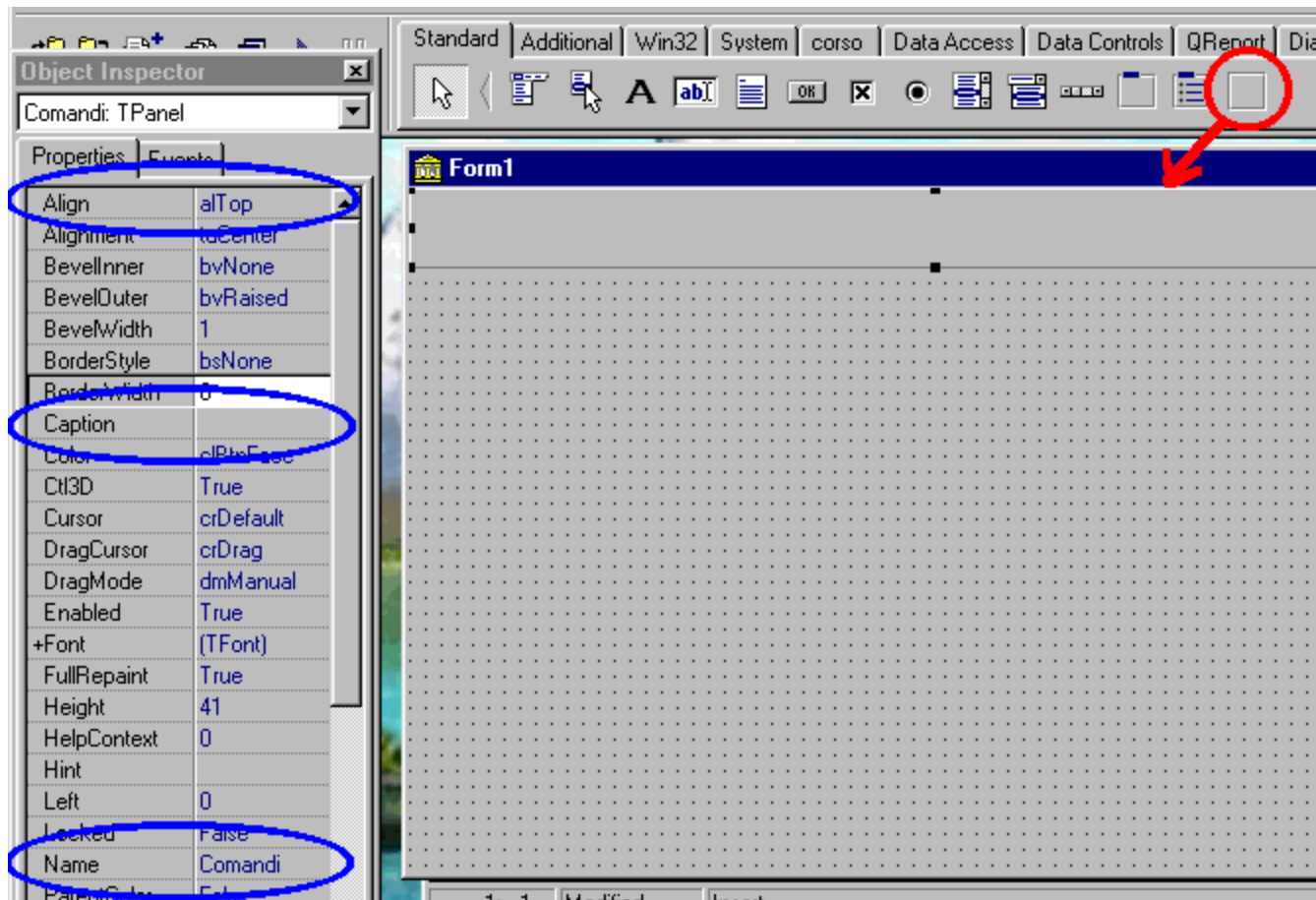


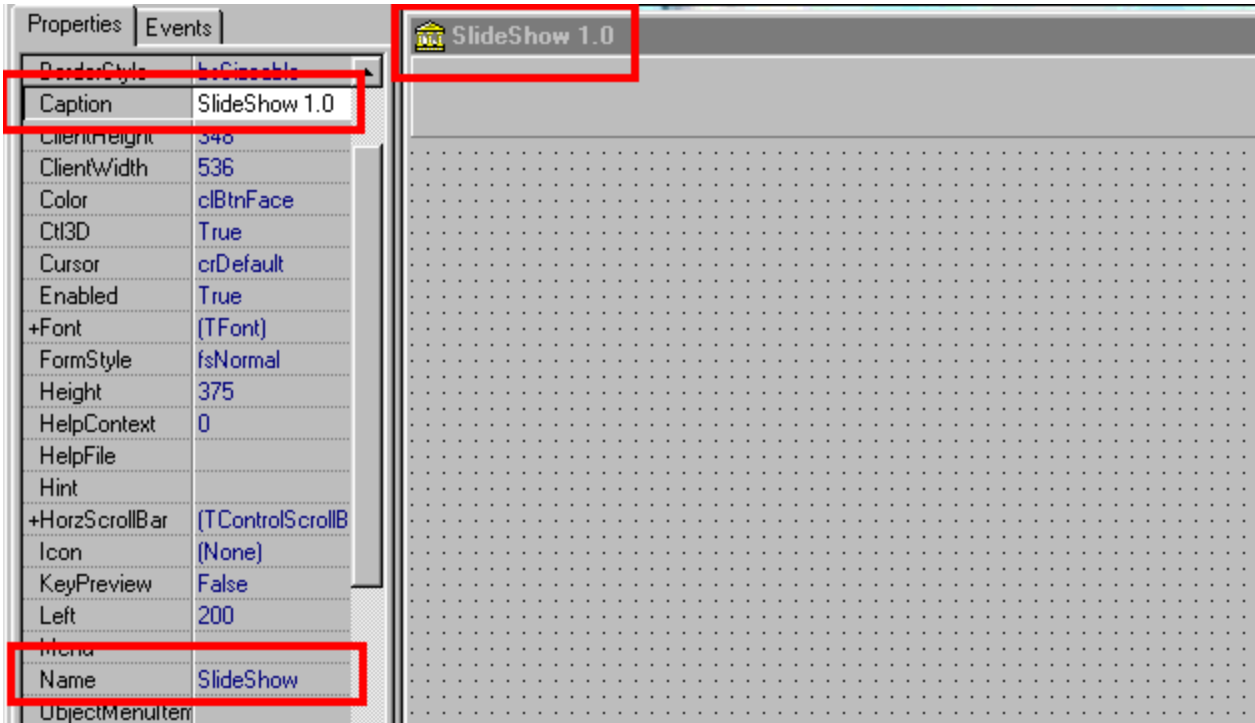
10. IL COMPONENTE TIMER: IL PROGRAMMA SLIDESHOW 1.0

=====

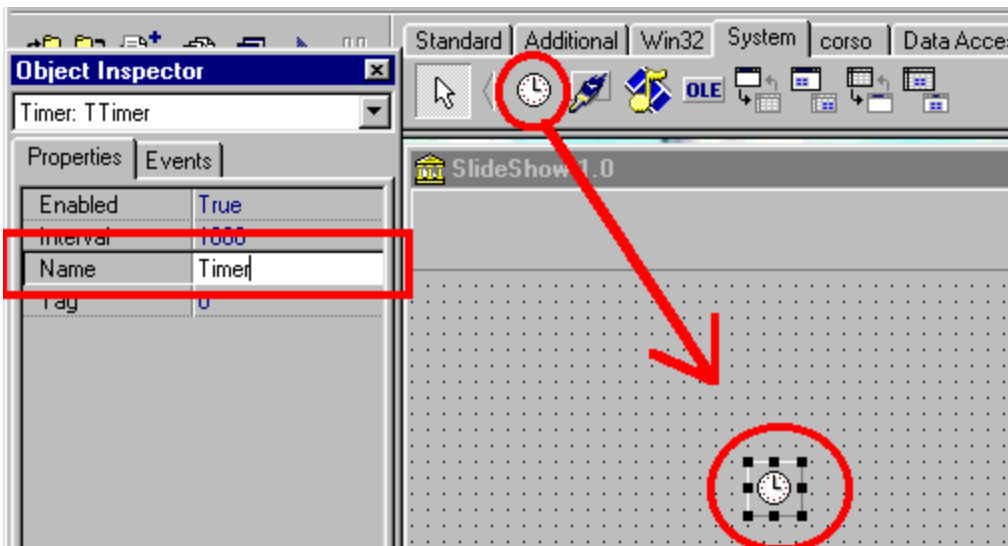
1. Il nostro corso, come detto, è soprattutto PRATICO e mi perdonerete qualche "salto"; ecco quindi il componente Timer (TTimer)...
- Avviare Delphi 3.
2. Inserire sulla form un componente TPanel (Panel) dalla cartella "Standard" e cambiarne queste proprietà:
Name ----> "Comandi"
Caption ----> "" (NIENTE - nessun carattere)
Align ----> alTop (ALLINEAMENTO IN ALTO)



Cliccare su un punto della form (FINESTRA) e cambiarne queste proprietà:
Name ----> "SlideShow"
Caption ----> "SlideShow 1.0"

