

Mão na massa

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

- 1) Crie, na classe `ListaDeContasCorrentes`, uma nova propriedade `Public`, com `Get` explícito, para que seja possível acessar o tamanho da lista de fora da classe:

```
Public ReadOnly Property tamanho() As Integer
    Get
        Return _proximaposicao
    End Get
End Property
```

- 2) Ainda na classe `ListaDeContasCorrentes`, um **indexador**, criando uma `Region` para indexadores:

```
#Region "Indexadores"

Default Public ReadOnly Property Item(indice As Integer)
    Get
        Return GetContaCorrente(indice)
    End Get
End Property

#End Region
```

- 3) Crie também o método `GetContaCorrente`:

```
Public Function GetContaCorrente(indice As Integer) As ContaCorrente

    If indice < 0 And indice >= _proximaposicao Then
        Throw New ArgumentOutOfRangeException(NameOf(indice))
    End If
    Return _items(indice)

End Function
```

- 4) Crie o método `AdicionarVariosUmAUm`, para adicionar diversas contas na lista de uma vez só:

```
Public Sub AdicionarVariosUmAUm(ParamArray contas As ContaCorrente())

    For Each conta As ContaCorrente In contas
        Adicionar(conta)
    Next

End Sub
```

- 5) Crie um último método, chamado `somaNumeros`, que irá somar todos os números recebidos por parâmetro:

```

Public Function somaNumeros(ParamArray numeros() As Integer) As Integer

    Dim soma As Integer = 0
    For Each numero As Integer In numeros

        soma += numero
    Next
    Return soma

End Function

```

6) No formulário `Frm_Principal2`, adicione um `GroupBox`, com Name `Grp_AdicionarVarios`. Dentro desse `GroupBox`, adicione oito `Label`, com Name `Lbl_Agencia1`, `Lbl_Conta1`, `Lbl_Agencia2`, `Lbl_Conta2`, `Lbl_Agencia3`, `Lbl_Conta3`, `Lbl_Agencia4` e `Lbl_Conta4`, oito `TextBox`, com Name `Txt_Agencia1`, `Txt_Conta1`, `Txt_Agencia2`, `Txt_Conta2`, `Txt_Agencia3`, `Txt_Conta3`, `Txt_Agencia4` e `Txt_Conta4`, e um `Button`, com Name `Btn_AdicionarVarios`.

7) No código fonte do formulário `Frm_Principal2`, dentro da sub-rotina `New()`, insira a inicialização da propriedade `Text` do `GroupBox` e dos novos `Label` e `Button`:

```

Grp_AdicionarVarios.Text = "Adicionar Vários"
Btn_AdicionarVarios.Text = "Adicionar"
Lbl_Agencia1.Text = "Agencia 1"
Lbl_Conta1.Text = "Conta 1"
Lbl_Agencia2.Text = "Agencia 2"
Lbl_Conta2.Text = "Conta 2"
Lbl_Agencia3.Text = "Agencia 3"
Lbl_Conta3.Text = "Conta 3"
Lbl_Agencia4.Text = "Agencia 4"
Lbl_Conta4.Text = "Conta 4"

```

8) Crie uma nova função para escrever na tela cada elemento da Lista de contas correntes, conforme abaixo:

```

Public Function EscreverElementosArray() As String

    Dim elementos As String = String.Empty
    For i As Integer = 0 To Lista.tamanho - 1
        elementos += Lista.GetContaCorrente(i).agencia.ToString + " - " +
            Lista.GetContaCorrente(i).numero.ToString + vbCrLf
    Next
    Return elementos

End Function

```

9) Crie uma segunda função com o mesmo objetivo, mas desta vez utilizando o indexador:

```

Public Function EscreverElementosArrayIndexador() As String

    Dim elementos As String = String.Empty
    For i As Integer = 0 To Lista.tamanho - 1
        elementos += Lista(i).agencia.ToString + " - " + Lista(i).numero.ToString + vbCrLf
    Next
    Return elementos

```

```
End Function
```

10) Modifique a rotina do clique do botão `Btn_Adicionar`, para chamar a nova rotina `EscreverElementosArrayIndexador`, ao invés do método `EscreverElementosArray`, da classe `ListaDeContaCorrente`. Substitua a linha:

```
Txt_Array.Text = Lista.EscreverElementosArray
```

Pela linha:

```
Txt_Array.Text = EscreverElementosArrayIndexador()
```

11) Da mesma maneira, modifique a rotina ddo clique do botão `Btn_Remover`. Substitua a linha:

```
Txt_Array.Text = Lista.EscreverElementosArray
```

Pela linha:

```
Txt_Array.Text = EscreverElementosArrayIndexador()
```

12) Na rotina do clique do novo botão `Btn_AdicionaVarios`, insira o seguinte código:

```
Private Sub Btn_AdicionaVarios_Click(sender As Object, e As EventArgs) Handles Btn_AdicionaVarios.C

    If Txt_Agencia1.Text <> "" And Txt_Conta1.Text <> "" Then
        Dim Conta1 As New ContaCorrente(Txt_Agencia1.Text, Txt_Conta1.Text)
    If Txt_Agencia2.Text <> "" And Txt_Conta2.Text <> "" Then
        Dim Conta2 As New ContaCorrente(Txt_Agencia2.Text, Txt_Conta2.Text)
    If txt_Agencia3.Text <> "" And Txt_Conta3.Text <> "" Then
        Dim Conta3 As New ContaCorrente(txt_Agencia3.Text, Txt_Conta3.Text)
    If Txt_Agencia4.Text <> "" And Txt_Conta4.Text <> "" Then
        Dim Conta4 As New ContaCorrente(Txt_Agencia4.Text, Txt_Conta4.Text)
        Lista.AdicionarVariosUmAUm(Conta1, Conta2, Conta3, Conta4)
        Txt_Array.Text = EscreverElementosArrayIndexador()
    Else
        Lista.AdicionarVariosUmAUm(Conta1, Conta2, Conta3)
        Txt_Array.Text = EscreverElementosArrayIndexador()
    End If
    Else
        Lista.AdicionarVariosUmAUm(Conta1, Conta2)
        Txt_Array.Text = EscreverElementosArrayIndexador()
    End If
    Else
        Lista.AdicionarVariosUmAUm(Conta1)
        Txt_Array.Text = EscreverElementosArrayIndexador()
    End If
End If

End Sub
```

13) Salve e execute. Teste a aplicaci o n, adicionando e removendo contas individualmente e em grupos.