## [委托](https://msdn.microsoft.com/zh-cn/library/900fyy8e.aspx)可以与命名方法关联。使用命名方法对委托进行实例化时，该方法将作为参数传递，例如：

// Declare a delegate:

delegate void Del(int x);

// Define a named method:

void DoWork(int k) { /\* ... \*/ }

// Instantiate the delegate using the method as a parameter:

Del d = obj.DoWork;

这被称为使用命名的方法。使用命名方法构造的委托可以封装[静态](https://msdn.microsoft.com/zh-cn/library/98f28cdx.aspx)方法或实例方法。在早期版本的 C# 中，命名方法是对委托进行实例化的唯一方式。但是，在不希望付出创建新方法的系统开销时，C# 使您可以对委托进行实例化，并立即指定委托在被调用时将处理的代码块。代码块可以包含 lambda 表达式或匿名方法。有关更多信息，请参见 [匿名函数（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/bb882516.aspx)。

**备注**

作为委托参数传递的方法必须与委托声明具有相同的签名。

委托实例可以封装静态或实例方法。

尽管委托可以使用 [out](https://msdn.microsoft.com/zh-cn/library/t3c3bfhx.aspx) 参数，但建议您不要将其用于多路广播事件委托，因为您无法知道哪个委托将被调用。

**示例 1**

以下是声明及使用委托的一个简单示例。注意，委托 Del 和关联的方法 MultiplyNumbers 具有相同的签名

// Declare a delegate

delegate void Del(int i, double j);

class MathClass

{

static void Main()

{

MathClass m = new MathClass();

// Delegate instantiation using "MultiplyNumbers"

Del d = m.MultiplyNumbers;

// Invoke the delegate object.

System.Console.WriteLine("Invoking the delegate using 'MultiplyNumbers':");

for (int i = 1; i <= 5; i++)

{

d(i, 2);

}

// Keep the console window open in debug mode.

System.Console.WriteLine("Press any key to exit.");

System.Console.ReadKey();

}

// Declare the associated method.

void MultiplyNumbers(int m, double n)

{

System.Console.Write(m \* n + " ");

}

}

/\* Output:

Invoking the delegate using 'MultiplyNumbers':

2 4 6 8 10

\*/

**示例 2**

在下面的示例中，一个委托被同时映射到静态方法和实例方法，并分别返回特定的信息。

// Declare a delegate

delegate void Del();

class SampleClass

{

public void InstanceMethod()

{

System.Console.WriteLine("A message from the instance method.");

}

static public void StaticMethod()

{

System.Console.WriteLine("A message from the static method.");

}

}

class TestSampleClass

{

static void Main()

{

SampleClass sc = new SampleClass();

// Map the delegate to the instance method:

Del d = sc.InstanceMethod;

d();

// Map to the static method:

d = SampleClass.StaticMethod;

d();

}

}

/\* Output:

A message from the instance method.

A message from the static method.

\*/

**请参阅**

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

[委托（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173171.aspx)

[匿名方法（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/0yw3tz5k.aspx)

[如何：合并委托（多路广播委托）（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173175.aspx)

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