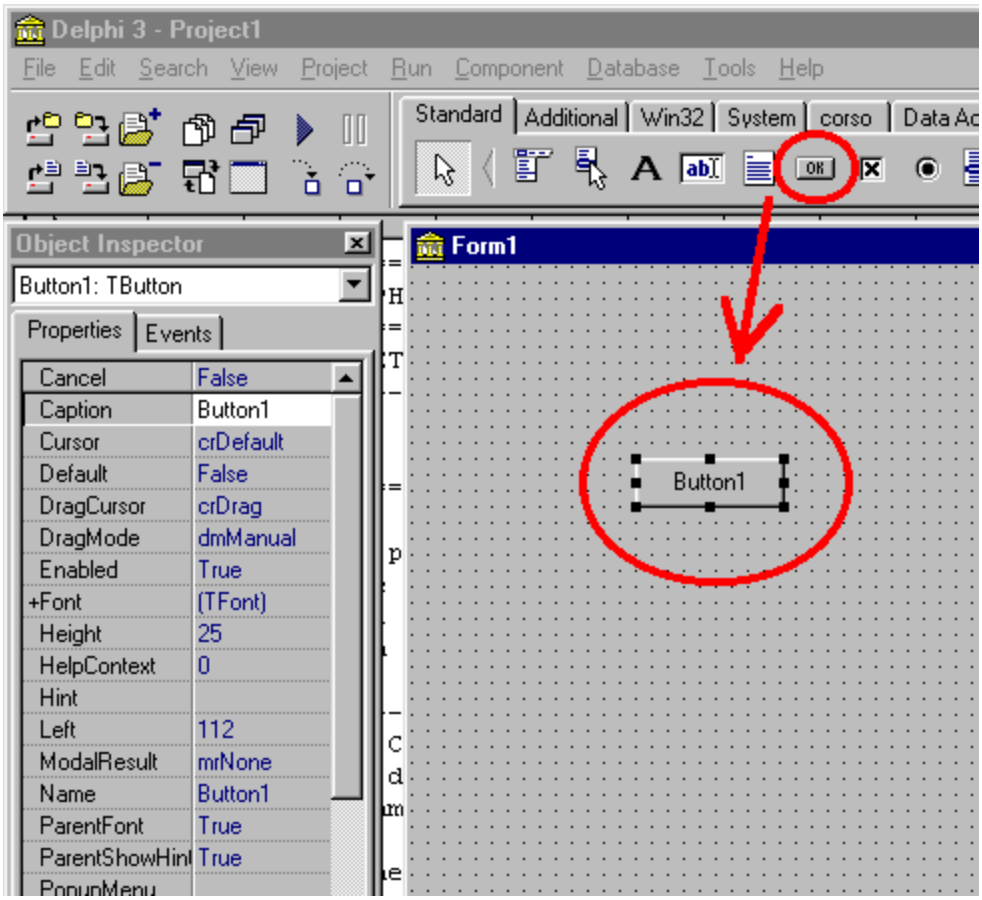
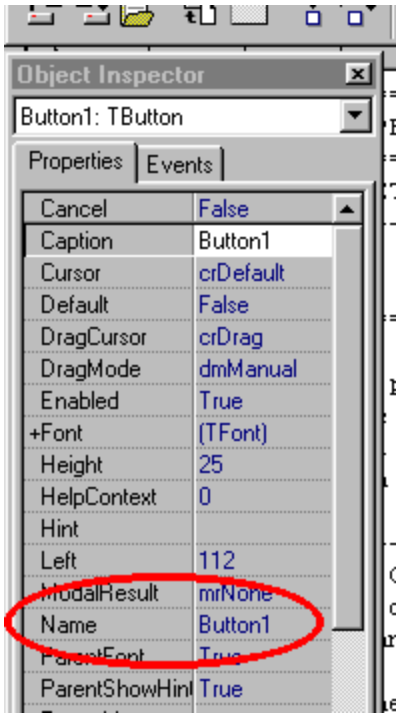
==========
CORSO PRATICO DI DELPHI
==========
2. PROPRIETA' DEI PULSANTI - PARTE PRIMA
==========

Impariamo ad usare i pulsanti (componente TButton). La lettera "T" indica i componenti, essenziali in una programmazione visuale. Ogni componente ha sua PROPRIETA' e suoi EVENTI; certe caratteristiche sono comuni a piu' componenti.

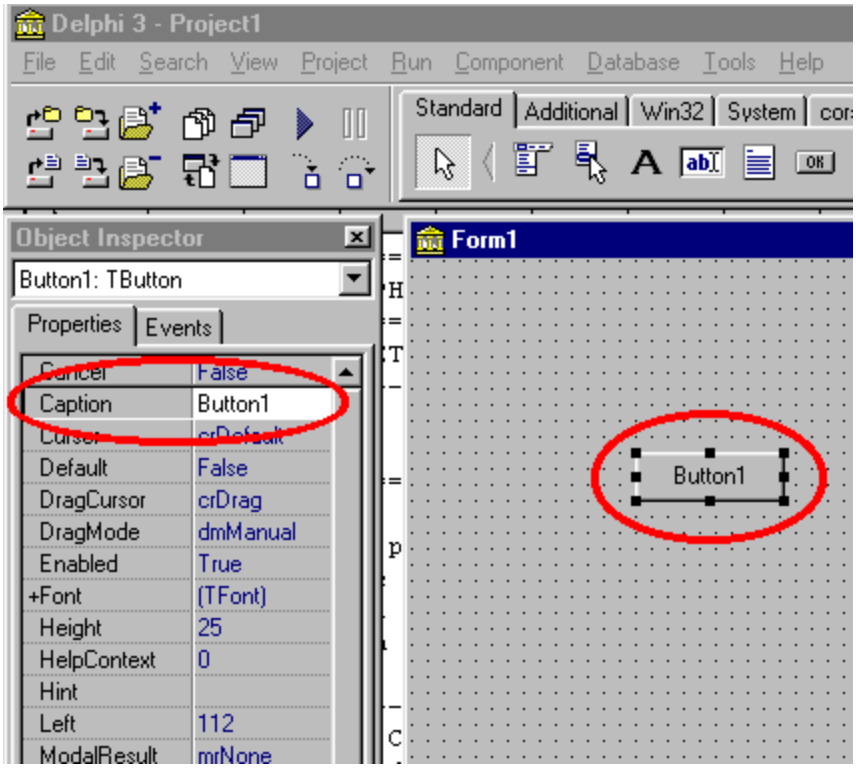
1. Avviare Delphi 3
2. Inserire sulla finestra un componente "Button" dalla cartellina "Standard"