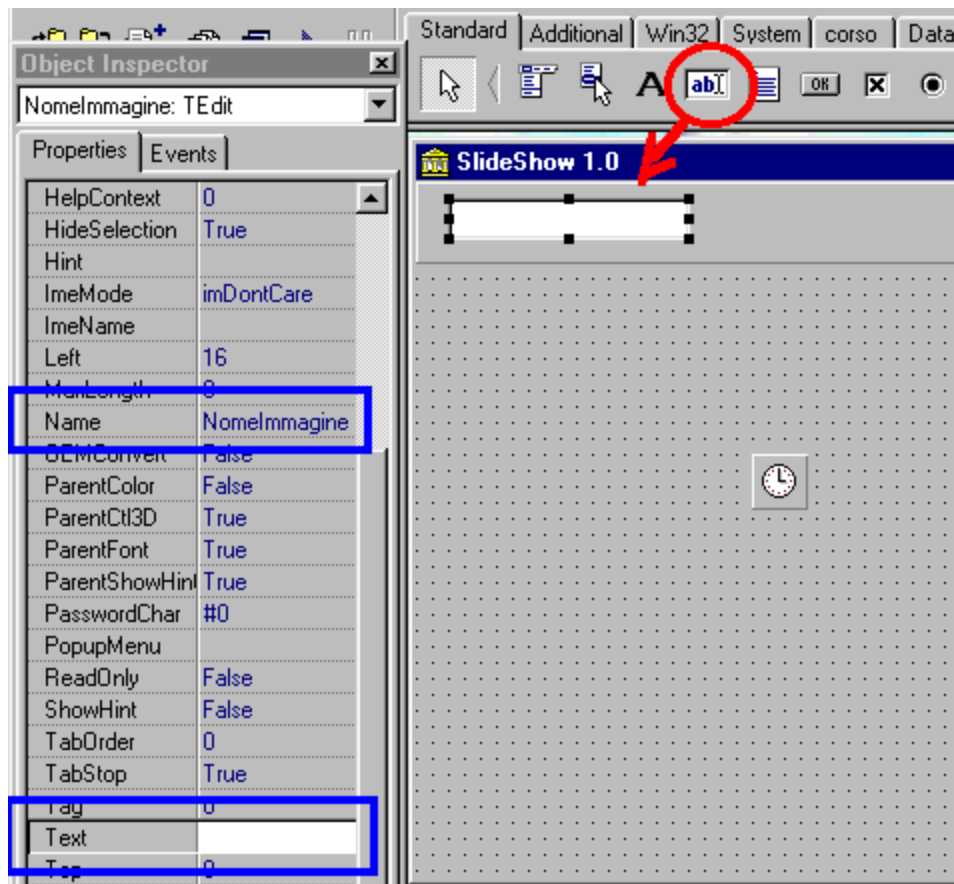


Inserire un componente TTimer (Timer) dalla cartella "System" e cambiare la proprietà "Name" da "Timer1" a "Timer". Da notare che allo stato attuale Timer è attivato (Enabled è impostato a True, cioè vero) e presenta un intervallo di 1 secondo (Interval è impostato a 1000, cioè a 1000 millisecondi).

