

# Corso Rapido di Visual Basic destinato a chi non sa programmare

*Il minicorso di Visual Basic si sviluppa in tre puntate.*

*Nella prima, pubblicata sul numero scorso di MC, abbiamo motivato il corso stesso, che, come risulta dai titoli, è destinato anche a chi non sa programmare, ed ha quindi un «taglio» molto semplificato.*

*Abbiamo trattato i primi rudimenti ed in particolare abbiamo esaminato quasi tutti i Control presenti nella Tool di Visual Basic.*

*Nella seconda puntata, questa, parliamo 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 esterno contenente dati, l'uso delle Subroutine, l'uso delle Funzioni Personalizzate, la gestione delle Variabili e delle Matrici.*

*Nella terza puntata 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**

seconda puntata

## Le funzioni di I/O

Il Basic, di qualsiasi casa e di qualsiasi versione, non è stato mai molto adatto per lo sviluppo di applicazioni di tipo DBMS. Possiede infatti poche e rudimentali istruzioni di Input/Output, che consentono solo poche operazioni fondamentali, ben limitate in confronto a quelle possibili in un moderno prodotto DBMS, anche sotto Windows.

Se l'applicazione da sviluppare è

orientata alla gestione, magari pesante, di un database complesso (molti archivi collegati), sarebbe quindi meglio non utilizzare il Visual Basic.

In alcuni casi, quelli in cui il Visual Basic consente soluzioni (ad esempio la gestione della Grafica) non raggiungibili con gli altri prodotti, si può pensare di utilizzare comunque il Visual Basic, tirandogli «il collo» sulle funzioni di Input ed Output, per arrivare a simularne un uso simile a quello consentito da un

DBMS.

Infatti, come abbiamo detto, Visual Basic possiede comunque i comandi fondamentali, cui l'utente più esperto può affiancare funzioni complementari, o scritte da lui stesso, o prelevate da librerie di mercato, specializzate nella gestione dell'Input/Output.

Noi limiteremo, per lo meno in questo minicorso, il nostro interesse alle sole funzioni interne, quelle che permettono di gestire file Sequenziali e quelle che permettono di gestire file Random. Su queste faremo una serie di esercizi che permettano al nostro lettore alle prime armi, di approfondire questo importante aspetto del Visual Basic.

## Tipi di operazioni consentite e tipi di operazioni non consentite

Il Basic in generale, e quindi anche il Visual Basic in particolare, consente due tipi di accesso ai file Dati. L'accesso Sequenziale, con il quale si leggono (o si scrivono) uno dopo l'altro e con una unica operazione tutti, o comunque un insieme, di Record dell'Archivio, e l'accesso Casuale (Random), con il quale si legge (o si scrive) un solo record per volta, identificato dal suo «indirizzo fisico», che però occorre conoscere.

Il concetto di Sequenziale e di Casua-



Figura 1 — MS Visual Basic 2.0 — Ambiente.

*Nella prima puntata del nostro minicorso abbiamo studiato tutti gli strumenti presenti nel Toolbox di Visual Basic. Abbiamo constatato come l'ambiente Visual Basic assomigli più a quello di un prodotto per il disegno che non a quello, meno colorato e grafico, di un normale linguaggio di programmazione. In questa puntata ci occupiamo di comandi e funzioni interne, che non provengono dal Toolbox.*



le è un concetto che si ritrova in vari campi. Se vogliamo raggiungere un dato e ci troviamo in una organizzazione Sequenziale, dobbiamo comunque leggere (anche se cerchiamo di farlo velocemente) tutti i dati precedenti. In una organizzazione Casuale possiamo saltare direttamente al dato voluto, conoscendone l'indirizzo. Pensate alla differenza tra una musicassetta e un CD.

Sperimenteremo quindi una serie di operazioni «consentite» sui due tipi di file.

Poi costruiremo alcune piccole applicazioni allo scopo di eseguire anche operazioni «non consentite». Con questo termine indichiamo quelle operazioni che in altri prodotti, più orientati al trattamento dei dati o alla gestione dei dati, si ottengono con un solo comando.

Realizzeremo quindi un Programma per l'Ordinamento dei dati e un programma per la costruzione e gestione di un File Indice, con il quale velocizzare la ricerca dei dati e il loro ordinamento.

Alla fine parleremo di Subroutine e di Funzioni Utente. Argomento che non c'entra nulla con le funzioni di I/O, ma che è altrettanto fondamentale.

Si tratta di due argomenti «sotterranei».

Sono ambedue importanti, ma non hanno nessun rapporto diretto con quello che succede in superficie, quindi nel Form.

