

Tarefa Orientada 3

Código do formulário *frmPrincipal*

Objectivos:

- Desenvolver código para responder a eventos
- Mostrar e Fechar formulários

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

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Click* associado ao item *Investimento* do menu *Ficheiro*. Este evento ocorre quando o utilizador seleccionar o item *Investimento* do menu *Ficheiro*. O código permite criar e mostrar uma instância do formulário filho *frmCalcularInvestimento*.

```
Public Class frmPrincipal

    Private Sub mnuFicheiroInvestimento_Click(ByVal sender As _
System.Object, ByVal e As System.EventArgs) Handles _
mnuFicheiroInvestimento.Click
        Dim frmCalcularInvestimento As New frmCalcularInvestimento
        frmCalcularInvestimento.MdiParent = Me
        frmCalcularInvestimento.Show()

    End Sub
```

Na primeira linha, a seguir ao cabeçalho do procedimento de evento, cria-se uma nova instância do formulário *frmCalcularInvestimento*. Para tal, utiliza-se o construtor do respectivo objecto (*New*).

Na segunda linha do procedimento define-se que o formulário actual (*frmPrincipal*), referido pela palavra reservada *Me*, é o formulário pai do formulário *frmCalcularInvestimento*.

Na terceira linha do procedimento utiliza-se o método *Show* do objecto formulário para mostrar o respectivo formulário.

A seguir, apresenta-se o código que irá ser executado quando ocorrer o evento *Click* associado ao item *Desvalorização* do menu *Ficheiro*. Este código permite criar e mostrar uma instância do formulário filho *frmCalcularDesvalorização*.

```
Private Sub mnuFicheiroDesvalorização_Click(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
mnuFicheiroDesvalorização.Click  
  
    Dim frmCalcularDesvalorização As New frmCalcularDesvalorização  
    frmCalcularDesvalorização.MdiParent = Me  
    frmCalcularDesvalorização.Show()  
  
End Sub
```

A análise deste código é análoga à do procedimento anterior.

Devido ao facto de poder estar aberta mais do que uma instância de um formulário filho ao mesmo tempo, não é possível referir o formulário pelo seu nome. Pode-se utilizar a propriedade *ActiveMdiChild* de um formulário pai para referir o formulário filho que tem o *focus*.

A seguir, apresenta-se o código que irá ser executado quando ocorrer o evento *Click* associado ao item *Fechar* do menu *Ficheiro*. Este código permite fechar o formulário (filho) activo, caso exista.

```
Private Sub mnuFicheiroFechar_Click(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
mnuFicheiroFechar.Click  
  
    Dim FormulárioActivo As Form = Me.ActiveMdiChild  
    If FormulárioActivo IsNot Nothing Then  
        FormulárioActivo.Close()  
    End If  
  
End Sub
```

Na primeira linha do procedimento de evento declara-se uma variável para referenciar o formulário filho que está activo.

Na segunda linha do procedimento verifica-se se existe algum formulário filho activo. A palavra reservada *Nothing*, quando associada a uma variável

que referencia um objecto, significa que a variável não está a referenciar nenhum objecto.

Na terceira linha do procedimento utiliza-se o método *Close* para fechar o formulário (filho) activo.

A seguir, apresenta-se o código que irá ser executado quando ocorrer o evento *Click* associado ao item *Sair* do menu *Ficheiro*. Este código permite fechar a aplicação.

```
Private Sub mnuFicheiroSair_Click(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) Handles mnuFicheiroSair.Click  
  
    Me.Close()  
  
End Sub
```

Neste procedimento, é utilizado o método *Close* do formulário pai para sair da aplicação.

A seguir, apresenta-se o código que irá ser executado quando ocorrer o evento *Click* associado ao item *BarraFerramentas* do menu *Ver*. Este código permite tornar visível ou invisível a barra de ferramentas no formulário *frmPrincipal*.

```
Private Sub mnuVerBarraFerramentas_Click(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
mnuVerBarraFerramentas.Click  
  
    If mnuVerBarraFerramentas.Checked Then  
        tbfrmPrincipal.Visible = True  
    Else  
        tbfrmPrincipal.Visible = False  
    End If  
  
    'tbfrmPrincipal.Visible = mnuVer.Checked  
  
End Sub
```

Neste procedimento é utilizada a propriedade *Checked* do item *BarraFerramentas* do menu *Ver* para testar se o item está, ou não, seleccionado. A propriedade *Visible* do objecto *tbfrmPrincipal* é usada para tornar visível ou invisível a barra de ferramentas no formulário.

