

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... 🔍

Local variables should not be declared and then immediately returned or thrown

Code Smell

Unused local variables should be removed

Code Smell

Private fields only used as local variables in methods should become local variables

Code Smell

"public static" fields should be constant

Code Smell

Loops should not contain more than a single "break" or "continue" statement

Code Smell

Declarations should use Java collection interfaces such as "List" rather than specific implementation classes such as "LinkedList"

Code Smell

"switch" statements should have at least 3 "case" clauses

Code Smell

A "while" loop should be used instead of a "for" loop

Code Smell

The default unnamed package should not be used

Code Smell

"equals(Object obj)" should be overridden along with the "compareTo(T obj)" method

Code Smell

Package names should comply with a naming convention

Deprecated annotations should include explanations

Analyze your code

Code Smell

Major ?

java9

Since Java 9, @Deprecated has two additional arguments to the annotation:

- since allows you to describe when the deprecation took place
- forRemoval, indicates whether the deprecated element will be removed at some future date

In order to ease the maintainers work, it is recommended to always add one or both of these arguments.

This rule reports an issue when @Deprecated is used without any argument.

Noncompliant Code Example

@Deprecated

Compliant Solution

@Deprecated(since="4.2", forRemoval=true)

Exceptions

The members and methods of a deprecated class or interface are ignored by this rule. The classes and interfaces themselves are still subject to it.

See Also

- {rule:java:S1123}

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-6355

1/2

 Code Smell
Nested code blocks should not be used  Code Smell
Array designators "[]" should be on the type, not the variable  Code Smell
Array designators "[]" should be located after the type in method signatures  Code Smell
Type parameter names should comply