

VISUAL BASIC

Cos'è VB

VB è un linguaggio di programmazione che consente di realizzare applicativi eseguibili nell'ambiente Microsoft Windows; i progetti che vengono realizzati appaiono e funzionano come tutti i programmi standard di Window.

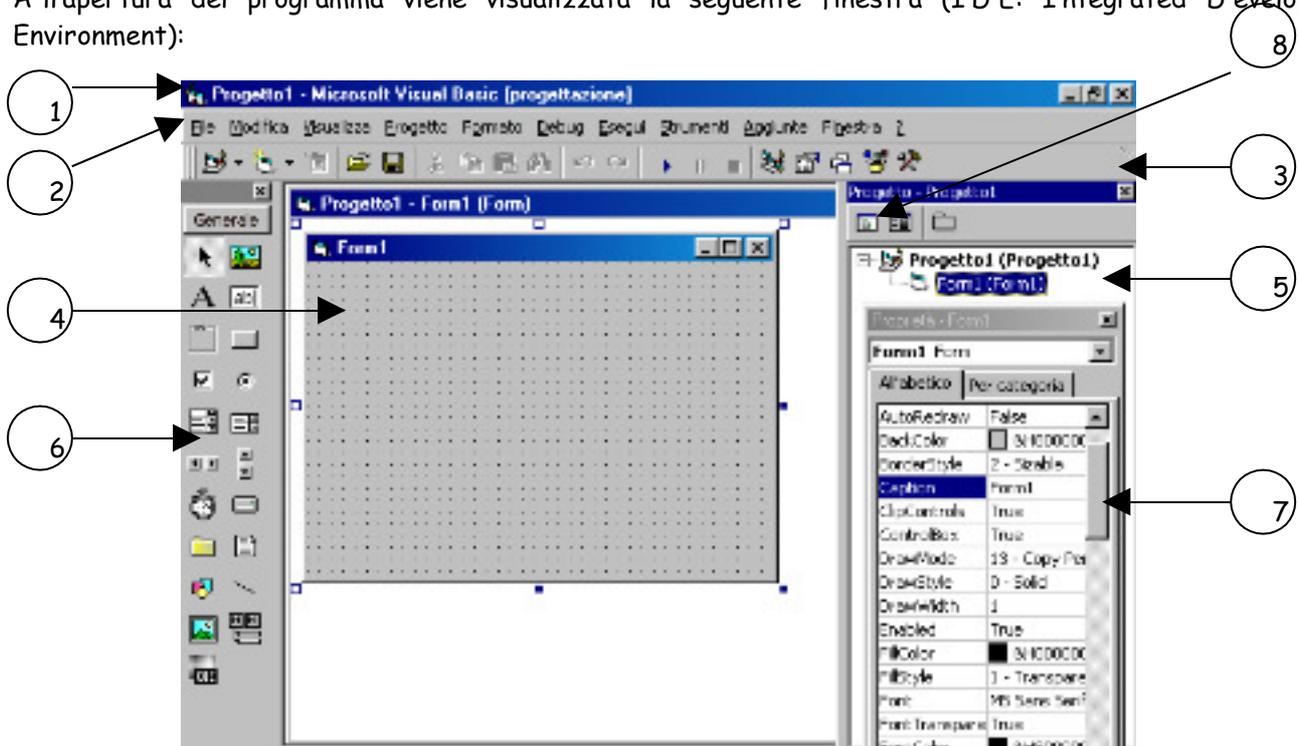
Esistono centinaia di linguaggi di programmazione e ognuno di essi è nato per risolvere particolari problemi. La maggior parte dei linguaggi tradizionali (come PASCAL, COBOL, BASIC) sono considerati linguaggi procedurali nel senso che il programma specifica passo passo la sequenza di tutte le operazioni (cioè la logica del programma determina la successiva istruzione da eseguire, in risposta a condizioni e a richieste dell'utente); tali linguaggi sono anche detti strutturati, in quanto basati su strutture di controllo fondamentali (sequenza, selezione e iterazioni) che consentono al programmatore di controllare il flusso del programma.

Abbiamo utilizzato inoltre il linguaggio SQL, che è un linguaggio non procedurale ma strutturato, che consente di realizzare le operazioni fondamentali sulle basi di dati (proiezione, selezione e congiunzione). Infatti esistono anche linguaggi di programmazione non procedurali che utilizzano un approccio alla programmazione diverso: la programmazione orientata agli oggetti (OOP, object oriented programming) e la programmazione basata su eventi; tali linguaggi vengono anche detti "programmazione visuale" (visual programming) e consentono di produrre software basato sull'uso dell'interfaccia grafica.

Microsoft considera VB un linguaggio basato sugli eventi, nel senso che ogni azione dell'utente (come la pressione di un tasto o il clic del mouse su un pulsante) può provocare l'attivazione di un evento che determina l'esecuzione di una routine (o procedura) Basic scritta dal programmatore; quindi l'ordine in cui viene eseguito il codice dipende dagli eventi che si verificano, che a loro volta dipendono dalle operazioni svolte dall'utente e non prevedibili dal programmatore (in questo senso il linguaggio è "non procedurale"). VB è comunque un linguaggio strutturato perché utilizza le strutture di controllo fondamentali.

L'ambiente VB

All'apertura del programma viene visualizzata la seguente finestra (IDE: Integrated Development Environment):



dove:

1. finestra principale VB
2. barra dei menu'
3. barra degli strumenti
4. finestra del form
5. finestra progetto
6. casella degli strumenti (controlli VB)
7. finestra delle proprieta' (dei form e dei controlli)
8. apre la finestra del codice

Nella casella degli strumenti sono presenti i seguenti controlli standard:

CONTROLLO	NOME	DESCRIZIONE	AZIONE
	Pointer	P untatore	Se non e' selezionato un altro controllo e' sempre attivo; consente di spostare e ridimensionare i form e i controlli
	Picture box	Casella immagine	Crea il contenitore per grafici .bmp, .ico, .jpg, .gif che l'utente non puo' modificare
	Label	Etichetta	Visualizza un messaggio o un'intestazione che l'utente non puo' modificare
	Text box	Casella di testo	Crea un box di testo in cui l'utente puo' inserire o modificare dati
	Frame	Cornice	Crea una cornice per contenere altri controlli
	Command button	Pulsante di comando	Pulsante di comando che esegue una azione associata (OK, annulla,...)
	Check box	Casella di controllo	Consente di gestire le scelte dell'utente (vero/falso, si/no)
	Option button	Pulsante di opzione	Consente di scegliere una sola azione tra un insieme di opzioni possibili
	Combo box	Casella combinata	Consente all'utente di operare una scelta scrivendola in un box di testo o selezionando una scelta in elenco
	List box	Casella di riepilogo	Presenta all'utente una lista all'interno della quale puo' scegliere
	Horizontal-vertical scroll bar	Barra scorrimento orizzontale-verticale	Crea una barra di scorrimento orizzontale o verticale per muoversi nelle liste o nei form di grande dimensione; puo' essere usata anche per indicare la posizione attuale in un range di valori
	Timer	Timer	Consente l'uso di un orologio per far partire un'esecuzione con uno specifico intervallo di tempo
	Drive list box	Casella di riepilogo delle unita'	Permette all'utente di selezionare il disco (A o C) durante l'esecuzione di un'applicazione
	Directory list box	Casella di riepilogo delle cartelle	Permette all'utente di selezionare cartelle del disco durante l'esecuzione di un'applicazione
	File list box	Casella di riepilogo dei file	Permette all'utente di scegliere i file durante l'esecuzione di un'applicazione
	Shape	Forma	Inserisce nel form figure tipo rettangoli, cerchi, ellissi
	Line	Linea	Disegna linee verticali, orizzontali, oblique
	Image	Immagine	Visualizza un'immagine BMP, JPEG, GIF, WMF o un'icona ICO e funziona da pulsante di comando
	OLE	Contenitore	Consente al programmatore di creare applicazioni

			OLE che possono visualizzare o permettere la modifica di dati provenienti da altre applicazioni: essi possono essere modificati all'interno dell'applicazione nella quale sono stati creati
	Data	Controllo dei dati	Per creare applicazioni in grado di visualizzare e manipolare le tabelle di un data base

Questi sono i controlli normalmente presenti nella casella degli strumenti; per aggiungere un controllo si seleziona la voce "componenti" dal menu' di scelta rapida della casella degli strumenti, che si apre facendo clic con il pulsante destro sulla casella stessa; viene visualizzata la finestra di dialogo "componenti" in cui scegliere il controllo.

Altri controlli:

	BD Grid	Griglia associata a database	Consente di inserire un controllo che permette all'utente di manipolare i dati di una tabella di database
	Crystal Report	Generatore di report	Consente di disegnare il layout di stampa, in autocomposizione, attraverso una serie di finestre di dialogo

Come si lavora in VB

In VB si lavora con oggetti, che possiedono proprietà e metodi. Gli oggetti sono form e controlli; un oggetto può essere un singolo elemento di un'applicazione, oppure una intera applicazione. Si può pensare ad un oggetto come ad un sostantivo (es. form: finestra, label: etichetta, ecc.)

Un form è una finestra che costituisce l'interfaccia per l'utente dell'applicazione; ogni form può contenere controlli, immagini grafiche o altri form.

Un controllo è uno strumento disegnato dal programmatore su un form utilizzando la casella degli strumenti, per ricevere input e visualizzare output

Le proprietà forniscono informazioni relative ad un oggetto (il nome, il colore, la dimensione, il funzionamento, ecc); si può pensare alla proprietà come ad un aggettivo che descrive l'oggetto.

Tra le proprietà presenti in ogni oggetto troviamo:

➤Name: il nome dell'oggetto, che è possibile modificare (label, text box... .)

➤Caption: è il nome che appare sul controllo (es pulsante) e viene assegnato dal programmatore

Le azioni associate ad un oggetto sono dette metodi; un metodo è un verbo della programmazione orientata agli oggetti (show, print, move, clear, ecc) e consente di definire il comportamento di un oggetto durante l'esecuzione dell'applicazione (più avanti vedremo diversi esempi).

Un progetto VB è l'insieme di form, codice e file dell'applicazione; la realizzazione di un progetto VB è suddivisa in tre fasi:

1. definizione dell'interfaccia utente: si imposta lo schermo che l'utente vedrà quando esegue l'applicazione; in questa fase si definiscono i form e i controlli, cioè tutti gli oggetti che saranno usati nel progetto.
2. impostazione delle proprietà degli oggetti: si descrive ogni oggetto presente nel progetto assegnando le proprietà, cioè scegliendole nella finestra delle proprietà.
3. scrittura del codice Basic: si scrivono le routine necessarie per poter eseguire il progetto utilizzando il linguaggio Basic (ad un oggetto può essere associato il codice: cliccando due volte sull'oggetto o utilizzando l'apposito pulsante, viene aperta la finestra del codice); si noterà quante poche istruzioni sono necessarie per creare un potente programma Windows (a differenza dei linguaggi procedurali).

Come impostare il progetto

➤creare una cartella per il progetto; ogni progetto VB presenta almeno due file:

	Nomeform.FRM (Visual Basic Form File): è il file contenente il form del progetto (inizialmente i nostri progetti avranno un solo form); se il progetto è composta da più form avremo tanti file di form
---	---

 Progetto1	Nomeprogetto.VBP (VB Project): e' il file del progetto, che contiene i nomi (metadati) degli altri file presenti nel progetto
--	---

Inoltre possono essere presenti altri file:

 Progetto1	Nomeprogetto.VBW: contiene informazioni sul form
 Progetto1	Nomeprogetto.EXE: e' il file eseguibile, che non necessita del linguaggio VB per essere eseguito; puo' essere creato al termine del progetto e fornito all'utente senza il file di progetto

- > disegnare il primo form dell'applicazione utilizzando la casella degli strumenti, associare le proprietà ai controlli e scrivere il codice
- > eseguire il progetto per controllare il funzionamento dell'applicazione (aprire il menu ESEGUI e scegliere A VVI A oppure premere il pulsante A VVI A nella barra degli strumenti oppure premere F5)
- > se il progetto funziona correttamente e' possibile creare il file eseguibile scegliendo FILE/ CREA nomeprogetto.EXE e specificando opzioni come il numero di versione, l'icona dell'applicazione, l'autore, ecc.; il file eseguibile consente di lanciare l'applicazione da Windows (dopo aver chiuso l'ambiente VB) cliccando due volte sull'icona, come per qualsiasi applicazione Windows.

