

## 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 formulário `Frm_Principal`, do projeto `ByteBank.SistemaAgencia`, adicione um terceiro botão, com `Name Btn_Principal3`.

2) No código do formulário, dentro da sub-rotina `New()`, inclua a inicialização da propriedade `Text` do novo botão:

```
Btn_Principal3.Text = "Clique Aqui"
```

3) Inclua o código do clique do botão `Btn_Principal3` que declara o array `Lista` do tipo `ListaDeContasCorrentes` e que insere as contas na lista uma a uma:

```
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))
    End Sub
```

4) Crie a classe `ListaDeContasCorrentes`, na pasta `Classes`, do projeto `ByteBank.SistemaAgencia`.

5) No código da nova classe, importe o `Namespace` da classe `ContaCorrente`, que está declarada no projeto `ByteBank.Bibliotecas`:

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

6) Declare as suas propriedades:

```
#Region "Propriedades"

    Private Property _items() As ContaCorrente()
    Private Property _proximaposicao As Integer

#End Region
```

7) Insira o construtor da classe:

```
#Region "Construtores"

    Public Sub New(Optional numeroPosicoesIniciais As Integer = 5)
        _items = New ContaCorrente(numeroPosicoesIniciais - 1) {}
        _proximaposicao = 0
    End Sub

#End Region
```

8) Insira o método MeuMetodo , utilizado para testar os parâmetros opcionais:

```
Public Sub MeuMetodo(Optional vString As String = "String Opcional", Optional vInteger As Integer =
    MsgBox("VString = " + vString + ", vInteger = " + vInteger.ToString)
End Sub
```

9) Insira o método que adiciona um novo item no array de contas correntes:

```
Public Sub Adicionar(item As ContaCorrente)

    VerificarCapacidadeArrayCopy(_proximaposicao + 1)
    Console.WriteLine("Novo membro adicionado na posição " + _proximaposicao.ToString)
    _items(_proximaposicao) = item
    _proximaposicao += 1

End Sub
```

10) Insira o método que verifica a capacidade do array, sem utilizar o método Array.Copy :

```
Private Sub VerificarCapacidadeArray(tamanhoNecessario As Integer)

    If _items.Length >= tamanhoNecessario Then
        Return
    End If

    Dim novoTamanho As Integer = _items.Length * 2

    If novoTamanho < tamanhoNecessario Then
        novoTamanho = tamanhoNecessario
    End If
```

```
Dim NovoArray(novoTamanho) As ContaCorrente

For i As Integer = 0 To _items.Length - 1
    Console.WriteLine(".")
    NovoArray(i) = _items(i)
Next
_items = NovoArray

End Sub
```

11) Insira o m  odo que verifica a capacidade do array e utiliza o m  odo `Array.Copy`:

```
Private Sub VerificarCapacidadeArrayCopy(tamanhoNecessario As Integer)

If _items.Length >= tamanhoNecessario Then
    Return
End If

Dim novoTamanho As Integer = _items.Length * 2

If novoTamanho < tamanhoNecessario Then
    novoTamanho = tamanhoNecessario
End If

Dim NovoArray(novoTamanho) As ContaCorrente

Array.Copy(_items, NovoArray, _items.Length)

_items = NovoArray

End Sub
```