STRUTTURA		
Campo	Lunghezza	Note
MATRICOLA	4 caratteri	solo numerici
NOME	15 caratteri	
COGNOME	15 caratteri	
CITTA	15 caratteri	
LOGICO	1 carattere	S/N oppure 1/0 Check Box
DATA	numero lungo	data in numero seriale
QUALIFICA	1 carattere	List Box per Qualifica
IMPORTO	numero doppio	
SCONTO	numero intero	

DATI								
"1022"	"LUIGI"	"ROSSI"	"ROMA"	"S"	25631	"H"	6538000	12
"1027"	"COSIMO"	"VERDI"	"MILANO"	"S"	20783	"B"	1835000	20
"1029"	"MARCO"	"BIANCHI"	"TORINO"	"N"	26775	"D"	3999000	18
"1034"	"MARIO"	"FUCSIA"	"FIRENZE"	"N"	23386	"H"	6394000	3
"1038"	"ALESSANDRO"	"GIALLI"	"NAPOLI"	"S"	26197	"C"	1223000	9
"1044"	"MARIANO"	"MARRONI"	"ROMA"	"S"	20816	"G"	6059000	6
"1047"	"LUDOVICO"	"ROSA"	"ROMA"	"N"	19513	"B"	3062000	11
"1051"	"MARTINO"	"NERI"	"TORINO"	"S"	27332	"C"	4868000	15

Figura 2 — MS Visual Basic 2.0 — Il nostro archivio commentato. Nell'articolo parliamo di funzioni di Input/Output. Lavoreremo con un solo archivio, che tratteremo sia in maniera «Sequenziale», e in questo caso è possibile anche vedere, da DOS, il contenuto dell'archivio stesso, sia in maniera «Random». In questo secondo caso l'archivio non è visibile «ad occhio nudo» in quanto i dati numerici vengono memorizzati in formato binario. Nella figura vediamo sia la struttura dell'archivio che usiamo per la prova sia il contenuto, in chiaro, dei primi record.

Il primo esercizio che faremo utilizzerà un solo Form e addirittura nessun Control.

### Premessa

Poiché pensiamo che un argomento di tale importanza debba essere studiato con calma ci siamo dati dei limiti di trattazione ben precisi:

- eseguiremo quasi esclusivamente operazioni di Lettura,
- faremo esercizi che lavorano su un solo File Dati, già predisposto e ben conosciuto come struttura,
- anche il contenuto del File Dati ci è perfettamente noto,
- non approfondiremo il problema del-

le Prestazioni dell'Applicazione.

Queste limitazioni ce le poniamo per poter fare, nel ristretto spazio di un articolo, comunque una trattazione concreta con esempi svilupparli anche dal neofita.

Prima di cominciare, quindi, analizziamo il nostro archivio e studiamo la sintassi delle poche, per fortuna, istruzioni che ci serviranno nei successivi esercizi.

In figura 2 vediamo il nostro archivio, in alto la sua struttura con nome, lunghezza e tipo dei campi, in basso i dati veri e propri in un formato che è il cosiddetto «comma delimited». Questo formato è quello utilizzato nei file Sequenziali del Visual Basic, ma è ottenibi-

Figura 3 — MS Visual Basic 2.0 — Funzioni di Input e Output. Per funzioni di Input e Output intendiamo quelle che permettono di leggere e di scrivere dei file dati esterni, anche di grandi dimensioni. I dati letti potranno essere manipolati e visualizzati nei Form Visual Basic così come, in senso inverso, i dati digitati o calcolati nei Form potranno essere registrati nel file esterno.

#### Tipi di Variabili

Tipo	Simbolo	Bytes	Da	A
Integer	%	2	-32.768	32.767
Long	4	-2.147.483.648	2.147.483.647	
Single	!	4	-3,37 E+38	3,37 E+38
Double	#	8	-1,67 D+308	1,67 D+308
Currency	@	8	-9,22 E+14	9,22 E+14
String	\$	max.32K		

#### Letture di 5 record in Sequenza

```
OPEN "DATIVB.DAT" FOR INPUT AS # 1
FOR I=1 TO 5
INPUT #1,A$(I),B$(I),C(I)
NEXT I
CLOSE #1
```

#### Scrittura di 5 record in Sequenza

```
OPEN "DATIVB.DAT" FOR OUTPUT AS # 1
FOR I=1 TO 5
PRINT #1,A$(I),B$(I),C(I)
NEXT I
CLOSE #1
```

#### Accodamento di 5 record in Sequenza

```
OPEN "DATIVB.DAT" FOR APPEND AS # 1
FOR I=1 TO 5
PRINT #1,A$(I),B$(I),C(I)
NEXT I
CLOSE #1
```

#### Letture di un Record in 27 posizione

```
NR=27
VAR=. . . Variabile che contiene l'intero Record
OPEN "DATIVB.SEQ" FOR RANDOM LEN=75 AS # 1
GET #1,NR,VAR
CLOSE #1
```

#### Scrittura di un Record in 27 posizione

```
NR=27
VAR=. . . Variabile che contiene l'intero Record
OPEN "DATIVB.SEQ" FOR RANDOM LEN=75 AS # 1
PUT #1,NR,VAR
CLOSE #1
```



