

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) No projeto **ByteBank.SistemaAgencia**, adicione um novo formulário, com as seguintes propriedades e valores:

Propriedade	Valor
Name	Frm_Principal2
Text	Formulário Principal 2
StartPosition	CenterScreen
WindowState	Normal

2) Esse formulário irá mostrar o conteúdo do seu array, com campos de texto e botões para adicionar e remover elementos. Então adicione dois *Label*, com Name `Lbl_Agencia` e `Lbl_Conta`, três *TextBox*, com Name `Txt_Agencia`, `Txt_Conta` e `Txt_Array` (esse também terá as propriedades `Multiline` e `ScrollBars` com os valores `True` e `Vertical`, respectivamente), e dois *Button*, com Name `Btn_Adicionar` e `Btn_Remover`.

3) Na Classe `ContaCorrente`, insira um novo construtor para permitir uma instanciação sem parâmetros:

```
Private Sub New()
    End Sub
```

4) Ainda na classe `ContaCorrente`, sobrescreva o método `Equals`, herdado da classe `Object`, com o seguinte código:

```
Public Overrides Function Equals(obj As Object) As Boolean
    Dim outraConta As New ContaCorrente()
    outraConta = TryCast(obj, ContaCorrente)
    If outraConta Is Nothing Then
        Return False
    End If
    If outraConta.agencia = agencia And outraConta.numero = numero Then
        Return True
    End If
    Return False
End Function
```

5) Na classe `ListaDeContasCorrentes`, inclua o método para identificar a posição de um elemento na lista, conforme abaixo:

```
Public Function posicaoElemento(Conta As ContaCorrente) As Integer
    For i As Integer = 0 To _proximaposicao - 1
        Dim contaAtual As ContaCorrente = items(i)
```

```

If contaAtual.Equals(Conta) Then
    Return i
End If
Next

Return -1

End Function

```

6) Ainda na classe `ListaDeContasCorrentes`, inclua o método para remover um elemento da Lista, conforme abaixo:

```

Public Sub Remover(Conta As ContaCorrente)

    Dim indice As Integer = posicaoElemento(Conta)
    If indice = -1 Then
        Return
    End If

    _proximaposicao -= 1
    For i As Integer = indice To _proximaposicao - 1
        _items(i) = _items(i + 1)
    Next

End Sub

```

7) E inclua o método para escrever na tela os elementos da lista, conforme abaixo:

```

Public Function EscreverElementosArray()

    Dim elementos As String = String.Empty
    For i As Integer = 0 To _proximaposicao - 1
        elementos += _items(i).agencia.ToString + " - " +
            _items(i).numero.ToString + vbCrLf
    Next
    For i As Integer = _proximaposicao To _items.Length - 1
        elementos += "NULL" + vbCrLf
    Next

    Return elementos

End Function

```

8) Modifique o código da rotina de clique do botão `Btn_Principal3`, do formulário `Frm_Principal`, para localizar e imprimir a posição de algum elemento:

```

Private Sub Btn_Principal3_Click(sender As Object, e As EventArgs) Handles Btn_Principal3.Click

    Dim Lista As New ListaDeContasCorrentes(10)

    Lista.Adicionar(New ContaCorrente(111, 11111))
    Lista.Adicionar(New ContaCorrente(222, 22222))
    Lista.Adicionar(New ContaCorrente(333, 33333))
    Lista.Adicionar(New ContaCorrente(444, 44444))

```

```

Lista.Adicionar(New ContaCorrente(555, 55555))
Lista.Adicionar(New ContaCorrente(666, 66666))

Dim vPosicao As Integer =
    Lista.posicaoElemento(New ContaCorrente(444, 44444))
MsgBox(vPosicao)

End Sub

```

9) Vá para o código fonte do formulário `Frm_Principal2`, dentro da sub-rotina `New()`, insira a inicialização da propriedade `Text` dos *Label*, dos *Button* e do formulário:

```

Me.Text = "Array Contas Correntes"
Lbl_Agencia.Text = "Agencia"
Lbl_Contra.Text = "Conta"
Btn_Adicionar.Text = "Adicionar"
Btn_Remover.Text = "Remover"

```

10) No início do código, coloque a declaração `Imports` para a classe `ContaCorrente`:

```
Imports ByteBank.Bibliotecas.Classes.Clientes
```

11) No início do formulário, declare a variável `Lista`:

```
Dim Lista As New ListaDeContasCorrentes()
```

12) Insira a rotina de clique do botão `Btn_Adicionar`:

```

Private Sub Btn_Adicionar_Click(sender As Object, e As EventArgs)
    Handles Btn_Adicionar.Click

    If Txt_Agencia.Text <> "" And Txt_Contra.Text <> "" Then
        Dim conta As New ContaCorrente(Txt_Agencia.Text, Txt_Contra.Text)
        Lista.Adicionar(conta)
        Txt_Array.Text = Lista.EscreverElementosArray
    End If

```

```
End Sub
```

13) E insira a rotina de clique do botão `Btn_Remover`:

```

Private Sub Btn_Remover_Click(sender As Object, e As EventArgs)
    Handles Btn_Remover.Click

    If Txt_Agencia.Text <> "" And Txt_Contra.Text <> "" Then
        Dim conta As New ContaCorrente(Txt_Agencia.Text, Txt_Contra.Text)
        Lista.Remover(conta)
        Txt_Array.Text = Lista.EscreverElementosArray
    End If

```

```
End Sub
```

- 14) Por fim, selecione o formul  rio `Frm_Principal2` como o formul  rio padr  o do projeto, clicando com o bot  o direito do mouse sobre o projeto `ByteBank.SistemaAgencia` e selecionando **Propriedades**. Em **Formul  rio de Inicializa  o**, selecione o `Frm_Principal2` como padr  o.