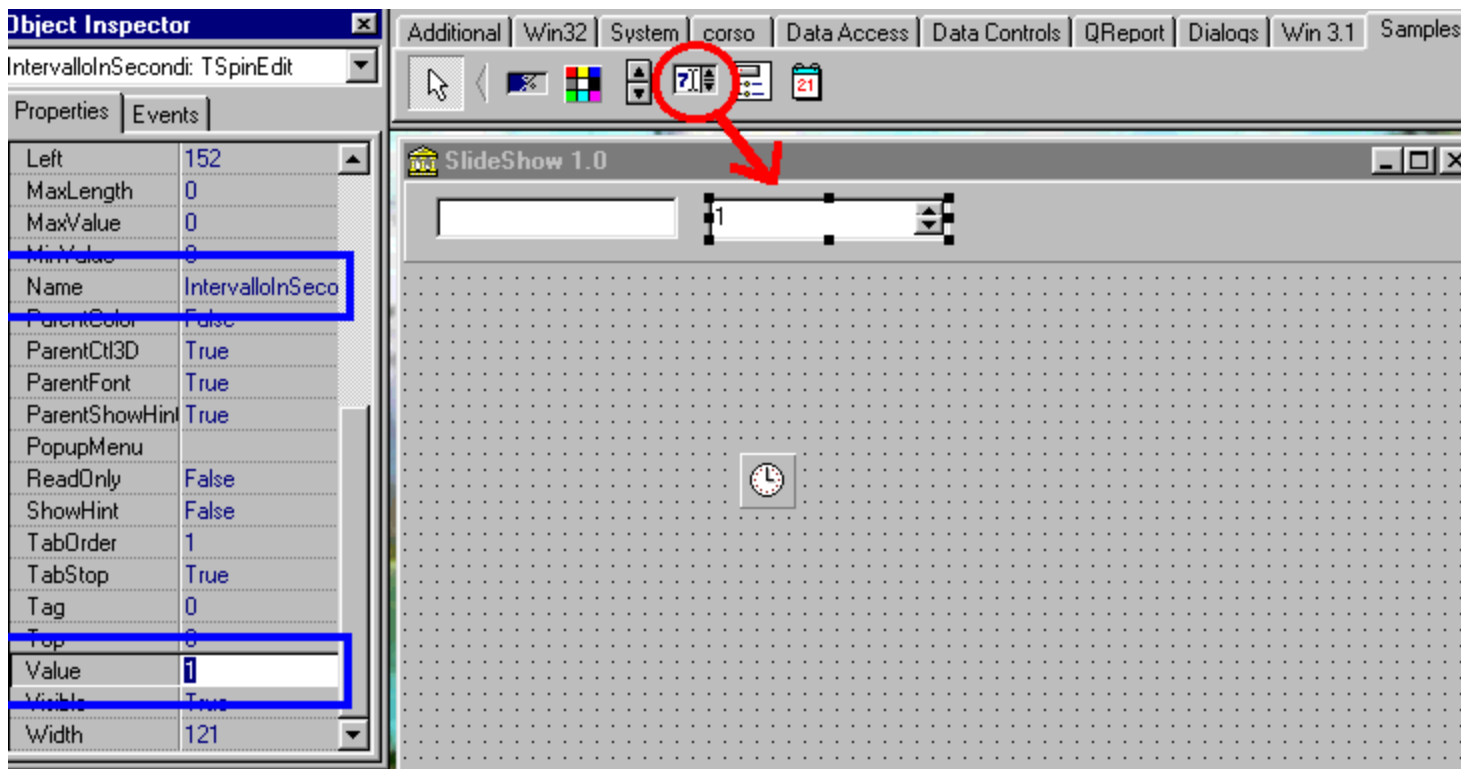


Inserire ora sul pannello "Comandi" gli altri componenti che ci servono:

- dalla cartella "Standard" inserire un TEdit (Edit) e cambiarne queste proprietà:
Name ----> "NomeImmagine" Text ----> "" (NIENTE - nessun carattere)



- dalla cartella "Samples" inserire un TSpinEdit (SpinEdit) e cambiarne queste proprietà:
Name ----> "IntervalloInSecondi" Value ----> "1"

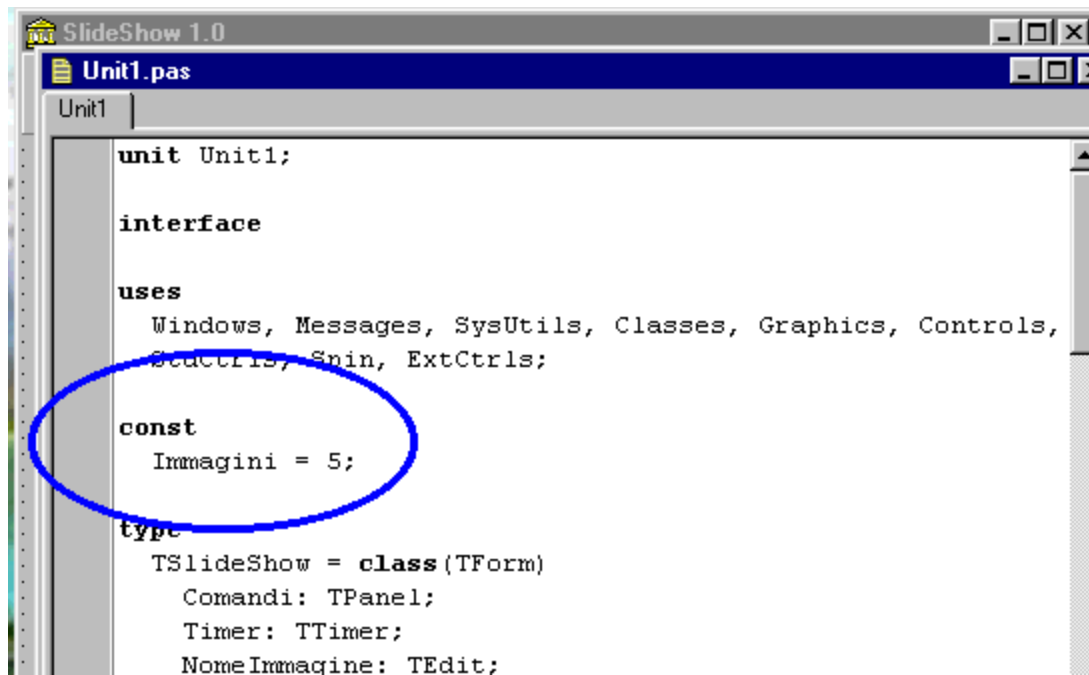


- dalla cartella "Standard" inserire un TCheckBox (CheckBox) e cambiarne queste proprietà:
Name ---> "Attivazione" Caption ---> "Attiva"
Checked ---> "True"

Salvare nella directory "Corso10" con il nome di progetto "Corso10".
F9 per provare la maschera che abbiamo creato.

3. Prima di procedere oltre, copiare 5 immagini .bmp nella directory Corso10 e rinominarle nel seguente modo: BITMAP1.BMP - BITMAP2.BMP - BITMAP3.BMP - BITMAP4.BMP - BITMAP5.BMP
4. Premere F12 per passare al listato del programma. Inserire la costante "Immagini":
dopo le "Uses" all'inizio del listato scrivere:

```
const  
Immagini = 5;
```



Le costanti sono valori che non cambiano nel corso dell'esecuzione del programma. Possono essere usate più volte. Il vantaggio principale: per cambiare tutti i valori nel programma basta cambiare semplicemente la costante (UNA VOLTA SOLA). Esempio: per moltiplicare per 2:

```
const  
  VALORE = 2;  
.....  
begin  
  B = 1 * VALORE;    otteniamo 2  
  C = 2 * VALORE;    otteniamo 4  
  D = 3 * VALORE;    otteniamo 6  
end;
```

per moltiplicare per 3 basta cambiare il valore della costante (UNA VOLTA SOLA):

```
const  
  VALORE = 3;  
.....  
begin  
  B = 1 * VALORE;    otteniamo 3  
  C = 2 * VALORE;    otteniamo 6  
  D = 3 * VALORE;    otteniamo 9  
end;
```

