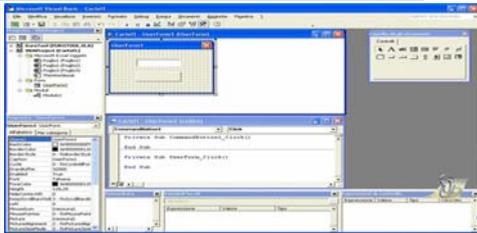


VISUAL BASIC FOR APPLICATION

LA PIATTAFORMA DI SVILUPPO



GLI ELEMENTI DI VBA

- **Finestra di progetto:** mostra l'elenco gerarchico dei progetti aperti e di tutti i moduli di codice e finestre di interfaccia appartenenti ad un determinato progetto. Ad ogni progetto sono associate quattro cartelle a seconda del tipo di modulo che viene sviluppato.
 - Microsoft Excel oggetti: contiene il codice legato ai fogli ed alla cartella di lavoro.
 - Form: contiene le interfaccia utente associate al progetto.
 - Moduli: contiene i moduli di codice generici.
 - Moduli di classe: contiene i moduli per la creazione di nuove classi di oggetti.
- **Finestra Userform (interfaccia utente):** Questa finestra ci permette di creare finestre di interfaccia utente personalizzate da applicare ai programmi VBA, simili in tutto e per tutto a quelle proprie di Microsoft Excel.

GLI ELEMENTI DI VBA

- **Finestra delle proprietà:** mostra l'elenco, alfabetico o per categorie, di tutte le proprietà (altezza, larghezza, colori, font, descrizioni, ecc...) appartenenti ad un controllo contenuto in uno "UserForm" o ad uno "UserForm" stesso.
- **Modulo di codice:** Luogo dove viene scritto il codice Visual Basic. Esistono tre tipi di moduli di codice: generici, di classe e di userform.
- **Casella degli strumenti:** contiene una serie di controlli necessari per costruire l'interfaccia utente. Per spostare i controlli nella "UserForm" basta cliccare sull'oggetto interessato e mantenendo premuto il tasto sinistro trascinarlo sulla finestra Userform.

GLI ELEMENTI DI VBA

- **Finestra immediata;**
- **Finestra Variabili locali;**
- **Finestra espressioni di controllo:** Finestre usate per il Debug (eliminazione degli errori sintattici e logici) del codice.
- **Visualizzatore degli oggetti:** elenco di tutti gli oggetti, metodi, proprietà ed eventi appartenenti ad un controllo o all'applicazione Excel.

LA SINTASSI BASIC

- Affinché le nostre applicazioni funzionino è necessario *inserire il codice Basic*, cioè le istruzioni che indicano al computer quali comandi eseguire ed in che ordine.
- Il codice viene inserito nelle finestre "Modulo di codice" all'interno dell'Editor di "Visual Basic".

CLASSIFICAZIONE DEI DATI

- I dati utilizzati dai moduli di codice durante l'esecuzione dei programmi vengono memorizzati all'interno delle "Costanti" e delle "Variabili".
- Le **variabili** rappresentano dati il cui valore cambia durante l'esecuzione del programma.
- Le **costanti** invece contengono sempre lo stesso valore predefinito durante tutta l'elaborazione.
- La sintassi di variabili e costanti è la seguente:
 - Dim NomeVariabile as TipoVariabile
 - Const NomeVariabile as TipoCostante = ValoreCostante

LA VARIABILE

- Il nome di una variabile non può:
 - Superare i 255 caratteri.
 - Contenere punti o operatori matematici.
 - Non deve essere una parola riservata di Visual Basic, cioè quelle parole che servono per impartire le istruzioni al programma; come ad esempio: Loop, Function, Sub, End, Do, Integer, Case, If, For, Else, Then, Select.

DIVERSI TIPI DI VARIABILI

Tipo	Descrizione
Byte	Accetta numeri da 0 a 255
Integer	Accetta numeri da -32768 a 32767
Long	Accetta numeri da -2147483648 a 2147483647
Single	Accetta numeri da -3,402823E38 a -1,401298E-45 per valori negativi Da 1,401298E-45 a 3,402823E38 per valori positivi
Double	Accetta numeri da -1,79769313486232E308 a -4,94065845841247E-324 per valori negativi Da 4,94065845841247E-324 a 1,79769313486232E308 per valori positivi
Boolean	Accetta solo 2 valori "True" o "False"
String	Accetta da 0 a 65000 caratteri
Date	Date dal 1 gennaio 100 al 31 dicembre 9999
Variant	Se contiene solo numeri: come Double. Se contiene anche caratteri: come String.

DICHIARAZIONE DI VARIABILI

- Anche se non è necessario, è buona norma **dichiarare sempre variabili** e costanti all'interno del codice, questo aiuta a ridurre gli errori di scrittura del codice.
- Quando viene inserita la parola chiave "Option Explicit" in cima al modulo di codice, Visual basic, in caso di mancata dichiarazione di una variabile, darà una segnalazione di errore.

LE ROUTINE

- Con la creazione di routine è possibile dividere il codice in blocchi separati. Questo permette di rendere il programma più leggibile ed individuare facilmente gli errori.
- Utilizzare i moduli permette anche di rendere comune parti di codice a più programmi. In Visual Basic esistono due tipi di routine.
 - Routine Sub: svolgono azioni ma non restituiscono un valore.
 - Routine Function: svolgono azioni e restituiscono un valore.
- NB: un modulo di codice non coincide necessariamente con una routine, in quanto un modulo può contenere più routine contemporaneamente.

ESEMPIO DI ROUTINE

- Sub PrimaRoutine()
- ' La riga successiva visualizza un messaggio
- MsgBox "Ciao Mondo", vbInformation, "Ciao"
- End Sub

VISIBILITA'

- La visibilità di una variabile, o di una costante, definisce la sua disponibilità all'interno del progetto. Si possono dichiarare variabili e costanti a tre livelli:
 1. **Livello locale:** vanno dichiarate all'interno della routine e sono disponibili solo all'interno di essa.
 2. **Livello di modulo:** Vanno dichiarate in testa alla finestra di modulo di codice e sono viste da tutte le routine contenute all'interno del modulo di codice.
 3. **Livello pubblico:** Vanno dichiarate in testa alla finestra di modulo di codice e sono viste da tutte le routine e da tutti i moduli di codice appartenenti al progetto.

ESEMPIO

- 'Le seguenti variabili sono visibili in tutto il progetto
- Public risultato1 As Integer
- Public risultato2 As Integer
- Public sottraendo As Integer
- 'La seguente variabile è visibile a livello di modulo
- Private operatore As Integer
- Sub somma()
 - 'La seguente variabile è visibile solo all'interno della routine
 - Dim Addendo As Integer
 - operatore = 5
 - Addendo = 5
 - risultato1 = Addendo + operatore
 - MsgBox "Il risultato della somma è: " & risultato1, vbInformation, "Somma"
- End Sub

PROSECUZIONE

- Sub sottrazione()
 - 'La seguente variabile è visibile solo all'interno della routine
 - Dim sottraento As Integer
 - sottraento = 2
 - risultato2 = operatore - sottraento
 - MsgBox "Il risultato della sottrazione è: " & risultato2, vbInformation, "Sottrazione"
- End Sub
