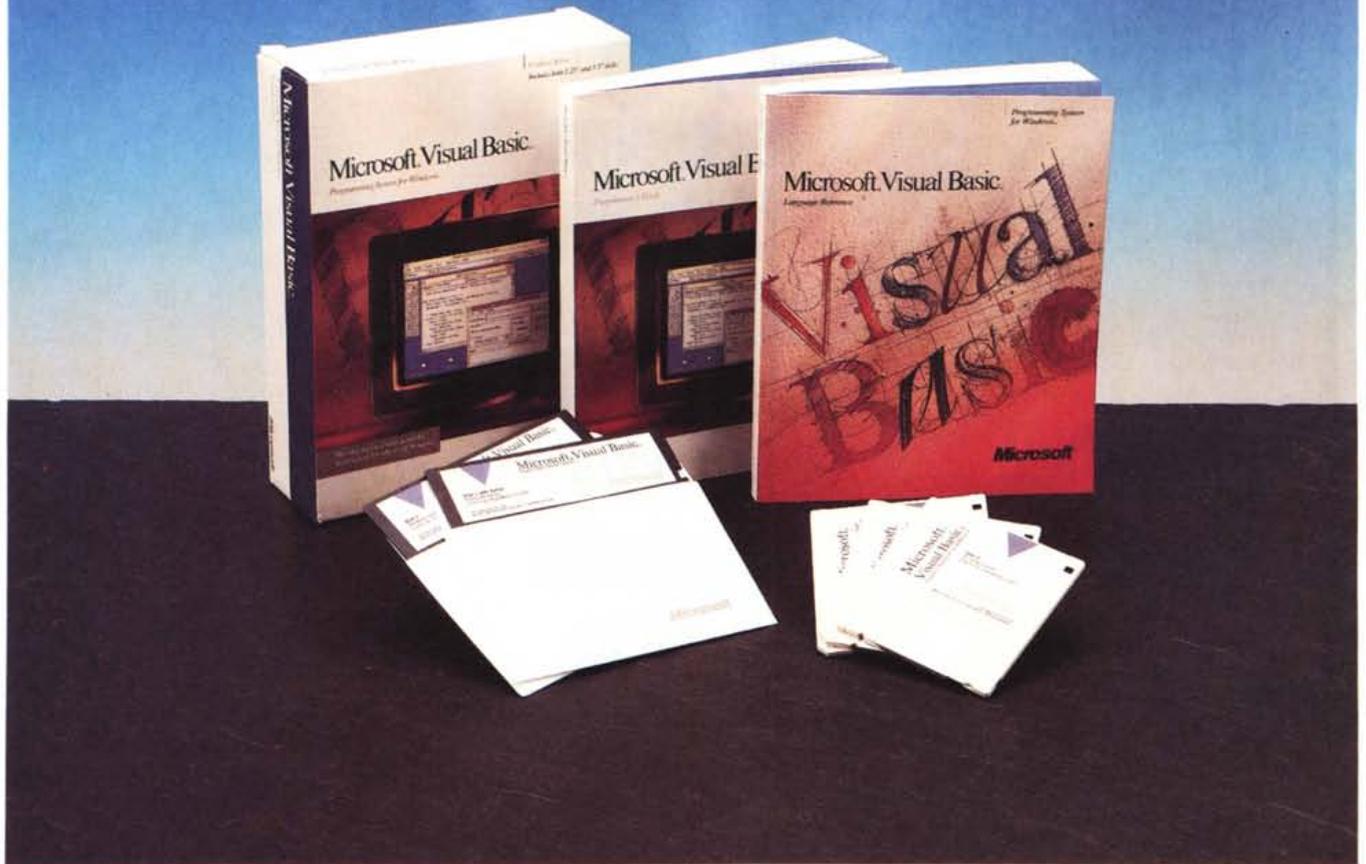


PROVA



# Microsoft Visual Basic per Windows

di Francesco Petroni

**A** chi di voi ha vissuto la nascita del Microcomputer (e perché no anche quella di MCmicrocomputer) e la sua successiva straordinaria evoluzione, la parola BASIC ricorderà molte cose.

La prime macchine, quelle in cui RAM e ROM si misuravano ancora in termini di multipli di 4 kbyte, accettavano su di sé solo un linguaggio elementare, residente, che svolgeva anche funzioni di sistema operativo. Si trattava del BASIC, parolina che oltre all'evidente significato di «basilare», «fondamentale» nasconde una sigla. BASIC significa in realtà Beginner All-purpose Symbolic Instru-

ction Code, è un linguaggio nato per usi didattici.

La sua iniziale dipendenza dalle varie macchine ha fatto necessariamente proliferare le sue versioni, che superano ora ampiamente il centinaio.

Anche le prime macchine IBM, parliamo dei primi 8088 di dieci anni fa, disponevano di un Basic interprete in ROM. A molti di voi sarà capitato qualche volta, se per qualche motivo alla accensione non era presente il DOS, di andare... a finire nel Basic.

Il fatto che il Basic dell'IBM fosse in ROM costituì anche uno dei primi ostacoli alla dif-

fusione delle standard IBM in quanto le case concorrenti non potevano copiare le ROM.

Per queste fu quindi sviluppato dalla Microsoft il GWBASIC, che era un Basic totalmente residente su disco e quindi utilizzabile su qualsiasi macchina non-IBM.

Con l'evoluzione dei PC e con la diffusione dello standard MS-DOS, il Basic è tramontato, ma non del tutto, in quanto vari produttori, primi tra tutti Microsoft (il cui presidente, il mitico Bill Gates, ha cominciato proprio scrivendo linguaggi Basic) lo hanno tramutato in un prodotto, con tanto di versioni successive e con tanto di implementazioni lega-

te alla contemporanea evoluzione delle macchine.

Oggi esistono e sono molto diffusi il Quick Basic della Microsoft e il Turbo Basic della Borland, che riconoscono mouse, lavorano in tutte le modalità video possibili, ecc. e sono dotati di ambienti Editor evoluti, di Help interattivi e sensitivi, di funzioni di Debugging sofisticate.

In particolare molte delle funzioni inserite nelle ultime versioni del Quick Basic (l'ultima è la 4.5) sono state inserite, con i dovuti adattamenti, nel Visual Basic.

Negli anni '90 il Basic vivrà sicuramente una nuova stagione d'oro, anche se ben poco rimane del vecchio Basic in ROM, quello di dieci anni fa.

I fatti che ci inducono ad azzardare questa previsione sono due, innanzitutto la presenza del Quick Basic, versione 4.5, ma privo del compilatore, nel nuovo DOS 5.0. E questo significa che i previsti 60 milioni di utilizzatori del DOS 5.0 disporranno gratuitamente anche di un Basic molto evoluto (fig.1).

Il secondo fatto è la nascita del Visual Basic, prodotto nuovissimo che, come diremo tra poco, si pone l'obiettivo di diventare per il mondo Windows, ormai indiscutibilmente l'ambiente operativo standard dei prossimi anni, e per l' OS/2 Presentation Manager, quello che il Basic è stato per i primi PC.

