

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

## 最終回 コレクション-その2-



### コレクションの実験

今回は、前回に引き続き、コレクションについて説明していきます。

コレクションに関して、どうしても紹介しておきたい概念があります。それはVisual Studio 2005で導入された「ジェネリック (Generics)」という機能です。

今回は、「clsHuman」というクラス

を作成し、さらにこの「clsHuman」のコレクションを作成しましたが、今回は、話を簡単にするために、もっと単純なコレクションで実験を行いたいと思います。



### 実験プログラムの作成

実験のために図1のようなフォームを作成してください。プロパティは表1

図1：  
実験用フォーム



表1：フォームに配置したコントロールのプロパティ設定

コントロール	プロパティ	値
Button1	(Name)	btnAdd
	Text	Add
Button2	(Name)	btnList
	Text	List
Button3	(Name)	btnPickUp
	Text	PickUp
Button4	(Name)	btnPickUpS
	Text	PickUpS
Button5	(Name)	btnAddG
	Text	AddG
Button6	(Name)	btnListG
	Text	ListG
Button7	(Name)	btnPickUpG
	Text	PickUpG
Button8	(Name)	btnQuit
	Text	Quit
ListBox1	(Name)	lstKazu
Label1	(Name)	lblKazu
	BackColor	White

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。



リスト1：コレクション要素の追加

```
Private Sub btnAdd_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnAdd.Click
    With mcolAList
        .Add(10)
        .Add(11)
        .Add(12)
    End With
End Sub
```

リスト2：コレクション要素の読み出し

```
Private Sub btnList_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnList.Click
    With lstKazu.Items
        For Each intKazu As Integer In mcolAList
            .Add(intKazu)
        Next
    End With
End Sub
```

のとおりです。

最初に、宣言部に「ArrayList」コレクションをひとつ宣言しておきます。

```
Public Class Form1
    Private mcolAList As New ArrayList
```

### コレクション要素の追加と読み出し

次に、[Add] ボタンの「Click」イベントハンドラを記述します（リスト1）。このコードについては、前回説明しましたね。単純に整数のコレクションに要素を3つ作成しているのです。

続いて、前回と同じく [List] ボタンをクリックすると、リストボックスに要素を表示することにします（リスト2）。

[Quit] ボタンはプログラムを終了させるだけです。

```
Private Sub btnQuit_Click(略) _
    Handles btnQuit.Click
    Me.Close()
End Sub
```

これでプログラムを実行し、[Add] [List] の2つのボタンを順番にクリックすると、リストボックスに3つの要素

を表示します（図2）。

ここまでは、何の問題もありませんね。

### 2番目の要素をラベルに表示する

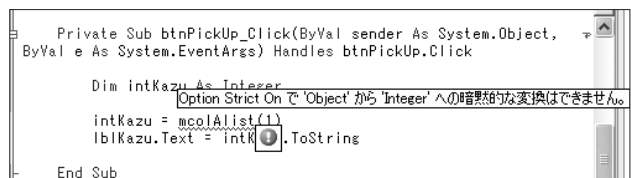
次に、コレクション内の2番目の要素を下のラベルに表示することにして、[PickUp] ボタンの「Click」イベントハンドラを記述します。

```
Dim intKazu As Integer

intKazu = mcolAList(1)
lblKazu.Text = intKazu.ToString
```

「mcolAList(1)」の下に青色の波線が

図3：エラーの内容を確認

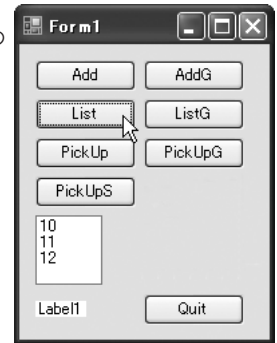


リスト3：コレクションの2番目の要素を読み出す

```
Private Sub btnPickUp_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnPickUp.Click
    Dim intKazu As Integer

    intKazu = DirectCast(mcolAList(1), Integer)
    lblKazu.Text = intKazu.ToString
End Sub
```

図2：  
[List] ボタンの  
実行結果



出ています。そこをマウスでポイントすると、

「Option Strict On で 'Object' から 'Integer' への暗黙的な変換はできません。」

というエラーであることがわかります（図3）。

この「mcolAList」コレクションの要素はObject型だったので、

そこで、

```
intKazu = mcolAList(1)
```

を、

```
intKazu = DirectCast(mcolAList(1), Integer)
```

と書き換えると、エラーが消えました（リスト3）。