

IMPOSTAZIONI DI EXCEL

Per rendere disponibile l'ambiente di sviluppo di Visual Basic for Application in Excel:

Excel Options / Popular

oppure

Excel Options / Customize ribbon

selezionare:

Show developer Tab in the Ribbon

oppure

Developer

Quando si salva il file

SAVE AS

Save as type

Excel Macro-Enabled Workbook

AMBIENTE DI SVILUPPO DI VBA

Per accedere all'editor di VBA, dalla scheda

Developer

selezionare: **Visual Basic**

La finestra di gestione progetti (**Project-VBAProject**) contiene un diagramma ad albero con le cartelle di lavoro aperte e gli oggetti Excel in esse contenuti (fogli Excel, le stesse cartelle, moduli, form, ecc.)

È raccomandabile scrivere il codice VBA nei moduli e nel caso di codici complessi, utilizzare più moduli separati, usando p. es. i nomi NumNomeModulo.

Per inserire i moduli dalla scheda

Insert

selezionare **Module**

Selezionare con doppio click il modulo per aprire la corrispondente **finestra del codice**.

Il **progetto VBA** indicato da **VBAProject(NomeCartella)** è diviso in due macroaree di elementi: oggetti e moduli.

Tra gli oggetti c'è sempre almeno un foglio di lavoro e la cartella corrente aperta (ThisWorkbook).

Tutti i fogli sono indicati da **Sheet# (NomeFoglio)**

I moduli contengono i programmi in Visual Basic ed è buona tecnica di programmazione suddividere programmi complessi in più moduli.

La finestra proprietà (**Properties-NomeOggetto**) contiene l'elenco di tutte le proprietà dell'oggetto selezionato.

FUNZIONI IN VBA DEFINITE DALL'UTENTE

Esempi di funzioni definite dall'utente

Function funzione1(parametro) As Double

 funzione1 = parametro * 3 + 1

End Function

Function funzione2(parametro)

 If parametro > 3 Then funzione2 = 0 Else funzione2 = parametro

End Function

NB: se si omette As TipoVariabile, la funzione restituisce il tipo Variant

Le funzioni definite dall'utente possono essere utilizzate come tutte le altre funzioni di Excel: possono essere inserite dalla scheda **Formulas** scegliendo **Insert Function**

Si può inserire una descrizione di che cosa fa la funzione:

dalla scheda **Developer**, scegliere **Macro**, scrivere il nome della funzione; mediante il pulsante **Options** si può inserire una descrizione della funzione.

Funzioni in VBA definite dall'utente

Esempio di funzione per calcolare le probabilità di una distribuzione binomiale

Function binomiale(n, p, x)

 binomiale = Application.Combin(n, x) * p ^ x * (1 - p) ^ (n - x)

End Function

Nel codice VBA si possono richiamare funzioni di Excel attraverso l'oggetto Application:

Application.NomeFunzione

La funzione Application.Combin è la funzione di Excel che calcola il coefficiente binomiale.

NB: le funzioni di Visual Basic possono invece essere utilizzate direttamente senza Application.NomeFunzione (es. sqr(), ...)

Funzioni in VBA definite dall'utente

Esempio di funzione con gestione di errore

Function divisione(a, b)

If b = 0 Then

 divisione = CVErr(xlErrDiv0)

Else

 divisione = a / b

End If

Gestione di errori

| | |
|----------------------------------|----------|
| NomeFunzione = CVErr(xlErrNum) | #NUM! |
| NomeFunzione = CVErr(xlErrNull) | #NULLO! |
| NomeFunzione = CVErr(xlErrDiv0) | #DIV/0! |
| NomeFunzione = CVErr(xlErrValue) | #VALORE! |

ALCUNI RICHIAMI SULLE ISTRUZIONI DI VISUAL BASIC

Option explicit

inserita nella prima riga del modulo rende obbligatoria la definizione delle variabili

Dichiarazione di variabili

Dim NomeVariabile As Tipo

IF

If condizione Then istruzione

If condizione Then istruzione1 Else istruzione2

Cicli

Do While test
 istruzioni

Loop

Do
 istruzioni

Loop Until test

For indice=1 to N [Step 1]
 istruzioni

Next indice

Alcuni richiami sulle istruzioni di Visual Basic

Array

Dim NomeVettore(IndiceMassimo) As Tipo

NB: gli indici vanno da 0 a IndiceMassimo, a meno che non si metta l'opzione che stabilisce come indice minimo 1:

Option Base 1

Esempio:

```
Dim vettore(5) As Double
```

```
For i=0 to 5
```

```
    Vettore(i)= ...
```

```
Next i
```

Le funzioni `LBound (NomeVettore)` e `UBound(NomeVettore)`

forniscono rispettivamente l'indice minimo e l'indice massimo del vettore.

Dim NomeVettore(Min To Max) As Tipo

Alcuni richiami sulle istruzioni di Visual Basic

Array multidimensionali

Dim Matrice(MaxRighe, MaxColonne) As Tipo

For i=0 to MaxRighe

....

For j=0 to MaxColonne

....

Array dinamici

Dim VettoreDinamico() As Tipo

...

ReDim VettoreDinamico (NumeroIntero)

Definisce un vettore di NumeroIntero + 1 elementi con indici che vanno da 0 a NumeroIntero.

Le macro subroutines in VBA

LE MACRO SUBROUTINES IN VBA

```
Sub NomeMacro()
```

```
    istruzioni
```

```
End Sub
```

Esempio

```
Sub saluti()
```

```
    MsgBox "ciao a tutti", , "SALUTI"
```

```
End Sub
```

Per eseguire una macro, dalla scheda

Developer

selezionare:

Macro

Run

Le macro subroutines in VBA

Esempio di macro

```
Sub valore_attuale()
```

```
    Dim C As Currency
```

```
    Dim i As Single
```

```
    Dim t As Single
```

```
    Dim v As Currency
```

```
    C = InputBox("inserisci il capitale", "Calcolo valore attuale", 1500#)
```

```
    i = InputBox("inserisci il tasso annuo", "Calcolo valore attuale", 0.02)
```

```
    t = InputBox("inserisci la durata in anni", "Calcolo valore attuale", 3.5)
```

```
    v = Round(Application.PV(i, t, , -C), 2)
```

```
    MsgBox ("il valore attuale di " & C & " al tasso " & i & " è: " & v)
```

```
End Sub
```

& concatena stringhe

Chr(9) inserisce un tabulatore nella stringa

Chr(13) inserisce un ritorno a capo nella stringa

Esercizio: macro per piano di ammortamento, usando Input Box e Message Box

Le macro subroutines in VBA

InputBox function

È una funzione che visualizza una casella di testo in una finestra di dialogo nella quale inserire un dato di input.

Restituisce una stringa con il contenuto della casella di testo.

InputBox (“TestoMessaggio” [, “Titolo finestra”] [, valore di default nella casella di testo])