

































-  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... 🔍


 Vulnerability


Server-side requests should not be vulnerable to forging attacks




 Bug


Collections should not be modified while they are iterated




 Bug


Equals method should be overridden in records containing array fields




 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



## The signature of "finalize()" should match that of "Object.finalize()"

Analyze your code

 Bug  Critical   pitfall

Object.finalize() is called by the Garbage Collector at some point after the object becomes unreferenced.

In general, overloading Object.finalize() is a bad idea because:

- The overload may not be called by the Garbage Collector.
- Users are not expected to call Object.finalize() and will get confused.

But beyond that it's a terrible idea to name a method "finalize" if it doesn't actually override Object.finalize().




### Noncompliant Code Example

```
public int finalize(int someParameter) {           // Noncompliant
    /* ... */
}
```

### Compliant Solution

```
public int someBetterName(int someParameter) { // Compliant
    /* ... */
}
```

Available In:





 |  | 

© 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-1175

1/2

<div>Unicode Grapheme Clusters should be avoided inside regex character classes</div> <div> Bug</div>
<div>Case insensitive Unicode regular expressions should enable the "UNICODE_CASE" flag</div> <div> Bug</div>
<div>Assertions should not compare an object to itself</div> <div> Bug</div>
<div>Regex alternatives should not be redundant</div> <div> Bug</div>