Ora apprenderemo l'uso del linguaggio svolgendo diversi esercizi, relativi ai seguenti argomenti

- Progetto 1: utilizzo di form, pulsanti ed etichette
- Progetto 2: utilizzo di caselle di testo e metodi
- Progetto 3: il codice VB: dati e strutture di controllo
- Progetto 4: un po' di relax, Splash screen e definizione di menu'
- Progetto 5: gestione di vettori
- Progetto 6: gestione di file
- Progetto 7: gestione di tabelle di database
- Progetto 8: VB e SQL
- Progetto 9: report di stampa

PROGETTO 1: utilizzo di form, pulsanti ed etichette

Questo semplice progetto visualizza il messaggio "ciao ciao ciao" quando l'utente clicca sul pulsante "premimi", e si chiuderà quando l'utente preme il pulsante "esci".

Innanzitutto creiamo un cartella PROGETTO1 e una sottocartella (ad es. PRIMO) che conterrà tutti i file del progetto; entrare in ambiente VB e scegliere il tipo di progetto EXE standard; se la finestra del form non è attiva, selezionarla cliccando su "finestra progetto" (v.5).

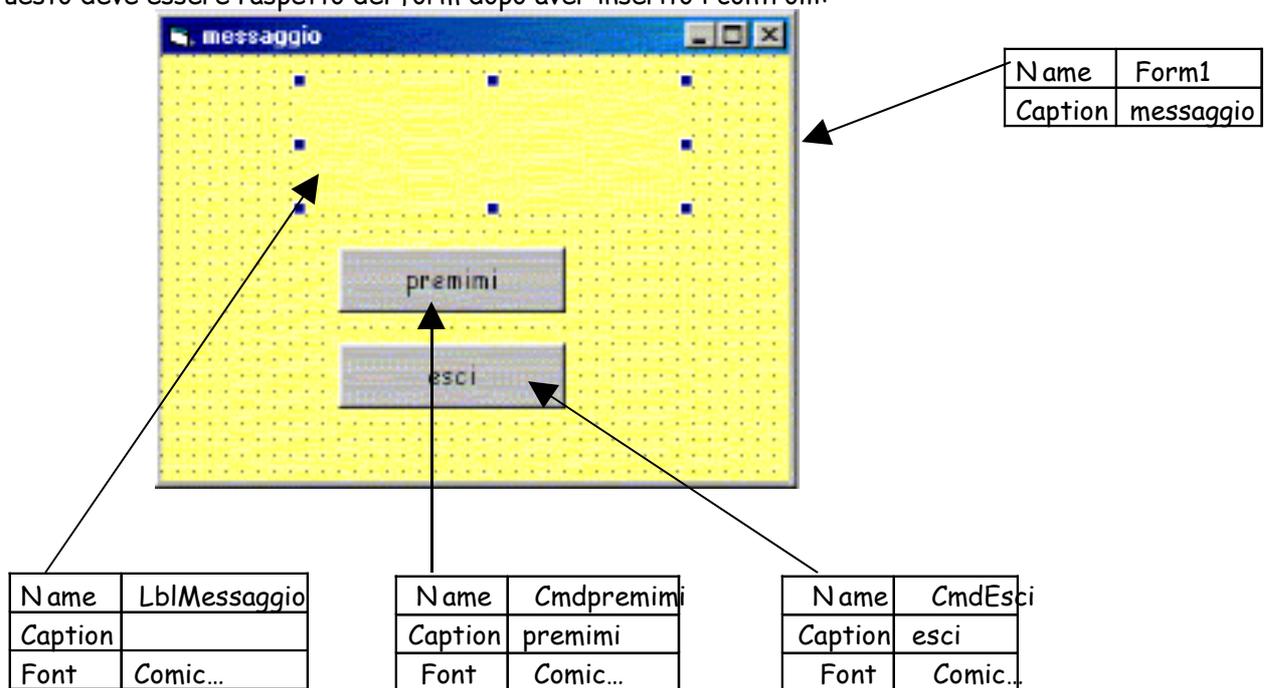
Ora disegniamo l'interfaccia grafica.

Il primo oggetto dell'applicazione è la finestra del form le cui proprietà sono evidenziate nella finestra delle proprietà: il nome è form1 (non lo modifichiamo) e la caption predefinita è form1; modifichiamo la caption scrivendo "messaggio": questo è il titolo che apparirà nella barra della finestra. Possiamo anche modificare il colore dello sfondo della finestra modificando la proprietà Backcolor.

Il secondo oggetto è un'etichetta (che conterrà il testo del messaggio); disegniamola sul form; il nome è label1, ma possiamo modificarlo in LblMessaggio; cancelliamo la caption in modo che non appaia nulla all'interno dell'etichetta e impostiamo la proprietà backcolor allo stesso colore della finestra.

Il terzo e quarto oggetto da disegnare sul form sono pulsanti di comando; il primo pulsante ha nome predefinito Command1 e lo modifichiamo in CmdPremimi; digitiamo la caption "premimi" e scegliamo il font del carattere; il secondo pulsante si chiama Command2 (lo modifichiamo in CmdEsci), impostiamo la caption "esci" e il font.

Questo deve essere l'aspetto del form dopo aver inserito i controlli:



NB è possibile copiare uno o più oggetti usando le solite funzioni "copia" e "incolla"!

Ora dobbiamo scrivere il codice, cioè dobbiamo associare ai controlli il codice che descrive le azioni da compiere in corrispondenza degli eventi che accadono durante l'esecuzione del programma.

Per aprire la finestra del codice basta fare doppio clic sull'oggetto.

Iniziamo dal pulsante "premi": quando l'utente clicca (provoca l'evento click) sul pulsante (CmdP remimi) deve apparire il messaggio nell'etichetta (LblMessaggio); il codice e' il seguente:

```
Private Sub CmdPremimi_Click()  
'visualizza il messaggio nell'etichetta  
LblMessaggio.Caption = "ciao ciao ciao"  
End Sub
```

Tutte le routine iniziano con la parola riservata "private sub" e terminano con "end sub"; "CmdP remimi_Click()" indica l'azione associata al pulsante; il commento inizia con ' ; LblMessaggio.caption=" ciao... " associa all'etichetta la scritta "ciao..", che appare q uando viene premuto il pulsante CmdP remimi.

O ra associamo il codice al pulsante "esci":

```
Private Sub CmdEsci_Click()  
'permette di terminare l'applicazione  
End  
End Sub
```

Q uando l'utente preme (click) il pulsante (CmdEsci), l'applicazione termina (end).

O ra verifichiamo il funzionamento dell'applicazione; per lanciare l'esecuzione si puo':

- Scegliere ESEGUI /A VVI A
- P remere il pulsante avvia nella barra degli strumenti
- P remere F5

D urante l'esecuzione il form assume q uesto aspetto:



Se il programma funziona correttamente, possiamo creare l'eseguibile (EXE): scegliere FILE/ "crea nomeprogetto.exe", decidere la destinazione (es. la cartella progetto1) e il nome del file eseguibile (es ciao).

O ra dobbiamo salvare l'intero progetto nella cartella progetto1: scegliere FILE/SA LVA P ROGET T O CON NOME; vengono salvati form1 e progetto1.

P er eseguire l'applicazione all'esterno dell'ambiente VB ora basta cliccare sull' icona denominata "ciao".

NOTA: gli oggetti presentano dei nomi predefiniti (es. Command1, Command2, Label1, ecc.); come abbiamo visto q uesti nomi possono essere modificati con nomi significativi. D i solito si usa il prefisso Cmd per i Command, Lbl per le label, Txt per le caselle di testo. Non e' obbligatorio farlo, ma e' conveniente q uando un'applicazione contiene molti controlli di cui si potrebbe dimenticare il nome.

Nel seguito si utilizzeranno, per comodita', i nomi predefiniti.

Esercizio: creare un'applicazione (v. figura) che visualizzi quattro diversi proverbi a seconda del pulsante che viene premuto e che permetta di terminare premendo il pulsante uscita.



soluzione (memorizzare tutti i file del progetto in una sottocartella della cartella P R O G E T T O 1):
 Descrizione dei controlli utilizzati :

FORM1: name: form1; caption: "proverbi"; font:...

ET ICHETTA : name: label1; caption: nessuna; font:...

PULSANTE1: name: command1; caption: "proverbio 1"; font:...

PULSANTE2 : name: command2; caption: "proverbio 2 " ;font:...

PULSANTE3: name: command3; caption: "proverbio 3"; font:...

PULSANTE4: name: command4; caption: "proverbio 4"; font:...

PULSANTE5: name: command5; caption: "esci"



Il codice associato ai controlli e':

```
Private Sub Command1_Click()
'visualizza un proverbio
Label1.Caption = "tanto va la gatta al lardo che ci lascia lo z ampino"
End Sub
```

```
Private Sub Command2_Click()
'visualizza un proverbio
Label1.Caption = "cielo a pecorelle acqua a catinelle"
End Sub
```

```
Private Sub Command3_Click()
'visualizza un proverbio
Label1.Caption = "meglio un uovo oggi che una gallina domani"
End Sub
```

```
Private Sub Command4_Click()
'visualizza un proverbio
Label1.Caption = "il mattino ha l'oro in bocca"
End Sub
```

```
Private Sub Command5_Click()
End
End Sub
```

Esercizio: creare un'applicazione con cinque pulsanti; premendo un pulsante appare un saluto in una lingua; premendo un altro pulsante appare lo stesso saluto in un'altra lingua; ecc. deve essere incluso un pulsante che permette di uscire (v.figura). Usare sempre una sottocartella



esercizio. Ora costruiamo una maschera menu' principale, che consente di aprire altre due finestre. Quindi realizziamo un progetto costituito da tre form, come nella figura:



Il primo form (form1) contiene:

- Un'etichetta (label1) con caption:" menu' principale"
- Un frame (frame1) con caption: scelte
- Un pulsante di comando (command1 del form1) a cui e' associato il codice per l'uscita dal programma
- Due pulsanti di opzione (option1 con caption: "prima finestra" e option2 con caption: "seconda finestra") a cui e' associato il codice per l'apertura delle rispettive finestre

Il secondo form (form2) contiene:

- Un'etichetta (label1 nel form2) con caption:" prima finestra"
- Un pulsante di comando (command1 del form2) a cui e' associato il codice per chiudere la finestra, cioe' scaricare il form da memoria centrale e consentire di visualizzare il menu' principale

Analogamente il terzo form (form3) contiene label1 e command1

Il codice associato e':

```
Private Sub Command1_Click()
'termina l'esecuzione del programma
End
End Sub
```

```
Private Sub Form_Load()
'controlla il caricamento del form1
WindowState = 2 'visualizza il form a tutto schermo, ma si puo' omettere
Option1.Value = False
Option2.Value = False
End Sub
```

```
Private Sub Option1_Click()
'visualizza il form2 "prima finestra"
Form2.Show
End Sub
```

```
Private Sub Option2_Click()
'visualizza il form3 "seconda finestra"
Form3.Show
End Sub
```

```
Private Sub Command1_Click()
'scarica il form2
Unload Me
End Sub
```

```
Private Sub Command1_Click()
'scarica il form3
Unload Me
End Sub
```

Show e' un metodo che permette la visualizzazione del form; Unload Me e' l'istruzione con cui il form scarica se stesso dalla memoria

A i fini della documentazione del progetto e' possibile stampare form e codice utilizzando il comando Stampa dal menu' File.

PROGETTO 2: utilizzo di caselle di testo e metodi

Questo progetto consente di eseguire la somma di due numeri inseriti dall'utente quando viene premuto il tasto "+" e si chiude quando l'utente preme il pulsante "esci".

Innanzitutto creiamo un cartella PROGETTO2 che conterra' tutti i file del progetto; entrare in ambiente VB e scegliere il tipo di progetto EXE standard; se la finestra del form non e' attiva, selezionarla cliccando su "finestra progetto" (v.5).

Ora disegniamo l'interfaccia grafica.

Il primo oggetto dell'applicazione e' la finestra del form le cui proprieta' sono evidenziate nella finestra delle proprieta': il nome e' form1 (non lo modifichiamo) e la caption predefinita e' form1; modifichiamo la caption scrivendo "calcola": questo e' il titolo che apparira' nella barra della finestra. Possiamo anche modificare il colore dello sfondo della finestra modificando la proprieta' Backcolor.

Il secondo oggetto e' un'etichetta (che conterra' il testo "somma di due numeri"); disegniamola sul form; il nome e' label1; cancelliamo la caption, digitiamo il testo desiderato e impostiamo la proprieta' backcolor allo stesso colore della finestra.

Il terzo e quarto oggetto da disegnare sul form sono pulsanti di comando; il primo pulsante ha nome Command1, digitiamo la caption "+" e scegliamo il font del carattere; il secondo pulsante si chiama Command2, impostiamo la caption "esci" e il font.

Ora disegniamo tre oggetti di tipo caselle di testo; le caselle di testo hanno nome textn (dove n indica il numero di etichetta), e text impostato a textn (non esiste la proprieta' caption); per fare in modo che la casella appaia vuota si cancella la proprieta' text.

la prima casella ha nome text1 e cancelliamo text1 dalla proprieta' text; procediamo in modo analogo per la seconda e la terza.

