

# *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*

- Foi inicialmente integrada com o Excel 5 em 1994 e 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 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

- **Exemplo: dado X quanto é a sua metade mais o seu dobro?**

```
metade = x / 2
```

```
dobro = 2 * x
```

```
resultado = metade + dobro
```

- **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**

# 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
- **Decimal** +/-79.228.162.514.264.337.593.543.950.335 (sem casas decimais)  
+/-7,9228162514264337593543950335 (com casas decimais)
- **Currency** -922.337.203.685.477,5808 até 922.337.203.685.477,5807
- **Date** 1 de Janeiro de 100 até 31 de Dezembro de 9999
- **String** 1 até aproximadamente 2 bilhões de caracteres

# Variáveis III

## ● Exemplos

Option Explicit

```
Dim metade As Decimal
Dim dobro As Decimal
Dim resultado As Decimal
metade = x / 2
dobro = 2 * x
resultado = metade + dobro
```

```
Dim val As Boolean
val = True
```

```
Dim texto As String
texto = "vba"
```

```
Dim aux As Variant
aux = True
aux = "vba"
```

# Operadores I

## ● **Aritméticos**

+ (adição)

/ (divisão)

**Mod** (resto da divisão)

- (subtração e negação)

\ (divisão inteira)

\* (multiplicação)

^ (exponenciação)

## ● **Texto**

& (concatenação)

## ● **Comparação**

= (igual a)

<> (diferente de)

> (maior que)

>= (maior ou igual)

< (menor que)

<= (menor ou igual)

## ● **Lógicos**

**And** (e lógico)

**Or** (ou lógico)

**Not** (negação)

# Operadores II

## ● Precedências

^

-

\*, /

\

Mod

+, -

&

=, >, <, <>, >=, <=

And, Or, Not

Maior precedência



exponenciação

negação

multiplicação e divisão

divisão inteira

resto da divisão

adição e subtração

concatenação

comparação

lógicos

Menor precedência

## ● Outros caracteres

\

:

-

comentários

múltiplas instruções na mesma linha

uma instrução em múltiplas linhas

# Caixas de mensagem

## ● **Caixas MsgBox**

- Forma básica de apresentar uma mensagem ao utilizador
- **MsgBox**(mensagem)

## ● **Caixas InputBox**

- Apresenta uma mensagem numa caixa de mensagem, permite que o utilizador introduza texto, e devolve a sequência de texto introduzida
- **InputBox**(mensagem)

## ● **Exemplo**

```
Dim nome As String
nome = InputBox("Introduza o seu nome...")
MsgBox "Bem vindo " & nome & "!"
```

# 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 da 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 ocorrência de erros

## ● *Tipo de procedimentos*

- **Sub**: são utilizados para executar tarefas independentes
- **Function**: são utilizados para calcular/retornar valores

# Procedimentos II

## ● *Procedimentos Sub*

```
Sub nome ([argumentos])  
    [...]  
    [Exit Sub]  
    [...]  
End Sub
```

## ● *Exemplo*

```
Sub boas_vindas()  
    Dim nome As String  
    nome = InputBox("Introduza o seu nome...")  
    MsgBox "Bem vindo " & nome & "!"  
End Sub
```

# Procedimentos III

## ● *Procedimentos Function*

```
Function nome ([argumentos]) [As tipo]
    [...]
    [nome = expressão]
    [...]
    [Exit Function]
    [...]
    [nome = expressão]
    [...]
End Function
```

## ● *Exemplo*

```
Function area(comp As Integer, alt As Integer) As Integer
    area = comp * alt
End Function
```