




 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... 🔍

Abstract class names should comply with a naming convention

Code Smell

Strings literals should be placed on the left side when checking for equality

Code Smell

Files should contain an empty newline at the end

Code Smell

Source code should be indented consistently

Code Smell

A close curly brace should be located at the beginning of a line

Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines

Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line

Code Smell

An open curly brace should be located at the beginning of a line

Code Smell

An open curly brace should be located at the end of a line

Code Smell

Tabulation characters should not be used

Code Smell

Functions should not be defined with a variable number of arguments

Code Smell

"enum" fields should not be publicly mutable

Analyze your code

Code Smell

Minor ?

bad-practice

enums are generally thought of as constant, but an enum with a public field or public setter is non-constant. Ideally fields in an enum are private and set in the constructor, but if that's not possible, their visibility should be reduced as much as possible.

#### Noncompliant Code Example

```
public enum Continent {  
  
    NORTH_AMERICA (23, 24709000),  
    // ...  
    EUROPE (50, 39310000);  
  
    public int countryCount; // Noncompliant  
    private int landMass;  
  
    Continent(int countryCount, int landMass) {  
        // ...  
    }  
  
    public void setLandMass(int landMass) { // Noncompliant  
        this.landMass = landMass;  
    }  
}
```

#### Compliant Solution

```
public enum Continent {  
  
    NORTH_AMERICA (23, 24709000),  
    // ...  
    EUROPE (50, 39310000);  
  
    private int countryCount;  
    private int landMass;  
  
    Continent(int countryCount, int landMass) {  
        // ...  
    }  
}
```

Available In:

sonarlint

sonarcloud





sonarqube

https://rules.sonarsource.com/java/RSPEC-3066

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR

1/2

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>Local-Variable Type Inference should be used</div> <div> Code Smell</div>
<div>Migrate your tests from JUnit4 to the new JUnit5 annotations</div> <div> Code Smell</div>
<div>Track uses of disallowed classes</div> <div> Code Smell</div>
<div>Track uses of "@SuppressWarnings" annotations</div> <div> Code Smell</div>