

## 第43章 仮想プロパティと仮想インデクサ



プロパティやインデクサもvirtual宣言して、継承先でoverrideすることができます。

基本クラスの参照変数に、継承されたクラスの参照を代入すると代入した参照のバージョンのプロパティやインデクサを呼び出すことができます。これもメソッドの時と同じです。

仮想プロパティの例を見てみましょう。

```
// virtualprop01.cs

using System;

class MyClass1
{
    int xx;

    public virtual int x
    {
        get
        {
            //Console.WriteLine("MyClass1");
            return xx;
        }
        set
        {
            xx = value;
        }
    }
}

class MyClass2 : MyClass1
{
    int yy;

    public override int x
    {
        get
        {
            return yy;
        }
        set
        {
            yy = value;
        }
    }
}

class virtualprop01
{
    public static void Main()
    {
        MyClass1 mc1 = new MyClass1();
        MyClass2 mc2 = new MyClass2();

        mc1.x = 10;
        mc2.x = 100;
        Console.WriteLine("mc1.x = {0}", mc1.x);
        Console.WriteLine("mc2.x = {0}", mc2.x);

        MyClass1 mcx;

        mcx = mc1;
        Console.WriteLine("mcx.x = {0}", mcx.x);

        mcx = mc2;
        Console.WriteLine("mcx.x = {0}", mcx.x);
    }
}
```

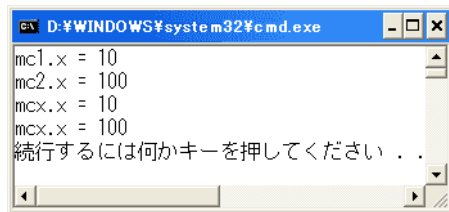
MyClass1では、プロパティxがvirtualで宣言されています。

MyClass2では、プロパティxをオーバーライドしています。

Main関数ではまず、MyClass1やMyClass2のインスタンスを生成して普通にxを利用して 値の代入をし、これを読み出しています。

次に、基本クラスのMyClass1の参照変数mcxを用意し、これにMyClass1やMyClass2の参照を代入し、mcx.xでどのバージョンのプロパティが呼ばれたかを確認しています。

実行結果は次のようになります。



```
D:\WINDOWS\system32\cmd.exe
mc1.x = 10
mc2.x = 100
mcx.x = 10
mcx.x = 100
続行するには何かキーを押してください...
```

mcx.xは、mcxに代入されている参照の型によって異なる値を返す点に注意してください。

インデクサについても、全く同様です。

```
// virtualindexer01.cs
```

```
using System;
```

```
class MyClass1
```

```
{
    string[] str = new string[2];

    public virtual string this[int n]
    {
        get
        {
            return str[n];
        }
        set
        {
            str[n] = value;
        }
    }
}
```

```
class MyClass2 : MyClass1
```

```
{
    string[] arr = new string[2];

    public override string this[int n]
    {
        get
        {
            return arr[n];
        }
        set
        {
            arr[n] = value;
        }
    }
}
```

```
class virtualindexer01
```

```
{
    public static void Main()
    {
        MyClass1 mc1 = new MyClass1();
        MyClass2 mc2 = new MyClass2();

        mc1[0] = "MyClass1";
        mc1[1] = "です";

        mc2[0] = "MyClass2";
        mc2[1] = "でございます";

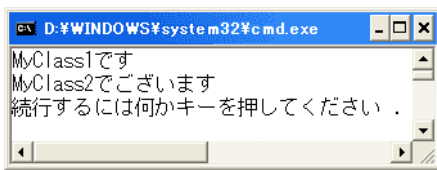
        MyClass1 mcx;

        mcx = mc1;
        Console.Write(mcx[0]);
        Console.WriteLine(mcx[1]);

        mcx = mc2;
```

```
        Console.Write(mcx[0]);  
        Console.WriteLine(mcx[1]);  
    }  
}
```

実行結果は次のようになります。



---

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

*Update 18/Sep/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。**