

Mãos na massa

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

- 1) Vá para o formulário `Frm_Principal` e adicione dois novos botões, com `Name` `Btn_Principal4` e `Btn_Principal5`.
- 2) Crie uma nova classe, chamada `ListaDeObjects`. Copie todo o código de `ListaDeContasCorrentes` e faça as seguintes modificações:

- Troque todas as referências à `ContaCorrente` por `Object`
- Renomeie o método `GetContaCorrente` por `GetObject`
- Nos métodos `PosicaoElemento` e `AdicionarVariosUmAUm`, substitua as variáveis `conta` / `contas` por `item` / `itens`
- Elimine os métodos `MeuMetodo`, `EscreverElementoArray` e `somaNumeros`
- Retire todos as linhas com `Console.WriteLine`

```
Public Class ListaDeObjects
```

```
#Region "Propriedades"
```

```
    Private Property _items() As Object()
```

```
    Private Property _proximaposicao As Integer
```

```
    Public ReadOnly Property tamanho() As Integer
```

```
        Get
```

```
            Return _proximaposicao
```

```
        End Get
```

```
    End Property
```

```
#End Region
```

```
#Region "Indexadores"
```

```
    Default Public ReadOnly Property Item(indice As Integer)
```

```
        Get
```

```
            Return GetObject(indice)
```

```
        End Get
```

```
    End Property
```

```
#End Region
```

```
#Region "Construtores"
```

```
    Public Sub New(Optional numeroPosicoesIniciais As Integer = 5)
```

```
        _items = New Object(numeroPosicoesIniciais - 1) {}
```

```
        _proximaposicao = 0
```

```
    End Sub
```

```
#End Region
```

```
#Region "Métodos"
```

```
Public Sub Adicionar(item As Object)
    VerificarCapacidadeArrayCopy(_proximaposicao + 1)
    _items(_proximaposicao) = item
    _proximaposicao += 1
End Sub

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 Object

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

    _items = NovoArray

End Sub

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 Object

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

End Sub

Public Function posicaoElemento(item As Object) As Integer

    For i As Integer = 0 To _proximaposicao - 1
        Dim itemAtual As Object = _items(i)
        If itemAtual.Equals(item) Then
            Return i
        End If
    Next

    Return -1

End Function
```

```
End Function
```

```
Public Sub Remover(item As Object)
```

```
    Dim indice As Integer = posicaoElemento(item)
```

```
    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
```

```
Public Function GetObject(indice As Integer) As Object
```

```
    If indice < 0 And indice >= _proximaposicao Then
```

```
        Throw New ArgumentOutOfRangeException(NumberOf(indice))
```

```
    End If
```

```
    Return _items(indice)
```

```
End Function
```

```
Public Sub AdicionarVariosUmAUm(ParamArray items As Object())
```

```
    For Each item As Object In items
```

```
        Adicionar(item)
```

```
    Next
```

```
End Sub
```

```
#End Region
```

```
End Class
```

3) Crie uma nova classe, chamada `Lista`. Copie todo o código de `ListaDeObjects` e faça as seguintes modificações:

- Defina a classe como `Of T`
- Troque todas as referências à `Object` por `T`
- Renomeie o método `GetObject` por `GetItem`

```
Public Class Lista(Of T)
```

```
#Region "Propriedades"
```

```
    Private Property _items() As T()
```

```
    Private Property _proximaposicao As Integer
```

```
    Public ReadOnly Property tamanho() As Integer
```

```
        Get
```

```
            Return _proximaposicao
```

```
        End Get
```

```
    End Property
```

```
#End Region
```

```
#Region "Indexadores"
```

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

```
#End Region
```

```
#Region "Construtores"
```

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

```
#End Region
```

```
#Region "Métodos"
```

```
Public Sub Adicionar(item As T)
    VerificarCapacidadeArrayCopy(_proximaposicao + 1)
    _items(_proximaposicao) = item
    _proximaposicao += 1
End Sub
```

```
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 T
```

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

```
    _items = NovoArray
```

```
End Sub
```

```
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
```

```
End If
```

```
Dim NovoArray(novoTamanho) As T
```

```
For i As Integer = 0 To _items.Length - 1
```

```
    NovoArray(i) = _items(i)
```

```
Next
```

```
_items = NovoArray
```

```
End Sub
```

```
Public Function posicaoElemento(item As T) As Integer
```

```
    For i As Integer = 0 To _proximaposicao - 1
```

```
        Dim itemAtual As T = _items(i)
```

```
        If itemAtual.Equals(item) Then
```

```
            Return i
```

```
        End If
```

```
    Next
```

```
    Return -1
```

```
End Function
```

```
Public Sub Remover(item As T)
```

```
    Dim indice As Integer = posicaoElemento(item)
```

```
    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
```

```
Public Function GetItem(indice As Integer) As T
```

```
    If indice < 0 And indice >= _proximaposicao Then
```

```
        Throw New ArgumentOutOfRangeException(NumberOf(indice))
```

```
    End If
```

```
    Return _items(indice)
```

```
End Function
```

```
Public Sub AdicionarVariosUmAUm(ParamArray items As T())
```

```
    For Each item As T In items
```

```
        Adicionar(item)
```

```
    Next
```

```
End Sub
```

```
#End Region
```

```
End Class
```

