

# Corso rapido di Visual Basic

## destinato a chi non sa programmare

*Pensiamo di sviluppare questo minicorso di Visual Basic in tre puntate.*

*Nella prima, questa, motiveremo il corso stesso, che, come risulta dai titoli, è destinato a chi non sa programmare, ed ha quindi un «taglio» molto semplificato. In questa puntata tratteremo i primi rudimenti ed in particolare parleremo dei Control presenti nella Toolbox di Visual Basic. Nella seconda puntata parleremo di argomenti più complessi, anche se fondamentali, come le funzioni di I/O, quelle per intenderci che permettono di interfacciare Visual Basic con un file di dati, l'uso delle Subroutine, l'uso delle Funzioni Personalizzate, la gestione delle Variabili e delle Matrici. In questa seconda puntata un pizzico di conoscenza del vecchio Basic non ci starebbe male.*

*Nella terza puntata vi proporremo, sempre sotto forma di esercizio, una serie di applicazioni più complete, che, avendo pur sempre finalità didattiche, ci permetteranno di mettere insieme, in un unico programma, i vari elementi studiati nelle prime due lezioni*

**di Francesco Petroni**

### Visual Basic Il punto della situazione

Il Visual Basic della Microsoft sta a Windows come il Basic, nelle sue varie versioni, dal Basica in ROM dell'IBM, al Turbo Basic della Borland, al QuickBasic della stessa Microsoft, stanno al DOS.

È un linguaggio di programmazione «all purpose» che serve un po' a tutto e che può essere produttivamente impiegato in applicazioni di complessità bas-

sa o media.

Non entra in concorrenza certo con gli strumenti più evoluti, come i vari C++ e SDK, ma è certamente più facile da maneggiare, anche da un utente che non abbia preconcoscienze di programmazione, ma che abbia una buona conoscenza operativa e concettuale del mondo Windows, e che desideri anche fare qualcosa da sé.

Il prodotto finale della programmazione in Visual Basic è un applicativo Win-

dows facilmente installabile sulle macchine dei destinatari e facilmente usabile, al pari delle normali applicazioni Windows.

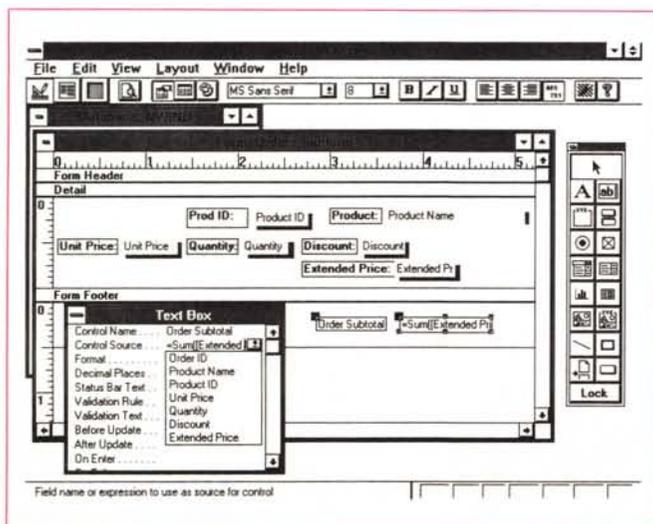
Pur trattandosi di un prodotto giovanissimo ha già un passato ricco di eventi. Ne sono uscite due versioni per Windows, la 1.0 e la 2.0 (quella che useremo), e una, recentissima, per DOS. E ogni versione è stata accompagnata (veramente della versione 2.0 è stato solo annunciato) da specifici Professional Toolkit, che ne allargano le aree di utilizzabilità.

Altro fenomeno connesso con il Visual Basic è stata la nascita di numerose «Libraries» con funzioni aggiuntive, destinate a potenziare i punti deboli del Visual Basic. In particolare ci riferiamo alle funzioni Data Base, che Visual Basic riprende purtroppo integralmente dal vecchio Basic.

Nel corso degli articoli parleremo anche di questo aspetto.

Un aspetto più tecnico che esula dalla finalità del corso, ma che fa «appizzare le orecchie» ai tecnici è costituito dall'esistenza di specifici prodotti che servono per sviluppare applicazioni Visual Basic che interfacciano prodotti di sistema come LAN Manager o SQL Server.

Chi dunque volesse utilizzare in maniera professionale il Visual Basic può già ora sfruttare una dozzina di prodotti, e può consultare una mezza dozzina di



*Figura 1 - MS Visual Basic 2.0 - Analogie con MS Access. Questa immagine è presa dal Microsoft Access che presenta, nell'ambiente Disegno della Form, strumenti di programmazione del tutto analoghi a quelli disponibili in Visual Basic 2.0. Il meccanismo sul quale si appoggia la programmazione è basato sugli oggetti e sulla definizione delle proprietà degli oggetti ed è assolutamente identico nei due prodotti.*

testi, in italiano (il Visual Basic pare invece che non sarà tradotto).

Altro aspetto da non trascurare è che il Visual Basic, con qualche variante, sarà adottato da tutti gli applicativi programmabili della Microsoft. A riprova di quanto detto vi mostriamo subito una videata presa da Microsoft Access (fig. 1) nella quale vediamo sia la Toolbox che la finestra delle Properties di cui dispone ciascun oggetto posto nella Form.

### Prerequisiti

Il programmatore Visual Basic deve conoscere molto bene Windows in quanto buona parte degli oggetti che gestisce Visual Basic sono oggetti Windows, quindi a lui già perfettamente noti. Non perderemo quindi tempo a spiegare cosa è e come funziona una Scroll Bar, una List Box, un Option Button.

Non perderemo neanche tempo a dire perché e per come ad un certo oggetto testuale si può attribuire quel carattere, o quel colore. Si può fare perché è Windows che lo permette, non il Visual Basic.

