

VB.NET

コンポーネント活用講座
～ 教養課程 ～



スモールパーツ プログラミング

第20回

小森 大輔 KOMORI, Daisuke

MainMenuコントロール、 ContextMenuコントロール — Part3 —

Technology Tools

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

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¥SMALLディレクトリに収録しています。

¥IMAGEMENUITEM
アイコンを表示するMenuItemの継承クラス

¥IMAGEMENUITEMFLAT
フラット形状のメニューを表示するクラス

本連載は、Visual Basic .NET (以下VB.NET) の標準コントロールにスポットを当て、とことん使い込んで、その機能や役割を探ってゆきます。

前回まで2回にわたって、メニューを表示するコントロールであるMainMenuの「基本的な使用方法」「MenuItemオブジェクトのオーナーローを使用したアイコンの表示」について解説してきました。

今回は応用編ということで、前回作成したアイコンを表示するMenuItemオブジェクトを、MenuItemクラスを継承する新しいクラスとして実装し、汎用性を持たせてみます。そして、OfficeやVisual Studio .NETライクなフラット形状のメニューの表示にも挑戦してみたいと思います。

MenuItemを継承する クラスの作成 ～サンプルMenuItem

まずはアイコンを表示するメニューアイテムを、MenuItemクラスを継承したクラスとして作成してゆきましょう。

“継承クラスの作成”なんて言葉だけを見ると、Visual Basic プログラマには何やら敷居が高いことのように思いますが、そんなことはありません。今回行なう作業は、次のとおり。継承クラスの作成なんて言ってもやることはこの程度です。

- ①クラスを作成
- ②InheritsステートメントでMenuItemクラスを継承
- ③必要なインターフェイスを考え、プロパティやメソッドとして実装
- ④変更して使用するMenuItemクラスのメソッドを、Overridesキーワードでオーバーライド

◆ クラスの作成とMenuItemクラスの継承

プロジェクトを新規作成し、このプロジェクトにクラスをひとつ追加します。クラス名は「ImageMenuItem」にしましょう。そして、Classステートメントでクラスが定義されているすぐ後の行に、

```
Inherits MenuItem
```

と、記述します。このコードは「これはMenuItemクラスを継承するクラスだよ」という意味になります。Inheritsステートメントを使って他のクラスを継承する場合、必ずClassステートメントのすぐ後に記述する必要があります。

次に、MenuItemクラスが標準で持つインターフェイス以外に必要なものを定義し、また、標準で持つインターフェイスの中で変更する必要があるものをオーバーライドしてゆきます。

◆ 表示するアイコンのイメージを指定するプロパティ

まずは、イメージを表示するメニューアイテムのクラスなんですから、そのイメージを指定するプロパティを持っていないければお話になりませんね。

リスト1のように、クラスのProtected変数として変数mImageを宣言し、Imageプロパティを新規に作成します。

◆ メソッドのオーバーライド

メニューアイテムを描画する前に発生するOnMeasureItemメソッドと、メニューアイテムの描画が必要なタイ

リスト1：Imageプロパティ

```
Protected mImage As Image

Public Property Image() As Image
    Get
        Return mImage
    End Get
    Set(ByVal Value As Image)
        mImage = Value
    End Set
End Property
```

ミングで発生するOnDrawItemメソッドは、オーナーロー (OwnerDraw) の“キモ”になります。したがって、この両メソッドをオーバーライドするのは必須です。

これらのメソッドをオーバーライドするには、Overridesキーワードを使用します。Overridesキーワードを指定したSubプロシージャは、「この派生クラスのメソッドは、基本クラス内の同じ名前のメソッドを書き換えています」という意味になります。ここで注意しなければならないのは、「メソッドの引数の数とデータ型が、オーバーライドする基本クラスのメソッドと完全に一致する必要がある」ということです。また、これらのメソッドをオーバーライドするときには、そのメソッドの中から基本クラスのOnMeasureItemメソッドを呼び出す必要があります。

リスト2は、これらのメソッドをオーバーライドしたコードです。

◆ Newコンストラクタ

この派生クラス (ImageMenuItem) がインスタンス化されたタイミングで、オーナーローに設定するなどの処理を行なう必要があるため、クラスの初期化を制御するNewコンストラクタにも手を加える必要があります。

しかし、このNewコンストラクタは特殊なメソッドで、他のメソッドとは異なり継承は行なわれません。ですから“手を加える”と言っても、新たに定義する形になります。試しにNewコンストラクタにOverridesキーワードを付けて定義しようとしても、「NewをOverridesとして宣言することはできない」と怒られてしまいます。

基本クラスのNewコンストラクタもそうなっています

リスト2：メソッドのオーバーライド

```
Protected Overrides Sub OnMeasureItem( _
    ByVal e As MeasureItemEventArgs)
    MyBase.OnMeasureItem(e)
End Sub

Protected Overrides Sub OnDrawItem( _
    ByVal e As DrawItemEventArgs)
    MyBase.OnDrawItem(e)
End Sub
```