

Text

T-SQL

VB.NET

VB₆

XML

TypeScript



C# static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C# code



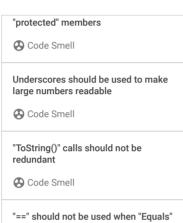
₩ Bug (76)

Security Hotspot

(28)

Quick 52 Fix

Tags



is overridden

Code Smell

An abstract class should have both abstract and concrete methods

Code Smell

Multiple variables should not be declared on the same line

Code Smell

Culture should be specified for "string" operations

Code Smell

"switch" statements should have at least 3 "case" clauses

Code Smell

break statements should not be used except for switch cases

Code Smell

String literals should not be duplicated

Code Smell

Files should contain an empty newline at the end

Code Smell

Unused "using" should be removed

Code Smell

Floating point numbers should not be tested for equality

Analyze your code

Search by name.

👬 Bug 🔷 Major 🕝

Floating point math is imprecise because of the challenges of storing such values in a binary representation. Even worse, floating point math is not associative; push a float or a double through a series of simple mathematical operations and the answer will be different based on the order of those operation because of the rounding that takes place at each step.

Even simple floating point assignments are not simple:

```
float f = 0.100000001f: // 0.1
double d = 0.1000000000000001; // 0.1
```

(Results will vary based on compiler and compiler settings)

Therefore, the use of the equality (==) and inequality (!=) operators on float or double values is almost always an error.

This rule checks for the use of direct and indirect equality/inequality tests on floats and doubles

Noncompliant Code Example

```
float myNumber = 3.146f;
if ( myNumber == 3.146f ) //Noncompliant. Because of floatin
{
  // ...
}
if (myNumber <= 3.146f && mNumber >= 3.146f) // Noncompliant
  // ...
}
if (myNumber < 4 \mid \mid myNumber > 4) // Noncompliant indirect i
{
  // ...
```

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

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

Code Smell

Tabulation characters should not be used

Code Smell

Methods and properties should be named in PascalCase

Code Smell

Track uses of in-source issue suppressions

Code Smell