

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) Na classe `ContaCorrente`, inclua a propriedade `NumeroSaquesErrados`, como mostrado abaixo:

```
Private m_NumeroSaquesErrados As Integer
Public Property NumeroSaquesErrados As Integer
    Get
        Return m_NumeroSaquesErrados
    End Get
    Private Set(value As Integer)
        m_NumeroSaquesErrados = value
    End Set
End Property
```

2) Inclua também a propriedade `NumeroTransferenciasErradas`:

```
Private m_NumeroTransferenciasErradas As Integer
Public Property NumeroTransferenciasErradas As Integer
    Get
        Return m_NumeroTransferenciasErradas
    End Get
    Private Set(value As Integer)
        m_NumeroTransferenciasErradas = value
    End Set
End Property
```

3) No método `Sacar`, se o saldo for menor que o valor a ser sacado, incremente a propriedade `NumeroSaquesErrados`:

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

    If ValorSacado < 0 Then
        m_NumeroSaquesErrados += 1
        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"
        m_NumeroSaquesErrados += 1
        Throw New ValorSacadoMenorSaldoException(ValorSacado, m_saldo, vMensagem)
    Else
        m_saldo -= ValorSacado
    End If

End Sub
```

4) No método `Transferir`, se o saque não puder ser feito, incremente a propriedade `NumeroTransferenciasErradas`:

```

Public Sub Transferir(ValorTransferencia As Double,
    ContaDestino As ContaCorrente)

    Try

        Sacar(ValorTransferencia, "da transferencia")
    Catch ex As ValorSacadoMenorSaldoException
        m_NumeroTransferenciasErradas += 1
        Throw New OperacaoFinanceiraException("Operação Inválida", ex)
    Catch ex As ArgumentException
        m_NumeroTransferenciasErradas += 1
        Throw
    End Try
    ContaDestino.Depositar(ValorTransferencia)

End Sub

```

5) Na classe `ValorSacadoMenorSaldoException`, modifique a herança da classe para:

```

Public Class ValorSacadoMenorSaldoException
    Inherits OperacaoFinanceiraException

```

6) Inclua mais um novo construtor da classe:

```

Sub New(Message As String, execucaoInterna As Exception)
    MyBase.New(Message, execucaoInterna)
End Sub

```

7) Crie a classe `OperacaoFinanceriaException`, com o seguinte código:

Namespace Classes

```

Public Class OperacaoFinanceiraException
    Inherits Exception

```

```

#Region "CONSTRUTORES"

```

```

    Sub New()

```

```

    End Sub

```

```

    Sub New(Message As String)

```

```

        MyBase.New(Message)

```

```

    End Sub

```

```

    Sub New(Message As String, execucaoInterna As Exception)

```

```

        MyBase.New(Message, execucaoInterna)

```

```

    End Sub

```

```

#End Region

```

```

End Class

```

```
End Namespace
```

8) No formulário `Frm_Video05`, abaixo do `GroupBox`, adicione dois `Label`, de `Name` `Lbl_SaquesErrados` e `Lbl_TransferErrados`, respectivamente, um `Button`, de `Name` `Btn_Stack` e um `TextBox`, de `Name` `Txt_Stack`.

9) No código fonte do formulário `Frm_Video05`, inclua a declaração de duas novas variáveis:

```
Dim vStack As String = ""
Dim vControle As Boolean = False
```

10) Na sub-rotina `New()` do formulário, inicialize as propriedades `Text` dos novos componentes:

```
Btn_Stack.Text = "Ver último StackTrace"
Txt_Stack.Visible = False
Lbl_SaquesErrados.Text = "Num. Saques Errados: " +
    Conta.NumeroSaquesErrados.ToString
Lbl_TransferErrados.Text = "Num. Transf Errados: " +
    Conta.NumeroTransferenciasErradas.ToString
```

11) No clique do botão `Btn_Saque`, modifique o código da ação do evento para adicionar a `StackTrace` da possível exceção à variável `vStack` e para exibir o número de saques errados:

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

    Try
        vStack = ""
        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
        Lbl_SaquesErrados.Text = "Num. Saques Errados: " + Conta.NumeroSaquesErrados.ToString
        vStack = ex.StackTrace
        MsgBox("Ocorreu um erro ValorSacadoMenorSaldoException: " + ex.Message +
            ". Valor Saque de " + ex.ValorSacado.ToString + ". Saldo de " + ex.Saldo.ToString)
    Catch ex As ArgumentException
        Lbl_SaquesErrados.Text = "Num. Saques Errados: " + Conta.NumeroSaquesErrados.ToString
        vStack = ex.StackTrace
        MsgBox("Ocorreu um erro ArgumentException: " + ex.Message)
    Catch ex As Exception
        vStack = ex.StackTrace
        MsgBox("Ocorreu um erro Exception: " + ex.Message)
    End Try

End Sub
```

12) No clique do botão `Btn_Transferir`, adicione/concatene a `StackTrace` da possível exceção à variável `vStack` e exiba a quantidade de transferências e saques errados:

```

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

    Try
        vStack = ""
        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 OperacaoFinanceiraException
        Lbl_SaquesErrados.Text = "Num. Saques Errados: " + Conta.NumeroSaquesErrados.ToString
        Lbl_TransferErrados.Text = "Num. Transf Errados: " + Conta.NumeroTransferenciasErradas.ToStr
        vStack = ex.StackTrace
        vStack += ex.InnerException.StackTrace
        MsgBox("Ocorreu um erro OperacaoFinanceiraException: " + ex.Message)
    Catch ex As ValorSacadoMenorSaldoException
        Lbl_SaquesErrados.Text = "Num. Saques Errados: " + Conta.NumeroSaquesErrados.ToString
        Lbl_TransferErrados.Text = "Num. Transf Errados: " + Conta.NumeroTransferenciasErradas.ToStr
        vStack = ex.StackTrace
        MsgBox("Ocorreu um erro ValorSacadoMenorSaldoException: " + ex.Message +
            ". Valor da Transferencia de " + ex.ValorSacado.ToString + ". Saldo de " + ex.Saldo.
    Catch ex As ArgumentException
        Lbl_SaquesErrados.Text = "Num. Saques Errados: " + Conta.NumeroSaquesErrados.ToString
        Lbl_TransferErrados.Text = "Num. Transf Errados: " + Conta.NumeroTransferenciasErradas.ToStr
        vStack = ex.StackTrace
        MsgBox("Ocorreu um erro ArgumentException: " + ex.Message)
    Catch ex As Exception
        vStack = ex.StackTrace + vbCrLf
        vStack += ex.InnerException.StackTrace
        MsgBox("Ocorreu um erro Exception: " + ex.Message +
            " - Erro original Interno: " + ex.InnerException.Message)
    End Try

End Sub

```

13) No clique do botão Btn_Stack , exiba/esconda o Txt_Stack e exiba nele o conteúdo da variável vStack :

```

Private Sub Btn_Stack_Click(sender As Object, e As EventArgs) Handles Btn_Stack.Click
    If vControle = False Then
        vControle = True
        Txt_Stack.Visible = True
    Else
        vControle = False
        Txt_Stack.Visible = False
    End If
    Txt_Stack.Text = vStack
End Sub

```

14) Salve e execute a aplicação. Verifique o funcionamento com os exemplos mostrados na aula.