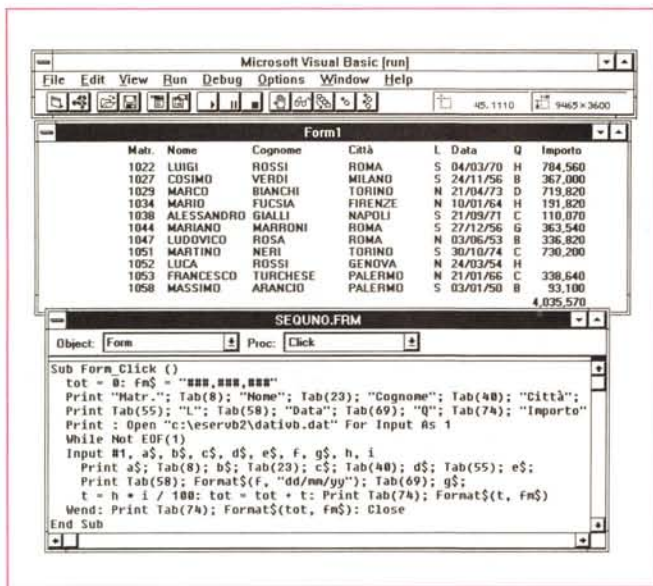


Figura 4 — MS Visual Basic 2.0 — Lettura del File e sua Visualizzazione nel Form. Il primo esercizio consiste nella semplice lettura del File (serve infatti proprio per studiare i comandi di lettura), nella contemporanea elaborazione di alcuni campi, nella totalizzazione dei campi numerici e nella loro visualizzazione direttamente sul Form. In questo caso non abbiamo disegnato oggetti. Tutto il processo avviene quando si fa Click sul Form stesso.

le direttamente da molti prodotti, ad esempio dal dBase III.

È evidente che i Record (ovvero le righe) nei file Sequenziali non hanno lunghezza fissa.

Nei file Random invece va dichiarata la lunghezza di ciascun campo, e la lunghezza, ovvero la lunghezza totale, del Record. Ed è proprio grazie alla lunghezza nota, uguale per tutti i record, che nel file Random può essere eseguito un «accesso diretto» al Record di interesse.

In altre parole la lunghezza dei campi è necessaria solo per i file Random, mentre quelli Sequenziali accettano qualsiasi lunghezza di campo, tanto il campo stesso è identificato dalle «virgolette», nel caso di campi alfabetici, e dalle virgole, in caso di dati numerici.

Nei file Sequenziali i dati vengono comunque lasciati in formato ASCII, invece nei file Random debbono venir convertiti in un formato binario (questo riguarda solo i numeri).

Ad un numero definito Intero vengono riservati 2 byte, ovvero 2 alla 16-ma, quindi da -32.768 a 32.767.

Se non bastano 2 si usano 4 byte e si ha un numero Long, se ancora non basta, si ricorre al Currency, che può andare bene, data la sua precisione, anche per il Bilancio dello Stato.

Altri tipi numerici sono quelli definiti «floating point», che variano maggiormente, a scapito però della precisione. Servono per scopi scientifici.

Tornando al nostro archivio c'è da notare la presenza, nella struttura dell'Archivio, di una Data. La Data può

essere espressa in formato alfanumerico «11/12/92» oppure in formato numerico seriale, secondo il «calendario» che usa Visual Basic, che è lo stesso che usano l'Excel e gli altri fogli elettronici, e che si basa sui giorni a partire dal 1 gennaio 1900.

Nel nostro archivio l'abbiamo lasciata in questo formato seriale, anche perché può essere trasformata in qualsiasi formato Data e può comunque essere manipolata con le funzioni che agiscono sulle date.

Chi dispone di un File dBase può facilmente, sia con il dBase III che con i dBase IV, generare un file «comma delimited», ma attenzione alle Date che vengono trasformate in un altro modo, diventano 19921215, un formato numerico su cui valgono anche gli operatori relazionali (maggiore, minore, ecc.).

## Le poche istruzioni di I/O

Sempre nella figura 3 abbiamo inserito, in alto, i tipi di Campo e, in basso, le istruzioni Visual Basic che utilizzeremo negli esercizi, dividendole in «blocchetti» di poche righe.

I blocchetti cominciano con il comando Open (leggetene attentamente la sintassi esatta nell'Help, o nel manuale), che accetta alcune varianti, per specificare il modo di lettura e il numero di file aperto.

Tale numero diventa indispensabile quando si lavora su più file aperti in contemporanea.

Tutti i blocchi si chiudono con una istruzione Close.

Nel caso del Sequenziale manipoliamo un file di 5 record, prelevati o scritti da o in alcuni vettori. Nel caso dei file Random, invece, leggiamo o scriviamo solo il record numero 27.

Le istruzioni di lettura e scrittura sono INPUT, PRINT, nel caso del Sequenziale, e GET e PUT, nel caso del Random.

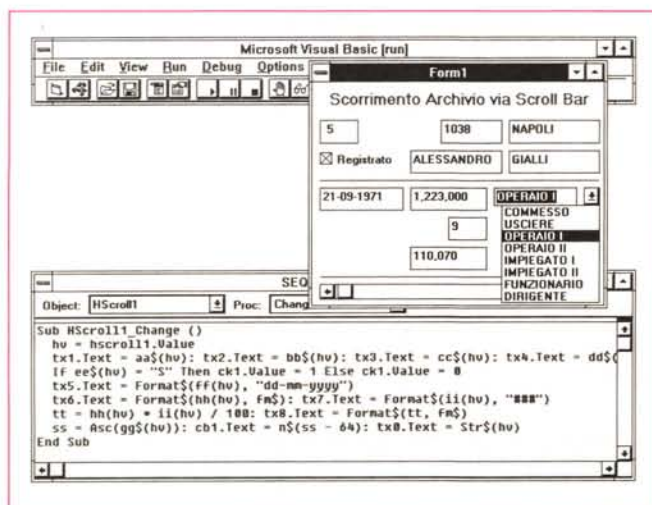
Il file Sequenziale può essere letto e scritto (e accodato) solo in sequenza. Questo va ribadito perché deve essere un meccanismo chiaro a tutti. Infatti se si esegue due volte il secondo blocchetto di istruzioni viene scritto due volte lo stesso file (la seconda sovrascrive). Se invece si esegue prima un Output e poi un Append si ottiene la somma dei record.

