

Visual Basic for Application: primi approcci

Nello scorso numero di MC abbiamo provato MS Excel versione 5.0 ed in tale occasione abbiamo annunciato un approfondimento riguardante una delle novità principali di Excel 5.0: il Visual Basic for Application, il nuovo linguaggio di programmazione, che in questa versione affianca, ma nelle prossime sostituirà totalmente, il vecchio linguaggio Macro. Altra caratteristica del VBA è quella di essere destinato a costituire il linguaggio unico di programmazione per tutti i prodotti per Windows della Microsoft e probabilmente anche di molte altre case

di Francesco Petroni e Raffaele Valensise

Una delle caratteristiche più «spinte» del VBA è quella di riconoscere OLE Automation, che, detto in parole semplici, consiste nella possibilità, per tutti i prodotti programmabili in VBA e OLE 2.0 compatibili, di essere programmati «dal di fuori». Questo significa che con VBA e OLE Automation sarà possibile costruire procedure «trasversali», che lavorano anche su più prodotti, indipendentemente dal prodotto con il quale il programma è stato scritto e indipendentemente dal prodotto dal quale il programma è stato lanciato.

È chiaro che questi obiettivi, ambiziosissimi, sono per ora praticabili a titolo di esperimento. È ancora presto per ipotizzare un uso affidabile e performante di VBA e OLE Automation in un'applicazione chiusa. Una cosa è sicura: tra pochissimo il VBA, con tutte le sue caratteristiche, sarà una specie di strato unificante i vari applicativi operanti nell'ambiente.

Spreadsheet e Macro, un sodalizio che funziona

Il primo spreadsheet di larghissima diffusione è stato l'123 versione 1.a. Risale al 1982, e già disponeva di un suo linguaggio Macro, basato sulla codifica dei normali comandi che potevano in tal modo essere trascritti, da qualche parte sul foglio, per poi essere eseguiti tutti insieme. E già nella prima versione dell'123 c'erano anche altri comandi, non usabili in modalità diretta, ma utili in una Macro. Ad esempio c'erano comandi per consentire un input dei dati, alternativo a quello eseguito direttamente nelle celle, c'erano comandi per la generazione di menu personalizzati, ecc. La caratteristica principale del si-

stema macro era che tutto avveniva sullo stesso foglio di lavoro, e questo causava al programmatore numerosi problemi di natura organizzativa.

Nelle versioni successive dell'123 il linguaggio si è ovviamente arricchito, ed è stato affiancato dall'utilissimo registratore di Macro, che facilita la fase di

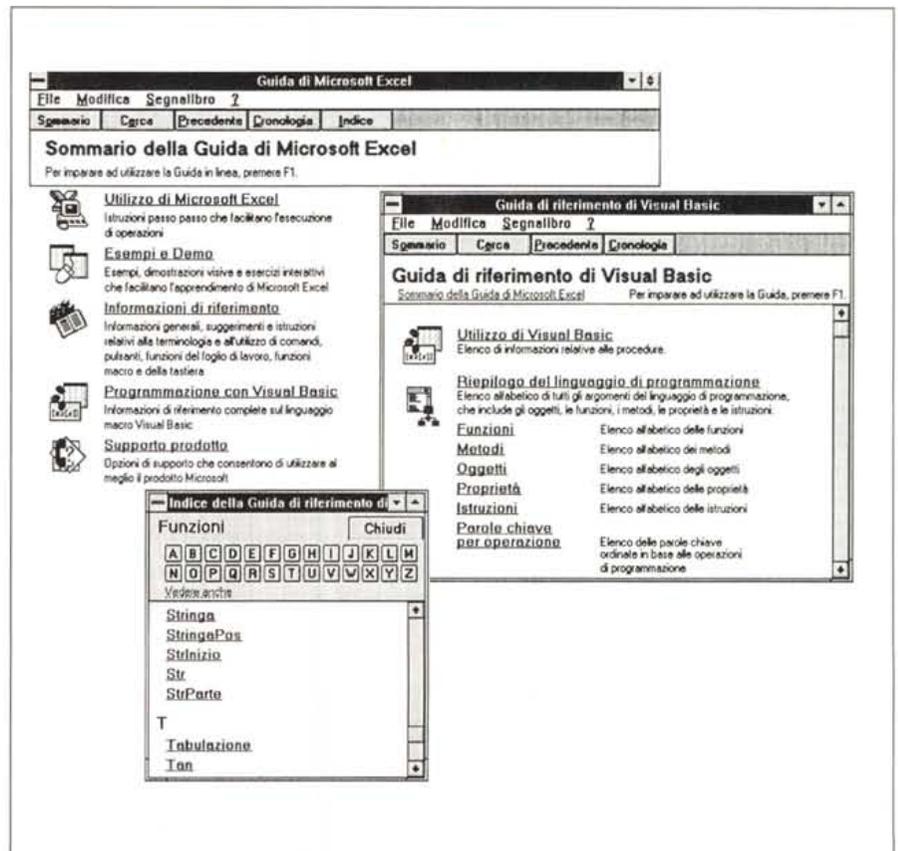


Figura 1 - MS Excel 5.0 - La Guida di Visual Basic for Application.

In queste tre schermate vediamo una delle modalità di approccio con l'Help del Visual Basic. Si parte dal Sommario generale della Guida, dal quale si passa alla sezione Programmazione con Visual Basic, che, a sua volta, ha un suo Sommario, dal quale si può, ad esempio, accedere all'elenco alfabetico delle Istruzioni, degli Oggetti, dei Metodi e delle Funzioni. Avvertiamo subito che le funzioni di VBA non sono, in generale, le stesse, né come numero né come sintassi, disponibili nel foglio normale.

trascrizione dei comandi. Excel, nella sua prima versione (ricordiamo che si tratta di un prodotto nato direttamente in Windows), ha da una parte conservato molte delle scelte fatte dalla Lotus, dall'altra ha introdotto una serie di novità molte delle quali imposte dall'ambiente grafico. Tra le più significative: il ricorso ad un tipo speciale di foglio, il foglio Macro, la possibilità di costruire delle Macro Funzioni, che costituiscono un livello intermedio di programmazione, la possibilità di codificare, anche ricorrendo ad un Editor Grafico, una Dialog Box che può essere richiamata dalla procedura.

Le versioni, sia dell'123 che di Excel, si sono succedute senza grosse novità. Nel frattempo anche la Borland, soprattutto con il suo Quattro Pro per Windows, ha dato ulteriori contributi, in particolare ha adottato la Cartella di Lavoro, che racchiude più fogli, e all'interno della quale può essere disegnata la Dialog Box, che come noto è lo strumento principe per il dialogo tra utente e applicazione.

Ed eccoci finalmente arrivati al VBA, nuovo strumento di programmazione, appartenente ad una nuova generazione di strumenti di programmazione (la tecnologia nel suo complesso si chiama Visual Programming), in cui vengono anche messi a frutto i quasi quindici anni di storia dello spreadsheet e i contributi delle varie case, che hanno, in questi anni, proposto propri spreadsheet, e le esperienze di centinaia di migliaia di utenti evoluti.