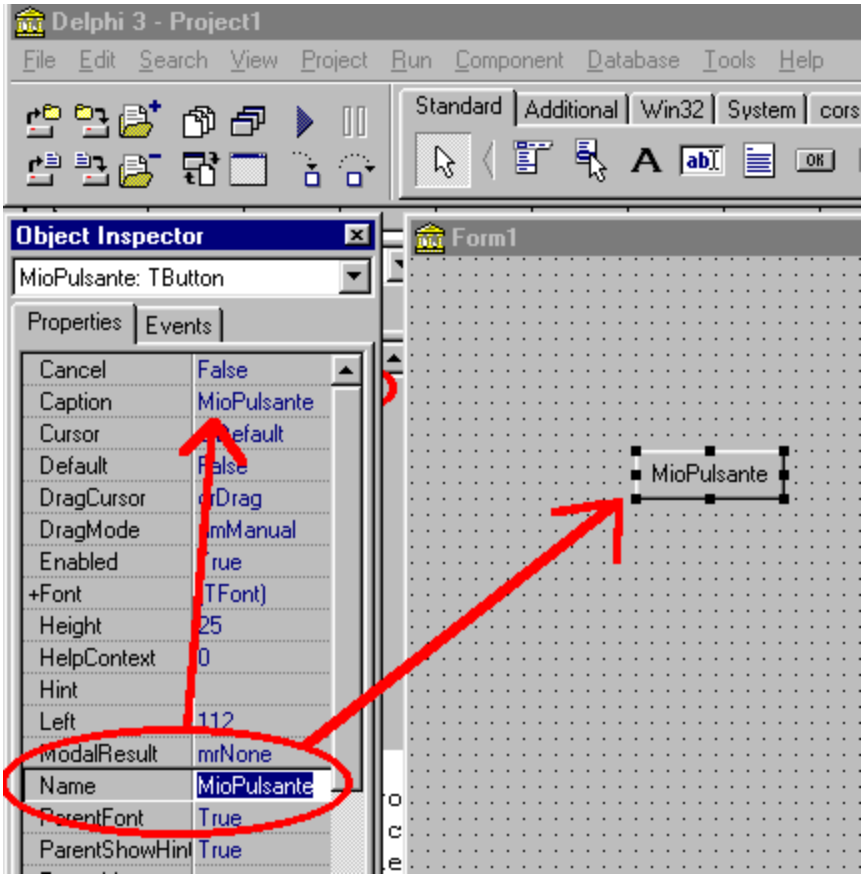
3. Modifichiamo alcune proprietà di questo componente... Il componente ha assunto automaticamente il nome ("name") Button1 ed è di tipo TButton;



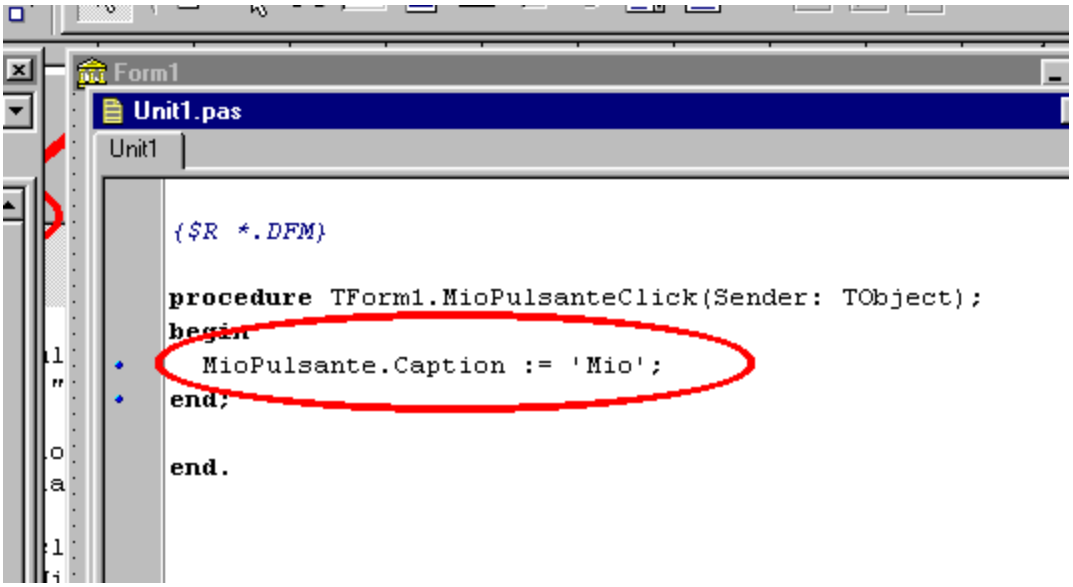
il pulsante presenta anche una didascalia visibile, che può essere modificata scrivendo un testo diverso per la proprietà "Caption"



Se si cambia la proprietà "Caption", il nome non cambia; se si cambia la proprietà "Name", cambia automaticamente anche la proprietà "Caption". Cambiare la proprietà "Name" in "MioPulsante"; la didascalia visibile cambia automaticamente in "MioPulsante".

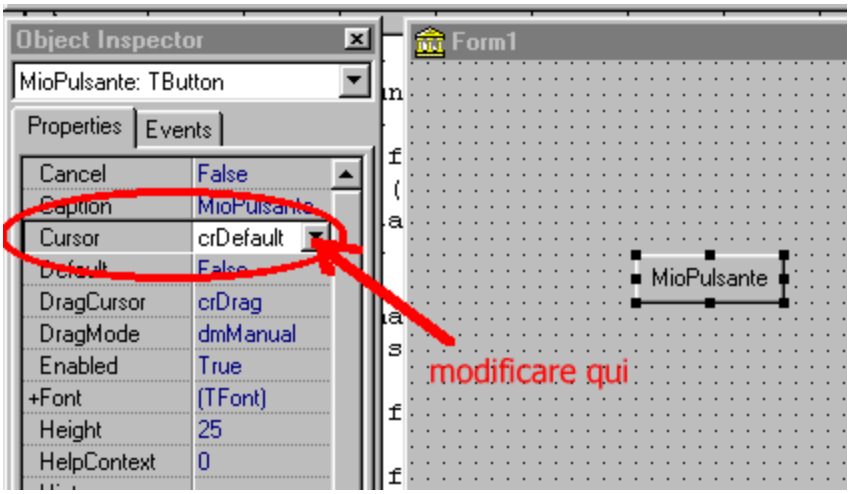


Doppio click ora sul pulsante; scriviamo questo testo tra "begin" (inizio) e "end" (fine):
  MioPulsante.Caption := 'Mio';
[cioè: la didascalia del pulsante MioPulsante è posta uguale a 'Mio']



in esecuzione, se cliccate sul pulsante, cambia la didascalia da "Mio pulsante" a "Mio";