Inoltre inseriamo un controllo linea dopo le due caselle di testo e impostiamo la proprieta' borderwidth=2 (consente di avere una linea piu' spessa).

Questo deve essere l'aspetto del form dopo aver inserito i controlli:

Name	Form1
Caption	calcola

Name	Label1
Caption	Somma...

Name	Text1
Text	

Name	Command1
Caption	+

Name	Text2
Text	

Name	Command2
Caption	esci

Name	Text3
Text	

Name	Line1
Borderwidth	2

ora associamo ai controlli il codice che descrive le azioni da compiere in corrispondenza degli eventi che accadono durante l'esecuzione del programma, in risposta alle azioni dell'utente.

Quando l'utente preme il pulsante "+" (command1), nella casella text3 deve apparire la somma dei due numeri inseriti; quando l'utente preme "esci" (command2) l'applicazione termina:

```
Private Sub Command1_Click()
```

```
'calcola la somma dei numeri inseriti nelle caselle
'di testo text1 e text2
Tex t3 .Text = Val(Text t1 .Text) + Val(Text t2.Text)
End Sub
```

```
Private Sub Command2_Click()
'esce dal programma
End
End Sub
```

Val e' una funzione che restituisce il valore numerico del contenuto di una casella di testo. Verifichiamo il funzionamento dell'applicazione, creiamo il file eseguibile e salviamo il progetto nella cartella **P R O G E T T O 2**.

Perfezioniamo l'applicazione! Riapriamo il nostro progetto.

Vogliamo inserire un pulsante che azzeri il contenuto delle tre caselle contenenti gli operandi e il risultato, per ripetere l'operazione; inoltre, dopo l'azzeramento, il cursore deve apparire nella prima casella di testo per inserire un nuovo numero.

Inseriamo sul form un pulsante **command3** impostandone la caption a "azzeri"; il codice da associare al comando e':

```
Private Sub Command3_Click()
'azzeri le caselle di testo
'e riporta il cursore sulla prima
Tex t1 .Text = " "
Tex t2.Text = " "
Tex t3 .Text = " "
Tex t1 .SetFocus
End Sub
```

Setfocus e' un metodo che riporta il cursore nella prima casella di testo. Possiamo anche fare in modo che, dopo aver premuto il tasto "+" ed aver visualizzato il risultato, sia attivo il tasto "azzeri" (**command3**); associamo alla casella del risultato (**text3**) il codice:

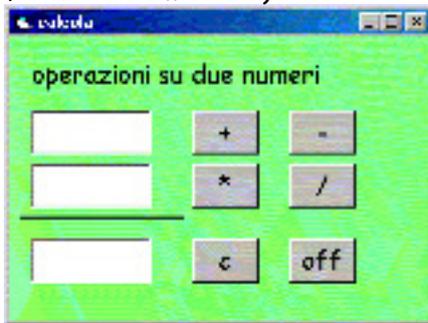
```
Private Sub Text3_Change()
'porta il cursore sul pulsante "azzeri"
Command3 .SetFocus
End Sub
```

Un ultimo perfezionamento: quando si lancia per la prima volta l'applicazione il cursore non appare nella prima casella di testo; per fare in modo che cio' avvenga bisogna associare al caricamento del form tale azione:

```
Private Sub Form_Load()
'sposta il cursore sulla prima casella di testo
Form1 .Show
Tex t1 .SetFocus
End Sub
```

Il metodo show visualizza il form dell'applicazione (**form1**); dopo aver mostrato il form, il cursore viene spostato sulla prima casella di testo (**text1**).

Esercizio creare un'applicazione (come in figura) che permetta di eseguire le quattro operazioni fondamentali premendo quattro pulsanti con i relativi simboli; deve essere presente anche un pulsante di azzeramento (c) e di uscita (off); (sappiamo che non e' possibile dividere un numero per zero; come fare? Vedremo oltre)



PROGETTO 3 : il codice VB: dati e strutture di controllo

A tutti gli oggetti e' associato un codice, cioe' istruzioni scritte in linguaggio VB; le istruzioni sono organizzate in routine, che in VB si chiamano SUB e FUNCTION.

La SUB ha la seguente struttura:

```
Sub nome()
```

```
....
```

```
End Sub
```

La FUNCTION ha la struttura:

```
Function nome (parametri) as valore
```

```
....
```

```
End Function
```

Di solito Sub e Function sono precedute dal tipo PRIVATE, per indicare una routine locale, cioe' visibile solo all'interno del form e vengono associate ai controlli cliccando due volte sull'oggetto. Ma il codice, oltre che associato ai controlli, puo' essere inserito nei progetti mediante i moduli (ricordate ACCESS?): essi sono componenti dell'applicazione VB registrati in file con estensione .BAS.

Il codice e' PUBLIC per indicare che i dati e le routine sono accessibili da qualunque form.

Una routine puo' richiamare al suo interno un'altra routine (scomposizione top-down):

```
Private sub operando1_Change()
```

```
    Azzera
```

```
End sub
```

```
Private Sub azzera()
```

```
    Label1.caption = ""
```

```
End sub
```

Il linguaggio VB e' interpretato (non compilato), nel senso che l'attivita' di controllo sintattico viene effettuata durante l'attivita' di editing, segnalando gli errori con un messaggio.

Le caratteristiche di VB sono:

>Le frasi di commento iniziano con un apostrofo, in qualsiasi punto del testo

>L'assegnazione di un valore ad una variabile avviene con l'istruzione: nome variabile = valore

>I nomi delle variabili devono iniziare con una lettera e possono contenere numeri e lettere fino a un massimo di 40 caratteri

>I tipi di dati sono:

tipo di dato	Dimensione	Intervallo
Byte	1 byte	Da 0 a 255
Boolean	2 byte	True o false
Integer	2 byte	Da -32768 a 32767
Long	4 byte	Da -2147483648 a 2147483647
Single	4 byte	Virgola mobile a precisione singola
Double	8 byte	Virgola mobile a precisione doppia
Date	8 byte	Da 1 gennaio 100 al 31 dicembre 9999
Currency	8 byte	Molto grande ! per la valuta
String	Lunghezza stringa	Da 1 a circa 65400

>In VB (come in Basic) non e' obbligatorio dichiarare le variabili prima di usarle (a differenza di Pascal, Cobol, ecc): la dichiarazione di una variabile avviene la prima volta che viene incontrata; puo' sembrare un vantaggio perche' rende piu' snello il codice, ma nasconde un'insidia: se il nome di una variabile viene successivamente scritto in modo errato, viene interpretato come una nuova variabile di cui e' possibile perdere il controllo.

E' buona regola di programmazione dichiarare sempre le variabili; si puo' fare in modo che venga richiesta la dichiarazione esplicita di tutte le variabili di un modulo, inserendo l'istruzione "Option Explicit" nella sezione dichiarazioni del modulo; inoltre e' possibile inserire automaticamente l'istruzione "Option Explicit" in tutti i nuovi moduli scegliendo "Opzioni" dal menu' "strumenti" e, nella scheda "Ambiente", attivare "dichiarazione di variabili obbligatoria".

La dichiarazione deve essere fatta nella sezione "generale" per le variabili che devono essere visibili da tutte le routine; se una variabile e' locale puo' essere dichiarata nella Sub che la utilizza.

➤ Per dichiarare i tipi di dato si usa l'istruzione DIM; per esempio, per dichiarare le seguenti variabili:
 dim nome as string, scelta as integer...

per i vettori e le matrici:

dim vettore(minimo to massimo) as long (es: dim vettore(1 to 50) as string)

dim matrice(minimo to Massimo, minimo to Massimo) as integer

➤ Nella dichiarazione delle variabili si può appendere al nome delle variabili il carattere che indica il tipo, in questo modo: nomevariabile\$ invece di DIM nomevariabile as string

carattere	Tipo di dato
%	Integer
&	Long
!	Single
#	Double
@	Currency
\$	String

➤ Le strutture di controllo sono:

• **Selezione**

Primo modo: IF condiz ione THEN istruz ioni
 ELSE istruz ioni
 END IF

Secondo modo: IF condiz ione THEN
 Istruz ioni
 ELSEIF condiz ione THEN
 Istruz ioni
 ELSE
 Istruz ioni
 ENDIF

• **Ripetizione:**

primo modo: DO [WHILE|UNTIL] condiz ione
 istruz ioni
 LOOP

secondo modo: DO
 istruz ioni
 LOOP [WHILE|UNTIL] condiz ione

con contatore: FOR contatore=valore iniziale TO valore finale [STEP incremento]
 istruz ioni
 NEXT contatore

➤ **Selezione multipla:** SELECT CASE selettore
 CASE valore1
 istruz ioni
 CASE valore2
 Istruz ioni
 ...
 CASE ELSE
 istruz ioni
 END SELECT

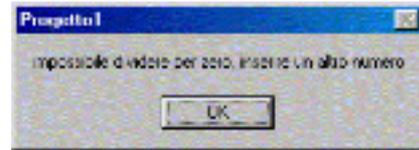
esempio di selezione: riprendiamo l'ultimo esercizio della lezione precedente. Sappiamo che non è possibile effettuare la divisione per zero; quindi se il secondo operando è zero, dobbiamo gestire l'errore per evitare il blocco del programma.

Assegniamo al pulsante relativo alla divisione (command2) un controllo sul contenuto della casella text2 :

```

Private Sub Command2_Click()
'calcola il q uoziente dei numeri inseriti nelle caselle
'di testo text1 e text2 e controlla che il secondo numero sia diverso da zero
If Val(Text t2.Text t) = 0 Then
MsgBox "impossibile dividere per z ero, inserire un altro numero"
Tex t2.SetF ocus
Else
Tex t3 .Text t = Val(Text t1 .Text t) / Val(Text t2.Text t)
End If
End Sub

```



MsgBox e' una funzione che consente l'output dei risultati: visualizza un box di dialogo (il cui testo e' scritto tra virgolette "... ..") e chiede all'utente di premere un pulsante di comando (v. figura: q uesto non e' un form e non deve essere disegnato dal programmatore).

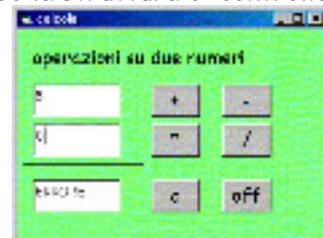
Text2.SetFocus riporta il cursore sulla casella di testo text2 per inserire un altro numero.

Il problema poteva essere risolto anche in altro modo, pur utilizzando la struttura di controllo I F.

```

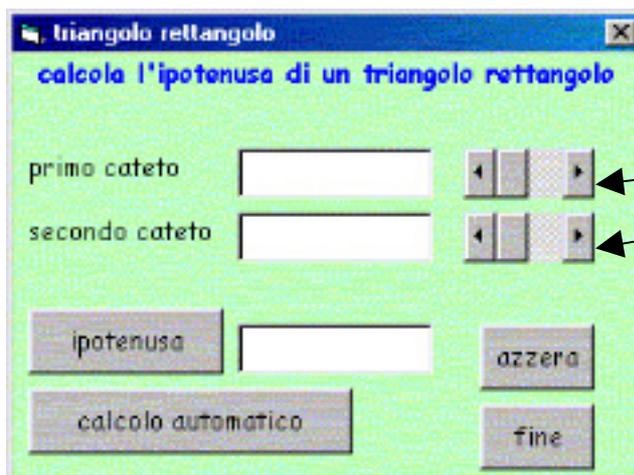
Private Sub Command2_Click()
'se il valore del secondo numero e' zero visualizza ERRO RE nella casella del risultato
If Val(Text t2.Text t) = 0 Then
Tex t3 .Text t = "ERRORE"
Tex t2.SetF ocus
Else
Tex t3 .Text t = Val(Text t1 .Text t) / Val(Text t2.Text t)
End If
End Sub

```



Esempio di ripetizione: creare un'applicazione per il calcolo dell'ipotenusa di un triangolo rettangolo; il programma offre all'utente diverse possibilita' per effettuare il calcolo:

1. inserendo i valori dei due cateti nelle apposite caselle di testo
2. scegliendo i valori dei due cateti attraverso le barre di scorrimento
3. attivando un generatore automatico di numeri interi che vengono assegnati ai due cateti



Name	Hscroll1
Name	Hscroll2

```

Private Sub Command1_Click()
'calcola l'ipotenusa q uando viene premuto il pulsante "ipotenusa"
Tex t3 .Text t = Sqr(Val(Text t1 .Text t) ^ 2 + Val(Text t2.Text t) ^ 2)
End Sub

