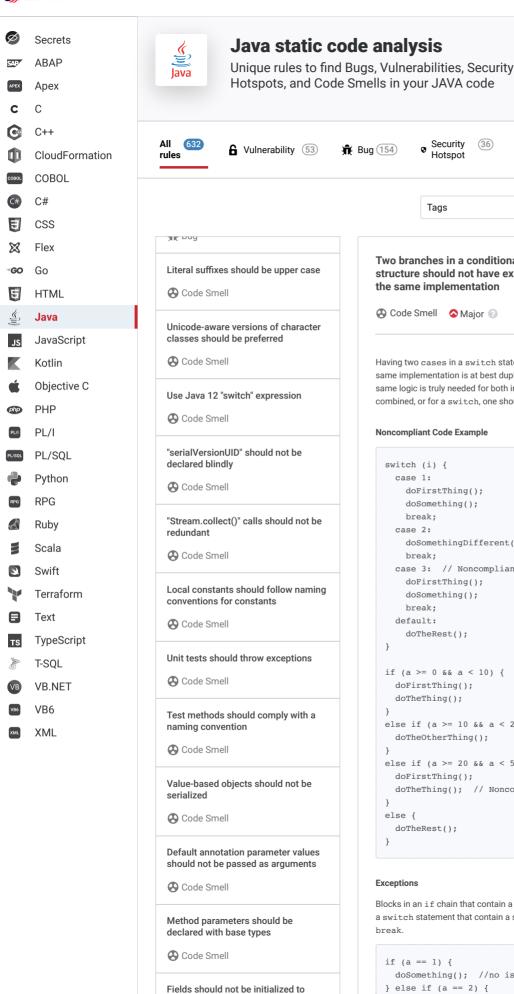


Products >



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

Security

Hotspot

Tags

(36)

Analyze your code

Quick 42 Fix

Having two cases in a switch statement or two branches in an if chain with the same implementation is at best duplicate code, and at worst a coding error. If the same logic is truly needed for both instances, then in an \mathtt{if} chain they should be combined, or for a switch, one should fall through to the other.

(389)

design suspicious

Search by name.

Noncompliant Code Example

```
switch (i) {
  case 1:
   doFirstThing();
    doSomething();
   break;
  case 2:
    doSomethingDifferent();
  case 3: // Noncompliant; duplicates case 1's implementati
    doFirstThing();
    doSomething();
    break:
  default:
    doTheRest();
if (a >= 0 && a < 10) {
  doFirstThing():
  doTheThing();
else if (a >= 10 \&\& a < 20) {
  doTheOtherThing();
else if (a >= 20 \&\& a < 50) {
  doFirstThing();
  doTheThing(); // Noncompliant; duplicates first condition
else {
  doTheRest();
```

Exceptions

Blocks in an if chain that contain a single line of code are ignored, as are blocks in a switch statement that contain a single line of code with or without a following break

```
if (a == 1) {
 doSomething(); //no issue, usually this is done on purpos
} else if (a == 2) {
 doSomethingElse();
 else {
```

default values

Java static code analysis: Two branches in a conditional structure should not have exactly the same implementation

Code Smell

Multiple loops over the same set should be combined

Code Smell

Classes without "public" constructors should be "final"

Code Smell

Unnecessary semicolons should be omitted

Code Smell

Literal boolean values and nulls should not be used in assertions

```
doSomething();
}
```

But this exception does not apply to if chains without else-s, or to switch-es without default clauses when all branches have the same single line of code. In case of if chains with else-s, or of switch-es with default clauses, rule {rule:java:S3923} raises a buq.

```
if (a == 1) {
  doSomething(); //Noncompliant, this might have been done
} else if (a == 2) {
  doSomething();
}
```

Available In:

sonarlint ⊕ | sonarcloud 💩 | sonarqube

© 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