## sonar RULES

**Products** ⌄

- 🚫 Secrets
- ⬛ ABAP
- 🟥 Apex
- Ⓒ C
- Ⓒ C++
- ⬛ CloudFormation
- COBOL COBOL
- Ⓒ C#
- CSS
- ✖ Flex
- ⟜GO Go
- HTML
- ☕ **Java**
- JS JavaScript
- Kotlin
- Objective C
- php PHP
- PL/I PL/I
- PL/SQL PL/SQL
- Python
- RPG RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TS TypeScript
- T-SQL
- VB 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

---

### The default unnamed package should not be used

**Analyze your code**

⊗ Code Smell    🍃 Minor ⓘ    🏷 convention

According to the Java Language Specification:

> Unnamed packages are provided by the Java platform principally for convenience when developing small or temporary applications or when just beginning development.

To enforce this best practice, classes located in default package can no longer be accessed from named ones since Java 1.4.

**Noncompliant Code Example**

```
public class MyClass { /* ... */ }
```

**Compliant Solution**

```
package org.example;

public class MyClass{ /* ... */ }
```

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