第60章 インターフェース　その5

複数のインターフェースを実装するとき、シグニチャが同じメソッドなどが存在したらどうなるのでしょうか。エラーになるのでしょうか。

実験してみるのが手っ取り早いですね。

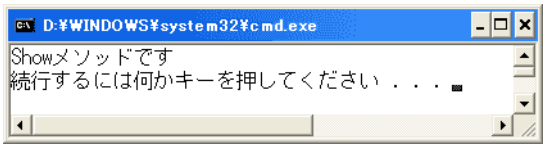
```
// interface05.cs

using System;

public interface IInterface1
{
    void Show();
}

public interface IInterface2
{
    void Show();
}

class MyClass : IInterface1, IInterface2
{
    public void Show()
    {
        Console.WriteLine("Showメソッドです");
    }
}

class interface05
{
    public static void Main()
    {
        MyClass mc = new MyClass();

        mc.Show();
    }
}
```

IInterface1もIInterface2もメンバは全く同じシグニチャを持つメソッドです。

MyClassは、この2つのインターフェースを実装しています。

これをコンパイルしても、コンパイラは何も文句を言いません。

実行結果は次のようになります。



このように、インターフェース間で同じシグニチャを持つメンバが存在したとき、曖昧さをなくすために、明示的な実装(Explicit interface implementation)を行うことができます。メソッドの場合は

インターフェース名 .メソッド名 (パラメータリスト)
{実装内容}

インターフェースを明示的に実装すると、そのクラスの参照からはアクセスできなくなります。さらに、publicなどのアクセス修飾子をつけることができなくなります。

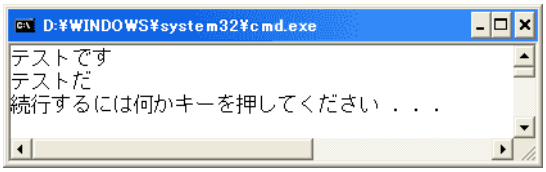
じゃあ、どうやってアクセスするの?

と、いうことになりますが、これはインターフェース参照変数を使います。この変数に、実装したクラスの参照を代入することになります。

```
// interface06.cs

using System;

public interface IInterface1
{
    void Show(string s);
}

public interface IInterface2
```

```
{
    void Show(string s);
}

class MyClass : IInterface1, IInterface2
{
    void IInterface1.Show(string s)
    {
        Console.WriteLine(s + "です");
    }

    void IInterface2.Show(string s)
    {
        Console.WriteLine(s + "だ");
    }
}

class interface06
{
    public static void Main()
    {
        MyClass mc = new MyClass();

        IInterface1 i1 = mc;

        i1.Show("テスト");

        IInterface2 i2 = mc;

        i2.Show("テスト");

    }
}
```

実行結果は次のようになります。



どのバージョンの実装が実行されるかは、どのインターフェースの型に実装クラスの 参照を代入したかによって決まります。

インターフェースの参照変数には、それを実装したクラスの参照を代入できることが わかりました。実は、これはすでに暗黙のうちに第54章で行っていました。

```
public string ToString (
 string format,
 IFormatProvider provider
)
```

DateTime構造体のToStringメソッドで、providerはIFormatProviderインターフェースの 参照変数です。したがって、これを実装したCultureInfoクラスの参照を代入できるのですね。

せっかく覚えた明示的実装ですが、なるべく避けてくださいとMSDNに書いてあります。(セキュリティ面で問題有り)

[C# Index] [総合Index] [Previous Chapter] [Next Chapter]