

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

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... 🔍

Abstract class names should comply with a naming convention

Strings literals should be placed on the left side when checking for equality

Files should contain an empty newline at the end

Source code should be indented consistently

A close curly brace should be located at the beginning of a line

Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line

An open curly brace should be located at the beginning of a line

An open curly brace should be located at the end of a line

Tabulation characters should not be used

Functions should not be defined with a variable number of arguments

Static non-final field names should comply with a naming convention

Analyze your code

Code Smell Minor convention

Shared naming conventions allow teams to collaborate efficiently. This rule checks that static non-final field names match a provided regular expression.

Noncompliant Code Example

With the default regular expression `^[a-z][a-zA-Z0-9]*$`:

```
public final class MyClass {
    private static String foo_bar;
}
```

Compliant Solution

```
class MyClass {
    private static String fooBar;
}
```

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-3008

1/2

<div>Local-Variable Type Inference should be used</div> <div> Code Smell</div>
<div>Migrate your tests from JUnit4 to the new JUnit5 annotations</div> <div> Code Smell</div>
<div>Track uses of disallowed classes</div> <div> Code Smell</div>
<div>Track uses of "@SuppressWarnings" annotations</div> <div> Code Smell</div>