

# Visual Basic for Applications

## ● **Introdução**

- É uma linguagem de programação baseada na conhecida linguagem BASIC
- Está concebida para funcionar em conjunto com diferentes aplicações de forma a potenciar a robustez das mesmas
- Enquadra-se nos ambientes de programação baseados no processamento de sequência de eventos (*event-driven programming*)

## ● **História**

- Em 1994, foi inicialmente integrado no Excel 5 e a partir daí a sua expansão para outras aplicações foi gradual
- Em 1997, com a saída do Office 97 a Microsoft concretizou um dos seus grandes objectivos, ter um ambiente de programação completamente integrado nos seus quatro produtos mais famosos: Word, Excel, Access e PowerPoint
- Actualmente, o VBA é já por si só um produto independente que outras companhias podem adoptar e incorporar nas suas aplicações

# Variáveis I

- ***Para que servem?***
  - Servem para guardar valores temporariamente em memória
- ***Declaração explícita de variáveis***
  - Declarar uma variável VAR: ***Dim VAR***
  - Declarar uma variável VAR como sendo do tipo TYPE: ***Dim VAR As TYPE***
  - Variáveis sem declaração de tipo têm por defeito o tipo ***Variant***
- ***Declaração implícita de variáveis***
  - Possibilidade de não declarar variáveis
  - Variáveis não declaradas têm por defeito o tipo ***Variant***
  - Não permitir o uso de variáveis implícitas: ***Option Explicit***
- ***Visibilidade e longevidade de uma variável***
  - Visível em todos os módulos e durante toda a execução: ***Public***
  - Visível apenas no próprio módulo e durante toda a execução: ***Private***
  - Visível apenas no próprio procedimento e durante a sua execução: ***Dim***

# Variáveis II

- **Tipo de variáveis**

- **Variant** tipo genérico
- **Boolean** True ou False
- **Byte** 0 até 255
- **Integer** -32.768 até 32.767
- **Long** -2.147.483.648 até 2.147.483.647
- **Single** -3,402823E38 até -1,401298E-45 (para valores negativos)  
1,401298E-45 até 3,402823E38 (para valores positivos)
- **Double** -1,79769313486232E308 até -4,94065645841247E-324 (negativos)  
4,94065645841247E-324 até 1,79769313486232E308 (positivos)
- **Currency** -922.337.203.685.477,5808 até 922.337.203.685.477,5807
- **Decimal** +/-79.228.162.514.264.337.593.543.950.335 (sem casas decimais)  
+/-7,9228162514264337593543950335 (com casas decimais)
- **Date** 1 de Janeiro de 100 até 31 de Dezembro de 9999
- **String** 1 até aproximadamente 2 bilhões de caracteres (65.400 se tamanho fixo)
- **Object** tipo genérico para referência a objectos

# Operadores I

- **Aritméticos**

**+** (adição)

**/** (divisão)

**Mod** (resto da divisão)

**-** (subtração e negação)

**\** (divisão inteira)

**\*** (multiplicação)

**^** (exponenciação)

- **Relacionais**

**=** (igual a)

**<>** (diferente de)

**Like** (como)

**>** (maior que)

**>=** (maior ou igual)

**<** (menor que)

**<=** (menor ou igual)

- **Lógicos**

**And** (e lógico)

**Imp** (implicação)

**Or** (ou lógico)

**Xor** (ou exclusivo)

**Not** (negação)

**Eqv** (equivalência lógica)

- **Texto**

**&** (concatenação)

# Operadores II

- **Precedências**

<u>Operador</u>	<u>Operação</u>
<b>^</b>	exponenciação
<b>-</b>	negação
<b>*, /</b>	multiplicação e divisão
<b>\</b>	divisão inteira
<b>Mod</b>	resto da divisão
<b>+, -</b>	adição e subtração
<b>&amp;</b>	concatenação
<b>=, &gt;, &lt;, &lt;&gt;, &gt;=, &lt;=, Like</b>	comparação
<b>And, Or, Not, Imp, Xor, Eqv</b>	lógica

- **Outros caracteres**

,

:

—

comentários

múltiplas instruções na mesma linha

uma instrução em múltiplas linhas

- **Caixas de mensagem**

**MsgBox** (mensagem)

**InputBox** (mensagem)

# Procedimentos I

- ***Para que servem?***

- Quando uma dada sequência de instruções (tarefa) é executada repetidamente em diferentes partes do código, deve ser criado um procedimento que substitua e concentre num único local a sequência de instruções relativa à tarefa a executar
- O uso de procedimentos aumenta a produtividade do programador pois diminui o tamanho global do código a escrever, facilita a edição da sequência de instruções relativa à tarefa em causa, e minimiza a potencial ocorrência de erros

