怒濤のTips101
最後の大逆襲

覚えておくと超便利なワザモノ
手軽に使えるカンタンTips

From VBPJ
訳：有限会社 風工舎

VB Tech Tips 101

Enumで大文字小文字の区別を強制する

VB3 VB4 VB5 VB6 VBA

列挙定数はすばらしいものだが、ちょっと不快な癖がある。統合開発環境（IDE）で大文字小文字の区別が保持されないのだ。多くの人が定数名のスペルミスの有無を目で見て確認するためにIDEを利用している。幸い、IDEを騙して大文字小文字の区別を保持させることができる。すなわち、Enumをパブリック変数として“も”宣言したうえで、それらがコンパイルされないようにコンパイラディレクティブの“#If False…#End If”で囲むという方法だ。

```
Public Enum MyEnum
   EnumOne=1
   EnumTwo
   EnumThree
End Enum
#If False Then
   Public EnumOne
   Public EnumTwo
   Public EnumThree
#End If
```

—Barry Garvin, Georgetown, Massachusetts

プロジェクトウィンドウにファイルをドラッグする

VB3 VB4 VB5 VB6 VBA

VB6（およびそれ以前のバージョン）で新規プロジェクトを作成する際、大きなモジュールライブラリを含めるのが大変だった。「ファイルの追加」ダイアログは複数選択をサポートしていないので、一度にひとつずつモジュールを追加することになるからだ。VBPファイルを手動で編集することもできるが、いずれにしても面倒な作業だ。

しかし、この方法を試してほしい。VBを開いてから、エクスプローラのウィンドウを開く。複数のBAS、FRM、またはCLS

モジュールを反転表示（選択）し、それらをVBのプロジェクトウィンドウにドラッグするだけだ。これで、単一選択の悪夢に悩まされることはなくなる。すべてのファイルが即座にプロジェクトに追加される。

—Darin Higgins, Fort Worth, Texas

StrReverseを使って最後に出現した文字を見つける

VB3 VB4 VB5 **VB6** VBA

長い文字列の中で特定の文字について最後に出現したものを探す場合、ふつうはループの中で「Mid$」や「InStr」を使う。VB6で新たに導入された文字列反転関数の「StrReverse」を使えばもっと簡単にできる。たとえば、完全に修飾されたファイルスペックには¥（円記号）がいくつか含まれるが、そのうち最後のものを見つけてファイル名だけを抽出したい場合がある。リスト1はStrReverseを使って必要な情報だけをすばやく抽出する。

同じロジックを使って、パスからファイル名を取り除いてフルパスだけを返すこともできる（リスト2）。

—Balaraman M. Sriram, Richardson, Texas

1回の代入で構造体データをクリアする

VB3 VB4 VB5 **VB6** VBA

特定の動作（ビヘイビア）をもたない構造化データを格納するときは、ユーザー定義型が役に立つ。（関連付けられた動作をもたせる場合は、データをそれ独自のクラスにカプセル化する必要がある）。リスト3を使えば、それぞれのサブ変数を設定することなく、ユーザー定義型の変数をすばやくクリアできる。

この方法は、ユーザー定義型に多数のサブ変数があるときにとくに便利である。

—Dave Doknjas, Surrey, British Columbia, Canada

すばやく簡単なキュー

VB3 VB4 VB5 **VB6** VBA

リストボックスを簡易キューとして動作させることができる。ListMyQueueという名前でリストボックスを作成し、リスト4-1でキューに追加し、リスト4-2でキューの内容を取得する。

—Eric Robuck, Lyon Station, Pennsylvania

数値の小数部分を返す

VB3 VB4 VB5 **VB6** VBA

VBには10進数の小数部分を返す関数はないが、Fix関数を使って元の値から整数部分を取り除くことで、簡単に小数部分が得られる。以下の例では、負の値を避ける

リスト1：StrReverseで必要な情報だけをすばやく抽出

```
Public Function GetFileName(ByVal FileSpec As String) As String
    Dim sRevName As String
    If Len(FileSpec) Then
        sRevName = StrReverse(FileSpec)
        GetFileName = StrReverse(Left$( _
            sRevName, InStr(1, sRevName, "¥") - 1))
    End If
End Function
```