Se si vuole leggere solo il 27-mo record si deve scorrere prima i 26 precedenti, in quanto non si sa a priori la posizione fisica, all'interno del file, del Record cercato.

Nel file Random il 27-mo record viene indicato direttamente nella istruzione di Lettura o Scrittura.

È chiaro che si utilizzeranno file se-

Figura 5 — MS Visual Basic 2.0 — Caricamento dei Dati in Array e loro scorrimento con una Scroll Bar. Nell'esercizio precedente non abbiamo utilizzato nessun Control. Ora invece prevediamo di leggere il nostro file dati, di appoggiarlo su alcune Array (cosa possibile e conveniente solo se i file sono di piccole dimensioni) e di visualizzarlo record per record, facendo apparire i dati in un Form con alcune Text Box e altri Control. Il movimento, avanti e indietro sull'archivio, è realizzato con una semplice Scroll Bar.





quenziali solo in caso di archivi piccoli, da caricare e da manipolare tutti insieme.

I file Random sono invece più «nobili», possono servire per applicazioni più impegnative. Il problema diventa quello di conoscere l'indirizzo di ciascun Record in modo da rendere produttivo il sistema di indirizzamento proprio del file Random.

Stiamo per parlare proprio di questo.

## Avvertenze

Il Visual Basic è molto divertente da usare, ma è difficile documentare il programma che è sempre scomposto in tanti pezzi, legati a ciascun evento su ciascun oggetto.

Dei nostri esercizi mostreremo solo gli aspetti più significativi, dando per scontato che ai vari pezzi che mancano ci penserete voi.

Inoltre diamo per scontata la conoscenza di quelle informazioni, legate all'uso dei Control del Toolbox, che dovrete aver acquisito leggendo il primo articolo.

Ricordiamo che gli esercizi proposti, tranne quello che usa la Griglia, possono essere sviluppati indifferentemente con Visual Basic 1 oppure con il 2.

Precisiamo anche il fatto che per poter scrivere dei listati corti (che rientrassero in una videata) abbiamo sempre dato dei nomi corti ai vari oggetti. Ad esempio l'oggetto Text1 è diventato TX1, Label10 LB10, Combo1 CB1, e così via.

Precisiamo infine che faremo una descrizione molto sommaria dei vari esercizi. Vi suggeriamo se necessario di approfittare dell'ottimo Help del Visual Basic, richiamabile anche per parola chiave.

## Sequenza

Il primo esercizio consiste nella lettura del file Sequenziale. Occorre aprire il file (OPEN #1), e leggerlo (INPUT #1, var1, var2, ...) record per record. Alla fine il file va chiuso (CLOSE #1).

Viene utilizzato un solo Form e l'unico evento gestito è il Click sul Form stesso che causa la Lettura dei Record, l'esecuzione di qualche Calcolo e la visualizzazione dei Dati così manipolati sul Form stesso.

Lo scorrimento dei record viene gestito con una coppia di istruzioni che generano un ciclo:

```
WHILE NOT EOF(1)
```

```
.omissis.
```

```
WEND
```

il ciclo viene ripetuto fino a quando è Vero che non è finito il File. Dovrebbe

Figura 6 — MS Visual Basic 2.0 — Ordinamento.

Le funzioni di Input e Output del Visual Basic sono «rudimentali», non sono quindi paragonabili a quelle presenti in un comune prodotto DBMS. A volte è quindi necessario scrivere delle routine per eseguire delle operazioni che in un DBMS sono eseguite direttamente dal comando. Ad esempio l'ordinamento, che nel dBASE III ha un suo comando specifico, deve essere risolto, dal programmatore VB, necessariamente con un sottoprogramma.

risultare chiaro anche al novellino che tale sistema serve per leggere tutto il file anche se non se ne conosce a priori la lunghezza, in termini di numero di record contenuti.

Alcuni dei dati letti vengono formattati, altri subiscono dei calcoli, poi tutti insieme vengono visualizzati con il co-

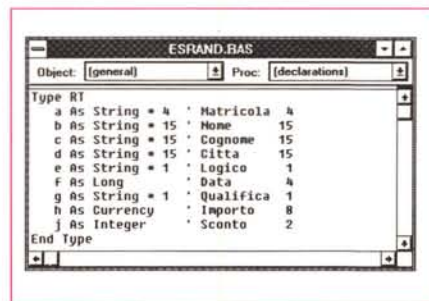
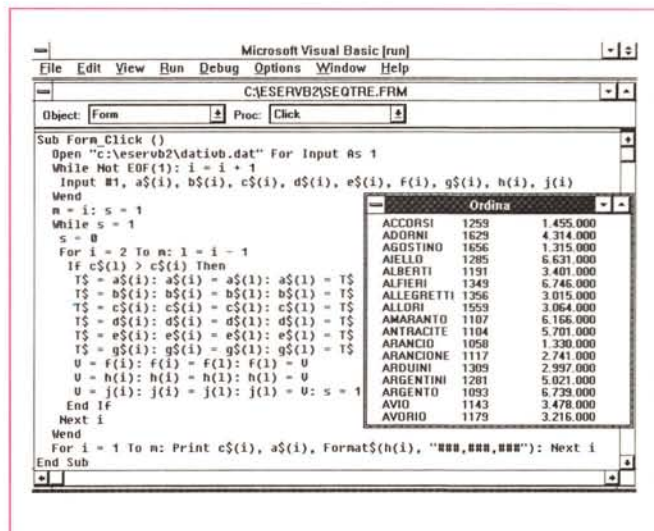