### Windows non si discute più

Con una frequenza ormai quasi quotidiana riceviamo nuovi prodotti per Windows. Tale tendenza continuerà per un bel po' di anni. Addirittura il nuovissimo DOS 5.0 viene pubblicizzato come il miglior partner per Windows.

Uno degli aspetti più interessanti di queste novità è che le varie case, ad esempio la Lotus, con i suoi 123 e Ami Pro 2 per Windows, danno comunque interpretazioni «personali» della interfaccia Windows. Lotus infatti risponde allo strumento ToolBar (le barre con i bottoni di MS Excel 3.0 e MS Word) con la Icon Panel, disponibile sia in 123/W che in Ami Pro 2.0, in cui l'utente sistema i bottoni-funzione che più gli interessano, scegliendoli tra una dotazione molto ricca o confezionandoli.

Citiamo un altro prodotto di grande interesse «storico» del quale abbiamo avuto solo notizia. Si tratta del PFS: Windows Works, primo esempio di prodotto per Windows appartenente alla tipologia Works. I Works sono prodotti multifunzionali (Word Processor, Spreadsheet, DataBase, Communications, Business Graphics, ecc.) di tipo Entry Level, non eccessivamente evoluti ma molto economici, destinati quindi ai principianti.

Invece tra i prodotti «novità» di maggior interesse, di cui abbiamo già parlato su MC, citiamo Borland Object Vision e Asymetrix ToolBook, prodotti innovativi, di difficile interpretazione e collocazione, in un mercato abituato ad una classificazione abbastanza rigida e consolidata.

In questo panorama così movimentato non potevano più mancare i linguaggi per Win-

### Microsoft Visual Basic

**Produttore e Distributore:**  
Microsoft S.p.A.  
Via Cassanese, 224  
20090 Milano Oltre - Segrate.  
Tel.: 02/269121.  
**Prezzo (IVA esclusa):**  
Microsoft Visual Basic L. 320.000.

dows, categoria di prodotti importantissima, ma fino a poco fa del tutto trascurata dai produttori.

Chi voleva programmare sotto Windows o doveva utilizzare il C più SDK (il Software Development Kit della Microsoft che permette di sfruttare le librerie DLL, comunque presenti in una macchina in cui sia stato installato Windows), oppure doveva utilizzare uno dei due linguaggi della Borland, il Turbo Pa-

scal o il C++, ambedue recentemente dotati di librerie Windows, che evitano all'utente di dover affrontare l'ostico SDK. In ogni caso queste sono soluzioni molto tecniche destinate solo ai professionisti della programmazione.

Per gli utilizzatori meno esperti non esistevano (prima del Visual Basic) linguaggi specifici. Costoro potevano ricorrere solo alla sezione Macro dei prodotti orizzontali più evoluti, con i quali però si riesce solo a programmare nel rispetto della filosofia (e quindi nel rispetto dei limiti) del prodotto da cui si parte.

Il Visual Basic si colloca in mezzo a tutto questo (fig. 2), non ha ambizioni di sostituire i linguaggi più evoluti, con i quali per intenderci vengono realizzate le applicazioni più complesse, ma non ha neanche i limiti delle sezioni Macro dei vari pacchetti.

### La ricetta Visual Basic

Gli ingredienti della ricetta che sta alla base di tale prodotto sono fondamentalmente

Figura 1 - MS QUICK BASIC - Due Basic per gli anni '90.

Microsoft, nel suo nuovo DOS, il 5.0, ha inserito come dotazione standard una serie di nuove utility. Una delle più significative è rappresentata dal Basic, anzi dal Quick Basic, 4.5 nella sola versione interprete, che, come diremo nell'articolo, ha parecchie affinità con il nuovo arrivato Visual Basic.

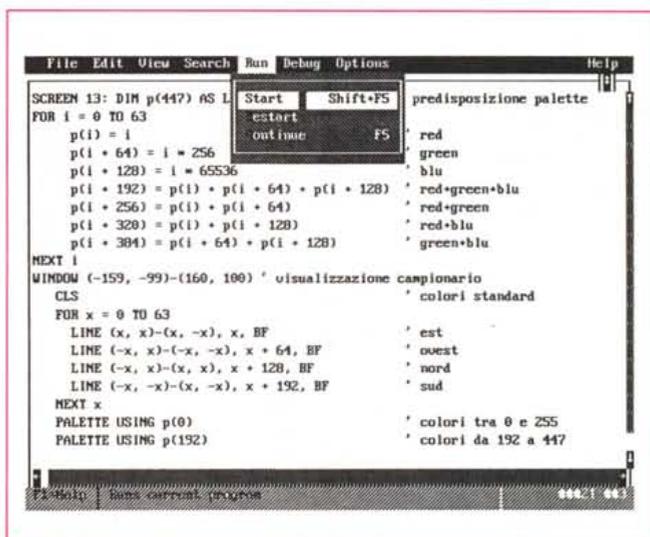
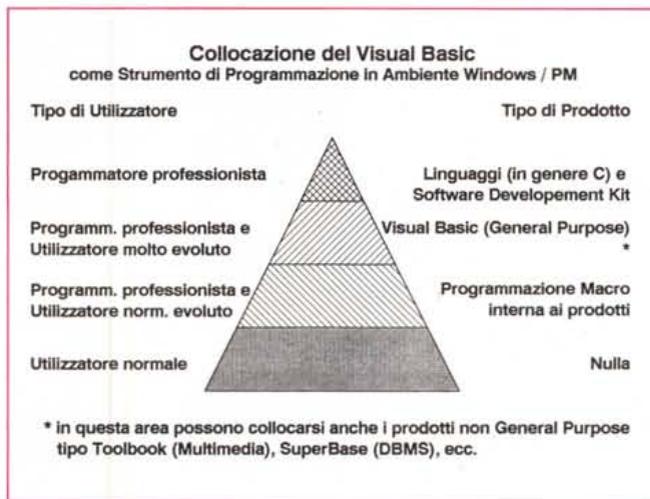


Figura 2 - MS Visual Basic - La sua collocazione. L'utilizzatore evoluto di un qualsiasi pacchetto sotto Windows può migliorare la sua produttività sfruttando le Macro che il prodotto stesso gli permette di costruire. Il programmatore professionista, che sviluppa un applicativo verticale sotto Windows, utilizza il linguaggio C e il Software Developer Kit per Windows. Lo spazio tra questi due estremi (costituito dalle applicazioni mediamente complesse affrontabili solo da utilizzatori mediamente evoluti) potrà essere colmato dal Microsoft Visual Basic.



due: 50% di Quick Basic e 50% di Windows.

Il Basic, il linguaggio dal glorioso passato, che, nonostante l'età e le diverse centinaia di versioni, è ancora molto adatto per risolvere, attraverso una programmazione abbastanza libera, problemi un po' di tutti i tipi.

Si possono sicuramente escludere le applicazioni di tipo gestionale, in cui occorre maneggiare pesantemente archivi (attività per la quale il Basic non è portato) o le applicazioni molto orizzontali, tipo Word Processor o tipo Spreadsheet, delle quali peraltro esistono già numerosi ed ottimi «esemplari».