Una conoscenza di alcune istruzioni del vecchio Basic sarebbe auspicabile, non fosse altro che per il fatto che l'utilizzatore del tutto a digiuno, dovrà comunque imparare una serie di istruzioni fondamentali, che c'erano prima e che ci sono pure adesso. Le principali istruzioni che servono e che comunque descriveremo nei vari esercizi sono:

FOR .. NEXT

IF .. ELSE .. END IF, e le sue varianti  
DO WHILE .. LOOP, e le sue varianti.

La conoscenza di un foglio elettronico aiuterebbe molto, in quanto in Visual Basic sono ripresi numerosi elementi presenti anche nel foglio elettronico. Ad esempio i formati numerici che seguono, nel Visual Basic, la stessa sintassi, o le date che sono gestite e gestibili alla stessa maniera.

### Il nostro obiettivo

Il nostro obiettivo è quello di imparare il Visual Basic imparando non tanto le centinaia di comandi quanto il mecca-

smo fondamentale che sta alla base di tutto, imparato il quale il successivo sfruttamento del prodotto diventa quasi banale.

Una applicazione normale Visual Basic si basa su una o più Form (Maschera, Videata, Dialog Box, chiamatela come volete) in cui vanno disegnati degli Oggetti Windows.

Ogni oggetto ha, intrinsecamente, un certo numero di proprietà, che gli vengono assegnate per il solo fatto che questo oggetto è stato disegnato nella Form, e che il programmatore può tene-

re per buone o può variare.

Compito del programma è quello di far variare qualcuna di queste proprietà al verificarsi di un certo Evento legato ad un altro Oggetto.

Facciamo un esempio. In una Form inseriamo due oggetti. Una «Text Box», il cui contenuto viene, al momento del disegno, posto dal Visual Basic uguale a «text1», e un «Command Button» che all'inizio si chiama Command1.

Possiamo, anche subito dopo aver disegnato ogni oggetto, cambiarne una proprietà. Possiamo ad esempio toglie-

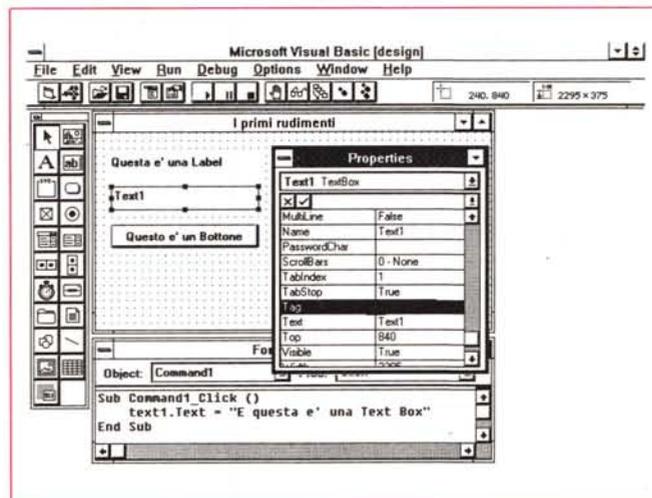


Figura 3 - MS Visual Basic 2.0 - Bottoni.

In questo esercizio studiamo due nuovi Control. Il Check Box, che può assumere due valori, Vero o Falso, e che conseguentemente mostra una croce nella relativa casellina, e l'Option Button, che serve per raggruppare una serie di scelte mutuamente esclusive. In un gruppo di Option Buttons uno solo può avere il valore Vero.

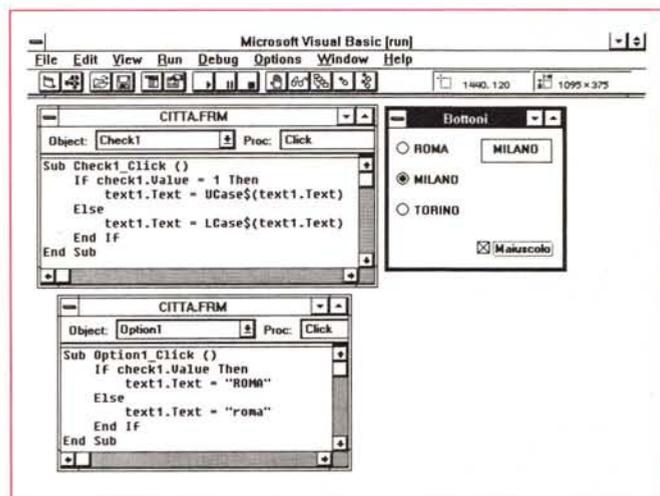


Figura 2 - MS Visual Basic 2.0 - I primi rudimenti.

La nostra teoria è questa. Per riuscire a programmare in Visual Basic non serve studiare tutti i comandi e tutte le istruzioni di programmazione, basta capire alcuni semplici meccanismi legati agli oggetti coinvolti in un programma, le FORM, i CONTROL, le PROPERTIES, gli EVENTS e i «pezzi» di CODICE. Una volta capiti questi si può già affermare di saper programmare in VB.

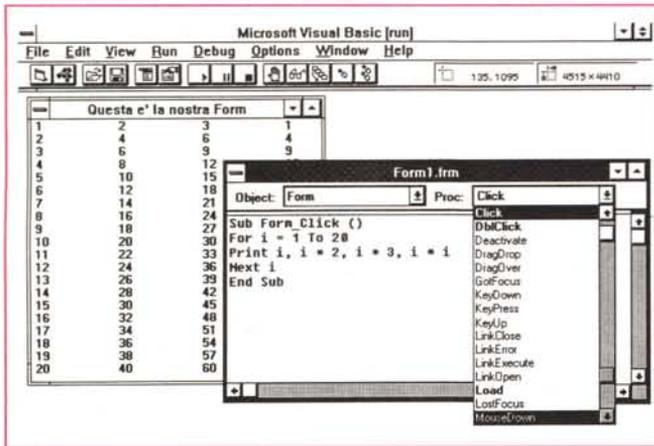


Figura 5 - MS Visual Basic 2.0 - Calcolatrice a Bottoni.