A seguir, apresenta-se o código que irá ser executado quando ocorrer o evento *Click* associado aos itens *Cascada*, *Horizontal* e *Vertical* do menu *Janela*. Este código permite determinar o modo de apresentação dos formulários filho que estão abertos.

```
Private Sub mnuJanelaCascada_Click(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) Handles mnuJanelaCascada.Click  
  
    Me.LayoutMdi (MdiLayout.Cascade)  
  
End Sub  
  
Private Sub mnuJanelaHorizontal_Click(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) Handles mnuJanelaHorizontal.Click  
  
    Me.LayoutMdi (MdiLayout.TileHorizontal)  
  
End Sub  
  
Private Sub mnuJanelaVertical_Click(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) Handles mnuJanelaVertical.Click  
  
    Me.LayoutMdi (MdiLayout.TileVertical)  
  
End Sub
```

Nestes procedimentos é utilizado o método *LayoutMdi* do formulário *frmPrincipal* para determinar o modo de apresentação dos seus formulários filho que estão abertos. Para tal, são usados os membros da enumeração (*enumeration*) *MdiLayout*.

Se definir a propriedade *MdiWindowListItem* do controlo *MenuStrip* (*MenufrmPrincipal*) com o nome do menu *mnuJanela*, este menu irá também mostrar uma lista de todos os formulários filho abertos.

A seguir, apresenta-se o código que irá ser executado quando ocorrer o evento *Click* associado aos botões *Investimento* e *Desvalorização* da barra de ferramentas do formulário *frmPrincipal*. Este código permite criar e mostrar uma instância dos formulários filho *frmCalcularInvestimento* e *frmCalcularDesvalorização*.

```
Private Sub tbbtnInvestimento_Click(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
tbfrmPrincipalbtnInvestimento.Click  
  
    mnuFicheiroInvestimento.PerformClick()  
  
End Sub  
  
Private Sub tbbtnDesvalorização_Click(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
tbfrmPrincipalbtnDesvalorização.Click  
  
    mnuFicheiroDesvalorização.PerformClick()  
  
End Sub
```

Dado que os botões *tbfrmPrincipalbtnInvestimento* e *tbfrmPrincipalbtnDesvalorização* da barra de ferramentas providenciam a mesma funcionalidade que os itens *mnuFicheiroInvestimento* e *mnuFicheiroDesvalorização* do menu *mnuFicheiro*, utiliza-se simplesmente o método *PerformClick* dos itens *mnuFicheiroInvestimento* e *mnuFicheiroDesvalorização*. Como resultado irá ser executado o código associado ao evento *Click* desses itens. Deste modo, não temos que duplicar o código para efectuar as mesmas funcionalidades.

A seguir, apresenta-se o código que irá ser executado em resposta ao evento *Tick* associado ao controlo *Timer* com o nome *TimerfrmPrincipal*. O controlo *Timer* causa este evento nos intervalos determinados pelos valores (em milissegundos) definidos na propriedade *Interval*. Deste modo, a informação da barra de estados é actualizada periodicamente.

```
Private Sub Timer1_Tick(ByVal sender As Object, ByVal e As _  
System.EventArgs) Handles TimerfrmPrincipal.Tick  
  
    sbPrincipal.Items("sblblData").Text = _  
DateTime.Today.ToShortDateString  
  
    sbPrincipal.Items("sblblHoras").Text = _  
DateTime.Now.ToShortTimeString  
  
End Sub  
  
End Class
```

Para poder trabalhar com datas e horas em VB utiliza-se a estrutura *DateTime*.

A propriedade *Now* da estrutura *DateTime* devolve a data e a hora actual.

A propriedade *Today* da estrutura *DateTime* devolve a data actual.

Para formatar datas e horas, pode utilizar-se as propriedades *ToLongDateString*, *ToShortDateString*, *ToLongTimeString* e *ToShortTimeString* da estrutura *DateTime*.

O método *ToLongDateString* converte o valor da estrutura *DateTime* para uma *string* que inclua a designação do dia da semana, a designação para o mês, o dia do mês e o ano.

O método *ToShortDateString* converte o valor da estrutura *DateTime* para uma *string* que inclua os números para mês, dia e ano.

O método *ToLongTimeString* converte o valor da estrutura *DateTime* para uma *string* que inclua as horas, os minutos e os segundos.

O método *ToShortTimeString* converte o valor da estrutura *DateTime* para uma *string* que inclua as horas e os minutos.

Note-se, contudo, que o formato que é utilizado para a data e para as horas depende das definições especificadas nas definições regionais do computador.

Assim, a primeira instrução deste procedimento permite definir a propriedade *Text* do item/panel *SblblData* da barra de estados *sbPrincipal* com a data actual no formato numérico abreviado.