```

```

Private Sub Command2_Click()
'esce dall'applicazione q uando viene premuto "fine"
End
End Sub

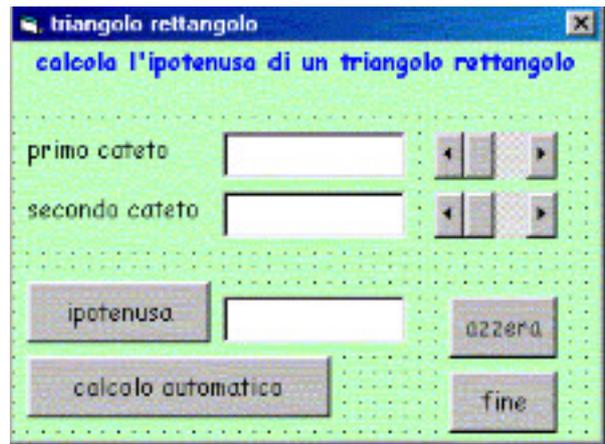
```

```
Private Sub Command3_Click()
'azzerare le caselle di testo q uando viene premuto "azzerare"
Tex t1 .Tex t = " "
Tex t2 .Tex t = " "
Tex t3 .Tex t = " "
Tex t1 .SetFocus
End Sub
```

```
Private Sub Command4_Click()
'effettua un calcolo automatico da 0 a 100 quando si preme
"calcolo automatico"
For i = 1 To 100
Tex t1 .Tex t = i
Tex t1 .Refresh
H Scroll1 .Value = i
For j = 1 To 100
Tex t2 .Tex t = j
Tex t2 .Refresh
H Scroll2 .Value = j
Tex t3 .Tex t = Sqr(i ^ 2 + j ^ 2)
Tex t3 .Refresh
Nex t j
Nex t i
End Sub
```

```
Private Sub H Scroll1_Change()
'assegna un valore alla prima casella di testo in base alla posizione
'della barra di scorrimento
Tex t1 .Tex t = H Scroll1 .Value
End Sub
```

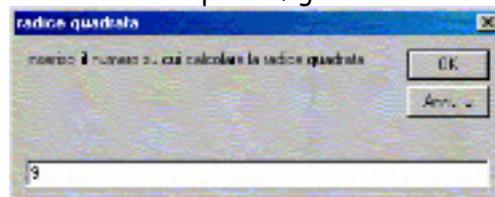
```
Private Sub H Scroll2_Change()
Tex t2 .Tex t = H Scroll2 .Value
End Sub
```



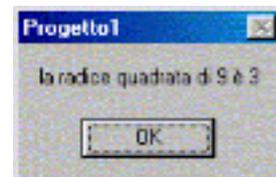
Effettivamente q uesto esempio di ciclo e' molto stupido!!!
Il metodo Refresh aggiorna il contenuto delle caselle di testo per ogni valore calcolato.

Esempio di selez ione multipla: calcolo della radice q uadrata di un numero.
Il programma accetta da tastiera un numero attraverso una finestra di input e ne calcola la radice q uadrata, intercettando le situazioni di errore (attraverso la struttura select case) che si verificano quando il numero e' negativo.

Disegniamo il form con i due pulsanti come nella prima figura:



finestra di input INPUT BOX



finestra di output MSGBOX

Le due finestre accanto al form non devono essere disegnate in quanto sono relative a funzioni definite nel codice.

si utilizzano:

➤ la funzione Inputbox, che consente l'input dei dati e in cui viene specificato il testo che deve comparire nella finestra e il titolo della finestra; la sintassi e' la seguente:

NO MEVA RI A BILE = INPUT BOX ("richiesta del dato", "testo della finestra (non obbligatorio)")

➤ la funzione Sgn, che restituisce il segno di un numero N; se assume valore 0 il segno è 0, se assume valore 1 il segno è positivo, altrimenti è negativo;

➤ Msgbox è la funzione che consente l'output attraverso una finestra di dialogo; la & serve a concatenare le stringhe; la sintassi è la seguente:
 MSGBOX "testo per la visualizzazione del dato" & NOMEVARIABILE, , "testo della finestra (non obbligatorio)"

Il codice associato ai controlli è il seguente:

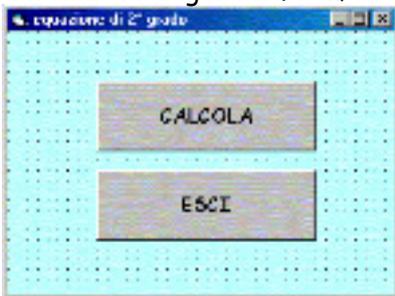
Dim n As Double 'dichiarazione della variabile nella sezione GENERALE

```
Private Sub Command1_Click()
'codice associato al pulsante "radice quadrata"
n = InputBox("inserisci il numero su cui calcolare la radice quadrata", "radice quadrata")
Select Case Sgn(n)
Case 1
MsgBox "la radice quadrata di " & n & " è " & Sqr(n)
Case 0
MsgBox "la radice quadrata di " & n & " è zero"
Case Else
MsgBox "la radice quadrata di " & n & " è immaginaria"
End Select
End Sub
```

```
Private Sub Command2_Click()
'codice associato al pulsante "uscita"
End
End Sub
```

Ora potete provare a svolgere un esercizio analogo: scrivere un'applicazione che consenta di calcolare le soluzioni di un'equazione di secondo grado.

Utilizzate il seguente form; se non ci riuscite, questa è la soluzione:



dichiarazione delle variabili nella sezione "generale"

```
Dim A As Double
Dim B As Double
Dim C As Double
Dim DELTA As Double
Dim X As Double
Dim X1 As Double
Dim X2 As Double
```

```
Private Sub CmdCalcola_Click()
A = InputBox("inserisci il 1° coefficiente", "inserimento coefficienti")
B = InputBox("inserisci il 2° coefficiente", "inserimento coefficienti")
C = InputBox("inserisci il 3° coefficiente", "inserimento coefficienti")
DELTA = B ^ 2 - 4 * A * C
Select Case Sgn(DELTA)
Case 0
X = - B / 2 * A
MsgBox "due soluzioni coincidenti; X=" & X, , "soluzione"
Case 1
X1 = (- B - Sqr(DELTA)) / 2 * A
X2 = (- B + Sqr(DELTA)) / 2 * A
MsgBox "due soluzioni X1=" & X1 & " e X2=" & X2, , "soluzione"
Case Else
MsgBox "nessuna soluzione reale", , "soluzione"
End Select
End Sub
```

```
Private Sub CmdEsci_Click()
End
End Sub
```

Effettivamente gli esempi fatti finora relativamente alle strutture di controllo sono abbastanza semplici per programmatori esperti come voi! L'intenzione, tuttavia, è quella di fornire le basi del linguaggio e la logica di lavoro, lasciando al programmatore tutti i relativi approfondimenti del linguaggio (utilizzando gli appositi testi e manuali).

Per tale motivo non tratteremo la gestione dei vettori e delle matrici e accenneremo soltanto alla gestione dei file; approfondiremo invece il trattamento delle tabelle di database e delle query in SQL all'interno del linguaggio VB.

PROGETTO 4 : Un po' di relax , Splash screen e definizione di menu'

Impariamo a gestire figure e spostamenti nel form; realizziamo un progetto contenente un'immagine in alto a sinistra, che può essere spostata utilizzando le barre di scorrimento orizzontale e verticale, per raggiungere un'altra figura in basso a destra.

Il form, con caption "aiutami..." dovrà avere questo aspetto:



I controlli presenti nel form sono:

- Prima immagine con nome Image1 (Calimero o altro) e picture a scelta
- Seconda immagine con nome Image2 (Priscilla o altro) e relativa picture
- Barra di scorrimento orizzontale (attenzione ad impostare le proprietà! provate)
- Barra di scorrimento verticale (idem)
- Pulsante con nome command1 e caption "esci"

Associamo agli oggetti il codice:

```
Private Sub Command1_Click()  
'esce dal programma  
End  
End Sub
```

```
Private Sub H Scroll1_Change()  
'assegna un valore alla posizione dell'immagine in base  
'alla posizione della barra di scorrimento orizzontale  
Image1.Left = H Scroll1.Value  
End Sub
```

```
Private Sub hscroll1_scroll()  
'rende visibile l'immagine quando si aziona la barra  
Image1.Left = H Scroll1.Value  
End Sub
```

```
Private Sub VScroll1_Change()  
Image1.Top = VScroll1.Value  
End Sub
```

```
Private Sub vscroll1_scroll()  
Image1.Top = VScroll1.Value  
End Sub
```

Ora vogliamo rendere un po' più sofisticata la nostra applicazione; avrete certamente osservato che nella maggior parte delle applicazioni che "girano" sotto Windows appare una schermata di presentazione alla partenza del software per dare informazioni sul nome del programma, sull'autore, sul numero di versione ecc. Questa schermata in genere scompare dopo un certo numero di secondi o dopo la pressione di un pulsante. In gergo informatico questa schermata iniziale si chiama Splash Screen. Quindi ora vogliamo inserire nella nostra applicazione una schermata di questo tipo.

Dal menu "progetto" scegliere "inserisci form" e selezionare "schermata iniziale". Lo splash che appare contiene diversi oggetti: teniamo l'immagine ma cambiamo la picture scegliendo l'immagine che preferiamo; cancelliamo tutti gli altri oggetti e inseriamo un timer; impostiamo le proprietà del timer (enabled : true; interval:5000; significa che l'azione associata verrà effettuata per 5000 millisecondo, circa 5 secondi).

Lo splash dovrebbe apparire così, e scriviamo il codice associato all'oggetto timer (cliccando sull'oggetto per aprire la finestra del codice):



```
Private Sub Timer1_Timer()
'chiude la splash form dopo un certo numero di millisec,rende visibile la
'form1 e disattiva il timer
Unload frmSplash
Form1.Visible = True
Timer1.Enabled = False
End Sub
```

N.B. l'oggetto TIMER non sarà più visibile in fase di esecuzione!

Ora dobbiamo aggiornare il codice associato al form1; clicchiamo sul form1 per aprire la finestra del codice e inseriamo:

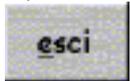
```
Private Sub Form_Load()
'rende invisibile la form1 e richiama la splash form
Form1.Visible = False
frmSplash.Show
End Sub
```

Proviamo il progetto, creiamo l'eseguibile e salviamo.

Tutte le applicazioni per Windows sono configurate in modo da poter essere utilizzate sia tramite mouse che tramite tastiera. Finora abbiamo realizzato applicazioni che venivano gestite tramite mouse, ma si può fare in modo di definire tasti di scelta rapida attraverso la pressione della lettera sottolineata.

Ad esempio, per uscire dall'applicazione precedente l'utente deve cliccare con il mouse sul tasto "esci"; ma possiamo fare in modo che l'applicazione termini premendo "E" (cioè il carattere che nel tasto appare sottolineato); per definire un tasto di questo tipo si deve inserire il carattere & (e commerciale) davanti alla lettera da usare come tasto di scelta rapida.

Quindi nell'esempio precedente modifichiamo la caption "esci", scrivendo "&esci":



L'utente avrà la possibilità di muoversi nell'applicazione utilizzando i tasti (la tabulazione per passare da un controllo ad un altro; la combinazione ALT + la lettera sottolineata).

Avremmo anche potuto definire un tasto diverso, ad esempio la S, in questo modo: "&esci".

Per definire i tasti di scelta bisogna utilizzare alcuni accorgimenti:

➤ Quando possibile usare i tasti standard di Windows (es.: la E per ESCI, la S per SALVA, la F per FILE, ecc.)

➤ Assicurarsi di non assegnare a due controlli lo stesso tasto di scelta

Un tipo di form aggiuntivo che si può usare in un progetto è la "finestra informazioni" come quella che si trova nella maggior parte dei programmi Windows nel menu ? . In genere la finestra informazioni fornisce il nome e la versione del progetto; per crearla scegliere "progetto", "inserisci form" e selezionare il form "finestra informazioni su".



Viene visualizzato il form "FrmAbout" di cui si imposta la caption, lo sfondo ecc. È possibile effettuare modifiche (modificare l'immagine inserendone una a scelta, immettere le informazioni e cancellare i controlli che non interessano come System); il controllo command contrassegnato con OK contiene già il codice per scaricare il form quando viene premuto e quindi non deve essere modificato.

Ora vedremo come attivare questa finestra!

Le applicazioni per Windows presentano in genere dei menu', contenenti dei comandi che possono essere utilizzati in sostituzione dei pulsanti di comando: ad esempio il menu' FILE, che a sua volta puo' contenere il sottomenu' A P R I , N U O V O , S T A M P A , C H I U D I ... , il menu' V I S U A L I Z Z A , ecc. Spesso le applicazioni presentano anche il menu' "?" contenente la "guida in linea" e una "finestra Informazioni".

