

Visual Basic for Applications

● **Introdução**

- É um 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**

- Foi inicialmente integrado como Excel 5 em 1994. A partir daí a sua expansão para outras aplicações foi gradual.
- Foi com a saída do Office 97 em 1997 que a Microsoft concretizou um dos seus grandes objectivos: ter um ambiente de programação completamente integrado entre os seus quatro famosos produtos: 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.

Visual Basic for Applications: # 1

Variáveis

● **Tipos**

- **Byte** 0 até 255
- **Boolean** True ou False
- **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 (até 65.400 se fixo)
- **Variant** qualquer valor numérico ou de texto
- **Object** qualquer referência a um objecto
- **Type** definido pelo utilizador

Visual Basic for Applications: # 2

VariáveisII

- **Declaração de variáveis implícita e explicitamente**
 - Necessidade de declarar ou não as variáveis antes de usá-las
 - Para não permitir o uso de variáveis implícitas: **Option Explicit**
 - Para declarar uma variável VAR: **Dim VAR**
 - Para declarar uma variável VAR com o sendo do tipo TYPE: **Dim VAR As TYPE**
 - Uma variável sem declaração de tipo temporária de tipo **Variant**
- **Visibilidade e longevidade de uma variável**
 - Visível em todos os módulos durante toda a execução: **Public**
 - Visível apenas no próprio módulo durante toda a execução: **Private** ou **Dim**
 - Visível apenas no próprio procedimento durante sua execução: **Dim**
 - Visível apenas no próprio procedimento durante toda a execução: **Static**
- **Valores constantes**
 - Symbolic constants : **Const PI=3,14**
 - Intrinsic constants (definido em bibliotecas do VBA): **Const LEFT_BUTTON=1**
 - System-defined constants : **True; False; Null; Empty; Nothing**

OperadoresI

- **Aritméticos**

+	(adição)	-	(subtração e negação)	*	(multiplicação)
/	(divisão)	\	(divisão inteira)	^	(exponenciação)
Mod	(resto da divisão)				
- **Relacionais**

=	(igual a)	>	(maior que)	<	(menor que)
<>	(diferente de)	>=	(maior ou igual)	<=	(menor ou igual)
Like	(como)	Is	(referência de objectos)		
- **Lógicos**

And	(e lógico)	Or	(ou lógico)	Not	(negação)
Imp	(implicação)	Xor	(ou exclusivo)	Eqv	(equivalência lógica)
- **Texto**

&	(concatenação)				
--------------	----------------	--	--	--	--

OperadoresII

● Precedências

Operador

^

-

***, /**

Mod

+, -

&

=, >, <, <>, >=, <=, Like, Is

And, Or, Not, Imp, Xor, Eqv

Operação

exponenciação

negação

multiplicação e divisão

divisão inteira

resto da divisão

adição e subtração

concatenação

comparação

lógica

● Outros caracteres

Rem ou ' (comentários)

: (múltiplas instruções em uma linha)

_ (uma instrução em múltiplas linhas)

Visual Basic for Applications: # 5

Procedimentos

● Procedimentos Sub

[Public | Private][Static] Sub nome [(lista_de_argumentos)]

[...]

[ExitSub]

[...]

EndSub

● Procedimentos Function

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

[...]

[nome=expressão]

[ExitFunction]

[...]

[nome=expressão]

EndFunction

Visual Basic for Applications: # 6

Argumentos

- **Declarar argumentos**

[*Optional*][*ByVal* | *ByRef*][*ParamArray*]argumento[()][*As* tipo][=valor_por_defeito]
Functionarea (*comp As Integer, alt As Integer*)As Integer

- **Passar argumentos**

- Correctamente: a= area(5,4) area 5,4 Callarea (5,4)
- Erradamente: a= area 5,4 area (5,4) Callarea 5,4

- **Nomear argumentos**

- areaalt :=4, comp:=5 a= area (alt:=4, comp:=5)

- **Opções**

- **Optional**:indicaqueoargumentonãoné necessário.
- **ByVal**:indicaqueoargumentoépassadoporvalor.
- **ByRef**:indicaqueoargumentoépassadoporreferência(pordefeito, osargumento são passadosporreferência).
- **ParamArray**:permiteaceitarum númeroarbitráriodeargumentos.

