

- 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... 🔍

Overloads with a "CultureInfo" or an "IFormatProvider" parameter should be used

Code Smell

Types should not extend outdated base types

Code Smell

Properties should be preferred

Code Smell

Generics should be used when appropriate

Code Smell

Type names should not match namespaces

Code Smell

Strings should be normalized to uppercase

Code Smell

Exceptions should provide standard constructors

Code Smell

Assemblies should be marked with "NeutralResourcesLanguageAttribute"

Code Smell

Interfaces should not be empty

Code Smell

Enumerations should have "Int32" storage

Code Smell

Generic methods should provide type parameters

Code Smell

Multidimensional arrays should not be used

### Inheritance tree of classes should not be too deep





Analyze your code

Code Smell Major ? design

Inheritance is certainly one of the most valuable concepts in object-oriented programming. It's a way to compartmentalize and reuse code by creating collections of attributes and behaviors called classes which can be based on previously created classes. But abusing this concept by creating a deep inheritance tree can lead to very complex and unmaintainable source code. Most of the time a too deep inheritance tree is due to bad object oriented design which has led to systematically use 'inheritance' when for instance 'composition' would suit better.

This rule raises an issue when the inheritance tree, starting from `Object` has a greater depth than is allowed.

Available In:  
sonarlint | sonarcloud | sonarqube

 Code Smell
<b>"static readonly" constants should be "const" instead</b>  Code Smell
<b>Strings or integral types should be used for indexers</b>  Code Smell
<b>Parameter names should not duplicate the names of their methods</b>  Code Smell
<b>Track use of "NotImplementedException"</b>