Corso di Visual Basic (Parte I)

Come muovere i primi passi con Microsoft Visual Basic, lo strumento da molti considerato ideale per la realizzazione di applicazioni di piccola e media entità in ambiente Windows. *di* ***Maurizio Crespi***

La nascita di Microsoft Visual Basic ha comportato una rivoluzione nel mondo degli strumenti di sviluppo per Windows, lanciando un nuovo modo di programmare, che vede per la prima volta prevalere l'uso del mouse nei confronti di quello della tastiera.

Tale approccio, detto *visuale*, si basa sull'uso di oggetti, ovvero di elementi attivi selezionabili su una barra degli strumenti (*toolbox*), caratterizzati dall'essere descritti da un insieme di valori, detti *proprietà* ed in grado di eseguire dei *metodi*, ovvero delle azioni che ne possono influenzare lo stato.

Un'ulteriore particolarità degli oggetti consiste nella capacità di generare degli eventi, a cui il programmatore può associare come risposta degli insiemi di istruzioni, detti *procedure* (ad esempio quando viene premuto un bottone il programmatore può associare la chiusura di una finestra).

Il linguaggio con cui esse sono scritte si basa sulle regole sintattiche previste dal BASIC, di cui mantiene la semplicità.

La creazione di un form

Un'applicazione Visual Basic è costituita da una o più finestre, dette *form*, sulle quali si trovano gli oggetti che formano l'interfaccia fra l'applicazione e l'utente.

Su un form è possibile inserire pressoché qualsiasi elemento, sia che si tratti di un'immagine, di un box di testo, di una lista o di un pulsante.

Il form principale dell'applicazione è generato automaticamente ogni volta che si agisce sul menu *File* per creare un nuovo progetto ed è visualizzato all'avvio dell'applicazione, di cui, in genere, rappresenta la schermata principale.

La sua chiusura determina la fine dell'esecuzione del programma.

Si supponga di voler realizzare un'applicazione che, alla pressione di un pulsante, faccia apparire in un punto preciso della finestra la scritta "*Benvenuto in Visual Basic*". Il primo passo consiste nel posizionare sul form gli elementi grafici necessari.

Si deve dapprima inserire un contenitore per il testo da visualizzare.

Visual Basic prevede due tipi di elementi adatti a questo scopo: si tratta delle *label* e delle *textbox*.

Le prime, come indica il nome, sono dedicate alla visualizzazione di "etichette", ovvero di porzioni di testo non modificabili dall'utente. Le seconde, invece, consentono la digitazione di un testo al proprio interno e si rivelano pertanto adatte all'acquisizione di dati.

Nel caso dell'esempio, dovendo visualizzare un testo modificabile solo dal programma, è conveniente optare per una *label*, che può essere inserita sul form rendendo visibile la barra degli strumenti di disegno, operazione effettuabile agendo sulla voce *Toolbox* del menu *View*, selezionando su di essa l'elemento contraddistinto dalla lettera "A" e tracciando con il mouse un rettangolo sull'area destinata ad ospitare l'etichetta.

Al rilascio del pulsante sinistro del mouse appare l'elemento desiderato, contenente la scritta "Label1".

Con questa operazione si crea il primo oggetto. Per adeguarlo alle proprie esigenze, è necessario variarne le proprietà. Ciò è possibile facendo clic su di esso con il mouse e visualizzando la finestra delle proprietà agendo sulla voce *Properties Window* del menu *View*.

Appare così una tabella, in cui la colonna di sinistra contiene i nomi di tutte le proprietà che descrivono l'elemento e la colonna di destra ne indica i rispettivi valori. Nel caso dell'etichetta testuale appena creata, si nota che due sue proprietà contengono il testo "Label1".

Esse sono denominate, rispettivamente, *Name* e *Caption*. Come è facile immaginare, esse definiscono rispettivamente il nome dell'oggetto, ovvero l'identificatore da usare ogniqualvolta si desideri fare riferimento ad esso e il testo visualizzato nell'etichetta.