Un primo esercizio classico consiste in una minicalcolatrice. In questo esercizio si «tocca con mano» (prometto solennemente che non utilizzerò più questo orrendo modo di dire) la differenza tra un linguaggio di programmazione tradizionale, che richiederebbe centinaia di righe di codice, e il Visual Basic, con il quale si disegnano letteralmente gli oggetti e si scrivono poche righe di codice legate all'evento subito dal singolo oggetto.

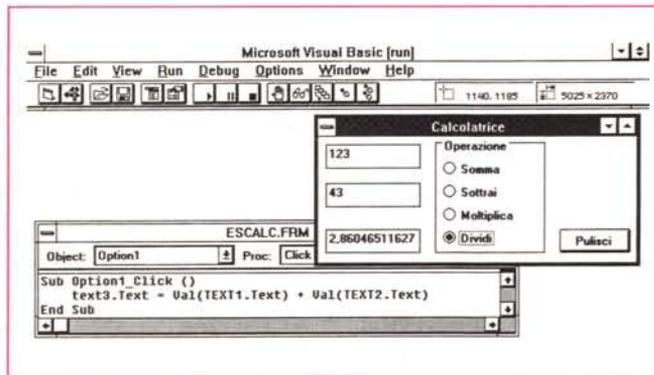
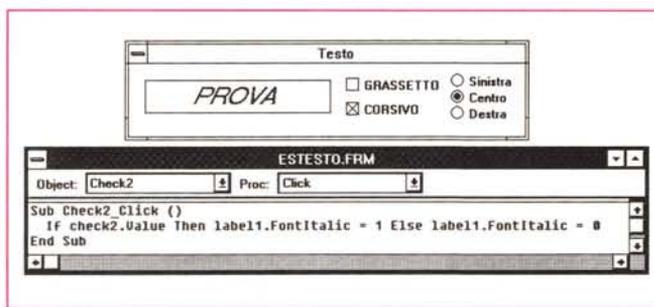


Figura 6 - MS Visual Basic 2.0 - Trattamento di una Label.

La Label, ovvero la semplice scritta sulla Form, è l'oggetto apparentemente più banale. In realtà essa stessa ha una serie di proprietà che ne permettono, anche da programma, una pesante manipolazione. In questo semplice esercizio ne gestiamo, con un gruppo di Option Box, l'allineamento, e con due Check Box, gli attributi estetici, che sono sommabili.



re il contenuto di testo alla Text Box, o variarne il Bordo o il Font, o la posizione (le proprietà sono 35, e hanno tutte già assunto un preciso valore). Possiamo inoltre chiamare l'altro oggetto, il Botto-ne, «Scrivi Ciao».

Abbiamo disegnato i due oggetti (in realtà anche la Form è un oggetto che ha, a sua volta, ben 44 proprietà). Dobbiamo decidere cosa deve fare, e quan-

do lo deve fare, il programma.

La cosa più ovvia è che scriva «Ciao» nella Text Box quando qualcuno clicca sul bottone «Scrivi Ciao».

Per ottenere questo effetto occorre individuare quale è l'evento (è il Click) e su quale oggetto tale evento si verifica (sul Command Button) e collegare a questo oggetto e a questo suo evento una istruzione. Visual Basic associa, an-

Figura 4 - MS Visual Basic 2.0 - Solo Form, Click e doppio Click. La Form è essa stessa un Oggetto pieno di... Proprietà. In questo esercizio realizziamo un programma che lavora solo sulla Form. Eseguendo un Click sulla Form stessa, si lancia una routine che scrive dei numeri. Eseguendo un doppio Click si pulisce semplicemente la Form.

che in questo caso automaticamente, ad ogni oggetto tutti i possibili eventi che può generare (nel caso del Command Button sono ben otto, oltre al semplice Click).

Stia ora a noi, individuati l'oggetto e l'evento, scrivere il comando che vogliamo sia eseguito. Per far questo basta fare click sull'oggetto, e appare una finestra in cui scrivere il programma, solo che già troviamo scritte le due istruzioni iniziali e finali, per cui si deve scrivere solo una riga. Il programma è dunque:

```
Sub Command1_Click ()
    text1.text="Ciao"
End Sub
```

Il che tradotto in italiano diventa:

Al verificarsi dell'evento Click sul bottone Command1, imposta la proprietà text della Text Box chiamata text1 pari a «Ciao».

In questo, come in tutti i casi, il programma viene eseguito solo al verificarsi dell'evento e il programma consiste nella variazione di una proprietà di un oggetto.

In tutti i programmi, anche i più complessi, quello che succede è sempre un susseguirsi di eventi e di azioni conseguenti a tali eventi.

Quindi il tutto funziona sulla base di un meccanismo semplice, imparato il quale si sa programmare in Visual Basic.

### Al lavoro

Andiamoci piano. Anche se abbiamo imparato concettualmente il meccanismo dobbiamo studiare le varie tipologie di oggetti, le varie proprietà ad essi collegate e i vari eventi possibili in ciascun oggetto.

In questa prima lezione vi proponiamo una serie di esercizi che servono ognuno per studiare un Oggetto presente nel Toolbox, non tanto per verificare il fatto che funziona proprio come un oggetto Windows, quando per cominciare a sensibilizzarci sulle varie problematiche connesse con ciascun oggetto.

Useremo il Visual Basic 2.0, uscito di recente, ma tutto quello che faremo «vale» anche per il Visual Basic 1.0.

