sonar RULES

Products ⌄

- ⊘ Secrets
- SAP ABAP
- APEX Apex
- C C
- C++ C++
- CloudFormation
- COBOL COBOL
- C# C#
- CSS CSS
- Flex Flex
- GO Go
- HTML HTML
- **Java**
- JS JavaScript
- Kotlin Kotlin
- Objective C
- php PHP
- PL/I PL/I
- PL/SQL PL/SQL
- Python Python
- RPG RPG
- Ruby Ruby
- Scala Scala
- Swift Swift
- Terraform Terraform
- Text Text
- TS TypeScript
- T-SQL T-SQL
- VB.NET VB.NET
- VB6 VB6
- XML 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

String.valueOf() should not be appended to a String

Analyze your code

⊕ Code Smell   ⊖ Minor ?   🏷 clumsy

Appending `String.valueOf()` to a `String` decreases the code readability.

The argument passed to `String.valueOf()` should be directly appended instead.

Noncompliant Code Example

```
public void display(int i){
    System.out.println("Output is " + String.valueOf(i));    /
}
```

Compliant Solution

```
public void display(int i){
    System.out.println("Output is " + i);                    /
}
```

Available In:

sonarlint | sonarcloud | sonarqube

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