Il secondo ingrediente è Windows con la sua interfaccia grafica, con la sua operatività orientata all'oggetto e con i suoi «oggetti» standard, ormai noti a tutti, il cui utilizzo fa parte della «cultura generale», già acquisita da qualsiasi utilizzatore di un qualsiasi prodotto Windows.

Dall'unione dei due componenti nasce un prodotto totalmente grafico, orientato agli oggetti, in cui si opera su varie tipologie di finestre, che contengono elementi Windows (sono le Form) o pezzi di codice (fig. 3).

**Il materiale e la sua installazione**

Il materiale, che nel caso specifico ci è stato fornito dalla società Quotha 32 di Firenze, consiste di due grossi manuali e tre piccoli dischetti.

La loro installazione si esegue da Windows ed è del tutto guidata. Viene richiesto attraverso una Dialog Box cosa installare e viene al momento eseguito un bilancio tra il fabbisogno di memoria e la quantità disponibile.

Alla fine dell'installazione rimane una directory, il cui nome di default è VB, il cui contenuto è di circa 2.500 kbyte, e alcune sottodirectory, che contengono il materiale di supporto, costituito da alcuni Examples, dal ricco ed efficace Tutorial e da una abbondante dotazione di Icone.

Nella directory Windows, da cui si lancia l'installazione, rimane anche il file VBRUN100.DLL, che serve per rendere eseguibili i file EXE generati con il Visual Basic.

Una volta realizzato e testato un programma Visual Basic lo si può infatti compilare, attraverso l'opzione MAKE EXE, presente nel menu FILE. Per eseguire lo stesso file su un'altra macchina occorre copiarvi anche il VBRUN100.DLL.

Help e Tutorial sono molto efficaci e particolarmente utili nelle fasi iniziali di apprendimento. L'Help, la cui gestione è delegata a Windows, è sensitivo (fig. 4), e può essere percorso utilizzando l'indice. Le istruzioni più importanti sono dotate di un Esempio, copiabile e direttamente rieseguibile.

Particolarmente ben fatto è il Tutorial, la cui durata viene stimata in 130 minuti, che spiega i vari argomenti con il massimo (per un Tutorial) dettaglio. Si addentra anche nella spiegazione delle istruzioni più importanti e consente l'accesso agli esercizi in dotazione (figg. 5, 6).

Passiamo ai manuali. Il primo è il Program-

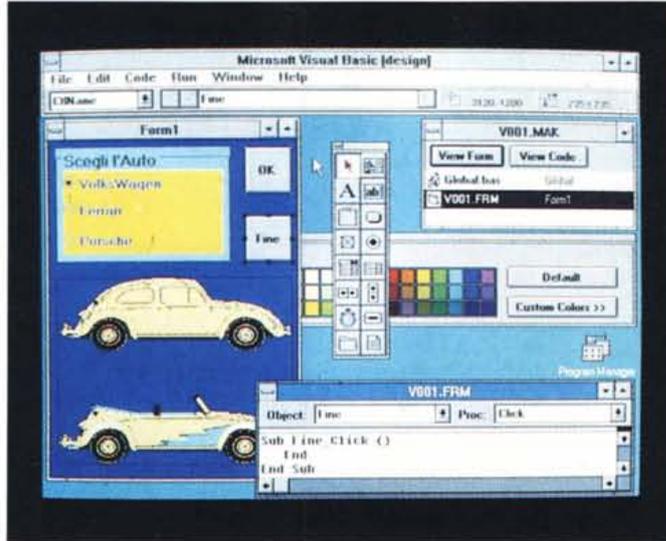


Figura 4 - MS Visual Basic - Help .. da saccheggare. Anche l'Help è un comodo strumento di apprendimento oltre che di consultazione. Può essere percorso in svariati modi, ma il suo più grande merito è quello di disporre di una serie di Examples su cui agisce anche una funzionalità di Cut che permette di eseguire il successivo Paste direttamente nell'Editor dove si sta scrivendo il codice del programma.

Figura 3 - MS Visual Basic - L'ambiente operativo. Il Visual Basic attiva una serie di finestre operative su cui si agisce in contemporanea. Accanto alla Form su cui si sta lavorando sono disponibili sia le finestre con i vari Tools che la finestra Project, che contiene l'elenco delle varie componenti, sia le finestre con i Codici dei vari programmi sottostanti ciascun oggetto della Form. Il tutto viene gestito da una finestra molto bassa, in cui è stato piazzato il menu.



Figura 5 - MS Visual Basic - Il Tutorial. Gli strumenti per l'apprendimento del Visual Basic messi a disposizione direttamente nel package sono due. Il Tutorial che, attraverso un indice ramificato, permette di analizzare abbastanza approfonditamente le caratteristiche e le modalità di utilizzazione del prodotto, e gli Esercizi che permettono all'utilizzatore di costruire passo passo una serie di applicazioni predefinite, che toccano semplici problematiche d'uso.

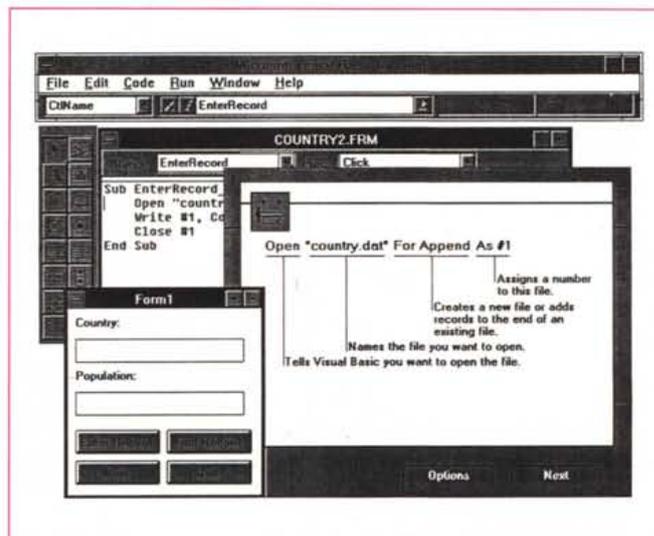
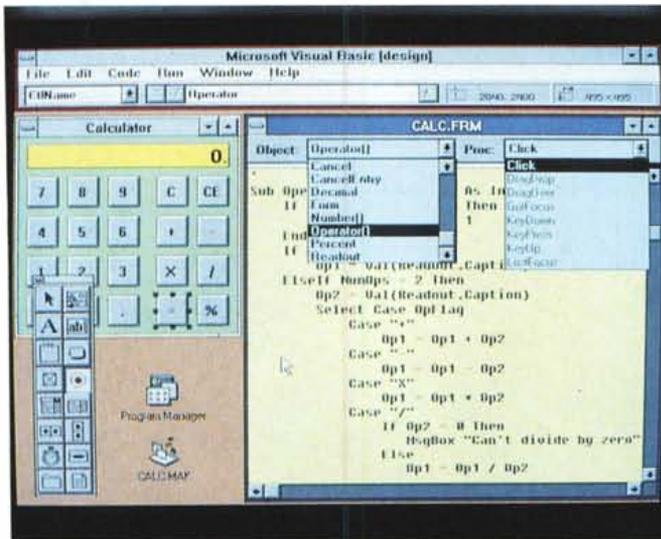


Figura 6 - MS Visual Basic - L'esecuzione di un Esercizio Tutorial. La costruzione di una calcolatrice è un esercizio ormai diventato classico per un prodotto orientato agli oggetti, in quanto ogni suo bottone diventa un oggetto. La programmazione è Event Driven in quanto vanno scritti i programmi conseguenti gli eventi che interessano ciascun bottone.



mer's Guide che tratta argomenti di complessità crescente:

- Getting Started, 4 capitoli, 32 pagine
- Visual Basic Essentials, 4 capitoli, 45 pagine
- Application's Building Techniques, 8 capitoli, 133 pagine
- Advanced Applications, 7 capitoli, 249 pagine e 3 appendici.