### I primi tre oggetti Label, Text Box e Command Button (figura 2)

Quando si lancia il Visual Basic appare subito una nuova Form. Premendo F4 o cliccando (solo la versione 2.0 ha

però la Toolbar) sull'icona con una manina, posta sotto il comando RUN, definiamone la Caption, che è il titolo della Form, che indichiamo come «I primi rudimenti».

Nel maneggiare le List Box con le proprietà della Form notiamo che alcune di esse sono numeriche, ad esempio tutte quelle che servono a definire la posizione della Form sul video, alcune sono logiche (vero o falso) ad esempio quelle che servono per decidere se far apparire in alto sulla cornice della Form i Bottoni di Minimo e di Massimo.

Altre proprietà hanno scelte predefinite, come quella che serve per determinare la forma del Mouse, in questo caso a fianco della riga di editazione appare una freccia verso il basso che mostra tutte le possibilità (quelle per il Mouse sono ben 13). Altre ancora si appoggiano a ulteriori Windows, come quella per scegliere il colore di fondo o quella che serve per scegliere la Icona da associare alla Form.

Il consiglio che vi diamo è di scorrere tutte le altre possibilità anche quando dovete scegliere una proprietà a voi già nota, perché è probabile che anche le altre, prima o poi, vi possano servire.

Disegniamo poi la Label, e cambiamone il Caption, poi la Text Box, e cancelliamone la proprietà Text, e infine il Command Button e ne cambiano la Caption.

Ora prima di scrivere il programma, che è lungo 1 (dico UNA) riga, facciamo il punto della situazione. Vi confermiamo il fatto che nel disegnare gli oggetti, poi spostabili e dimensionabili come se fossero semplici oggetti grafici, sono state impostate automaticamente le varie proprietà. Queste possono essere cambiate anche a mano direttamente nella Window Properties.

La Form ha già il suo aspetto. Il programma si può già eseguire (il comando è RUN START) e non dovrebbero esserci errori. Se tutto è OK possiamo bloccare l'esecuzione (RUN END) e scrivere la riga di programma.

Per fare questo si fa doppio click sul Bottone e si passa nella Window con il programma in cui sono già impostati il nome dell'oggetto e la Procedura / Evento che ci interessa.

Le due List Box di tale nuova finestra propongono la prima tutti gli Oggetti (compreso General, che come vedremo, serve per le impostazioni generali) e la seconda tutti gli eventi relativi all'oggetto selezionato a sinistra.

La riga da scrivere, che vedete nella foto, serve per assegnare alla proprietà Text dell'oggetto Text1 il valore «È questa è una Text Box».

Figura 7 - MS Visual Basic 2.0 - Uso delle Scroll Bar.

Le Scroll Bar sono uno strumento utilissimo in quanto possono risolvere un gran numero di problemi, in particolare tutti quelli in cui ci sia qualche cosa che può variare da un minimo ad un massimo. Le proprietà MIN e MAX, come tutte le proprietà in genere, possono essere definite «al volo» dall'interno del programma. Ad esempio se la Scroll Bar viene destinata ad una funzione di scorrimento dei Record di un Archivio occorre impostare il MIN a 1, il primo record, e il MAX al valore, che è variabile, dell'ultimo RECORD.

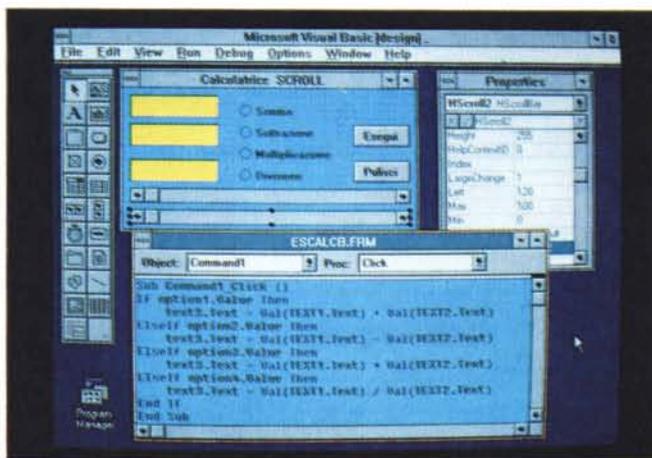
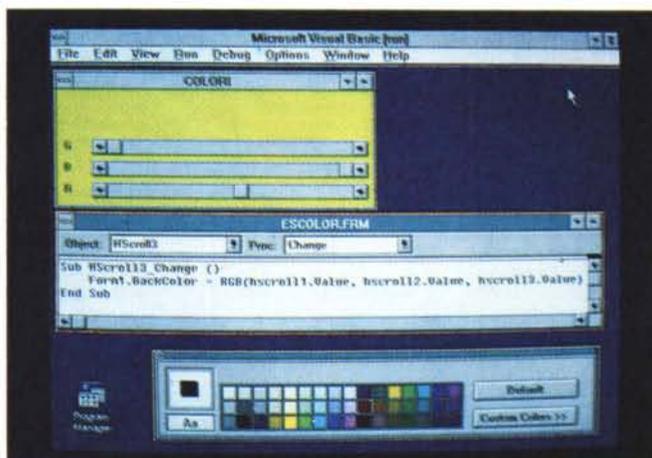


Figura 8 - MS Visual Basic 2.0 - Scroll Bar per gestire il Colore.

Una delle «properties» comuni a quasi tutti i Control è il colore che viene gestito dal Visual Basic in svariate maniere. La più semplice si basa sulla Window Color Palette, ma non si può usare all'interno del programma, in cui invece servono istruzioni per definire il colore. L'istruzione è Control.Color = RGB (r,g,b) in cui r, g e b, sono i valori delle componenti Rosso, Giallo e Blu del colore voluto. Poiché ogni colore può assumere valori numerici da 0 a 255, le possibilità sono 256 alla terza, ovvero i famosi 16 milioni di colori permessi dalle schede video a 24 bit e «spettacolarmente» riconosciuti da Windows.