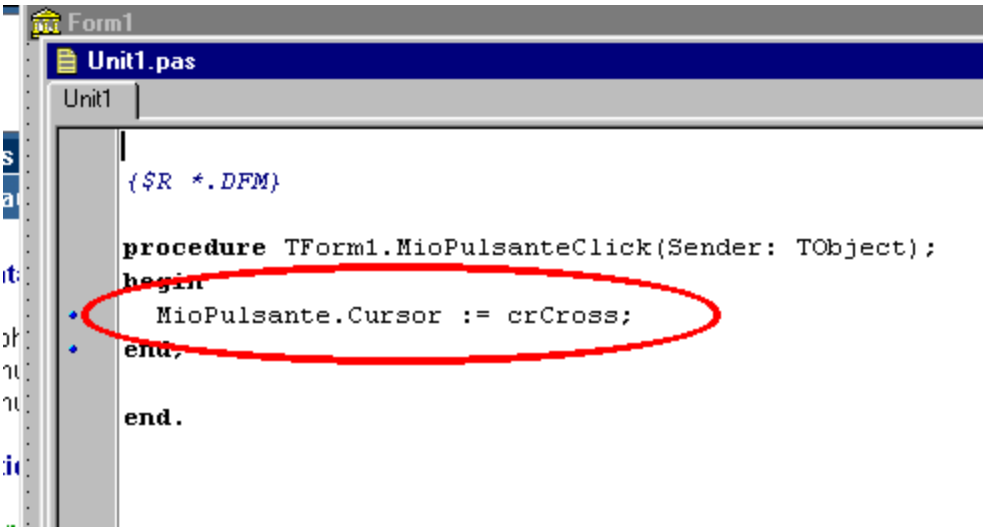
4. Proprietà "Cursor": cosa succede al cursore del mouse quando passa sopra al pulsante?
   E' possibile cambiare la proprietà "Cursor" da "crDefault" a:
   crAppStart ---> clessidra (cursore inizio applicazioni)
   crArrow ---> freccia (cursore freccia)
   crCross ---> croce (cursore croce)
   crDrag ---> oggetto trascinato (cursore trascinamento)
   crHandPoint ---> mano che indica (cursore selezione)
   crHelp ---> punto interrogativo (cursore aiuto)
   crHourGlass ---> clessidra grande (cursore lunga attesa)
   crHSplit ---> frecce divergenti orizzontali (cursore separazione orizzontale)
   crBeam ---> stanghetta (cursore inserimento)
   crMultiDrag ---> tanti oggetti trascinati (cursore trascinamento multiplo)
   crNo ---> segnale di divieto (cursore no)
   crNoDrop ---> segnale di divieto piccolo (cursore no trascinamento)
   crSizeNESW ---> frecce da NordEst a SudOvest
   crSizeNS ---> frecce da Nord a Sud
   crSizeNWSE ---> frecce da NordOvest a SudEst
   crSizeWE ---> frecce da Ovest a Est
   crSQLWait ---> clessidra SQL (NON ci interessa)
   crUpArrow ---> freccia sottile verso sopra
   crVSplit ---> frecce divergenti (cursore separazione verticali)
   "cr" sta per "cursore", come avrete capito...



Provate a inserire questi valori ed eseguite ogni volta, passando
sopra il pulsante per vedere l'effetto.

Anche qui i valori possono essere cambiati in fase di esecuzione:
Doppio click sul pulsante; al posto di
    MioPulsante.Caption := 'Mio';
scrivere
    MioPulsante.Cursor := crCross;
[cioè: il cursore per il pulsante deve avere l'aspetto di una croce]

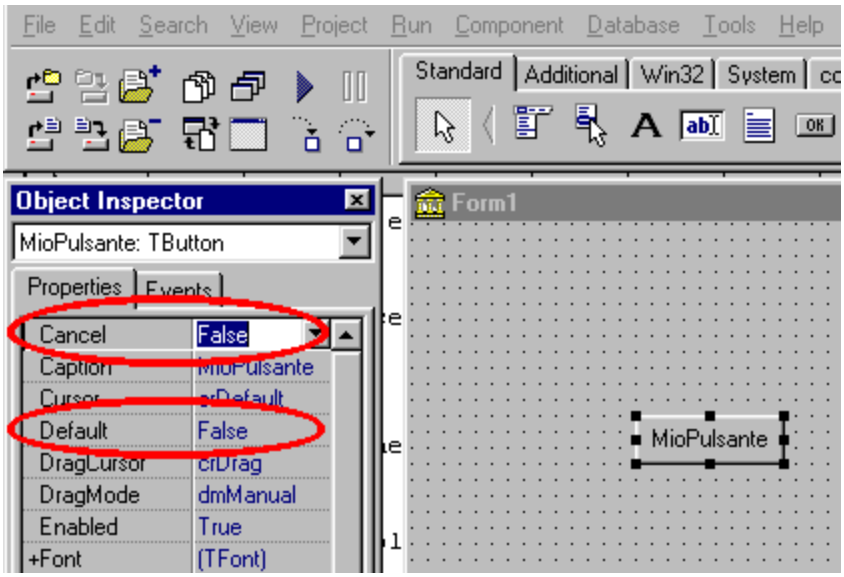
(potete scegliere qualsiasi altro valore)



Dare esecuzione (F9) e cliccare sul pulsante.

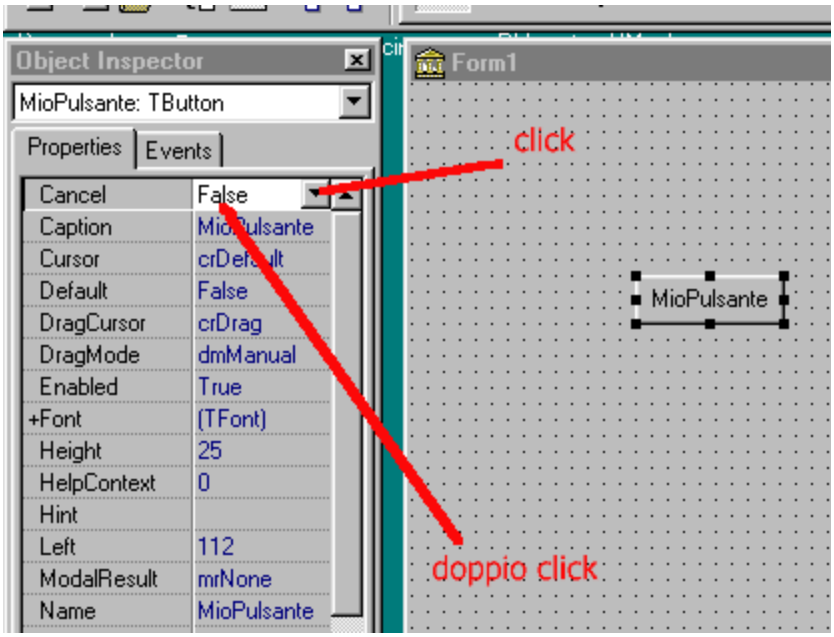
Adesso salviamo il nostro "lavoro" in una directory di nome "corso2";
1. File - Save project as
2. navigare fino alla directory corso;
3. creare una directory "corso2"
4. Cliccare su "Salva"
5. Scrivere come nome progetto "Corso2" e cliccare ancora su "Salva"

5. Le proprietà "Cancel" e "Default" sono impostate a "False" (FALSO),



ma possono essere modificate (sia nell'Object Inspector, sia tramite codice) a "True" (VERO).
Cosa significano queste proprietà?
La proprietà "Cancel": se si imposta a "True", quando si preme il tasto Esc viene eseguito il codice del click sul pulsante. Se ci sono più pulsanti con la proprietà "Cancel" impostata a "True", viene eseguito solo il primo pulsante visibile (si segue il "tab order", l'ordine di creazione, che può essere modificato).
"False" e "True" sono valori Boolean: hanno un'importanza enorme nella programmazione... Il concetto è questo: SE è vero ALLORA, se è falso INVECE...