Il secondo Manuale è il Reference, che contiene l'elenco alfabetico completo dei comandi (dalla funzione ABS alla funzione YEAR). La trattazione è schematica, in quanto per ogni comando viene indicata Action, Syntax, Remarks, See-Also e, in molti casi, viene proposto un Example, ed è in complesso ricca (334 pagine).

### Caratteristiche fondamentali del Visual Basic

Un'applicazione sviluppata con il Visual Basic si chiama Project.

Un Project è un insieme di Forms e di Procedure, ogni Form può contenere una serie di Controls. La Form coincide con una Windows e costituisce ovviamente l'interfaccia verso l'utilizzatore.

Al suo interno lo sviluppatore inserisce i Controls, oggetti Windows, definiti e posizionati nel Form utilizzando una modalità operativa totalmente Grafica, analoga a quella di un prodotto di tipo Drawing, e per ciascuno dei quali lo sviluppatore deve definire una serie di «Properties» (dipendenti dal tipo di Control) e per ciascuna di queste le relative specifiche (dipendenti dal tipo di Property).

Materialmente l'Object va scelto dal ToolBox (fig. 7) e poi va posizionato nella Form, sulla quale è possibile attivare e visualizzare una griglia. Poi occorre definire le Property che interessano e per ciascuna di queste le sue specifiche, operazione che si esegue utilizzando la Property Bar (fig. 8), che è posta immediatamente al di sotto del menu principale.

La programmazione è Event Driven e quindi

di i «pezzi» di programma vanno collegati al singolo Control e al verificarsi di uno specifico Event su di esso.

La programmazione oltre che al singolo Control può essere legata alla Form, al verificarsi di un evento su di essa (ad esempio alla sua apertura), oppure legata a tipologie di eventi che non coinvolgono Control (ad esempio la pressione di un Tasto), oppure può avere effetto Globale su tutta l'applicazione (in questo caso è slegata dall'Evento).

Cosa ci sia in un Project viene evidenziato dalla Project Window, in cui sono elencate le

varie Form presenti nell'applicazione (che possono mostrare sia la «faccia» Form, che la faccia «Code») e i vari Moduli di solo codice.

La Form, in quanto Window a tutti gli effetti, dispone di specifiche funzioni che ne definiscono le caratteristiche dimensionali e funzionali. Inoltre esistono comandi per l'attivazione e la disattivazioni di Form, necessarie in caso di applicazioni MultiForm.

A ciascuna Form si può attribuire un suo menu (come lo vedremo tra poco).

Una Form standard contiene sempre la Control Box in alto a sinistra, che presenta l'opzione Chiudi (che può servire in ogni caso per terminare l'esecuzione dell'applicazione). Presenta sempre le due/tre icone per l'ingrandimento, il Ridimensionamento e la Riduzione ad Icona.

Il Menu può essere confezionato con l'utilità Menu Design Window, che, attraverso un semplicissimo Editor basato su un Outliner a più livelli (ma in genere ne bastano due), permette di definire le voci principali della barra e le opzioni della singola tendina (fig. 9). Ne parliamo tra un po'.

### Il Menu Design Window

Il ToolBox serve per confezionare le Form, che rappresentano l'interfaccia dell'applicazione verso l'utente. Il secondo potente strumento a disposizione dello sviluppatore è il Menu Design Window, che serve per costruire il menu della singola Form.

In questo caso lo strumento ha fini esclusivamente organizzativi in quanto, come noto

**Microsoft Visual Basic**  
Descrizione degli strumenti del Toolbox

Consente di selezionare uno o più oggetti (va premuto anche il tasto Ctrl) presenti nel Form	Pointer		Picture Box	Riquadro che può contenere vari tipi di Immagine. Fissa o variabile (si deve indicare il nome del file)
Etichetta, scritta fissa inserita nel Form. Si possono definire le sue caratteristiche estetiche	Label		Text Box	Riquadro il cui contenuto va definito via programma. E' assimilabile al Campo di un Record
cornice per contenere due o più Option Button, che diventano mutuamente esclusive	Frame		Command Button	Il Bottone è l'oggetto più classico. Il Click è l'evento più classico. E' analogo al Button di Excel 3
Si tratta di un bottone di tipo Logico. Se il suo valore è Vero il quadratino presenta una crocetta	Check Box		Option Button	Serve per scegliere tra varie opzioni che si escludono vicendevolmente
Combinazione tra un Text Box e una List Box, che suggerisce una serie di voci prestabilite	Combo Box		List Box	Elenco di scelte prefissate. Nel caso che non entrino nello spazio predisposto appare la Scroll Bar
Classica Barra di Scorrimiento di cui si possono definire Valori Minimo, Massimo e di incremento	Horizontal Scroll Bar		Vertical Scroll Bar	Classica Barra di Scorrimiento di cui si possono definire Valori Minimo, Massimo e di incremento
Orologio programmabile. Il raggruppamento dell'orario impostato è un Event	Timer		Drive List Box	Elenco dei Drive disponibili nella macchina
Elenco delle Directories presenti nel Drive scelto tra quelli indicati dal Drive List Box	Directory List Box		File List Box	Elenco dei Files presenti nella Directory e nel Drive scelti tra quelli indicati dagli altri Controls

Figura 7 - MS Visual Basic - Il Toolbox. Svilupperemo la prova del Visual Basic praticamente in due puntate. La prima, la prova classica, è questa che non può spingersi sugli aspetti pratici. La seconda, che pubblichiamo in questo stesso numero, qualche pagina più in là, nella rubrica Windows, presenta una serie di «esercizi» pratici, che permettono tra l'altro di approfondire l'argomento Controls.

a tutti, una applicazione può distribuire le sue varie funzionalità in un albero dei menu.

I più importanti applicativi per Windows, citiamo i classici Excel e Word per Windows, dispongono tutti di un proprio menu, e per ogni voce del menu principale attivano una sottostante tendina.

Opzionando la singola voce può essere eseguito il comando relativo o, se si tratta di un comando complesso, viene mandata in esecuzione una Dialog Box in cui l'operatore deve inserire contestualmente tutte le sue ulteriori scelte.

Il generatore di Menu del Visual Basic dispone di uno specifico Editor, assolutamente intuitivo, che permette di aggiungere, inserire e cancellare voci.

Il livello cui appartiene ogni singola Voce è definibile attraverso un Outliner, che fa rientrare le voci delle tendine rispetto alle voci principali (da non rientrare). Si possono eseguire più rientri ed in tal caso le tendine secondarie si affiancano a quelle in cui sta la voce che le richiamano.