Scritta l'istruzione si può eseguire il programma.

Ora dieci minuti di ricreazione che vi invitiamo a passare modificando le varie proprietà «estetiche» della Form, della Label, della Text Box e del Command Button.

### Option Button e Check Button (figura 3)

Anche questi due oggetti sono ben noti agli utilizzatori di Windows. Gli Option Button sono quelli con il cerchietto. La loro caratteristica è che sono mutuamente esclusivi. Premendone uno viene attivato mentre gli altri vengono disattivati.

La Property che gestisce la situazione ON / OFF del bottone è di tipo Logico e

si chiama Value. Se occorre da programma settare uno degli Option Button il comando è:

Option1.value=true (o false)

Se invece si deve leggere lo stato del Bottone il comando è

If Option1.value = true Then .. ovvero

If Option1.value = 1 Then .. ovvero

If Option1.value Then ..

I vari Option1.Button possono essere inseriti in un Group Box. Questa operazione è indispensabile quando in una stessa Form vengono messi più gruppi di Option Button.

Il Check Box invece vive autonomamente. Può essere anche esso posto in ON o in OFF, senza influenzare gli altri Check Box eventualmente presenti sulla stessa Form.

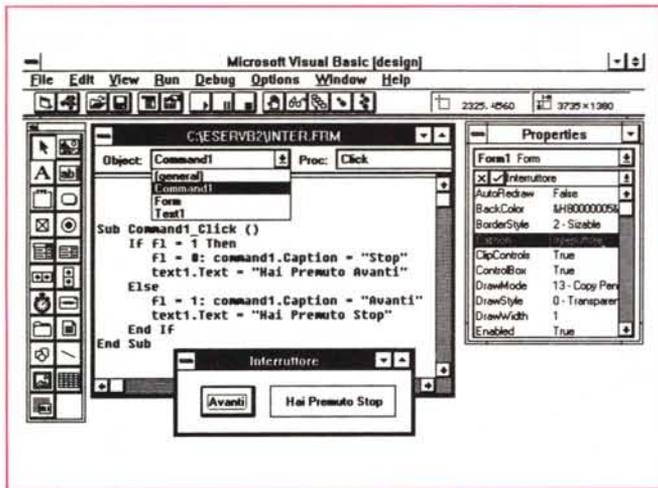
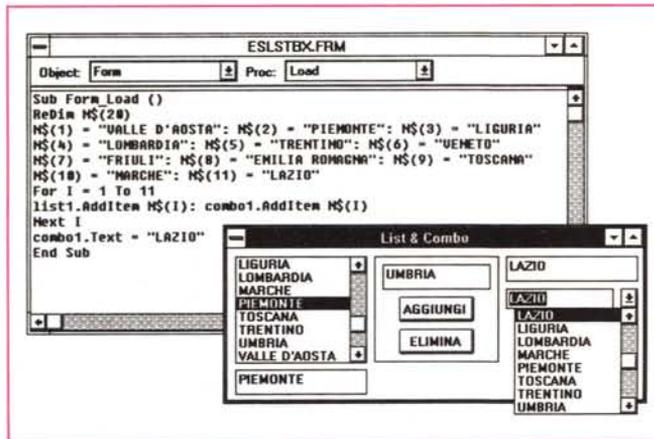


Figura 9 - MS Visual Basic 2.0 - Interruttore. In questo esercizio simuliamo un bottone-interruttore. In pratica usiamo un «Command Button» al quale cambiamo la proprietà «Caption», ovvero la scritta che appare al suo interno, quando lo clicchiamo. Per ottenere l'effetto contrario, che cioè riappaia la scritta iniziale quando si preme di nuovo il Button, utilizziamo un Flag, che è uno strumento classico di programmazione.

Figura 10 - MS Visual Basic 2.0 - List e Combo. Chi già conosce Windows già conosce, potenzialmente, gli Oggetti inseribili in un programma Visual Basic, in quanto si tratta degli stessi oggetti che troviamo nel sistema e nelle applicazioni e che già sappiamo usare. Nel fare l'esercizio, descritto nel testo, sulle List Box e le Combo Box, possiamo dare per scontato il fatto che il lettore conosca già il funzionamento di tali oggetti.



Con l'occasione possiamo provare a cambiare alcune proprietà, quelle di cui intuimmo facilmente il significato, ad esempio quelle che servono per gestire i Bottoni sulla Cornice (attenzione a non togliere il Control Button, che serve per uscire dall'applicazione, anche quando ci siamo dimenticati di inserire da qualche parte una Istruzione END), o quelle che servono per posizionare e colorare la Form.

Per scrivere i numeri usiamo l'istruzione FOR .. NEXT, che serve per «contare». In pratica per I che va da 1 a 20 (e quindi contiamo da 1 a 20) scriviamo, con l'istruzione PRINT, varie formule che coinvolgono I.

Ne risultano 20 righe piene di numeri, ben incolonnate.

Tra gli eventi che caratterizzano la Form il più importante è sicuramente il Load. In una applicazione reale sono infatti molte le Azioni che occorre compiere quando si apre una Form. Ad esempio è in tale momento che si possono assegnare ai vari Control i valori iniziali.

**Una microapplicazione: una calcolatrice bottoni (figura 5)**

Già con le conoscenze minime acquisite fino ad ora possiamo lanciarsi in una microapplicazione. Una calcolatrice con due «Text Box», in cui digitare due numeri, una «Text Box» in cui far scrivere dal programma il risultato, le quattro Option Button per scegliere una delle quattro Operazioni e un Bottone per ripulire le tre Text Box.