ESERCIZIO: Impostare a "True" il valore della proprietà "Cancel" (doppio click sullo spazio bianco, o scelta dalla freccetta)



ed eseguire (F9): premendo il tasto Esc, noterete che il mouse passando sul pulsante cambia forma, come se avessimo fatto click sul pulsante stesso.

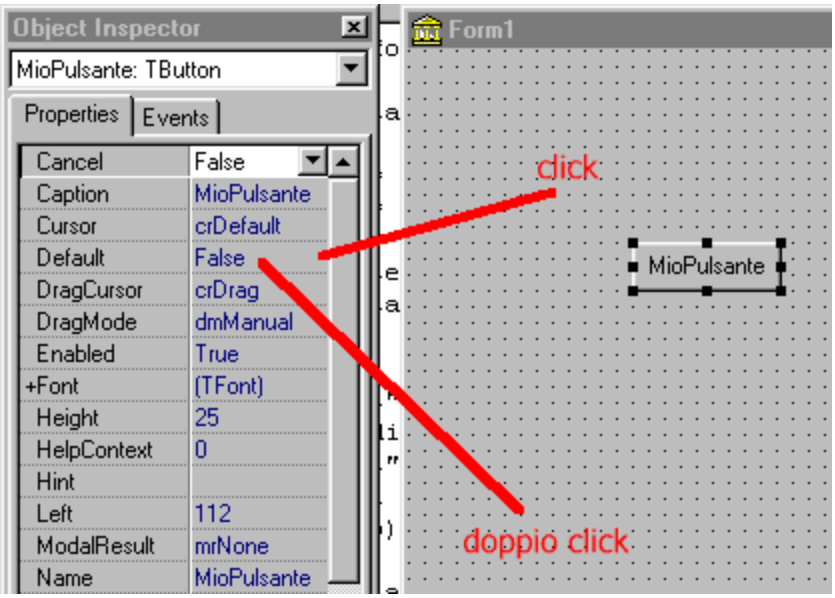
Tramite codice, questa proprietà si modifica nel seguente modo:

```
MioPulsante.Cancel := True;  {Attivazione di Esc per il pulsante}
MioPulsante.Cancel := False;  {Disattivazione di Esc per il pulsante}
```

Il codice, come vedete, è molto "discorsivo" (anche se in inglese) ed è uno dei grandi vantaggi di Delphi.

La proprietà "Default": se si imposta a "True", quando si preme il tasto Enter viene eseguito il codice del click sul pulsante. Se ci sono più pulsanti con la proprietà "Default" impostata a "True", viene eseguito solo il primo pulsante visibile (si segue sempre il "tab order", l'ordine di creazione, che può essere modificato).
IMPORTANTE: qualsiasi Pulsante (TButton) che ha il fuoco (cioè è attivo ed evidenziato), diventa pulsante "di default" temporaneamente; in conclusione, quando si preme Enter possono succedere due cose: 1. viene eseguito il codice del Pulsante su cui ci si trova; 2. viene eseguito il codice del primo pulsante creato (o con "tab order" più basso).

ESERCIZIO: Impostare a "True" il valore della proprietà "Default" (doppio click sullo spazio bianco, o scelta dalla freccetta)



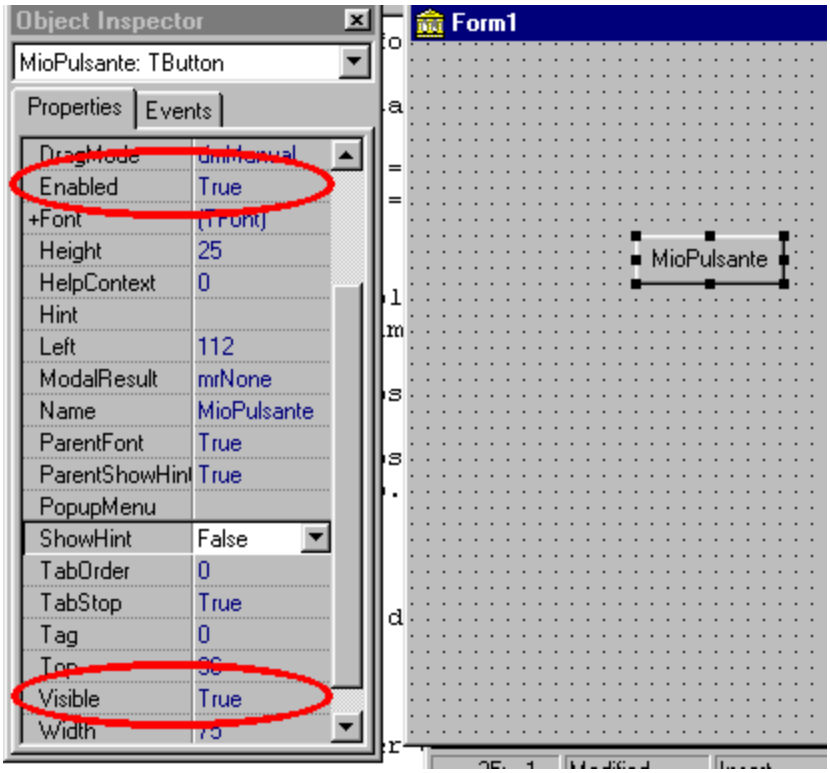
ed eseguire (F9): premendo il tasto Enter (Invio), noterete che il mouse passando sul pulsante cambia forma, come se avessimo fatto click sul pulsante stesso.

Tramite codice, questa proprietà si modifica nel seguente modo:

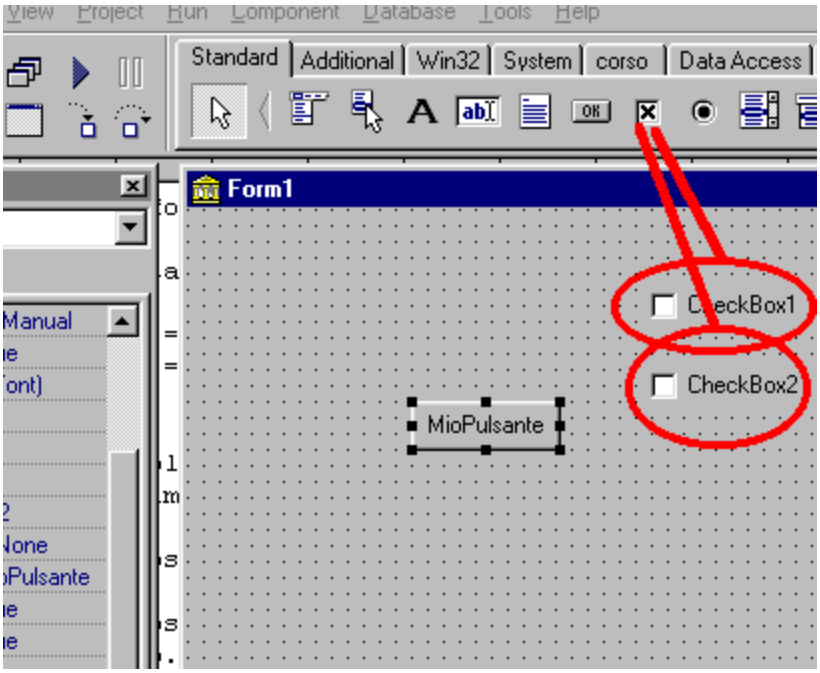
```
MioPulsante.Default := True;  {Attivazione di Enter per il pulsante}
MioPulsante.Default := False;  {Disattivazione di Enter per il pulsante}
```