Figure 7, 8 — MS Visual Basic 2.0 — Randomizzazione del file sequenziale.

Compito di questo programma è quello di leggere il File con i Dati (che è in un formato ASCII delimitato) e di trasformarlo (con una procedura che prima legge e poi scrive) in un file Random, che è un formato codificato e non leggibile se non con un programma VB di lettura. Questi tipi di file permettono un accesso diretto al Record di cui si conosca l'indirizzo.



mando PRINT. Durante lo scorrimento viene anche gestita la totalizzazione degli Importi.

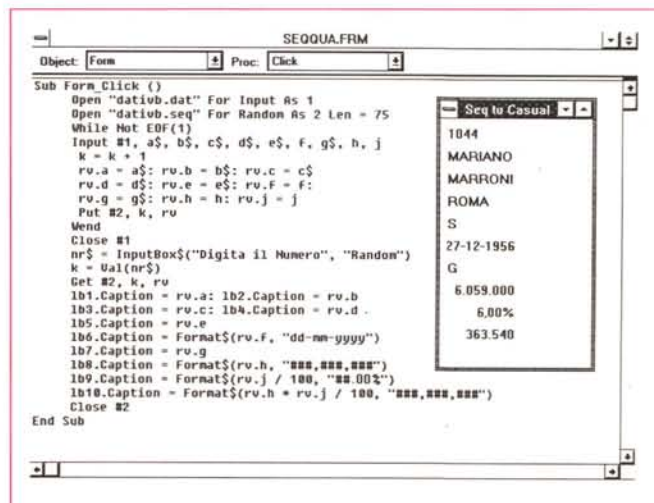
Insomma vengono eseguite una serie di operazioni normali in questi casi, che non stiamo, per mancanza di spazio e perché il programma è comunque visibile sulla figura 4, a descrivere nel dettaglio. Nella figura si vede anche il Form con i Dati visualizzati.

## Scorrimento con Scroll Bar

Nel primo esercizio abbiamo letto i dati e li abbiamo via via manipolati. Ora li leggiamo e li appoggiamo su una serie di vettori, per poterli più facilmente vedere.

Li vediamo in un Form, in cui abbiamo messo alcune Text.Box.

Nel Form ci sono anche un Check Button, che scodifica il campo Logico,





una Combo Box, che invece usiamo per scodificare gli Otto codici della Qualifica (per svolgere questo compito usiamo un vettore n\$(x) che li contiene), una Scroll Bar, che usiamo per scorrere avanti e indietro i Record.

La figura 5 mostra solo il Form con i dati e il programma relativo all'evento «Change» subito dalla Scroll Bar.

Mancano i «pezzi» relativi alla lettura dei record e al loro trasferimento nei vettori che dovete realizzare da soli. Si tratta di operazioni viste nella lezione precedente.

## Ordinamento

Le funzioni di gestione dei file dati del Visual Basic sono «rudimentali», non sono quindi assolutamente paragonabili a quelle presenti in un comune prodotto DBMS.

A volte è addirittura necessario scrivere delle routine per eseguire delle operazioni che in un DBMS sono eseguite direttamente dal comando. Ad esempio l'ordinamento, che nel dBASE III ha un suo comando specifico, deve essere risolto, dal programmatore VB, necessariamente con un sottoprogramma, che sfrutti una delle routine di ordinamento che sono presenti a decine nei libri di programmazione, sia nei Testi Sacri, sia nei manuali dedicati ai vari prodotti.

Per Ordinare il nostro file utilizzeremo la routine più semplice, che è però anche la meno veloce (in un programma reale bisogna invece scegliere la routine più veloce), anche perché è citata dall'Help del Visual Basic, alla voce WHILE WEND, che ha come esempio proprio la routine di Sort.

Nell'esercizio, in figura 6, vediamo sia la fase di lettura che la fase di ordina-

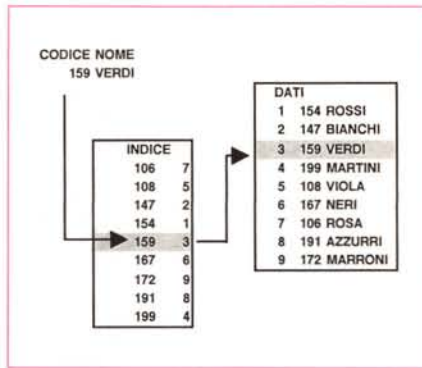


Figura 9 — MS Visual Basic 2.0 — Costruzione di un Indice.

