sonar RULES

Products ⌄

Java static code analysis
Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

Sidebar	
🚫	Secrets
SAP	ABAP
APEX	Apex
C	C
C++	C++
C#	CloudFormation
COBOL	COBOL
C#	C#
CSS	CSS
Flex	Flex
GO	Go
HTML	HTML
Java	**Java**
JS	JavaScript
Kotlin	Kotlin
Objective C	Objective C
php	PHP
PL/I	PL/I
PL/SQL	PL/SQL
Python	Python
RPG	RPG
Ruby	Ruby
Scala	Scala
Swift	Swift
Terraform	Terraform
Text	Text
TS	TypeScript
T-SQL	T-SQL
VB	VB.NET
VB6	VB6
XML	XML

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

Tags ⌄   Search by name... 🔍

🐞 **Bug**

Reflection should not be used to increase accessibility of records' fields
🐞 Bug

AssertJ assertions with "Consumer" arguments should contain assertion inside consumers
🐞 Bug

**The regex escape sequence \cX should only be used with characters in the @-_ range**
🐞 Bug

Regular expressions should not overflow the stack
🐞 Bug

**Tests method should not be annotated with competing annotations**
🐞 Bug

Assertions should not be used in production code
🐞 Bug

**DateTimeFormatters should not use mismatched year and week numbers**
🐞 Bug

Unicode Grapheme Clusters should be avoided inside regex character classes
🐞 Bug

Case insensitive Unicode regular expressions should enable the "UNICODE_CASE" flag
🐞 Bug

Assertions should not compare an object to itself
🐞 Bug

"super.finalize()" should be called at the end of "Object.finalize()" implementations

Analyze your code

🐞 Bug   🔴 Critical ❓   🏷 cwe cert

Overriding the `Object.finalize()` method must be done with caution to dispose some system resources.

Calling the `super.finalize()` at the end of this method implementation is highly recommended in case parent implementations must also dispose some system resources.

Noncompliant Code Example

```
protected void finalize() {    // Noncompliant; no call to su
  releaseSomeResources();
}

protected void finalize() {
  super.finalize();  // Noncompliant; this call should come
  releaseSomeResources();
}
```

Compliant Solution

```
protected void finalize() {
  releaseSomeResources();
  super.finalize();
}
```

See

- MITRE, CWE-568 - finalize() Method Without super.finalize()
- CERT, MET12-J. - Do not use finalizers

Available In:

sonarlint ⊖ | **sonar**cloud ♻ | **sonar**qube ⌇

5/26/22, 8:08 PM   Java static code analysis: "super.finalize()" should be called at the end of "Object.finalize()" implementations

2/2

Regex alternatives should not be redundant

🐞 Bug

Alternatives in regular expressions should be grouped when used with anchors

🐞 Bug

AssertJ methods setting the assertion context should come before an assertion

🐞 Bug

AssertJ configuration should be applied