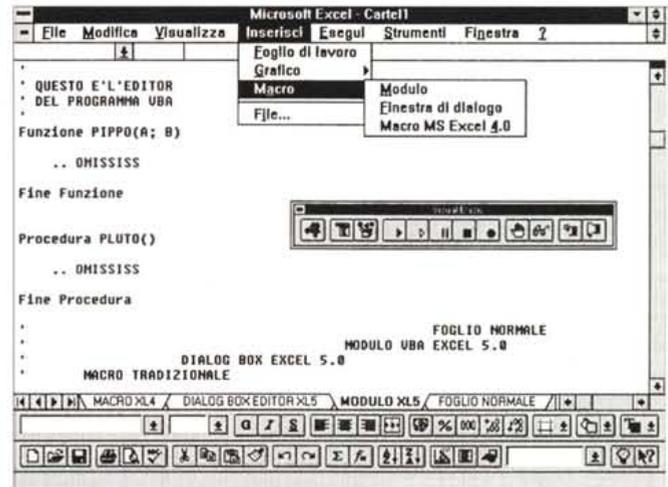
Un paio di anni fa Microsoft ha lanciato anche il Visual Basic e successivamente il Visual C. Questo lo diciamo ad ulteriore conferma della globalità della strategia legata al Visual Programming. Comunque, nel parlare del VBA di Excel non faremo, per ora, riferimento al Visual Basic normale, mentre dovremo ogni tanto, ma solo per indicare alcune differenze sostanziali, citare il vecchio linguaggio Macro.

La nuova organizzazione interna di Excel 5.0

Prima di parlare di VBA è indispensabile precisare la nuova organizzazione interna di Excel, basata sul concetto di Cartella di Lavoro. Con Excel 5.0 si lavora su Cartelle di Lavoro. Il file con la cartella di lavoro ha estensione XLS (come i vecchi file Excel 2.0, 3.0, 4.0). Una cartella di lavoro però contiene fogli, da 1 fino ad un massimo di 256. I fogli vengono fisicamente evidenziati dalla classica linguetta, che mostra il nome del

Figura 2 - MS Excel 5.0 Accesso alle varie funzioni di programmazione.

Excel 5.0 lavora con il concetto di «cartella». In una cartella ci possono essere più tipi di foglio, il Foglio Normale, il Foglio Macro per le macro XL4, scritte cioè alla vecchia maniera, il foglio con le Dialog Box, usabili in una procedura, e il Modulo, che è quello che ci interessa di più, in cui vanno scritte le Funzioni o Procedure. In caso di uso del registratore di Macro, vengono prodotti automaticamente i Fogli Macro del vecchio tipo, oppure Moduli con le istruzioni VBA. Dipende dall'opzione impostata. Un ultimo tipo di foglio è quello che ospita i Grafici.



foglio. Esistono numerosi comandi per gestire i fogli (evidenziazione, inserimento, cancellazione, spostamento, ecc.). È anche possibile scrivere formule a cavallo di più fogli (es. la somma della cella A1 di dodici fogli).

Esistono cinque tipi di foglio:

- Foglio Dati tradizionale. È lo spreadsheet classico;
- Foglio Macro alla vecchia maniera. Assomiglia al precedente solo che le colonne appaiono più larghe e si possono inserire anche le funzioni di programmazione;
- Foglio Dialog Box. È nuovo. È un foglio a quadretti piccoli, in cui si può inserire il disegno di una sola dialog box (una sola per foglio). Il foglio dialog box in pratica sostituisce il vecchio Dialog Box Editor. Da una parte permette, come vedremo tra un po', di disegnare delle box più ricche (ad esempio c'è an-

che la Scroll Bar), e dall'altra facilita l'uso, da parte del programma, della box che viene identificata semplicemente dal suo nome;

- Foglio Grafico. Un grafico prodotto con le nuove potentissime funzionalità di Excel 5.0 può essere piazzato su un foglio Dati o su un foglio Grafico;

- Foglio Modulo. È il foglio nel quale vanno scritti i programmi che possono essere funzioni oppure procedure.

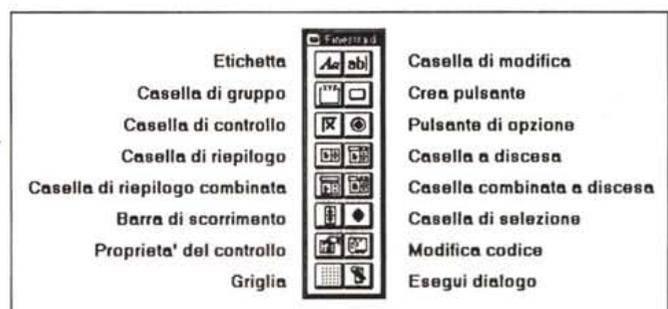
Al lancio di Excel viene proposta una Cartella con 16 fogli dati. Essendo tale valore impostabile come opzione generale conviene subito portarlo ad 1, tanto poi è possibile inserire i fogli in più, quando effettivamente servono.

Il menu, per quanto riguarda la gestione dei fogli, in concreto propone le voci:

- Inserisci Grafico su questo foglio,
- Inserisci Grafico crea un nuovo foglio,

Figura 3 - MS Excel 5.0 Barra dei pulsanti per la Finestra di Dialogo.

In Excel 5.0, così come in Word per Windows vers. 6.0, è possibile inserire direttamente sul foglio di lavoro alcuni oggetti alla Windows (caselle di controllo, pulsanti di opzione, caselle di riepilogo, ecc.) la cui gestione non deve necessariamente essere legata alle macro. Nella figura presentiamo il campionario degli oggetti proposti dalla barra dei pulsanti chiamata Finestra di dialogo, precisando che non tutti gli oggetti sono inseribili in una box posta direttamente sul foglio, alcuni di questi sono usabili solo in un foglio di tipo Dialog Box.



- Inserisci Macro Modulo,
- Inserisci Macro Finestra di Dialogo,
- Inserisci Macro Macro MS Excel 4.0.

Inoltre se si attiva il Registratore, cosa che si fa con il comando Strumenti Registra Macro, viene creato automaticamente un nuovo foglio del tipo Modulo. A seconda del tipo di foglio attivo è presente uno specifico Menu, proprio di quel tipo di foglio, ed in particolare sono attivi altri comandi per creare gli altri tipi di foglio. Ad esempio partendo da un foglio con un grafico si può inserire un nuovo foglio dati o modulo, e così via.

In figura 1 vi proponiamo un collage di comandi, di fogli e di toolbar. Siamo piazzati su un foglio di tipo Modulo, in pratica si tratta di un ambiente Editor nel quale possiamo scrivere due tipi di codice, Funzioni oppure Procedure.

Per chi vuole assemblare più fogli preesistenti, magari realizzati con il vecchio Excel 4.0, in un'unica cartella, c'è il comando Inserisci File.

