

Objectos I

- **Conceito**
 - Qualquer coisa que numa aplicação se pode 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
- **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

Objectos II

● **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`

● **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"`

Objectos III

● **Eventos**

- Os eventos são algo que acontece aos objectos
- Ocorrem em resultado de acções do utilizador, do sistema ou do próprio código
- Os procedimentos correspondentes aos eventos têm todos a seguinte forma:
 - `Private Sub Objecto_Evento(argumentos)`
- É necessário não confundir métodos com eventos. Por exemplo, o método `Activate` quando executado sobre um objecto implica a execução do evento `Activate` do mesmo objecto:
 - `Worksheets(1).Activate`
 - `Private Sub Worksheet_Activate()`

Objectos IV

● **Hierarquia de objectos**

- **Application**: objecto de topo da hierarquia de objectos
- Caminhos hierárquicos
 - `Application.Workbooks("Livro1").Worksheets("Folha1")`
- Caminhos únicos podem ser abreviados
 - `Workbooks("Livro1")`
 - `ActiveCell`

● **Colecção de objectos (Collection)**

- Tipo de objecto que resulta da colecção de outros objectos, não necessariamente do mesmo tipo
- Por exemplo, os objectos `Workbooks` e `Worksheets` são objectos do tipo **Collection** que contêm respectivamente todos os objectos do tipo `Workbook` e `Worksheet` da aplicação

Objectos V

● **Variáveis como objectos**

- Tipo de dados **Object**
- A instrução **Set**
- A constante **Nothing**
- O operador **Is**

● **Exemplo**

```
Dim bool as Boolean
Dim rng1, rng2 As Object
Set rng1 = Cells(1,1)
Set rng2 = Cells(2,2)
bool = rng1 Is rng2
Set rng1 = Nothing
bool = rng1 Is Nothing
```

```
`As Range
```

```
`bool = False
```

```
`para libertar memória
```

```
`bool = True
```

Uma variável do tipo **Object** também fica com o valor **Nothing** se o objecto a ela associado for fechado/terminado/eliminado

Objectos VI

● **A variável implícita Me**

- É automaticamente disponibilizada em todos os módulos de classe
- Referencia a instância da classe onde o código está a ser executado
`MsgBox Me.Name`

● **Múltiplas acções sobre um objecto**

```
With objecto  
[...]  
End With
```

● **Exemplo**

```
With ActiveCell  
  .Formula = "=A1*10"  
  With .Font  
    .Italic = True  
    .Bold = True  
  End With  
End With
```

```
`ActiveCell.Formula = "=A1*10"  
`ActiveCell.Font.Italic = True  
`ActiveCell.Font.Bold = True
```

Objectos Workbook I

● **Referenciação**

- [Application.]: caminho hierárquico implícito
- Workbooks ("Livro1"): referência directa ao nome do livro
- Workbooks (1): referência ao primeiro livro a ser aberto
- ActiveWorkbook: referência ao livro que tem o foco
- ThisWorkbook: referência ao livro onde está o código a ser executado

● **Workbooks Collection**

- Workbooks.Open(FileName): método que abre um livro existente
- Workbooks.Close: método que fecha todos os livros abertos
- Workbooks.Count: propriedade que indica o número de livros abertos

● **Exemplo**

```
ChDir "\My documents\Excel"  
Workbooks.Open "Livro1"
```

Objectos Workbook II

● **Propriedades**

- `workbook.Name`: nome do livro (só de leitura)
- `workbook.Saved`: indica se existem alterações por guardar (pode ser colocada a `True` para evitar o menu de guardar alterações)

● **Métodos**

- `workbook.Activate`: torna o livro activo
- `workbook.Save`: guarda as últimas alterações
- `workbook.Close (SaveChanges)`: fecha o livro
- `workbook.Protect (Password, Structure, Windows)`: protege o livro no que diz respeito à sua estrutura (as folhas não podem ser movidas, ocultas, mostradas, eliminadas, os nomes não podem ser alterados nem podem ser inseridas novas folhas) e/ou às janelas (as janelas não podem ser movidas, redimensionadas, ocultas, mostradas ou fechadas)
- `workbook.Unprotect (Password)`: desprotege o livro