- ***Procedimentos Sub***

- ***São utilizados para executar tarefas independentes***

***[Public | Private] Sub*** nome ( [argumentos] )

***[...]***

***[Exit Sub]***

***[...]***

***End Sub***

# Procedimentos II

- **Procedimentos Function**

- São utilizados para calcular/retornar valores

[**Public** | **Private**] **Function** nome ( [argumentos] ) [**As** tipo]

[...]

[nome = expressão]

[**Exit Function**]

[...]

[nome = expressão]

**End Function**

- **Argumentos**

- Declarar argumentos: argumento [**As** tipo]

Function area (**comp As Integer, alt As Integer**) As Integer

- Passar argumentos: var = area(5, 4)

- Nomear argumentos: var = area (alt:= 4, comp:= 5)

- Argumentos opcionais: **Optional** argumento [**As** tipo] = valor\_por\_defeito

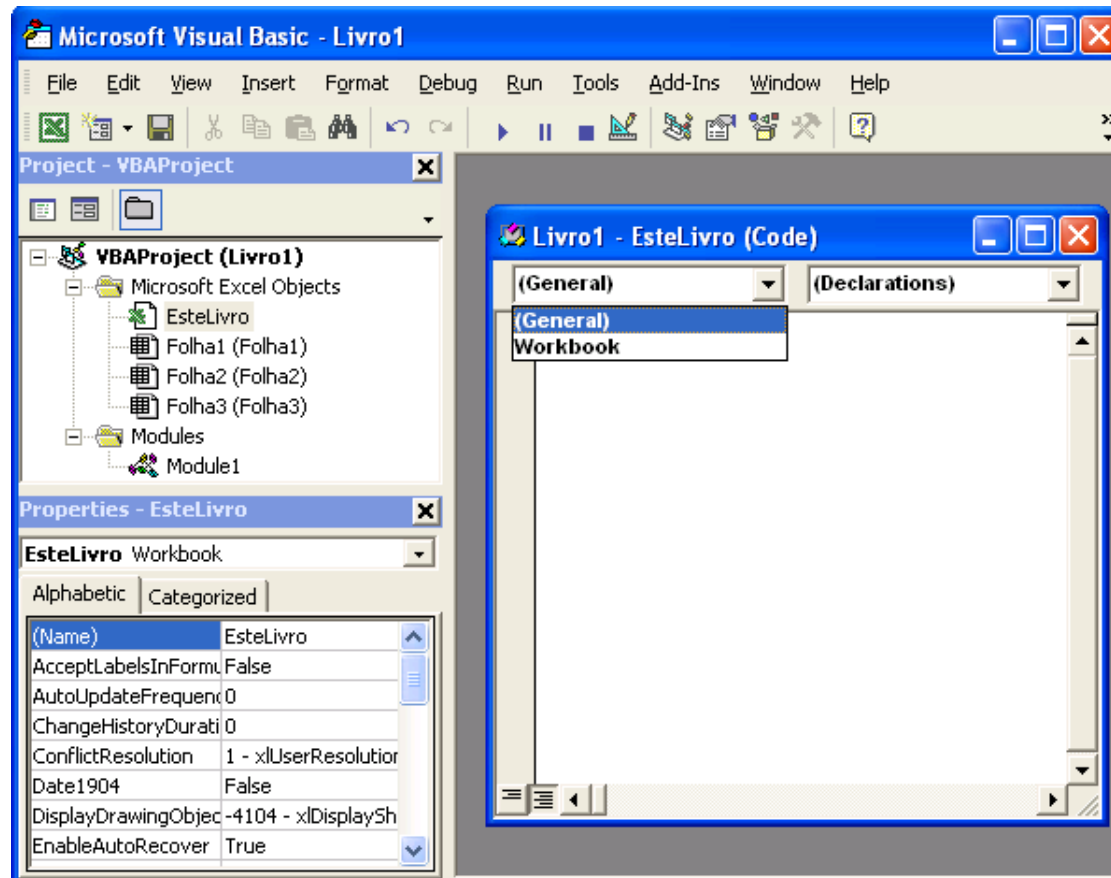
Function area (comp As Integer, **Optional** alt As Integer = 1) As Integer

