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...

Guava
Code Smell

Nullness of parameters should be guaranteed
Code Smell

"Integer.toHexString" should not be used to build hexadecimal strings
Code Smell

Asserts should not be used to check the parameters of a public method
Code Smell

Assignments should not be redundant
Code Smell

Methods should not have identical implementations
Code Smell

"java.nio.Files#delete" should be preferred
Code Smell

Unused "private" classes should be removed
Code Smell

"Stream.peek" should be used with caution
Code Smell

"Map.get" and value test should be replaced with single method call
Code Smell

"@RequestMapping" methods should not be "private"
Code Smell

Raw types should not be used
Code Smell

## Tests method should not be annotated with competing annotations

Analyze your code

Bug Major tests

Annotating unit tests with more than one test-related annotation is not only useless but could also result in unexpected behavior like failing tests or unwanted side-effects.

This rule reports an issue when a test method is annotated with more than one of the following competing annotation:

- @Test
- @RepeatedTest
- @ParameterizedTest
- @TestFactory
- @TestTemplate

**Noncompliant Code Example**

```
@Test
@RepeatedTest(2) // Noncompliant, this test will be repeated
void test() { }

@ParameterizedTest
@Test
@MethodSource("methodSource")
void test2(int argument) { } // Noncompliant, this test will
```

**Compliant Solution**

```
@RepeatedTest(2)
void test() { }

@ParameterizedTest
@MethodSource("methodSource")
void test2(int argument) { }
```

Available In:
sonarlint sonarcloud sonarqube

"Arrays.stream" should be used for primitive arrays

Code Smell

Printf-style format strings should be used correctly

Code Smell

Assertion arguments should be passed in the correct order

Code Smell

Ternary operators should not be nested

Code Smell