

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) No projeto **ByteBank.Bibliotecas**, abra a classe **Agencia**.

2) No seu código, inclua a importação do `Namespace` das classes:

```
Imports ByteBank.Bibliotecas.Classes.Funcionarios
```

3) Inclua as novas propriedades para esta classe:

```
Private Shared m_NumeroClienteFila As Integer
Public Shared ReadOnly Property NumeroClienteFila As Integer
    Get
        Return m_NumeroClienteFila
    End Get
End Property
```

```
Private Shared m_TempoTotalFila As Integer
Public Shared ReadOnly Property TempoTotalFila As Integer
    Get
        Return m_TempoTotalFila
    End Get
End Property
```

```
Public Property Caixas As List(Of Caixa)
Public Property Fila As Queue(Of ClienteAgencia)
Public Property NumeroCliente As Integer
```

4) No seu construtor, inclua as inicializações das novas propriedades, como mostrado abaixo:

```
Public Sub New(codigo As Integer)

    agencia = codigo
    _contasCorrentes = New HashSet(Of ContaCorrente)

    ' ##### SIMULAÇÃO #####

    Caixas = New List(Of Caixa)
    Fila = New Queue(Of ClienteAgencia)
    m_NumeroClienteFila = 0
    m_TempoTotalFila = 0
    NumeroCliente = 0

    ' #####

End Sub
```

5) E implemente os métodos `EntrarNaFila`, `ExecutaAtendimento`, `SairDaFila` e `TamanhoFila`:

```
Public Sub EntrarNaFila(_TempoServico As Integer)
    NumeroCliente += 1
    Dim Cliente As New ClienteAgencia(_TempoServico, NumeroCliente)
    Fila.Enqueue(Cliente)
End Sub

Public Sub ExecutaAtendimento(_TempoIncremento As Integer)
    For I As Integer = 0 To Caixas.Count - 1
        If Caixas(I).Ocupado = True Then
            Caixas(I).EfetuarAtendimento(_TempoIncremento)
        End If
    Next
    For I As Integer = 0 To Fila.Count - 1
        Fila(I).AdicionarTempoEspera(_TempoIncremento)
    Next
End Sub

Public Sub SairDaFila()
    If Fila.Count > 0 Then
        For I As Integer = 0 To Caixas.Count - 1
            If Caixas(I).Ocupado = False Then
                Dim Cliente As ClienteAgencia = Fila.Dequeue()
                m_NumeroClienteFila += 1
                m_TempoTotalFila += Cliente.TempoEspera
                Caixas(I).IniciarAtendimento(Cliente.TempoServico)
            End If
        Next
    End If
End Sub

Public Function TamanhoFila() As Integer
    Return Fila.Count
End Function
```

6) Ainda no projeto **ByteBank.Bibliotecas**, dentro da pasta **Classes\Clientes**, crie uma nova classe, chamada **ClienteAgencia**, com o seguinte código:

```
Namespace Classes.Clientes

    Public Class ClienteAgencia

        #Region "PROPRIEDADES"

        Public Property Nome As String
        Public Property TempoServico As Integer
        Public Property TempoEspera As Integer

        #End Region

        #Region "CONSTRUTORES"

        Public Sub New(_TempoServico As Integer, Senha As Integer)
            Nome = "Cliente Senha: " + Senha.ToString
        End Sub

    End Class

End Namespace
```

```

        Nome = _Nome
        TempoServico = _TempoServico
        TempoEspera = 0
    End Sub

#End Region

#Region "MÉTODOS"

    Public Sub AdicionarTempoEspera(_TempoIncremento As Integer)
        TempoEspera += _TempoIncremento
    End Sub

#End Region

End Class

End Namespace