- A declaração de um argumento opcional implica que os argumentos subsequentes sejam igualmente declarados como opcionais

# VBA e Excel I

- **Editor do Visual Basic**

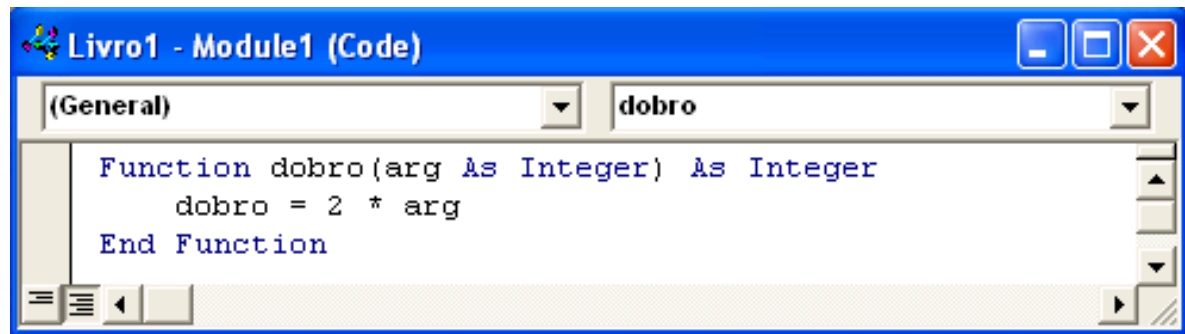
- Menu <Ver> seguido das opções <Barras de ferramentas> e <Visual Basic>
- Janelas de projecto, de propriedades e de código



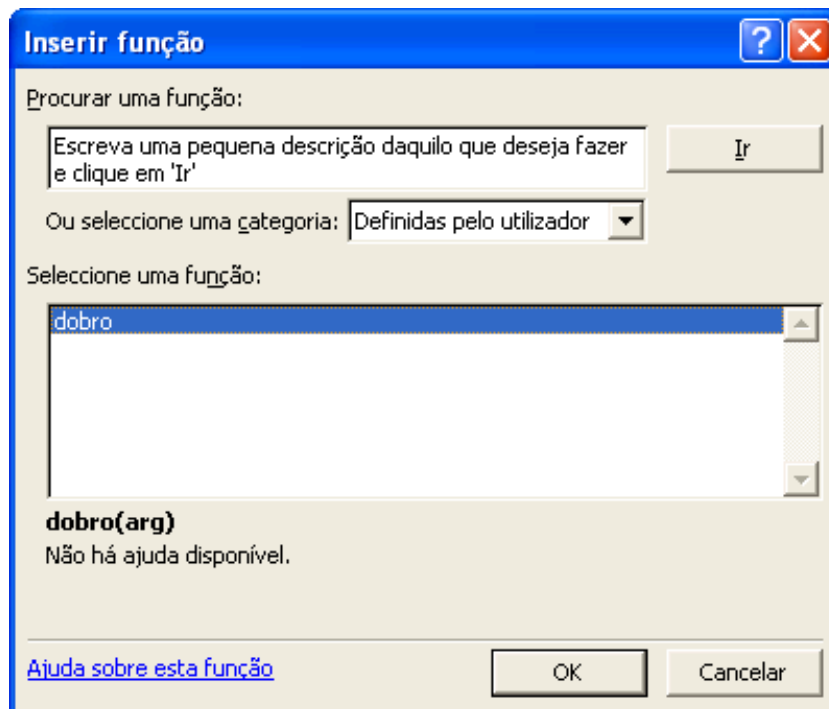


# VBA e Excel II

- **Código VBA como função do Excel**



```
Function dobro(arg As Integer) As Integer
    dobro = 2 * arg
End Function
```

