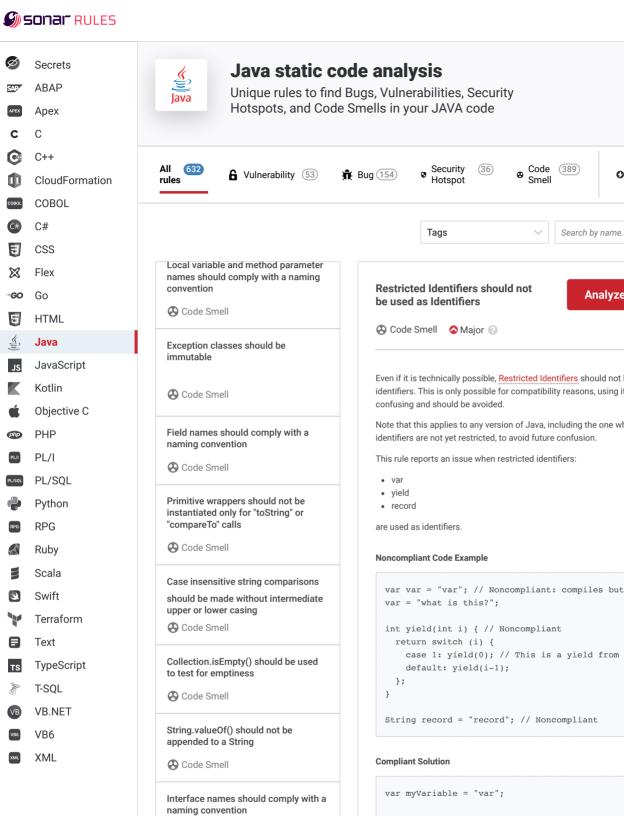


Products >

O Quick 42 Fix

Q



```
Analyze your code
Even if it is technically possible, Restricted Identifiers should not be used as
identifiers. This is only possible for compatibility reasons, using it in Java code is
Note that this applies to any version of Java, including the one where these
identifiers are not yet restricted, to avoid future confusion.
This rule reports an issue when restricted identifiers:
  var var = "var"; // Noncompliant: compiles but this code is
      case 1: yield(0); // This is a yield from switch express
  String record = "record"; // Noncompliant
  int minusOne(int i) {
    return switch (i) {
      case 1: yield(0);
       default: yield(i-1);
```

String myRecord = "record";

sonarlint ⊕ | sonarcloud ♦ | sonarqube

• JLS16, 3.8: Identifiers

Available In:

Code Smell

superfluous Code Smell

removed

statement

Code Smell

"throws" declarations should not be

Unnecessary imports should be

Return of boolean expressions should

not be wrapped into an "if-then-else"

Boolean literals should not be redundant
Code Smell

Modifiers should be declared in the correct order
Code Smell

Empty statements should be removed
Code Smell

Class variable fields should not have public accessibility
Code Smell

© 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