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...

"URL.hashCode" and "URL.equals" should be avoided
Code Smell

Two branches in a conditional structure should not have exactly the same implementation
Code Smell

Unused assignments should be removed
Code Smell

"Object.wait(...)" should never be called on objects that implement "java.util.concurrent.locks.Condition"
Code Smell

A field should not duplicate the name of its containing class
Code Smell

JUnit4 @Ignored and JUnit5 @Disabled annotations should be used to disable tests and should provide a rationale
Code Smell

Anonymous inner classes containing only one method should become lambdas
Code Smell

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

"for" loop stop conditions should be invariant
Code Smell

Sections of code should not be commented out
Code Smell

Non-constructor methods should not have the same name as the enclosing class

All branches in a conditional structure should not have exactly the same implementation

Analyze your code

Bug Major

Having all branches in a `switch` or `if` chain with the same implementation is an error. Either a copy-paste error was made and something different should be executed, or there shouldn't be a `switch/if` chain at all.

Noncompliant Code Example

```
if (b == 0) {  // Noncompliant
  doOneMoreThing();
} else {
  doOneMoreThing();
}

int b = a > 12 ? 4 : 4;  // Noncompliant

switch (i) {  // Noncompliant
  case 1:
    doSomething();
    break;
  case 2:
    doSomething();
    break;
  case 3:
    doSomething();
    break;
  default:
    doSomething();
}
```

Exceptions

This rule does not apply to `if` chains without `else`-s, or to `switch`-es without `default` clauses.

```
if(b == 0) {    //no issue, this could have been done on pur
  doSomething();
} else if(b == 1) {
  doSomething();
}
```

Available In:

sonarlint sonarcloud sonarqube

Code Smell

Exception types should not be tested using "instanceof" in catch blocks

Code Smell

Classes from "sun.*" packages should not be used

Code Smell

Throwable and Error should not be caught

Code Smell

Unused method parameters should be removed