



# .NET Framework だからわかる！

## 多言語で味わう.NETクラスライブラリ

吉田 弘一郎 YOSHIDA, Koichiro

### 多言語対応の.NET

.NETのなにが楽しいかというと、プログラミングを行っていると、まるで多種多様なクラス満載の大型ショッピングセンターでお買い得品を漁る、

という感じを受ける点です。

あのクラスにこのオブジェクト、目移りしないほうがどうかしてます。ここで、気になるのはプログラム言語です。Javaの場合にはJavaしかなかったのですが、.NETになるとさまざまな言語が使えます。ですから、自分はVBユ

ーザーであるからC#のクラスは気にしなくてもよいのではないかというような誤解も生じかねません。ここで、.NET用であれば、言語の差を気にしなくてもよいのです。いや、言語の差を気にしなくてよいどころか、自由に混在して使えるという状況を、簡単な例題で示して、言語間の壁を取り払いたいと思います。その上で、プログラムの実行時間の測定に用いるStop Watchクラスと、書式付プリントに便利なPrintfクラスをご紹介しますことにします。前者は「かなり見つけにくいルーチン」を利用したものであり、後者は「C言語のルーチンをC#から呼び出す」実例になっています。ともに、本稿のために作ったものです。

#### 本稿で前提となるもの

- OS Windows 2000 Professional (SP3) 以降
- 開発環境 Visual Studio.NET
- .NET Framework 1.0.3705.288 (SP2)
- Visual Basic .NET Compiler 7.00.9466
- Internet Explorer 6.0.2800.1106
- Visual J#



この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥F01\_01フォルダ以下に収録しています

¥CODE1：4つの言語でクラスを作り、それを3つの言語によるアプリケーションで使用するサンプル

¥CODE2：独自クラスライブラリをC#で使用するコマンドラインアプリケーション

### .NET4大言語

現段階で、マイクロソフト社からは

次の4つの.NET対応言語が出ており、すべてVisual Studio.NET環境で使えます。これらを“.NET4大言語”と呼ぶことにしましょう。

- Visual Basic
- Visual C#
- Visual C++
- Visual J#

これ以外にも.NETで使えるプログラム言語は、APL、COBOL、Eiffel、Forth、Fortranなどなど、16種類にもおよびます<sup>[注1]</sup>。また、Borland社のDelphiやC++ Builderも.NET対応になりつつあります。

一方、Java仮想マシン (JVM) で利用できるプログラム言語にも、次のWebページにあるように、数多くものがあります。

<http://grunge.cs.tu-berlin.de/~talk/vmlanguages.html>

それでは、これらのJVM用言語をもって「Javaも多言語に対応している」と言うことができるのでしょうか？残念なことに、これらのJVM用非Java言語の多くは多分に「趣味的」ないしは「学術的」なもので、実際にそれらが活用されたケースというのは非常に少ないのではないかと思います。また、Java陣営そのものが、Java以外の言語に冷淡であるという現実もあります。

注1) .NETで使えるプログラム言語：http://msdn.microsoft.com/vstudio/partners/language/default.asp

Javaとは対照的に、.NETの場合は言語の選択に関しては非常に開放的で、非常に基礎的なレベルから多言語環境であることに対応しています。データ型とサブルーチン呼び出しの互換性はもちろんのこと、最適化や例外処理までもが互換になっています。ですから、言語そのものからくる制約はありません。

それを実証するために、本稿では先にあげた4つの言語で同じクラスを作り、それらをVisual Basic.NET、C#、J#の3つの言語で使ってみることにします。

## クラスライブラリの作成

「Speak」という関数メソッドをもつ非常に簡単なクラスを、.NET4大言語で作成します。

### Visual Basic.NETによるクラス作成

オブジェクト指向が少々不得手だったVisual Basicですが、.NETになって非常に得意になりました。この記事の例題を作りながら実感したのが、

「Visual Basicでは、細かいことを気にすることなく.NETプログラミングができる」

という事実です。決してVisual Basic派ではない私を感じたのですから、これ

リスト1：VBclass.vb

```
Public Class VBclass
    Public Function Speak() As String
        Speak = "Visual Basic is best."
    End Function
End Class
```

は本当です。実際、意外な感じがしました。

それでは、リスト1をご覧ください。「Speak」というメンバ関数(あるいは関数メソッド)が“Visual Basic is best.”という文字列を返します。実に簡単に、説明の必要はないでしょう。

ここで、クラスの作り方の手順を述べておきます。

Visual Studio.NETを起動して[新しいプロジェクト] ボタンをクリックし、プロジェクトの種類は「Visual Studioソリューション」を、テンプレートは「空のソリューション」を選択します(図1)。プロジェクト名は任意でかまいません。この空のプロジェクト内に、まずは「Visual Basic.NETのクラスライブラリのプロジェクト」を新規に作成してゆきます。

ソリューションエクスプローラ内の「ソリューション」アイコンを右クリックし、「追加」 - 「新しいプロジェクト」を選択します。「新しいプロジェクトの追加」ダイアログボックスが表示されたら、プロジェクトの種類に「Visual Basicプロジェクト」、テンプレートに「クラスライブラリ」を選択し、名前を付けます(図2)。

ここでは、そのプロジェクトの名前を「VB\_ClassLibrary」としました。「OK」ボタンをクリックすると、VB\_ClassLibrary.vbというファイルが生成され、その中に“中身が空の骨格プログラム”である「Class1.vb」が収まります。せっかくだからクラスの名前も「VBclass」としておきましょう(図3)。プロパティの「ファイル名」を「VBclass.vb」に変更し、コードエディタ内