

































-  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... 🔍


 Bug


Regex boundaries should not be used in a way that can never be matched




 Bug


Regex patterns following a possessive quantifier should not always fail




 Bug


Regular expressions should be syntactically valid




 Bug


Assertions comparing incompatible types should not be made




 Bug


JUnit5 inner test classes should be annotated with @Nested




 Bug


Only one method invocation is expected when testing checked exceptions




 Bug


Assertion methods should not be used within the try block of a try-catch catching an Error




 Bug

Getters and setters should access the expected fields



 Bug

Zero should not be a possible denominator



 Bug


Locks should be released


 Bug


"runFinalizersOnExit" should not be called

Switch cases should end with an unconditional "break" statement

Analyze your code

 Code Smell

 Blocker

 cwe cert suspicious

When the execution is not explicitly terminated at the end of a switch case, it continues to execute the statements of the following case. While this is sometimes intentional, it often is a mistake which leads to unexpected behavior.

Noncompliant Code Example

```
switch (myVariable) {
    case 1:
        foo();
        break;
    case 2: // Both 'doSomething()' and 'doSomethingElse()' w
        doSomething();
    default:
        doSomethingElse();
        break;
}
```






Compliant Solution

```
switch (myVariable) {
    case 1:
        foo();
        break;
    case 2:
        doSomething();
        break;
    default:
        doSomethingElse();
        break;
}
```

Exceptions

This rule is relaxed in the following cases:

```
switch (myVariable) {
    case 0: // Empty case used
    case 1:
        doSomething();
        break;
    case 2: // Use of a fallthr
        // fallthrough
    case 3: // Use of return st
        return;
    case 4: // Use of throw sta
        throw new IllegalStateException();
    case 5: // Use of continue
        continue;
}
```

 Bug
"ScheduledThreadPoolExecutor" should not have 0 core threads
 Bug
"Random" objects should be reused
 Bug
The signature of "finalize()" should match that of "Object.finalize()"
 Bug
Jump statements should not occur in "finally" blocks
 Bug

```
default:                                // For the last cas
    doSomethingElse();
}
```

See

- [MITRE, CWE-484](#) - Omitted Break Statement in Switch
- [CERT, MSC17-C](#) - Finish every set of statements associated with a case label with a break statement
- [CERT, MSC52-J](#) - Finish every set of statements associated with a case label with a break statement

Available In:

sonarlint  | **sonarcloud**  | **sonarqube** 