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...

"@CheckForNull" or "@Nullable" should not be used on primitive types

Analyze your code

Code Smell | Minor | Quick Fix

By definition, primitive types are not Objects and so they can't be `null`. Adding `@CheckForNull` or `@Nullable` on primitive types adds confusion and is useless.

This rule raises an issue when `@CheckForNull` or `@Nullable` is set on a method returning a primitive type: byte, short, int, long, float, double, boolean, char.

Noncompliant Code Example

```
@CheckForNull
boolean isFoo() {
 ...
}
```

Compliant Solution

```
boolean isFoo() {
 ...
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Local-Variable Type Inference should be used

Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

Code Smell

Track uses of disallowed classes

Code Smell

Track uses of "@SuppressWarnings" annotations

Code Smell