

第56章 インターフェイス その1



インターフェイスとは、クラスで実装すべきメソッド、プロパティ、インデックス、イベントなどを示し、それ自身では一切実装しない抽象的な存在です。

インターフェイスの宣言は次のように行います。

```
interface インターフェイス名
{
    戻り値の型 メソッド名 (パラメータリスト);
    データ型 プロパティの名前 {get;set;}
    データ型 this[データ型 インデックス] {get;set;}
    ...
}
```

インターフェイスの名前は好きにつけてよいのですが、慣習で大文字のIから始まる名前にする人が多いようです。

インターフェイスのメンバは全部自動的にpublicとなります。(アクセス修飾子を自分でつけることはできない)

また、staticなメンバも定義することはできません。

インターフェイスは、フィールドを持つことはできません。コンストラクタやデストラクタも持つことはできません。

interface自体にpublic修飾子をつけると、このインターフェイスは無条件であらゆるクラスに実装できることになります。

クラスにインターフェイスを実装するには、次のようにします。

```
class クラス名 : インターフェイス名
{
    ... (実装)
}
```

この場合、インターフェイス中のすべてのメンバを実装しなくてははいけません。

複数のインターフェイスを実装することも可能です。この場合カンマ(,)で区切ります。

```
class クラス名 : インターフェイス1, インターフェイス2, ...
```

どうして、インターフェイスのようなものが必要なのでしょう。これは、あらかじめクラスがどのように振る舞うべきかを定めておくことにより、プログラミングの方向性を指し示す指針となるのです。

難しいことはさておき、インターフェイスを作って、実際にこれを実装してみましょう。

```
// interface01.cs

using System;

public interface IMyInterface
{
    double Pulus(double a, double b);
}

class MyClass : IMyInterface
{
    public double Pulus(double x, double y)
    {
        return x + y;
    }
}

class interface01
{
    public static void Main()
    {
        MyClass mc = new MyClass();
        string strFormat = "{0} + {1} = {2}";

        Console.WriteLine(strFormat, 100, 200, mc.Pulus(100, 200));
        Console.WriteLine(strFormat, 10.5, 12.5, mc.Pulus(10.5, 12.5));
    }
}
```

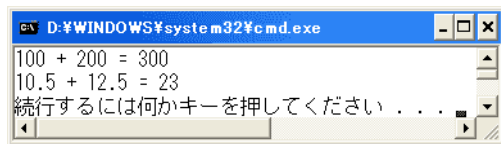
インターフェイスIMyInterfaceのメンバは、double型の引数を2つ持ち戻り値がdouble型のメソッドPulusを持ちます。

これを見るだけで、IMyInterfaceを実装するクラスは、double型の演算をするクラスなのだなー、と想像できます。

MyClassクラスではIMyInterfaceインターフェイスを実装します。

Mainメソッドでは、MyClassのインスタンスを生成して足し算の計算をさせていますね。

実行結果は次のようになります。



[\[C# Index\]](#) [\[総合Index\]](#) [\[Previous Chapter\]](#) [\[Next Chapter\]](#)

Update 01/Oct/2006 By Y.Kumei

当ホーム・ページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。