6. Le proprietà "Enabled" e "Visible", comuni a quasi tutti i componenti, sono importantissime.

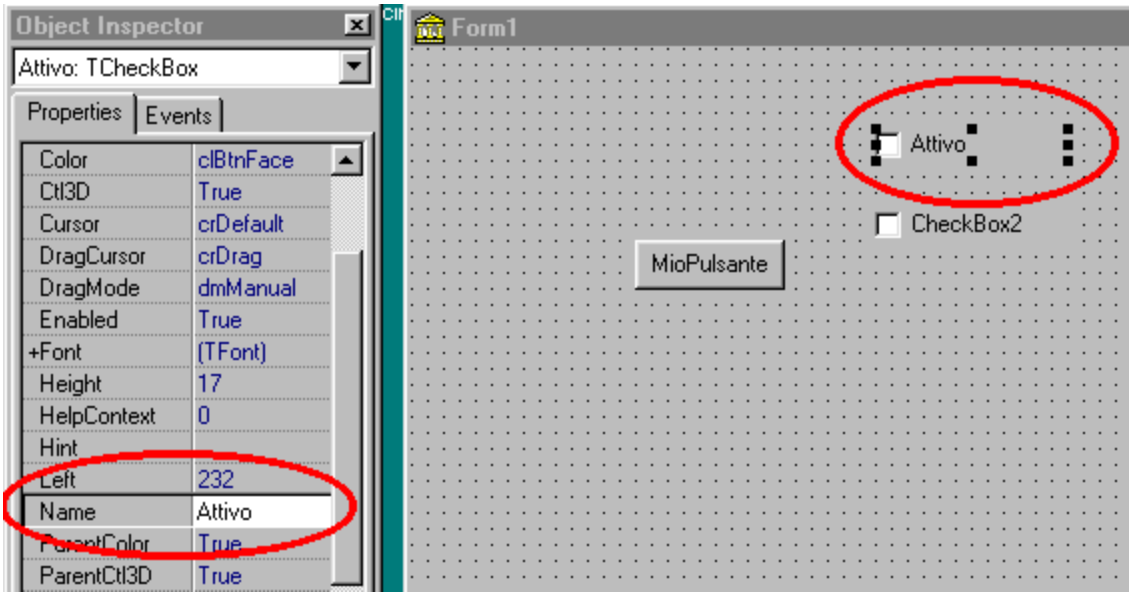
   Con "Enabled" impostato a "False" si disattiva il componente, ma resta visibile.
   Con "Visible" impostato a "False" si rende invisibile il componente, che pero' resta attivo.



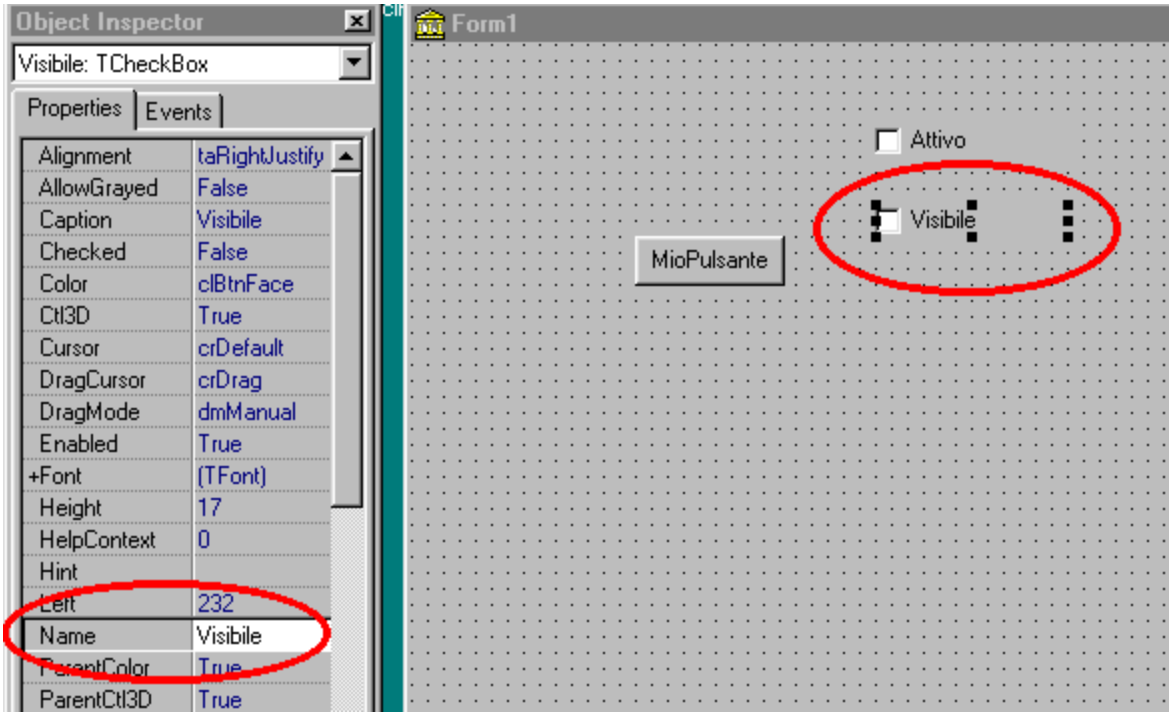
Facciamo però qualcosa di pratico, altrimenti vi annoiate...
Incolliamo 2 componenti CheckBox dalla cartella Standard:



Click semplice su CheckBox1 e modifica della proprietà "Name" in "Attivo".



Click semplice su CheckBox2 e modifica della proprietà "Name" in "Visibile".