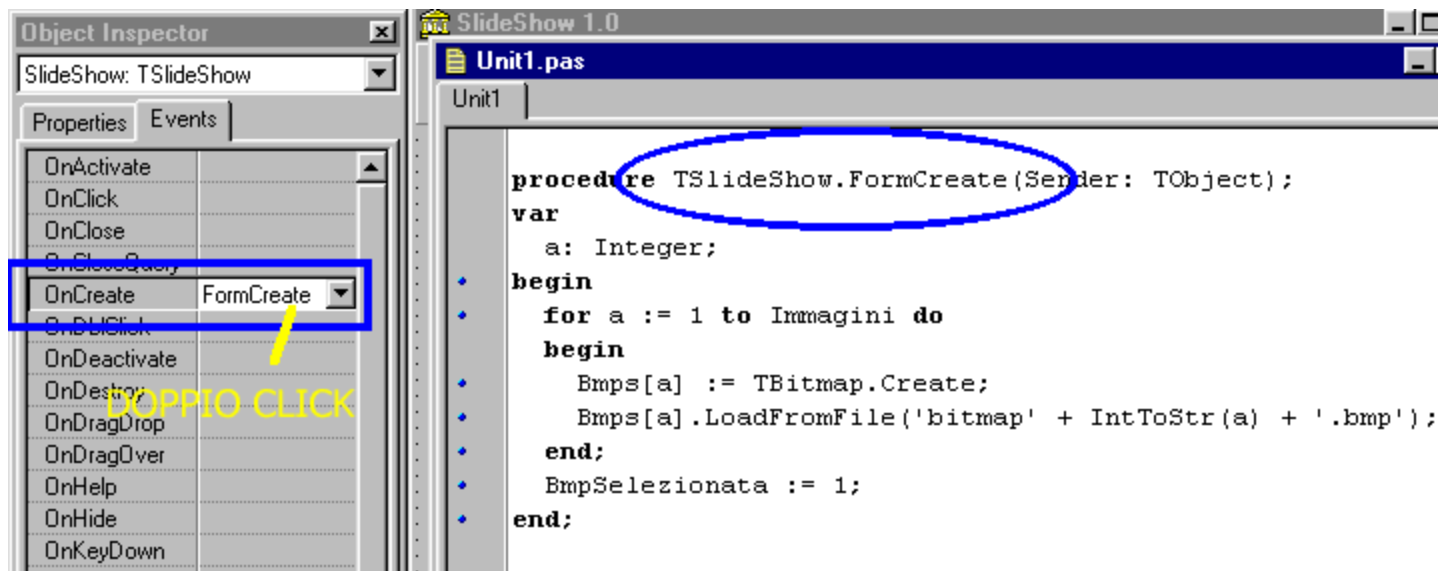
Nella sezione "Private" aggiungere queste linee:

```
Bmps: Array[1..NumBmps] of TBitmap;  
BmpSelezionata: Integer;
```

Cioè "Bmps" è uno schieramento di (5) immagini .BMP (tipo TBitmap), mentre BmpSelezionata è un numero intero.

