Visual Basic .NETのツボ 2003

第20回 クラス作成の基礎を総まとめ —その2—

西田 雅昭
NISHIDA, Masaaki

前回からクラスによるプログラミングのまとめとして、汎用の「Human」クラスを作成しています。今回は、「Human」クラスのプロパティにデータチェックを行なう処理を追加してクラスを完成させます。

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¥TUBOディレクトリに収録しています。

¥0311
前回作成した「Human」クラスとテストプログラム

¥HUMAN2
「Human」クラス（完成版）

¥TESTHUMAN2
「Human」クラスのテスト用プログラム（完成版）

¥SHAIN

簡単なおさらい

前回、人の情報を管理する「Human」クラスを作成し、表1のようなプロパティを実装しましたが、各プロパティプロシージャのコードは以下のような単純なものでした。

```
Public Property pstrRName() As String
  Get
    Return mstrRName
  End Get
  Set(ByVal Value As String)
    mstrRName = Value
  End Set
End Property
```

繰り返しになりますが、データ処理

表1：「Human」クラスのプロパティ

プロパティ	データ型	説明
pblnkanji	Boolean	入力モードの判別（漢字：True、カナ／英字：False）
pstrFamilyName	String	姓（漢字）
mstrFirstName	String	名（漢字）
pstrFurigana	String	ふりがな
pstrRName	String	ローマ字名
pblnSex	Boolean	性別（男：True、女：False）
pdteBirth	Date	誕生日（プログラムで年齢を算出する際に利用）
pstrZip	String	郵便番号
pstrAddress1	String	住所1
pstrAddress2	String	住所2
pstrTele	String	電話番号
pstrKeitai	String	携帯番号
pstrMail	String	メールアドレス

などにおけるデータのチェックは、**ユーザーインターフェイスから切り放して、クラスの内部で行なう**のが望ましいのです。これにより、システムの管理が楽になりますし、ビジネスルールなどの訂正が1箇所で済むことになります。

そこで、このクラスがプロパティの値を受け取る際に、データをチェックし、それをこのクラスのインスタンスを処理しているルーチンに知らせることにします。そのためには、各プロパティプロシージャの「Set」文を書き変える必要があります（これについては今回説明します)。

データが正しくないことを知らせる、一番よい方法は「エラーを発生する」ことです。そのため、前回、「Incorrect PropertyException」クラスを作成しました。今回は、このクラスを踏まえた内容になるので、コードをリスト1に再掲しておきましょう。

プロパティでデータをチェックする

早速、表1のプロパティの「Set」文について、どのようにして例外を発生するかを見てゆきましょう。前回作成した「Human」クラスのプロジェクトを開いてください。

参考

前回までのコードは、付録CD-ROM（¥DOTNET¥TUBO¥0311フォルダ）に収録しています。今回から読み始めた方は、それを元にこの記事を読み進めてください。

リスト1：「IncorrectPropertyException」クラス

```
Public Class IncorrectPropertyException
   Inherits ApplicationException
   Private mstrMessage As String

   Public Sub New(ByVal strMessage As String)
      MyBase.New(strMessage)
   End Sub
End Class
```

リスト2：「ローマ字名」のデータチェック

```
Public Property pstrRName() As String
   Get
      Return mstrRName
   End Get
   Set(ByVal Value As String)
      If Value.Length > mcintRNameLength Then
         Throw New IncorrectPropertyException( _
          "ローマ字は" & mcintRNameLength.ToString & "文字以内で入力してください")
      Else
         mstrRName = Value
      End If
   End Set
End Property
```

◉ローマ字名のチェック

まずは、簡単な「ローマ字名」からプロパティプロシージャを書き変えることにします（リスト2)。「ローマ字名」では、文字数制限のチェックのみ行ないます。「Throw New IncorrectProperty Exception(…)」の部分では、例外クラスのインスタンスを作成する際にエラーメッセージの内容を指定しています。

「Throw」がエラーを発生するステートメントであることはすでにおわかりですね。以前、同じ実装の例外クラスを利用しましたが、「Throw」以降の記述がよくわからない、というご意見をいただきました。そこで、少し補足しておきます。

リスト1を見るとわかりますが、例外クラス「IncorrectPropertyException」では、親クラス「ApplicationException」の「Message」プロパティをそのまま利用しており、指定したエラーメッセージを使用してインスタンスを初期化するコンストラクタを実装しています。

```
Public Sub New(ByVal strMessage As String)
   MyBase.New(strMessage)
End Sub
```

この引数「strMessage」は、エラーを説明するメッセージです。「MyBase」は、「Me（自分自身を参照)」と違って現在のクラス（IncorrectPropertyException）の基本クラス（ApplicationException）を参照しているので、

```
Throw New IncorrectPropertyException( _
 "メッセージ")
```

という記述が可能となるわけです。