Vogliamo quindi inserire dei menu' al di sotto della barra del titolo di un form; utilizziamo come applicativo il progetto precedente (Calimero). In particolare vogliamo creare un menu "file" contenente il sottomenu' "esci" e un menu "?" contenente "informazioni su". I menu' devono presentare una struttura utilizzando gli indici (in genere si parte da 0, ma si puo' anche partire da 1); nel nostro caso prevediamo tale struttura:

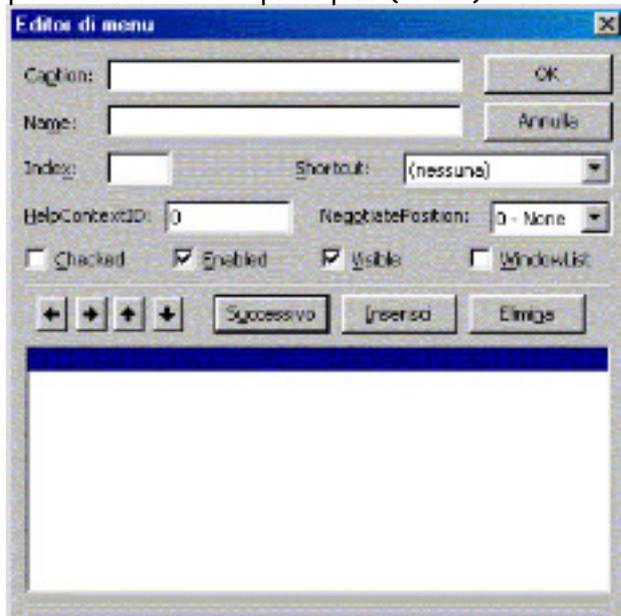
0.File (o &File, per rendere attivo il pulsante di scelta rapida attraverso il tasto F)

1.Esci (o &Esci)

1.? (o &?)

1.informazioni (o &I nformazioni)

Per impostare i menu' si utilizza la finestra EDITOR DI MENU' dal menu STRUMENTI, essendo posizionati sul form principale (form1) in cui deve essere inserito il menu'.



Procediamo in questo modo:

>Compilare la caption con "&File" e assegnare Name MnuFile; digitare 0 nella casella Index e premere su "successivo" (l'utilizzo di & permette l'uso del pulsante di scelta rapida)

>Per impostare il sottomenu' "esci" premere la freccia verso destra, che imposta il livello di rientro del menu

>Nella casella caption scrivere "&Esci" e nella casella Name "MnuEsci"; assegnare Index: 1

>Cliccare sulla freccia verso sinistra per tornare al precedente livello di menu'

>Nella caption digitare "&?" , il nome "Mnu?" e assegnare Index 2 ; premere su successivo

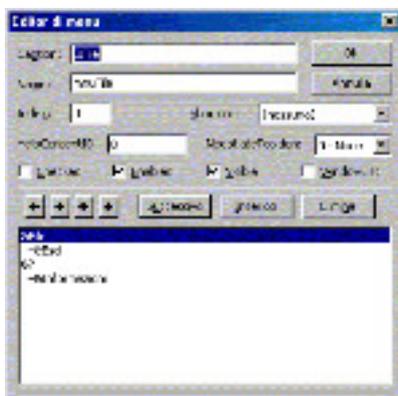
>Per impostare il sottomenu' "?" premere la freccia verso destra, che imposta il livello di rientro del menu

>Nella caption scrivere "&informazioni su", assegnare il nome "Mnuinfo" e index 1

>L'editor dovrebbe apparire come nella figura successiva

>Al termine premere OK

Nell'editor di menu' avrete senz'altro notato la casella "Shortcut" (scorciatoie): permette di assegnare ad una combinazione di tasti (es CTRL+V) una determinata azione (pensate ai menu' di Word!!!). Se volete, provate ad usarla.



Dopo aver premuto OK, il form contiene la barra dei menu' (v. figura a destra); non abbiamo finito! Dobbiamo associare ad ogni comando del menu' il relativo codice.

Con il form aperto aperto in fase di progettazione aprire il menu File e scegliere "esci"; viene visualizzata la finestra del codice relativa alla routine MnuEsci_click()



Associamo il codice che permette di terminare l'applicazione:

```
Private Sub mnuEsci_Click(Index As Integer)
End
End Sub
```

Procediamo in modo analogo per il codice relativo all'apertura della finestra informazioni:

```
Private Sub MnuInfo_Click(Index As Integer)
    frmAbout.Show
End Sub
```

(la finestra informazione e' un form di nome FrmAbout, a cui si associa il metodo Show).

Dopo aver controllato l'esecuzione dell'applicazione, create l'eseguibile e salvate l'intero progetto (naturalmente avrete creato una apposita cartella per contenere tutti i file del progetto).

Esercizio: ora potete provare a creare uno Splash screen, una finestra informazioni e dei menu' anche per altri progetti; prendete ad esempio il progetto 2.

In questo caso possiamo prevedere la seguente struttura di menu':

O.&File

1.&Esci

2.&azzera

1.&Operazioni

1.&somma

2.&ottra

3.&moltiplica

4.&dividi

3.&?

1.&informazioni

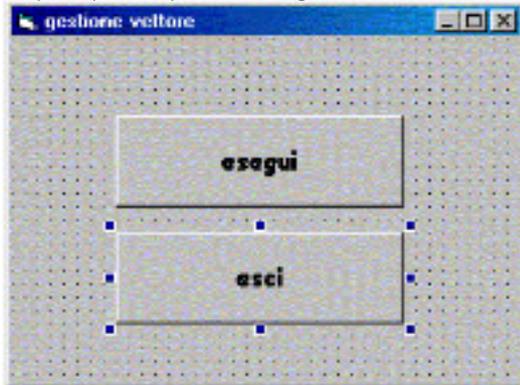
Ora che abbiamo familiarizzato con l'ambiente VB, lo utilizzeremo per la gestione dei file sequenziali (li ricordate?) anche se solo parzialmente. Il nostro obiettivo e' di utilizzare VB come interfaccia nell'uso con database; vedremo come aggiornare e visualizzare i dati di tabelle create con un DBMS come Microsoft access e come visualizzare i risultati di interrogazioni SQL.

PROGETTO 5: gestione dei vettori

La gestione dei vettori e delle matrici puo' essere realizzata in diversi modi; in ogni caso si deve fare riferimento ai concetti teorici, alla programmazione strutturata e alla scomposizione top-down.

Esempio 1: caricare in un vettore N numeri interi (max 10) e visualizzarli

Si puo' predisporre il seguente form, associando agli oggetti il codice:



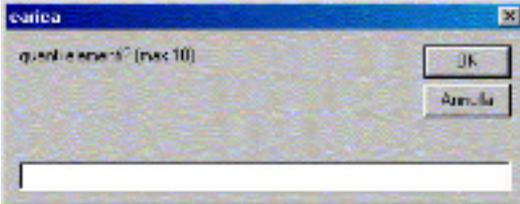
```
Dim vet(0 To 10) As Integer
Dim I As Integer
Dim N As Integer

Private Sub Cmdesci_Click()
'consente di uscire dall'applicazione
End
End Sub
```

```
Private Sub carica()
'carica max 10 numeri in un vettore
Do
N = InputBox ("quanti elementi? (max 10)", "carica")
Loop Until N <= 10
I = 0
Do Until I = N
I = I + 1
vet(I) = InputBox ("inserire il numero ", "carica")
Loop
End Sub
```

durante il funzionamento appariranno le seguenti finestre:

>per la richiesta del numero di elementi:



```
Private Sub stampa()
'visualizza gli elementi del vettore
I = 0
Do Until I = N
I = I + 1
Print vet(I)
Loop
End Sub
```

>per inserire ogni numero:



```
Private Sub Cmdesegui_Click()
'programma principale
carica
stampa
End Sub
```

>per visualizzare il risultato:

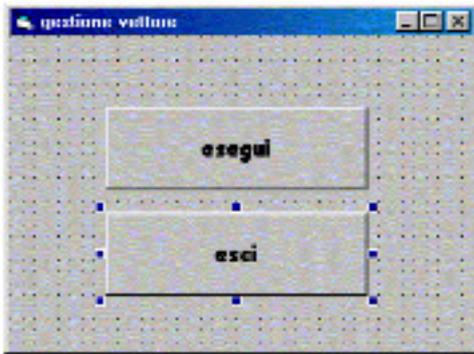


Esempio 2: caricare in un vettore N numeri (max 10) e comunicare la media

Si puo' predisporre lo stesso form del progetto precedente; per l'input dei dati si utilizza la INPUT BOX e per l'output (essendo un solo valore) si puo' usare la MSGBOX:

dichiarazione delle variabili in "generale":

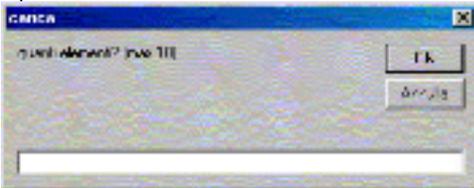
```
Dim vet(0 To 10) As Integer
Dim I As Integer
Dim N As Integer
Dim ACC As Long
Dim MEDIA As Single
```



```
Private Sub Cmdesci_Click()
End
End Sub

Private Sub carica()
'carica max 10 numeri in un vettore
Do
N = InputBox ("quanti elementi? (max 10)", "carica")
Loop Until N <= 10
For I = 1 To N
vet(I) = InputBox ("inserire il numero ", "carica")
Next I
End Sub
```

>per definire il numero di elementi del vettore



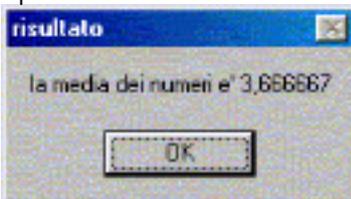
```
Private Sub calcola()
'somma i numeri inseriti nel vettore e calcola la media
ACC = 0
For I = 1 To N
ACC = ACC + vet(I)
Next I
MEDIA = ACC / N
MsgBox "la media dei numeri e' " & MEDIA, "risultato"
End Sub
```

>per caricare gli N elementi:

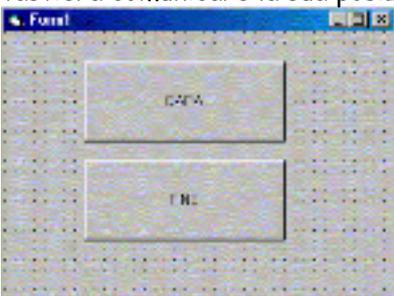


```
Private Sub Cmdesegui_Click()
'programma principale
carica
calcola
End Sub
```

>per visualizzare il risultato:



Esempio 3 : caricare in un vettore i nomi di n partecipanti ad una gara (max 10); inserendo un nome da tastiera comunicare la sua posizione.

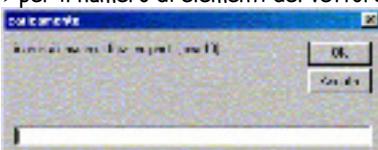


```
Private Sub Command1_Click()
'principale
carica
ricerca
End Sub

Private Sub Command2_Click()
End
End Sub
```

durante l'esecuzione vengono visualizzate le seguenti finestre:

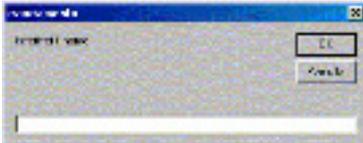
>per il numero di elementi del vettore:



```
Dichiarazione delle variabili:
Dim NOME(1 To 10) As String
Dim I As Integer
Dim N As Integer
Dim CERCA As String
Dim TROVATO As Integer
```

```
Sub carica()
'carica il vettore
Do
N = InputBox ("inserisci il numero di partecipanti (max 10)", "caricamento")
```

>per l'inserimento dei nomi:

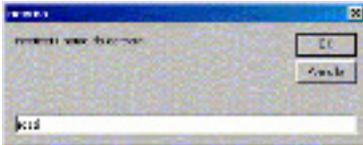


```

Loop Until N <= 10
For I = 1 To N
    NOME(I) = InputBox ("inserisci il nome", "caricamento")
Next I
End Sub

```

>per l'inserimento del nome da cercare:

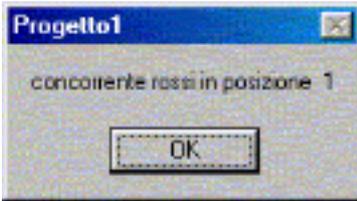


```

Sub ricerca()
CERCA = InputBox ("inserisci il nome da cercare", "ricerca")
TROVATO = 0
I = 0
Do
    I = I + 1
    If NOME(I) = CERCA Then
        TROVATO = 1
        stringa = "concorrente " + CERCA + "in posizione " + Str(I)
        MsgBox (stringa)
    End If
Loop Until (I = N) Or (TROVATO = 1)
If TROVATO = 0 Then
    MsgBox ("concorrente inesistente")
End If
End Sub

```

>per visualizzare la posizione se il nome viene trovato:



>per comunicare che il nome inserito e' inesistente:



PROGETTO 6: gestione dei file

I file di dati li conosciamo già: li abbiamo trattati utilizzando un linguaggio procedurale adatto alle applicazioni gestionali (COBOL); di conseguenza presumo che vi ricordiate quali sono le organizzazioni dei file (sequenziali e a indici) e le operazioni che possono essere effettuate su un file. Vediamo quindi solo alcuni semplici esempi per gestire file di dati in VB.

