Learning VB.Net

Tutorial 15 – Structures

Hello everyone… welcome to vb.net tutorials. These are going to be very basic tutorials about using the language to create simple applications, hope you enjoy it. If you have any notes about it, please send them to **notes@mka-soft.com** I will be happy to answer them. Finally if you find these tutorials are useful, it would be nice from you to send a small donation via PayPal to **donation@mka-soft.com**.

Tutorial posted on 2010-April-17.

Structures

In VB.NET, you can combine a number of related variables together and treat them as one unit. This simplifies programming and makes updating the source of your applications easier. Let us first consider the following example:

You want to store the information about a person/employee. The information include name, telephone number and salary. So far we learned that to store such information you should define 3 distinct variables:

```
Dim Name As String
Dim TEL As String
Dim Sal As Decimal
```

Later on you fill these variables with values, and uses them. Now what is you have two employees? Obviously you define another 3 variables:

```
Dim Name2 As String
Dim TEL2 As String
Dim Sal2 As Decimal
```

Now what if you have a 1000 employee? Well a better solution is to use arrays. However now you need to define 3 arrays:

```
Dim Names() As String
Dim TELs() As String
Dim Sals() As Decimal
```

The first array stores the names, the second stores telephone numbers, and last one stores salary. So arrays handles the information for large amount of data pretty well. But what is you need to add another property such as address? The solution is to add another array:

```
Dim Address() As String
```

And if you need to store another property you need to store another array. If you have 14 property for an employee, then you have to store and manage 14 different arrays. In such situations structures are useful. You define a structure to combine the different properties like this:

```
Structure PersonInfo
    Dim Name As String
    Dim Tel As String
    Dim Sal As Decimal
    Dim Address As String
End Structure
```

Now you have a new data type called **PersonInfo** which contains inside it a name, a telephone, a salary and an address for that particular employee/person.

```
Dim A As PersonInfo
A.Name = "Smith"
```

```
A.Tel = "555-22-332"
A.Sal = 400
```

So **A** here is the name of the variable and it stores all the attributes or properties of employee/person. To access a specific property you use the dot (.) followed by the property. For example **A.NAME** access the name property of that employee. Now to define another employee/person you write:

```
Dim B As PersonInfo
B.Name = "Micehl"
B.Tel = "111-22-332"
B.Sal = 700
```

So now you have two employees **A** & **B**. Now if you want to define an array of such structure you can do so by:

```
Dim Info() As PersonInfo
```

And you can access the elements of the array normally

```
Dim I As Integer
Dim N As Integer

N = InputBox("Enter the number of people")
ReDim Info(0 To N - 1)

' read info here
For I = 0 To N - 1
    Info(I).Name = InputBox("enter the name of person")
    Info(I).Tel = InputBox("enter the telephone number")
    Info(I).Sal = InputBox("enter the salary")
Next
```

So as you can see structures are used the same way as normal variables do. Now if you want to copy the information of a structure then :

```
A = B
```

This will allow you to copy all the content and attributes of structure B into A. So it does not matter how many attributes you have, they will all be copied in one single step.

Now let us work on an example. Create a simple form containing two buttons and a data grid view. One of the buttons should read the data and the other should sort the data. Create the following columns in the data grid :

- Name
- TEL
- Sal
- License

Then go to the code page of the form and define the following structure:

```
' this is the structure to store person information
Structure PersonInfo
    Dim Name As String
    Dim Tel As String
    Dim Sal As Decimal
    Dim LincenseNumber As String
End Structure
```

Then define an array of structure:

```
' this is the array to store persons' info
Dim Info() As PersonInfo
```

Next create a subroutine to read the information of the array

```
' read the information and store it in an array
Public Sub ReadInfo()
    Dim I As Integer
    Dim N As Integer

    N = InputBox("Enter the number of people")
    ReDim Info(0 To N - 1)

    ' read info here
    For I = 0 To N - 1
        Info(I).Name = InputBox("enter the name of person")
        Info(I).Tel = InputBox("enter the telephone number")
        Info(I).Sal = InputBox("enter the salary")
        Info(I).LincenseNumber = InputBox("enter license number")
    Next
End Sub
```

After that add the following code to display the content of the array

```
' fill the data grid with array info
Public Sub FillDGV(ByVal DAT() As PersonInfo, ByVal DGV As DataGridView)
    DGV.Rows.Clear()
    Dim I As Integer
    For I = 0 To DAT.Length - 1
        DGV.Rows.Add(DAT(I).Name, DAT(I).Tel, DAT(I).Sal, DAT(I).LincenseNumber)
    Next
End Sub
```

Next is to add the code of the read button

```
    ReadInfo()
    FillDGV(Info, DataGridView1)
```

Next is the sort method:

```
' the sort subroutine
Public Sub Sort(ByRef Arr() As PersonInfo)
    Dim Flg As Boolean
    Dim I As Integer
    Dim Tmp As PersonInfo
    Do
        Flg = False
        For I = 0 To Arr.Length - 2
            If Arr(I).Name > Arr(I + 1).Name Then
                Tmp = Arr(I)
                Arr(I) = Arr(I + 1)
                Arr(I + 1) = Tmp
                Flg = True
            End If
        Next
    Loop While Flg
```

```
    End Sub
```

Look how simple the swap operation is. No need to swap the 4 attributes, you just use one swap operation. For the second button add the following code:

```
        Sort(Info)
        FillDGV(Info, DataGridView1)
```

So the code should be:

```
Public Class Form1

    ' this is the structure to store person information
    Structure PersonInfo
        Dim Name As String
        Dim Tel As String
        Dim Sal As Decimal
        Dim LincenseNumber As String
    End Structure

    ' this is the array to store persons' info
    Dim Info() As PersonInfo

    ' read the information and store it in an array
    Public Sub ReadInfo()
        Dim I As Integer
        Dim N As Integer

        N = InputBox("Enter the number of people")
        ReDim Info(0 To N - 1)

        ' read info here
        For I = 0 To N - 1
            Info(I).Name = InputBox("enter the name of person")
            Info(I).Tel = InputBox("enter the telephone number")
            Info(I).Sal = InputBox("enter the salary")
            Info(I).LincenseNumber = InputBox("enter license number")
        Next
    End Sub

    ' fill the data grid with array info
    Public Sub FillDGV(ByVal DAT() As PersonInfo, ByVal DGV As DataGridView)
        DGV.Rows.Clear()
        Dim I As Integer
        For I = 0 To DAT.Length - 1
            DGV.Rows.Add(DAT(I).Name, DAT(I).Tel, DAT(I).Sal, DAT(I).LincenseNumber)
        Next
    End Sub

    ' the sort subroutine
    Public Sub Sort(ByRef Arr() As PersonInfo)
        Dim Flg As Boolean
        Dim I As Integer
        Dim Tmp As PersonInfo
        Do
            Flg = False
            For I = 0 To Arr.Length - 2
                If Arr(I).Name > Arr(I + 1).Name Then
                    Tmp = Arr(I)
                    Arr(I) = Arr(I + 1)
                    Arr(I + 1) = Tmp
                    Flg = True
                End If
            Next
        Loop While Flg
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        ReadInfo()
        FillDGV(Info, DataGridView1)
    End Sub


    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        Sort(Info)
        FillDGV(Info, DataGridView1)
    End Sub
End Class
```

The source file contains a simple added code, you should try to use that to add more functionality to the example. Try to include car information.

So this is all for today. If you need the source file, you can get it from the web site. If you have notes about this tutorial, email me at: **notes@mka-soft.com**.

Thanks.

mkaatr