Il programma lo dividiamo in quattro pezzi, ciascuno legato al Click su ciascun Option Button. L'istruzione sottostante è banale:

```
text3.text = Str$(Val(text1.text) + Val(text2.text))
```

La funzione Val(.) serve per tradurre il contenuto della «Text Box», che è ovviamente un testo, in un numero sul quale eseguire dei calcoli da riconvertire in testo.

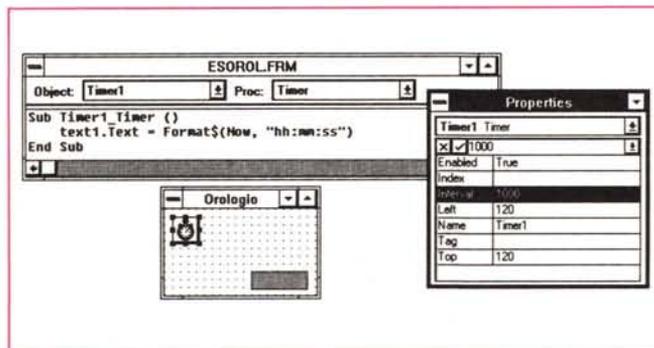
Ogni bottone avrà il suo programma. Occorre considerare il fatto che è la struttura stessa degli Option Buttons che permette di risolvere l'organizzazione della applicazione, senza che debba occuparsene il programmatore.

Il Command Button Pulisci serve per svuotare le Text Box. Anche in questo caso l'istruzione consiste nella definizione di una proprietà del Control:

```
text1.text = «»
```

Non abbiamo affrontato il problema

Figura 11 - MS Visual Basic 2.0 - Un timer. In questa applicazione sintetizziamo il funzionamento del Timer che è un Control «nascosto», nel senso che in fase di disegno appare sulla Form, ma in fase di esecuzione non ha nessun bisogno di apparire e non appare. La proprietà principale del Timer è l'Intervallo, nel quale indichiamo ogni quanto tempo scatta «l'evento». Se impostiamo l'Intervallo a 1000 indichiamo che l'«Evento Timer» accade ogni secondo. Nell'esercizio sfruttiamo questo evento per scrivere l'ora esatta sul Form. Questa applicazione può facilmente essere aggiunta in qualsiasi altra Form, per dotarla di un orologio.



**Le proprietà di una Form (figura 4)**

La Form non è un oggetto passivo, anzi. Ha, come detto, moltissime proprietà e può «subire» moltissimi eventi. Vi proponiamo un semplice esercizio in

cui una Form, senza nessun oggetto al suo interno, può subire un Click o un doppio Click.

Con un Click facciamo scrivere un po' di numeri, con un doppio Click lanciamo il comando CLS, che pulisce la Form.

dei formati numerici. Lo tratteremo in seguito.

### La formattazione di una Label (figura 6)

Sulla sinistra una Label «riempita» con una parola e incorniciata. Sulla destra tre Option Box per gestirne l'allineamento e al centro due Check Box per gestirne due attributi.

L'effetto delle due Check Box può sommarsi (possono essere in ON entrambe), mentre l'effetto delle tre Option Button non può sommarsi (l'allineamento è unico).

Nella figura vediamo il pezzo di programma (da ora in poi la chiamiamo Subroutine) in cui attiviamo o disattiviamo l'attributo Corsivo. Usiamo l'istruzione:

```
IF (check2.value) THEN (azione se si) ELSE
(azione se no)
```

Il test è SI se la Check Box ha la crocetta. L'azione è la definizione della proprietà della Label, ad esempio FontItalic, che, essendo di tipo logico, può essere 1 o 0.

### Un Control molto simpatico: la Scroll Bar (figura 7, 8)

Le Scroll Bar, verticale e orizzontale, sono uno strumento utilissimo in quanto possono essere sfruttate per risolvere un gran numero di problemi, in particolare tutti quelli in cui ci sia qualche cosa che deve variare da un minimo ad un massimo.

Le due proprietà MIN e MAX, oltre a quella che serve per definire lo STEP, possono essere, come tutte le proprietà in genere, anche definite «al volo» all'interno del programma.

Vi proponiamo due esercizi, il primo è una variante della calcolatrice. La variante consiste nel fatto che i due valori da digitare nelle due Text Box possono essere inseriti con le Scroll Bar (fatte variare solo da 0 a 100). Gli eventi da gestire sono il Change di ciascuna Scroll Bar e il Click sul Command Button «Esegui» che esegue appunto il calcolo desiderato a seconda dell'Option Button attivo (usiamo una struttura IF .. ELSEIF .. END IF per separare le quattro possibili operazioni).

Nel secondo esercizio (fig. 8) invece inseriamo tre Scroll Bar, i cui valori vanno da 0 a 255. In corrispondenza dell'evento Change di ciascuna Scroll Bar eseguiamo una istruzione che serve per cambiare il colore dello sfondo della Form. L'istruzione, che nella immagine non appare, è:

Figura 12 - MS Visual Basic 2.0 - Picture o Image.

Visual Basic tratta anche oggetti grafici. Esistono due Control, Picture e Image, che accettano immagini Bit-Mapped o Metafile, e che diventano in tal modo oggetti manipolabili a tutti gli effetti. Anche la Form può contenere una immagine che diventa lo sfondo dell'applicazione.