リスト2：フルパスの取得

```
Public Function GetPath(ByVal FileSpec As String) As String
    Dim sRevName As String
    Dim sPathName As String
    If Len(FileSpec) Then
        sRevName = StrReverse(FileSpec)
        sPathName = StrReverse(Right$(sRevName, _
            Len(sRevName) - InStr(sRevName, "¥")))
        If Right$(sPathName, 1) = ":" Then
            '¥付きルートディレクトリ
            sPathName = sPathName & "¥"
        End If
        GetPath = sPathName
    End If
End Function
```

リスト3：ユーザー定義型変数をクリア

```
' ユーザー定義型を定義する
Private Type udtSomeType
    SubVariableOne As Integer
    SubVariableTwo As String
    SubVariableThree As Long
End Type
' クラスレベルのユーザー定義型の
' 変数を2つ定義する
Private TypeVariableOne As udtSomeType
Private TypeVariableTwo As udtSomeType
' クラス内のメソッド
Private Sub ResetData()
    Dim CleanTypeVariable As udtSomeType
    TypeVariableOne = CleanTypeVariable
    TypeVariableTwo = CleanTypeVariable
End Sub
```

リスト4-1：キューに追加

```
Public Sub Enqueue(StringToAdd As String)
  If Len(String_to_Add) > 0 Then
    ParentForm.ListMyQueue.AddItem StringToAdd
  End If
End Sub
```

リスト4-2：キューから取得

```
Public Function Dequeue() As Variant
    If ParentForm.ListQueue.ListCount > 0 Then
        Dequeue = ParentForm.ListQueue.List(0)
        Parent_Form.ListQueue.RemoveItem (0)
    Else
        MsgBox "Queue is Empty"
    End If
End Function
```

ために計算結果の絶対値を返すようにしてある。

```
Public Function Frac( _
 ByVal Value As Double) As Double
   Frac = Abs(Value - Fix(Value))
End Function
```

—William Powell Jr., Lanham, Maryland

コントロール配列をエラーなしで反復処理する

VB3 VB4 VB5 VB6 VBA

コントロール配列には妙なところがあり、要素の欠落を許容する。コントロール配列を反復処理する場合、最も簡単なのは次の方法だ。

```
Dim i As Integer
For i = Text1.LBound To _
      Text1.UBound ...
```

しかし、配列内に穴がある場合には、この方法ではエラーが発生する。それを避けるために、配列をコレクションのように扱う。

```
Dim txt As TextBox
For Each txt In Text1
   txt.Text = "My Index is " _
            & txt.Index
Next txt
```

—Guy Dafny, Tel Aviv, Israel

API を使わずに地域の小数点文字を取得する

VB3 VB4 VB5 VB6 VBA

次の関数を使うと、地域の設定から10進数の小数点のためのシンボルを読み取ることができる。

```
Sub Form_Load()
  Dim DecS As String
  DecS = _
    ReadDecimalSymbol()
End Sub

Function ReadDecimalSymbol() _
   As String
  ReadDecimalSymbol = _
    Mid$(CStr(1.1), 2, 1)
End Function
```

—Gianfranco Callino, Verona, Italy

リストビューでラジオボタンを使う

VB3 VB4 VB5 VB6 VBA

簡単なコードで、リストビュー（ListView）コントロール内のチェックボックスをラジオボタンのように動作させることができる。ListViewのCheckboxesプロパティをTrueに設定し、ItemCheckイベントプロシージャに次のコードを記述する。

ユーザーがリスト項目のひとつをチェックするたびに、前にチェックされていた項目のチェックが外される。

```
Private Sub ListView1_ItemCheck( _
 ByVal Item As MSComctlLib.ListItem)
   Dim li As MSComctlLib.ListItem
   For Each li In ListView1.ListItems
      If li.Checked = True Then
         If li <> Item Then _
          li.Checked = False
      End If
   Next li
End Sub
```

—James D. Murray, Huntington Beach, California

リソース文字列をテキストファイルに出力する

VB3 VB4 32 VB5 VB6 VBA

リソースエディタ（Resource Editor）アドインは、リソースファイルのテキストエントリを追加するのに便利だが、現在の内容を出力する機能はない。リスト5をプロジェクトモジュールに追加すれば、必要なドキュメントを生成することができる。

必要なリソースファイル番号の範囲をテキストファイルに抽出するには、イミディエイトウィンドウを表示してDumpResStringsを呼び出し、適切なパラメータを渡す。

```
Call DumpResStrings(1, 2999, _
      "resdat.txt")
```

