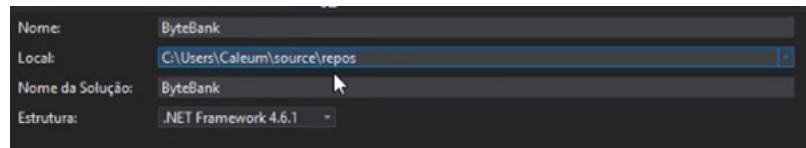


Mão na massa

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

1) Crie um novo projeto, acessando **Visual Basic** --> **Windows Desktop** --> **Aplicativo do Windows Forms (.NET Framework)**. O nome do projeto e o nome da solução será **ByteBank**. Certifique-se que você está usando o mesmo local que vem sendo usado desde o início dos treinamentos e que o **.NET Framework** seja o **4.6.1 ou superior**:



2) Seguindo a convenção de treinamentos anteriores, renomeie o `Form1` para `Frm_Principal`, e lembre-se de checar se a propriedade `WindowState` está com o valor `Normal` e se a propriedade `StartPosition` está com o valor `CenterScreen`.

3) Ao formulário, adicione um *Label*, com `Name Lbl_Principal`, e um *Button*, com `Name Btn_Click`.

4) Na sub-rotina `New()` do formulário, modifique as propriedades `Text` do formulário e dos componentes recém-adicionados:

```
Public Sub New()

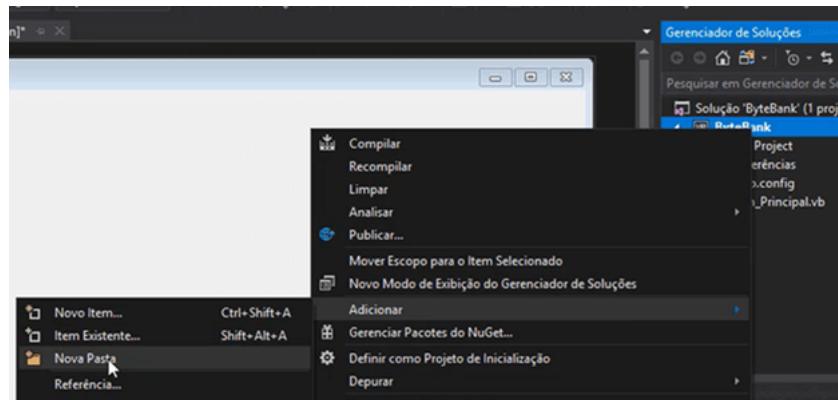
    ' Esta chamada é requerida pelo designer.
    InitializeComponent()

    ' Adicione qualquer inicialização após a chamada InitializeComponent().

    Me.Text = "Principal"
    Lbl_Principal.Text = "Principal"
    Btn_Click.Text = "Clique aqui"

End Sub
```

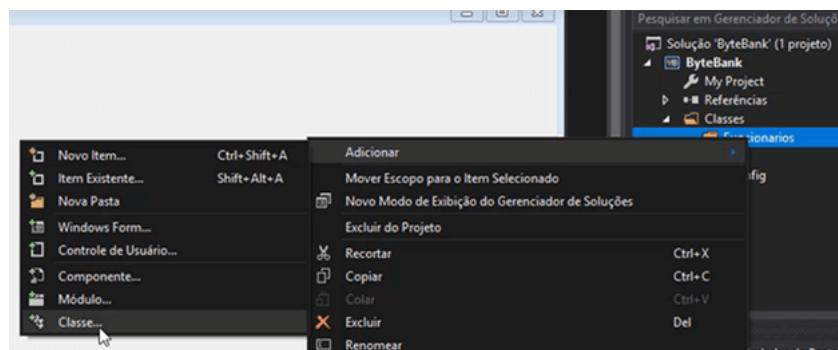
5) Para tornar o código mais claro e organizado, e facilitar a manutenção, ficando mais fácil encontrar o arquivo desejado no código fonte, crie uma pasta para cada tipo de componente, ou seja, uma para os **Forms** (e arraste para dentro dela o **Frm_Principal**, criado anteriormente) e uma pasta para as **Classes**, na qual você irá criar uma sub-pasta chamada **Funcionários**, onde será armazenada a classe `Funcionario`. Então, clique com o botão da direita do mouse sobre o projeto `ByteBank` e selecione **Adicionar --> Nova Pasta**:



6) O nome da nova pasta será **Forms**. Em seguida, arraste o *Form* já existente no projeto (**Frm_Principal**) para dentro da pasta.

7) Repita o processo para a criação da pasta **Classes** e criando também uma sub-pasta chamada **Funcionários**.