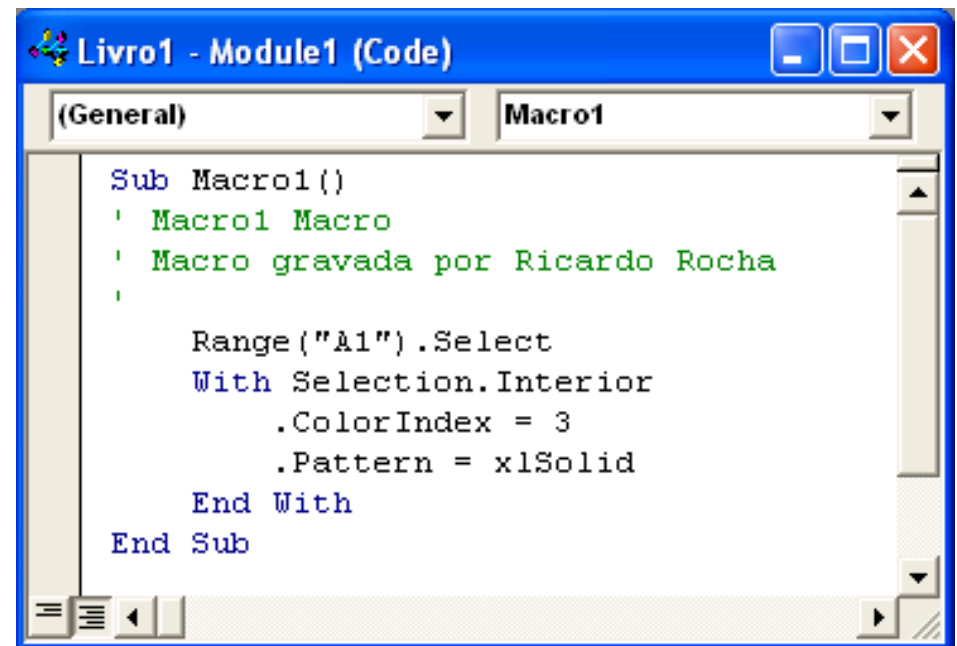
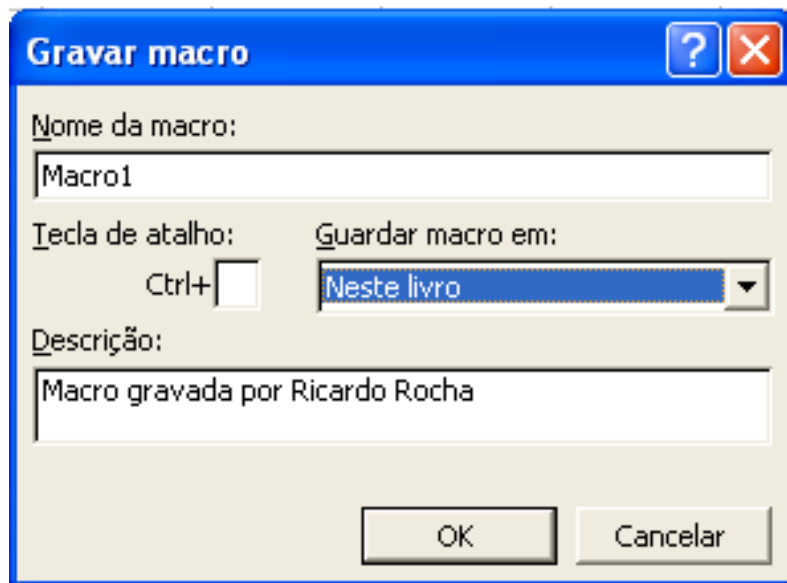
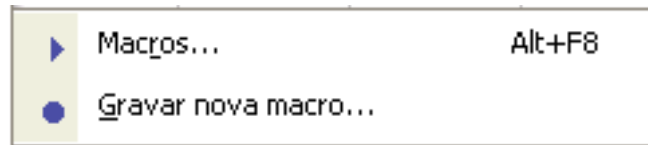


