C#基础之接口：

接口的定义是 指定一组函数成员而不实现成员的引用类型，其他类型和接口可以继承这个接口。

接口主要有以下**特点**：

1. 通过接口可以实现多重继承，C#接口的成员不能有public、protected、internal、private等修饰符。原因很简单，接口里面的方法要由外面接口实现去实现方法体，那么其修饰符必然是public。C#接口中的成员默认是public。
2. 接口成员不能有new、static、abstract、override、virtual修饰符。（特例）有一点要注意，当一个接口实现一个接口，这2个接口中有相同的方法时，可用new关键字隐藏父接口中的方法。
3. 接口中只包含成员的签名，接口没有构造函数，所以不能直接使用new对接口进行实例化。接口中只能包含方法、属性、事件和索引的组合。接口一旦实现，实现类必须实现接口中的所有成员，除非实现类本身就是抽象类。
4. C#是单继承，接口是解决C#里面类可以同时继承多个基类的问题。

接口用来在C#中实现多继承，接口中的方法是没有实体的，不能直接创建实例。只能通过实现类来使用接口中的方法，实现类必须实现接口中的所有成员。

接口和抽象类的区别：

**接口用于规范，抽象类用于共性**。

抽象类是类，只能被单继承（一个子类只有一个父类）。但是接口却可以依次实现多个。

接口中只能声明方法、属性、事件和索引器。而抽象类中可以有方法的实现（非抽象方法），也可以定义非静态的类变量（还是具有类的功能）。

抽象类可以提供某些方法的部分实现，接口不可以。抽象类的实例是它的子类给出的。接口的实例是实现接口的类给出的。

在抽象类中加入一个方法，那么它的子类就同时有了这个方法。而在接口中加入新的方法，那么实现它的类就要重新编写。

接口成员被定义为公共的，但抽象类中的成员也可以是私有的、受保护的。

C#接口的隐式与显示实现

通俗的来讲，“显示接口实现”就是使用接口名称作为方法名的前缀；而传统的实现方法称之为：“隐式接口实现”。

 interface IAnimal  
　　　　{  
　　　　　　　void Dog();  
　　　　}  
　　　　class Animal:IAnimal  
　　　　{  
　　　　　　public void Dog()  
　　　　　　{  
　　　　　　　　Console.WriteLine("dog...");  
　　　　　　}  
　　　　 }

定义了一个IAnimal接口，一般我们都会这么调用：

// 通过类调用

Animal animal = new Animal();

Animal.Dog();

// 或者通过接口调用

IAnimal animal = new Animal();

Animal.Dog();

类和接口都能调用到，事实上这就是“隐式接口实现”

那么“显示接口实现”是什么呢？

interface IAnimal

{

Void Dog();

}

Class Animal:IAnimal

{

Void IAnimal.Dog()

{

Console.WriteLine(“…”);

}

}

// 只能通过接口调用

IAnimal animal = new Animal();

animal.Dog();

对于显示接口实现只运行接口实现。不能像上面一样使用类的方法去实现了。

显示接口实现的应用场景：

在实际项目中，有时某个类往往会继承多个接口，而接口中往往会有一些相同名称、参数与类型的值。通过显示接口实现可以避免一些不必要的歧义。