Program Example 10  Recursion 1

Objectives

**Program Purpose**

- Display a series in a list box
- Clear the display
- Adjust series start and end values
- Adjust series interval

**Learning Goals**

- To demonstrate ‘recursion’ as a method of looping
- To use procedures in the general section
- To introduce a parameter variable
- To use the ‘Val’ function to convert strings to numbers
- To use a ‘Delay’ procedure
- To use the ‘Input Box’ function
- To use multiple selection (SELECT CASE...END SELECT)

Design Notes

**Control Arrays**

In this project, we will create a control array of three text boxes. These will hold the input values for ‘Start’, ‘End’ and ‘Interval’. Use the following method:

1. Create the first text box.
2. Name the box ‘txtInput’.
3. Copy and paste the box twice, answering ‘Yes’ to the question "Do you want to create a control array?" You should notice that the text boxes are named txtInput(0), txtInput(1) and txtInput(2).

Arrays always start with 0, if not numbered otherwise.

**Parameter variables**

The values in the txtInput boxes are ‘passed’ (sent as parameters) to the 'AddNumbers' procedure. The AddNumbers procedure has three variables declared in its header. These variables are set to ‘accept’ the values from the three txtInput boxes.

**Recursion**

Recursion is the process by which a procedure calls itself. The program creates another copy of the procedure in memory until the procedure reaches its endpoint. As with indefinite loops, it is easy to accidentally create an endlessly recursive loop, using up more and more memory until the system crashes. Be careful!

**Interface**

Create the interface as shown below.
Use 4 command buttons, 6 labels and 1 listbox.
Names of Objects

<table>
<thead>
<tr>
<th>Type of Object</th>
<th>Number</th>
<th>Names of Objects</th>
<th>Simple Initial Properties of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>1</td>
<td>Form1</td>
<td>Caption – “Recursion 1”</td>
</tr>
<tr>
<td>List Box</td>
<td>1</td>
<td>picDisplay</td>
<td>Font – Bold, 12 Column - 3</td>
</tr>
<tr>
<td>Text Box</td>
<td>3</td>
<td>txtInput(0), txtInput(1), txtInput(2)</td>
<td>Font – Bold 12</td>
</tr>
<tr>
<td>Label</td>
<td>3</td>
<td>Label1, Label2, Label3</td>
<td>Font – Bold, 12 Captions – Start, End, Interval</td>
</tr>
<tr>
<td>Command Button</td>
<td>4</td>
<td>cmdExample1, cmdExample2, cmdExample3, cmdClear</td>
<td>Font – Bold, 12: Captions – Display&amp;1, Display&amp;2, &amp;Clear, E&amp;xit</td>
</tr>
</tbody>
</table>

Further Initial Properties of Objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Property</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>Startup Position</td>
<td>2 – Center Screen</td>
</tr>
<tr>
<td>hsbMax</td>
<td>Min</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>LargeChange</td>
<td>5</td>
</tr>
<tr>
<td>hsbInterval</td>
<td>Min</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>LargeChange</td>
<td>2</td>
</tr>
<tr>
<td>Label1</td>
<td>Caption</td>
<td>Interval</td>
</tr>
<tr>
<td>Label2</td>
<td>Caption</td>
<td>Maximum</td>
</tr>
<tr>
<td>lblMax</td>
<td>Caption</td>
<td>**</td>
</tr>
<tr>
<td></td>
<td>BorderStyle</td>
<td>Fixed Single</td>
</tr>
<tr>
<td>lblInterval</td>
<td>Caption</td>
<td>**</td>
</tr>
<tr>
<td></td>
<td>BorderStyle</td>
<td>Fixed Single</td>
</tr>
</tbody>
</table>

Events – Code

```vbnet
Dim x As Integer

Sub AddNumbers(x, y, z)
    'recursive procedure which adds numbers to a label
    'x is the starting number
    'y is the end number
    'z is the interval
    lstNum.AddItem x
    If x < y Then
        Delay 0.3
        AddNumbers x + z, y, z 'recursive call
    End If
End Sub

Private Sub cmdClear_Click()
    lstNum.Clear
End Sub

Sub Delay(Wait As Single)
    'delays execution of your program by the amount of
    ' time "Wait"
    Dim Start As Single
    Start = Timer 'number of seconds since midnight
    While Timer - Start < Wait
        DoEvents
        Wend
End Sub
```
Private Sub cmdExample1_Click()
' A recursively repeating loop
' Stopped by entering the word "Stop"
' Uses a recursive call inside an IF-THEN-END IF statement
Dim Msg As String
    cmdExample1.Enabled = False
    Msg = InputBox("Enter a message", "Type "Stop" to Stop", "Message")
    lstNum.AddItem Msg
    Delay 1
    If Msg <> "Stop" Then   'condition which ends loop
        cmdExample1.Value = True  'recursive call
        cmdExample1.Enabled = True
    End If
End Sub