B1		fx =dobro(A1)	
	A	B	
1	10	20	

# VBA e Excel III

## ● **Macros**

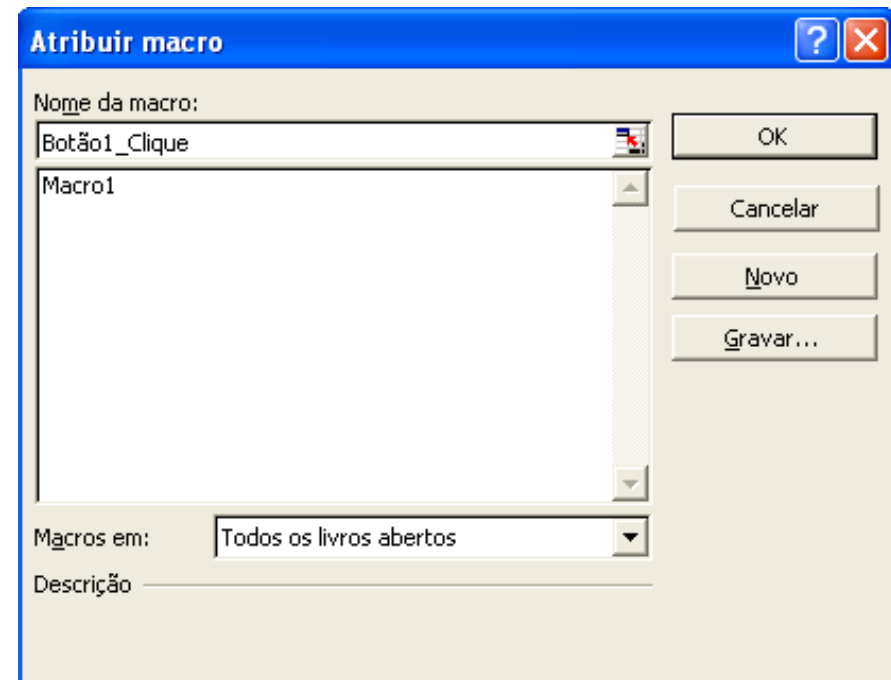
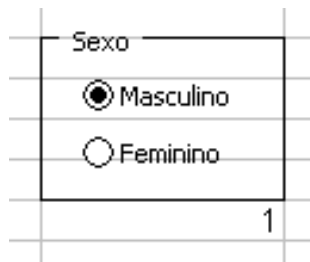
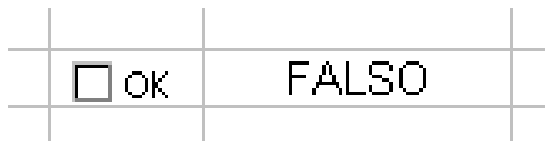
- Menu <Ferramentas> seguido da opção <Macro>
- Gravar macros: botões <Terminar gravação> e <Referência relativa>



# VBA e Excel IV

## ● **Formulários**

- Menu <Ver> seguido das opções <Barras de ferramentas> e <Formulários>
- Ligar células aos controlos: menu <Formatar> seguido da opção <Controlo...> e do separador <Controlo>
- Atribuir macros aos controlos: Menu específico sobre o controlo seguido da opção <Atribuir macro...>



# Estruturas de Decisão

- **Execução condicional**

**If** condição\_1 **Then**

    [...]

...

[**Elseif** condição\_n **Then**

    [...]]

[**Else**

    [...]]

**End If**

- **Múltiplos testes**

**Select Case** expressão\_a\_testar

**Case** lista\_de\_expressões\_1

        [...]

...

    [**Case** lista\_de\_expressões\_n

        [...]]

    [**Case Else**

        [...]]

**End Select**

# Código em Ciclo

## ● Ciclos condicionais

```
Do {While | Until} condição  
    [...]  
    [Exit Do]  
    [...]  
Loop
```

```
Do  
    [...]  
    [Exit Do]  
    [...]  
Loop {While | Until} condição
```

## ● Ciclos numeráveis

```
For contador = início To fim [Step incremento]  
    [...]  
    [Exit For]  
    [...]  
Next [contador]
```

```
For Each elemento In colecção  
    [...]  
    [Exit For]  
    [...]  
Next [elemento]
```

# Objectos I

- **Conceito**

- Qualquer coisa que numa aplicação se pode ver e manipular de algum modo

- **Formas de manipular um objecto**

- Alterar o seu conjunto de **propriedades**
- Activar **métodos** específicos do objecto para executar determinadas tarefas
- Associar procedimentos aos **eventos** que podem ocorrer sobre o objecto

- **Propriedades**

- As propriedades são os atributos que definem as características dos objectos
  - **ActiveCell.Formula = "=A1\*10"**
- Certas propriedades são elas mesmas objectos
  - **ActiveCell.Font.Italic = True**
- Existem propriedades que são só de leitura
  - **col = ActiveCell.Column**

# Objectos II

## ● **Métodos**

- Os métodos são acções que descrevem o que os objectos podem fazer
- São executados sobre os objectos e podem conter ou não argumentos
  - ***ActiveCell.Clear***
  - ***ActiveCell.AddComment "comentário"***

## ● **Eventos**

- Os eventos são algo que acontece aos objectos
- Ocorrem como resultado de acções do utilizador, do sistema ou do próprio código
  - ***Private Sub Workbook\_Open()***
  - ***Private Sub Worksheet\_Activate()***