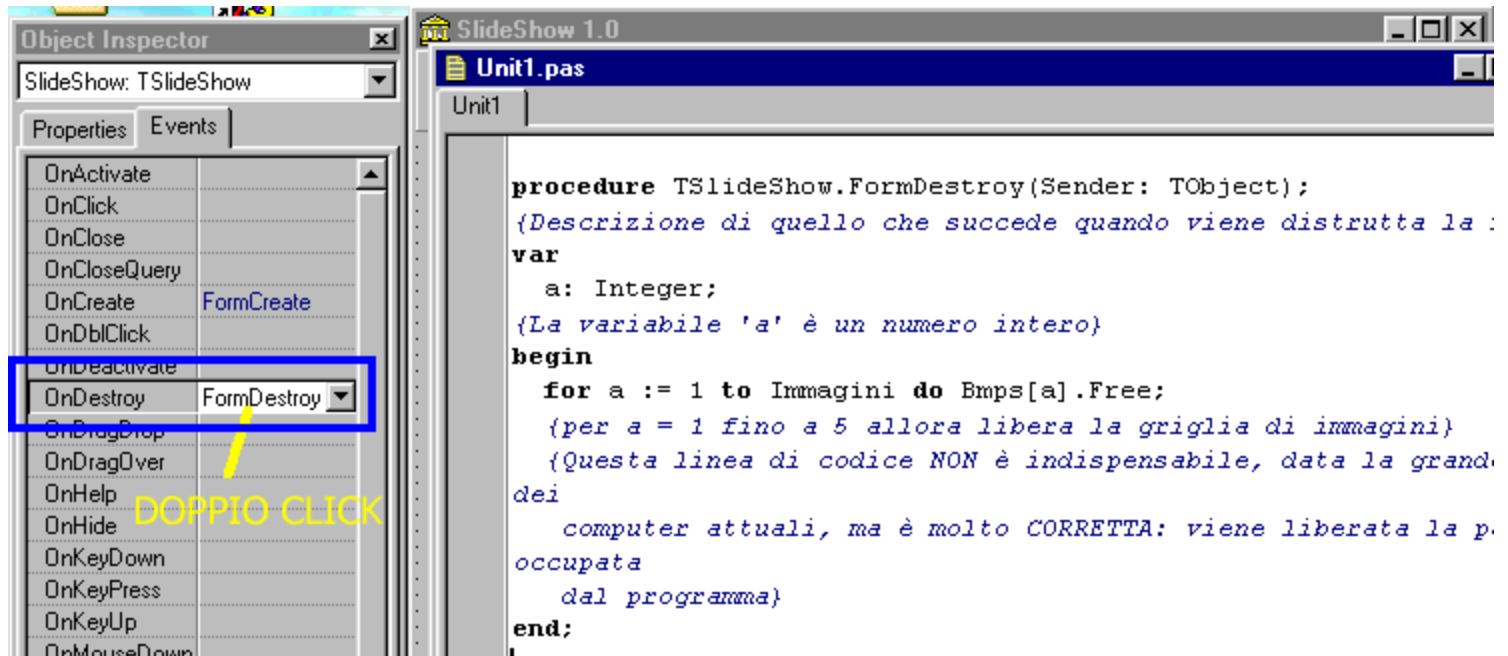
5. F12 per tornare alla maschera. Click semplice sulla form:
scrivere questo codice per l'evento OnCreate della form (doppio click sullo spazio vicino a OnCreate):

```
procedure TSlideShow.FormCreate(Sender: TObject);  
{Descrizione di quello che succede quando viene creata la finestra}  
var  
  a: Integer;  
{La variabile 'a' è un numero intero}  
begin  
  for a := 1 to Immagini do  
    {per a = 1 fino a 5 allora...}  
    begin  
      Bmps[a] := TBitmap.Create;  
      {crea lo spazio per accogliere l'immagine con il numero 'a'}  
      Bmps[a].LoadFromFile('bitmap' + IntToStr(a) + '.bmp');  
      {carica l'immagine con il nome 'BITMAP' + 'a' + '.BMP'}  
    end;  
    BmpSelezionata := 1;  
end;
```



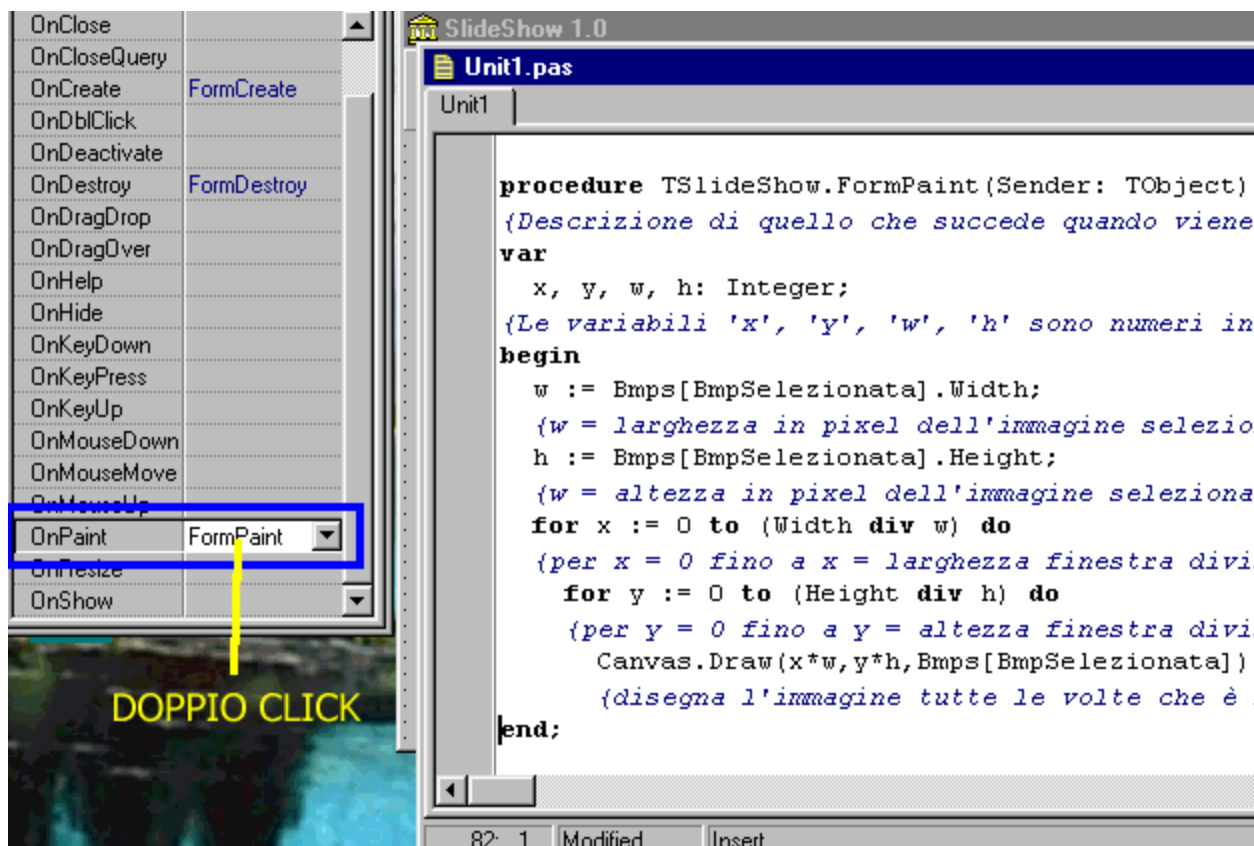
6. Scrivere questo codice per l'evento OnDestroy della form (doppio click sullo spazio vicino a OnDestroy):

```
procedure TSlideShow.FormDestroy(Sender: TObject);
{Descrizione di quello che succede quando viene distrutta la finestra}
var
  a: Integer;
{La variabile 'a' è un numero intero}
begin
  for a := 1 to Immagini do Bmps[a].Free;
  {per a = 1 fino a 5 allora libera la griglia di immagini}
  {Questa linea di codice NON è indispensabile, data la grande potenza e memoria dei
   computer attuali, ma è molto CORRETTA: viene liberata la parte di memoria occupata
   dal programma}
end;
```



