Security

Hotspot

Java static code analysis

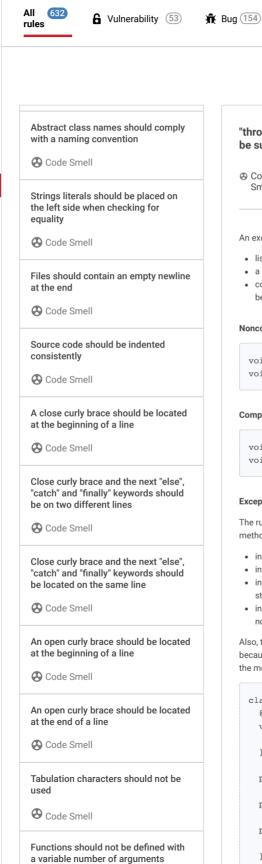
Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

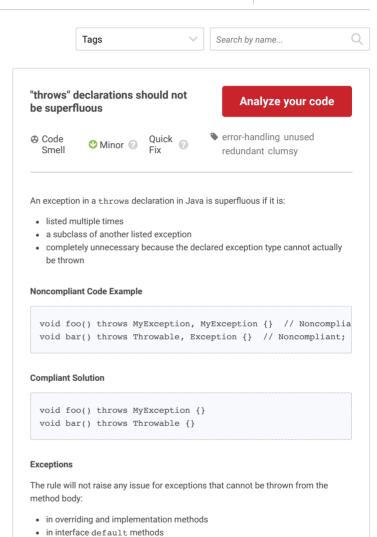


Products ✓

O Quick 42 Fix







in non-private methods that only throw, have empty bodies, or a single return

• in overridable methods (non-final, or not member of a final class, non-static, non-private), if the exception is documented with a proper JavaDoc

Also, the rule won't raise issues on RuntimeException, or one of its descendants,

because explicating runtime exceptions which could be thrown can ultimately help

the method's users, and can even be considered as good practice.

void doSomething() throws IOException {

⊗ Code

Smell

(389)

class A extends B {

@Override

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

© 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