

特集
1-2

Visual Basic 6.0
ユーザーのための

Visual C#入門

Visual C#の プログラミング書法

これなら安心 C#言語構文完全マスター

日向 俊二
HYUGA, Shunji

Level



Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:
 - Visual Studio .NET 2003
 - Visual Basic 6.0

Samples

はじめに

あるプログラミング言語をマスターするためには、プログラムの書き方を一通り知っておく必要があります。C#をマスターするためにはC#のプログラムの書法を覚える必要があります。

C#の構文には、たとえば、変数やクラスの宣言のための構文、実行するプログラムを選択する構文であるif文、繰り返しのためのfor文などがあって、これらはVisual Basic 6.0の構文に似ていますが、C#とVisual Basicでは少し異なる場合があります。また、C#独自のキーワードもあって、そのうちのいくつかはC#のプログラミングを始めるためにぜひとも知っておきたいキーワードです。

ここでは、C#のプログラムを理解するために知っておきたいC#のプログラムの基本的な書き方と、C#のキーワードのうち、Visual Basic 6.0のプログラマが特に注意を払いたい主なキーワードについて解説します^[注1]。

宣言

最初に、変数の宣言方法から見てみましょう。

Visual Basicでは変数を宣言するときにDimステートメントを使います、一方、C#では、型と宣言する変数の名前だけで変数を宣言できます。たとえば、double型で名前がheightという変数を宣言するときには次のようにします。

```
double height;
```

行の最後に「; (セミコロン)」が付いている点に注目してください。C#では原則として文 (ステートメント) の最後に「;」を付けます。この変数宣言は最後に「;」が付いているので、C#ではひとつの文として認識されます (宣言文)。

「;」で文が終了するという決まりはC#のプログラムの重要な性質です

注1) 本稿はC#の言語仕様を解説するものではありません。C#の言語仕様については、<http://www.microsoft.com/japan/msdn/library/default.asp?url=/japan/msdn/library/ja/csspec/html/CSharpSpecStart.asp>を参照してください。

ら、ここで覚えておいてください。

次の例のように、同じ型の複数の変数を一度に宣言することもできます。

```
double height, weight, BMI;
```

変数を宣言するのと同時に値を代入することもできます(リスト1)。これを初期化といいます。

リスト1の最後の行の「new」は、クラスのインスタンス(オブジェクト)を作成するためのキーワードです。C#では大文字/小文字が厳密に区別されるので、この場合の「new」は小文字でなければなりません^[注2]。

さらに、次の例のようにfor文の中で変数を宣言して使うこともできます。

```
for (int i=0; i<10; i++)
    x += i;
```

このようなfor文のカッコの中の最初の式(この場合は「int i=0」)を初期化式といい、真ん中の式(「i<10」)を条件式、最後の式(「i++」)を反復式といいます。for文の中で変数を宣言したいときには初期化式として記述します。そして、このようにしてfor文の中で宣言

注2) C#のキーワードの先頭の文字はすべて小文字です。また、Microsoftの命名ガイドラインによると、C#のインスタンス変数名は小文字で、メソッド名は大文字で始めることになっています。しかし、BMIのような頭字語をインスタンス変数だからといって小文字で書くかどうかはプログラマ次第でしょう。なお、プロパティは先頭の文字を大文字にします。

リスト1：初期化の例

```
int x = 0;
System.Drawing.Graphics g = e.Graphics;
System.Drawing.Pen p =
    new System.Drawing.Pen(
        System.Drawing.Color.Black);
```

した変数は、for文の中だけで有効です。

次の例のように複数の変数を宣言して使うこともできます。

```
for (int i=0, j=0; i<10; i++, j++)
    x += i * j * 2;
```

初期化式の「int i=0, j=0;」や反復式の「i++, j++」の中の「,」を、一般にカンマ演算子といいます。C#のfor文の初期化式や反復式は、カンマでいくつでも繋ぐことができます。ですから、たとえば次のようにしてもかまいません。

```
for (int x=0, y=0, z=100, Alpha=1;
     x<100;
     x++, y++, Alpha += 2)
```

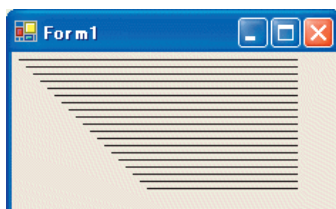
これらの書き方を上手に使うと、Visual Basicでは冗長になりがちなプログラムを、C#では簡潔に記述することができます。そのひとつの例をリスト2に示します。図1はリスト2の実行画面です。

配列

配列は同じ型の複数の要素からなるデータ構造です。

Visual Basicでは、配列はDimステートメントを使って宣言します。たとえば、Visual Basic 6.0で要素が6個の配列を宣言するときには、次のようにし

図1：リスト2で描いた図形



ます^[注3]。

```
Dim v(5) As Integer
```

C#では、配列であることを宣言するために「[]」と「[]」を使います。しかし、「[]」と「[]」を使って宣言しただけでは、配列として使うことはできません。キーワードnewを使って配列のための領域を確保する必要があります。

```
int[] v = new int [5];
```

こうして作成した配列の要素には、添え字(インデックス)で参照することができます。

```
for (int i=0; i<5; i++)
    v[i] = i;
```

また、キーワード「foreach」を使って各要素に同じコードを適用することができます。次の例は配列「v[]」の中の各要素を文字列に変換して結合します。

```
foreach (int i in v)
    s += i.ToString();
```

多次元配列の宣言には「,」を使いません。

```
int[,] v2d;
```

宣言のときには、配列の次元の数より「1少ない,」を指定するだけで、サイズを指定しない点に注目してください。配列のサイズは、newを使って実際に配列のためのメモリを確保するときに指定します。

注3) Visual Basic 6.0では配列の添え字(インデックス)はデフォルトでゼロから始まり、Option Base 1を宣言すると添え字は1から始まります。一方、Visual Basic .NETやC#では配列の添え字は常にゼロから始まります。