8) Crie a classe **Funcionario**, dentro da pasta **Funcionários**, clicando com o botão da direita do mouse sobre a pasta **Funcionarios** e selecionando **Adicionar --> Classe**:



9) Na classe **Funcionario**, defina as propriedades `nome`, `cpf`, todas `String`, e `salario`, que é `Double`:

```
Public Class Funcionario

    Public Property nome As String
    Public Property cpf As String
    Public Property salario As Double

End Class
```

10) Além disso, os funcionários do **ByteBank** recebem uma bonificação anual de 10% dos seus salários. Defina a bonificação no método `GetBonificacao()`, calculando o seu valor a partir da propriedade `Salario`, conforme abaixo:

```
Public Class Funcionario

    Public Property nome As String
    Public Property cpf As String
    Public Property salario As Double

    Public Function GetBonificacao() As Double
        Return salario * 0.1
    End Function

End Class
```

11) Quando os arquivos do projeto são separados em diretórios, existe uma convenção que é a de criar, no nome do Namespace , o caminho dos subdiretórios onde eles se encontram. Então coloque o código da classe dentro do Namespace :

```
Namespace Classes.Funcionarios
```

```
    Public Class Funcionario
```

```
        #Region "PROPRIEDADES"
```

```
            Public Property nome As String
            Public Property cpf As String
            Public Property salario As Double
        #End Region
```

```
        #Region "CONSTRUTORES"
```

```
        #End Region
```

```
        #Region "MÉTODOS"
```

```
            Public Function GetBonificacao() As Double
                Return salario * 0.1
            End Function
        #End Region
```

```
    End Class
```

```
End Namespace
```

12) Na mesma pasta que Funcionario , crie a classe Diretor , que tem uma bonificação diferente, o seu próprio salário:

```
Namespace Classes.Funcionarios
```

```
    Public Class Diretor
```

```
        #Region "PROPRIEDADES"
```

```
            Public Property nome As String
            Public Property cpf As String
            Public Property salario As Double
        #End Region
```

```
        #Region "CONSTRUTORES"
```

```
        #End Region
```

```
        #Region "MÉTODOS"
```

```
            Public Function GetBonificacao() As Double
                Return salario * 1
            End Function
        #End Region
```

```
    End Class
```

```
End Namespace
```

13) Crie também a classe `Gerente` , que tem como bonificação 10% do seu salário:

```
Namespace Classes.Funcionarios

    Public Class Gerente

        #Region "PROPRIADES"
            Public Property nome As String
            Public Property cpf As String
            Public Property salario As Double
        #End Region

        #Region "CONSTRUTORES"

        #End Region

        #Region "MÉTODOS"
            Public Function GetBonificacao() As Double
                Return salario * 0.1
            End Function
        #End Region

    End Class

End Namespace
```

14) Dentro do projeto, crie a pasta **Bibliotecas** e dentro dela crie a classe `GerenciadorBonificacao` . Nesta classe, você irá criar uma variável chamada `TotalBonificacao` , que irá acumular o valor das bonificações. Crie também dois métodos, chamados `Registrar` , um para receber um `Gerente` e outro para receber um `Diretor` , e irá acumular o valor da bonificação na variável. Por fim, crie outro método para retornar o valor da variável que acumulou as bonificações. Como as classes `GerenciadorBonificacao` e `Funcionario` não se encontram no `Namespace` original (que é o projeto), não esqueça de adicionar o `Imports` :

```
Imports ByteBank.Classes.Funcionarios

Namespace Classes.Bibliotecas

    Public Class GerenciadorBonificacao

        Private _totalBonificacao As Double

        Public Sub Registrar(Gerente As Gerente)
            _totalBonificacao += Gerente.GetBonificacao()
        End Sub

        Public Sub Registrar(Diretor As Diretor)
            _totalBonificacao += Diretor.GetBonificacao()
        End Sub

        Public Function getBonificacao() As Double
            Return _totalBonificacao
        End Function

    End Class

End Namespace
```

End Namespace

15) No modo *Designer*, dê um duplo clique sobre o botão `Btn_Click`, para programar a ação que será efetuada no clique no botão. Dentro dele, instancie um `Gerente` e um `Diretor`, preenchendo as suas propriedades, e imprima-as. Em seguida, instancie um `GerenciadorBonificacao`, registre o gerente e o diretor e imprima o valor total de bonificação a ser pago. O código do formulário ficará da seguinte forma:

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

Public Class Frm_Principal

    Public Sub New()

        ' Esta chamada é requerida pelo designer.
        InitializeComponent()

        ' Adicione qualquer inicialização após a chamada InitializeComponent().

        Me.Text = "Principal"
        Lbl_Principal.Text = "Principal"
        Btn_Click.Text = "Clique aqui"

    End Sub

    Private Sub Btn_Click_Click(sender As Object, e As EventArgs) Handles Btn_Click.Click

        Dim gerenciadorBonificacao As New GerenciadorBonificacao()

        Dim Carlos As New Gerente()
        Carlos.nome = "Carlos Antônio"
        Carlos.cpf = "543.445.434-00"
        Carlos.salario = 10000

        gerenciadorBonificacao.Registrar(Carlos)

        Dim Bia As New Diretor()
        Bia.nome = "Beatriz Regina"
        Bia.cpf = "522.476.980-12"
        Bia.salario = 20000

        gerenciadorBonificacao.Registrar(Bia)

        MsgBox("O salário do " + Carlos.nome + " é " + Carlos.salario.ToString)
        MsgBox("A bonificação de " + Carlos.nome + " será de " +
               Carlos.GetBonificacao().ToString)

        MsgBox("O salário da " + Bia.nome + " é " + Bia.salario.ToString)
        MsgBox("A bonificação da " + Bia.nome + " será de " +
               Bia.GetBonificacao().ToString)

        MsgBox("O valor total de bonificação a ser pago será de " +
               gerenciadorBonificacao.getBonificacao.ToString)

    End Sub

```

```
End Sub
```

```
End Class
```

16) Salve, execute, clique no bot  o e veja as janelas serem exibidas.