Si noti che Visual Basic provvede ad assegnare a queste proprietà un valore predefinito. Spesso, però, è necessario fare in modo che la *label* visualizzi un dato diverso.

Nell'esempio, l'etichetta deve visualizzare del testo solo dopo la pressione del pulsante. È pertanto necessario che all'avvio del programma il suo contenuto sia nullo. Affinché ciò avvenga, si deve fare clic con il mouse sulla colonna di destra in corrispondenza della proprietà *Caption* e si deve sostituirne il contenuto con uno spazio.

Sebbene sia possibile mantenere per gli oggetti i nomi predefiniti, è consigliabile, al fine di rendere più facilmente leggibile il programma, assegnare ad essi degli identificatori che permettano ad un altro programmatore, o allo stesso autore dopo che sia trascorso un considerevole lasso di tempo, di poterne facilmente comprendere la funzione. Ciò rende più semplice ogni eventuale modifica successiva.

Nel caso dell'esempio, è pertanto conveniente assegnare all'etichetta il nome *lblMessaggio*. Si noti l'uso del prefisso "lbl", abbreviazione di "label". Il ricorso

a un prefisso è assolutamente facoltativo, ma si rivela estremamente utile nel caso di applicazioni complesse per fare sì che a colpo d'occhio sia possibile comprendere il tipo di componente a cui si sta facendo riferimento.

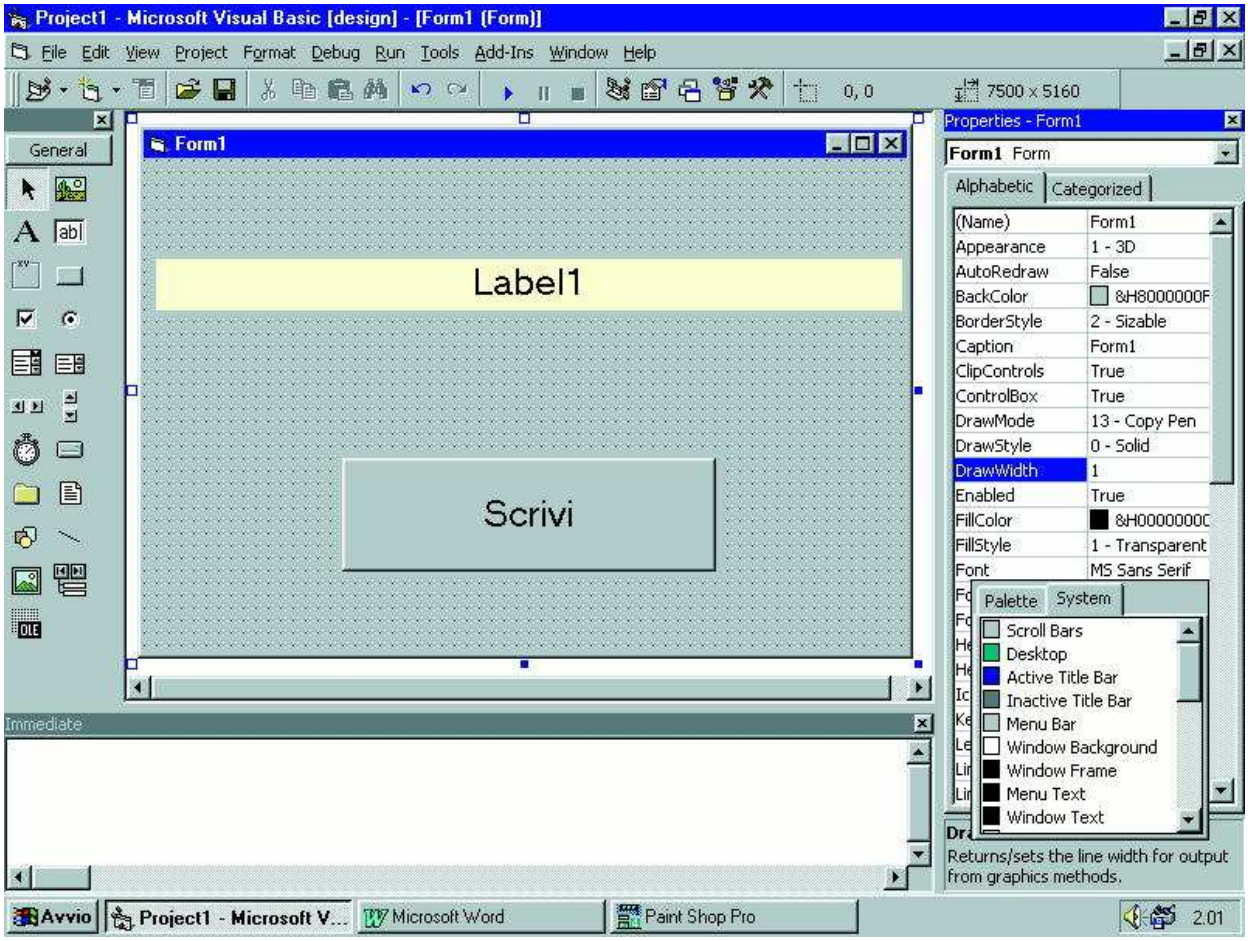
La modifica del nome dell'oggetto avviene digitando il nuovo identificatore nella colonna di destra della finestra *Properties*, in corrispondenza della riga denominata *Name*.

