第31章 プロパティとインデクサをもう少し詳しくみてみる

何とも長たらしい表題です。前章および前々章で大急ぎでプロパティとインデクサについて見てきました。この章では、もう少し詳しくこれらについてみてみます。さらに、フィールドとメソッドの関係にも少し触れます。

プロパティやインデクサはstaticにすることができるのでしょうか。結論から言うと、プロパティはOKですが、インデクサはNOです。

さらに、staticなフィールドに対してインスタンスメソッドはアクセスできるのでしょうか。結論はできます。しかし、逆にインスタンスフィールドに対してstaticなメソッドはアクセスできません。

サンプルを次に示します。

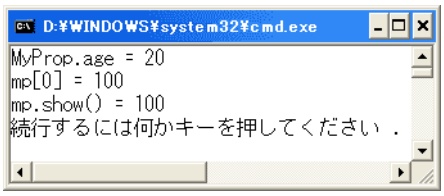
```
// staticprop01.cs

using System;

class MyProp
{
    static int x;
    static int[] y = new int[10];
    public static int z;

    // プロパティはstaticにできる
    public static int age
    {
        get
        {
            return x;
        }
        set
        {
            x = value;
        }
    }

    //インデクサはstaticにできない
    public int this[int index]
    {
        get
        {
            return y[index];
        }
        set
        {
            y[index] = value;
        }
    }

    // インスタンスメソッドはstaticなフィールドにアクセスできる
    // 逆にインスタンスフィールドにstaticなメソッドはアクセスできない
    public int show()
    {
        return z;
    }
}

class staticprop01
{
    public static void Main()
    {
        MyProp.age = 20;
        Console.WriteLine("MyProp.age = {0}", MyProp.age);

        MyProp mp = new MyProp();
        mp[0] = 100;
        Console.WriteLine("mp[0] = {0}", mp[0]);

        MyProp.z = 100;
        Console.WriteLine("mp.show() = {0}", mp.show());
    }
}
```

プロパティはstaticにできます。しかし、このプロパティに関連したフィールドもstaticで有る必要があります。

これは、当たり前といえば当たり前です。staticなメンバはそのクラス全部に共通の性質や行動を表しています。staticなプロパティでインスタンスフィールドの値を設定できるとしたら矛盾が有りますね。

メソッドとフィールドの関係についても似ています。staticなフィールドをインスタンスメソッドでアクセスできます。たとえば、クラス全体の性質を有るインスタンスから、変更したいこともあるでしょう。しかし、インスタンスフィールドをstaticなメソッドで変更できたら、これはおかしいですね。

では、実行結果を見てみましょう。



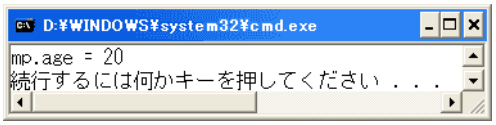
次に、インデクサのように関連したフィールドの無いプロパティというのは許されるのでしょうか。結論は大丈夫です。ただし、setアクセッサは使えません。(setされても格納するフィールドとが無いので当たり前)

では、関連するフィールドを持たないプロパティの例を見てみましょう。

```
// nofieldprop01.cs

using System;

class MyProp
{
    public int age
    {
        get
        {
            return 20;
        }
    }
}

class nofield01
{
    public static void Main()
    {
        MyProp mp = new MyProp();

        Console.WriteLine("mp.age = {0}", mp.age);
    }
}
```

ageプロパティは関連するフィールドがありません。getアクセッサではいつも20を返しています(年齢がいつまでも20歳という意味です)。

実行結果は次のようになります。



次に、getやsetアクセッサ内で勝手に変数を宣言して使ってもよいのでしょうか。これは使えます。

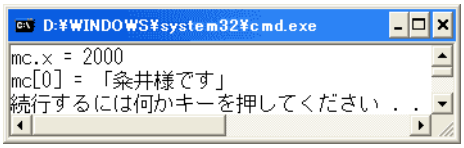
サンプルを次に示します。

```
// valprop01.cs

using System;

class MyClass
{
    int n = 2;
    string[] arr = new string[3];

    public int x
    {
        get
        {
            int val; //アクセッサ内で変数の宣言可能
            val = n * 10;
            return val;
        }
```

```
        set
        {
            int y;
            y = value * 20;
            n = y;
        }
    }

    public string this[int n]
    {
        get
        {
            string add = "です";
            return arr[n] + add;
        }
        set
        {
            string sama;
            sama = value + "様";
            arr[n] = sama;
        }
    }
}

class valprop01
{
    public static void Main()
    {
        MyClass mc = new MyClass();

        mc.x = 10;
        Console.WriteLine("mc.x = {0}", mc.x);

        mc[0] = "粂井";
        Console.WriteLine("mc[0] = 「{0}」", mc[0]);
    }
}
```

このプログラムを実行するとどうなるか、想像してみてください。

実行結果は次のようになります。思っていたとおりになったでしょうか。



プロパティでも、インデクサでもget,setアクセッサ内で変数を宣言できます。

しかし、そのすぐ外側では宣言を許されていません。

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

Update 06/Sep/2006 By Y.Kumei