Arrays

- **Declarar arrays**

- **Dim** nome_array(n_elementos) **As** tipo
- **Option Base1**
- **Dim** nome_array(limite_inferior **To** limite_superior) **As** tipo

- **Preencher arrays**

- Atribuição directaporelemento
- Usandoafunção **Array**

- **Utilizar arrays commaisdoqueumadimensão**

- **Dim** nome_array(n_elementos,n_elementos,...) **As** tipo

- **Arrays dinâmicos**

- Declaração **ReDim**
- Declaração **Preserve**
- Limitesinferioresuperiordeum array: **LBound** e **UBound**

EstruturasdeDecisão

● Execuçãocondicional

If condição *Then*

[...]

[*Elseif* n_condição *Then*

[...]]

...

[*Else*

[...]]

EndIf

IIf (condição,valor_se_verdade,valor_se_falso)

● Múltiplostestes

Select Case expressão_a_testar

[*Case* n_lista_de_expressões

[...]]

...

[*Case Else*

[...]]

End Select

Choose (índice,valor_1,...,valor_n)

Switch (expr_1,valor_1,..., expr_n,valor_n)

Visual Basic for Applications:# 9

CódigoemCiclo

● Cicloscondicionais

Do [{ *While* | *Until*}condição]

[...]

[*Exit Do*]

[...]

Loop

Do

[...]

[*Exit Do*]

[...]

Loop [{ *While* | *Until*}condição]

While condição

[...]

Wend

● Ciclosnumeráveis

For contador=início *To* fim[*Step* incremento]

[...]

[*Exit For*]

[...]

Next [contador]

For Each elemento *In* grupo

[...]

[*Exit For*]

[...]

Next [elemento]

Visual Basic for Applications:# 10

Funções Básicas I

● Conversão de dados

CBool (expressão)

CCur (expressão)

Cdbl (expressão)

CLng (expressão)

CVar (expressão)

CByte (expressão)

CDate (expressão)

CInt (expressão)

CSng (expressão)

CStr (expressão)

● Testes sobre dados

IsArray (variável)

IsEmpty (variável)

IsNull (variável)

TypeName (variável)

IsDate (expressão)

IsNumeric (expressão)

IsMissing (argumento)

VarType (variável)

● Caixas de mensagem

MsgBox (mensagem[,...])

InputBox (mensagem[,...])

Visual Basic for Applications: # 11

Funções Básicas II

● Manipulação de strings

Asc (string)

LCase (string)

Len (string | variável)

Left (string, comprimento)

Trim (string)

Str (número)

InStr ([início,] string_geral, string_procura)

Chr (código)

UCase (string)

Space (número)

Mid (string, início[, comprimento])

LTrim (string)

Val (string)

StrComp (string1, string2[, método])

● Manipulação de data e horas

Date

Date\$

DateSerial (ano, mês, dia)

Time

Time\$

TimeSerial (hora, minuto, segundo)

Now

Year (data)

Month (data)

Day (data)

Weekday (data)

Hour (hora)

Minute (hora)

Second (hora)

