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

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...

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

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

Exception types should not be tested using "instanceof" in catch blocks

Analyze your code

Code Smell Major error-handling cert clumsy

Multiple catch blocks of the appropriate type should be used instead of catching a general exception, and then testing on the type.

Noncompliant Code Example

```
try {
  /* ... */
} catch (Exception e) {
  if(e instanceof IOException) { /* ... */ }         // Nonc
  if(e instanceof NullPointerException{ /* ... */ }  // Nonc
}
```

Compliant Solution

```
try {
  /* ... */
} catch (IOException e) { /* ... */ }                // Comp
} catch (NullPointerException e) { /* ... */ }       // Comp
```

See

- CERT, ERR51-J. - Prefer user-defined exceptions over more general exception types

Available In:

sonarlint sonarcloud sonarqube

Code Smell

"==" and "!=" should not be used when "equals" is overridden

Code Smell

An abstract class should have both abstract and concrete methods

Code Smell

Sets with elements that are enum values should be replaced with EnumSet

Code Smell

String operations should not rely on the default system locale