

Tarefa Orientada 6

Código do formulário frmCalcularDesvalorização

Objectivos:

- Desenvolver código para responder a eventos
- Trabalhar com caixas de combinação e com caixas de listagem

Nesta tarefa iremos analisar o código desenvolvido para o formulário *frmCalcularDesvalorização*.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Load* associado ao formulário *frmCalcularDesvalorização*. Este evento ocorre quando o formulário é carregado para a memória principal do computador. O código permite adicionar valores inteiros (1..40) para a caixa de combinação relativa ao tempo de vida do item.

```
Public Class frmCalcularDesvalorização
    Inherits System.Windows.Forms.Form

    Private Sub frmCalcularDesvalorização_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Dim iÍndice As Integer
        For iÍndice = 1 To 40
            cboTempoVida.Items.Add(iÍndice)
        Next iÍndice
    End Sub
```

Neste procedimento utiliza-se o método *Add* para adicionar um item no fim da *ComboBox cboTempoVida*.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Activated* associado ao formulário *frmCalcularDesvalorização*. Este evento ocorre quando o formulário *frmCalcularDesvalorização* se torna no formulário activo. O código permite mostrar uma nova mensagem no painel central *sblblDescrição* da barra de estados do formulário.

```
Private Sub frmCalcularDesvalorização_Activated(ByVal sender As _  
Object, ByVal e As System.EventArgs) Handles Me.Activated  
  
    frmPrincipal.sbPrincipal.Items("sblblDescrição").Text =  
        "Calcular a desvalorização anual de um bem"  
  
End Sub
```

Note-se que por a barra de estados *sbPrincipal* pertencer ao formulário (pai) *frmPrincipal*, e por a propriedade *Text* estar a ser definida a partir do formulário (filho) *frmCalcularDesvalorização*, a instrução de atribuição inicia-se com a identificação do formulário *frmPrincipal* seguido da barra de estados *sbPrincipal*. Depois é utilizada a propriedade *Items* da barra de estados com o valor *sblblDescrição*, de modo a identificar o painel central da barra de estados que irá mostrar o texto indicado na parte direita da instrução de atribuição.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Click* associado ao botão *btnCalcular*.

```
Private Sub btnCalcular_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCalcular.Click

    If ValidarEntradas(txtCustoInicial.Text, 500, 1000000) Then
        If ValidarEntradas(txtValorFinal.Text, 0, CDec(txtCustoInicial.Text)) Then
            If cboTempoVida.SelectedIndex <> -1 Then
                Dim iÍndice As Integer
                lstDesvalorização.Items.Clear()
                For iÍndice = 1 To cboTempoVida.Text
                    lstDesvalorização.Items.Add("Ano " & iÍndice & ": " & FormatCurrency(SYD(txtCustoInicial.Text, txtValorFinal.Text, cboTempoVida.Text, iÍndice)))
                Next iÍndice
                txtCustoInicial.Focus()
            Else
                MessageBox.Show("Deve Seleccionar um valor para o " & "tempo de vida.", "Erro")
                cboTempoVida.Focus()
            End If
        Else
            MessageBox.Show("Deve introduzir um valor " & "entre 0 e o valor do custo inicial.", "Erro")
            txtValorFinal.Focus()
        End If
    Else
        MessageBox.Show("Deve introduzir um número " & "entre 500 e 1000000.", "Erro")
        txtCustoInicial.Focus()
    End If

End Sub
```

Neste procedimento de evento utiliza-se a função *ValidarEntradas* para verificar se todas as entradas contêm valores válidos.

Note-se que na segunda chamada à função *ValidarEntradas* se utilizou a função *CDec* para converter o seu argumento (valor introduzido na caixa de texto relativa ao valor final do bem) para decimal. Isto faz sentido porque o terceiro argumento da função *ValidarEntradas* pode ser do tipo de dados decimal.

Se as entradas forem todas válidas, verifica-se se está algum item seleccionado na caixa de combinação relativa ao tempo de vida do item. Para tal, utiliza-se a propriedade *SelectedIndex* da *ComboBox* *cboTempoVida*. A numeração é feita a partir do zero. O valor -1 indica que não está nenhum item seleccionado.

Se as entradas forem todas válidas e se estiver seleccionado um item na caixa de combinação relativa ao tempo de vida do item, “limpa-se” a caixa de listagem onde vai ser apresentado o resultado e calcula-se a desvalorização que o bem vai sofrer em cada ano de vida.

Para “limpar” a caixa de listagem, usa-se o método *Clear*.

Para calcular a desvalorização que o bem vai sofrer em cada ano de vida, conhecendo o seu valor inicial, o seu valor final e o seu tempo de vida, utiliza-se e a função *SYD*. Pode consultar a ajuda on-line para obter mais informações sobre esta função. O resultado é apresentado para cada ano, em formato monetário, na caixa de listagem *lstDesvalorização*. Note-se que o método *Add* permite adicionar um item no fim da caixa de listagem.

Depois, coloca-se o focus na caixa de texto relativa ao valor inicial do bem, através da utilização do método *focus*.

Caso não tenha sido seleccionado nenhum item na caixa de combinação relativa ao tempo de vida do bem, apresenta-se uma mensagem de erro apropriada e coloca-se o focus nesse controlo.

Sempre que exista uma entrada não válida, apresenta-se a mensagem de erro correspondente, através da utilização do método *Show* da classe *MessageBox*, e define-se o focus para o controlo apropriado, através da utilização do método *focus*.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Click* associado ao botão *btnFechar*.

```
Private Sub btnFechar_Click(ByVal sender As System.Object, ByVal _  
e As System.EventArgs) Handles btnFechar.Click  
  
    txtCustoInicial.Text = ""  
    txtValorFinal.Text = ""  
    cboTempoVida.SelectedIndex = -1  
    lstDesvalorização.Items.Clear()  
    Me.Hide()  
  
End Sub  
  
End Class
```

Neste procedimento começa-se por “limpar” as caixas de texto relativas aos valores inicial e final do bem. Define-se a propriedade *SelectedIndex* da *ComboBox cboTempoVida* com o valor -1, de modo a que não apareça nenhum item seleccionado nesta caixa de combinação. Utiliza-se o método *Clear* para “limpar” a caixa de listagem de apresentação de resultados. Finalmente, utiliza-se o método *Hide* do formulário para esconder o formulário, mas não o retirar da memória. Assim, o formulário pode voltar a ser mostrado quando for executado o método *Show*. Se fosse utilizado o método *Close*, então teria que ser criada uma nova instância do formulário.