Visual Basic for Applications: # 12

VBAeExcell

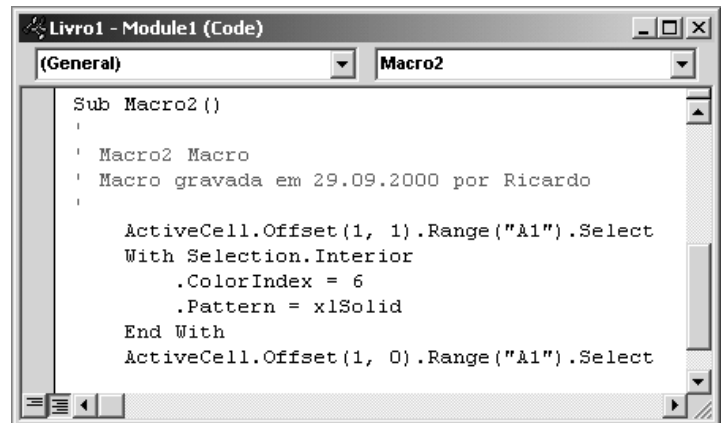
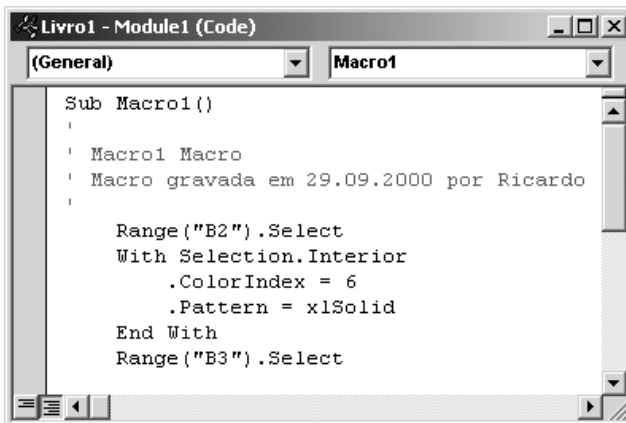
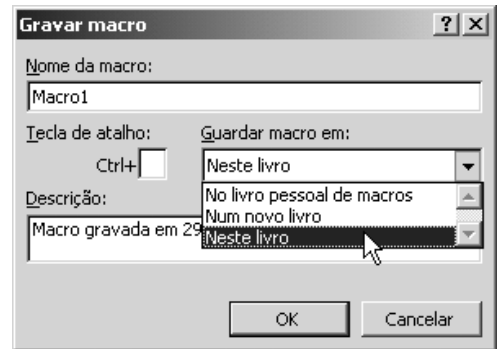
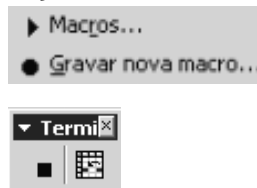
- **Macros**

- Menu<Ferramentas>seguidodaopção <Macro>

- **Gravarmacros**

- Botões<Terminargravação>e <Referênciarelativa>

- **Executareeditarmacros**



Visual Basic for Applications:# 13

VBAeExcell

- **Formulários**

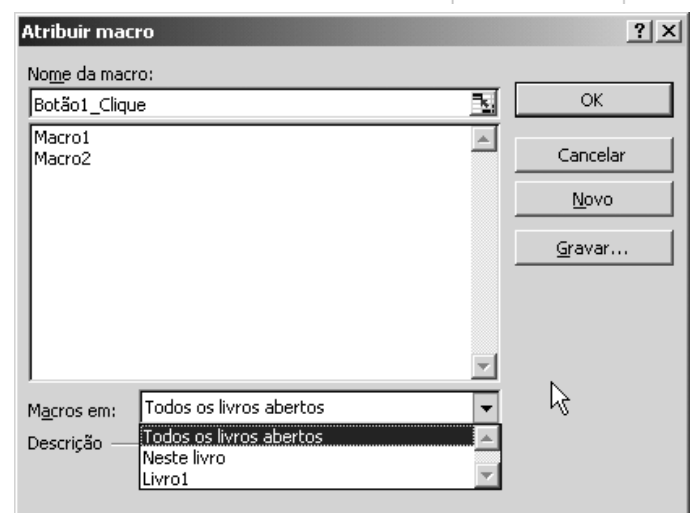
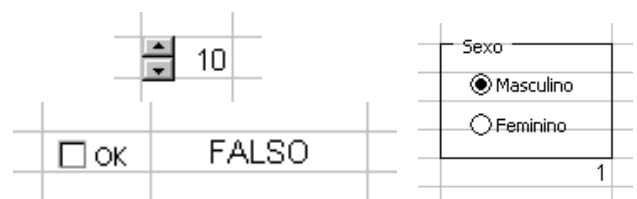
- Menu<Ver>seguidodas opções<Barrasde ferramentas>e<Formulários>

