

けん太の プログラミン 修行記



第7回

「けん太、名前空間について考える」の巻 - その1 -

名前空間の使い方

碗仔 けん太 (Pochi Company)
WANKO, Kenta

Visual Basic.NETでは、「System.Windows.Forms.Form」とか「System.Drawing」のような、ドット(.)で連結された名前をたくさん使います。このような長い名前の全体または一部は、「名前空間」というものだそうです。また、Visual Basic.NETで頻繁に使うようになったXMLにも、名前空間というものが使われているそうです。今回は最近良く耳にし目にもする名前空間について調べてみます。



前回までの あらすじ

ポチ・カンパニーのゲーム「困ったワンワン」^[注1]のプログラムを、Visual Basic.NET用に移植するという仕事を任されたボクは、Visual Basic 6.0のプログラムをVisual Basic.NETに移植するために必要なことをいろいろ調べてきました。「移植後のプログラムが以前より効率的できれいなプログラムになるようにすること」という条件が付いているので、Visual Basic.NETの特徴

を活かしたプログラムにするためにいろいろ調べた結果、.NET環境でよりよく機能するプログラムを作るためには、Visual Basic.NETのオブジェクト指向プログラミングの側面について詳しく知る必要があることに気がきました。

オブジェクト指向プログラミングの キーポイント

オブジェクト指向プログラミングでは、さまざまなクラスから作成したインスタンス、つまり、オブジェクトを使います。オブジェクトを使うということ自体は、それほど難しいことではないような気がします。しかし、実際に作業に取りかかってみると、カンタンではないことに遭遇してしまうのです。

オブジェクト指向プログラミングのキーポイントは何なのでしょう？あ

本稿で前提となるもの

OS Windows 2000 Professional (SP3)
開発環境 Visual Basic.NET
Microsoft Development Environment 2002 Version 7.0.9486
.NET Framework 1.0 Version 1.0.3705 (SP1)




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¥KENTAフォルダ以下に収録しています


¥SPACES：今回のサンプルプロジェクト


注1) 本稿に出てくる会社や登場人物およびここで取り上げているゲームプログラム「困ったワンワン」などはフィクションであり、実在する製品、会社、人物、犬種、そのほかモロモロあれこれアレコレやらとはいっさい関係がありません。念のため。


るとき、そんな疑問をもって迷子になりそうなほど大きなカンファレンス会場でポチさんの後にくっついて歩いていると、ふと、気付いたことがあります。オブジェクトにはなぜか「名前空間」が付いて歩くらしい、ということです。


 ポチさん、オブジェクト指向プログラミングでは、なぜいつも名前空間という言葉が出てくるんですか？

 こんなところで、突然なんだい？

 ですから、オブジェクト指向プログラミングと名前空間にはどういう関係があるんですか？

 勉強熱心だね。だけど、こんな人ごみの中でそんな大問題を簡単に説明できると思う？

 思いません。

 じゃ、自分で調べてごらん。ついだから、Visual Basic.NETの名前空間だけじゃなくて、名前空間というものの全般について調べてみるといいよ。

というわけで、今回は名前空間というものについて、Visual Basic.NETだけに限らず、C#やマネージドC++、XMLなどを含めて広く調べてみることにします。



名前空間とは

付録CD-ROMにも収録されているポチのプログラミング用語集で「名前空間」を調べたところ、次のようなことがわかりました。

- 名前空間は、ネームスペース (Name space)、ネーム空間ともいう
- 名前空間は、名前 (識別子) を区別するものこと
- 名前空間が異なれば、同じ名前の識別子を定義して、それぞれ別のものとして識別できる
- .NET Frameworkのクラスライブラリにあるクラスは、どのクラスもすべて種類ごとに階層構造をもつ名前空間に定義されている
- Visual Basic.NET、C#、マネージドC++などCLR (共通言語ランタイム) 上で実行されるプログラムは、同じ名前空間の同じクラスを使う
- XMLでも名前空間が使われ、異なる仕様やボキャブラリを識別することができる

これを読んだだけでも、それなりにわかったような気になりました。でも、万全を期すためにVisual Studio.NETのドキュメントで名前空間についてさらに調べた結果、次のようなことがわかりました。

名前空間の宣言方法

名前空間をVisual Basic.NETで宣言するには、次のようにキーワードName spaceを使って「Namespace 名前空間名」と「End Namespace」で囲みます。

```
Namespace AruName
```

```
End Namespace
```

そして、この中に次のように定義したクラスMyClassは、この名前空間AruNameのものになるそうです。

```
Namespace AruName
```

```
Public Class MyClass
```

```
End Class
```

```
End Namespace
```

したがって、他のコードからこのクラスにアクセスするにはAruName.MyClassでできるというわけです。名前空間の考え方そのものは、とても単純ですね。

整理してみると、

- 名前空間を宣言するときには、キーワード「Namespace」^[注2]を使って宣言する
- 名前空間の中にクラスを定義するときには、「Namespace 名前空間名」と「End Namespace」の間にクラスを宣言する

名前空間の使い方も、難しくはありません。

- ある名前空間に定義したクラスを使うときには、「名前空間名.クラス」の形式で使う

素朴な疑問

名前空間の定義の仕方と使い方はわかりました。しかし、ひとつの疑問が湧きあがってきました。それは、Visual Basic 6.0までの世界には名前空間はありませんでしたが、それでもやってこられたということです。名前空間など

注2) C#やC++では「namespace 名前空間名」と「{ }」の間にその名前空間に定義したいものを定義します。