

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**, inclua novos itens no menu:

- Aula 03
 - Vídeo 05

2) Inclua um novo formulário, de `Name Frm_Video05`, e com as propriedades `StartPosition` e `WindowState` com os valores `CenterScreen` e `Normal`, respectivamente.

3) Nesse novo formulário, adicione três `Label`, de `Name Lbl_Video05`, `Lbl_ContaCorrente` e `Lbl_ContaCorrente2`, respectivamente, e um `Button`, de `Name Btn_Fechar`. Além disso, adicione um `GroupBox`, de `Name Grp_Movimentacao`, e dentro dele, adicione três `Label`, de `Name Lbl_Movimentacao`, `Lbl_Saldo` e `Lbl_Saldo2`, respectivamente, um `TextBox`, de `Name Txt_Valor`, e dois `Button`, de `Name Btn_Saque` e `Btn_Transferir`, respectivamente.

4) Vá na classe `ContaCorrente` e edite o método `Sacar`, para prevenir que o valor a ser sacado não seja menor que 0 e nem maior que o saldo. Mas caso esse(s) caso(s) ocorra(m), lance uma exceção:

```
Public Sub Sacar(ValorSacado As Double, ValorLabel As String)

  If ValorSacado < 0 Then
    Throw New ArgumentException("Valor " + ValorLabel + " é negativo. " +
      ValorSacado.ToString, NameOf(ValorSacado))
  End If
  If m_saldo < ValorSacado Then
    Dim vMensagem As String
    vMensagem = "Valor " + ValorLabel + " é maior que o saldo"
    Throw New ValorSacadoMenorSaldoException(ValorSacado, m_saldo, vMensagem)
  Else
    m_saldo -= ValorSacado
  End If

End Sub
```

5) Vá no método `Transferir` e simplifique-o:

```
Public Sub Transferir(ValorTransferencia As Double,
  ContaDestino As ContaCorrente)
  Sacar(ValorTransferencia, "da transferencia")
  ContaDestino.Depositar(ValorTransferencia)
End Sub
```

6) Crie uma nova classe chamada `ValorSacadoMenorSaldoException`, com o seguinte código:

Namespace Classes

```

Public Class ValorSacadoMenorSaldoException
    Inherits Exception

#Region "PROPRIEDADES"
    Public ReadOnly Property ValorSacado As Double
    Public ReadOnly Property Saldo As Double
#End Region

#Region "CONSTRUTORES"

    Sub New()
        End Sub

    Sub New(_ValorSacado As Double, _Saldo As Double, Message As String)
        Me.New(Message)
        ValorSacado = _ValorSacado
        Saldo = _Saldo
    End Sub

    Sub New(message As String)
        MyBase.New(message)
        ' Algum tipo de implementação
    End Sub

#End Region

End Class

```

```
End Namespace
```

7) No formulário principal (`Frm_Principal`), dê um duplo clique na opção **Vídeo 05** do menu. Inclua o código abaixo, para abrir o formulário `Frm_Video05` :

```

Private Sub Video05ToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles Video05ToolStripMenuItem.Click
    Dim f As New Frm_Video05
    f.ShowDialog()
End Sub

```

8) Agora, acesse o código fonte do formulário `Frm_Video05` . Inclua no seu começo o `Imports` das classes:

```
Imports ByteBank.Classes
```

9) Em seguida, inclua a declaração de duas instâncias da classe `ContaCorrente` :

```

Dim Conta As New ContaCorrente(237, 144422)
Dim Conta2 As New ContaCorrente(237, 166433)

```

10) Inclua, dentro da sub-rotina `New()` , a inicialização da propriedade `Text` do formulário, dos `Label`, dos `Button` e do `GroupBox`:

```

Me.Text = "Formulário do Vídeo 05"
Lbl_Video05.Text = "Formulário do Vídeo 05"
Lbl_ContaCorrente.Text = "Conta: " + Conta.agencia.ToString +
    "-" + Conta.numero.ToString
Lbl_ContaCorrente2.Text = "Conta: " + Conta2.agencia.ToString +
    "-" + Conta2.numero.ToString
Btn_Fechar.Text = "Fechar"
Grp_Movimentacao.Text = "Movimentação da conta"
Lbl_Movimentacao.Text = "Digite o valor"
Btn_Saque.Text = "Sacar"
Btn_Transferir.Text = "Transferir"
Lbl_Saldo.Text = "Saldo da conta = " + Conta.saldo.ToString
Lbl_Saldo2.Text = "Saldo da conta2 = " + Conta2.saldo.ToString

```

11) No evento de clique do `Btn_Fechar` , feche o formulário:

```

Private Sub Btn_Fechar_Click(sender As Object, e As EventArgs) Handles Btn_Fechar.Click
    Me.Close()
End Sub

```

12) No evento de clique do `Btn_Saque` , faça o saque de uma conta, tratando as possíveis exceções dentro do Try-Catch :

```

Private Sub Btn_Saque_Click(sender As Object, e As EventArgs) Handles Btn_Saque.Click

    Try
        Dim ValorSacar As Double = Val(Txt_Valor.Text)
        Conta.Sacar(ValorSacar, "sacado")
        Lbl_Saldo.Text = "Saldo da conta = " + Conta.saldo.ToString
    Catch ex As ValorSacadoMenorSaldoException
        MsgBox("Ocorreu um erro ValorSacadoMenorSaldoException: " + ex.Message +
            ". Valor Saque de " + ex.ValorSacado.ToString +
            ". Saldo de " + ex.Saldo.ToString)
    Catch ex As ArgumentException
        MsgBox("Ocorreu um erro ArgumentException: " + ex.Message)
    Catch ex As Exception
        MsgBox("Ocorreu um erro Exception: " + ex.Message)
    End Try
End Sub

```

13) No evento de clique do `Btn_Transferir` , faça a transferência entre as contas, tratando as possíveis exceções dentro do Try-Catch :

```

Private Sub Btn_Transferir_Click(sender As Object, e As EventArgs) Handles Btn_Transferir.Click

    Try
        Dim ValorTransferencia As Double = Val(Txt_Valor.Text)
        Conta.Transferir(ValorTransferencia, Conta2)
        Lbl_Saldo.Text = "Saldo da conta = " + Conta.saldo.ToString
        Lbl_Saldo2.Text = "Saldo da conta2 = " + Conta2.saldo.ToString
    Catch ex As ValorSacadoMenorSaldoException
        MsgBox("Ocorreu um erro ValorSacadoMenorSaldoException: " + ex.Message +

```

```
    ". Valor da Transferencia de " + ex.ValorSacado.ToString +  
    ". Saldo de " + ex.Saldo.ToString)  
Catch ex As ArgumentException  
    MsgBox("Ocorreu um erro ArgumentException: " + ex.Message)  
Catch ex As Exception  
    MsgBox("Ocorreu um erro Exception: " + ex.Message)  
End Try  
  
End Sub
```

14) Salve e execute a aplicação. Verifique o funcionamento dos exemplos vistos nesta aula.