Altro «pilastro», concettuale ed operativo, nei prodotti DBMS è l'Indice, che serve a vedere, nell'ordine voluto, e quindi indipendentemente dal loro ordine fisico, i dati dell'archivio.

L'indice serve sia per ordinare, senza ricorrere all'Ordinamento Fisico, i dati, sia per cercarli più velocemente con una tecnica di ricerca «a salti». In Visual Basic l'indice non c'è, va quindi in qualche modo simulato. Vi suggeriamo un modo molto semplice.

mento vero e proprio che viene eseguito confrontando coppie di Cognomi (che risiedono nel vettore C\$(i)).

Se il Cognome posizionato prima viene dopo in ordine alfabetico i due (e conseguentemente tutti gli altri campi) si scambiano di posto. Le istruzioni, è un classico, che eseguono lo scambio sono:

```
C$(i)=T$: C$(l)=C$(l): C$(l)=T$
```

Maggiori dettagli ed esempi sugli ordinamenti si trovano anche in alcuni dei libri, già usciti in Italia, sul Visual Basic.

## Random

Per manipolare un file Random occorre definire, in un file \*.BAS da associare al programma, una speciale variabile, da noi chiamata RT, alla quale vengono associati direttamente tutti i campi presenti nella struttura del file.

Questa variabile semplifica sia la ge-

stione delle variabili, sia le istruzioni di scrittura e di lettura del file.

L'istruzione con la quale si definisce la variabile è:

```
TYPE RT
  A As String * 4
```

```
  F As Long
```

```
END TYPE
```

la vediamo in figura 7 applicata al nostro archivio.

Poi nella sezione Global del Form in cui si vedrà il file, si può definire un'altra variabile di appoggio di identica struttura con l'istruzione:

```
DIM RT as RV
```

Da ora in poi RT.A indicherà il campo A del record su cui ci si è posizionati. Ad esempio si potrà scrivere:

```
Text1.text = RT.A
```

per visualizzare la variabile RT.A in una Text1.box. E viceversa

```
RT.A = Text1.text
```

per trasferire la stringa digitata nella Text.box nella sua variabile.

L'esercizio che vi proponiamo (fig. 8) fa tre cose.

Legge il file Sequenziale. Contemporaneamente, record per record, alimenta il file Random, in modo che alla fine abbiamo due file di contenuto uguale, uno Sequenziale e uno Random.

Il secondo file viene lasciato aperto in modo da poter richiedere un record voluto.

Per richiedere il record viene usata l'istruzione:

```
X$ = InputBox$(..)
```

che esegue una mini Dialog Box che fa assumere alla variabile X\$ il valore digitato nella Dialog Box stessa. Tale valore viene subito tradotto in numero e usato per richiamare il record voluto, usando finalmente l'istruzione:

```
GET #1,K,RV
```

in cui K è il numero di record voluto e RV è la variabile di appoggio in cui transitano tutti i dati, che abbiamo citato prima.

## Indice

Altro «pilastro», concettuale ed operativo, nei prodotti DBMS è l'Indice. Concretamente l'Indice è un file, di appoggio rispetto al file con i dati, in cui sono memorizzate due cose, la chiave dell'Indice e il corrispondente numero del Record nel file dei Dati. Tale file d'appoggio deve venir tenuto ordinato per chiave, in modo che, data una chiave, sia immediatamente rintracciabile il relativo Record nell'archivio Dati.

In figura 9 schematizziamo la funzione del File Indice. La Chiave è la Matricola, viene trovata nel file Indice, e viene poi trovato il record cui corrisponde.

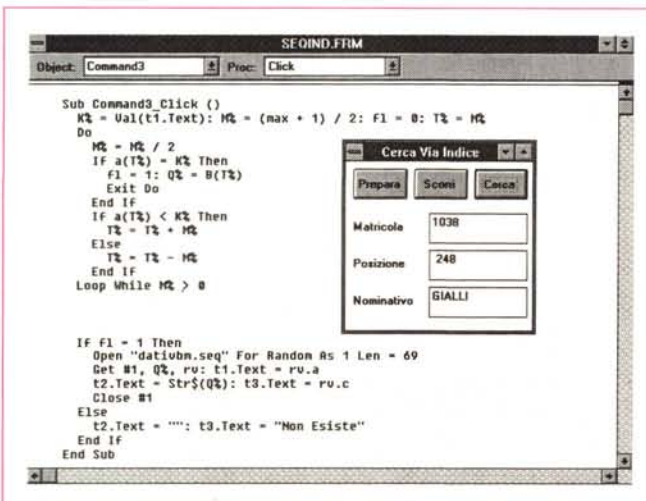


Figura 10 — MS Visual Basic 2.0 — Ricerca conoscendo l'Indirizzo.

Un file Random è utile e facilmente maneggiabile quando si conoscano gli indirizzi dei Record che si vogliono leggere. Si fa in questo caso un puntamento immediato (come in questo esercizio).



Nei prodotti di tipo DBMS esistono varie modalità per creare, gestire e usare i File Indice. Nel Basic invece gli Indici non esistono. Occorre «arrangiarli» o realizzandoli in proprio o utilizzando programmi aggiuntivi che servono a lavorare meglio con i file.