```

7) No mesmo projeto, dentro da pasta **Classes\Funcionarios**, crie a classe **Caixa**, com o seguinte código:

```

Namespace Classes.Funcionarios

    Public Class Caixa

#Region "PROPRIEDADES"

        Public Property Nome As String
        Public Property Ocupado As Boolean
        Public Property TempoAtendimento As Integer
        Public Property TempoMaximo As Integer

#End Region

#Region "CONSTRUTORES"

        Public Sub New(_Nome As String)
            Nome = _Nome
            Ocupado = False
        End Sub

#End Region

#Region "MÉTODOS"

        Public Sub IniciarAtendimento(_TempoServico As Integer)
            TempoMaximo = _TempoServico
            Ocupado = True
            TempoAtendimento = 0
        End Sub

        Public Sub EfetuarAtendimento(_TempoIncremento As Integer)
            If Ocupado = True Then
                TempoAtendimento += _TempoIncremento
                If TempoAtendimento >= TempoMaximo Then
                    Ocupado = False
                End If
            End If
        End Sub
    End Class
End Namespace

```

```

        End If
    End Sub

    Public Function MostrarStatus() As String
        Dim vStatus As String = "Caixa " + Nome
        If Ocupado Then
            vStatus += " Ocupado."
        Else
            vStatus += " Livre."
        End If
        Return vStatus
    End Function

```

```
#End Region
```

```
End Class
```

```
End Namespace
```

8) Agora no projeto **ByteBank.SistemaAgencia**, dentro da pasta **Classes**, crie a classe **Aleatorio**, com o seguinte código:

```

Imports ByteBank.Bibliotecas.Classes.Funcionarios

Namespace classes

    Public Class Aleatorio

        #Region "PROPRIEDADES"

            Public Property MenorNumero As Double
            Public Property MaiorNumero As Double

        #End Region

        #Region "CONSTRUTORES"

            Public Sub New(_MenorNumero As Integer, _MaiorNumero As Integer)
                MenorNumero = _MenorNumero
                MaiorNumero = _MaiorNumero
            End Sub

        #End Region

        #Region "MÉTODOS"

            Public Function getNumeroAleatorio() As Double
                Dim NumeroRandom As New Random()
                Dim NumeroAtual As Double = NumeroRandom.Next(MenorNumero, MaiorNumero)
                Return NumeroAtual
            End Function

        #End Region

    End Class

End Namespace

```

9) Baixe [aqui \(https://caelum-online-public.s3.amazonaws.com/1025-vb-net-listas-dicionarios-conjuntos/06/Frm_Simulacao.zip\)](https://caelum-online-public.s3.amazonaws.com/1025-vb-net-listas-dicionarios-conjuntos/06/Frm_Simulacao.zip) um ZIP dos arquivos do formulário `Frm_Simulacao`. Feito o download, extraia esse ZIP dentro da pasta do projeto **ByteBank.SistemaAgencia**.

10) No **Visual Studio**, clique com o botão direito do mouse sobre o projeto **ByteBank.SistemaAgencia** e acesse **Adicionar --> Item Existente**. Em seguida, no diretório do projeto, selecione o arquivo **Frm_Simulacao.vb**

