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...

Abstract class names should comply with a naming convention
Code Smell

Strings literals should be placed on the left side when checking for equality
Code Smell

Files should contain an empty newline at the end
Code Smell

Source code should be indented consistently
Code Smell

A close curly brace should be located at the beginning of a line
Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines
Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line
Code Smell

An open curly brace should be located at the beginning of a line
Code Smell

An open curly brace should be located at the end of a line
Code Smell

Tabulation characters should not be used
Code Smell

Functions should not be defined with a variable number of arguments
Code Smell

"equals(Object obj)" and "hashCode()" should be overridden in pairs

Analyze your code

Bug Minor cwe cert

According to the Java Language Specification, there is a contract between `equals(Object)` and `hashCode()`:

> If two objects are equal according to the `equals(Object)` method, then calling the `hashCode` method on each of the two objects must produce the same integer result.
> It is not required that if two objects are unequal according to the `equals(java.lang.Object)` method, then calling the `hashCode` method on each of the two objects must produce distinct integer results.
> However, the programmer should be aware that producing distinct integer results for unequal objects may improve the performance of hashtables.

In order to comply with this contract, those methods should be either both inherited, or both overridden.

Noncompliant Code Example

```
class MyClass {    // Noncompliant - should also override "h

  @Override
  public boolean equals(Object obj) {
    /* ... */
  }

}
```

Compliant Solution

```
class MyClass {    // Compliant

  @Override
  public boolean equals(Object obj) {
    /* ... */
  }

  @Override
  public int hashCode() {
    /* ... */
  }

}
```

See

- MITRE, CWE-581 - Object Model Violation: Just One of Equals and Hashcode Defined
- CERT, MET09-J. - Classes that define an equals() method must also define a hashCode() method

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

Available In:

sonarlint sonarcloud sonarqube