Chi invece vuole disassemblare un foglio, estraendolo dalla cartella per dargli dignità di foglio a sé stante, c'è il comando di salvataggio con nome, che permette di salvare i fogli individualmente, magari in altro formato. È chiaro che si perdono le formule «trasversali», quelle a cavallo tra più fogli.

Documentazione

VBA è una tecnologia nuovissima, quindi si deve in qualche modo pagare lo scotto della novità. La documentazione cartacea è costituita dal Manuale dell'Utente di Visual Basic, fornito con Excel, di circa trecentocinquanta pagine. Il manuale, per quanto ben organizzato e completo, non può raggiungere un livello di approfondimento sufficiente per le necessità più spinte del programmatore professionista, essendo ovviamente il VBA un linguaggio complesso, non fosse altro per il fatto che comprende anche tutti i comandi di Excel normale.

Anche la Guida è ben organizzata e completa. Nel trattare i vari argomenti vengono presentati numerosi esempi di codice, copiabili per trasferirli nel proprio modulo di programmazione (fig. 2), ma di esempi ne servirebbero cento volte tanti.

Va anche chiarito il fatto che non è possibile sfruttare immediatamente il materiale disponibile, ed in questo caso è abbastanza voluminoso, relativo all'altro VB, il Visual Basic quello normale, chiamiamolo così.

Il VBA è infatti allineato al Visual Basic quello normale, ma solo limitatamente alle istruzioni di programmazione standard. È chiaro che però cambiano totalmente gli oggetti trattati, che in Excel,

Figura 4 - MS Excel 5.0 Un nuovo tipo di finestra di dialogo.

Siamo dell'idea che prima di ricorrere alla programmazione occorra esplorare tutte le possibili alternative. In questo esempio, meglio descritto nel testo, vediamo come sia possibile in Excel 5.0, piazzare degli oggetti alla Windows, ad esempio dei Pulsanti, delle Liste, ecc. direttamente sul foglio, che in tal modo diventa una sorta di Dialog Box. È chiaro che in questa box alcuni elementi sono normalissime celle del foglio, altri sono elementi esterni, aggiunti prelevandoli dalla Toolbar. In tal caso i vari oggetti debbono far riferimento ad altre celle, in modo che gli eventuali calcoli possano essere eseguiti con ambedue i tipi di dati.

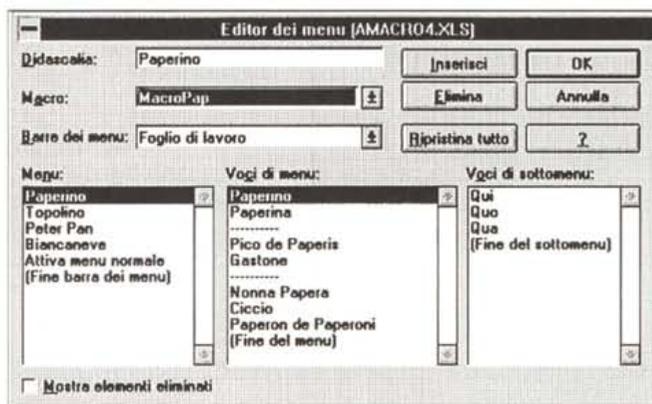
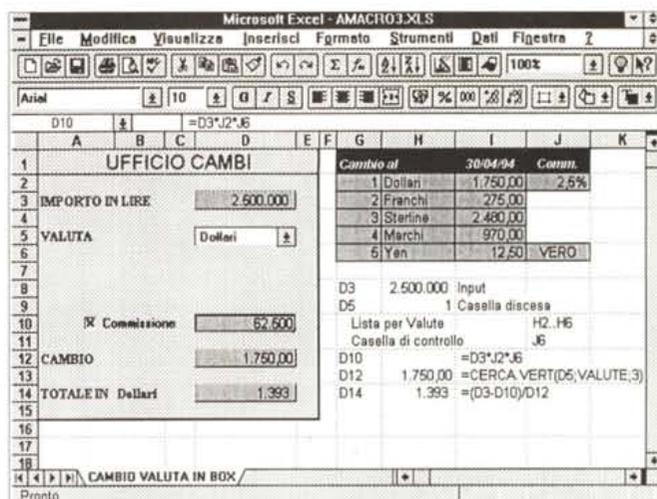
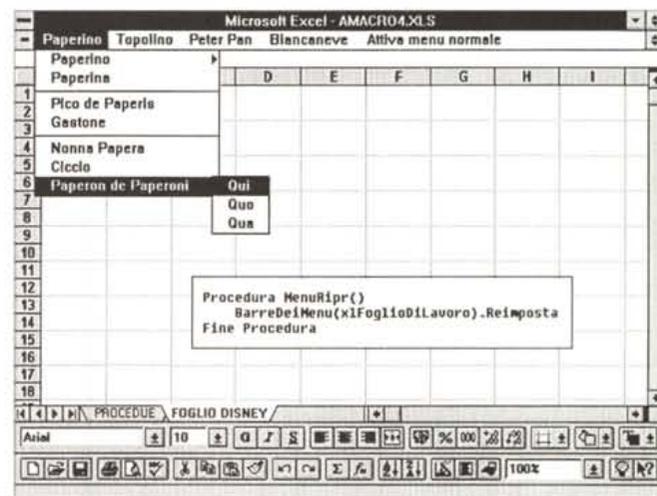


Figura 6 - MS Excel 5.0 Menu personalizzato.

Nella nostra applicazione che mostra il menu, il problema non sarà tanto quello di costruire il menu, cosa che si fa facilmente con l'editor visto prima, quanto quello di collegare ogni voce del menu ad una Macro. Una voce di menu, che dovrà essere sempre presente in un menu personalizzato, è quella che riattiva il menu normale. Al centro della figura vediamo proprio la Procedura, di una sola istruzione, che ripristina il menu normale. Oltre all'editor ci sono numerose istruzioni che permettono di gestire, via programma, il menu, a seconda delle esigenze.



sono Fogli, Dialog Box, Zone di celle, ecc. Altra differenza tra VBA e VB sta nel fatto che il primo è stato tradotto in italiano il secondo no, anche nei comandi.

Le due differenze rendono praticamente intrasferibile eventuali programmi

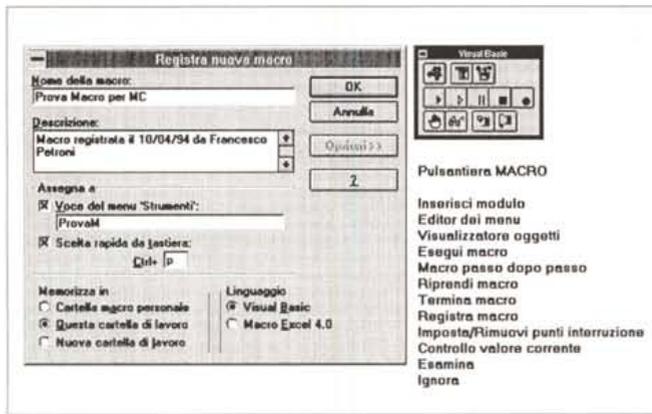
scritti con il Visual Basic in un modulo Excel.

