




 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... 🔍

comply with a naming convention

Code Smell

JUnit rules should be used

Code Smell

Nested "enum"s should not be declared static

Code Smell

"catch" clauses should do more than rethrow

Code Smell

Mutable fields should not be "public static"

Code Smell

The diamond operator ("<>") should be used

Code Smell

"finalize" should not set fields to "null"

Code Smell

Subclasses that add fields should override "equals"

Code Smell

Catches should be combined

Code Smell

Methods of "Random" that return floating point values should not be used in random integer generation

Code Smell

Parsing should be used to convert "Strings" to primitives

Code Smell

Classes should not be empty

Code Smell

The Object.finalize() method should not be called

Analyze your code

Bug

Major

cwe cert

According to the official javadoc documentation, this Object.finalize() is called by the garbage collector on an object when garbage collection determines that there are no more references to the object. Calling this method explicitly breaks this contract and so is misleading.

Noncompliant Code Example

```
public void dispose() throws Throwable {
    this.finalize();
} // Noncompliant
```

See

- MITRE, CWE-586 - Explicit Call to Finalize()
- CERT, MET12-J. - Do not use finalizers

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-1111

1/2

<div>Fields in non-serializable classes should not be "transient"</div> <div> Code Smell</div>
<div>Boolean checks should not be inverted</div> <div> Code Smell</div>
<div>Redundant casts should not be used</div> <div> Code Smell</div>
<div>"@Deprecated" code should not be used</div> <div> Code Smell</div>
<div>"toString()" should never be called on</div>