**internal（C# 参考）**

**Visual Studio 2005**

[其他版本](javascript:void(0))

https://i-msdn.sec.s-msft.com/Areas/Epx/Content/Images/ImageSprite.png?v=636555423371300703

* [Visual Studio 2015](https://msdn.microsoft.com/zh-cn/library/7c5ka91b(v=vs.140).aspx)
* [Visual Studio 2013](https://msdn.microsoft.com/zh-cn/library/7c5ka91b(v=vs.120).aspx)
* [Visual Studio 2012](https://msdn.microsoft.com/zh-cn/library/7c5ka91b(v=vs.110).aspx)
* [Visual Studio 2010](https://msdn.microsoft.com/zh-cn/library/7c5ka91b(v=vs.100).aspx)
* [Visual Studio 2008](https://msdn.microsoft.com/zh-cn/library/7c5ka91b(v=vs.90).aspx)

**internal** 关键字是类型和类型成员的访问修饰符。只有在同一程序集的文件中，内部类型或成员才是可访问的，如下例所示：

[复制](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_7597fb97-38e6-4b34-970c-4c2b46953f3d');" \o "复制到剪贴板。)

public class BaseClass

{

// Only accessible within the same assembly

internal static int x = 0;

}

有关 **internal** 与其他访问修饰符的比较，请参见[可访问性级别](https://msdn.microsoft.com/zh-cn/library/ba0a1yw2(v=vs.80).aspx)和[访问修饰符（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173121(v=vs.80).aspx)。

有关程序集的更多信息，请参见[程序集和全局程序集缓存（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173099(v=vs.80).aspx)：

内部访问通常用于基于组件的开发，因为它使一组组件能够以私有方式进行合作，而不必向应用程序代码的其余部分公开。例如，用于生成图形用户界面的框架可以提供“控件”类和“窗体”类，这些类通过使用具有内部访问能力的成员进行合作。由于这些成员是内部的，它们不向正在使用框架的代码公开。

从定义具有内部访问能力的类型或成员的程序集外部引用该类型或成员是错误的。

|  |
| --- |
| **Note注意** |
| 尽管不能用 C# 重写 **internalvirtual** 方法，但可以用某些语言（如使用 Ilasm.exe 的文本 Microsoft 中间语言 (MSIL)）重写它。 |

**示例**

该示例包含两个文件：Assembly1.cs 和 Assembly2.cs。第一个文件包含内部基类 BaseClass。在第二个文件中，实例化 BaseClass 的尝试将产生错误。

[复制](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_cc764ea0-f363-498e-9abb-7fadb10ec1a3');)

// Assembly1.cs

// compile with: /target:library

internal class BaseClass

{

public static int intM = 0;

}

[复制](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_bf82fd83-91e7-4658-a069-81a9c5ccbae7');)

// Assembly1\_a.cs

// compile with: /reference:Assembly1.dll

class TestAccess

{

static void Main()

{

BaseClass myBase = new BaseClass(); // CS0122

}

}

在此示例中，使用与示例 1 中所用的文件相同的文件，并将 BaseClass 的可访问性级别更改为 **public**。还将成员 IntM 的可访问性级别更改为 **internal**。在此例中，您可以实例化类，但不能访问内部成员。

[复制](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_cffecf03-5a7a-467c-a3aa-2a2811750ff1');)

// Assembly2.cs

// compile with: /target:library

public class BaseClass

{

internal static int intM = 0;

}

[复制](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_c84fe1fa-43e3-4fd3-8db1-4fe584b14d98');)

// Assembly2\_a.cs

// compile with: /reference:Assembly1.dll

public class TestAccess

{

static void Main()

{

BaseClass myBase = new BaseClass(); // Ok.

BaseClass.intM = 444; // CS0117

}

}

**C# 语言规范**

有关更多信息，请参见 [C# 语言规范](https://msdn.microsoft.com/zh-cn/library/ms228593(v=vs.80).aspx)中的以下各章节：

* 3.5.1 声明的可访问性
* 3.5.4 可访问性约束
* 10.2.3 访问修饰符
* 10.2.6.2 声明的可访问性