Indice dei nostri esercizi

Cominceremo col vedere una forma

Figura 7 - MS Excel 5.0
Registrazione della Macro.

Per registrare una macro occorre attivare il comando **Strumenti Registra Macro** che permette una serie di impostazioni, quali nome della macro, commento, tasto scorciatoia e una serie di opzioni. Va scelto innanzitutto il linguaggio desiderato, se VBA o Macro Excel 4.0, e poi che fine far fare al codice, che può essere memorizzato nella cartella personale delle macro, nella cartella corrente, oppure in un'ulteriore cartella. Sulla destra mostriamo la barra con i pulsanti relativi alla programmazione, controllo ed esecuzione delle Macro.



possibile inserire direttamente sul foglio di lavoro (sul foglio normale) alcuni oggetti alla Windows (caselle di controllo, pulsanti di opzione, caselle di riepilogo, ecc.) la cui gestione non deve necessariamente essere legata a macro.

In figura 3 vediamo il campionario degli oggetti, proposti dalla barra dei pulsanti chiamata Finestra di dialogo, e con i nomi che usa Excel, precisando che non tutti gli oggetti sono inseribili in una box posta direttamente sul foglio, ma sono comunque presenti perché la barra è la stessa che si usa (e che usiamo) nel foglio di tipo Dialog Box.

Per capire meglio il meccanismo, riferiamoci alla figura 4, in cui presentiamo una piccola gestione di Cambio Valuta in cui l'operatore deve semplicemente digitare l'importo in lire, che desidera trasformare in moneta straniera, scegliere da una lista la valuta desiderata, e specificare se alla transazione deve essere applicato o meno un tasso di commissione, che è del 2,5%.

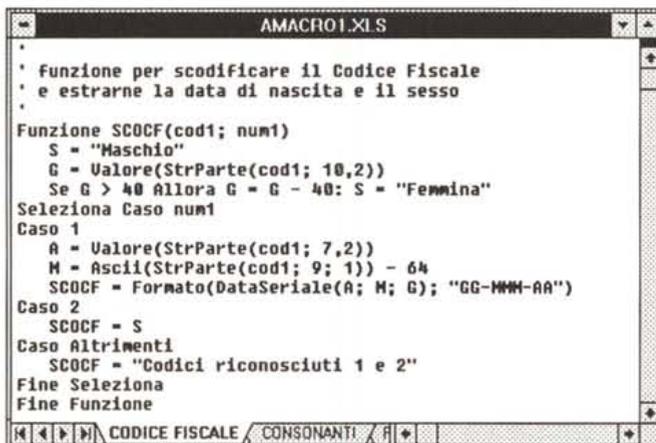
Vediamo in dettaglio la struttura di questa prima semplice procedura.

La cella D3 contiene l'input, ovvero l'importo in lire. Proprio nella cella D5 è posizionata una «Combo Box», collegata all'intervallo (H2:H6), che contiene l'elenco delle valute di cui è disponibile il valore di cambio.

Un clic sul pulsante «Casella a discesa» consente di tracciare direttamente sul foglio di lavoro la dimensione

Figura 8 - MS Excel 5.0
Uso delle Macro Funzioni.

Siamo stati sempre favorevolissimi alle funzioni «personalizzate», perché permettono di risolvere problemi di calcolo, anche molto complessi e con molte ramificazioni, senza dover ricorrere, ancora, alla programmazione tradizionale. Ne costruiamo due, la prima che estrae dal codice fiscale la data di nascita, oppure il sesso, del proprietario. La seconda che estrae, da una frase, una stringa, tutta in maiuscolo, che contiene solo le sue consonanti. Es. da Roma e Lazio la stringa sarebbe RMLZ.



più elementare di dialog box, quella che si può «piazzare» direttamente sul foglio dati, senza «scomodare» le macro.

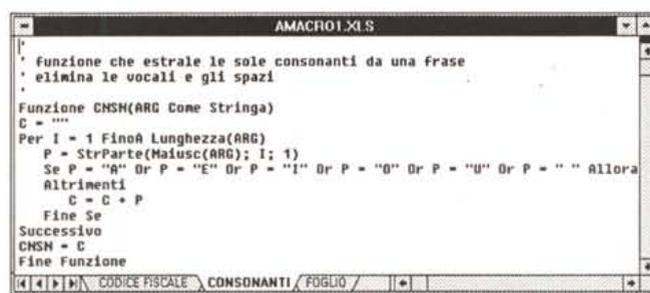
Poi utilizzeremo l'editore del menu, per realizzare un menu, anche in questo caso senza dover ancora ricorrere alle macro.

Passeremo quindi alla programmazione vera e propria realizzando dapprima un paio di funzioni personalizzate, scritte a mano, e poi creando una procedura, più tradizionale, di conteggio e confronto.

Poi costruiremo una dialog box che scambia i dati con il foglio e per finire una ulteriore dialog box più complessa, che serva da interfaccia tra utente e una tabella di dati.

Nel documentare i vari esercizi preferiamo usare delle hard copy delle idee, guadagnando in precisione di immagini ma perdendo i colori.

Non tratteremo in questo articolo, per mancanza di spazio, gli aspetti generali ed organizzativi del Visual Basic, che lavora con Oggetti, Proprietà e Metodi. Non mancherà occasione di riparla-



Il valore restituito è «maschio» oppure «femmina». Se il codice non è né 1 né 2, viene restituito un messaggio di errore.

re di VBA, probabilmente già nel prossimo numero di MC, e faremo in modo di analizzarlo anche da un punto di vista più... filosofico.

Una Dialog Box posata direttamente sul foglio di lavoro

Anche in Excel 5.0, così come nella versione 6.0 di Word per Windows, è

dell'oggetto che ci interessa. Nel caso specifico dimensioniamo un rettangolo sovrapposto alla cella D5. Ultimo il tracciamento, l'oggetto appare contornato da otto quadratini neri che ne indicano, al solito, la possibilità di modifica (l'elemento è selezionato). È a questo punto che possiamo specificarne le proprietà, attraverso il comando Formato Oggetto, disponibile sia con un clic sul

pulsante destro che nei menu standard.

La finestra «Formato Oggetto» è suddivisa nelle tre sottoschede: Protezione, Proprietà e Controllo, le prime due definiscono il blocco dell'oggetto per evitare modifiche del suo contenuto (se il foglio è protetto), della sua posizione e della sua dimensione da parte dell'utente e la sua eventuale assenza dagli output su carta.

La terza scheda, Controllo, ci permette varie cose:

- di collegare la nostra Casella all'elenco di valute disponibili specificando nel riquadro «Intervallo di Input» la zona di celle (H2:H6);
- di indicare la cella D5 (scelta da noi) nel «Collegamento cella». La moneta straniera scelta per l'operazione di cambio, verrà riportata sul foglio, nella cella indicata, sotto forma di valore numerico;
- di impostare lo spazio verticale occupato dalla Casella a Discesa attivata;
- di dare un aspetto «3D» all'oggetto.

