

Estruturas de Repetição II

Conteúdo

- Ciclo For ... Next
 - Apresentação da sintaxe e de exemplos deste ciclo.
- Validação de entrada de dados
 - Apresentação de dois algoritmos de validação de entrada de dados através de InputBox.
- Ciclos encadeados
 - Exemplos de aplicação com ciclos encadeados.

Ciclo For ... Next

- O ciclo for permite definir de uma forma compacta ciclos que percorrem o conjunto de valores de um intervalo predefinido.

- Sintaxe (geral)

```
→ Para <variável> = <valor inicial> até <valor final>  
    <bloco de instruções>  
Próximo valor <variável>  
    <instrução>
```

- Explicação

- A variável do ciclo vai percorrer o intervalo de valores compreendidos entre o valor inicial e o valor final. O ciclo termina quando o valor da variável ultrapassa o valor final (num ciclo crescente). O incremento da variável tem que ser um valor inteiro positivo.

Sintaxe do ciclo For ... Next

- Há duas variantes deste ciclo.

- Ciclo com incremento unitário

```
For <variável> = <val_inicial> To <val_final>  
    <bloco de instruções>  
Next
```

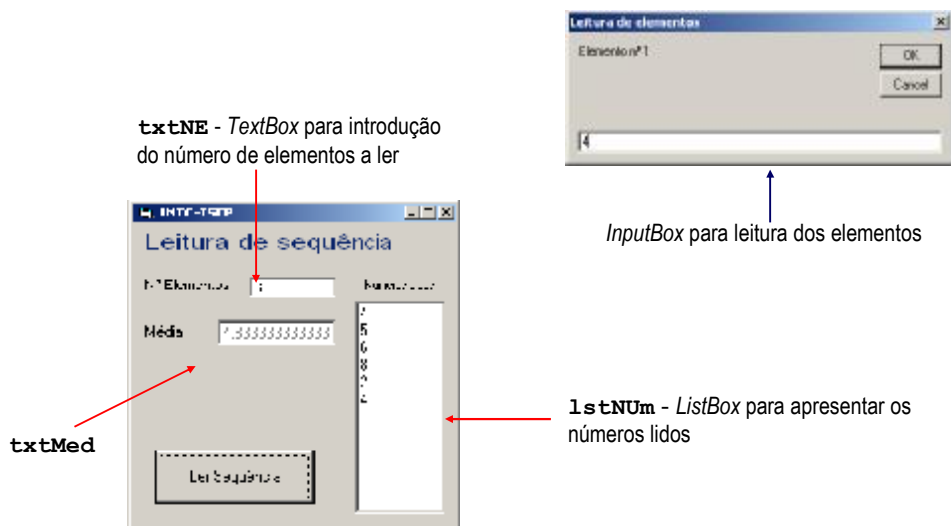
- Ciclo com incremento definido pelo programador (positivo ou negativo)

```
For <variável> = <val_inicial> To <val_final> Step <valor>  
    <bloco de instruções>  
Next
```

Exemplo de aplicação

- Elabore um programa que leia uma sequência de números inteiros, em que número de elementos é dado pelo utilizador. O programa deverá apresentar a média dos números introduzidos, o maior e o menor número. O programa deve ainda apresentar cada um dos números lidos.
- Resolução:
 - Como o número de elementos a ler é dado pelo utilizador pode ser utilizado um ciclo For.

Resolução - Form



Resolução - Código

```

Option Explicit
Private Sub cmdLer_Click()
    Dim ne As Integer, x As Integer, s As Long, num As Integer

    lstNum.Clear
    txtMed.Text = ""

    ne = Val(txtNE.Text)
    If ne <= 0 Then
        MsgBox "Não há elementos a considerar!", vbCritical
        Exit Sub
    End If

    For x = 1 To ne
        num = Val(InputBox("Elemento n° " & x))
        lstNum.AddItem num
        s = s + num
    Next

    txtMed.Text = s / ne
End Sub

```

Leitura e validação do número de elementos da sequência.

Pára quando x ultrapassar o valor de ne

Incrementa x e volta ao início do ciclo.

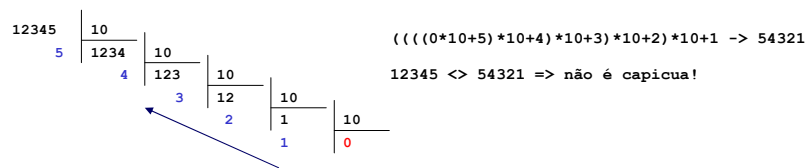
Execução passo a passo

- Teste com sequência de 3 elementos: 2, 3, 4

Iteração	ne	x	x <= ne	num	s
	3	0		0	0
1		1	True	2	0+2 -> 2
2		2	True	3	2+3 -> 5
3		3	True	4	5+4 -> 9
4		4	False		

Ciclos encadeados

- Elabore um programa que apresente numa ListBox todos as capicuas existentes num intervalo dado pelo utilizador (os limites do intervalo devem ser validados).
- Resolução
 - Um número é capicua se for idêntico à sequência dos seu algarismos, por ordem inversa.



- É preciso verificar se cada um dos números do intervalo é capicua.