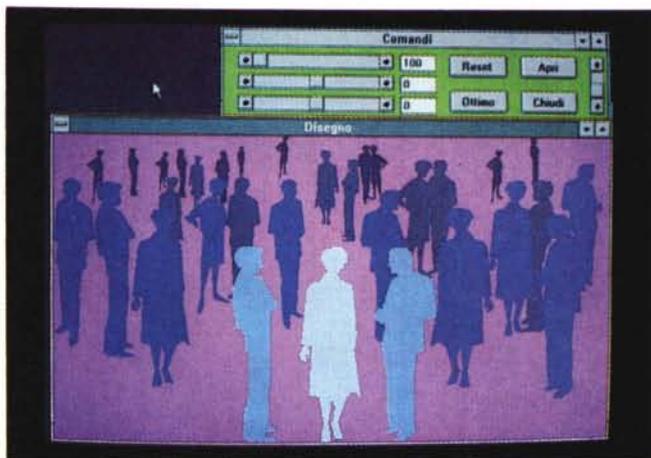
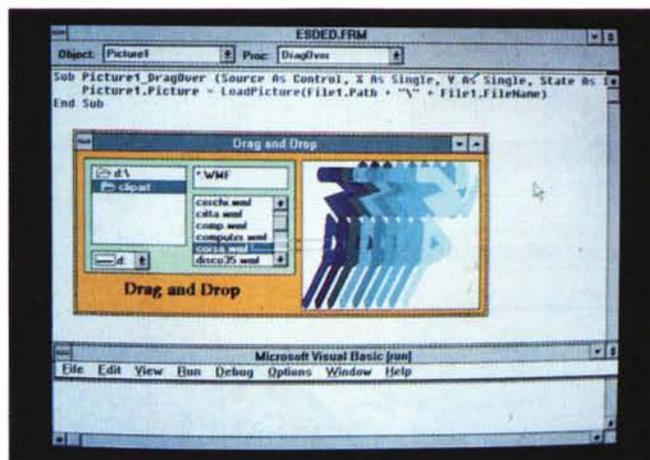


Figura 13 - MS Visual Basic 2.0 - File Manager e Drag and Drop. In questo esercizio verificiamo i tre Control che servono per realizzare un mini File Manager all'interno della propria applicazione, per consentire all'utente la scelta e la lettura di un file. Nell'esercizio attiviamo anche una funzione di Drag and Drop per portare il nome del file con il disegno all'interno della Picture.



```
Form1.backcolor=RGBcolor(r,g,b)
```

in cui a r,g e b vanno sostituiti i valori assunti da ciascuna delle tre Barre (proprietà Hscroll.Value).

Nella figura vediamo invece la subroutine legata all'evento Load della Form. Il suo compito è quello di settare ad un determinato valore iniziale le tre Barre.

### Interruttore (figura 9)

Questo esercizio, in cui simuliamo un interruttore, è dedicato al «Command Button» al quale cambiamo «al volo» la proprietà «Caption», che sarebbe la scritta che appare al suo interno.

L'effetto desiderato è che facendo Click sul bottone la sua scritta varia da Avanti a Stop e viceversa.

Per risolvere questo problema comune (che può servire ad attribuire ad un solo Bottone due funzioni, in alternativa

tra di loro), usiamo un Flag. Un Flag è una variabile che viene posta uguale a 1 o a 0, e che viene «girata» da 0 a 1 o viceversa quando si fa click sul bottone, usando una struttura di programmazione IF .. ELSE .. END IF.

### Due control utilissimi: la List Box e la Combo Box (figura 10)

Diamo per scontato il fatto che il lettore conosca già il funzionamento di tali oggetti.

Il problema è sapere come sono gestiti dal Visual Basic, e quindi come si alimentano, come si individua l'elemento selezionato, come si elimina un elemento, ecc.

Nella figura vediamo l'istruzione List1.AddItem = X, che è quella che serve per alimentare di elementi la Lista. Nel nostro caso la alimentiamo al verificarsi del caricamento della Form.

Gli altri eventi gestiti dai vari oggetti

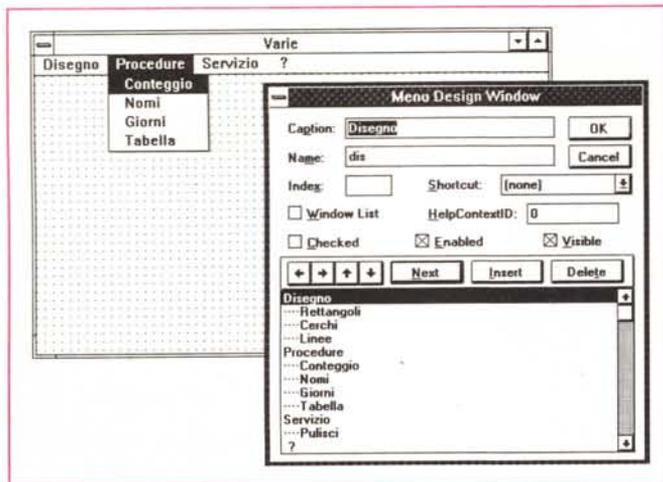
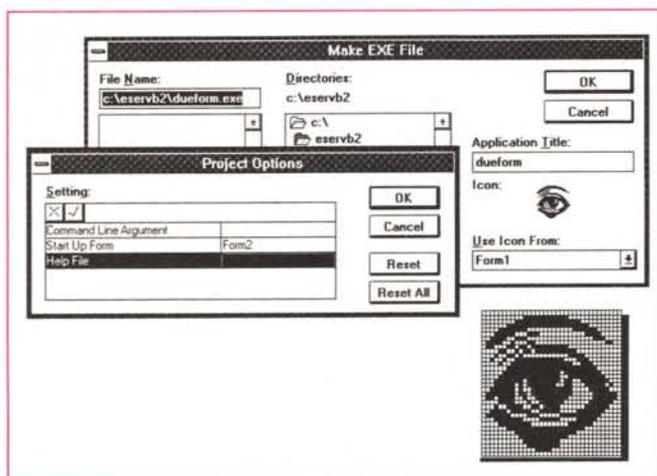


Figura 14 - MS Visual Basic 2.0 - Menu. Anche la costruzione e il successivo uso di un Menu di Windows è semplificata al massimo. Basta «compilare» una Dialog Box in cui si indicano Nome del Comando (Caption, quello che appare nel Menu), Name, nome «lato programma» e, poi con semplici comandi di Outlining, l'organizzazione delle varie Voci.