Completate le operazioni di formattazione, l'oggetto è già funzionante. Basta un clic sul foglio e di nuovo un clic sulla freccia della Casella a Discesa per attivare il collegamento con l'intervallo di celle selezionato.

Analoghe operazioni sono necessarie per implementare la Casella di controllo che applica il 2,5% di commissione. In questo caso impostiamo il collegamento con la cella J6. Il risultato è un operatore logico del tipo «VERO=1/FALSO=0» che sfruttiamo nella cella D10 con la formula:

=D3*J2*J6

che moltiplica la percentuale di commissione per l'importo in lire e il tutto per il dato proveniente dalla Casella di controllo (che vale 0 o 1).

Occupiamoci ora del dato proveniente dalla scelta operata con la Casella a discesa.

Il risultato posto nella cella D5 è un numero, nel nostro caso compreso fra 1 e 5 poiché 5 sono le monete straniere presenti nell'intervallo (H2:H6). Nella cella D12 inseriamo quindi una formula di lookup per passare dal dato numerico disponibile in D5, al valore in lire corrispondente alla valuta scelta. La formula, da piazzare in D12, sarà:

=CERCA.VERT(D5;Valute;3)

dove «Valute» è il nome di zona con il quale abbiamo precedentemente identificato l'intervallo di celle (G2:I6), mentre «3» indica la colonna di questo intervallo contenente il cambio in lire.

Per amore di completezza inseriamo una funzione CERCA.VERT anche nella cella B14 con la quale ricaviamo il tipo di valuta:

=CERCA.VERT(D5;Valute;2)

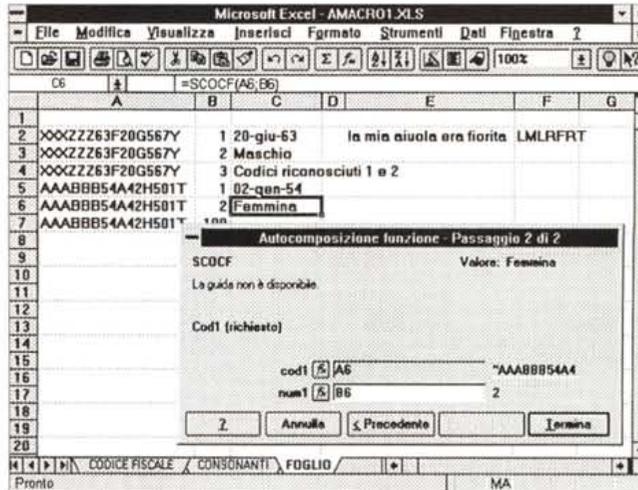
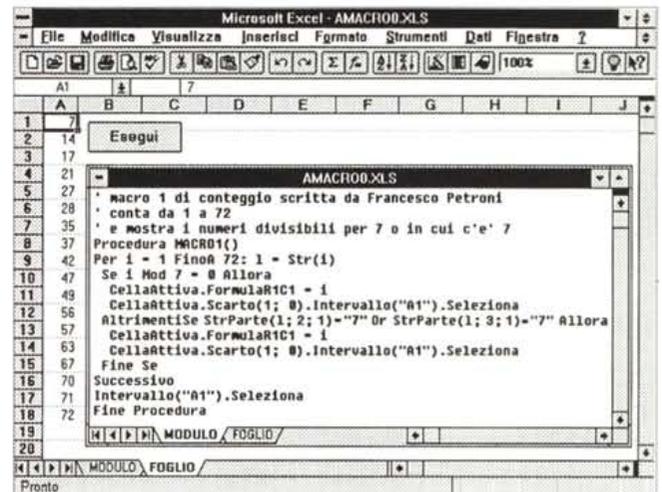


Figura 11 - MS Excel 5.0 - Procedura di conteggio e confronto.

Un classico programma «didattico», che riproponiamo ogni volta che ci capita di affrontare un nuovo linguaggio di programmazione. Vogliamo contare da 1 fino a 72 (per motivi di videata), e vogliamo visualizzare solo i numeri divisibili per 7 e quelli che contengono, in prima o seconda posizione, un 7. Il risultato sarà quindi 7, 14, 17, 21, 27, 28... La procedura, che risiede in un foglio modulo, dovrà essere eseguita da un Pulsante presente sul foglio.



La formula con il calcolo finale va posta nella cella D14:

=(D3-D10)/D12

ovvero, l'importo in lire, decurtato delle eventuali spese di commissione, diviso per il cambio della valuta prescelta.

Con questo sistema, che chiameremo Dialog Box senza programmazione e senza Macro, sarà possibile realizzare delle applicazioni di bassa complessità, chiuse e controllabili, anche grazie al fatto di poter impostare la protezione delle celle, da far usare ad un utente non esperto.

Un menu alternativo

Per creare un menu personalizzato che modifichi o sostituisca totalmente quello normale (in realtà i menu normali sono tanti quanti sono i tipi di foglio) ci sono tre strade. O si usa l'Editor del Menu, attivabile dal menu del foglio di tipo Modulo (non andate a cercarlo nel menu del foglio normale), o si usano le istruzioni di programmazione che permettono di creare nuove barre, di inserire o eliminare voci di primo, secondo

Figura 10 - MS Excel 5.0 - Uso delle Macro Funzioni. Funzione CSCN.

Diciamo subito che le macro funzioni in VBA possono essere molto più complesse e sofisticate di quelle scritte con il vecchio linguaggio Macro, soprattutto per il fatto che ora possono contenere cicli, ramificazioni, salti. In questo caso, con un ciclo che scorre tutta la stringa in input, vengono testati, uno per uno, tutti i caratteri, per poi ricostruire la stringa, fatta di sole consonanti, in uscita.

e terzo livello. La terza strada è quella mista, che consiste nel realizzare l'impianto del nuovo menu con l'Editor, e nel lasciare alle varie fasi del programma la sua gestione. Ad esempio la disattivazione di quelle voci non usabili ad un dato momento logico (le voci in grigio).

Nella figura 5 vediamo proprio l'editor del menu. La figura fa vedere molto chiaramente il fatto che è possibile creare tre livelli di menu e che è possibile, con un carattere «meno», tracciare una riga che funge da divisore tra le varie voci del menu.

È evidente che il grosso del lavoro non è creare il menu ma scrivere le varie procedure sottostanti ciascuna voce. Nella successiva figura 6, nella quale vediamo il menu in azione nel foglio dati, c'è anche la procedura che serve a ripristinare il menu normale, e che va assegnata all'ultima voce del menu (come realizzarla, lo vedremo dopo).

