

اینترفیس و کاربرد آن در برنامه نویسی

اینترفیس‌ها شامل مجموعه‌ای از متدها، خواص و رویدادها هستند که کلاس‌ها ملزم به پیاده‌سازی آن‌ها می‌باشند، بدون اینکه پیاده‌سازی خاصی را تعیین کنند. این ویژگی به توسعه‌دهندگان اجازه می‌دهد تا کدهایی منعطف و قابل گسترش ایجاد کنند، زیرا کلاس‌های مختلف می‌توانند از یک اینترفیس مشترک پیروی کنند و در نتیجه امکان استفاده از چندشکلی (Polymorphism) فراهم می‌شود.

مثال : با توجه به اینترفیس داده شده کلاس **Student** را طوری پیاده سازی نماید تا نام و نمره دانشجویی را با استفاده از دو متد **GetName** , **GetMark** بازگرداند.

```
C# IStudent.cs > ...  
interface IStudent{  
    string GetName();  
    double GetMark();  
}
```

```
C# Student.cs > ...  
class Student:IStudent{  
    private string name="Farhad";  
    private double mark=19.5;  
    public string GetName(){  
return this.name;}  
    public double GetMark(){  
return this.mark;}  
}
```

مثال : با توجه به **interface** داده شده کلاس **Test** را طوری بنویسید که با استفاده از متدی بنام **generate** ده عدد تصادفی دورقمی تولید و در لیستی بنام **ls** که بصورت **private** در آن تعریف شده قرار داده و با استفاده از متد **shownums** داده های لیست را در خروجی نمایش دهد.

```
C# ITest.cs > ...  
interface ITest{  
    void generate();  
    void shownums();  
}
```

```
C# Test.cs > ...  
class Test{  
    private List<int> ls=new List<int>();  
    public void generate(){  
        Random r=new Random();  
        for(int i=0;i<10;i++){  
            this.ls.Add(r.Next(10,100));}  
    public void shownums(){  
        foreach(int n in ls){  
            Console.WriteLine(n);}  
    }  
}
```