I File Indice sono importantissimi sia per mettere facilmente in ordine i dati, senza alterarne il reale ordine fisico, sia per cercare rapidamente i Record per mezzo della chiave, in quanto, essendo il file Indice ordinato, si può utilizzare un metodo di ricerca rapido, analogo a quello che usiamo noi «umani», per cercare un numero di telefono nell'elenco.

Anche la ricerca in un elenco ordinato è un argomento classico nei libri di informatica.

Nel nostro esercizio supponiamo di aver un archivio «disordinato», in cui cioè i vari Record non siano ordinati neanche per codice (al contrario di quello che abbiamo visto nella figura 2). Poi, con un programmino di lettura e scrittura, costruiamo un secondo archivio che contenga due campi, la Matricola (è questa la nostra chiave di ricerca) e il numero del record corrispondente, che in questo momento è un numero progressivo.

Terminata questa operazione ordiniamo il file secondo la chiave, realizzando in tal modo il nostro Indice, in cui la Matricola è in ordine e il numero del record.

Nella figura 10 vediamo l'utilizzazione dell'Indice.

Scriviamo la chiave da cercare nella prima Text.box e facciamo click su Cerca.

La chiave, cioè la Matricola, viene cercata nel file Indice, se c'è, viene prelevato il corrispondente indirizzo e, tramite questo, pescato nel File Dati il Record. Se non c'è, nella Text.box «Nominativo» scriviamo il messaggio «Non Esiste».

L'uso di un Indice comporta la necessità di avere i due file sempre perfettamente allineati.

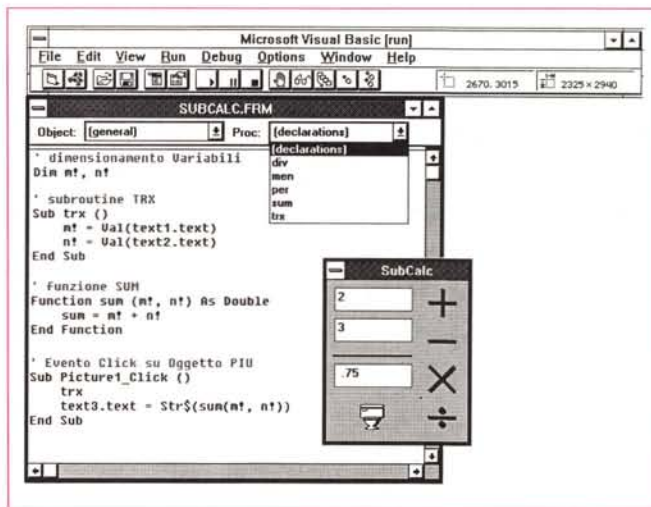
Altro problema può essere costituito dal fatto che la chiave può essere o meno ripetuta.

Una Matricola non può esserlo, mentre ad esempio un Cognome sì. In tal caso il programma di ricerca deve prevedere evenienza.

Per eseguire velocemente la ricerca della chiave nel file indice abbiamo usato un sistema «binario» che punta a metà del file, e poi confronta la chiave cercata con quella presente a metà. Poiché l'archivio è ordinato si continua la ricerca a metà della prima metà, se la chiave cercata è inferiore a quella

Figura 11 — Visual Basic 2.0.

*Ci siamo un po' dilungati sulle funzioni di // O. Quindi dedicheremo un solo esercizio al secondo argomento, che in realtà è molto importante nella programmazione «seria». Parleremo delle Function e delle Subroutine, che sono due strumenti che servono per ottimizzare il programma, mettendo in comune i processi di calcolo ripetitivi. Esemplicheremo tali concetti realizzando la «solita» calcolatrice.*



trovata a metà. Viceversa si punta alla metà della seconda metà e così via. In tal modo si raggiunge molto velocemente (si fanno pochi confronti anche se i Record sono migliaia) il record voluto.

### Ordinamento e griglia

Uno dei pochi Control che abbiamo «saltato» nell'articolo del numero scorso è il Grid.

È possibile definire una sorta di minispreadsheet su cui agiscono un certo numero di Proprietà che ne permettono una completa gestione. Si possono aggiungere righe e colonne, si possono definire larghezza delle colonne e estetica delle celle.

Riguardo al contenuto, ogni cella della griglia può essere considerata una Textbox, nella quale è possibile scrivere e della quale è possibile leggere il contenuto.

L'esercizio proposto comporta la lettura dei File Dati, il suo ordinamento per città. Poi con una routine di scorrimento si totalizzano sia gli importi per città (approfittando del fatto che le città sono in sequenza), sia quelli generali.

Alla fine si utilizzano le celle con i totali per città per calcolare le relative percentuali. Il tutto viene riversato nella griglia.

### Subroutine e Function

Alla fine dell'articolo cambiamo radicalmente argomento, parleremo della possibilità di definire e di usare delle Subroutine, pezzi di programmi richiamabili da varie parti, e delle Function, ovvero dei calcoli ripetitivi che assumono la struttura tipica delle Funzioni.

Come «Caso Studio» useremo la so-

lita Calcolatrice semplificata (fig. 11).

Due Textbox, per inserire i due valori, una per ricevere il risultato e cinque bottoni Grafici (delle Icone prelevate dalla vasta libreria in dotazione) che servono per eseguire le quattro operazioni e per pulire la videata (scusate il disegno poco elegante).