In modo analogo a quanto visto per la label, è possibile inserire il pulsante all'interno del form selezionando sulla toolbox l'apposita icona e tracciando un rettangolo avente le dimensioni dell'oggetto da creare.

Queste ultime possono essere modificate in qualsiasi momento trascinando i bordi dell'elemento con il mouse. Sempre con l'ausilio di questo strumento, è possibile spostare qualsiasi oggetto in una diversa posizione. Agendo sulla finestra delle proprietà, è possibile denominare l'oggetto *btScrivi* (ove "bt" rappresenta l'abbreviazione di bottone) e associare alla proprietà *Caption*, che contiene il testo che deve apparire sul pulsante, il valore "Scrivi".

Gli eventi

Il form appare come visualizzato in Figura 1.



A questo punto è necessario rendere "attivo" il bottone, ovvero fare in modo che al verificarsi dell'evento costituito dalla sua pressione sia eseguito del codice che provveda a modificare il contenuto della label. A tal fine, occorre selezionare il pulsante e fare su di esso un doppio clic con il mouse.

Si provoca così l'apertura di una finestra caratterizzata da una barra nella parte superiore, in cui è possibile osservare due liste a scomparsa.

Quella posta a sinistra contiene il nome dell'oggetto, mentre in quella di destra sono elencati tutti i possibili eventi che esso può generare.

Volendo fare in modo che la risposta avvenga in occasione della pressione del pulsante, occorre selezionare nella lista l'evento *Click*. Si provoca così la creazione della procedura *btScrivi_Click*, di cui è automaticamente visualizzata l'intestazione nella parte inferiore della finestra, seguita dalla frase *End Sub*, che indica la fine del blocco di codice.

Le istruzioni che devono essere eseguite in risposta all'evento vanno inserite nello spazio compreso fra le righe generate automaticamente, avendo l'accortezza di indicare un solo comando per linea. Per fare in modo che il programma visualizzi per mezzo della label *lblMessaggio* il testo "*Benvenuto in Visual Basic*", è necessario variare la proprietà *Caption* di quest'ultima. La procedura da associare al pulsante è quindi la seguente:

```
Sub btScrivi_Click ()

lblMessaggio.Caption = "Benvenuto in Visual Basic"

End Sub
```

Si noti che il nome della proprietà segue quello dell'oggetto, da cui è separato per mezzo di un punto.

Per verificare il funzionamento dell'applicazione, è necessario selezionare la voce *Start* del menu *Run*, oppure premere il tasto *F5*.