Figura 15 - MS Visual Basic 2.0 - La compilazione.

Una volta messo a punto il programma lo si può compilare. In tal modo risulta trasportabile su qualsiasi altra macchina ed eseguibile al pari di qualsiasi altro programma Windows. Oltre al file EXE, al quale si può associare, ovviamente, una Icona, occorre però portarsi appresso la libreria VBRUN x00.DLL da piazzare all'interno della directory System di Windows.



**Apri** serve per aprire la seconda Form (comando Form.Show)

**Chiudi** per chiudere la seconda Form (comando Form.Hide)

**Reset** per impostare la dimensione iniziale della Immagine

**Ottimo** per impostare la dimensione dell'immagine a quella che meglio riempie la Form.

Con le tre Scroll Bar simuliamo uno Zoom sull'Immagine e un suo spostamento, all'interno della Form alla ricerca del particolare che si vuol vedere meglio.

Le proprietà interessate sono quelle di tipo numerico come WIDTH (larghezza), HEIGHT (altezza), LEFT (spostamento da sinistra), TOP (spostamento dall'alto).

La piccola Scroll verticale viene fatta variare da 0 a 15 e viene usata per impostare il colore di sfondo, settandolo con l'istruzione QBcolor(x).

### Conclusioni

Per assoluta mancanza di spazio non vi commentiamo gli ultimi due esercizi. Il primo serve fondamentalmente a due cose. La prima è la gestione di un File Manager, fatto con i Control che permettono di impostare o leggere Drive, Directories, File. La seconda è l'esecuzione di un Drag and Drop, tipo di evento gestibile con il Visual Basic, anche se è un po' più complesso degli altri in quanto coinvolge due Control.

Usiamo Drag and Drop per visualizzare nella Picture Box la figure scelta dalla Lista dei File (fig. 13).

In una applicazione Windows è sempre presente un Menu. Anche Visual Basic permette di generare menu e lo fa, come al solito, attraverso uno strumento specifico assolutamente semplice da usare (fig. 14).

Basta scrivere il nome delle varie Voci, e, rientrati di un livello, i nomi delle Voci di secondo livello (sono possibili anche ulteriori livelli) e il menu è fatto. In realtà oltre alla proprietà Caption, che è la scritta che appare sul menu, occorre impostare anche il Name, che è il nome da usare nel programma.

Facendo Click sulla singola voce si apre la finestra dell'Evento Click sulla Voce che è del tutto analogo a qualsiasi altro Click su qualsiasi altro oggetto.

Anche in questo caso il meccanismo è quindi lo stesso a dimostrazione della nostra tesi iniziale secondo la quale per lavorare con Visual Basic occorre solo imparare la regola generale, che trovia applicata in tutti i Control, gli Eventi e le Istruzioni di programmazione.

presenti nella Form sono:

List1.Removeitem = X che vuole un valore numerico,

List1.Value che restituisce il numero progressivo dell'elemento evidenziato,

List1.Text che restituisce il valore dell'elemento evidenziato

List1.Countitem che restituisce il numero totale degli elementi della lista.

### Il Timer (figura 11)

Dopo un esercizio un po' complesso ve ne proponiamo uno molto più semplice.

Il Timer è un Control nascosto, nel senso che in fase di disegno appare sulla Form, ma in fase di esecuzione non si vede.

La proprietà principale del Timer è l'Intervallo, nel quale indichiamo ogni quanto tempo scatta l'«evento». Se impostiamo l'Intervallo a 1000 indichiamo che l'«Evento Timer» accade ogni se-

condo. Nell'esercizio sfruttiamo questo evento per scrivere l'ora esatta in una Label posta in un angolo della Form. Questa applicazione può essere aggiunta in qualsiasi altra Form, per dotarla di un orologio.

### Picture o Image (figura 12)

Visual Basic tratta anche oggetti grafici. Esistono due control, Picture e Image, che accettano immagini Bit-Mapped o Metafile, e che diventano in tal modo oggetti manipolabili a tutti gli effetti. Anche la Form può contenere una immagine che diventa lo sfondo dell'applicazione.

Nell'applicazione di cui vi mostriamo solo l'effetto finale usiamo più di una Form (cosa che accade sempre nelle applicazioni più complesse). Nella prima inseriamo una serie di Control che servono per gestire le caratteristiche geometriche della figura posta nella seconda Form:



# WINDOWS WORLD '93

M O S T R A  
C O N V E G N O  
P R O F E S S I O N A L E  
P E R G L I U T E N T I  
D I W I N D O W S

**INTERFACCE GRAFICHE E SOLUZIONI APPLICATIVE**

**Milano, 31 marzo - 3 aprile 1993**

*1<sup>a</sup> EDIZIONE*



**SPAZIO MILANONORD**

**via Pompeo Mariani 2 - Milano**

orario:

9.30 - 18.00 / sabato 3 aprile: 9.30 - 14.00



*è un'iniziativa*



**ASSOEXPO**

e

**WORLD EXPO**  
CORPORATION

*promossa da*

**IDG COMMUNICATIONS ITALIA**

**comitato tecnico:**

AST RESEARCH Italia

BORLAND Italia

DELL Computer

LOTUS Italia

MICROSOFT

TEXAS INSTRUMENTS Italia

WORD PERFECT Italia

**segreteria generale:**

WINDOWS WORLD

via Domenichino 11

20149 Milano

c.p. 15117 - 20150 Milano

tel. 02. 4815541

fax 02.4980330

telex 313627