7. Scrivere questo codice per l'evento OnPaint della form (doppio click sullo spazio vicino a OnPaint):

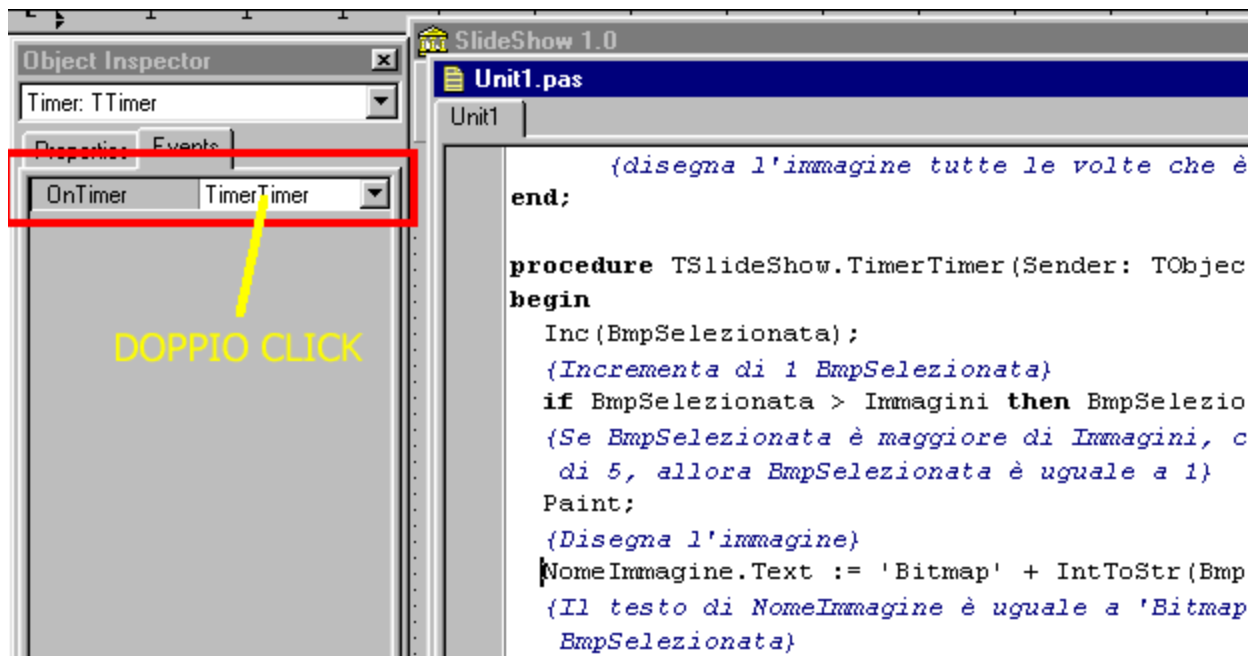
```
procedure TSlideShow.FormPaint(Sender: TObject);
{Descrizione di quello che succede quando viene ridisegnata la finestra}
var
  x, y, w, h: Integer;
{Le variabili 'x', 'y', 'w', 'h' sono numeri interi}
begin
  w := Bmps[BmpSelezionata].Width;
  {w = larghezza in pixel dell'immagine selezionata}
  h := Bmps[BmpSelezionata].Height;
  {h = altezza in pixel dell'immagine selezionata}
  for x := 0 to (Width div w) do
  {per x = 0 fino a x = larghezza finestra diviso larghezza immagine selezionata allora...}
    for y := 0 to (Height div h) do
    {per y = 0 fino a y = altezza finestra diviso altezza immagine selezionata allora...}
      Canvas.Draw(x*w, y*h, Bmps[BmpSelezionata]);
    {disegna l'immagine tutte le volte che è necessario per riempire la finestra}
end;
```