Esercizio 1: realizziamo un'applicazione che permetta di inserire dati in un file sequenziale e di leggerli.

Create una cartella PROGETTO5 e una sottocartella FILE, in cui verranno inseriti tutti i file del progetto.

L'applicazione è costituita da tre form:

form iniziale

name:frminizio

caption: principale

contiene due pulsanti di opzione, due etichette e un pulsante

Form per l'inserimento

Name: form1

Caption:gestione di file sequenziale

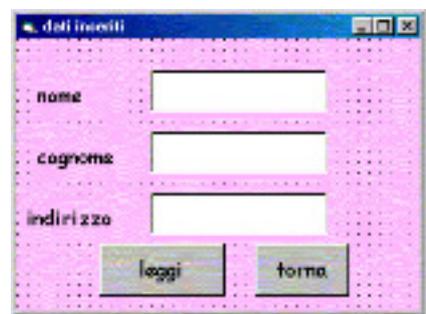
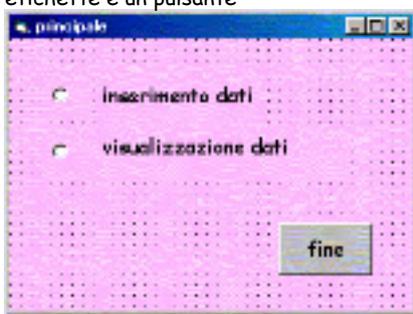
Contiene tre etichette, tre caselle di testo e due pulsanti

Form per la visualizzazione

Name: form2

Caption:dati inseriti

Contiene gli stessi elementi del form1



N.B. finora abbiamo utilizzato i diversi oggetti (form, etichette, pulsanti...) dell'applicazione VB con i nomi di default; ora che conosciamo meglio il linguaggio possiamo abituarci a definire nomi significativi per gli oggetti; nel nostro esempio il form (schermata principale) viene chiamato FRMINIZIO e i due form successivi vengono rinominati FORM1 e FORM2. In modo analogo abbiamo rinominato i pulsanti: il pulsante con la caption "fine" è stato nominato Cmdfine, il pulsante con caption "inserisci" è Cmdinserisci; le caselle di testo si chiamano "nome", "cognome", "indirizzo" (come i campi che contengono)

Dopo aver disegnato l'interfaccia dobbiamo associare il codice ai diversi controlli; per quanto riguarda il form iniziale (frminizio), non c'è nulla di nuovo (v. progetto 1):

```
Private Sub Cmdfine_Click()
```

```
'termina l'applicazione
```

```
End
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
'controlla il caricamento del form di inizio
```

```
Option1.Value = False
```

```
Option2.Value = False
```

```
End Sub
```

```
Private Sub Option1_Click()
```

```
'visualizza il form1 per inserimento dati
```

```
Form1.Show
```

```
End Sub
```

```
Private Sub Option2_Click()
```

```
'visualizza il form2 per visualizzazione dati
```

```
Form2.Show
```

```
Form2.Cmdleggi.setFocus 'porta il cursore sul pulsante Cmdleggi
```

```
'poiche' Cmdleggi appartiene al form2 deve essere specificato con form2!
```

```
End Sub
```

Per quanto riguarda il form1 e form2 dobbiamo considerare le operazioni relative al trattamento dei file sequenziali.

Per poter essere utilizzato, un file deve essere aperto; l'apertura può essere in OUTPUT (per la creazione), APPEND (per aggiungere dati in fondo al file; equivale EXTEND in Cobol) e INPUT (per leggere i dati); per inserire i dati utilizziamo la modalità APPEND, perché in VB può essere usata anche per creare il file, se ancora non esiste. L'istruzione di apertura è: OPEN "NOMEFILE.DAT" FOR [OUTPUT | APPEND | INPUT] AS #N (dove #N è il numero del buffer di memoria assegnato al file). L'istruzione WRITE #N o PRINT #N consente di scrivere i dati nel file sequenziale; INPUT #N è l'istruzione per la lettura.

La chiusura del file viene realizzata con l'istruzione CLOSE #N.

Associamo al form1 e ai relativi oggetti il seguente codice:

```
Private Sub Form_Load()  
'apre il file sequenziale per creazione e aggiunta dati; deve essere specificato il percorso  
Open "a:\progetto5\file\anagrafe.dat" For Append As #1  
End Sub
```

```
Private Sub CmdInserisci_Click()  
'scrive i dati nel file  
Write #1, nome.Text 'o PRINT #1  
Write #1, cognome.Text  
Write #1, indirizzo.Text  
nome.Text = "  
cognome.Text = "  
indirizzo.Text = "  
CmdInserisci.SetFocus 'dopo aver premuto tab porta il cursore sul pulsante inserisci, per memorizzare i dati  
End Sub
```

```
Private Sub Cmdesci_Click()  
'chiude il file e scarica il form1  
Close #1  
Unload Me  
End Sub
```

Il codice associato al form2 e relativi oggetti differisce per le istruzioni di lettura:

```
Private Sub Form_Load()  
'apre il file per la lettura  
Open "a:\progetto5\file\anagrafe.dat" For Input As #1  
End Sub
```

```
Private Sub Cmdleggi_Click()  
'consente la lettura del file assegnando i campi alle caselle di testo  
Dim varnome As String, varcognome As String, varindirizzo As String  
If Not EOF(1) Then  
    Input #1, varnome, varcognome, varindirizzo  
    nome.Text = varnome  
    cognome.Text = varcognome  
    indirizzo.Text = varindirizzo  
    Cmdleggi.SetFocus 'il pulsante leggi deve essere premuto per leggere il successivo record  
Else  
    MsgBox "non ci sono altri dati", , Form2.Caption  
'Form2.Caption inverte il msgbox con la caption di form2; si può omettere  
    Cmdtorna.SetFocus  
End If  
End Sub
```

```
Private Sub Cmdtorna_Click()  
'chiude il file e scarica il form2  
Close #1  
Unload Me  
End Sub
```

Questo progetto potrebbe essere completato con una Splash form, una finestra informazioni e un menu sotto la barra del titolo; provate!

Abbiamo trattato solo la scrittura e la lettura dei record di un semplice file sequenziale (in cui i record vengono letti sempre a partire dal primo), ma sappiamo che le operazioni possono anche essere di altro tipo (modifica, ricerca, elaborazioni varie); esistono inoltre anche i file ad accesso diretto, che offrono maggiore velocità ed efficienza nella gestione. Se avrete occasione di doverli gestire, queste conoscenze di base e, soprattutto, l'utilizzo di un buon manuale vi potrà essere di aiuto!

Esercizio 2: (molto carino) realizzare un'applicazione che simuli un programma di elaborazione testi (ad esempio il blocco note di Windows), per gestire testi in formato .txt. Tale programma permette di aprire un nuovo file, aprire un file già esistente, di salvarne uno nuovo e di uscire dal programma stesso; inoltre permette anche di tagliare o copiare testo in un nuovo file.

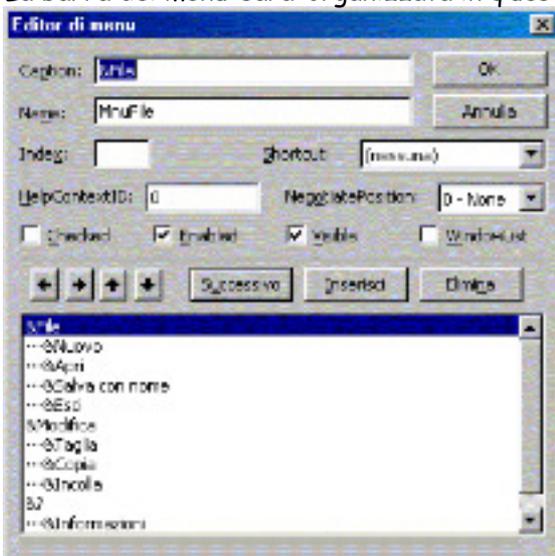
Il file in cui viene scritto il testo è un file di dati perché dovrà essere aperto (in output per permettere la memorizzazione, in input per la lettura) e chiuso al termine delle elaborazioni. Il nostro progetto è costituito dal form principale (form1), da uno splash form (frmsplash; se volete, sapete già come gestirlo) e da una finestra informazioni (frmabout).

Nel form1 inseriamo i seguenti oggetti:



- Una casella di testo adatta a contenere più righe: name text1; multiline true (consente l'inserimento di testo di grandi dimensioni su più righe); scrollbars both (produce l'attivazione automatica delle barre di scorrimento orizzontali e verticali quando il testo eccede le dimensioni della finestra)
- Un controllo COMMON DIALOG per le finestre di dialogo comuni; è un controllo standard che consente di visualizzare finestre di dialogo di uso comune nelle applicazioni Windows (quali la finestra Apri, Salva con nome, ecc.); se non è presente nella casella degli strumenti, può essere aggiunto scegliendo "componenti" dall'elenco che si apre cliccando con il tasto destro sulla casella stessa e mettendo il segno di spunta sul controllo stesso (Microsoft Common Dialog Control). Il controllo deve essere inserito in un punto qualsiasi del form, ma rimarrà nascosto all'utente durante l'esecuzione dell'applicazione.
- Una barra dei menu (che sapete costruire)

La barra dei menu sarà organizzata in questo modo:



CAPTION	NAME	INDEX
&File	MnuFile	
&Nuovo	MnuFileitem	0
&A pri	MnuFileitem	1
&Salva con nome	MnuFileitem	2
&Esci	MnuFileitem	3
&Modifica	MnuModificaItem	
&T aglia	MnuModificaItem	0
&Copia	MnuModificaItem	1
&I ncolla	MnuModificaItem	2
&?	MnuInfo	
&I nformazione	MnuInfoitem	0

Dopo aver definito i menu (v. progetto 4) occorre associare ad ogni voce e sottovoce il codice delle routine che devono essere eseguite quando l'utente fa una scelta nei menu.

Le routine utilizzano i seguenti nomi per gli oggetti: Form1 per il form, Text1 per la casella di testo e CFileDialog1 per il controllo delle finestre di dialogo comuni (poiche' in fase di disegno degli oggetti abbiamo lasciato i nomi di default, non li modifichiamo; ma avremmo anche potuto cambiarli!...).

Per aprire la finestra del codice si clicca su una voce di menu'.

Le routine utilizzano variabile di lavoro Filename che dichiariamo nella parte "generale" delle finestra del codice e contiene il nome del file utilizzato:



Ora associamo il seguente codice al menu "File", utilizzando la selezione multipla per controllare l'indice associato ai sottomenu' (i commenti al sorgente sono preceduti da '):

```
Private Sub MnuFileItem_Click(Index As Integer)
Select Case Index
'Index e' il selettore per controllare la voce di menu' selezionata
'corrisponde all'indice assegnato nell'editor di menu'
Case 0
'NUOVO
Text1.Text = "" 'svuota la casella di testo
filename = "senza nome"
Form1.Caption = "trattamento di testi:" & filename 'assegna alla finestra il nome del file
Case 1
'APRI
CFileDialog1.Filter = "tutti i file (*.*)|*.*|file di testo(*.txt)|*.txt|"
CFileDialog1.FilterIndex = 2 'visualizza i file *.* e *.txt
CFileDialog1.ShowOpen 'visualizza la finestra di dialogo APRI
filename = CFileDialog1.filename
Open filename For Input As #1 'apre il file
Form1.Caption = "trattamento di testi" & filename 'assegna alla finestra il nome del file aperto
Text1.Text = Input(LOF(1), 1) 'legge il contenuto del file e lo assegna alla casella di testo
Close #1 'chiude il file
Case 2
'SALVA CON NOME
CFileDialog1.Filter = "tutti i file (*.*)|*.*|file di testo (*.txt)|*.txt|"
CFileDialog1.FilterIndex = 2
CFileDialog1.ShowSave 'visualizza la finestra di dialogo SALVA con nome
filename = CFileDialog1.filename
Open filename For Output As #1 'apre il file per l'output
Print #1, Text1.Text 'scrive il testo nel file aperto; si puo' usare anche WRITE
Close #1 'chiude il file
Case 3
'ESCI
End
End Select
End Sub
```

In questa routine Filename e' il nome del file di testo che viene creato o modificato.

La gestione del file fisico su disco e' realizzata con le istruzioni OPEN per l'apertura, CLOSE per la chiusura, INPUT per la lettura e PRINT per la scrittura; la funzione LOF restituisce la lunghezza in byte del file aperto e #1 indica il numero del buffer di I/O (ricordate?) utilizzato per le operazioni sul file.