● **Exemplo**

```
ThisWorkbook.Protect "sap", True, False  
ThisWorkbook.Close SaveChanges:=True
```

Objectos Workbook III

● **Eventos**

- **Private Sub Workbook_Open()**: ocorre quando o livro é aberto por acção do método **Open**, ou por intermédio do utilizador
- **Private Sub Workbook_Activate()**: ocorre quando o livro passa a ser o livro activo por acção do método **Activate**, ou por intermédio do utilizador quando abre o livro ou o selecciona
- **Private Sub Workbook_NewSheet(ByVal Sh As Object)**: ocorre quando é inserida uma nova folha (**Sh** referencia a nova folha)

● **Exemplos**

```
Private Sub Workbook_Open()  
    Application.Visible = False  
End Sub  
Private Sub Workbook_Activate()  
    ThisWorkbook.Worksheets("Folha1").Activate  
End Sub  
Private Sub Workbook_NewSheet(ByVal Sh As Object)  
    Sh.Move After:=Worksheets(Worksheets.Count)  
End Sub
```

Objectos Worksheet I

● Referenciação

- [`ActiveWorkbook.`]: caminho hierárquico implícito
- `Worksheets("Folha1")`: referência directa ao nome da folha
- `Worksheets(1)`: referência à primeira folha no separador de folhas
- `ActiveSheet`: referência à folha que tem o foco

● Worksheets Collection

- `Worksheets.Add(Before, After, Count)`: método que adiciona novas folhas. `Before` especifica a folha antes da qual as novas folhas devem ser adicionadas; `After` especifica a folha depois da qual as novas folhas devem ser adicionadas (por defeito são adicionadas antes da folha activa); `Count` indica o número de folhas a adicionar (por defeito uma)
- `Worksheets.Count`: propriedade que indica o número de folhas existentes

● Exemplo

`Worksheets.Add after:=Worksheets(Worksheets.Count)`

Objectos Worksheet II

● **Propriedades**

- `worksheet.Name`: nome da folha
- `worksheet.Visible`: visibilidade da folha
 - `xlSheetVisible`: visível
 - `xlSheetHidden`: oculta (corresponde a ocultar a folha utilizando os menus)
 - `xlSheetVeryHidden`: invisível (só utilizando código é possível reverter o estado)

● **Métodos**

- `worksheet.Activate`: torna a folha activa
- `worksheet.Delete`: elimina a folha
- `worksheet.Copy(Before, After)`: copia a folha
- `worksheet.Move(Before, After)`: move a folha
- `worksheet.Protect(Passwd, DrawingObjs, Contents)`: protege a folha no que diz respeito aos seus objectos gráficos e/ou ao seu conteúdo
- `worksheet.Calculate`: recalcula todas as fórmulas presentes na folha. Útil para fórmulas que utilizam funções voláteis como **AGORA()** e **HOJE()**

Objectos Worksheet III

● **Eventos**

- **Private Sub Worksheet_Change(ByVal Target As Excel.Range):**
ocorre quando uma ou várias células são modificadas

● **Exemplos**

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    For Each cel In Target
        cel.Interior.ColorIndex = 3
    Next
End Sub
```

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
    For Each cel In Target
        cel.Value = ...           `entra em ciclo
    Next
End Sub
```

Objectos Range I

● Referenciação

- [`ActiveSheet.`]: caminho hierárquico implícito
- `Range("B2:C3")`: referência directa ao intervalo B2:C3
- `Range("A1, B2:C3")`: referência directa à célula A1 mais o intervalo B2:C3
- `Range("Lucro")`: referência directa à célula ou intervalo com o nome "Lucro"
- `Range("B2", "C3")`: referência ao intervalo B2:C3 (B2 é o canto superior esquerdo e C3 é o canto inferior direito)
- `Cells(1, 2)`: referência à célula na 1ª linha e na 2ª coluna (célula B1)
- `Range("B2:C3").Cells(1, 2)`: referência à célula na 1ª linha e na 2ª coluna dentro do intervalo B2:C3 (célula C2)

● Exemplo

```
For ano = 1 To 100
    Cells(ano, 1).Value = 1999 + ano
Next ano
```

