sonar RULES

Products ⌄

- 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

Class variable fields should not have public accessibility

Analyze your code

⚙ Code Smell   ⊖ Minor ⍰   🏷 cwe

Public class variable fields do not respect the encapsulation principle and has three main disadvantages:

- Additional behavior such as validation cannot be added.
- The internal representation is exposed, and cannot be changed afterwards.
- Member values are subject to change from anywhere in the code and may not meet the programmer's assumptions.

By using private attributes and accessor methods (set and get), unauthorized modifications are prevented.

Noncompliant Code Example

```java
public class MyClass {

  public static final int SOME_CONSTANT = 0;     // Complian

  public String firstName;                       // Noncompl

}
```

Compliant Solution

```java
public class MyClass {

  public static final int SOME_CONSTANT = 0;     // Complian

  private String firstName;                      // Complian

  public String getFirstName() {
    return firstName;
  }

  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }

}
```

Exceptions

Because they are not modifiable, this rule ignores `public final` fields. Also, annotated fields, whatever the annotation(s) will be ignored, as annotations are often used by injection frameworks, which in exchange require having public fields.

See

- MITRE, CWE-493 - Critical Public Variable Without Final Modifier

Local-Variable Type Inference should be used

⊗ Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

⊗ Code Smell

Track uses of disallowed classes

⊗ Code Smell

Track uses of "@SuppressWarnings" annotations

⊗ Code Smell

Available In:

sonarlint | sonarcloud | sonarqube