8. Premere F12 e click semplice sul componente "Timer"; scrivere il seguente codice per l'evento OnTimer:

```

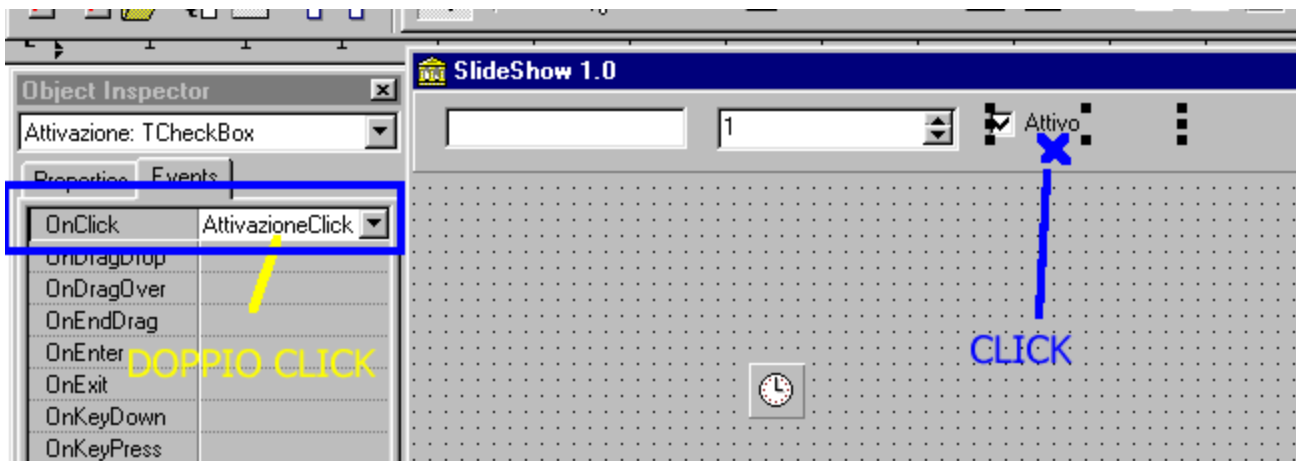
procedure TSlideShow.TimerTimer(Sender: TObject);
begin
    Inc(BmpSelezionata); {Incrementa di 1 BmpSelezionata}
    if BmpSelezionata > Immagini then BmpSelezionata := 1;
    {Se BmpSelezionata è maggiore di Immagini, cioè maggiore di 5, allora BmpSelezionata = 1}
    Paint; {Disegna l'immagine}
    NomeImmagine.Text := 'Bitmap' + IntToStr(BmpSelezionata);
    {Il testo di NomeImmagine è uguale a 'Bitmap' + DaInteroAStringa di BmpSelezionata}
end;
  
```



Salvare, quindi F9 per provare: il programma già funziona, ma non possiamo ancora fermare e riavviare la visualizzazione, nè determinarne l'intervallo.

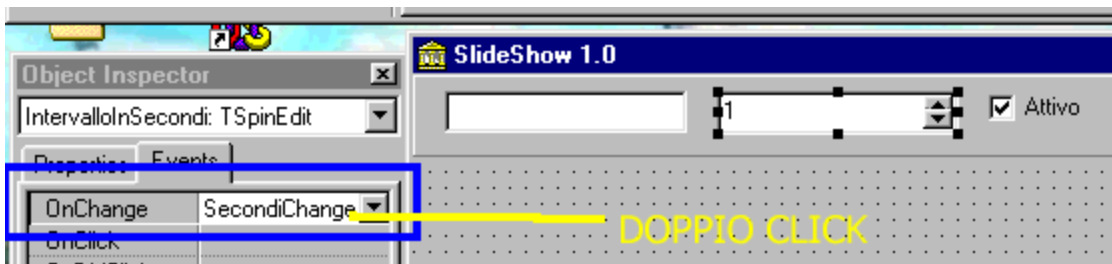
9. Premere F12 e click semplice sul componente "Attivazione"; scrivere il seguente codice per l'evento OnClick:

```
procedure TSlideShow.AttivazioneClick(Sender: TObject);
begin
  If Attivazione.Checked = True then    {Se il pulsante è selezionato allora...}
  begin
    Timer.Enabled := True;    {... Timer viene attivato}
  end;
  If Attivazione.Checked = False then    {Se il pulsante NON è selezionato allora...}
  begin
    Timer.Enabled := False;    {... Timer viene disattivato}
  end;
end;
```



10. Premere F12 e click semplice sul componente "IntervalloInSecondi"; scrivere il seguente codice per l'evento OnChange:

```
procedure TSlideShow.IntervalloInSecondiChange(Sender: TObject);
{Cosa succede quando CAMBIA il valore di IntervalloInSecondi?}
begin
  Timer.Interval := IntervalloInSecondi.Value * 1000;
  {Il valore "Interval" di Timer, cioè l'intervallo di
  visualizzazione tra le immagini, è uguale al
  valore di IntervalloInSecondi per 1000}
end;
```



Salvare, quindi F9 per provare.