09

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) Comece baixando o projeto aqui (https://caelum-online-public.s3.amazonaws.com/1016-vb-net-lambda-linq-dataset/01/ByteBank.zip). Descompacte o arquivo e abra a solução **ByteBank.sln**.

2) No projeto **ByteBank.Bibliotecas**, dentro da pasta **Classes**, crie uma pasta chamada **Criterios**. Dentro dessa nova pasta, crie a classe `CriterioContaCorrenteAgenciaNumero` , com o seguinte código:

```
Imports ByteBank.Bibliotecas.Classes.Clientes

Namespace Classes.Criterios

    Public Class CriterioContaCorrenteAgenciaNumero
            Implements IComparer(Of ContaCorrente)

        Public Function Compare(x As ContaCorrente, y As ContaCorrente) As Integer
                Implements IComparer(Of ContaCorrente).Compare

            ' SE X < Y ---> -1
            ' SE X = Y ---> 0
            ' SE X > Y ---> 1

            Dim vAgenciaX As String = x.agencia.ToString
            Dim vContaX As String = x.numero.ToString

            Dim vAgenciaY As String = y.agencia.ToString
            Dim vContaY As String = y.numero.ToString

            Dim vCriterioX As String = vAgenciaX + "-" + vContaX
            Dim vCriterioY As String = vAgenciaY + "-" + vContaY

            If vCriterioX < vCriterioY Then
                Return -1
            End If
            If vCriterioX = vCriterioY Then
                Return 0
            End If
            Return 1

            'Return x.titular.nome.CompareTo(y.titular.nome)

        End Function

    End Class

End Namespace
```

3) Na mesma pasta **Criterios**, crie também a classe `CriterioContaCorrenteNome` , com o seguinte código:

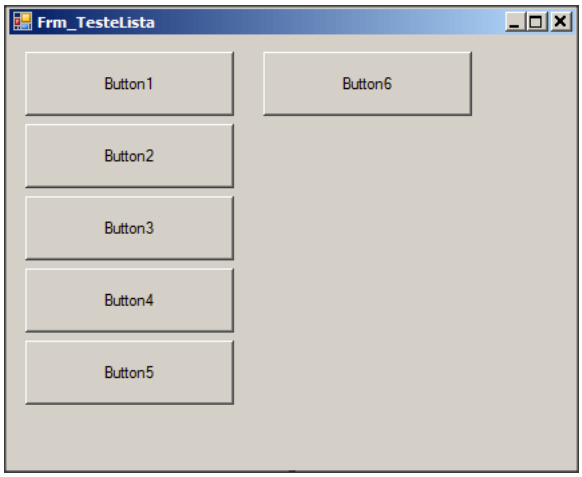
```
Imports ByteBank.Bibliotecas.Classes.Clientes

Namespace Classes.Criterios

    Public Class CriterioContaCorrenteNome
            Implements IComparer(Of ContaCorrente)


        Public Function Compare(x As ContaCorrente, y As ContaCorrente) As Integer
                Implements IComparer(Of ContaCorrente).Compare

            ' SE X < Y ---> -1
            ' SE X = Y ---> 0
            ' SE X > Y ---> 1

            If x.titular.nome < y.titular.nome Then
                Return -1
            End If

            If x.titular.nome = y.titular.nome Then
                Return 0
            End If

            Return 1

            'Return x.titular.nome.CompareTo(y.titular.nome)

        End Function

    End Class

End Namespace
```

4) Ainda na pasta **Criterios**, crie a classe `CriterioContaCorrenteSaldo` , com o seguinte código:

```
Imports ByteBank.Bibliotecas.Classes.Clientes

Namespace Classes.Criterios

    Public Class CriterioContaCorrenteSaldo
            Implements IComparer(Of ContaCorrente)

        Public Function Compare(x As ContaCorrente, y As ContaCorrente) As Integer
                Implements IComparer(Of ContaCorrente).Compare

            ' SE X < Y ---> -1
            ' SE X = Y ---> 0
            ' SE X > Y ---> 1

            If x.saldo < y.saldo Then
                Return -1
            End If

            If x.saldo = y.saldo Then
                Return 0
            End If
```

```
            Return 1

            ' Return x.saldo.CompareTo(y.saldo)

        End Function

    End Class

End Namespace
```

5) Agora, no projeto **ByteBank.SistemaAgencia**, adicione um botão ao formulário `Frm_TesteLista` , com `Name Button6` . Você terá:



6) No código-fonte do formulário **`Frm_TesteLista`** . Inclua o `Imports` do `NameSpace` das novas classes:

```
Imports ByteBank.Bibliotecas.Classes.Criterios
```

7) Por fim, implemente o código do evento de `Click` do botão `Button6` :

```
Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click

    Dim Conta1 As New ContaCorrente(277, 223444, "João")
    Conta1.Depositar(10000)

    Dim Conta2 As New ContaCorrente(277, 255645, "Pedro")
    Conta2.Depositar(5000)

    Dim Conta3 As New ContaCorrente(500, 166323, "Alberto")
    Conta3.Depositar(7000)

    Dim ListaContasCorrentes As New List(Of ContaCorrente)
    ListaContasCorrentes.Add(Conta1)
    ListaContasCorrentes.Add(Conta2)
    ListaContasCorrentes.Add(Conta3)

    MsgBox("Lista de contas original: " + String.Join(",", ListaContasCorrentes))

    ListaContasCorrentes.Sort(New CriterioContaCorrenteNome())
    MsgBox("Lista de contas ordenada por nome: " + String.Join(",", ListaContasCorrentes))
```

```
        ListaContasCorrentes.Sort(New CriterioContaCorrenteSaldo())
        MsgBox("Lista de contas ordenada por saldo: " + String.Join(",", ListaContasCorrentes))

        ListaContasCorrentes.Sort(New CriterioContaCorrenteAgenciaNumero())
        MsgBox("Lista de contas ordenada por agencia/numero: " + String.Join(",", ListaContasCorrentes))

    End Sub
```