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...

"switch/Select" statements should contain a "default/Case Else" clauses
Code Smell

"if ... else if" constructs should end with "else" clauses
Code Smell

Control structures should use curly braces
Code Smell

Expressions should not be too complex
Code Smell

ASP.NET HTTP request validation feature should not be disabled
Vulnerability

Serialization constructors should be secured
Vulnerability

Calculations should not overflow
Bug

Floating point numbers should not be tested for equality
Bug

Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression
Code Smell

Use a testable date/time provider.
Code Smell

Property names should not match get methods
Code Smell

Locales should be set for data types

"abstract" classes should not have "public" constructors

Analyze your code

Code Smell | Major | confusing

Since `abstract` classes can't be instantiated, there's no point in their having `public` or `internal` constructors. If there is basic initialization logic that should run when an extending class instance is created, you can by all means put it in a constructor, but make that constructor `private`, `private protected` or `protected`.

Noncompliant Code Example

```
abstract class Base
{
    public Base() // Noncompliant, should be private, privat
    {
      //...
    }
}
```

Compliant Solution

```
abstract class Base
{
    protected Base()
    {
      //...
    }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Code Smell

Literals should not be passed as localized parameters

Code Smell

Operators should be overloaded consistently

Code Smell

Method signatures should not contain nested generic types

Code Smell

Enumeration members should not be named "Reserved"