È così possibile osservare che, pur digitando una sola riga di codice, è stato creato un vero programma operante in ambiente Windows con un'interfaccia grafica.

I metodi

Come si è già accennato, ogni oggetto è in grado di ricevere dei comandi che possono provocare la variazione di alcune proprietà o la generazione di eventi.

Si tratta dei metodi. L'invocazione di un metodo avviene secondo la sintassi:

```
<oggetto>.<metodo> [<parametro>, ... ,<parametro>]
```

Al nome dell'oggetto è necessario far seguire, separato da un punto, quello del metodo e gli eventuali parametri.

Ad esempio, l'aggiunta della riga

```
lblMessaggio.Move 0,0
```

nella procedura vista in precedenza, provoca l'esecuzione del metodo *Move* da parte dall'oggetto *lblMessaggio*. La coppia *0, 0* rappresenta l'insieme dei parametri.

L'effetto che si ottiene consiste nello spostamento dell'etichetta testuale nell'angolo superiore sinistro del form, ovvero nel punto di cui i parametri rappresentano le coordinate.

Le variabili

Anche Visual Basic, come tutti i linguaggi di programmazione, prevede l'uso delle variabili, mediante le quali è possibile memorizzare dei valori testuali o numerici in strutture a cui il programma può accedere grazie a un nome assegnato loro in fase di creazione.

Una variabile è detta *locale* quando è definita all'interno di una procedura; la sua creazione avviene quando si fa riferimento ad essa per la prima volta, oppure quando è eseguita l'istruzione *Dim*, che presenta la seguente sintassi:

```
Dim <nome> [As <tipo>]
```

in cui <*nome*> rappresenta il nome da assegnare alla variabile.

Esso non si sottrae alla regola valida per tutti gli identificatori, ivi compresi i nomi degli oggetti, che impone loro di essere costituiti da delle sequenze di caratteri alfabetici o numerici prive di spazi ed aventi per iniziali delle lettere dell'alfabeto.

È inoltre possibile, anche se non indispensabile (come sottolineato dalla presenza delle parentesi quadre), aggiungere all'istruzione *Dim* un'indicazione del tipo di dato da creare. Visual Basic prevede numerosi tipi standard. I più utilizzati sono *integer*, usato per rappresentare i dati numerici interi compresi fra -32,768 a 32,767, *string*, che può contenere delle sequenze (stringhe) alfanumeriche composte al massimo da circa 65500 caratteri, *long*, in grado di ospitare numeri interi compresi fra -2,147,483,648 e 2,147,483,647, *single* e *double*, con cui è possibile memorizzare numeri reali anche di notevole entità.

Se invece è omessa la dichiarazione del tipo, la variabile è definita *variant*. Questo formato non è previsto dalla maggior parte dei linguaggi di programmazione tradizionali. La sua caratteristica fondamentale è l'universalità. Una variabile *variant*, infatti, può contenere dei dati aventi qualsiasi formato.

È buona norma, tuttavia, non fare largo uso di strutture di questo tipo, in quanto la loro gestione da parte dell'interprete è poco efficiente.

Una variabile locale può anche non essere dichiarata. In questo caso, la sua creazione avviene la prima volta in cui si fa riferimento ad essa.

In assenza di dichiarazione, è necessario aggiungere al nome un identificatore di tipo, costituito da un carattere. In caso di sua omissione, la variabile è creata di tipo *Variant*. I principali identificatori di tipo sono i seguenti: $ (*String*), % (*Integer*), & (*Long*), ! (*Single), #* (*Double*).

Ad esempio, è possibile creare una variabile di tipo numerico intero ed assegnarle il valore 5 sia scrivendo

```
Dim A as Integer

A = 5
```

sia con la riga

```
A% = 5
```

Dopo l'esecuzione dell'ultima istruzione presente nella procedura in cui sono state create, le variabili locali sono automaticamente distrutte.

Per fare in modo che il loro valore sia conservato anche in occasione delle chiamate successive, è necessario dichiarare le variabili sostituendo *Static* alla parola chiave *Dim*.

