STACK SIMULATOR TASK

Write a program that will visually simulate a working stack.

You can choose whether:

* its first slot is numbered zero or one.
* pushes and pops occur at the visual bottom or top of the stack.
* the stackpointer is updated after operations or before operation.

These decisions should be displayed on the simulator’s interface.

**Functional requirements**

* The contents of the stack must be visible onscreen.
* The state of the stackpointer must be visible.
* It must support the commands PUSH and POP.
* It must indicate what values are popped.
* It must prevent stack overflow and underflow conditions.

**Nonfunctional requirements**

* It should require minimal keystrokes and mouse actions.

**Extensions**

* Add PEEK, SWAP and ROTATE commands.

STA‘stack simulator by Mark Kelly 2011-07-26

Public Class frmStackSim

Dim Stackpointer As Integer = 0

Const MAXSTACK = 4

Private Sub txtPushValue\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtPushValue.TextChanged

cmdPush.Text = "Push " + txtPushValue.Text

cmdPush.Enabled = Len(cmdPush.Text) > 0 'disable push while box is empty!

End Sub

Private Sub cmdPush\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdPush.Click

If txtPushValue.Text = "" Then Exit Sub 'empty value?

If Stackpointer = MAXSTACK Then 'would overflow?

MsgBox("Stack overflow prevented")

Exit Sub

End If

'MsgBox("Pushing " & txtPushValue.Text)

box.Items.Add(txtPushValue.Text) 'push!

txtPushValue.Text = "" 'clear box

Stackpointer += 1 'update SP

txtTop.Text = Stackpointer 'show SP to the human

box.Refresh()

End Sub

Private Sub cmdPop\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdPop.Click

If Stackpointer = 0 Then

MsgBox("Prevented stack underflow")

Exit Sub

End If

Stackpointer -= 1 'go back to top item

MsgBox("POPPED " & box.Items(Stackpointer)) 'be informative

box.Items.RemoveAt(Stackpointer) 'take item off stack

txtTop.Text = Stackpointer 'show SP to the human

box.Refresh()

End Sub

Private Sub cmdExit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdExit.Click

End

End Sub

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

End Class

