

# C# 接口 ( Interface )

接口定义了所有类继承接口时应遵循的语法合同。接口定义了语法合同 "是什么" 部分，派生类定义了语法合同 "怎么做" 部分。

接口定义了属性、方法和事件，这些都是接口的成员。接口只包含了成员的声明。成员的定义是派生类的责任。接口提供了派生类应遵循的标准结构。

接口使得实现接口的类或结构在形式上保持一致。

抽象类在某种程度上与接口类似，但是，它们大多只是用在当只有少数方法由基类声明由派生类实现时。

定义接口: MyInterface.cs

接口使用 `interface` 关键字声明，它与类的声明类似。接口声明默认是 `public` 的。下面是一个接口声明的实例：

```
interface IMyInterface{  
    void MethodToImplement();}
```

以上代码定义了接口 `IMyInterface`。通常接口命令以 `I` 字母开头，这个接口只有一个方法 `MethodToImplement()`，没有参数和返回值，当然我们可以按照需求设置参数和返回值。

值得注意的是，该方法并没有具体的实现。

接下来我们来实现以上接口：InterfaceImplementer.cs

```
using System;  
  
interface IMyInterface{  
    // 接口成员  
  
    void MethodToImplement();}  
  
class InterfaceImplementer : IMyInterface{  
    static void Main()
```

```

{
    InterfaceImplementer iImp = new InterfaceImplementer();
    iImp.MethodToImplement();
}

public void MethodToImplement()
{
    Console.WriteLine("MethodToImplement() called.");
}
}

```

InterfaceImplementer 类实现了 IMyInterface 接口，接口的实现与类的继承语法格式类似：

```
class InterfaceImplementer : IMyInterface
```

继承接口后，我们需要实现接口的方法 MethodToImplement()，方法名必须与接口定义的方法名一致。

---

接口继承: InterfaceInheritance.cs

以下实例定义了两个接口 IMyInterface 和 IParentInterface。

如果一个接口继承其他接口，那么实现类或结构就需要实现所有接口的成员。

以下实例 IMyInterface 继承了 IParentInterface 接口，因此接口实现类必须实现 MethodToImplement() 和 ParentInterfaceMethod() 方法：

```

using System;

interface IParentInterface{
    void ParentInterfaceMethod();}

interface IMyInterface : IParentInterface{
    void MethodToImplement();}

class InterfaceImplementer : IMyInterface{
    static void Main()
    {

```

```
InterfaceImplementer iImp = new InterfaceImplementer();  
iImp.MethodToImplement();  
iImp.ParentInterfaceMethod();  
}
```

```
public void MethodToImplement()  
{  
    Console.WriteLine("MethodToImplement() called.");  
}
```

```
public void ParentInterfaceMethod()  
{  
    Console.WriteLine("ParentInterfaceMethod() called.");  
}}
```

实例输出结果为：

MethodToImplement() called.ParentInterfaceMethod() called.