Interface in C

An Interface is defined as a contract with syntactical contract that all Interface inheritance classes should follow. Interface defines what part of the covenant and inheritance classes define which part of the covenant.

The interface in C # is defined as a syntactical contract that all Interface inheritance classes should follow. The interface defines the " **What** " section of the contract and the derived classes define the " **How** " section of the contract.

Interface defines attributes, methods and events, which are members of that Interface. The Interface contains only declarations of these members. Defining members is the responsibility of the derived class. It often helps provide a standard Structure that derived classes should follow.

The abstract classes to some extent usually serve the same purpose, however, they are mainly used when there are only a few methods declared by the base class and the derived class implements function.

Declaring Interface in C

To declare the Interface in C # we use the `interface` keyword. It is similar to class declaration. By default, the Interface command is public. This is the Interface declaration example in C #:

```
 public interface GiaoDich { //các thành viên c?
a interface void  xemGiaoDich (); double  laySoLuong (); }
```

For example:

The following example will illustrate the Interface implementation above:

```
 using System . Collections . Generic ; using System . Linq ; using System . Tex
?
a interface void  xemGiaoDich (); double  laySoLuong (); } public class GiaoDich
?p GiaoDich k? th?a t?
 QTMGiaoDich public GiaoDich () {  ma_giao_dich  = " " ;  ngay_giao_dich  = " "
?ch: {0}" ,  ma_giao_dich ); Console . WriteLine ( "Ngày giao d?
ch: {0}" ,  ngay_giao_dich ); Console . WriteLine ( "S? l??
ng: {0}" ,  laySoLuong ()); } } class Tester { static void Main ( string []  args
```

Running the above code we get the following result:

```
  Transaction: 001
  Transaction date: 08/06/2018
  Quantity: 789000
  Transaction: 002
  Transaction date: 09/06/2018
```

```
Quantity: 4519000
```

According to Tutorialspoint

Previous article: Operator overloading in C #

Next lesson: Namespace in C #