特性可以应用于泛型类型中，方式与应用于非泛型类型相同。有关应用特性的更多信息，请参见 [特性（C# 和 Visual Basic）](https://msdn.microsoft.com/zh-cn/library/z0w1kczw.aspx)。

自定义特性只允许引用开放泛型类型（未提供类型参数的泛型类型）和封闭构造泛型类型（为所有类型参数提供参数）。

下面的示例使用此自定义特性：

class CustomAttribute : System.Attribute

{

public System.Object info;

}

特性可以引用开放式泛型类型：

public class GenericClass1<T> { }

[CustomAttribute(info = typeof(GenericClass1<>))]

class ClassA { }

使用数目适当的若干个逗号指定多个类型参数。在此示例中，GenericClass2 有两个类型参数：

public class GenericClass2<T, U> { }

[CustomAttribute(info = typeof(GenericClass2<,>))]

class ClassB { }

特性可以引用封闭式构造泛型类型：

public class GenericClass3<T, U, V> { }

[CustomAttribute(info = typeof(GenericClass3<int, double, string>))]

class ClassC { }

引用泛型类型参数的特性将导致编译时错误：

//[CustomAttribute(info = typeof(GenericClass3<int, T, string>))] //Error

class ClassD<T> { }

不能从 [Attribute](https://msdn.microsoft.com/zh-cn/library/system.attribute.aspx) 继承泛型类型：

//public class CustomAtt<T> : System.Attribute {} //Error

若要在运行时获得有关泛型类型或类型参数的信息，可以使用 [System.Reflection](https://msdn.microsoft.com/zh-cn/library/system.reflection.aspx) 的方法。有关更多信息，请参见[泛型和反射（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173128.aspx)

**请参阅**

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)

[泛型（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/512aeb7t.aspx)

[利用特性扩展元数据](https://msdn.microsoft.com/zh-cn/library/5x6cd29c.aspx)