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 shadow parent class fields

Analyze your code

Code Smell | Blocker | confusing

Having a variable with the same name in two unrelated classes is fine, but do the same thing within a class hierarchy and you'll get confusion at best, chaos at worst.

Noncompliant Code Example

```
public class Fruit
{
  protected Season ripe;
  protected Color flesh;

  // ...
}

public class Raspberry : Fruit
{
  private bool ripe; // Noncompliant
  private static Color FLESH; // Noncompliant
}
```

Compliant Solution

```
public class Fruit
{
  protected Season ripe;
  protected Color flesh;

  // ...
}

public class Raspberry : Fruit
{
  private bool ripened;
  private static Color FLESH_COLOR;
}
```

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.

```
public class Fruit
{
  private Season ripe;
  // ...
}

public class Raspberry : Fruit
```

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

```
{
  private Season ripe;  // Compliant as parent field 'ripe'
  // ...
}
```

Available In:

sonarlint | sonarcloud | sonarqube