sonar RULES

Products

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 rules 632 | Vulnerability 53 | Bug 154 | Security Hotspot 36 | Code Smell 389 | Quick Fix 42

Tags

Search by name...

Code Smell

Extensions and implementations should not be redundant
Code Smell

"==" and "!=" should not be used when "equals" is overridden
Code Smell

An abstract class should have both abstract and concrete methods
Code Smell

Sets with elements that are enum values should be replaced with EnumSet
Code Smell

String operations should not rely on the default system locale
Code Smell

Comments should not be located at the end of lines of code
Code Smell

Track uses of "CHECKSTYLE:OFF" suppression comments
Code Smell

Loggers should be "private static final" and should share a naming convention
Code Smell

Track uses of "NOPMD" suppression comments
Code Smell

Packages should have a javadoc file 'package-info.java'
Code Smell

The members of an interface or class declaration should appear in a pre-defined order

## "@Override" should be used on overriding and implementing methods

Analyze your code

Code Smell Major Quick Fix bad-practice

Using the `@Override` annotation is useful for two reasons :

- It elicits a warning from the compiler if the annotated method doesn't actually override anything, as in the case of a misspelling.
- It improves the readability of the source code by making it obvious that methods are overridden.

### Noncompliant Code Example

```
class ParentClass {
  public boolean doSomething(){...}
}
class FirstChildClass extends ParentClass {
  public boolean doSomething(){...}  // Noncompliant
}
```

### Compliant Solution

```
class ParentClass {
  public boolean doSomething(){...}
}
class FirstChildClass extends ParentClass {
  @Override
  public boolean doSomething(){...}  // Compliant
}
```

### Exceptions

This rule is relaxed when overriding a method from the `Object` class like `toString()`, `hashCode()`, ...

Available In:

sonarlint | sonarcloud | sonarqube

Code Smell

Abstract class names should comply with a naming convention

Code Smell

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

Code Smell

Files should contain an empty newline at the end

Code Smell

Source code should be indented