- **Ligarcélulasaoscontroles**

- Menu<Formatar>seguidoda opção<Controlo...>edo separador<Controlo>

- **Atribuirmacroaoscontroles**

- Menuespecíficosobreo controloseguidodaopção <Atribuirmacro...>

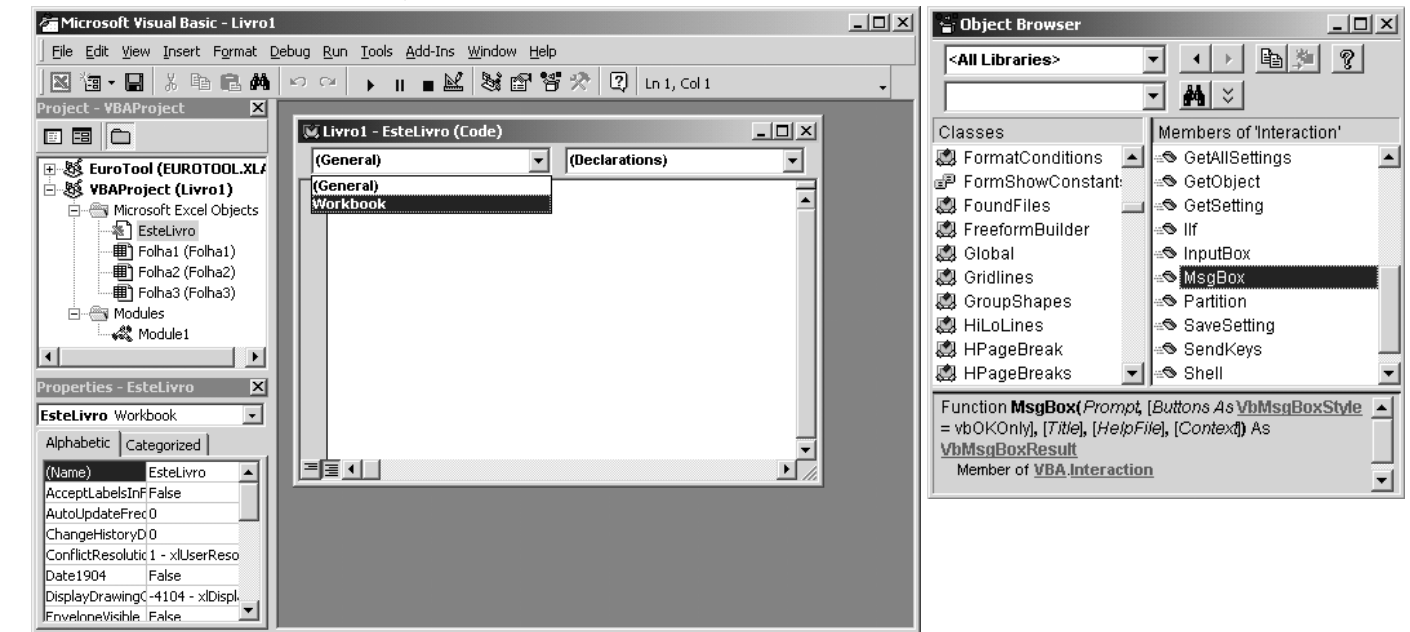


Visual Basic for Applications:# 14

VBAeExcelIII

● *Editor do Visual Basic*

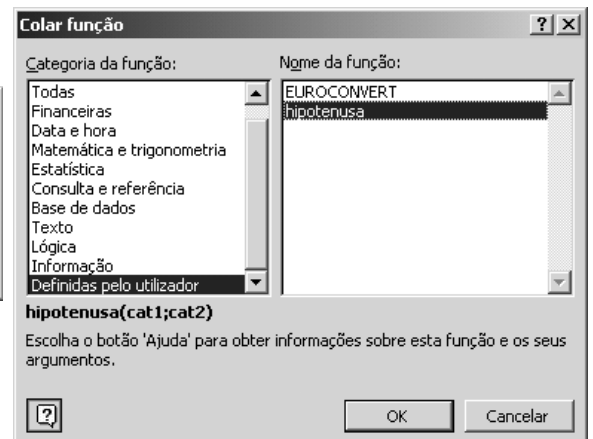
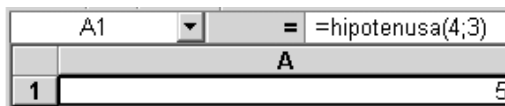
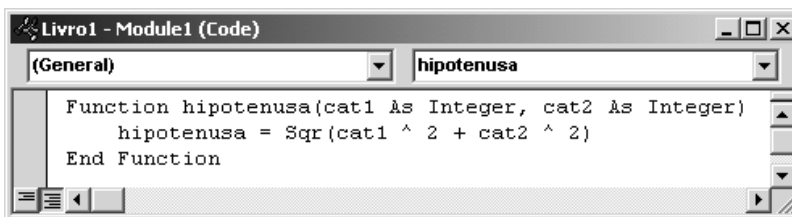
- Janelado projecto, janelade propriedadese janeladecódigo
- Localizador de objectos



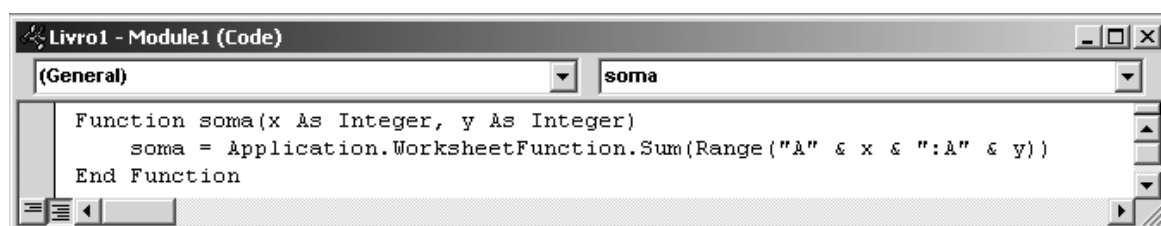
Visual Basic for Applications:# 15

CódigoVBAeFunçõesdoExcel

● *CódigoVBAcomofunçãodoExcel*



● *FunçõesdoExcelcomcódigoVBA*



Visual Basic for Applications:# 16

ObjectosI

- **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 um **método** específico do objecto para executar uma dada tarefa
 - Definir um procedimento que é executado sempre que um dado **evento** ocorrer sobre o objecto
- **Classes e instâncias**
 - **Classe de objectos** : conjunto de objectos com as mesmas propriedades, com os mesmos métodos e que respondem aos mesmos eventos
 - **Instância**: todo o objecto particular de uma classe
- **Hierarquia de objectos**
 - Objecto mais geral: **Application**
 - Conceito de **Collection**
 - Caminhos hierárquicos
 - **Application.Workbooks("Livro1").Worksheets("Folha1")**
 - **Application.Forms("Formulário1")**

Visual Basic for Applications:# 17

ObjectosII

- **Propriedades**
 - As propriedades são os atributos que definem as características dos objectos
 - Propriedades que referenciam objectos
 - **ActiveCell.Font.Italic**
 - Consultar e alterar propriedades
 - **var = ActiveCell.Font.Italic**
 - **ActiveCell.Font.Italic = True**
 - Propriedades só de leitura
- **Métodos**
 - Os métodos são acções que descrevem o que os objectos podem fazer
 - São executados sobre os objectos e podem conter ou não argumentos
 - **Application.CloseCurrentDatabase**
 - **Combo1.Add "novo item"**
- **Eventos**
 - Os eventos são algo que acontece aos objectos
 - Ocorrem como resultado de acções do utilizador, do sistema ou do próprio código
 - **Private Sub Form_Load()**

