

Learning VB.Net

Tutorial 14 – Do Loop

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-09.

Do Loop

In VB.NET, you can use For loop which we saw how to use in one of the previous tutorials. Another type of loop is the Do loop. The format of this loop can be one of the following:

```
Do
    Statement...
    Statement...
    ...
Loop While Condition
```

Or

```
Do
    Statement...
    Statement...
    ...
Loop Until Condition
```

Or

```
Do While Condition
    Statement...
    Statement...
    ...
Loop
```

Or

```
Do Until Condition
    Statement...
    Statement...
    ...
Loop
```

The location of the condition tells when the checking of exiting or staying in the loop should be performed. For example:

```
Do While K<10
    Statement...
    Statement...
    ...
Loop
```

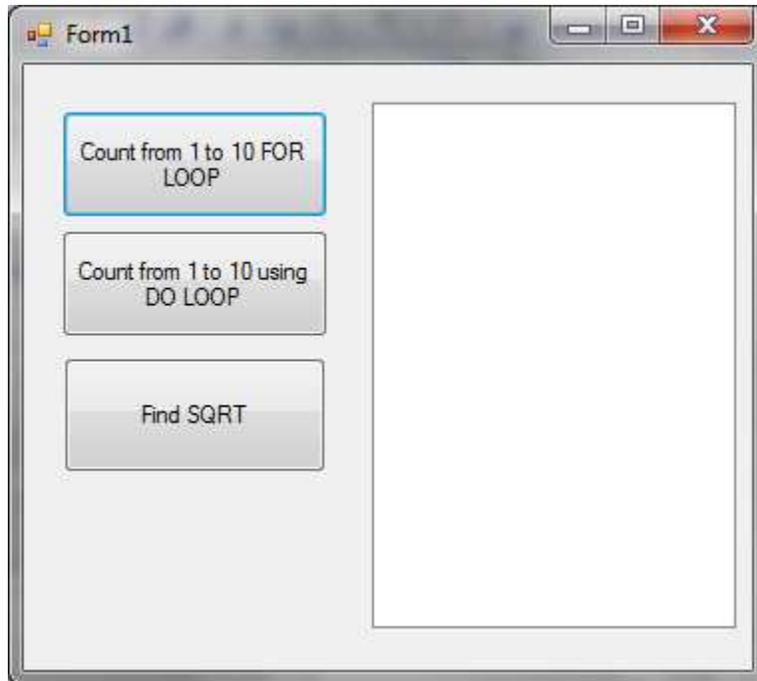
Here the check is performed every time before the loop is executed. It is possible in this example for the loop to never get executed. However if you write it like this:

```
Do
    Statement...
    Statement...
    ...
Loop While K<10
```

<http://www.mka-soft.com>

The check is performed after the loop is executed. So here the loop is executed at least once.

Now let us test the Do Loop with an example:



Create a form similar to the above picture, and then for the first button write the code:

```
Dim I As Integer
ListBox1.Items.Clear()
For I = 1 To 10
    ListBox1.Items.Add(I)
Next
```

This is the standard For loop, Now we will create a similar loop using the Do loop. Add the following code for the second button.

```
Dim I As Integer
I = 1
ListBox1.Items.Clear()
Do
    ListBox1.Items.Add(I)
    I = I + 1
Loop While (I < 11)
```

Run the application and see how both loops give the same result. The thing about the Do loop here is that you must initialize the counter (I) yourself, and increase its value, and check when you should stop the loop. This is something you don't have to do with the For loop.

The Do loop is not used to replace the For loop, but it is used when you don't know how many times you need to execute the code. For example, assume you want to find the square root for an unknown number of values, and you want to stop when you enter a negative number. The way you do it is as follows:

```
Dim I As Double
Do
    I = InputBox("Enter a +ve number:")
    If I >= 0 Then MsgBox("the root of your number is:" & Math.Sqrt(I))
Loop While I > 0
```

Here you can stop after entering one value, or after entering ... say 500 +ve numbers. Try to add the code to the third button in the form above and check it out. The full code for the form should be similar to the following:

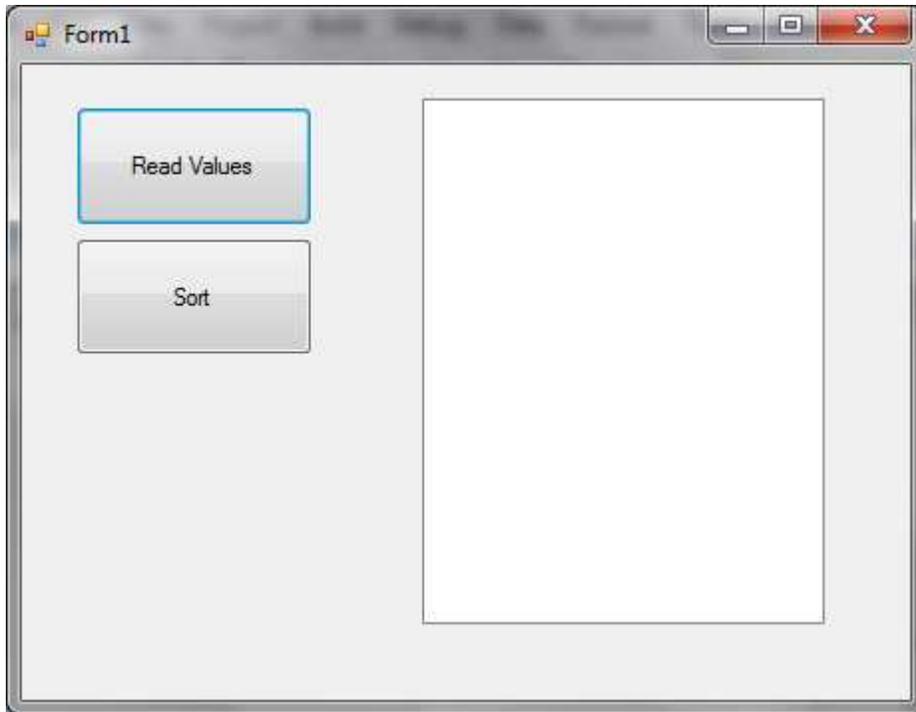
```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim I As Integer
        ListBox1.Items.Clear()
        For I = 1 To 10
            ListBox1.Items.Add(I)
        Next
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        Dim I As Integer
        I = 1
        ListBox1.Items.Clear()
        Do
            ListBox1.Items.Add(I)
            I = I + 1
        Loop While (I < 11)
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        Dim I As Double
        Do
            I = InputBox("Enter a +ve number:")
            If I >= 0 Then MsgBox("the root of your number is:" & Math.Sqrt(I))
        Loop While I > 0
    End Sub
End Class
```

Now let us try another example with the Do Loop, this time we use it to perform bubble sort. Create a form similar to the one below:



Next write down the code of the form to be as follows:

```
Public Class Form1
    Dim A() As Integer

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim Count As Integer
        Count = InputBox("enter the number of values:")
        ReDim A(0 To Count - 1)

        Dim I As Integer
        For I = 0 To Count - 1
            A(I) = InputBox("Enter the value " & I.ToString)
        Next

        ViewArray(A, ListBox1)
    End Sub

    Public Sub ViewArray(ByVal Ar() As Integer, ByVal L As ListBox)
        L.Items.Clear()
        Dim I As Integer
        For I = 0 To Ar.Length - 1
            L.Items.Add(Ar(I))
        Next
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        ' sort
        Dim I As Integer
        Dim J As Integer
        Dim Flag As Boolean

        Do
            Flag = False
```

<http://www.mka-soft.com>

```
        For I = 0 To A.Length - 2
            If A(I) > A(I + 1) Then
                Flag = True
                J = A(I)
                A(I) = A(I + 1)
                A(I + 1) = J
            End If
        Next

        Loop Until Flag = False

        ViewArray(A, ListBox1)

    End Sub
End Class
```

Check out the code for the sort operation:

```
' sort
Dim I As Integer
Dim J As Integer
Dim Flag As Boolean

Do
    Flag = False

    For I = 0 To A.Length - 2
        If A(I) > A(I + 1) Then
            Flag = True
            J = A(I)
            A(I) = A(I + 1)
            A(I + 1) = J
        End If
    Next

    Loop Until Flag = False
```

The Flag is used to tell when any two values in the array are exchanged, which tells the computer to perform another loop on array elements. Notice that this bubble sort is not very efficient, and it can be improved. However it is used here to just show you an example of when to use the Do Loop.

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