11) Acesse o código-fonte desse novo formulário. No clique do botão `Btn_FazSimulacao` inclua o código abaixo:

```
Private Sub Btn_FazSimulacao_Click(sender As Object, e As EventArgs)
    Handles Btn_FazSimulacao.Click

    Dim TempoMenorServico As Integer = Txt_TempoMenorServico.Text
    Dim TempoMaiorServico As Integer = Txt_TempoMaiorServico.Text
    Dim NumClieChegamMinutoMinimo As Integer = Txt_NumeroClientesMinimo.Text
    Dim NumClieChegamMinutoMaximo As Integer = Txt_NumeroClientesMaximo.Text
    Dim NumeroCaixas As Integer = Txt_NumCaixas.Text
    Dim TempoPasso As Integer = Txt_TempoPasso.Text
    Dim TempoMaximoSimulacao As Integer = Txt_TempoSimulacao.Text
    Dim vHora As New TimeSpan(10, 0, 0)

    Dim AgenciaBancaria As New Agencia(277)
    For I As Integer = 1 To NumeroCaixas
        AgenciaBancaria.Caixas.Add(New Caixa("CAIXA " + I.ToString))
    Next

    Dim SerieHistoricaServico As New Aleatorio(TempoMenorServico, TempoMaiorServico + 1)
    Dim SerieHistoricaNumClientes As New Aleatorio(NumClieChegamMinutoMinimo, NumClieChegamMinutoMa)

    Dim vContadorMinuto As Integer = 1

    For I As Integer = 1 To TempoMaximoSimulacao

        'SE FOR UM MINUTO COMPLETO

        If vContadorMinuto = 1 Then

            ' Quandoos clientes chegaram na agencia?
            Dim vNumClientes As Integer = SerieHistoricaNumClientes.getNumeroAleatorio

            '   CHEGAR CLIENTES NA AGENCIA
            '   COLOCA-LOS NA FILA

            For J As Integer = 1 To vNumClientes
                Dim vTempoServico As Integer = SerieHistoricaServico.getNumeroAleatorio
                AgenciaBancaria.EntrarNaFila(vTempoServico)
                System.Threading.Thread.Sleep(100)
            Next
        End If

        'FIM DO SE

        vContadorMinuto += 1
        If vContadorMinuto = 60 Then
```

```

        vContadorMinuto = 1
    End If

    'INCREMENTAR TEMPO DE SERVICO E TEMPO DE ESPERA
    AgenciaBancaria.ExecutaAtendimento(1)

    'VER SE EXISTE CAIXA DISPONIVEL PARA TIRAR CLIENTES DA FILA
    AgenciaBancaria.SairDaFila()

    'ESCREVER O STATUS ATUAL
    EscreverAgencia(AgenciaBancaria)

    'ESCREVER AS ESTATISTICAS
    Lbl_Tempo.Text = "TEMPO DA SIMULAÇÃO: " + ConverteSegundosEmHoras(I)
    Lbl_TamanhoFila.Text = "TAMANHO DA FILA: " + AgenciaBancaria.Fila.Count.ToString
    Lbl_TempoMedia.Text = "TEMPO MÉDIO DE ESPERA: " +
        ConverteSegundosEmHoras(Agencia.TempoTotalFila / Agencia.NumeroClienteFila)

    Me.Refresh()
    System.Threading.Thread.Sleep(TempoPasso)
Next

End Sub

```

12) Inclua também as funções `EscreverAgencia` e `ConverteSegundosEmHoras` :

```

Sub EscreverAgencia(AgenciaBancaria As Agencia)

    Txt_Caixas.Text = ""
    Txt_Fila.Text = ""
    For I As Integer = 0 To AgenciaBancaria.Caixas.Count - 1
        Txt_Caixas.Text += AgenciaBancaria.Caixas(I).MostrarStatus + vbCrLf
    Next

    For I As Integer = 0 To AgenciaBancaria.Fila.Count - 1
        Txt_Fila.Text = AgenciaBancaria.Fila(I).Nome + vbCrLf
    Next

End Sub

Function ConverteSegundosEmHoras(Segundos As Integer) As String
    Dim iSecond As Double = Segundos
    Dim iSpan As TimeSpan = TimeSpan.FromSeconds(iSecond)

    Return iSpan.Hours.ToString.PadLeft(2, "0"c) & ":" &
        iSpan.Minutes.ToString.PadLeft(2, "0"c) & ":" &
        iSpan.Seconds.ToString.PadLeft(2, "0"c)
End Function

```

13) Por fim, no formulário `Frm_Principal01` , adicione mais um item de menu, chamado **Simulação**. E no seu código-fonte, adicione o código a ser executado no clique desta nova opção do menu:

```

Private Sub SimulaçãoToolStripMenuItem_Click(sender As Object, e As EventArgs)
    Handles SimulaçãoToolStripMenuItem.Click
    Dim F As New Frm_Simulacao

```

```
Dim F As New Form_Simulacao
```

```
F.MdiParent = Me
```

```
F.Show()
```

```
End Sub
```