Al controllo CFileDialog1 vengono associati i seguenti metodi:

- SHOWOPEN: per aprire la finestra di dialogo "A pri" delle applicazioni Windows
- SHOWSAVE: per aprire la finestra di dialogo "salva con nome"
- FILTERINDEX: imposta l'elenco dei tipi di file nelle finestre di dialogo "A pri" e "salva con nome".

Analogamente associamo il seguente codice al menu "Modifica":

```
Private Sub MnuModificaItem_Click(Index As Integer)
Select Case Index
'Index e' il selettore per controllare la voce di menu' selezionata
'corrisponde all'indice assegnato nell'editor di menu'
```

```

Case 0      'e' stato scelto T A GLI A
Clipboard.Clear
Clipboard.SetText Text1.SelText      'copia il testo selezionato
Text1.SelText = " "                  'cancella il testo selezionato
Case 1      'e' stato scelto COP I A
Clipboard.Clear
Clipboard.SetText Text1.SelText      'copia il testo selezionato negli appunti
Case 2      'e' stato scelto INCOLLA
Text1.Text = Text1.Text + Clipboard.GetText() 'incolla il testo dagli appunti nel punto desiderato o in nuovo doc.
End Select
End Sub

```

Le scelte "taglia, copia, incolla" utilizzano l'oggetto `CLIPBOARD`, che rappresenta gli `APPUNTI` di `Windows`, e nel quale vengono conservati i caratteri selezionati e copiati/tagliati.

Per la gestione del menu "Modifica" e' necessaria una routine per controllare la presenza di testo selezionato; cio' significa che, se non e' stato selezionato testo, le opzioni `TAGLIA` (voce di menu' 0) e `COPIA` (voce di menu' 1) non devono essere attive (devono apparire in grigio chiaro perche' non possono essere scelte).

Si utilizza la proprieta' `ENABLED` del menu (che puo' assumere valore `true` o `false`).

La proprieta' della casella di testo `SELLENGTH` contiene la lunghezza del testo selezionato; se non e' stato selezionato testo, la condizione `TEXT.SELLENGHT > 0` restituisce valore `FALSE` e quindi rende inabilitata la voce di menu corrispondente.

La routine associata al menu modifica e' quindi la seguente:

```

Private Sub MnuModifica_click()
'disattiva taglia e copia se non c'e testo selezionato
MnuModificaItem(0).Enabled = (Text1.SelLength > 0)
MnuModificaItem(1).Enabled = (Text1.SelLength > 0)
End Sub

```

Associamo alla terza voce di menu (?) il seguente codice:

```

Private Sub MnuInfoitem_Click(Index As Integer)
'visualizza la finestra informazioni
frmAbout.Show
End Sub

```

Naturalmente dovete aver creato la finestra informazioni (`frmabout`)!

Se avete creato anche lo splash screen, associate il relativo codice (vi ricordate come si fa? v. progetto 4)

Ora e' il momento di provare l'applicazione; carina vero?

Create l'eseguibile e salvate. Questo e' il vostro blocco notes personale!

PROGETTO 7: gestione di tabelle di database

E' possibile utilizzare VB per scrivere progetti che consentono di visualizzare, modificare e inserire informazioni in un database relazionale.



Per poter gestire le tabelle di una database (es. Access) e' necessario utilizzare il controllo DATA presente nella casella degli strumenti

Primo esempio: Realizziamo un'applicazione che consente di visualizzare cognome e nome dei cittadini. (consiglio: create una cartella PROGETTO6, in cui inserirete le sottocartelle relative agli esempi che svilupperemo).

Innanzitutto occorre creare il database ANAGRAFE.MDB con Access (attenzione: VB 5.0 supporta versione di Access fino alla '97, quindi se create il database con Access 2000 salvatelo nella versione '97); il database contiene la tabella CITTADINO (COD, COGNOME, NOME, SESSO, ...).

Disegniamo l'interfaccia dell'applicazione:



Essa contiene :

- > Il form1 con caption "visualizzazione dati anagrafici"
- > Due etichette (label1, label1) rispettivamente con caption "cognome" e "nome"
- > Due caselle di testo a cui assegniamo name COGNOME e NOME e text azzerato
- > Un controllo data1
- > Un pulsante di comando con caption "esci" a cui verra' associato il codice per uscire dall'applicazione

Ora occorre assegnare le seguenti proprietà al controllo DATA 1:

- > proprietà CONNECT : si seleziona il tipo di database a cui connettersi; VB propone Access di default, ma se il database e' diverso (DBIII o altro) deve essere specificato il nome del database corrispondente
- > proprietà DATABASENAME: si specifica il percorso e il nome del database, corrispondente al file di Access con estensione .mdb (nel nostro caso ANAGRAFE.MDB); per specificare correttamente il percorso, premere sul pulsante con i puntini e scegliere
- > proprietà RECORDSOURCE: si specifica il nome della tabella contenuta nel database; aprendo il menu' a tendina vengono visualizzate le tabelle disponibili, nel nostro caso CITTADINO; (il recordsource puo' essere anche una query del database o un comando select di SQL)
- > Possiamo anche impostare la caption del controllo al nome della tabella di cui vengono visualizzati i dati e scegliere il font di caratteri desiderato.

Le caselle di testo devono visualizzare i dati della tabella; dobbiamo quindi assegnare le seguenti proprietà:

PRIMA CASSELLA DI TESTO	DI COGNOME	SECONDA CASSELLA DI TESTO	DI NOME
Name:	cognome	Name:	nome
DataSource	Data1	DataSource	Data1
DataField	COGNOME	Datafield	NOME

DATA SOURCE indica il controllo DATA a cui il campo si riferisce

DATA FIELD indica il nome del campo da associare alla casella di testo; si puo' scegliere il campo desiderato nella casella combinata, in cui viene mostrato l'elenco dei campi disponibile, se e' gia' stata definita la proprietà DataSource.

Possiamo impostare anche il font dei caratteri per entrambe le caselle di testo.

Assegniamo al pulsante Command1 la routine per uscire dall'applicazione (e' l'unica riga di codice che dovete scrivere!); proviamo ad eseguirla e, se funziona correttamente, creiamo il file eseguibile.



Quando viene eseguita l'applicazione, viene visualizzato il primo record della tabella; le frecce presenti nel controllo DATA permettono la navigazione tra i record.

Ora potete provare a visualizzare i dati provenienti anche da tabelle o query di altri database!

Secondo esempio: presentare i dati della tabella CITTADINO in forma tabellare.

Il controllo aggiuntivo DBGRID consente di visualizzare i dati della tabella secondo una griglia di righe e colonne (come nella modalita' "visualizzazione foglio dati" di Access).



Di solito tale controllo non e' presente nella casella degli strumenti, ma ora sappiamo come si fa ad aggiungerlo (scegliere Microsoft Data Bound Grid Control).

Disegnate un form con i seguenti controlli:

➤ Una LABEL con caption "archivio cittadini"

➤ Un controllo DATA1 con le proprieta' viste nell'esempio precedente; tale controllo serve solo per definire la tabella di riferimento per il controllo DBGrid e puo' restare nascosta durante l'esecuzione, impostandone la proprieta' Visibile a False

➤ Un pulsante di comando COMMAND1 a cui verra' associato il codice per uscire dall'applicazione

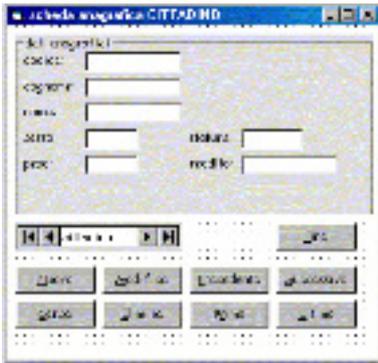
➤ un controllo DBGRID con nome DBGRID1, stabilendo le dimensioni adeguate con il trascinarsi del mouse; tale controllo deve essere associato al controllo DATA gia' presente nel form: si definisce la proprieta' DATA SOURCE: Data1.



Eseguendo l'applicazione si visualizzano i dati contenuti nella tabella CITTADINO secondo una griglia di righe e colonne, tra le quali ci si puo' muovere con i tasti freccia, con il mouse o con le barre di scorrimento; con questa modalita' di rappresentazioni dei dati sono ammesse modifiche sui valori dei campi, ma non e' possibile inserire o eliminare record.

Terzo esempio: realizziamo un'applicazione che consenta di effettuare le operazioni di manipolazione (inserimento, modifica, cancellazione e ricerca) e di navigazione sui dati di una tabella con apposite istruzioni scritte in codice VB.

Disegniamo un form (form1 con caption "scheda anagrafica cittadino") con i seguenti controlli:



- > un frame con caption "dati anagrafici"
- > sette etichette con caption "codice", "cognome", "nome", ...
- > sette caselle di testo (text1, text2, ...) con proprieta' text azzerata
- > un controllo DATA1 impostandone le proprieta' per collegarlo al database ANAGRAFE e alla tabella CITTADINO; tale controllo serve solo per definire il database da utilizzare e puo' restare nascosto durante l'esecuzione, impostandone la proprieta' Visibile a False
- > nove pulsanti di comando ("precedente", "successivo", "primo" e "ultimo" per la navigazione nel database; "nuovo" per aggiungere un record; "modifica" per modificare i dati; "cerca" per ricercare un record in base al cognome; "elimina" per cancellare il record corrente; "esci" per terminare l'applicazione)

Nel definire la caption di ogni pulsante potete usare il simbolo & per definire i tasti di scelta rapida (v. progetto4); inoltre e' sempre possibile definire una barra dei menu' che consenta di svolgere le stesse operazioni previste con i pulsanti.

Ora associamo il codice VB; nella sezione "generale" dichiariamo le seguenti variabili:

```
Dim CONT As Integer      'contatore che consente di controllare se il database ' vuoto o no
Dim TROVATO As Integer  'deviatore che controlla se il record cercato esiste
Dim CERCA As String     'variabile che conterra' il cognome per la ricerca del record
```

Le operazioni di manipolazione sulla tabella (inserimento, modifica e cancellazione) sono rappresentate dai metodi ADDNEW, EDIT e DELETE; inoltre le operazioni di inserimento e modifica devono essere seguite dal metodo UPDATE (ed e' opportuno anche aggiornare il contenuto della tabella al controllo DATA1 con il metodo REFRESH); la navigazione tra i record viene realizzata con i metodi MOVEFIRST (primo), MOVELAST (ultimo), MOVENEXT (successivo) e MOVEPREVIOUS (precedente), prevedendo opportuni controlli quando si e' posizionati sul primo (BOF= beginning of file) o ultimo record (EOF= end of file).

Ai controlli quindi associamo il seguente codice:

```
Private Sub Form_Load()
'all'esecuzione controlla che ci siano record in archivio
Data1.Refresh
CONT = 0
If Not Data1.Recordset.EOF Then
CONT = Data1.Recordset.RecordCount
End If
If CONT = 0 Then
MsgBox "archivio vuoto"
End If
End Sub

Private Sub Command1_Click()
'inserisce un nuovo record
Data1.Recordset.AddNew 'aggiunge un nuovo record
Data1.Recordset.Update 'registra il nuovo record
Data1.Recordset.MoveLast 'si posiziona sul record vuoto per
End Sub 'consentire l'inserimento dei
dati
```



```
Private Sub Command2_Click()
'modifica un record
If CONT = 0 Then
MsgBox "archivio vuoto, impossibile modificare"
Else: MsgBox "posizionarsi sul record da modificare"
Data1.Recordset.Edit 'predispone per la scrittura
Data1.Recordset.Update 'registra il record
End If
End Sub
```

```
Private Sub Command3_Click()
'ricerca sequenziale in base al cognome
```

```

CERCA = InputBox ("inserire cognome:", "ricerca")
Data1 .Recordset.MoveF irst 'si posiziona sul primo record
TROVATO = 0
Do 'ripete per ogni record
  If CERCA = Data1 .Recordset("cognome") Then
    TROVATO = 1
  Else: Data1 .Recordset.MoveNex t 'passa al record successivo
  End If
Loop Until (TROVATO = 1 ) Or (Data1 .Recordset.EOF )
If TROVATO = 0 Then
  MsgBox "record inesistente", , "ricerca" 'instesta il msgbox
End If
End Sub

Private Sub Command4 _Click()
'elimina un record
Data1 .Recordset.Delete 'cancella il record
Data1 .Refresh 'aggiorna il contenuto della tabella
Data1 .Recordset.MoveLast 'si posiziona sull'ultimo record memorizzato
End Sub

Private Sub Command5_Click()
'si posiziona sul record precedente, controllando l'inizio del file BO F
Data1 .Recordset.MovePrevious
If Data1 .Recordset.BOF Then
  Data1 .Recordset.MoveF irst
  MsgBox "non ci sono altri record precedenti", , "record"
End If
End Sub

Private Sub Command6_Click()
'si posiziona sul record successivo, controllando la fine del file EO F
Data1 .Recordset.MoveNex t
If Data1 .Recordset.EOF Then
  Data1 .Recordset.MoveLast
  MsgBox "non ci sono altri record successivi", , "record"
End If
End Sub

Private Sub Command7_Click()
'permette il posizionamento sul primo record
Data1 .Recordset.MoveF irst
End Sub

Private Sub Command8_Click()
'permette il posizionamento sull'ultimo record
Data1 .Recordset.MoveLast
End Sub

Private Sub Command9_Click()
'esce dall'applicazione
End
End Sub

```

PROGETTO 8: VB e SQL

Negli esempi precedenti l'insieme dei dati che venivano associati al controllo `DATA` provenivano da una tabella del database, il cui nome era specificato nella proprietà `recordsource`; in realtà l'insieme dei dati che possono essere associati al controllo `DATA` possono anche risultare da un'interrogazione `SQL` scritta all'interno del codice VB associato ad un controllo (es. pulsante).

