

- 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... 🔍

"protected" members

Code Smell

Underscores should be used to make large numbers readable

Code Smell

"ToString()" calls should not be redundant

Code Smell

"==" should not be used when "Equals" 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

Child class fields should not differ from parent class fields only by capitalization

Analyze your code

Code Smell Critical ? pitfall

Having a field in a child class with a name that differs from a parent class' field only by capitalization is sure to cause confusion. Such child class fields should be renamed.

Noncompliant Code Example

```
public class Fruit
{
    protected string plantingSeason;
    //...
}

public class Raspberry : Fruit
{
    protected string plantingseason; // Noncompliant
    // ...
}
```

Compliant Solution

```
public class Fruit
{
    protected string plantingSeason;
    //...
}

public class Raspberry : Fruit
{
    protected string whenToPlant;
    // ...
}
```

Or

```
public class Fruit
{
    protected string plantingSeason;
    //...
}

public class Raspberry : Fruit
{
    // field removed; parent field will be used instead
    // ...
}
```

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

Exceptions

This rule ignores same-name fields that are `static` in both the parent and child classes. It also ignores `private` parent class fields, but in all other such cases, the child class field should be renamed.

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](#)