

初音 玲
HATSUNE, Akira

コードから見える .NETとJavaの微妙な関係

.NETでJavaする方法、伝授します

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
 - Visual J# .NET

Level



Samples

この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥F04ディレクトリに収録しています。

¥0311VJ01

VJ#.NETフォームサンプル

¥0311VJ02

VJ#.NETクラスサンプル

¥0311VJ03

VJ#.NET複数プロジェクトサンプル

¥0311VJ04

VJ#.NETライブラリサンプル

¥0311VBVJ

VB.NETからVJ#.NETライブラリを利用するサンプル

はじめに

マイクロソフト社とJavaの関係はいつも微妙だ。.NET Framework上で動作するJava言語系であるVisual J#.NET (以下VJ#.NET) も最初に発売されたVisual Studio .NETには含まれておらず、Visual Studio .NET 2003から含まれるようになったという経緯がある。

JavaからみたVJ#.NET

VJ#.NETは、Java言語系であってJava処理系ではない。一体どんな違いがあるのか戸惑うかもしれないが、まずは図1をみてほしい。

本来、Javaは、JDKに含まれているコンパイラにより中間言語 (Javaバイトコード) にコンパイルされ、その中間言語がさまざまなOS用JRE (ランタイムとJavaVMなど) 上で共通して動作するまでを含めてJavaと言う。つまり、Java開発環境から実行環境までのトータルの処理系がJavaなのだ。

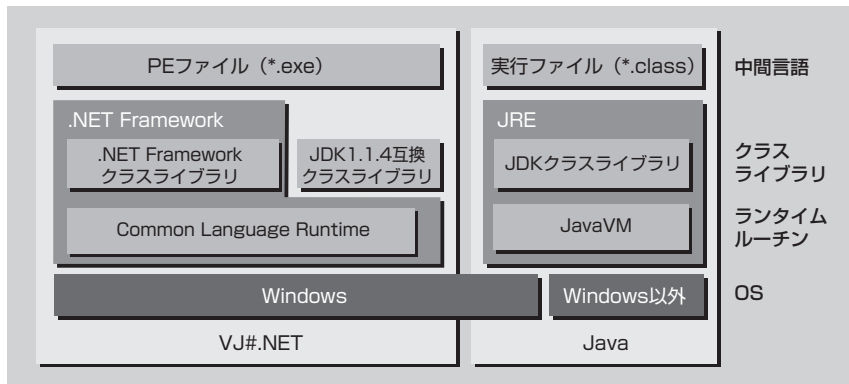
では、VJ#.NETはどうか。VJ#.NETには、JDK1.1.4に含まれているランタイムルーチンと互換性のあるランタイムルーチンが付属している。このため、VJ#.NETは、JDK1.1.4相当のコードを記述してコンパイルすることができる。しかしながら、そのコンパイル結果は、MSIL (Microsoft Intermediate Language: 中間言語) であり、JRE上で動作するのではなく、.NET Framework上で動作するので、厳密な意味ではJavaとは言えないのだ。

さらに、VC++.NETは、MFCやATLなどを使って、.NET Frameworkから切り離されたアンマネージコードな実行ファイルを生成できるが、今のところVJ#.NETでJavaバイトコードを生成する方法はない。

VJ#.NETの基本文法

VJ#.NETの基本文法は、Java言語そのものだ。しかし、実際にコードを書いてゆくと問題が発生することがある。それは、標準クラスライブラリの違い

図1: VJ#.NETとJavaの違い



だ。Javaの場合、JREに含まれているクラスライブラリが基本となるが、このクラスライブラリと.NET Frameworkクラスライブラリの間には互換性がない。さらに、互換性がないにもかかわらず文法的には同じため、たとえば、変数型などは非常に混乱を招くような状態になっている。

変数型の違い

たとえば、倍精度浮動小数点型の変数を宣言する場合、

```
Double dblAns;
```

と記述すると、「java.lang.Double」というJavaの倍精度浮動小数点型として宣言されるし^[注1]、

```
double dblAns;
```

と記述すると、「System.Double」というCLR (共通言語ランタイム) の倍精度浮動小数点型構造として宣言され

注1) コンパイル時に警告メッセージが出力される。

る。これは、Doubleだけではなく他の変数型についても同様だ。

CLRとの親和性

変数型のように人の目には同じ意味に見えるものでも、使われるクラスライブラリが別になってしまうというパターン以外にも注意が必要ことがある。それは、JDK1.1.4互換ライブラリとCLRとの親和性だ。.NET Frameworkクラスライブラリとの使い勝手の違いは、「JDK1.1.4互換ライブラリで考慮してCLRを使っている」ということにつながる。それだったら、素直に.NET Frameworkクラスライブラリを選択したほうが良いと思うのは、手元にJavaで書いた大規模コードがないからだろうか。

Windowsアプリケーションの作成

VJ#.NETの開発環境は、Visual Basicで定評のあったIDEという統合開発環境の進化系であるMicrosoft Development Environment (MDE) だ。

MDEの起動

Windowsの「スタート」メニューからVisual Studio .NET 2003を起動すると、「Microsoft Development Environment [デザイン]」画面が立ち上がる。

新しいプロジェクト

MDEのスタートページで、「新しいプロジェクト」ボタンをクリックし、「新しいプロジェクト」ダイアログから「Windowsアプリケーション」を選択すると、自動的にフォーム (画面) をひとつ持ったアセンブリ (実行ファイル) のプロジェクトが作成できる (図2)。フォームの名前は、デフォルトのForm1.jslから「0311VJ01.jsl」に変更した。

アセンブリの実行

この時点で、メニューから [デバッグ] - [開始] を選択すると、内部的に図3のような流れで、ソースコードからネイティブコードへと変換が行なわれて、フォームだけのアプリケーションが起動する。

そのときのコードはリスト1のようになっていて、ポイントは以下の通りだ。

- ①名前空間を指定するためにpackage宣言を行なう
- ②import宣言により、いくつかのクラスライブラリを単純名で指定できるようにして、コードをみやすくする
- ③extends指定でFormクラスを継承して、frm0311VJ01を定義する
- ④初期化ルーチンを呼び出すPublic関