Visual Basic for Applications:# 18

ObjectosIII

- **Variáveis como objectos**
 - Tipodados **Object**
 - A instrução **Set**
 - A constante **Nothing**
 - O operador **Is**

- **Múltiplas acções sobre um objecto**
 - With** objecto
 - [...]
 - EndWith**

- **Referenciar instância corrente de uma classe**
 - A variável **Me**

Visual Basic for Applications:# 19

Objectos Workbook

- **Referenciação**
 - **[Application.]Workbooks(1)** **[Application.]Workbooks("Livro1")**
 - **[Application.]ActiveWorkbook** **[Application.]ThisWorkbook**
- **WorkbooksCollection**
 - **[Application.]Workbooks.Add** **[Application.]Workbooks.Open(FileName)**
 - **[Application.]Workbooks.Close** **[Application.]Workbooks.Count**
- **Propriedades**
 - **workbook.FullName** **workbook.Name**
 - **workbook.Path** **workbook.Saved**
- **Métodos**
 - **workbook.Activate** **workbook.Close(SaveChanges, FileName)**
 - **workbook.Save** **workbook.SaveAs(FileName)**
 - **workbook.Protect(Password, Structure, Windows)** **workbook.Unprotect(Password)**
- **Eventos**
 - **PrivateSubWorkbook _Activate()**
 - **PrivateSubWorkbook _Open()**
 - **PrivateSubWorkbook _Newsheet(ByValSh As Object)**
 - **PrivateSubWorkbook _BeforeSave(ByValSaveAsUI As Boolean, Cancel As Boolean)**

Visual Basic for Applications:# 20

Objectos Worksheet

● Referenciação

- *Sheets, Worksheets e Charts*

- *[ActiveWorkbook.]Worksheets(1)* *[ActiveWorkbook.]Worksheets("Folha1")*

- *[ActiveWorkbook.]Worksheets(Array("Folha4","Folha5"))*

- *[ActiveWorkbook.]ActiveSheet*

● WorksheetsCollection

- *[ActiveWorkbook.]Worksheets.Add(Before, After, Count)*

- *[ActiveWorkbook.]Worksheets.Count*

● Propriedades

- *worksheet.Name*

worksheet.Visible

● Métodos

- *worksheet.Activate*

worksheet.Copy(Before, After)

- *worksheet.Move(Before, After)*

worksheet.Delete

- *worksheets.FillAcrossSheets(Range, Type)*

worksheet.Calculate

- *worksheet.Protect(Password, DrawingObjects, Contents)*

● Eventos

- *PrivateSubWorksheet _Calculate()*

- *PrivateSubWorksheet _Change(ByValTarget AsExcel.Range)*

Visual Basic for Applications:# 21

ObjectosRange

● Referenciação

- *[ActiveSheet.]Range("A1")*

[ActiveSheet.]Range("A1:B5")

- *[ActiveSheet.]Range("Lucro")*

[ActiveSheet.]Range("A1","B5")

- *[ActiveSheet.]Cells(1,2)*

[ActiveSheet.].Range("B2:C5").Cells(2,1)

● Propriedades&Métodos

- *range.Row*

range.Column

- *worksheet/range.Rows(Index)*

worksheet/range.Columns(Index)

- *range.Value*

range.Formula

- *range.Address(RowAbsolute, ColumnAbsolute)*

range.Select

- *range.Offset(RowOffset, ColumnOffset)*

range.Count

- *range.Copy(Destination)*

range.Cut(Destination)

- *range.Clear*

range.ClearComments

- *range.ClearContents*

range.ClearFormats

- *range.FillDown*

range.FillUp

- *range.FillLeft*

range.FillRight

- *[ActiveWorkbook.]Names.Add(Name, RefersTo)*

name.Delete

Visual Basic for Applications:# 22