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

"Stream.collect()" calls should not be redundant

Code Smell

Local constants should follow naming conventions for constants

Code Smell

Unit tests should throw exceptions

Code Smell

Test methods should comply with a naming convention

Code Smell

Value-based objects should not be serialized

Code Smell

Default annotation parameter values should not be passed as arguments

Code Smell

Method parameters should be declared with base types

Code Smell

Fields should not be initialized to default values

Code Smell

Multiple loops over the same set should be combined

Code Smell

Classes without "public" constructors should be "final"

Code Smell

Unnecessary semicolons should be omitted

Code Smell

Unused assignments should be removed

Analyze your code

Code Smell Major cwe cert unused

A dead store happens when a local variable is assigned a value that is not read by any subsequent instruction. Calculating or retrieving a value only to then overwrite it or throw it away, could indicate a serious error in the code. Even if it's not an error, it is at best a waste of resources. Therefore all calculated values should be used.

Noncompliant Code Example

```
i = a + b; // Noncompliant; calculation result not used befo
i = compute();
```

Compliant Solution

```
i = a + b;
i += compute();
```

Exceptions

This rule ignores initializations to -1, 0, 1, `null`, `true`, `false` and `""`.

See

- MITRE, CWE-563 - Assignment to Variable without Use ('Unused Variable')
- CERT, MSC13-C. - Detect and remove unused values
- CERT, MSC56-J. - Detect and remove superfluous code and values

Available In:

sonarlint sonarcloud sonarqube

Literal boolean values and nulls should not be used in assertions

Code Smell

Test assertions should include messages

Code Smell

Mutable members should not be stored or returned directly

Code Smell

Redundant modifiers should not be used

Code Smell