Va detto che anche in un'applicazione semplice, composta magari dai soli fogli dati e modulo, quest'ultimo può essere «nascosto» alla vista con uno

Figura 12 - MS Excel 5.0 - Istruzione Mentre. Per contare è possibile utilizzare più di una istruzione. Qui vediamo, in una videata di Help (l'Help in linea è copiabile nel senso che è possibile trasferire il codice dalla videata di Help direttamente nel Modulo su cui si sta lavorando), l'istruzione Mentre... FineMentre, alternativa al Per... Successivo. Da citare anche una... promettente Esegui... Ripeti finché.

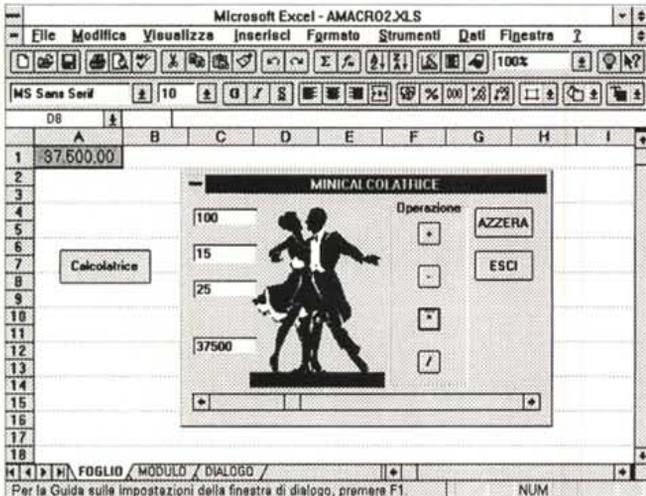
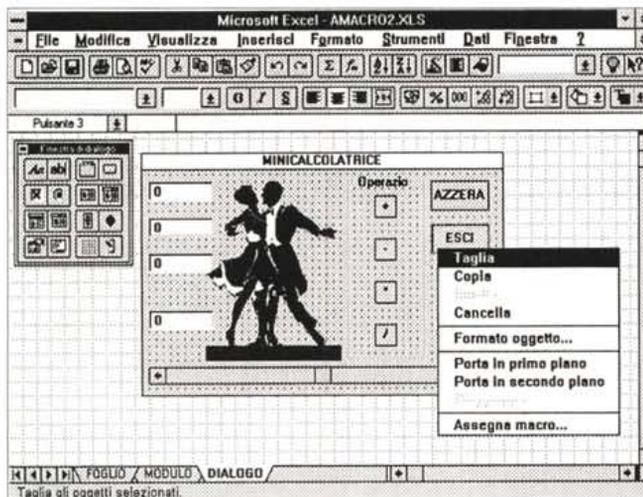


Figura 14 - MS Excel 5.0 - La calcolatrice al lavoro. Si tratta di un'applicazione inutile dal punto di vista pratico ma che ci permette di studiare un argomento importantissimo e cioè come trasferire i dati da un oggetto all'altro. Sono oggetti le celle, anzi le zone del foglio, che possono essere anche di una sola cella, sono oggetti le Caselle di Modifica della calcolatrice.



specifico comando. Non è invece necessario lanciare l'esecuzione del menu, che appare automaticamente sul foglio dati per il semplice fatto che è il menu di stato creato.

Per registrare

Passiamo ora alle macro vere e proprie, quelle che fanno uso del foglio di tipo modulo. Il primo passo, nell'avvicinamento ai moduli, consiste nella realizzazione di una macro, sfruttando il registratore.

Il comando, che è Strumenti Registra Macro, prevede una serie di opzioni: quale linguaggio utilizzare, VBA o Macro Excel 4.0, e dove si vuole che vada a finire il codice scritto da registratore. Abbiamo scelto di farlo trascrivere sulla stessa cartella di lavoro.

Nella figura 7, che è un «collage» in

cui vediamo la citata box, abbiamo messo anche la barra dei pulsanti che appare quando si è alle prese con la macro. In basso la descrizione dei vari pulsanti.

La registrazione di una macro è opportuna in due casi. Quando effettivamente la procedura da automatizzare è totalmente percorribile via comandi diretti, oppure quando si stia scrivendo a mano una procedura, non ci si ricorda un comando che però è presente come comando normale. In questo caso si accende un attimo il registratore, si esegue il comando, si ferma la macro e si copia il comando trascritto dal registratore nella propria procedura. Questo sistema è in genere più rapido di quello che consiste nell'andare a trovare il comando nell'help o in un manuale.

Due funzioni personalizzate

Chi ci segue nei nostri articoli divulgativi riguardanti l'uso degli spreadsheet, sa che siamo stati sempre favorevoli alle funzioni «personalizzate», perché permettono di risolvere problemi di calcolo, anche molto complessi e con molte ramificazioni logiche, senza dover ricorrere alla programmazione tradizionale.

Diciamo subito che le macro funzioni in VBA possono essere molto più complesse e sofisticate di quelle scritte con il vecchio linguaggio Macro, soprattutto per il fatto che ora possono contenere cicli, ramificazioni, salti.

Nel successivo esercizio, documentato dalle figure 8, 9 e 10, presentiamo due macro funzioni: la prima che estrae da un codice fiscale o la data di nascita o il sesso del proprietario, la seconda che estrae da una frase, una stringa tutta in maiuscolo, che contiene solo le sue consonanti. Es. da Roma e Lazio la stringa risultante è RMLZ.

Per scrivere una macro funzione occorre inserire un nuovo foglio di tipo modulo e scrivere subito le due istruzioni, quella iniziale e quella finale, della funzione. Es:

```
Funzione SCOCF (cod1;num1)
Fine Funzione
```

In questa maniera viene identificata la nuova funzione SCOCF, alla quale vanno passati due parametri, che andrebbero a rigore dichiarati per indicarne il tipo, operazione che però non è obbligatoria per il funzionamento della funzione (in tal modo però rinunciando al controllo sul tipo dei dati).

All'interno di tali due istruzioni va scritta la procedura vera e propria.

Nel nostro caso a seconda del valore passato come secondo argomento, estraiano, con funzioni di manipolazione delle stringhe, la data di nascita del

proprietario del C.F. oppure il suo sesso.

Usiamo la struttura di programmazione Selezione Caso, che è molto più comoda di una struttura Se... Altrimenti Se... Fine Se, in caso di più ramificazioni nella funzione. Con il «Caso Altrimenti», alternativo ai due possibili valori (1 e 2) del secondo argomento, intercettiamo gli errori.

Non intercettiamo invece gli errori sul primo argomento che deve essere un codice fiscale, che, come noto, contiene la data di nascita, dal settimo all'undicesimo carattere, anno, mese e giorno, con il mese indicato con una lettera. Il sesso del proprietario del C.F. è indicato dal fatto che il giorno del mese di nascita è aumentato di 40, se il tizio è... una tizia.

La seconda funzione, CNSN, lavora su una frase (è l'unico argomento da passare) dalla quale estrae, e le ripropone in maiuscolo, le sole consonanti. In questo caso, con un ciclo che scorre tutta la stringa in input, vengono testati, uno per uno, tutti i caratteri, per poi ricostruire la stringa, fatta di sole consonanti, in uscita.