Fare doppio click sul tasto "Attivo" e scrivere/incollare questo codice tra "begin" (inizio) e "end" (fine):

```
  if Attivo.Checked = True then
  {se il tasto di scelta "Attivo" HA il segno di spunta allora...}
  begin
    MioPulsante.Enabled := True;
    {MioPulsante viene attivato}
  end;
  if Attivo.Checked = False then
  {SE il tasto di scelta "Attivo" NON HA il segno di spunta ALLORA...}
  begin
    MioPulsante.Enabled := False;
    {MioPulsante viene disattivato}
  end;
```

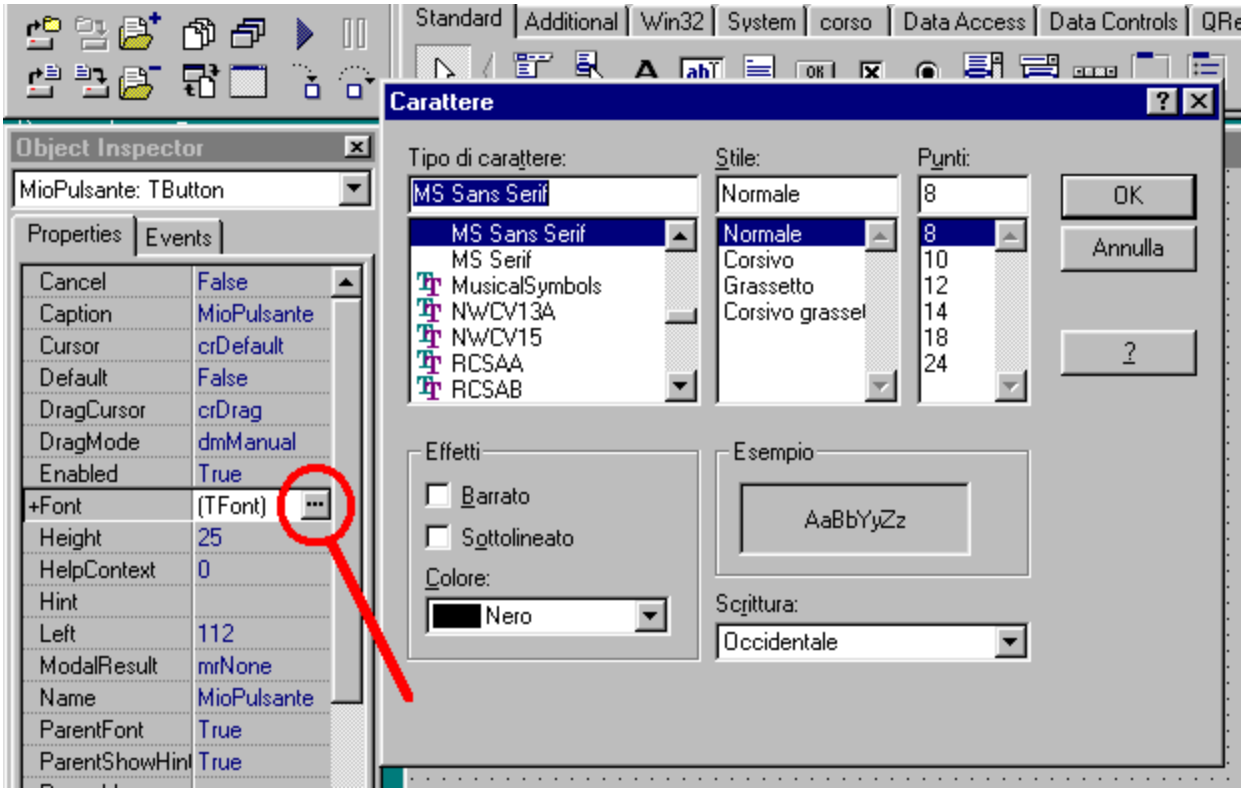
Fare doppio click sul tasto "Visibile" e scrivere/incollare questo codice tra "begin" (inizio) e "end" (fine):

```
  if Visibile.Checked = True then
  {se il tasto di scelta "Visibile" HA il segno di spunta allora...}
  begin
    MioPulsante.Visible := True;
    {MioPulsante viene reso visibile}
  end;
  if Visibile.Checked = False then
  {SE il tasto di scelta "Visibile" NON HA il segno di spunta ALLORA...}
  begin
    MioPulsante.Visible := False;
    {MioPulsante viene reso invisibile}
  end;
```

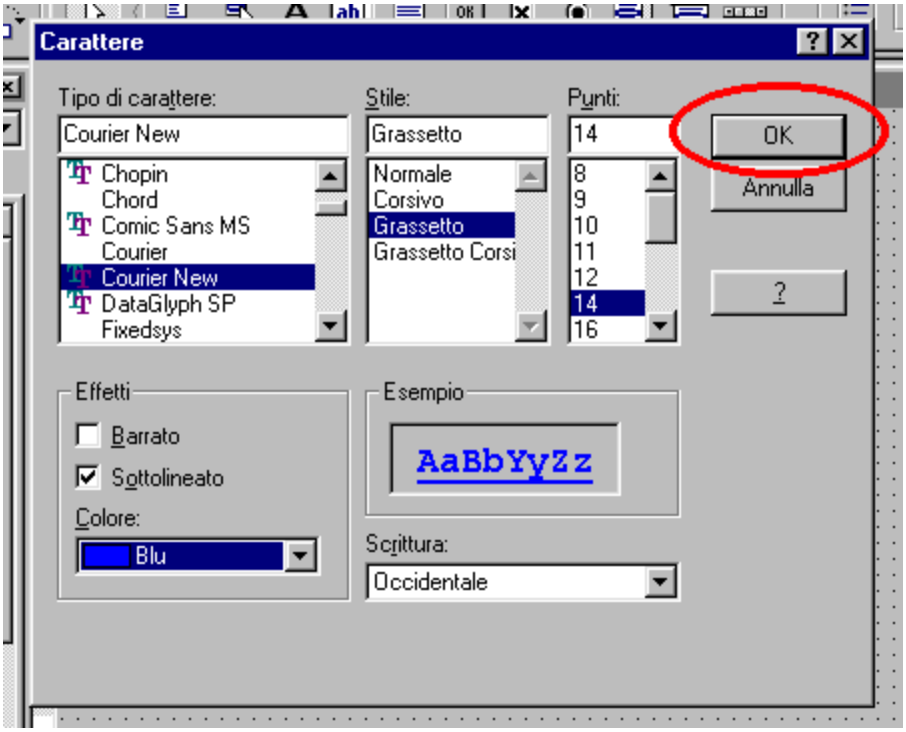
Eseguire il programma (F9) e verificare le conseguenze quando si clicca sui pulsanti di scelta.

