




 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... 🔍

Code Smell

Package names should comply with a naming convention

Code Smell

Nested code blocks should not be used

Code Smell

Array designators "[]" should be on the type, not the variable

Code Smell

Array designators "[]" should be located after the type in method signatures

Code Smell

Type parameter names should comply with a naming convention

Code Smell

Overriding methods should do more than simply call the same method in the super class

Code Smell

Classes that override "clone" should be "Cloneable" and call "super.clone()"

Code Smell

Public constants and fields initialized at declaration should be "static final" rather than merely "final"

Code Smell

Local variable and method parameter names should comply with a naming convention

Code Smell

Exception classes should be immutable

Code Smell

String multiline concatenation should be replaced with Text Blocks

Analyze your code

Code Smell

Major ?

java15

In Java 15 Text Blocks are now official and can be used. The most common pattern for multiline strings in Java < 15 was to write String concatenation. Now it's possible to do it in a more natural way using Text Blocks.

Noncompliant Code Example

```
String textBlock =
    "<html>\n" +
    "    <body>\n" +
    "        <tag>\n" +
    "        </tag>\n" +
    "    </body>\n" +
    "</html>";
```

Compliant Solution

```
String textBlock = """
    <html>
        <body>
            <tag>
            </tag>
        </body>
    </html>""";
```

See

- JEP 378: Text Blocks
- Programmer's Guide To Text Blocks, by Jim Laskey and Stuart Marks

Available In:

sonarlint




sonarcloud

sonarqube

© 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](#)

https://rules.sonarsource.com/java/RSPEC-6126

1/2

<div>Field names should comply with a naming convention</div> <div> Code Smell</div>
<div>Primitive wrappers should not be instantiated only for "toString" or "compareTo" calls</div> <div> Code Smell</div>
<div>Case insensitive string comparisons should be made without intermediate upper or lower casing</div> <div> Code Smell</div>
<div>Collection.isEmpty() should be used to test for emptiness</div>