Objectos Range II

● **Propriedades**

- `range.Row`: número da primeira linha do *range*
- `range.Column`: número da primeira coluna do *range*
- `range.Rows()`: *collection* de todas as linhas do *range*
- `range.Rows(Index)`: linha de ordem `Index` do *range*
- `range.Columns()`: *collection* de todas as colunas do *range*
- `range.Columns(Index)`: coluna de ordem `Index` do *range*

● **Exemplo**

```
Function LastRow(rng As range)
    With rng
        LastRow = .Rows.Count + .Row - 1
        `LastRow = .Rows(.Rows.Count).Row
    End With
End Function
```

Objectos Range III

● **Propriedades**

- `range.Value`: valores das células do *range*

● **Exemplo**

```
Function MaxValue(rng As range)
    aux = rng.Value
    `rng especifica uma célula vazia
    If IsEmpty(aux) Then Exit Function
    `rng especifica uma célula não vazia
    If Not IsArray(aux) Then
        MaxValue = aux : Exit Function
    End If
    `rng especifica mais do que uma célula
    MaxValue = aux(1, 1)
    For Each aux In aux_array
        If aux > MaxValue Then MaxValue = aux
    Next aux
End Function
```

Objectos Range IV

● Propriedades

- `range.Formula` | `FormulaLocal` | `FormulaR1C1` | `FormulaR1C1Local`:
fórmulas das células do *range*

● Exemplos

```
Range("A9").Formula = "=SUM(A1:A8)"
```

```
Range("A9").FormulaLocal = "=SOMA(A1:A8)"
```

`aspas nas fórmulas devem ser duplicadas

```
Range("A1").Formula = "=COUNTIF(B1:B9, "">0"")"
```

```
Range("A1").FormulaLocal = "=CONTAR.SE(B1:B9; "">0"")"
```

`o exemplo que se segue leva a que B9 ← =SUM(B1:B8)

```
Range("A9:B9").Formula = "=SUM(A1:A8)"
```

`o exemplo que se segue leva a que A9 ← =SUM(\$A\$1:\$A\$8)

```
Range("A9").FormulaR1C1 = "=SUM(R1C1:R8C1)"
```

```
Range("A9").FormulaR1C1Local = "=SOMA(L1C1:L8C1)"
```

`o exemplo que se segue leva a que A9 ← =SUM(A1:A8)

```
Range("A9").FormulaR1C1 = "=SUM(R[-8]C:R[-1]C)"
```

Objectos Range V

● Propriedades

- `range.Offset(RowOffset, ColumnOffset)`: desloca o *range* `RowOffset` linhas para cima/baixo e `ColumnOffset` colunas para a direita/esquerda
- `range.Address(RowAbsolute, ColumnAbsolute)`: endereço do *range*
- `range.Count`: número de células do *range*

● Exemplos

```
Range("C5:D10").Select  
Selection.Offset(-4, -2).Select           `A1:B6  
Selection.Offset(3, 1).Range("B2").Select `C5  
  
Set cel = Cells(1, 1)  
MsgBox cel.Address()                     ` $A$1  
MsgBox cel.Address(RowAbsolute:=False)   ` $A1
```

Objectos Range VI

● Métodos

- `range.Select`: selecciona (`Selection` referencia o conjunto seleccionado)
- `range.Calculate`: recalcula fórmulas
- `range.Copy(Destination)`: copia o conteúdo
- `range.ClearComments`: remove comentários
- `range.ClearContents`: remove conteúdos
- `range.ClearFormats`: remove formatações
- `range.Clear`: remove comentários, conteúdos e formatações
- `range.FillDown`: copia o conteúdo da 1ª linha para as restantes células (idêntico a seleccionar a 1ª linha e arrastar com o rato)
- `range.FillUp`: copia o conteúdo da última linha para as restantes células
- `range.FillLeft`: copia o conteúdo da última coluna para as restantes células
- `range.FillRight`: copia o conteúdo da 1ª coluna para as restantes células

● Exemplo

```
Range("A1:B5").Copy Range("E5")  
Range("A1:A10").FillDown
```

'copia para E5:F9