Nel listato che appare nella figura abbiamo inserito, facendo un «Collage», alcuni «brani» di programmazione.

— Il dimensionamento delle variabili (nella sezione General del Form),

— la subroutine TRX che traduce il contenuto delle due Textbox nei corrispondenti valori numerici,

— la funzione SUM, in rappresentanza delle altre tre. Alla funzione vengono passati due parametri (i due addendi) e restituisce un parametro (la somma appena calcolata),

— il programma eseguito al verificarsi dell'evento Click sul bottone PIU. Tale programma esegue la subroutine TRX, di traduzione delle variabili, e usa la funzione SUM. Poi traduce in stringa il risultato e lo piazza nella terza Textbox.

E così in tale esercizio abbiamo visto in azione sia delle Subroutine che delle Function.

### Conclusioni

Nel prossimo numero vedremo questo e una serie di esercizi più complessi, che però sfrutteranno solo gli elementi visti nelle prime due puntate. È probabile che prepareremo un'ulteriore lezione che affronterà i rapporti tra Visual Basic e Windows, nella quale sperimenteremo DDE, OLE, il comando Shell, ecc.

A presto.



# E.T.S. L'ALTERNATIVA VINCENTE

## 486-50 MHZ L. 2.580.000

Intel 486-50 Mhz processor.  
256K Static Cache RAM.  
4MB High Speed RAM.  
AMI Bios con Shadow RAM Support.  
1.44MB Floppy Drive.  
100MB 14ms IDE Hard Disk.  
Svga Windows Accelerator (16 mil. colori).  
14" Interlacciato Svga Monitor.  
ATBus 2 Seriali / 1 Parallela / 1 Game.  
Tastiera 102 tasti.

## 486-33 MHZ L. 2.175.000

Intel 486-33 Mhz processor.  
128K Static Cache RAM.  
4MB High Speed RAM.  
AMI Bios con Shadow RAM Support.  
1.44MB Floppy Drive.  
100MB 14ms IDE Hard Disk.  
16 Bit Svga Card 1MB  
14" Interlacciato Svga Monitor.  
ATBus 2 Seriali / 1 Parallela / 1 Game.  
Tastiera 102 tasti.

## 386-40 MHZ L. 1.620.000

AMD 386-40 Mhz processor.  
64K Static Cache RAM.  
4MB High Speed RAM.  
AMI Bios con Shadow RAM Support.  
1.44MB Floppy Drive.  
100MB 14ms IDE Hard Disk.  
16 Bit Svga Card 1MB.  
14" Interlacciato Svga Monitor.  
ATBus 2 Seriali / 1 Parallela / 1 Game.  
Tastiera 102 tasti.

## PARTI SCIOLTE

• M/B 386-40 cache	L. 310.000
• M/B 486-33 cache	L. 830.000
• M/B 486-50 cache	L. 1.290.000
• Case Desk display	L. 85.000
• Case Minitower dis.	L. 109.000
• 1.44MB Floppy Drive	L. 80.000
• 1.2MB Floppy Drive	L. 91.000
• HD 80MB ATBus	L. 370.000
• HD 100MB ATBus	L. 490.000
• HD 200MB ATBus	L. 750.000
• SIMM 1MB 70/60 ns	L. 55.000
• SIMM 4MB 70/60 ns	L. 220.000
• SVGA 1MB	L. 109.000
• SVGA 1MB ET4000 64K	L. 190.000
• SVGA Windows acc.	L. 175.000

## OFFERTE SPECIALI

M/B 486 DX2-66 256k cache	L. 1.390.000
MS-DCS 5.0 + Windows 3.1	L. 150.000
Controller intell. ATBUS 512k	L. 190.000

Permuta la tua CPU 486-DX 33  
con la nuovissima 486-DX2 66  
per sole L. 555.000



GARANZIA FINO A 5 ANNI - ASSISTENZA RAPIDA - LABORATORIO IN SEDE  
PREZZI AL NETTO DI IVA - AMPI SHOW-ROOM - POSSIBILITA' LEASING -  
TRASFORMAZIONI - PREVENTIVI PERSONALIZZATI.

I MARCHI RIPORTATI SONO REGISTRATI DAI LEGITTIMI PROPRIETARI.

CONSEGNA RAPIDA IN TUTTA L'ITALIA.

A CAUSA DELLE CONTINUE OSCILLAZIONI VALUTARIE IL LISTINO PREZZI FA RIFERIMENTO A DOLLARO = 1500 LIRE.

I COMPUTER ETS SONO DISTRIBUITI DA :

### MILANO

viale Lucania, 15  
Tel. 02- 5696046  
Fax. 02-90687040

### MILANO

via S.Gregorio, 8  
Tel. 02-29519990  
Fax. 02-29512854

### LIVORNO

Corso Matteotti, 48  
Cecina  
Tel. 0586-631240

### ANCONA

Via delle Grazie, 46  
Tel. 071-2802243

### BERGAMO

via Marconi, 50  
Petosino  
Tel. 035-571506

### CINISELLO B.

via M.te Spluga, 13  
Tel. 02- 66014703

### VICENZA

via Giardino, 81  
S.Giacomo  
Tel. 0424-513340