sonar RULES

Products

- 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...

Code Smell

Redundant modifiers should not be used
Code Smell

"private" and "final" methods that don't access instance data should be "static"
Code Smell

Files should not be empty
Code Smell

Collection methods with O(n) performance should be used carefully
Code Smell

"Exception" should not be caught when not required by called methods
Code Smell

"collect" should be used with "Streams" instead of "list::add"
Code Smell

Switches should be used for sequences of simple "String" tests
Code Smell

"final" classes should not have "protected" members
Code Smell

Underscores should be used to make large numbers readable
Code Smell

"Serializable" inner classes of "Serializable" classes should be static
Code Smell

Member variable visibility should be specified
Code Smell

Sections of code should not be commented out

Analyze your code

Code Smell Major unused

Programmers should not comment out code as it bloats programs and reduces readability.

Unused code should be deleted and can be retrieved from source control history if required.

Available In:

sonarlint sonarcloud sonarqube

Classes and methods that rely on the default system encoding should not be used

Code Smell

Simple class names should be used

Code Smell

Variables should not be declared before they are relevant

Code Smell

Extensions and implementations should not be redundant

Code Smell