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...

mixed with other assertions

Code Smell

"@Deprecated" code marked for removal should never be used

Code Smell

Vararg method arguments should not be confusing

Code Smell

Whitespace for text block indent should be consistent

Code Smell

'List.remove()' should not be used in ascending 'for' loops

Code Smell

Collection constructors should not be used as java.util.function.Function

Code Smell

"else" statements should be clearly matched with an "if"

Code Smell

"Class.forName()" should not load JDBC 4.0+ drivers

Code Smell

Java features should be preferred to 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

AssertJ assertions with "Consumer" arguments should contain assertion inside consumers

Analyze your code

Bug | Major | tests

AssertJ assertions taking `Consumer` objects as arguments are expected to contain "requirements", which should themselves be expressed as assertions. This concerns the following methods: allSatisfy, anySatisfy, hasOnlyOneElementSatisfying, isInstanceOfSatisfying, noneSatisfy, satisfies, satisfiesAnyOf, zipSatisfy.

These methods are assuming the `Consumer` will do the assertions itself. If you do not do any assertion in the `Consumer`, it probably means that you are inadvertently only partially testing your object.

This rule raises an issue when a `Consumer` argument of any of the above methods does not contain any assertion.

Noncompliant Code Example

```
assertThat(myObject).isInstanceOfSatisfying(String.class, s
assertThat(myObject).satisfies("Hello"::equals); // Noncompl
```

Compliant Solution

```
assertThat(myObject).isInstanceOfSatisfying(String.class, s
assertThat(myObject).satisfies(obj -> assertThat(obj).isEqua
```

Available In:

sonarlint | sonarcloud | sonarqube

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