Ad ogni singola voce va attribuito un Caption, ovvero il testo che appare nel menu, e un CtlName, il nome da usare nel programma. Il click sulla voce di menu è, manco a dirlo, un «Event» al verificarsi del quale viene mandata in esecuzione una specifica subroutine.

Esistono inoltre ulteriori comandi, utilizzabili però solo in programmazione, che permettono sia di gestire dinamicamente l'albero dei menu, sia di attivare o disattivare voci (con il solito sistema di farle diventare grigie), sia di mostrare un simboletto di «visto» in caso di voci attivabili in alternativa l'una rispetto all'altra.

Insomma con la Design Menu Window si possono gestire i menu nella stessa modalità evoluta che l'utilizzatore di Windows è abituato a trovare realizzata nei vari applicativi che conosce.

Cogliamo infine l'occasione per ricordare che l'organizzazione della applicazione è a totale carico dello sviluppatore, che potrà ad esempio decidere di sfruttare una complessa ramificazione del menu oppure di alleggerirla e di ricorrere a Dialog Box ricche di Command Button (fig. 10).

### Le funzioni grafiche

Oltre alle normali funzioni Windows, che sono ovviamente tutte orientate alla grafica, esistono anche delle funzioni specificamente Grafiche, che possono intervenire sia a livello di Form, che in tal caso diventa un foglio di disegno, sia a livello di Control di tipo Picture (fig. 11). Cominciamo a parlare di quest'ultima.

La Control Picture può contenere una immagine già disponibile su File, oppure può contenere un disegno fatto, all'interno della Control, con una serie di istruzioni grafiche, riprese in buona parte dal Quick Basic.

Nel primo caso l'immagine può essere sia di tipo BitMapped, formato e desinenza BMP, realizzabile quindi con il normale PaintBrush disponibile con Windows, che come

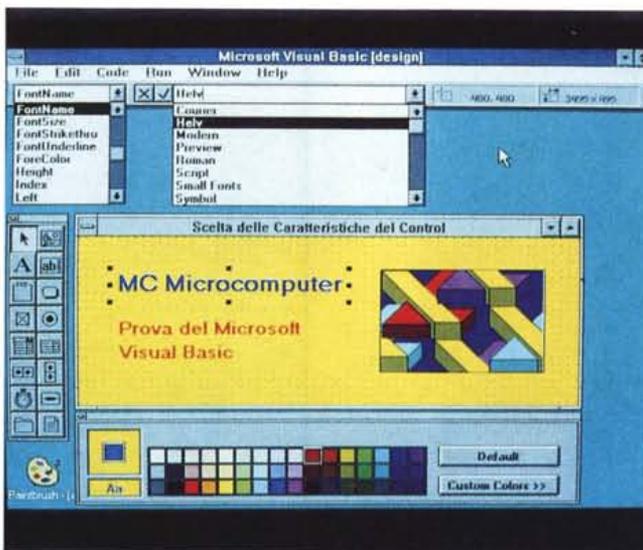


Figura 9 - MS Visual Basic - Il Generatore di Menu Formidabile. Semplicemente formidabile.

Il primo livello di organizzazione di una qualsiasi applicazione risiede nel menu, che con il Visual Basic si costruisce utilizzando una finestra di dialogo in cui si scrivono una ad una le varie voci. Poi pigliando bottoni freccia si definiscono i livelli delle singole opzioni o si modifica il loro ordine. Alla fine non occorre fare altro che «appiccicare» il singolo programma sotto la singola voce.

Figura 8 - MS Visual Basic - La Property Bar.

Per ciascun Control, così in Visual Basic si chiamano gli oggetti presenti in ciascuna Form, è possibile definire una serie di caratteristiche, e per ciascuna di queste è possibile definire il relativo parametro, utilizzando le due Combo Box, presenti sotto il menu della finestra principale. Accanto alle due Combo Box appaiono le coordinate e le dimensioni dell'oggetto selezionato nella Form.

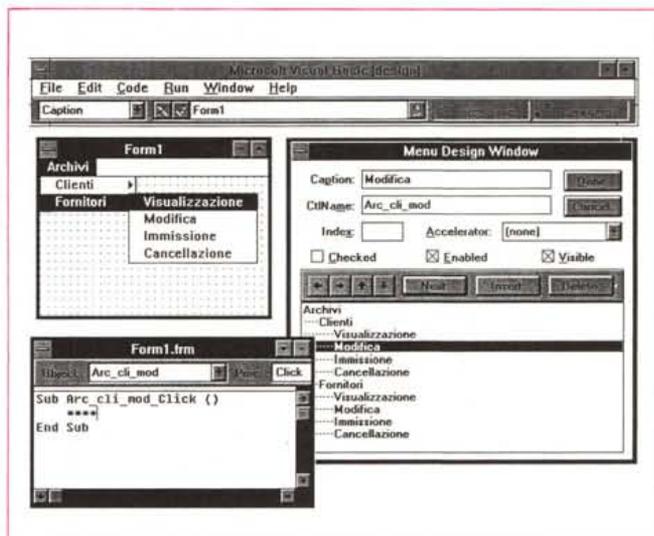


Figura 10 - MS Visual Basic - Organizzazione orientata ai Bottoni.

Cogliamo l'occasione per dire che l'organizzazione della applicazione è a totale carico dello sviluppatore, che potrà ad esempio decidere di sfruttare una complessa ramificazione del menu oppure di alleggerire il menu e ricorrere a Form Box piene di Command Buttons.

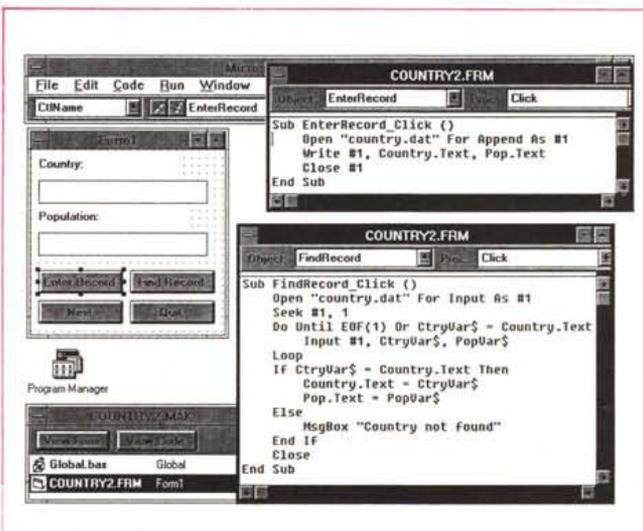


Figura 11 - MS Visual Basic - Prestazioni Grafiche.

Oltre ad utilizzare le funzioni di disegno proprie del Visual Basic, è possibile caricare in una Form o nel Control di tipo Picture immagini esterne, in formato Icona (desinenza ICO), Windows Metafile (WMF) o Bitmapmed (BMP). Il caricamento può essere statico, ovvero avviene una tantum al momento del disegno della Form, oppure dinamico e l'immagine caricata va definita al volo con una specifica istruzione.