Resolução - Form

txtInf e txtSup -
TextBoxes com limite inferior
e superior

MsgBox com a mensagem
de erro no intervalo.



lstCap - ListBox com as
capicuas do intervalo

Option Explicit

```
Private Sub cmdSair_Click()  
End  
End Sub
```

Resolução - Código

```
Private Sub cmdProc_Click()  
    Dim num As Long, aux As Long, cap As Long, alg As Integer  
    Dim inf As Long, sup As Long  
  
    lstCap.Clear  
    inf = Val(txtLI.Text)  
    sup = Val(txtLS.Text)  
    If inf < 0 Or inf > sup Then  
        MsgBox "Atenção: Intervalo errado!", vbCritical  
        Exit Sub  
    End If  
  
    For num = inf To sup  
        aux = num  
        cap = 0  
        Do While aux > 0  
            alg = aux Mod 10  
            aux = aux \ 10  
            cap = cap * 10 + alg  
        Loop  
        If cap = num Then  
            lstCap.AddItem cap  
        End If  
    Next  
End Sub
```

Leitura e validação dos limites do intervalo

É calculado o "inverso" de cada número do intervalo. Utiliza-se a variável **aux** para não afectar o valor de **num**.

Se o número for igual ao inverso então é capicua.

Validação de entrada de dados

- Problema:
 - Pretende-se validar os dados introduzidos pelo utilizador durante um ciclo, através de InputBoxes. Caso os dados estejam errados, o utilizador deve voltar a introduzi-los.
- Solução
 - Utiliza-se um ciclo para controlar a entrada de dados. Há duas abordagens possíveis:
 - Ciclo DLU / DLW
 - A leitura é feita dentro do ciclo.
 - Ciclo DWL / DUL
 - A primeira leitura é feita fora do ciclo. Só entra no ciclo se o valor lido estiver errado.

Validação - Exemplo

- Exemplo: leitura de notas entre 0 e 20

- DLU

```
Do
  nota = Val(InputBox("Nota nº " & x))
Loop Until nota >= 0 And nota <= 20
```

O ciclo só termina quando a nota for válida.

- DWL

```
nota = Val(InputBox("Nota nº " & x))
Do While nota < 0 Or nota > 20
```

O ciclo só é executado se a nota for inválida.

```
  nota = Val(InputBox("Erro: Introduza novamente " _
    & " a nota nº " & x))
```

Loop

A *InputBox* informa também o utilizador do erro.

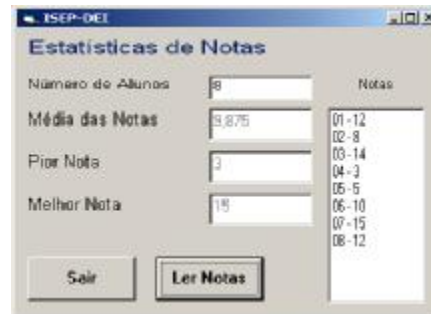
Exemplo de aplicação

- Elabore um programa que leia as notas dos alunos de uma disciplina (número de alunos dado pelo utilizador) e apresente a média das notas e a melhor e a pior nota. Devem também ser apresentadas numa *ListBox* todas as notas lidas.

- Resolução

- O número de alunos deve ser validado (>0)
- Como o número de alunos é conhecido, pode utilizar-se um ciclo For para leitura das notas. Para cada nota usa-se um ciclo para validar a nota.
- A leitura das notas é efectuada através de uma *InputBox*.

Resolução - Form



Resolução – Código (I)

Option Explicit

```
Private Sub cmdSair_Click()
    End
End Sub
```

```
Private Sub cmdLer_Click()
    Dim na As Integer, nota As Integer, soma As Integer
    Dim max As Integer, min As Integer, x As Integer
```

```
na = Val(txtNA.Text)
If na <= 0 Then
    MsgBox "Erro: Número de alunos inválido!"
    Exit Sub
End If
```

```
soma = 0
```

```
max = -1
min = 21
```

Comentário

' valores iniciais

Os valores iniciais devem, neste caso, ser impossíveis. Assim, na primeira iteração são de certeza alterados.

Resolução – Código (II)

```
For x = 1 To na
  Do
    nota = Val(TextBox("Nota n° " & x)) ← Validação da nota.
  Loop Until nota >= 0 And nota <= 20

  lstNotas.AddItem Format(x, "00") & " - " & nota
  soma = soma + nota

  If nota > max Then
    max = nota
  End If

  If nota < min Then
    min = nota
  End If

Next
txtMed.Text = soma / na
txtMelhor.Text = max
txtPior.Text = min
End Sub
```

' compara com melhor anterior
Se a nota é maior do que o máximo actual passa a ser o novo máximo.

' compara com pior anterior
Se a nota é menor do que o mínimo actual passa a ser o novo mínimo.

Conclusão

- Estudamos dois ciclos:
 - Do ... Loop
 - 4 variantes – DWL, DUL, DLW, DLU
 - Pode ser utilizado em qualquer situação.
 - For ... Next
 - 2 variantes – ciclo crescente e decrescente
 - Só pode ser aplicado a intervalos de valores.
 - Os limites do intervalo têm que ser numéricos, inteiros e constantes.
 - O incremento tem que ser constante.
- Introduzimos o conceito de padrão
 - O que permite agilizar a resolução de problemas.