

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14回

小森 大輔 KOMORI, Daisuke

List View コントロール — Part 2 —

～ アイテムを並べ替えるための5つの手法

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。

¥SORT1
Sortingプロパティを使用した並べ替え

¥SORT2
バブルソートを使用した並べ替え

¥SORT3
ListViewItemSorterを使用した並べ替え

¥SORT4
カスタムプロシージャを使用した文字列の比較

¥SORT5
Windows APIを使用した文字列の比較

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みます。

前号は「ListViewコントロールの列ヘッダーのクリック時に、通常はアイテム並べ替えを行いませんよ」と言ったところで終わってしまったのでしたよね。今回はこのアイテムの並べ替えに絞ってお話しようと思います。

列ヘッダークリック時のイベント



前号でもお話ししたように、HeaderStyleプロパティを「ColumnHeaderStyle.Clickable」に設定したListViewコントロールは、列ヘッダーをクリックした際に「ListView_ColumnClick」というイベントが発生します。

通常、列ヘッダーがクリックさ

れたときには、クリックされた列をキーにしてアイテムが並べ替えられるのが自然ですが、ListViewコントロール自身がこの処理を行ってくれるわけではありません。このイベントが発生したときの処理は、開発者が独自に実装しなければなりません。

“汎用性を高くするため”というのにはわからなくもありませんが、プロパティのオプションで並べ替えを実装してくれていても良いんじゃないのと思うのは私だけ？ 不満は残りますが、ここはぐっとこらえて、数ある並べ替えの実装方法を見てみることにしましょう。

Sortingプロパティを使用した並べ替え



アイテムを並べ替える簡単な方法として、ListViewコントロールのSortingプロパティを使用する方法

表1：SortOrder列挙体

メンバ名	内容
SortOrder.Ascending	昇順に並べ替える
SortOrder.Descending	降順に並べ替える
SortOrder.None	並べ替えを行わない

があります。このプロパティには表1のようなSortOrder列挙体の値を指定します。ただしこの表を見てわかるように、このプロパティでは、「アイテムをキーにした」昇順／降順の並べ替えしかできず、サブアイテムをキーに指定することはできません。

リスト1は、このSortingプロパティを使用してアイテムを並べ替えるサンプルです。ListView1_ColumnClick イベントの引数であるColumnClickEventArgsクラスには、クリックされた列ヘッダーの“0”から始まるインデックスを表わすColumnプロパティがあります。このプロパティの値を利用して、クリックされた列ごとに分岐した処理を行なうことができます。サンプルではクリックされた列のインデックスが“0”、すなわちアイテム列の列ヘッダーがクリックされた場合のみ、交互に昇順／降順を表わす静的フラグ変数に基づいた値をSortingプロパティに指定して、並べ替えを行ないます。

ソートアルゴリズムを使用した並べ替え



リスト1のSortingプロパティでは、アイテム列をキーとした並べ替えしか行なうことができません。では、サブアイテムがクリックされたときにその列の値に基づいて並べ替えを行なうには、どのような処理をすれば良いのでしょうか？

まず考えられるのが、「クリックされた列のすべてのサブアイテムをループで確認して、ソートルーチンにかけて並べ替えを行なう」というものです（ソートルーチンに関してはコラムを参照）。ここでは一番簡単なバブルソートを使用して、並べ替えを行なってみることにします。

アイテム文字列の比較は、Stringクラスに用意されているCompareメソッド（指定した2つのStringオブジェクトを比較する）を使用すれば良さそうです。

リスト1：Sortingプロパティを使用したアイテムの並べ替え（サンプルSort1）

```
Private Sub ListView1_ColumnClick(ByVal sender As Object, _
    ByVal e As ColumnClickEventArgs) _
    Handles ListView1.ColumnClick
    ' 静的フラグ変数を宣言
    Static blnFlg As Boolean
    ' 昇順／降順を決定するSortOrder格納変数
    Dim so As SortOrder

    ' クリックされた列がアイテム列の場合
    If e.Column = 0 Then
        ' フラグを見て昇順／降順を決定
        If blnFlg Then
            so = SortOrder.Ascending
        Else
            so = SortOrder.Descending
        End If
        ' Sortingプロパティを指定
        ListView1.Sorting = so
        ' フラグ変数を反転
        blnFlg = Not blnFlg
    End If
End Sub
```

リスト2は、アイテムを追加する際に、サブアイテム文字列にランダムな数を含めたListViewコントロールで並べ替えを行なうサンプルです。ListView1_ColumnClick イベントプロシージャからコールするカスタムプロシージャでバブルソートを使用して並べ替えを行ない、引数に並べ替えのキーとする列と、昇順／降順を表わすSortOrder型の値を指定するようにしてあります。このプロシージャ内で比較を行ない、アイテムを入れ替える

図1：バブルソートでのアイテムの並べ替え