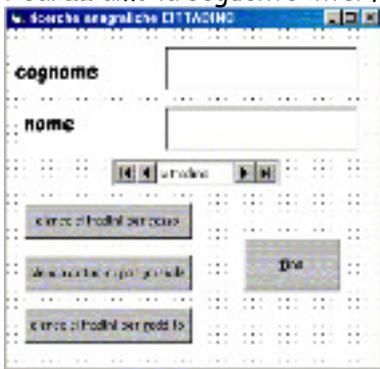
Esempio: data la tabella `CITTA DINO` del database `A N A G R A F E`, realizziamo un progetto che consenta di realizzare tre tipi di interrogazioni:

1. elenco dei cittadini di un certo sesso
2. elenco dei cittadini il cui cognome inizia con una certa lettera
3. elenco dei cittadini con reddito superiore ad un certo valore

visualizzando cognome e nome dei cittadini che corrispondono alle caratteristiche richieste.

Sappiamo come realizzare queste istruzioni in `SQL`; ora vediamo come associarle al codice VB.

Realizziamo la seguente interfaccia:



Con i seguenti controlli:

- un controllo `DATA1` con caption "cittadino", `DATA BASE NAME` associato al file di database su disco (specificare il percorso) e `RECORD SOURCE` associato a `CITTA DINO`
- due etichette con caption `COGNOME`, `NOME`
- due caselle di testo con `text` azzerata e `DATA FIELD` associata rispettivamente a `COGNOME` e `NOME`
- quattro pulsanti di comando in cui inserire una opportuna caption ("elenco cittadini...", "fine") prevedendo i tasti di scelta rapida

Ora associamo ai quattro pulsanti il seguente codice:

'dichiarazione delle variabili nella sezione "generale"

```
Dim MIOSQL As String      'stringa di codice SQL
Dim CERCAS As String      'variabile contenente il sesso da cercare
Dim INIZIALE As String    'variabile contenente l'iniziale del cognome da cercare
Dim CONT As Integer       'contatore dei record
Dim VALORE As Double      'variabile contenente il valore minimo di reddito
```

```
Private Sub Command1_Click()
'ricerca i cittadini di un certo sesso
CERCAS = InputBox ("sesso:", "ricerca") 'visualizza un box per inserire il sesso da cercare
MIOSQL = " " 'istruzione SQL
MIOSQL = MIOSQL + "SELECT COGNOME, NOME "
MIOSQL = MIOSQL + "FROM CITTADINO "
MIOSQL = MIOSQL + "WHERE SESSO= '" + CERCAS + "' "
MIOSQL = MIOSQL + "ORDER BY COGNOME, NOME;"
Data1.RecordSource = MIOSQL 'associa alla proprietà il codice SQL
Data1.Refresh 'aggiorna l'origine dei dati associata a DATA1
CONT = 0 'controlla che ci sia corrispondenza nella ricerca
If Not Data1.Recordset.EOF Then
CONT = Data1.Recordset.RecordCount
End If
If CONT = 0 Then
MsgBox "nessun record trovato", 16, "risultato" 'visualizza un messaggio di tipo 16 se non ci sono record
End If
End Sub
```

```
Private Sub Command2_Click()
'ricerca i cittadini con cognome avente una certa iniziale
INIZIALE = InputBox ("iniziale del cognome:", "ricerca")
MIOSQL = " "
MIOSQL = MIOSQL + "SELECT COGNOME, NOME "
MIOSQL = MIOSQL + "FROM CITTADINO "
```

```

MIOSQ L = MIOSQ L + "W H ERE COGNOME LIKE '" + INIZIALE + "' & '*' "
MIOSQ L = MIOSQ L + "ORDER BY COGNOME, NOME;"
Data1 .RecordSource = MIOSQ L
Data1 .Refresh
CONT = 0
If Not Data1 .Recordset.EOF Then
    CONT = Data1 .Recordset.RecordCount
End If
If CONT = 0 Then
    MsgBox "nessun record trovato", 16, "risultato"
End If
End Sub

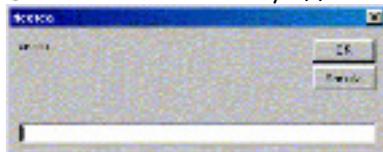
Private Sub Command3_Click()
'ricerca dei cittadini con reddito superiore ad un certo valore
VALORE = InputBox ("valore minimo di reddito:", "ricerca")
MIOSQ L = " "
MIOSQ L = MIOSQ L + "SELECT COGNOME, NOME "
MIOSQ L = MIOSQ L + "FROM CITTADINO "
MIOSQ L = MIOSQ L + "W H ERE REDDITO >" + Str(VALORE) + " "
MIOSQ L = MIOSQ L + "ORDER BY COGNOME, NOME;"
Data1 .RecordSource = MIOSQ L
Data1 .Refresh
CONT = 0
If Not Data1 .Recordset.EOF Then
    CONT = Data1 .Recordset.RecordCount
End If
If CONT = 0 Then
    MsgBox "nessun record trovato", 16, "risultato"
End If
End Sub

Private Sub Command4_Click()
'esce dall'applicazione
End
End Sub

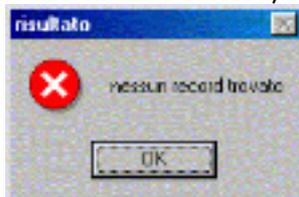
```

Le istruzioni SQL sono costruite su piu' righe per comodita' di lettura e devono rispettare la sintassi, compresi gli spazi e i punti e virgola (infatti al termine di ogni riga, prima di " c'e' uno spazio). Il Valore del codice inserito da tastiera viene concatenato alla stringa SQL con l'operatore + (normalmente tale parametro apparirebbe tra parentesi quadre nelle istruzioni [...])

Durante l'esecuzione, effettuando la ricerca sul sesso, appaiono le seguenti finestre:



per l'inserimento del sesso (M o F) da cercare



questo messaggio appare se viene inserito un valore assurdo



Se invece esistono dei record corrispondenti, viene visualizzato il primo (possibile visualizzare tutti gli altri utilizzando i pulsanti del controllo D A T A 1):

Ora potete provare a realizzare altre applicazioni con diverse istruzioni SQL, utilizzando anche altri database (ne abbiamo realizzati tantissimi; avete solo l'imbarazzo della scelta!). Provate ad utilizzare tutti i controlli che abbiamo esaminato: splash screen, menu, finestra informazioni, form principale da cui vengono aperti altri form, ecc.

PROGETTO 9 : Report di stampa

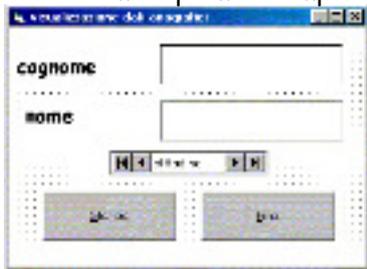
I dati di una tabella o i risultati di una query possono essere stampati su carta utilizzando due modalita'.

Primo modo: il modo piu' semplice consiste nell'utilizzare l'oggetto `Printer`, che corrisponde alla stampante predefinita nell'ambiente Windows; le righe di stampa vengono inviate alla stampante con il metodo `Print` seguito dalla variabile da stampare. Si possono definire le caratteristiche di stampa utilizzando le proprieta' dell'oggetto `Printer`:

- `FontName`: per il nome del font dei caratteri
- `FontSize`: per la dimensione
- `FontBold`: valore booleano per il grassetto (true stampa con grassetto, false senza)
- `FontItalic`: valore booleano per il corsivo
- `FontUnderline`: valore booleano per il sottolineato

Tutte le righe che usano il metodo `Print` vengono trasferite in un buffer, che viene inviato alla stampante quando viene eseguito il comando `EndDoc`.

Utilizziamo il primo esempio di `PROGETTO 6` per stampare i cognomi e nomi dei cittadini.



Aggiungiamo al form un pulsante e cui assegniamo nome `CmdStampa` e `caption &Stampa`; al pulsante deve essere assegnato il seguente codice:

```
Private Sub CmdStampa_Click()  
'stampa l'elenco dei cittadini con cognome e nome  
Data1.Recordset.MoveFirst 'si posiziona sul primo record  
Printer.FontName = "times new roman"  
Printer.FontSize = 12  
Printer.FontBold = False  
Do  
Printer.Print Data1.Recordset("COGNOME"); Tab; Data1.Recordset("NOME")  
Data1.Recordset.MoveNext 'si sposta al record successivo  
Loop Until Data1.Recordset.EOF  
Printer.EndDoc  
End Sub
```

Abbiamo gia' visto il significato delle diverse proprieta' utilizzate; "Tab" serve ad inserire un salto di tabulazione nella riga di stampa.

Secondo modo: e' un metodo un po' piu' sofisticato. L'ambiente di programmazione VB offre la possibilita' di costruire dei report di stampa in modo veloce utilizzando un componente aggiuntivo denominato Crystal report: e' un generatore di report simile a quello contenuto in `Access` e consente di disegnare il layout di stampa, in autocomposizione, attraverso una serie di finestre di dialogo. (viene attivato dal menu' "aggiunte" di VB con la scelta "progettazione report"). Sul risultato si puo' poi intervenire manualmente.

Il file cosi' creato viene salvato su disco in un file con estensione `.rpt`.

Proviamo a generare un report che permetta di stampare il contenuto di tutti i campi della tabella `CITTA DINO` del database `ANAGRAFICA`; prima di tutto progettiamo il seguente form:



Poi attiviamo il generatore di report inserendo tutti i campi quando ci vengono richieste durante la procedura; salviamo il report con il nome `ANAGRAFIC.RPT`.

Ora dobbiamo inserire nel form un controllo in grado di richiamare il report creato. Tale controllo si chiama "Crystal report" e, se non e' presente nella casella degli strumenti, sappiamo come inserirlo. Il controllo viene posizionato nel form ma rimane invisibile durante l'esecuzione. Al controllo dobbiamo associare la proprieta' `ReportFilename`, scegliendo il nome del report nella finestra di dialogo.

Poi inseriamo nel form un pulsante CmdStampa con caption &Stampa per consentire all'utente di eseguire la stampa; il codice associato e' il seguente:

```
Private Sub CmdStampa_Click()
' esegue la stampa
CrystalReport1.Action = 1
End Sub
```

Naturalmente dovete ricordarvi anche del pulsante Esci.

Questo e' il risultato quando si attiva la stampa:

CODICE	COGNOME	ESPE	SESSO	DATA	ALIQUOTA	REDDITO
1	VERDI	ROSA	F	55	15000	L. 1000000
2	ROSSI	GIULIA	F	45	18000	L. 1000000
3	BIANCHI	LUCA	M	52	12000	L. 1000000
4	GRANZI	FRANCA	F	58	14000	L. 1000000
5	FRANZI	FRANCO	M	48	19000	L. 1000000
6	DELLI	LEON	M	3	000	L. 1000000
7	VERDI	LEON	F	45	18000	L. 1000000
8	ROSSI	GIULIA	F	55	15000	L. 1000000

Con il codice viene associato alla proprieta' action del crystal report il valore 1, che attiva la stampa su carta; in alternativa 0 stampa in una finestra di anteprima e 2 stampa su file.

Ora che siete diventati bravissimi potete svolgere la seguente esercitazione.

Utilizzando Access create un database AZIENDA contenete le seguenti tabelle:

CLIENTE(CODCLI, RAGIONESOCIALE, TELEFONO, INDIRIZZO..., CODLOC)

LOCALITA(CODLOC, NOME)

Create un'applicazione VB contenente un form principale che consenta di accedere ai dati delle tabelle, ad alcune interrogazioni e alle stampe (scegliete voi quali); inserite anche i menu', lo splash screen e la finestra informazioni