Salviamo il nostro "lavoro";
1. File - Save

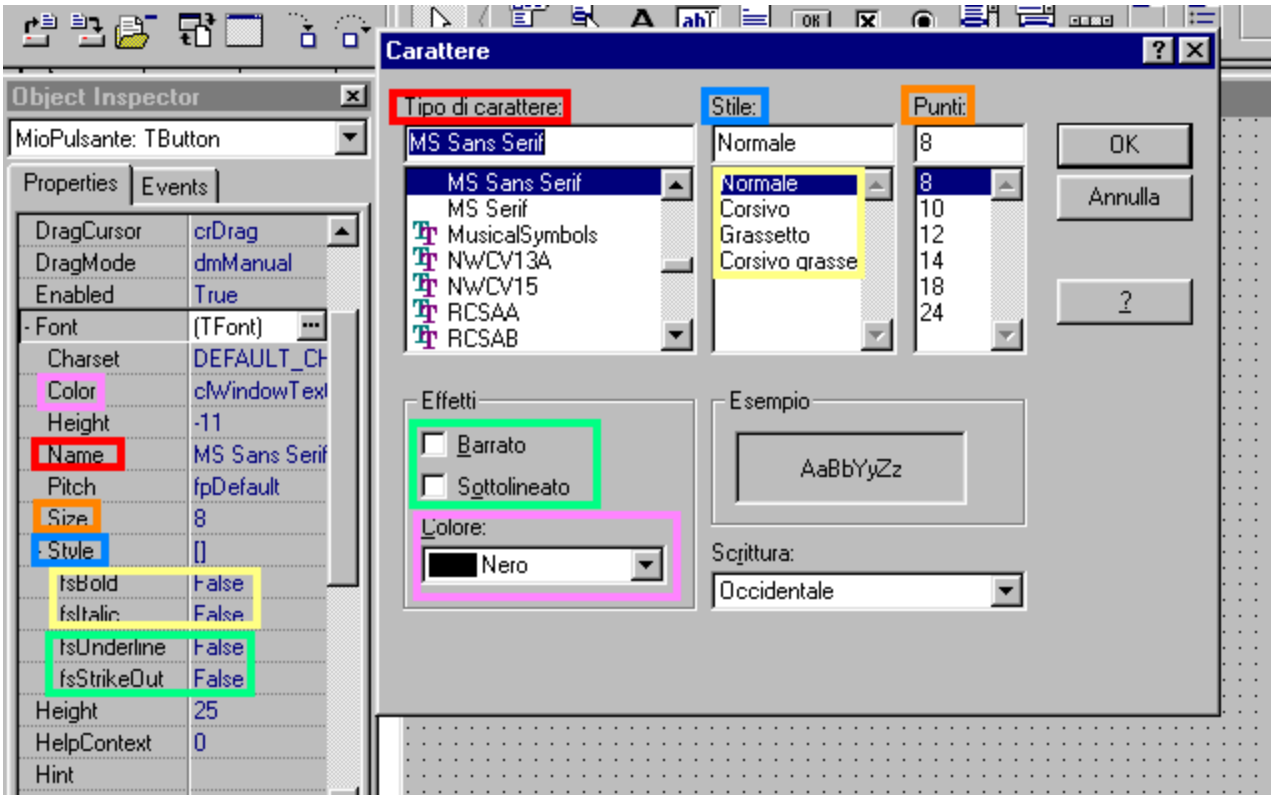
7. Stiamo per vedere altre proprietà di "MioPulsante", molte delle quali comuni a tanti componenti. E' il caso della proprietà "Font", per la quale si può usare una maschera standard (come negli editor di Windows); click semplice su "MioPulsante", quindi doppio click sui tre puntini vicino alla proprietà "Font":



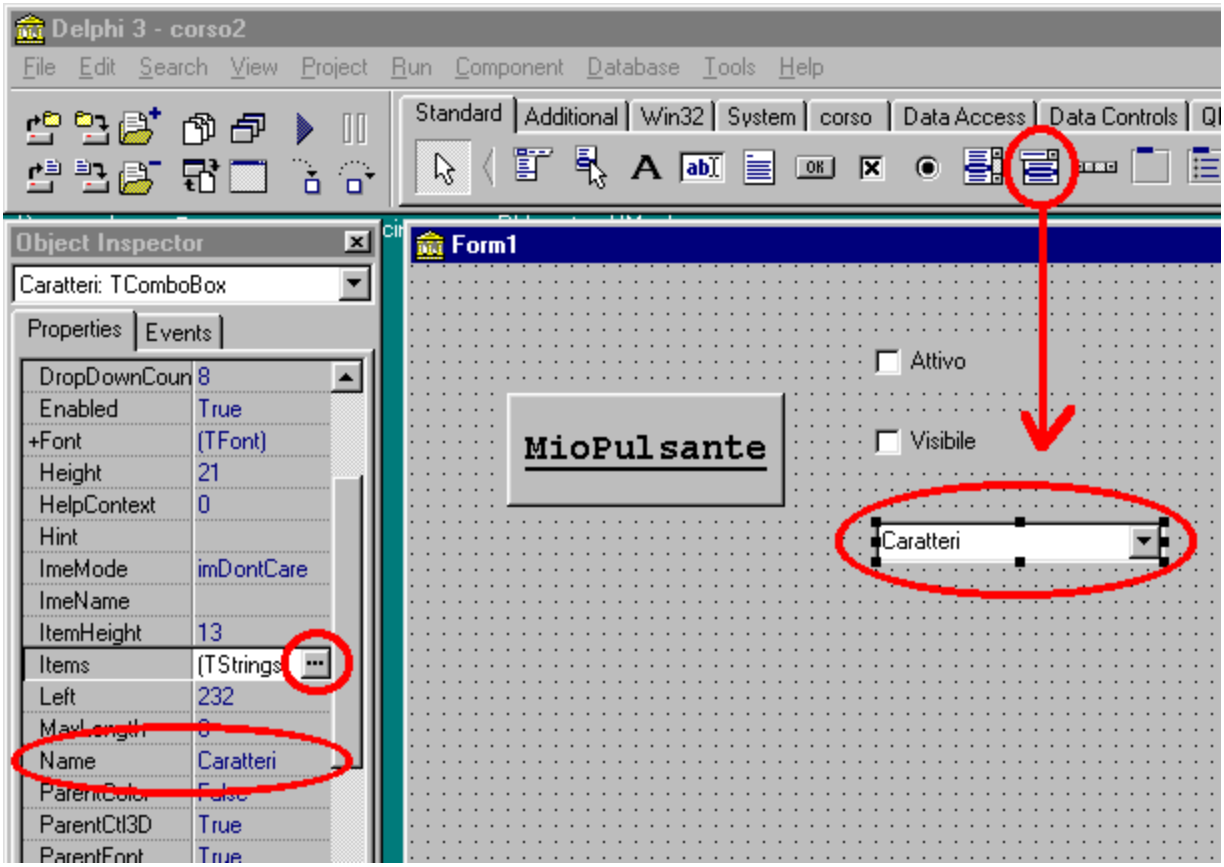
Cambiare il tipo di carattere da "MS Sans Serif" a "Courier New", lo stile da "Normale" a "Grassetto", i "Punti" da "8" a "14". Nel settore "Effetti", cliccare su "Sottolineato" e scegliere "Blu" per il colore. Cliccare su "OK". Ingrandire il pulsante per adattarlo al testo più grande.



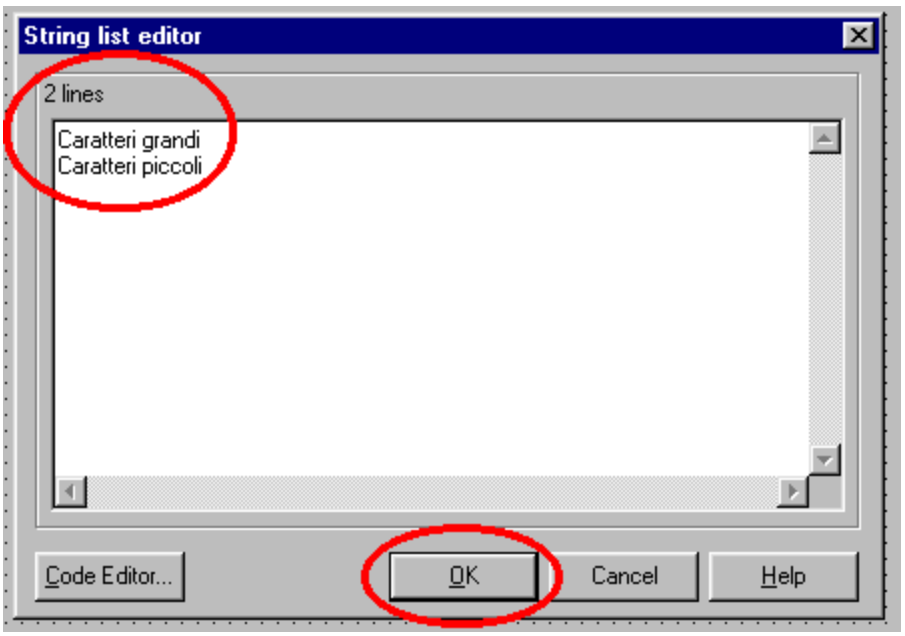
Si possono modificare tutte le proprietà (o solo alcune) direttamente dall"Object Inspector"; l'immagine mostra la corrispondenza delle funzioni:



