




 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

Inner classes which do not reference their owning classes should be "static"

Analyze your code

 Code Smell

 Major ?

 performance

A non-static inner class has a reference to its outer class, and access to the outer class' fields and methods. That class reference makes the inner class larger and could cause the outer class instance to live in memory longer than necessary.

If the reference to the outer class isn't used, it is more efficient to make the inner class `static` (also called nested). If the reference is used only in the class constructor, then explicitly pass a class reference to the constructor. If the inner class is anonymous, it will also be necessary to name it.

However, while a nested/static class would be more efficient, it's worth noting that there are semantic differences between an inner class and a nested one:

- an inner class can only be instantiated within the context of an instance of the outer class.
- a nested (static) class can be instantiated independently of the outer class.

Noncompliant Code Example

```
public class Fruit {
    // ...

    public class Seed { // Noncompliant; there's no use of th
        int germinationDays = 0;
        public Seed(int germinationDays) {
            this.germinationDays = germinationDays;
        }
        public int getGerminationDays() {
            return germinationDays;
        }
    }
}
```

Compliant Solution

```
public class Fruit {
    // ...

    public static class Seed {
        int germinationDays = 0;
        public Seed(int germinationDays) {
            this.germinationDays = germinationDays;
        }
        public int getGerminationDays() {
            return germinationDays;
        }
    }
}
```

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:  |  | 

© 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](#)