# C# 委托（Delegate）

C# 中的委托（Delegate）类似于 C 或 C++ 中函数的指针。**委托（Delegate）** 是存有对某个方法的引用的一种引用类型变量。引用可在运行时被改变。

委托（Delegate）特别用于实现事件和回调方法。所有的委托（Delegate）都派生自 **System.Delegate** 类。

## 声明委托（Delegate）

委托声明决定了可由该委托引用的方法。委托可指向一个与其具有相同标签的方法。

例如，假设有一个委托：

public delegate int MyDelegate (string s);

上面的委托可被用于引用任何一个带有一个单一的 *string* 参数的方法，并返回一个 *int* 类型变量。

声明委托的语法如下：

delegate <return type> <delegate-name> <parameter list>

## 实例化委托（Delegate）

一旦声明了委托类型，委托对象必须使用 **new** 关键字来创建，且与一个特定的方法有关。

下面的实例演示了委托的声明、实例化和使用，该委托可用于引用带有一个整型参数的方法，并返回一个整型值。

using System;

delegate int NumberChanger(int n); //声明委托

namespace DelegateAppl{

class TestDelegate

{

static int num = 10;

public static int AddNum(int p)

{

num += p;

return num;

}

public static int MultNum(int q)

{

num \*= q;

return num;

}

public static int getNum()

{

return num;

}

static void Main(string[] args)

{

// 创建委托实例

NumberChanger nc1 = new NumberChanger(AddNum); //实例委托

NumberChanger nc2 = new NumberChanger(MultNum);

// 使用委托对象调用方法

nc1(25); //调用委托

Console.WriteLine("Value of Num: {0}", getNum());

nc2(5);

Console.WriteLine("Value of Num: {0}", getNum());

Console.ReadKey();

}

}}

## 委托的多播（Multicasting of a Delegate）

委托对象可使用 "+" 运算符进行合并。一个合并委托调用它所合并的两个委托。只有相同类型的委托可被合并。"-" 运算符可用于从合并的委托中移除组件委托。

使用委托的这个有用的特点，您可以创建一个委托被调用时要调用的方法的调用列表。

NumberChanger nc; //创建委托指针

NumberChanger nc1 = new NumberChanger(AddNum); //实例化委托一

NumberChanger nc2 = new NumberChanger(MultNum); //实例化委托二

nc = nc1; //将委托指针指向委托一

nc += nc2; //委托一加上委托二