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...

Security Hotspot

Setting loose POSIX file permissions is security-sensitive
Security Hotspot

Formatting SQL queries is security-sensitive
Security Hotspot

Deprecated annotations should include explanations
Code Smell

Restricted Identifiers should not be used as Identifiers
Code Smell

Redundant constructors/methods should be avoided in records
Code Smell

Records should be used instead of ordinary classes when representing immutable data structure
Code Smell

"Stream.toList()" method should be used instead of "collectors" when unmodifiable list needed
Code Smell

Operator "instanceof" should be used instead of "A.class.isInstance()"
Code Smell

String multiline concatenation should be replaced with Text Blocks
Code Smell

Single-character alternations in regular expressions should be replaced with character classes
Code Smell

Reluctant quantifiers in regular

The Object.finalize() method should not be overridden

Analyze your code

Code Smell | Critical | unpredictable cert

The `Object.finalize()` method is called on an object by the garbage collector when it determines that there are no more references to the object. But there is absolutely no warranty that this method will be called AS SOON AS the last references to the object are removed. It can be few microseconds to few minutes later. So when system resources need to be disposed by an object, it's better to not rely on this asynchronous mechanism to dispose them.

Noncompliant Code Example

```
public class MyClass {
  ...
  protected void finalize() {
    releaseSomeResources();    // Noncompliant
  }
  ...
}
```

See

- CERT, MET12-J. - Do not use finalizers

Available In:

sonarlint | sonarcloud | sonarqube

expressions should be followed by an expression that can't match the empty string

Code Smell

Constructors of an "abstract" class should not be declared "public"

Code Smell

Similar tests should be grouped in a single Parameterized test

Code Smell

Tests should be stable

Code Smell