Nell'ultima figura, la 10, vediamo un foglio in cui abbiamo usato le due nuove funzioni e vediamo come anche l'Autocomposizione delle funzioni tenga conto della presenza delle due nuove nate.

Una procedura più tradizionale

Affrontiamo ora un classico programma «didattico», che non ha nessuna finalità pratica, ma serve a sperimentare alcune istruzioni fondamentali di programmazione (fig. 11).

Lo scopo del programma è quello di contare da 1 fino a 72 (per motivi di videata) e di visualizzare solo i numeri divisibili per 7 e quelli che contengono, in prima o seconda posizione, un 7. Il risultato sarà quindi 7, 14, 17, 21, 27, 28...

Per scrivere la procedura occorre aprire un foglio Modulo e poi su questo foglio bianco scrivere:

```
Procedura CONTA()
  Se l'istruzione si scrive in minuscolo al momento dell'invio si può avere la conferma della correttezza sintattica dal fatto che VBA mette l'iniziale maiuscola alle istruzioni riconosciute. Se invece l'istruzione è sintatticamente errata VBA invia un messaggio chiarificatore dell'errore. Ogni Procedura, che inizia con l'istruzione Procedura, finisce con un'istruzione collegata, che è:
  Fine Procedura
```

Le istruzioni che abbiamo usato, e che potete vedere nella figura citata, sono innanzitutto il classico For... Next,

```

' *****
' Macro "MiniCalcolatrice"
' Attiva la finestra di dialogo, gestisce i pulsanti e restituisce
' sul foglio di lavoro il risultato del calcolo
' *****
Dim D1; D2; D3; R
' Definisce le variabili condivise da tutte le procedure nel modulo
' D1, D2, D3 sono i valori impostati dall'utente, R e' il risultato

Procedura MiniCalc()
  Azzera
  ' Richiama la procedura Azzera()
  FogliDialogo("Dialogo").CaselleDiModifica(4).Abilitato = Falso
  ' Inibisce le modifiche alla Casella di Modifica del risultato
  FogliDialogo("Dialogo").Mostra
  ' Attiva la Finestra di Dialogo nel foglio "DIALOGO"
  Fogli("Foglio").Celle(1; 1).Valore = Formato(R; "###.###.###,###")
  ' Esce dalla Finestra di Dialogo riportando nella cella A1 il risultato
  Fine Procedura

Procedura Slider()
  ' Gestisce la Barra di Scorrimento e imposta il valore
  ' risultante nella Casella di Modifica 3
  FogliDialogo("Dialogo").CaselleDiModifica(3).Testo =
  FogliDialogo("Dialogo").BarreDiScorrimento(1).Valore
  Fine Procedura

Procedura Somma()
  ' Opera la somma dei valori impostati dall'utente
  Trasforma
  R = D1 + D2 + D3
  FogliDialogo("Dialogo").CaselleDiModifica(4).Testo = R
  Fine Procedura

Procedura Sottrazione()
  Trasforma
  R = D1 - D2 - D3
  FogliDialogo("Dialogo").CaselleDiModifica(4).Testo = R
  Fine Procedura

Procedura Moltiplicazione()
  Trasforma
  R = D1 * D2 * D3
  FogliDialogo("Dialogo").CaselleDiModifica(4).Testo = R
  Fine Procedura

Procedura Divisione()
  Trasforma
  R = D1 / D2 / D3
  FogliDialogo("Dialogo").CaselleDiModifica(4).Testo = R
  Fine Procedura

Procedura Azzera()
  ' Imposta a zero il contenuto di tutte le Caselle di Modifica
  Per Conta = 1 FinoA 4
    FogliDialogo("Dialogo").CaselleDiModifica(Conta).Testo = 0
  Successivo
  Fine Procedura

Procedura Trasforma()
  ' Trasforma in numeri il contenuto testuale delle Caselle di Modifica
  D1 = Valore(FogliDialogo("Dialogo").CaselleDiModifica(1).Testo)
  D2 = Valore(FogliDialogo("Dialogo").CaselleDiModifica(2).Testo)
  D3 = Valore(FogliDialogo("Dialogo").CaselleDiModifica(3).Testo)
  Fine Procedura

```

che in VBA in italiano diventa:
Per Contatore = X FinoA Y Passo Z
Successivo

Al suo interno abbiamo usato una nomenclatura di Se, per testare se visualizzare o meno il numero corrente:

```
Se test1
AltrimentiSe test2
Fine Se
```

I test sono due. Per il primo, che è di tipo numerico, abbiamo usato l'operatore Mod, che restituisce il resto della divisione:

```
Se X Mod 7 = 0 Allora
  verifica che X diviso 7 sia uguale a zero
  e quindi che X sia multiplo di 7.
```

Poiché il secondo test è di tipo alfanumerico (presenza di un 7 nel numero convertito in stringa) abbiamo dovuto

usare due funzioni:

```
L=Str(I)
che converte il numero I in una stringa.
StrParte(L;x;y)
che estrae, dalla stringa L, y caratteri a
partire dall'x-esimo.
```

Nella Macro sono inserite anche le istruzioni che servono a riversare i vari numeri sul foglio:
CellaAttiva.FormulaR1C1=I
riversa nella cella attiva il valore corrente di I, se ha superato i vari test.
CellaAttiva.Scarto(x,y),

che serve per selezionare la cella in posizione relativa x,y rispetto alla corrente.

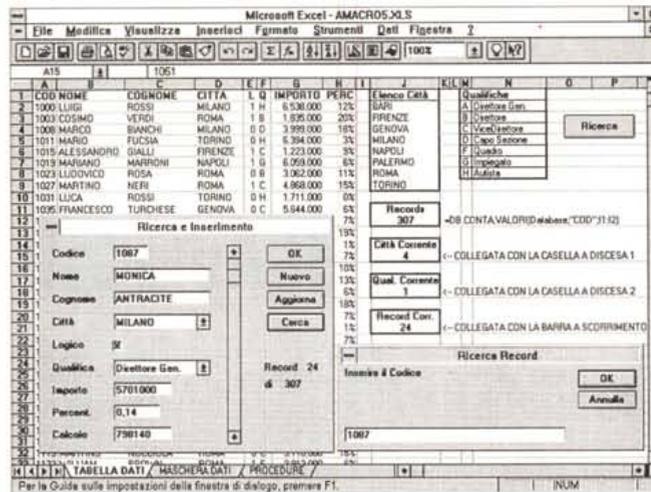
Terminata la macro, che non va battezzata in quanto è l'istruzione Procedura che si incarica di definirne il nome, ci si può spostare su un foglio normale per eseguirla.

Si tratta di una macro «relativa», che viene eseguita quindi a partire dalla cel-

Figura 15 - MS Excel 5.0 - Le procedure relative alla calcolatrice commentate.