4) No código fonte do formulário `Frm_Principal`, dentro da sub-rotina `New()`, insira a inicialização da propriedade `Text` dos novos botões:

```
Btn_Principal4.Text = "Clique Aqui"
Btn_Principal5.Text = "Clique Aqui"
```

5) No código da rotina de clique do botão `Btn_Principal4`, insira o seguinte código:

```
Private Sub Btn_Principal4_Click(sender As Object, e As EventArgs)
    Handles Btn_Principal4.Click

    Dim Lista As New ListaDeObjects()

    Lista.Adicionar(12)
    Lista.Adicionar(13)
    Lista.Adicionar(15)
    Lista.AdicionarVariosUmAUm(12, 15, 14, 13, 18, 20)

    Dim soma As Integer = 0
    For i As Integer = 0 To Lista.tamanho - 1
        soma += Lista(i)
    Next

    MsgBox(soma)

    Dim Lista2 As New ListaDeObjects()

    Lista2.Adicionar("a1")
    Lista2.Adicionar("a2")
    Lista2.Adicionar("a3")
    Lista2.AdicionarVariosUmAUm("a4", "a5", "a6", "a7")

    Dim somaStr As String = ""
    For i As Integer = 0 To Lista2.tamanho - 1
        somaStr += Lista2(i)
    Next

    MsgBox(somaStr)

    Dim Lista3 As New ListaDeObjects()

    Lista3.Adicionar(New ContaCorrente(1, 1))
    Lista3.Adicionar(New ContaCorrente(1, 1))
    Lista3.Adicionar(New ContaCorrente(1, 1))
    Lista3.AdicionarVariosUmAUm(
        New ContaCorrente(1, 1),
        New ContaCorrente(1, 1),
        New ContaCorrente(1, 1),
        New ContaCorrente(1, 1)
    )

    Dim somaStr2 As String = ""
    For i As Integer = 0 To Lista3.tamanho - 1
        Dim contaAtual As ContaCorrente = TryCast(Lista3(i), ContaCorrente)
        somaStr2 += contaAtual.agencia.ToString + " - " +
```

```

        somaStr2 += contaAtual.agencia.ToString + " - " +
        contaAtual.numero.ToString + " ; "

    Next

    MsgBox(somaStr2)

End Sub

```

6) No código da rotina de clique do botão `Btn_Principal5`, insira o seguinte código:

```

Private Sub Btn_Principal5_Click(sender As Object, e As EventArgs)
    Handles Btn_Principal5.Click

    Dim Lista As New Lista(Of Integer)

    Lista.Adicionar(12)
    Lista.Adicionar(13)
    Lista.Adicionar(15)
    Lista.AdicionarVariosUmAUm(12, 15, 14, 13, 18, 20)

    Dim soma As Integer = 0
    For i As Integer = 0 To Lista.tamanho - 1
        soma += Lista(i)
    Next

    MsgBox(soma)

    Dim Lista2 As New Lista(Of String)

    Lista2.Adicionar("a1")
    Lista2.Adicionar("a2")
    Lista2.Adicionar("a3")
    Lista2.AdicionarVariosUmAUm("a4", "a5", "a6", "a7")

    Dim somaStr As String = ""
    For i As Integer = 0 To Lista2.tamanho - 1
        somaStr += Lista2(i)
    Next

    MsgBox(somaStr)

    Dim Lista3 As New Lista(Of ContaCorrente)

    Lista3.Adicionar(New ContaCorrente(1, 1))
    Lista3.Adicionar(New ContaCorrente(1, 1))
    Lista3.Adicionar(New ContaCorrente(1, 1))
    Lista3.AdicionarVariosUmAUm(
        New ContaCorrente(1, 1),
        New ContaCorrente(1, 1),
        New ContaCorrente(1, 1),
        New ContaCorrente(1, 1)
    )

    Dim somaStr2 As String = ""
    For i As Integer = 0 To Lista3.tamanho - 1
        Dim contaAtual As ContaCorrente = TryCast(Lista3(i), ContaCorrente)
        somaStr2 += contaAtual.agencia.ToString + " - " + contaAtual.numero.ToString + " ; "
    Next

```

Next

MsgBox(somaStr2)

End Sub

7) Salve, execute e faça os testes, como feitos durante a aula.