

































-  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... 🔍

 Code Smell


Classes without "public" constructors should be "final"




Unnecessary semicolons should be omitted




Literal boolean values and nulls should not be used in assertions




Test assertions should include messages




Mutable members should not be stored or returned directly




Redundant modifiers should not be used




"private" and "final" methods that don't access instance data should be "static"




Files should not be empty




Collection methods with O(n) performance should be used carefully




"Exception" should not be caught when not required by called methods





"collect" should be used with "Streams" instead of "list::add"



"switch" statements should not have too many "case" clauses

 Code Smell

 Major ?




 brain-overload

When `switch` statements have large sets of `case` clauses, it is usually an attempt to map two sets of data. A real map structure would be more readable and maintainable, and should be used instead.





Exceptions

This rule ignores `switches` over `Enums` and empty, fall-through cases.

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](#)

<div>Switches should be used for sequences of simple "String" tests</div> <div> Code Smell</div>
<div>"final" classes should not have "protected" members</div> <div> Code Smell</div>
<div>Underscores should be used to make large numbers readable</div> <div> Code Smell</div>
<div>"Serializable" inner classes of "Serializable" classes should be static</div> <div> Code Smell</div>