

第35章 ジャグ配列風インデクサ



今までの配列は、たとえば2次元であればどの行の列も同じでした。これを行ごとに列の 数を変えることができます。これをジャグ配列といいます。(ジャグ(jag)とはギザギザという意味です。衣服のかぎ裂きもjagといいます。)

この章では、ジャグ配列風インデクサを作ってみます。

まず、最初にジャグ配列の作り方を2次元で考えてみます。

通常の2次元配列は

```
データ型[,] 名前 = new データ型[要素数,要素数];
```

のように作りました。ジャグ配列では(1次元の配列)の配列という考え方で

```
データ型[][] 名前 = new [要素数][];
```

と宣言しておき、

```
名前[0] = new データ型[要素数];  
名前[1] = new データ型[要素数];  
.....
```

のように、各行をそれぞれnewしていきます。これだと、行ごとに要素数を変えることができますね。また、配列の表記方法に十分注意してください。

まずは、ジャグ配列のサンプルを見てみましょう。

```
// jag01.cs  
  
using System;  
  
class jag01  
{  
    public static void Main()  
    {  
        string[][] name = new string[3][];  
        int i, j;  
  
        name[0] = new string[2]{"田中", "佐藤"};  
        name[1] = new string[3]{"横井", "吉田", "工藤"};  
        name[2] = new string[1]{"太田"};  
  
        for (i = 0; i < 3; i++)  
            foreach (string str in name[i])  
                Console.WriteLine(str);  
    }  
}
```

配列の要素の表示には、for文とforeach文を組み合わせて使っている点に注意してください。

実行結果は次のようになります。

```
C:\WINDOWS\system32\cmd.exe  
田中  
佐藤  
横井  
吉田  
工藤  
太田  
続行するには何かキーを押してください . . .
```

インデクサは、`public int this[int a][int b]`のように宣言できません。

そこで、無理矢理ジャグ配列風インデクサを作ってみます。(あまり意味はありません)

```
// jag02.cs  
  
using System;  
  
class MyClass  
{  
    string[][] name;  
    public int i;  
  
    public int len
```

```

    {
        get
        {
            return name[i].Length;
        }
    }

    public string this[int a]
    {
        get
        {
            return name[i][a];
        }
        set
        {
            name[i][a] = value;
        }
    }

    public MyClass()
    {
        i = 0;
        name = new string[3][];
        name[0] = new string[2];
        name[1] = new string[3];
        name[2] = new string[2];
    }
}

class jag02
{
    public static void Main()
    {
        MyClass mc = new MyClass();

        mc.i = 0;
        mc[0] = "佐藤";
        mc[1] = "田中";
        mc.i = 1;
        mc[0] = "太田";
        mc[1] = "工藤";
        mc[2] = "吉田";
        mc.i = 2;
        mc[0] = "加藤";
        mc[1] = "伊藤";

        for (int i = 0; i < 3; i++)
        {
            mc.i = i;
            for (int j = 0; j < mc.len; j++)
                Console.WriteLine(mc[j]);
        }
    }
}

```

まずは、MyClassクラスの

```
public string this[int a]{...}
```

のところを見てください。

「これ、普通のインデクサじゃないの」

そうです。でも、get,setアクセッサを見てください。

name[i][a]を返したり、これに値を設定したりしていますね。

iは何かというと、このクラスのインスタンスフィールドです。

コンストラクタを見ると、

```

name = new string[3][];
name[0] = new string[2];
name[1] = new string[3];
name[2] = new string[2];

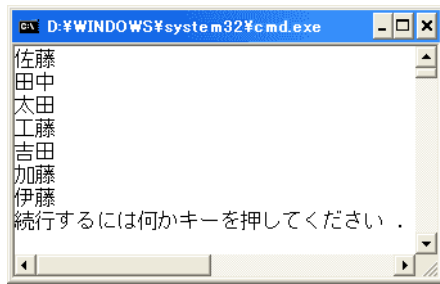
```

と、ようにジャグ配列を作っています。結局インデクサでは (name[i])の配列に対して値を返したり、設定したりしているわけです。ですから、このインデクサを使う前にiの値を設定し

ておく必要があります。また、name[j]の配列の長さをプロパティlenで返しています。

要するに、このプログラムのインデクサはごく普通のインデクサです。

実行結果を見てみましょう。



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

Update 10/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。