

西田雅昭  
の

新

# Visual Basic!

西田 雅昭  
NISHIDA, Masaaki

基本からの  
業務アプリケーション構築技法

「ロボ」

## 第7回 クラスの継承



### 「clsHuman2」クラス

まずは、簡単に前回のおさらいをしておきましょう。

前回は、「MyClassLibrary」名前空間に「clsHuman2」というクラスを作成しました(図1)。

この「clsHuman2」クラスでは、以下の5つの属性(プロパティ)とメソッド、イベントを定義しています。( )内

は、プロパティ名とデータ型です。

- ・ 姓 (pstrFamilyName / String)
- ・ 名 (pstrFirstName / String)
- ・ ふりがな (pstrfurigana / String)
- ・ 誕生日 (pdteBirthday / Date)
- ・ 年収 (plngIncome / Long)
- ・ 「fncAge」メソッド：現時点での年齢を返す
- ・ 「FutureBirth」イベント：「誕生日」に当日より後の日付を入力すると発生する

このようにクラスで定義したプロパティ、メソッド、イベントは、クラスの「メンバ」と呼びます。

今回は、このクラスを使用して新しいクラスを作成してみます。

なお、今回のサンプルプログラムには前回のサンプルプログラムが含まれていますので、本誌Webサイトからサンプルプログラムをダウンロードして、実際に作りながら読み進めてください。

レベル >>> Level

1 2 3 4 5

サンプル >>> Sample

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、<http://www.shoeisha.com/mag/windev/>からダウンロード可能です。

言語 >>> Language

Visual Basic

ツール >>> Tool

Visual Studio 2005 Professional

\* ) 本連載ではVisual Studio 2005 Professionalを使用していますが、解説内容は特に明記しない限り、Visual Basic 2005 Expressでも利用できます。Express Editionをお持ちの方もぜひ挑戦してみてください。

図1: 「clsHuman2」クラスの動作テストプログラム

「Human」クラスのテスト

姓	西田
名	雅昭
フリガナ	ニシダ マサアキ
誕生日	1970/01/01
年収	¥1,000,000
年齢	36

書き込み    クリア    読み込み    終了

## ● クラスの継承

今回作成するのは、企業の社員を表わす「clsEmployee」というクラスです。このクラスの属性は、仮に、

- ・ 氏名
- ・ ふりがな
- ・ 部課名
- ・ 役職
- ・ 誕生日
- ・ 給与
- ・ ボーナス

ということにしましょう。

この属性は、前回作成した「clsHuman2」クラスによく似ています。「ふりがな」「誕生日」はまったく同じですし、「氏名」は「姓」と「名」を使用し作成できそうです。

「clsHuman2」クラスのソースコードをコピー／修正して新たに「clsEmployee」クラスを作るというもひとつの手ですが、もう少し楽な方法はないでしょうか。

Visual Basicでは「継承」という便利な機能がサポートされています。

これは、あるクラスを継承して別のクラスを作成すると、継承元のクラスのプロパティ、メソッド、イベントをそのまま受け継ぐことができるという機能です。継承元のクラスを「基本クラス」、継承して作成したクラスを「派生クラス」と呼びます。

この機能を使えば、基本クラスで定義したメンバは、派生クラスで新たに

定義しなくても使えるのです。

ここで簡単な例を見てみましょう。「点 (clsPoint)」というクラスが、

- ・ Xプロパティ (X座標)
- ・ Yプロパティ (Y座標)
- ・ Colorプロパティ (色)
- ・ Paintメソッド (点を描く)

というメンバを持っているとします。たとえば、この「点」クラスを継承して「線 (clsLine)」という派生クラスを作成する場合、すでに上記4つのメンバがあるので、新たに定義するのは、

- ・ Radianプロパティ (角度)
- ・ Lengthプロパティ (長さ)

という2つのメンバだけで済みます。

派生クラスでは、「Inherits」ステートメントを使って、どの基本クラスから派生しているかを宣言します。次のようなコードを記述すれば、「点」クラスを継承し、計5つのメンバを持つ「線」クラスを作成することができます。

```
Public Class clsLine
    Inherits clsPoint

    ' 2つのプロパティを定義
    ' (Radian / Length)

End Class
```

## ● オーバーライド

この継承という機能を使えば、簡単に「clsHuman2」クラスから「clsEm

リスト1: 「pdteBirthday」Property プロシージャ

```
Private mdteBirthday As Date

Public Property pdteBirthday() As Date
    Get
        Return mdteBirthday
    End Get
    Set(ByVal value As Date)
        If value > DateTime.Today Then
            RaiseEvent FutureBirth(value)
        Else
            mdteBirthday = value
        End If
    End Set
End Property
```

\*) 前回紹介した「pdteBirthday」Property プロシージャでは、「RaiseEvent FutureBirth(mdteBirthday)」と記述していましたが、これは「RaiseEvent FutureBirth(Value)」の誤りです。ここで訂正させていただきます。

ployee」クラスが作成できそうです。

しかしここでひとつ問題があります。「clsHuman2」クラスの「pdteBirthday」プロパティでは「誕生日」が未来の日付だったときだけチェックしていますが、「clsEmployee」クラスでは15歳未満と70歳以上のときにチェックしたいのです。そのため、「pdteBirthday」プロパティの処理を変更しなければなりません。

Visual Basicでは、基本クラスから継承したプロパティやメソッドの処理を派生クラスで変更する必要がある場合、派生クラスでそれらの処理を定義し直すことができます。この機能を「オーバーライド」と呼びます。

ただし、プロパティやメソッドをオーバーライドするには、基本クラスのプロパティやメソッドを「Overridable」修飾子を付けて宣言しなければなりません。

ここで前回作成した「MyClassLibrary」プロジェクトを開いてみましょう。「pdteBirthday」プロパティの定義は、リスト1のようになっています。