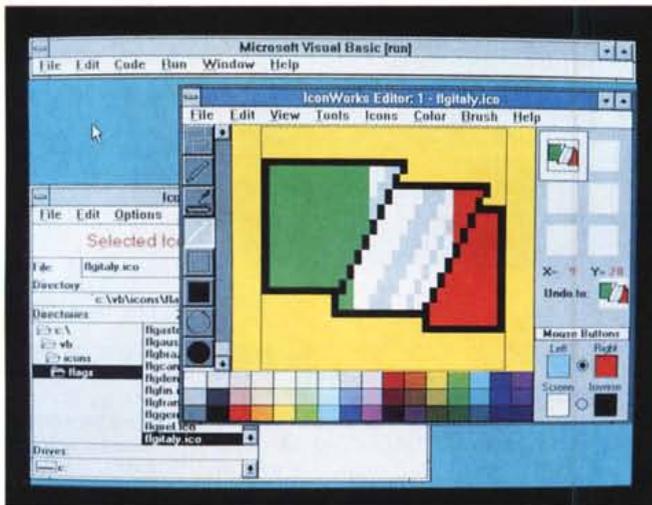
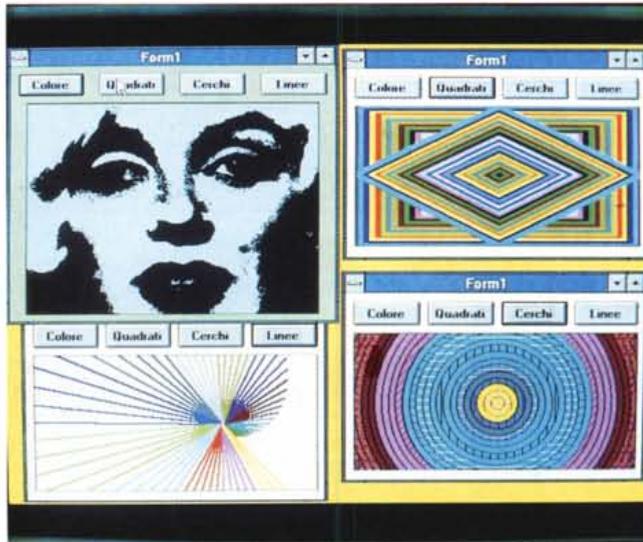


Figura 12 - MS Visual Basic - Il Generatore di Icone.

Anche l'Icona è uno degli elementi fondamentali nel mondo Windows. Il Visual Basic permette sia di utilizzarle in maniera «istituzionale», ad esempio quando si riduce ad Icona, appunto, una qualsiasi Form, sia di utilizzarle come Botto-ne Grafico in una Form, sia infine di confezionarle, disegnandole con un programma per la Generazione di Icone disponibile tra gli Examples di programmi fatti col Visual Basic stesso.

noto può servire anche da convertitore dal più diffuso formato PCX, sia di tipo vettoriale, ed in tal caso il formato è solo il Windows Metafile, WMF (in tal caso il miglior convertitore, da qualsiasi formato vettoriale, è il solito Corel Draw!).

Il terzo formato è l'ICO, che, come noto, è il formato bitmapped utilizzato da Windows per definire l'icone. I file ICO contengono un disegno di 32 per 32 pixel e occupano 766 byte (fig. 12).

I Control di tipo Picture sono Control a tutti gli effetti e quindi se vi si agisce con il mouse possono generare Eventi.

Due utilizzi possibili: il primo è un Picture Control il cui contenuto varia al variare del record di un file (applicazione multimediale). Il secondo è un Bottone grafico che cambia aspetto al verificarsi di certe condizioni nel programma (ad esempio un semaforo che cambia i suoi colori).

### Il «non ambiente» del Visual Basic il suo Menu

L'effetto più singolare che si avverte lanciando il Visual Basic è il fatto che appare solo una minifinestra. Larga quanto il Video e alta... tre righe, quella con il titolo, quella con il menu e quella con la Property Bar.

La riga del titolo è di tipo standard, con la barretta che attiva il Menu Controllo, i pulsanti Ingrandisci, Ripristina e Riduci ad icona. Il Menu (fig. 13) ha solo sei voci, che elenchiamo e descriviamo brevemente:

**File**, che permette di salvare il Progetto (tutti i Form e i Moduli dell'applicazione) o i suoi elementi singoli. Permette di iniziare nuovi Progetti, Form o Moduli, inoltre di stampare Form e Codici e di realizzare l'eseguibile del programma, con l'opzione Make Exe.

**Edit**, contiene tutti i classici Undo, Cut, Paste e Paste Link, nonché i comandi di gestione della griglia del foglio del Form. Del Paste Link parleremo tra un po'.

**Code**, attiva il menu che contiene i comandi da utilizzare in fase di Editazione di un modulo di codice. Load e Save, Find e Replace. L'organizzazione dei moduli del singolo Form è tale che i vari «pezzi» di programma possono essere visualizzati come pagine successive di un singolo documento e stampati tutti insieme con un unico comando. L'ultima voce permette l'attivazione del Syntax Checking contestuale alla scrittura, che si può considerare un primo livello di Debug.

**Run**, permette di lanciare e stoppare l'esecuzione del programma e contiene alcuni comandi utili per il Debug, come quelli per lanciare l'esecuzione passo passo e quelli per definire dei BreakPoint. È da citare la presenza di funzionalità che permettono di verificare il valore delle variabili e di testare anche singole istruzioni scritte lì per lì all'interno del programma.

**Window**, serve per attivare le varie Windows con i vari tool, che ripetiamo: Toolbox, Menu Design, Project, Color Palette, e la Immediate Window, che è quella che permette in fase di Debug di leggere i valori delle variabili.

Infine il menu **Help**, che contiene i vari comandi per i vari accessi al materiale di supporto, Help ed Tutorial.

Ci permettiamo di dare un consiglio a chi voglia «accelerare» i tempi di apprendimento del Visual Basic. È quello di realizzare una mini applicazione organizzata su un Menu in cui inserire i numerosi programmi di Example presenti nell'Help. Ad esempio una voce principale Grafica, con una voce secondaria Linee, cui collegare semplicemente (con il Taglia e Cuci, per intenderci) il programma di Esempio presente nella pagina dell' Help relativa a Line.

### Molto altro

Fino a questo punto dell'articolo abbiamo consumato circa il 90 per cento dello spazio a disposizione e quindi gli ultimi argomenti, i più evoluti, non possiamo fare altro che accennarli brevemente. Ricordiamo che tra poche pagine troverete una decina di esercizi elementari e che possono essere considerati complementari rispetto a questo articolo.

Per descrivere un prodotto così complesso, i cui soli manuali, tralaltro molto sintetici, occupano circa 800 pagine, occorrerebbero ancora molte pagine. Ma la sua importanza è tale che sicuramente avrà successo e quindi ne ripareremo più volte. Ecco comunque altri temi evoluti:

Altri Eventi. Abbiamo parlato di programmazione Event Driven riferendoci in genere all'evento più semplice che è il fatidico «click» del mouse sul Control. Ma che succede se sul Control facciamo un Doppio Click, oppure, tenendo contemporaneamente premuto il tasto shift, facciamo Click, oppure se cerchiamo di spostare, tenendo premuto il tasto del mouse, il Control stesso, oppure

se, abbandonando la tastiera, premiamo improvvisamente Ctrl e F5.