リソースファイルのドキュメント化が終了したら、標準のアプリケーション開発用

リスト5：リソース文字列をテキストファイルに出力

```
Public Sub DumpResStrings(Start As Long, _
                          Finish As Long, _
                          FileSpec As String)
   Dim hFile As Long
   Dim sText As String
   Dim i As Long
   On Error Resume Next
   hFile = FreeFile
   Open FileSpec For Output As #hFile
   For i = Start To Finish
      sText = LoadResString(i)
      If Err.Number = 0 Then
         Print #hFile, i & vbTab & sText
      Else
         Err.Clear
      End If
   Next i
   Close #hFile
End Sub
```

にこの関数をプライベートに設定する。

—*Trevor Marr, Chessington, Surrey, England*

短いファイル名を長いファイル名に変換する

VB3 VB4 32 VB5 VB6 VBA

Dir関数を使って長いファイル名を返すことはできるが、それにはパス情報は含まれない。与えられた短いパス／ファイル名をいったんその構成ディレクトリに分解することで、APIの助けを借りずに、32ビット版のVBでDir関数を使って長いパス／ファイル名を作成することができる。

見てのとおり、リスト6はドライブに基づく完全に修飾された標準のファイルスペックを前提にしている。

—*Alex Leyfman, Brooklyn, New York*

リストビューを昇順または降順に並べ替える

VB3 VB4 32 VB5 VB6 VBA

リスト7は、Windowsエクスプローラや Outlookなど多くの市販アプリケーションで見られるリストビュー（ListView）コントロールに関して標準の列の並べ替えを行なう。このルーチンを使用すると、ユーザーが列をクリックするたびに、リストビューが自動的に並べ替えられる。同じ列をクリックすると、並べ替え順序が昇順と降順の間で切り替わる。このルーチンをリストビューコントロールのColumnClickイベントプロシージャから呼び出すには、リストビューへの参照と、呼び出し元のイベントに渡されたColumnHeaderへの参照の両方を渡す。

—*Jim Pragit, Glen Ellyn, Illinois*

アルファベット文字だけであるかどうかを検査する—その1

VB3 VB4 VB5 VB6 VBA

VBにはIsNumeric関数はあるが、IsAlpha関数はない。このリスト8を使用すれば、文字または文字列がアルファベット文字（A-ZまたはA-z）であるかどうかを判定できる。ハイフン、アポストロフィなど、その他の文字で有効としたいものがある場合

リスト6：短いファイル名を長いファイル名に

```
Public Function GetLongFilename(ByVal sShortName As String) As String
    Dim sLongName As String
    Dim sTemp As String
    Dim iSlashPos As Integer
    ' InStrが失敗しないように短い名前に“¥”を付ける
    sShortName = sShortName & "¥"
    ' “[ドライブ文字]:¥”の部分を無視するために
    ' 4文字目から始める
    iSlashPos = InStr(4, sShortName, "¥")
    ' 変換のために“¥”文字の間の文字列を取り出す
    Do While iSlashPos
        sTemp = Dir(Left$(sShortName, _
            iSlashPos - 1), vbNormal Or vbHidden _
            Or vbSystem Or vbDirectory)
        If sTemp = "" Then
            ' Error 52 - Bad File Name or Number
            GetLongFilename = ""
            Exit Function
        End If
        sLongName = sLongName & "¥" & sTemp
        iSlashPos = InStr(iSlashPos + 1, sShortName, "¥")
    Loop
    ' ドライブ文字を先頭に付加する
    GetLongFilename = Left$(sShortName, 2) & sLongName
End Function
' この関数を使う場合は、次の行を追加する
GetLongFilename("C:¥PROGRA~1¥COMMON~1")
```

リスト7：リストビューのソート

```
Public Sub ListView_ColumnClick( _
        ByRef MyListView As ListView, _
        ByVal ColumnHeader As ColumnHeader)
    With MyListView
        .Sorted = False
        If .SortKey <> ColumnHeader.Index - 1 Then
            .SortKey = ColumnHeader.Index - 1
            .SortOrder = lvwAscending
        Else
            If .SortOrder = lvwAscending Then
                .SortOrder = lvwDescending
            Else
                .SortOrder = lvwAscending
            End If
        End If
        .Sorted = True
    End With
End Sub
```