# Objectos III

- **Hierarquia de objectos**

- **Application**: objecto de topo da hierarquia de objectos
- Caminhos hierárquicos
  - **Application.Workbooks("Livro1").Worksheets("Folha1")**
- Caminhos únicos podem ser abreviados
  - **Worksheets("Folha1")**
  - **ActiveCell**
- **Collection**: tipo de objecto que resulta da colecção de outros objectos

- **Variáveis como objectos**

- Tipo de dados **Object**
- A instrução **Set**



# Objectos Workbook

## ● **Referenciação**

- Referência directa ao nome do livro: ***[Application.]Workbooks("Livro1")***
- Referência ao livro que tem o foco: ***[Application.]ActiveWorkbook***

## ● **Propriedades**

- ***workbook.Name*** – nome do livro (só de leitura)
- ***workbook.Saved*** – indica se existem alterações por guardar

## ● **Métodos**

- ***workbook.Save*** – guarda as últimas alterações
- ***workbook.Close(SaveChanges, FileName)*** – fecha o livro

## ● **Eventos**

- ***Private Sub Workbook\_Open()*** – é executado quando o livro é aberto
- ***Private Sub Workbook\_Activate()*** – é executado quando o livro passa a activo

# Objectos Worksheet

## ● **Referenciação**

- Referência directa ao nome da folha: ***[ActiveWorkbook.]Worksheets("Folha1")***
- Referência à folha que tem o foco: ***[ActiveWorkbook.]ActiveSheet***

## ● **Propriedades**

- ***worksheet.Name*** – nome da folha
- ***worksheet.Visible*** – indica a visibilidade da folha

## ● **Métodos**

- ***worksheet.Delete*** – elimina a folha
- ***worksheet.Calculate*** – recalcula todas as fórmulas presentes na folha

## ● **Eventos**

- ***Private Sub Worksheet\_Activate()*** – é executado quando a folha passa a activa
- ***Private Sub Worksheet\_Calculate()*** – é executado sempre que a folha é recalculada

# Objectos Range

## ● **Referenciação**

- Referência directa a células ou intervalo de células:
  - **[ActiveSheet.]Range("A1")**
  - **[ActiveSheet.]Range("B2:C3")**
  - **[ActiveSheet.]Range("A1,B2:C3")**
- Referência directa a células:
  - **[ActiveSheet.]Cells(1,1)**
  - **[ActiveSheet.]Range("B2:C3").Cells(1,1)**

## ● **Propriedades**

- **range.Value** – indica o valor presente no range
- **range.Formula** – indica a fórmula presente no range (notação standard)
- **range.FormulaLocal** – indica a fórmula presente no range (notação da instalação)
- **range.Offset(RowOffset, ColumnOffset)** – especifica um deslocamento do range

## ● **Métodos**

- **range.Select** – selecciona o range (**Selection** referencia o range seleccionado)
- **range.ClearComments** – remove os comentários das células do range
- **range.ClearContents** – remove o conteúdo das células do range
- **range.ClearFormats** – remove a formatação das células do range
- **range.Clear** – remove tudo o que diz respeito às células do range

# Objectos Control I

- **Propriedades comuns**

- **Name** – nome que identifica o control
- **Left / Top** – posição relativa ao canto superior esquerdo do form
- **Height / Width** – altura e comprimento do control
- **BackColor / ForeColor** – cor do fundo e cor do texto do control
- **Font** – tipo de letra do control
- **Caption** – texto presente no control
- **ControlTipText** – texto de ajuda para quando se mantém o foco sobre o control
- **TabIndex** – ordem de navegação do control (utilizando a tecla *Tab*)
- **TabStop** – se *False* previne a navegação por intermédio da tecla *Tab*
- **Visible** – visibilidade do control
- **Enabled** – se *False* previne o control de receber o foco e responder a eventos
- **Locked** – se *True* previne o utilizador de editar o valor presente no control

# Objectos Control II

- **Command button**

- **Default** – botão seleccionado por defeito quando se abre o formulário
- **Cancel** – botão seleccionado por defeito no caso do formulário ser cancelado
- **CommandButton\_Click( )** – evento associado à acção de premir o botão

- **Text box**

- **Text** – texto presente na caixa de texto
- **PasswordChar** – caracter especial de edição para introdução de passwords
- **TextBox\_Change( )** – ocorre sempre que o conteúdo da caixa é alterado

- **Combo / List boxes**

- **AddItem** – adiciona uma nova entrada à caixa
- **ListIndex** – entrada seleccionada na caixa (a primeira entrada é a zero)
- **ComboBox\_Change( )** – ocorre sempre que o conteúdo da caixa é alterado