Private Sub cmdExample2_Click()
' A indefinite pretested loop
' Stopped by entering the word "Stop"
Dim Msg As String
    cmdExample2.Enabled = False
    lstNum.Clear
    While Msg <> "Stop"   'condition which ends loop
        Msg = InputBox("Enter a message", "Type "Stop" to Stop", "Continue")
        lstNum.AddItem Msg
        Delay 1
        Wend
    cmdExample2.Enabled = True
End Sub

Private Sub cmdExample3_Click()
Dim x, y, z As Single
    x = Val(txtInput(0).Text)   'convert text to number
    y = Val(txtInput(1).Text)
    z = Val(txtInput(2).Text)
    AddNumbers x, y, z   'initial call to Recursive Procedure
End Sub

Private Sub txtInput_Change(Index As Integer)
    'Perform data validation
    Select Case Index
        Case 0 'Start must be greater than Start
            If Val(txtInput(1)) <= Val(txtInput(0)) Then
                txtInput(1).Text = Val(txtInput(0)) + 10
            End If
        Case 1  'End must be greater than Start
            If Val(txtInput(1)) <= Val(txtInput(0)) Then
                txtInput(1).Text = Val(txtInput(0)) + 10
            End If
        Case 2
            If Val(txtInput(2)) < 1 Or Val(txtInput(2)) > Val(txtInput(1)) Then
                txtInput(2).Text = 1
            End If
    End Select
End Sub
Further Notes

The Input Box is used to obtain user input. It has three values, separated by commas:

e.g.  \texttt{Msg} = \texttt{InputBox("Enter a message", "Type "Stop" to Stop", "Continue")}

1. The text ‘\texttt{Enter a message}’ appears in the main grey section of the box.
2. The text ‘\texttt{Type “Stop” to Stop}’ appears in the blue title bar (double sets of quotation marks are necessary when the text you are displaying itself contains quotation marks e.g. ‘\texttt{Type “Stop” to Stop}’).
3. The text ‘\texttt{Continue}’ appears as a default value that the user may or may not type over.

NB. ‘\texttt{Msg}’ is merely a string variable used to hold the value entered in the Input Box by the user.

Multiple Selection ....\texttt{SELECT CASE END SELECT}.

The ‘Select Case’ statement is used in place of multiple ‘IF ... THEN ... ELSE IF’ statements. It is much tidier to use.

Suggestions for Consolidation and Extension

1. To test your understanding of the program, especially your knowledge of ‘IF ... THEN ... ELSE IF’ statements, rewrite the event ‘\texttt{txtInput\_Change}’ without using the ‘\texttt{SELECT CASE ... END SELECT}’ statements.
2. Change the delay value to see what affect it has at different points in your code e.g. ‘\texttt{Delay 3}’, ‘\texttt{Delay 4}’ and so on. The ‘\texttt{Delay}’ procedure is also very useful for slowing down the running of the code, so that you can investigate what is happening inside loops. Codes normally execute so fast that the designer can’t properly see what is happening.
3. Modify the interface to use scroll bars instead of text boxes for input. You are advised to leave the working text boxes on the interface while testing the scroll bars.

Questions

1. Why are the command buttons disabled and enabled inside the events \texttt{cmdExample1} and \texttt{cmdExample2}?
2. What does the ‘\texttt{Val()}’ function do? Why is it needed?
3. Write code that uses an \texttt{Input Box} to ask the user’s name. The input value should be displayed in a label named ‘\texttt{lblUserName}’.
4. Which line in the recursive procedure ‘\texttt{ADDNUMBERS}’ performs the vital test to see if the procedure is recursively called again?
5. What is the meaning of ‘\texttt{Timer}’?
6. What is the parameter variable in the ‘\texttt{Delay}’ procedure called? What \texttt{datatype} is it?
7. What are the parameter variables in the ‘\texttt{AddNumbers}’ procedure called? What \texttt{datatype} are they?