




 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... 🔍


 Bug


Unary prefix operators should not be repeated







"=" should not be used instead of "++"







"read" and "readLine" return values should be used







Inappropriate regular expressions should not be used







Conditionally executed code should be reachable





"notifyAll" should be used





Blocks should be synchronized on "private final" fields

Non-serializable objects should not be stored in "HttpSession" objects


"wait", "notify" and "notifyAll" should only be called when a lock is obviously held on an object


Null pointers should not be dereferenced


Loop conditions should be true at least once

"default" clauses should be last

Analyze your code

 Code Smell

 Critical



switch can contain a default clause for various reasons: to handle unexpected values, to show that all the cases were properly considered.

For readability purpose, to help a developer to quickly find the default behavior of a switch statement, it is recommended to put the default clause at the end of the switch statement. This rule raises an issue if the default clause is not the last one of the switch's cases.




Noncompliant Code Example

```
switch (param) {
  case 0:
    doSomething();
    break;
  default: // default clause should be the last one
    error();
    break;
  case 1:
    doSomethingElse();
    break;
}
```

Compliant Solution

```
switch (param) {
  case 0:
    doSomething();
    break;
  case 1:
    doSomethingElse();
    break;
  default:
    error();
    break;
}
```

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-4524

1/2

<div>A "for" loop update clause should move the counter in the right direction</div> <div> Bug</div>
<div>Non-public methods should not be "@Transactional"</div> <div> Bug</div>
<div>Servlets should not have mutable instance fields</div> <div> Bug</div>
<div>"toString()" and "clone()" methods should not return null</div> <div> Bug</div>