Insomma Visual Basic permette di programmare anche una serie di Eventi, collegati ai Control, collegati alla tastiera, collegati al tempo.. o non collegati a nulla.

**Il Timer.** È un Control che permette di definire un Event che si verifica ogni tanto, a seconda di un certo valore di Intervallo impostato. Cito un'applicazione che ho dovuto sviluppare per motivi di... famiglia. Poiché i miei due figli ogni tanto si mettono a giocare o a disegnare con il mio computer, ho fatto un programmino, regolato proprio dal Timer che dopo 15 minuti fa apparire una Form con una bella scritta «Adesso basta!, fate lavorare papà».

**Tool di File Manager.** Altra presenza «inquietante» (la prima è il Timer) tra i Tools del ToolBox è quella dei tre Control che permettono di definire Drive, Directory e File. Questi strumenti servono sicuramente in una applicazione in cui l'utilizzatore, ad un certo punto, decida di leggere un file.

Il fatto che si tratti di strumenti di uso frequente, in quanto presenti in qualsiasi applicativo, ce li rende familiari. In realtà, vedendoli da dietro le quinte, diventano strumenti molto evoluti, con i quali si possono realizzare applicazioni con sofisticate funzionalità di interazione con i file su disco.

**Funzioni I/O.** Il Visual Basic dispone delle stesse funzioni di gestione dei File presenti nel Quick Basic. Si possono gestire file di tipo Sequenziale o di tipo Random o di tipo Binario. È chiaro che l'applicazione più immediata è quella di realizzare una Form, in cui siano inseriti dei Control di tipo Text Box, i cui dati variabili siano letti via via da uno o più file esterni. Nel citato articolo pratico presentiamo un paio di esercizi su tale argomento.

**DDE.** Il Dynamic Data Exchange è una delle caratteristiche più evolute (e per molti di noi, più affascinanti) di Windows. Gli utilizzatori dei prodotti Windows più attenti probabilmente già conoscono le tecniche del DDE, in quanto sanno eseguire operazioni di «Copia» e «Incolla Collegamento», tra applicativi differenti. Tali tecniche sono basate su funzioni interne di Windows che, con i prodotti più programmabili, come Visual Basic ovviamente, sono direttamente sfruttabili attraverso istruzioni di programmazione.

Nella figura 14 si vede abbastanza chiaramente come, aperto un altro applicativo, nel nostro caso Excel 2.1, sia possibile eseguire il Copia e Incolla tra una cella di Excel e un Text Box di Visual Basic, o tra un Grafico di Excel e una Picture di Visual Basic.

Esistono, sia nella Text Box che nella Picture Box, delle property che servono proprio a definire le specifiche del collegamento.

Oltre ad un uso diretto del DDE è possibile un uso via programma. Esistono istruzioni per attivare e gestire il collegamento. È ad esempio possibile attivare l'applicativo collegato e poi eseguire dei comandi all'interno dello stesso.

Insomma potremmo, gestendo il tutto da Visual Basic, aprire Excel, farlo lavorare un

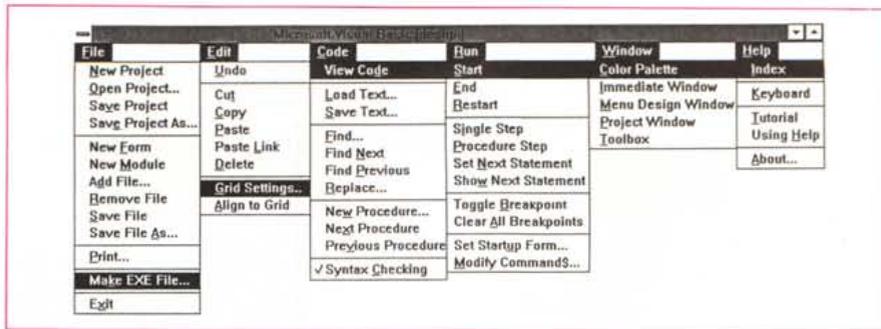


Figura 13 - MS Visual Basic - Il Menu.. al completo. In questo Collage possiamo verificare l'organizzazione del menu principale, in cui accanto a voci presenti in tutti i prodotti (File, Edit, Windows, Help), troviamo voci tipiche in un linguaggio di programmazione (Code e Run).

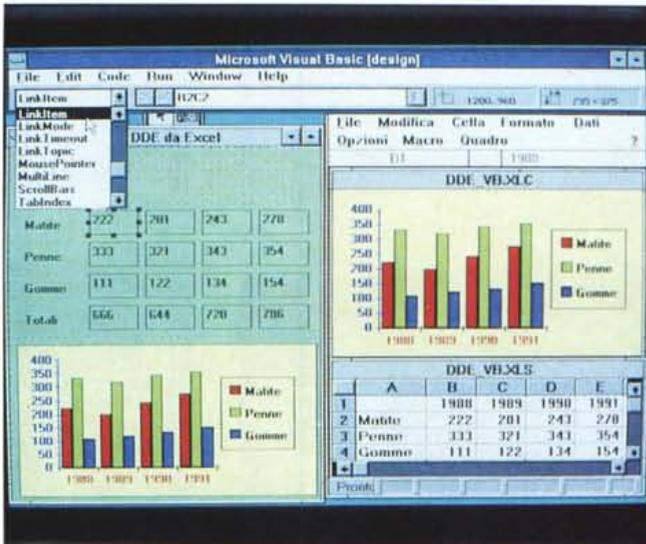


Figura 14 - MS Visual Basic - Teoria e Pratica del DDE. Gli utilizzatori dei prodotti Windows più evoluti probabilmente già conoscono le tecniche del DDE, in quanto sanno eseguire operazioni di «Copia» e «Incolla Collegamento» tra applicativi differenti. Forse però non sanno che tali tecniche sono basate su funzioni interne di Windows e che, con i prodotti più programmabili, come Visual Basic ovviamente, sono direttamente sfruttabili attraverso istruzioni di programmazione.

pochetto, e poi ricevere i dati calcolati da Excel stesso.

**DLL.** Mentre le funzioni di Dynamic Data Exchange sono proprie di Windows, per sfruttare le funzionalità permesse da Librerie Dinamiche (la famose DLL), occorre disporre sia di comandi in grado di farlo, sia di librerie dinamiche e della relativa documentazione.

Visual Basic dispone di tutta la potenzialità, in termini di istruzioni, del caso. Non ha purtroppo proprie DLL di esempio, che potrebbero costituire un buon banco di prova per l'utilizzatore che volesse cimentarsi in questa materia interessantissima.

**Conclusioni**

Siamo di fronte ad un prodotto cui va pronosticato un lungo futuro, sia perché ne vedremo nel corso degli anni un bel po' di versioni successive (questo è «solo» il Visual Basic versione 1.0) sia perché lo stesso si presta ad essere dotato di accessori, sviluppati da terzi, che ne amplificano le potenzialità. E le conclusioni che possiamo trarre da queste prime settimane con il Visual Basic non possono prescindere da queste considerazioni.

