

## 第29章 プロパティ



プロパティについては、今までも暗黙のうちに使っていました。使う側にとっては、フィールドだと思って使っても差し支え有りませんね。

まずは、次の簡単なプログラムを見てください。

```
// prop01.cs

using System;

class MyClass
{
    int x;

    public void setvalue(int y)
    {
        x = y;
    }
    public void show()
    {
        Console.WriteLine("x = {0}");
    }
}

class prop01
{
    public static void Main()
    {
        MyClass mc = new MyClass();

        mc.setvalue(100);
        mc.show();
    }
}
```

MyClassクラスには、privateなインスタンスフィールドxがあります。

xはprivateなので、クラス外からはアクセスできません。

そこで、xに値を設定するsetvalueメソッドを設置しました。xの値を見るための showメソッドも用意しました。

これらの、メソッドは単にフィールドに値を設定したり、値を読み出したりしている だけです。このような場合、わざわざメソッドを用意したりせずに、もっと簡単な方法があります。それには、プロパティを使います。プロパティは次のように宣言します。

### データ型 名前

```
{
    get
    {
        return フィールド名;
    }
    set
    {
        フィールド名 = value;
    }
}
```

getとか、setのことをアクセッサ(accessor)といいます。ユーザーがプロパティの名前を指定して、値を取得しようとしてきたときgetアクセッサが働きます。つまり、return フィールド名;でわかるようにフィールドの値を返します。

ユーザーが値を格納しようとしてきたときは、setアクセッサが働きフィールドに値を格納します。valueというのは、キーワードでユーザーが設定しようとしている値が格納されています。

getやsetアクセッサは常に両方無いとダメかという、そんなことはありません。getアクセッサのみであれば読み取り専用、setアクセッサのみであれば書き込み専用となります。

アクセッサを利用して、書き込み、読み取り制限もできます。たとえば、有るフィールドには負の値を設定してもらっては困る場合、ユーザーが負の値を指定してきたら、これを無視すればよいわけです。

**int x; // privateなインスタンスフィールド**

```
public int myx //プロパティは通常クラス外からアクセスされるのでpublic
{
    get
    {
        return x;
    }
}
```

```

    set
    {
        if (value >= 0)
            x = value;
    }
}

```

このようにすれば、ユーザーがプロパティを介してxに負の値を設定することはできません。また、xはprivateなのでクラス外から直接値を設定することもできません。

このように、フィールドに対して安全な値のみを格納することが可能となります。

次のプログラムは、今の説明を実現したものです。フィールドxは年齢を格納するものとして、ユーザーが負の値を格納できないようにしています。

```

// prop02.cs

using System;

class MyProp
{
    int x;

    public int age
    {
        get
        {
            return x;
        }
        set
        {
            if (value >= 0)
                x = value;
        }
    }
}

class prop02
{
    public static void Main()
    {
        MyProp mp = new MyProp();
        mp.age = 20;

        Console.WriteLine("年齢は{0}歳です", mp.age);

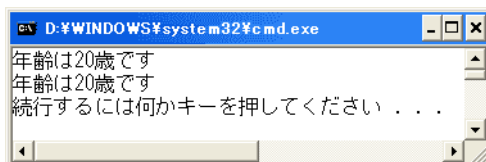
        mp.age = -20;

        Console.WriteLine("年齢は{0}歳です", mp.age);
    }
}

```

Mainメソッドでmp.age = -20;とわざとに負の値を格納しようとしています。

では、実行結果を見てみましょう。



mp.age = -20;を実行後も、「年齢は20歳です」と表示されますね。

また、使う側から見るとあたかもageというフィールドに値を設定したり、読み出したりするのと同じ感覚ですね。

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

Update 04/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。