Un foglio modulo può contenere anche più procedure, che possono essere commentate con delle spiegazioni che cominciano con un segno «apice». Una procedura può essere richiamata o perché associata ad un pulsante, o più genericamente perché associata ad un «evento». Può essere anche richiamata da un'altra procedura semplicemente indicandone il nome.

Figura 16 - MS Excel 5.0 - Applicazione gestione dati - I dati e la Dialog Box. Nel prossimo numero di MC continueremo le manovre di avvicinamento al VBA, presentando un'applicazione più complessa che fa uso di una Dialog Box più complessa: questa. L'obiettivo sarà non solo quello di sperimentare più a fondo le possibilità delle Dialog Box ma anche quello di riaffrontare il VBA, parlando in maniera più sistematica di oggetti, di proprietà degli oggetti, di metodi che intervengono sulle proprietà degli oggetti, ecc. ecc.



la su cui ci si è posizionati. Per eseguirla si può attivare il comando di menu: Strumenti Macro Esegui, oppure più elegantemente, si può piazzare un pulsante sul foglio e collegare a questo la macro, ormai riconosciuta. Lo vediamo in secondo piano sul foglio.

Abbiamo già usato un paio di volte le istruzioni per contare Per... Successivo.

Nel Basic tradizionale ci sono altre istruzioni comode per contare. Nella figura 12 vediamo l'esempio, un po' miserello, presente nell'Help del VBA, relativo all'istruzione Mentre... FineMentre. Come al solito gli esempi proposti nelle videata di Help sono copiabili direttamente nel foglio modulo.

La Dialog Box Calcolatrice

A differenza di quanto accadeva nelle precedenti versioni, Excel 5.0 permette all'utente di costruire eleganti e funzionali Dialog Box senza utilizzare utility esterne (quali il glorioso «Dialog Box Editor»): è infatti possibile inserire direttamente nella cartella di lavoro un tipo speciale di foglio, chiamato Finestra di Dialogo.

Questo foglio consiste in un vero e proprio ambiente operativo grafico, dotato di una fitta griglia, contenente una finestra nuda, dotata dei soli pulsanti «OK» e «ANNULLA», che costituisce il piano d'appoggio in cui inserire, muovere, cancellare, modificare gli oggetti alla Windows, disponibili nella Barra dei Pulsanti «Finestra di Dialogo» di cui abbiamo parlato prima.

Per sperimentare questa particolare funzionalità di Excel 5.0, realizziamo la piccola calcolatrice che vedete in figura 13. Si tratta, ovviamente, di una finestra di dialogo ad hoc, attivata dal pulsante «Calcolatrice», ed interamente gestita da una macro VBA. Il risultato del calcolo eseguito con la calcolatrice viene poi riportato nella cella A1 del foglio attivo.

È evidente l'inutilità pratica di macro del genere dato che qualsiasi calcolo

può essere eseguito direttamente nella cella di Excel.

Il primo passo da compiere è la costruzione della finestra e degli elementi che la compongono (fig. 14). Quindi, attivato il foglio finestra di dialogo nella cartella, provvediamo all'inserimento delle 4 caselle di modifica che conterranno gli input numerici e il risultato delle operazioni. Questi oggetti non necessitano di ulteriori settaggi poiché saranno interrogati e gestiti dalla macro.

Disegnati i quattro Pulsanti, un doppio clic al centro di ciascuno di essi consente di editarne i nomi e sostituirli con «+», «-», «*», «/». In seguito abbineremo ad ognuno di essi una procedura in VBA attivata dall'evento «pulsante premuto».

Tracciamo, quale puro estetismo, la Casella di Gruppo che circonda i pulsanti operatori aritmetici: un doppio clic sul bordo ci consente di digitarne il titolo. Altro estetismo è la figura decorativa posta sulla calcolatrice.

Eliminiamo il pulsante «OK» e, al suo posto, inseriamo un pulsante «AZZERARE». Un doppio clic sul pulsante «Annulla» per trasformarlo in «ESCI».

L'ultimo oggetto che inseriamo è la Barra di Scorrimento che tracciamo orizzontalmente: servirà ad impostare (incrementando o decrementando) il valore contenuto nella terza Casella di Modifica.

Doppio clic sulla barra del titolo della finestra di dialogo e inseriamo il testo «MINICALCOLATRICE».

Tocco finale: cambiamo nome al foglio (Formato / Foglio / Rinomina) da «Dialogo 1» in, più semplicemente, «DIALOGO».

Per avere un'anteprima di come la finestra apparirà agli occhi dell'utente, facciamo clic sull'ultimo pulsante della barra Finestra di Dialogo, quello a forma di interruttore.

Il listato che si trova in figura 15 è il codice VBA che serve a gestire tutti gli eventi pilotati dalla Finestra di Dialogo. I

commenti dovrebbero essere sufficienti ad interpretare il flusso delle istruzioni. Ci limitiamo pertanto ad analizzarne i punti più interessanti.

L'indicazione FogliDialogo («Dialogo») è il riferimento alla Dialog Box costruita nel foglio chiamato «Dialogo»: si noti come gli oggetti Caselle di Modifica e Barre di Scorrimento vengano identificati da questo prefisso.

La presenza delle quattro procedure Somma(), Sottrazione(), Moltiplicazione() e Divisione() nasce dall'esigenza di dover abbinare un controllo ad ognuno dei quattro pulsanti «operatori aritmetici». Torniamo quindi al foglio Finestra di Dialogo per impostare questi abbinamenti.

Iniziamo dalla somma; un clic con il tasto destro sul pulsante permette di attivare l'opzione Assegna Macro. Nell'elenco di procedure da noi impostate scegliamo quella denominata «Somma» e ripetiamo la sequenza per gli altri tre pulsanti.

Alla Barra di Scorrimento abbiniamo invece la procedura Slider(), il cui unico compito è di riportare nella terza Casella di Modifica il valore impostato scorrendo la barra.

Il pulsante «Azzerare», viene abbinato alla procedura omonima che ha il compito di impostare a zero tutte le caselle di modifica.

Cosa vedremo nel prossimo articolo

Nel prossimo numero di MC continueremo le manovre di avvicinamento al VBA presentando un'applicazione più complessa che fa uso di una Dialog Box più complessa, che possiamo per ora solo vedere in figura 16.

Un primo obiettivo sarà quindi quello di sperimentare più a fondo le possibilità delle Dialog Box, come interfaccia che facilita il rapporto tra utente e dati del foglio. Come già si può intuire dalla figura cercheremo di utilizzare il più possibile l'oggettistica Windows, dotando la Box di Caselle a Discesa, di Pulsanti, ecc. che collegheremo alle celle del foglio.

Per quanto riguarda VBA in sé... ricominceremo la trattazione da capo, nel senso che riaffronteremo, in maniera più sistematica e schematica, gli elementi tipici di tale ambiente, e quindi parleremo di oggetti, di proprietà degli oggetti, di metodi che intervengono sulle proprietà degli oggetti, di istruzioni speciali per identificare gli oggetti, ecc. ecc. Vi consigliamo di non mancare.