In alcuni casi, si rivela necessario fare in modo che una variabile non sia mai distrutta e sia accessibile anche dalle altre procedure presenti in un form.

In questo caso, è necessario selezionare la sezione *General* nella lista a discesa posta nella parte superiore sinistra della finestra contenente il codice associato al form, la voce *Declarations* nella lista di destra e dichiarare la variabile per mezzo dell'istruzione *Dim*.

Quando invece si presenta la necessità di fare in modo che una struttura sia accessibile da tutte le procedure presenti nell'applicazione, indipendentemente dal form in cui esse si trovano, è necessario utilizzare per la dichiarazione la parola chiave *Global*.

Una variabile dichiarata in questo modo è detta *globale*. La sua dichiarazione è impossibile all'interno di un form. Essa deve essere effettuata in un modulo, ovvero in un file di testo caratterizzato dall'estensione .BAS, che è aggiunto al progetto selezionando la voce *Add Module* del menu *Project*. La riga di codice rappresenta un esempio di dichiarazione di una variabile globale:

```
Global NomeCognome as String
```

Un esempio di utilizzo di una variabile

Si consideri l'applicazione precedentemente realizzata. Si desidera modificarla per creare un semplice strumento in grado di visualizzare il tempo trascorso dall'ultima pressione del pulsante *btScrivi*.

Per fare ciò, è necessario aggiungere al form un *timer*. Si tratta di un oggetto in grado di generare un evento ad intervalli regolari, la cui frequenza è decisa dal programmatore.

Per inserire tale elemento, occorre selezionare sulla *toolbox* l'icona a forma di cronografo e disegnare un rettangolo sul form. Si provvede poi a selezionare il nuovo oggetto, a visualizzare la finestra *Properties* e ad assegnare il valore 1000 alla proprietà *Interval*.

In tal modo si definisce il periodo (in millisecondi) che intercorre fra due eventi. L'impostazione effettuata provoca la generazione di un evento ogni secondo.

Anche in questo caso, è possibile associare al verificarsi di questi eventi un insieme di istruzioni da eseguire. Ciò è possibile facendo doppio clic sull'oggetto.

Si provoca così l'apertura della finestra del codice sorgente e la creazione della procedura *Timer1_Timer*, in cui è possibile scrivere:

```
Sub Timer1_Timer ()

NumSecondi = NumSecondi + 1

lblMessaggio.Caption = Str$(NumSecondi)

End Sub
```

Essa ha lo scopo di incrementare di un'unità il valore della variabile *NumSecondi*, usata per conteggiare i secondi e di visualizzare tale valore all'interno della label. A tal fine, il risultato del conteggio è convertito in testo tramite la funzione *Str$* ed è assegnato alla proprietà *Caption* dell'oggetto che provvede a visualizzarlo sullo schermo.

Per consentire l'azzeramento del contasecondi in seguito alla pressione del pulsante *btScrivi*, è necessario modificare la procedura associata a tale evento.

Ciò è possibile facendo doppio clic sul bottone e sostituendo le istruzioni precedentemente inserite con le seguenti:

```
Sub btScrivi_Click ()

NumSecondi = 0
```

```
End Sub
```

Affinché la variabile *NumSecondi* possa essere utilizzata da entrambe le procedure, è necessario che sia accessibile in tutto il form. Come visto in precedenza, perché ciò sia possibile, si deve digitare nella sezione *Global/Declarations* la riga

```
Dim NumSecondi as Long
```

Conclusioni

Come dimostrato dagli esempi sopra descritti, si può realizzare un'applicazione funzionante in ambiente Windows in un batter d'occhio, semplicemente facendo uso del mouse e scrivendo un numero davvero ridotto di righe di codice.

È proprio grazie alla notevole facilità d'uso, oltre che all'affidabilità dell'interprete di cui è dotato, che Visual Basic è da molti considerato lo strumento ideale per la realizzazione di programmi di piccola e media entità.