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

C# static code analysis

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

All rules 409 | Vulnerability 34 | Bug 76 | Security Hotspot 28 | Code Smell 271 | Quick Fix 52

Tags

Search by name...

Value types should implement "IEquatable<T>"
Code Smell

Finalizers should not be empty
Code Smell

"[ExpectedException]" should not be used
Code Smell

"this" should not be exposed from constructors
Code Smell

Types should not have members with visibility set higher than the type's visibility
Code Smell

Fields should be private
Code Smell

"try" statements with identical "catch" and/or "finally" blocks should be merged
Code Smell

NullReferenceException should not be caught
Code Smell

Functions should not have too many lines of code
Code Smell

"for" loop stop conditions should be invariant
Code Smell

Statements should be on separate lines
Code Smell

Classes should not be coupled to too many other classes (Single Responsibility Principle)

Parameters should be passed in the correct order

Analyze your code

Code Smell | Major

When the names of parameters in a method call match the names of the method arguments, it contributes to clearer, more readable code. However, when the names match, but are passed in a different order than the method arguments, it indicates a mistake in the parameter order which will likely lead to unexpected results.

Noncompliant Code Example

```
public double Divide(int divisor, int dividend)
{
    return divisor/dividend;
}

public void DoTheThing()
{
    int divisor = 15;
    int dividend = 5;

    double result = Divide(dividend, divisor);  // Noncompli
    //...
}
```

Compliant Solution

```
public double Divide(int divisor, int dividend)
{
    return divisor/dividend;
}

public void DoTheThing()
{
    int divisor = 15;
    int dividend = 5;

    double result = Divide(divisor, dividend);
    //...
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Responsibility Principle)

☢ Code Smell

"switch case" clauses should not have too many lines of code

☢ Code Smell

Magic numbers should not be used

☢ Code Smell

Standard outputs should not be used directly to log anything

☢ Code Smell

Files should not have too many lines of code