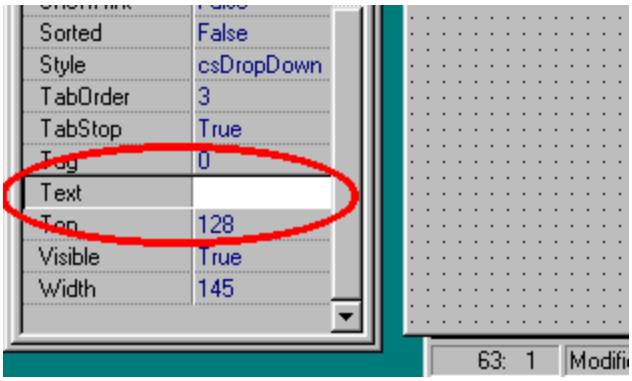
Come si modifica da proprietà questo codice? Niente di più semplice: Inserire un componente ComboBox (ComboBox1) dalla cartellina "Standard"; cambiarne la proprietà name da "ComboBox1" a "Caratteri"; cliccare sui tre puntini vicino alla proprietà "Items" (cioè elementi).



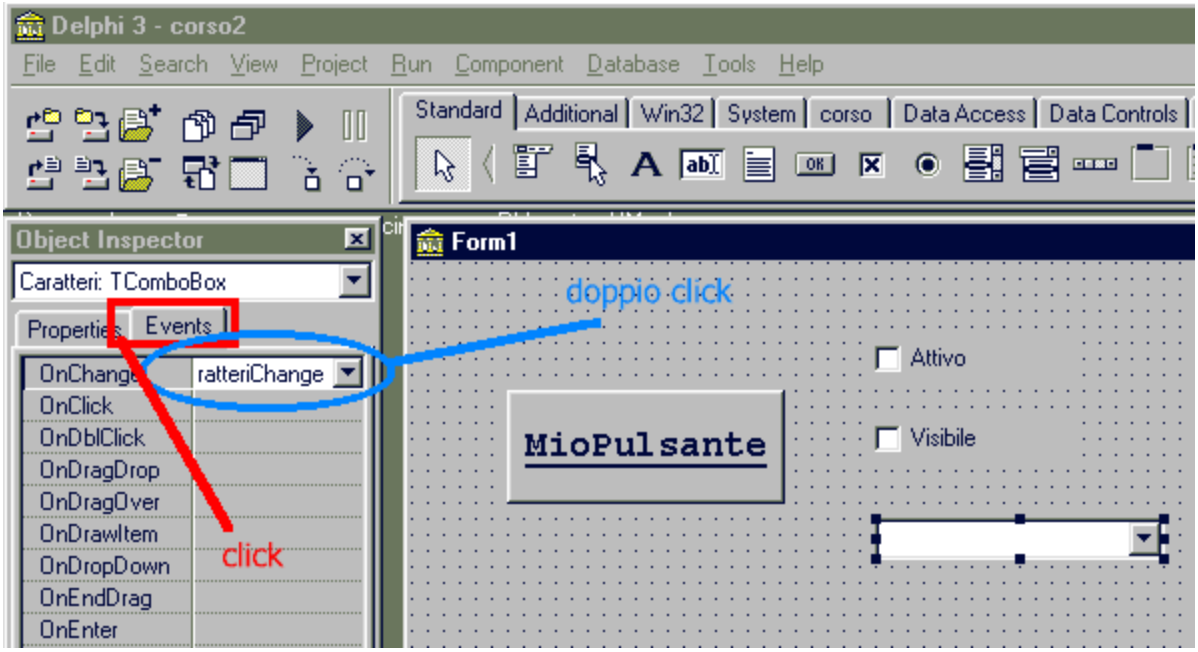
Scrivere queste due linee:
Caratteri grandi
Caratteri piccoli
e cliccare su OK



Azzerare la proprietà "Text" (cioè TESTO - corrisponde alla proprietà "Caption" di MioPulsante).



Click semplice sulla linguetta "Events", quindi doppio click sullo spazio vicino a "OnChange" (cioè QUANDO CAMBIA...);



scrivere/incollate questo codice tra "Begin" e "End":

```
  if Caratteri.Text = 'Caratteri grandi' then
  {se abbiamo selezionato "Caratteri grandi", cioè se il testo del componente
   "Caratteri" è 'Caratteri grandi' allora...}
    begin
    MioPulsante.Font.Size := 14;
    {la grandezza (size) del carattere (font) del pulsante (MioPulsante) viene
     posta uguale a 14}
    MioPulsante.Font.Name := 'Courier New';
    {il nome (name) del carattere (font) del pulsante (MioPulsante) viene
    posta uguale a 'Courier New'}
    with MioPulsante do begin
    {consideriamo MioPulsante, quindi impostiamo...}
    Font.Style := Font.Style
                  {+ o - per togliere o aggiungere}
                  - [fsStrikeOut]
                  - [fsItalic]
                  + [fsUnderline]
                  + [fsBold];
    {inseriamo il sottolineato e il grassetto}
    end;
  end;
  if Caratteri.Text = 'Caratteri piccoli' then
```

```
  {se abbiamo selezionato "Caratteri piccoli", cioè se il testo del componente
   "Caratteri" è 'Caratteri piccoli' allora...}
    begin
      MioPulsante.Font.Size := 8;
      {la grandezza (size) del carattere (font) del pulsante (MioPulsante) viene
       posta uguale a 8}
      MioPulsante.Font.Name := 'MS Sans Serif';
      {il nome (name) del carattere (font) del pulsante (MioPulsante) viene
       posta uguale a 'MS Sans Serif'}
      with MioPulsante do begin
      {consideriamo MioPulsante, quindi impostiamo...}
      Font.Style := Font.Style
                    {+ o - per togliere o aggiungere}
                    - [fsStrikeOut]
                    - [fsItalic]
                    - [fsUnderline]
                    - [fsBold];
      {togliamo il sottolineato e il grassetto}
    end;
  end;
```

Eseguire il programma (F9) e verificare le conseguenze quando si scelgono
le opzioni dalla finestra a tendine.

Può bastare per ora...
   Salviamo il nostro "lavoro";
   1. File - Save