code Smell
Default annotation parameter values should not be passed as arguments  Code Smell
Method parameters should be