Ne abbiamo comunque subito apprezzato l'interfaccia assolutamente intuitiva, a tutti i livelli, quindi il ToolBox, il Menu Design, ecc.

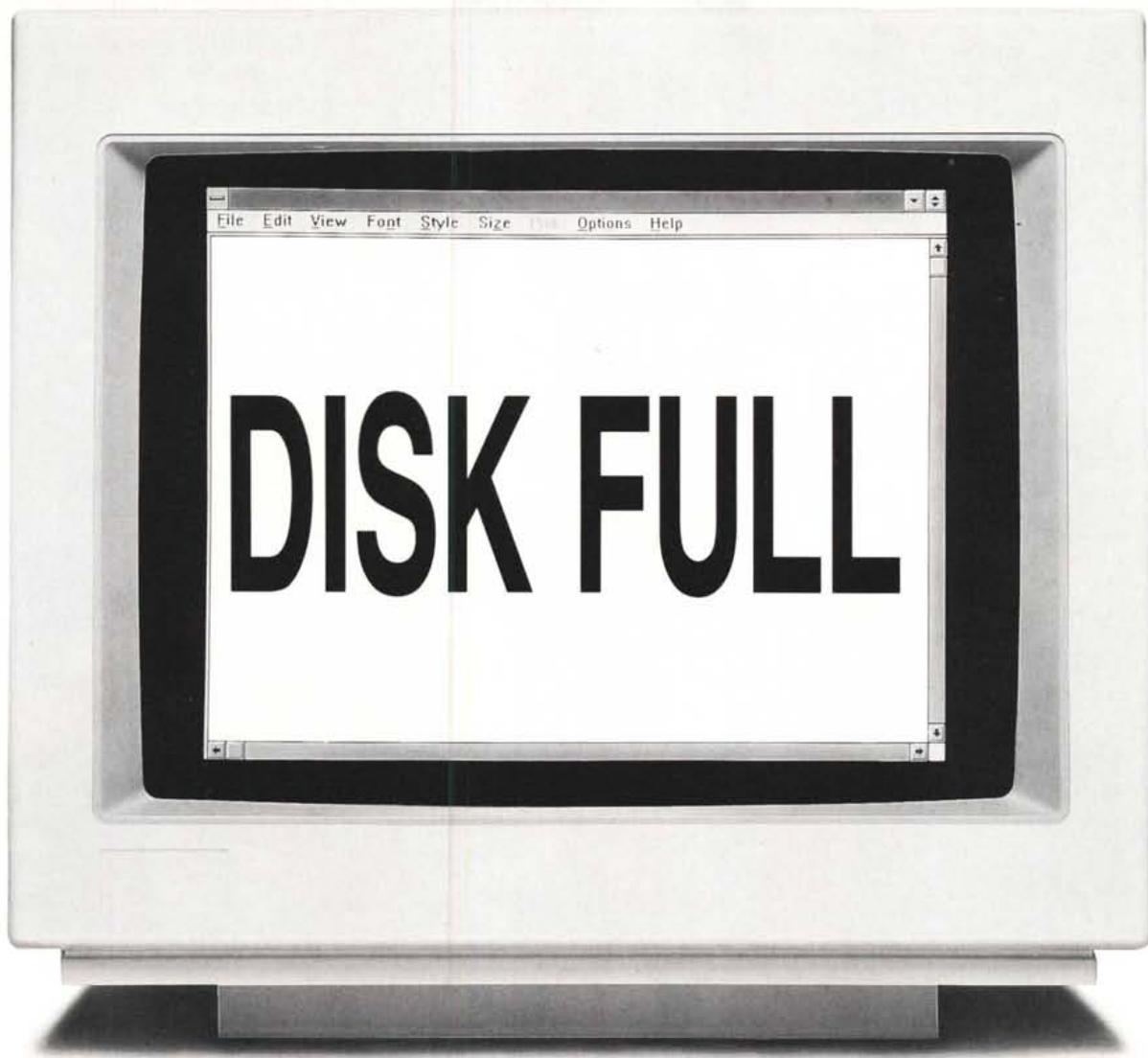
che permette sicuramente un uso immediato per fare le prime semplici applicazioni.

Si lavora molto con i Tools, con i Control, con le Property, ecc. La programmazione, che costringe alla fin fine a scrivere righe di programma, diventa la fase finale, quasi accessoria, e comunque ben localizzata nel lavoro. E tale fase è comunque agevolata dai vari strumenti di Help, Tutorial e dalle sofisticate funzioni di Debug.

Ne abbiamo apprezzato, anche se dovremmo ritornarci per i necessari approfondimenti, la ricchezza delle istruzioni di tipo «interoperativo». Esistono strumenti per costruire Form di tipo File Manager, funzioni per dialogare in varie maniere, non solo attraverso il DDE, con le altre applicazioni Window. Esistono istruzioni per attivare funzioni esterne messe a disposizione dalle DLL.

Se Windows ha reso il lavoro con il PC divertente oltre che utile, il Visual Basic può rendere anche la programmazione divertente e probabilmente un po' più facile.

Ma attenzione! La programmazione è una attività troppo seria per poter essere solo o soprattutto divertente. Ricordiamoci che il «successo» di una applicazione dipende da tanti fattori, non solo dal fatto che l'interfaccia utente sia di tipo GUI e che il programmatore sia stato «veloce» e «creativo» nel disegnarla.



## HARDCARD: IL SISTEMA PIU VELOCE PER DARE MEMORIA AL VOSTRO PC.

Il PC è cresciuto: con le nuove interfacce grafiche, i DTP e, soprattutto, Windows, le prestazioni si sono moltiplicate, ma anche la memoria richiesta.

Logico, quindi, che 20 o 40 Mb non vi bastino più. Oggi però, con le nuove Hardcard II XL, anche l'hard disk è cresciuto.

Le Hardcard II XL sono dei veri hard disk su scheda per PC286 e 386, affidabili e facili da installare. Disponibili in versioni da 50 e 105

Mb, sono 8 volte più veloci dei modelli convenzionali ed hanno un tempo di accesso di soli 9 ms: l'ideale per

operare in un ambiente grafico come Windows, che richiede alte prestazioni.

Dalle vostre finestre non si affaceranno più problemi di spazio, e il vostro PC sarà libero di crescere ancora, in tutta tranquillità.



**Plus** 

**Editrice Italiana Software**



**DIVISIONE PRODOTTI SPECIALI**

Via Fieno 8  
20123 - Milano - MI  
Tel. (02) 809961

Via Spalato 7  
10141 - Torino - TO  
Tel. (011) 389934

Via L. Ariosto 16/2  
35128 - Padova - PD  
Tel. (049) 773939

V.le dei Cappuccini 12  
40026 - Imola - BO  
Tel. (0542) 41665

P.za Barberini 52.  
00187 - Roma - RM  
Tel. (06) 4880288

Via G. Amendola 106/T  
70126 - Bari - BA  
Tel. (080) 583744

SIAMO PRESENTI ALLO SMAU - PAD. 25 - SAL. 1 - STAND A40 - B29