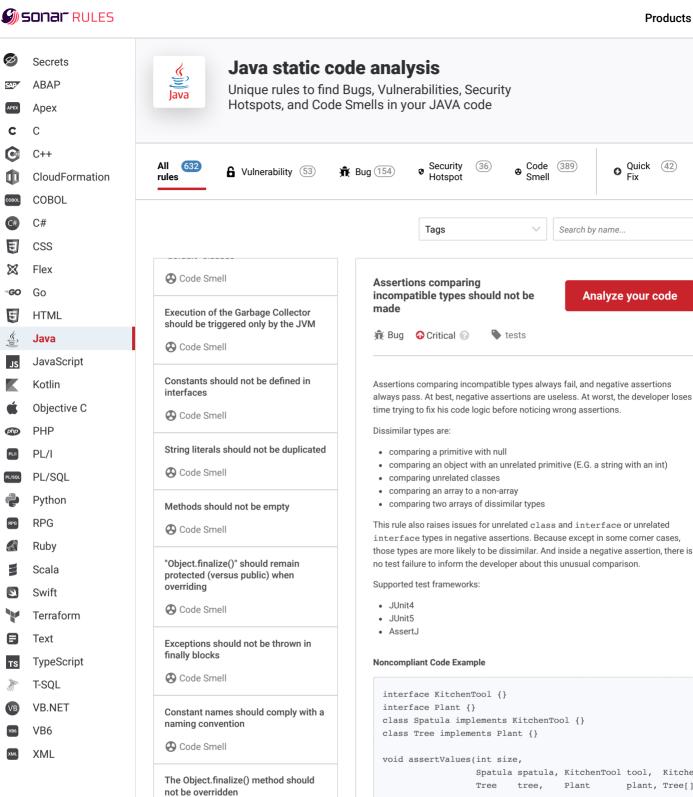


Products ✓



```
Spatula spatula, KitchenTool tool, Kitche
                                            plant, Tree[]
// Whatever the given values, those negative assertions wi
                                       // Noncompliant; p
assertThat(size).isNotNull();
assertThat(spatula).isNotEqualTo(tree); // Noncompliant; u
assertThat(tool).isNotSameAs(tools);
                                       // Noncompliant; a
assertThat(trees).isNotEqualTo(tools); // Noncompliant; i
// Those assertions will always fail
assertThat(size).isNull();
                                                 // Noncom
assertThat(spatula).isEqualTo(tree);
// Those negative assertions are more likely to always pas
assertThat(spatula).isNotEqualTo(plant); // Noncompliant;
assertThat(tool).isNotEqualTo(plant);
                                         // Noncompliant;
```

See

Code Smell

6 Vulnerability

Vulnerability

Vulnerability

securely

XML operations should not be

vulnerable to injection attacks

JSON operations should not be vulnerable to injection attacks

XML signatures should be validated

Java static code analysis:
XML parsers should not be vulnerable to Denial of Service attacks
<b>6</b> Vulnerability
XML parsers should not load external schemas
<b>6</b> Vulnerability
Mobile database encryption keys should not be disclosed
<b>☆</b> Vulnerability
Reflection should not be vulnerable to injection attacks
<b>₲</b> Vulnerability

{rule:java:S2159} - Silly equality checks should not be made

Available In:

sonarlint ⊖ | sonarcloud ♦ | sonarqube |

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy