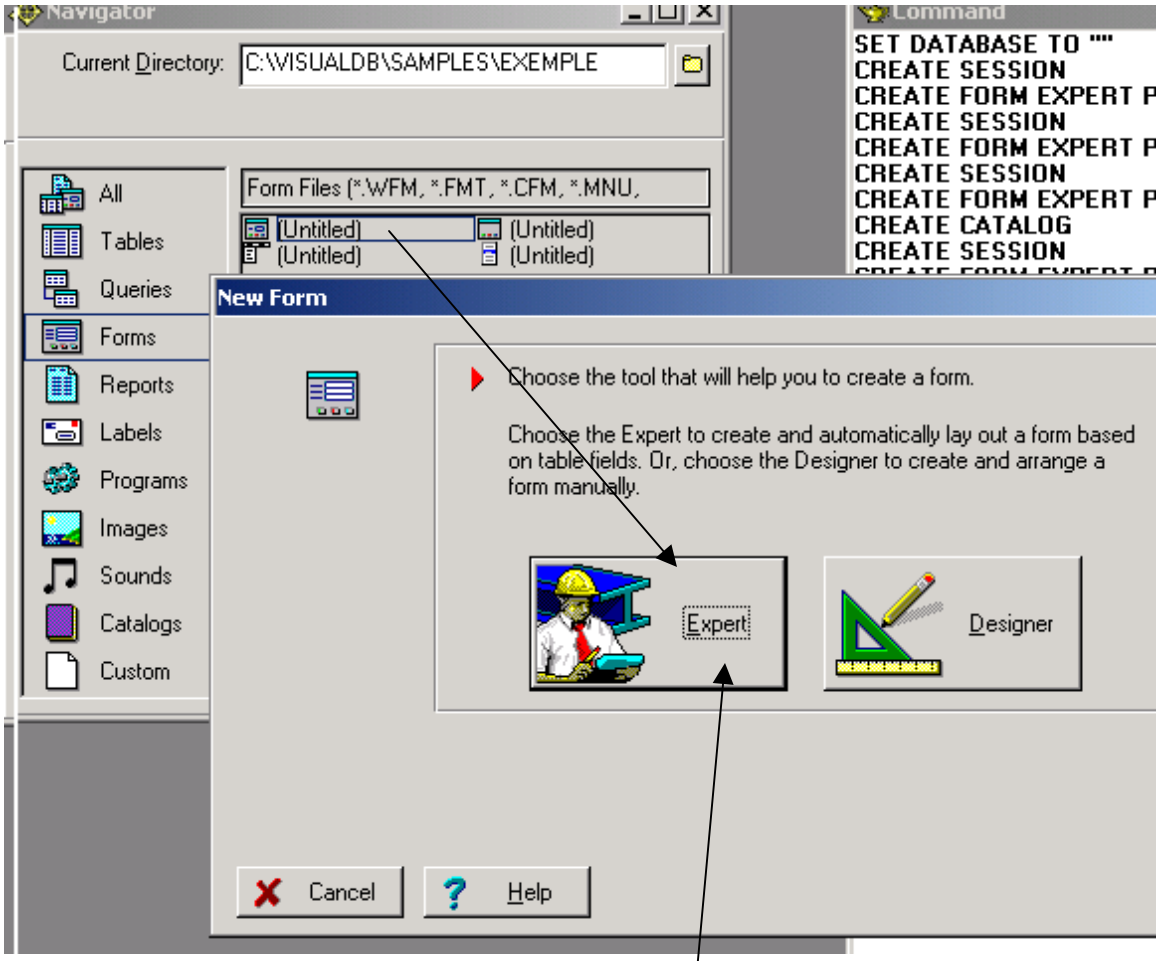
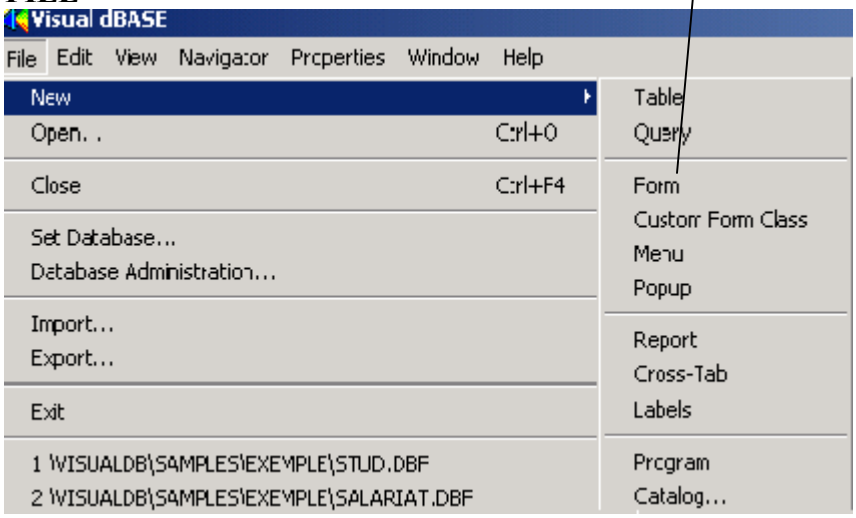
7. UTILIZARE DESIGNER PENTRU PROIECTARE INTERFETE GRAFICE

7.1. Proiectare FORM



FILE



In Visual dBase există un Designer (wizard) pentru proiectarea interfețelor grafice, care ajută la realizarea unor programe ce folosesc **obiecte standard Windows** şi utilizează **programarea orientată pe evenimente**. Până acum s-a folosit programarea clasică care utilizează comenzile procedurale pentru a selecta diferite funcții (DO CASE). Interfața grafică permite utilizarea obiectelor TEXT la care se pot alege diferite dimensiuni, fonturi, culori. Designer-ul este un generator de cod de program dBase, care defineşte o clasă FORM pe care se gasesc obiecte derivate din alte clase Windows (Text, EntryField, PushButtone, RadioButton, ListBox, ComboBox, Browse, Image,...). Programatorul proiectează o machetă de formular (FORM) pe care plasează obiectele de pe o **Paletă de obiecte,** pe care le personalizează modificându-le proprietățile cu ajutorul unei **Palete de proprietăți** afişate de **Inspector** (obiect al designer-ului).

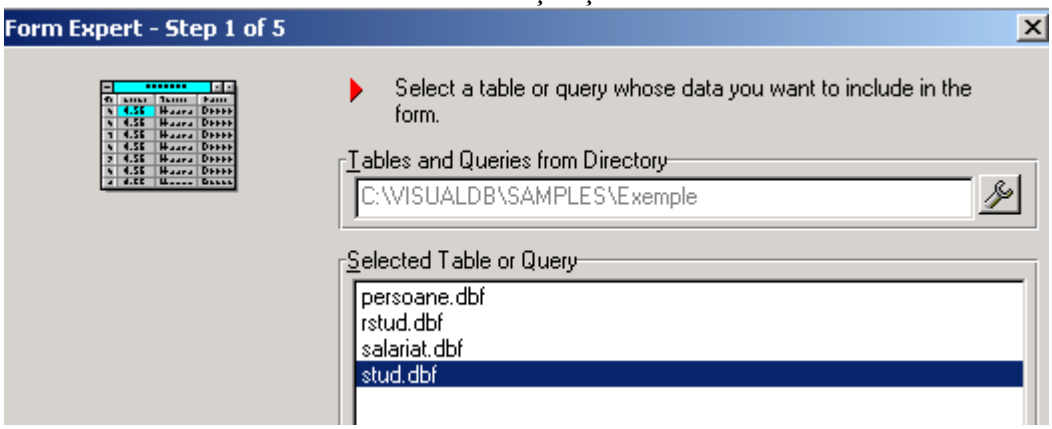
Activarea Designer-ului se poate face:

- Prin selectare FORMS în Navigator şi DblClick pe Untitled
- Din meniul FILE se selectează NEW şi apoi FORM

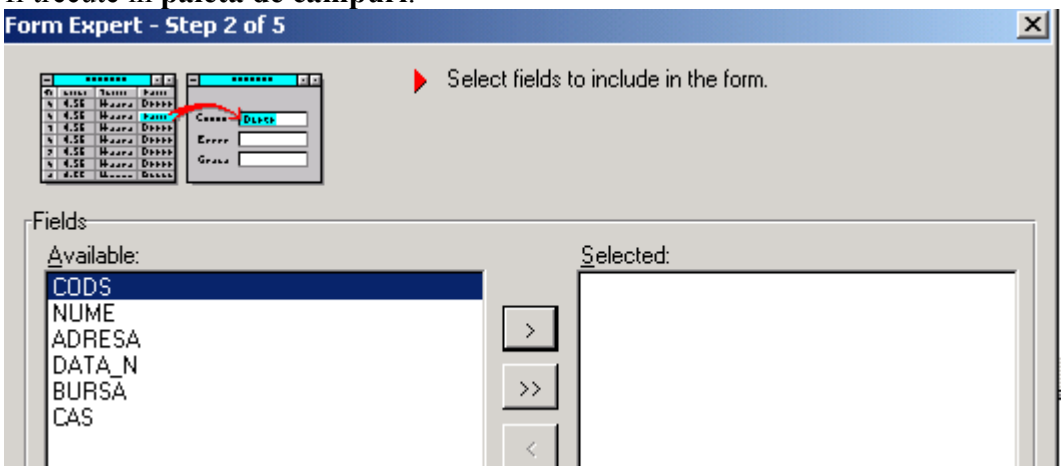
Se va afişa **meniul New Form** în care butonul DESIGNER deschide un Form gol pe care se pot completa obiecte de către programator cu o asistență minimă.

Se va selecta butonul **EXPERT** care permite legarea Form-ului de un fişier şi referirea directă la câmpurile. Acest meniu are 5 paşi.

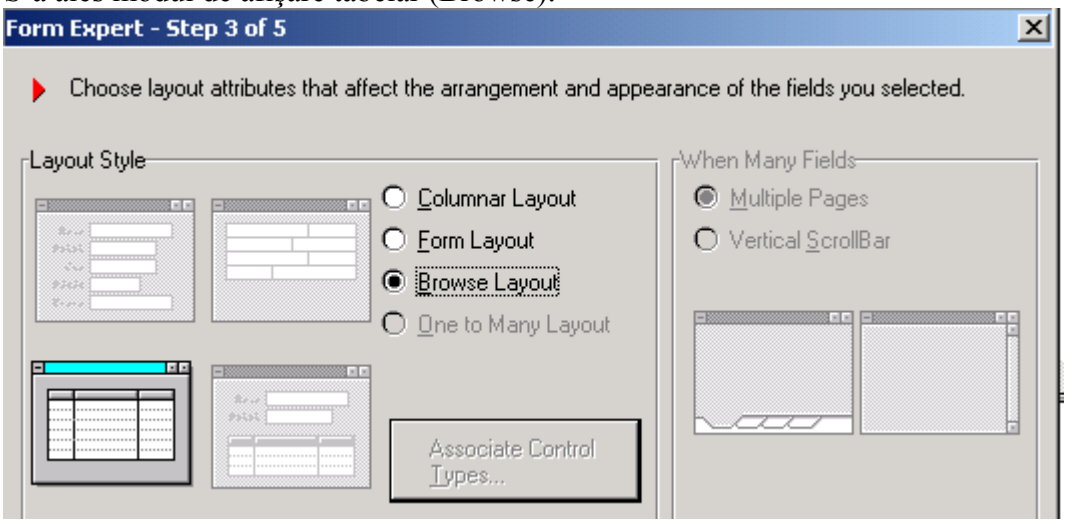
Pasul 1 selectează directorul curent şi fişierul asociat Form-ului



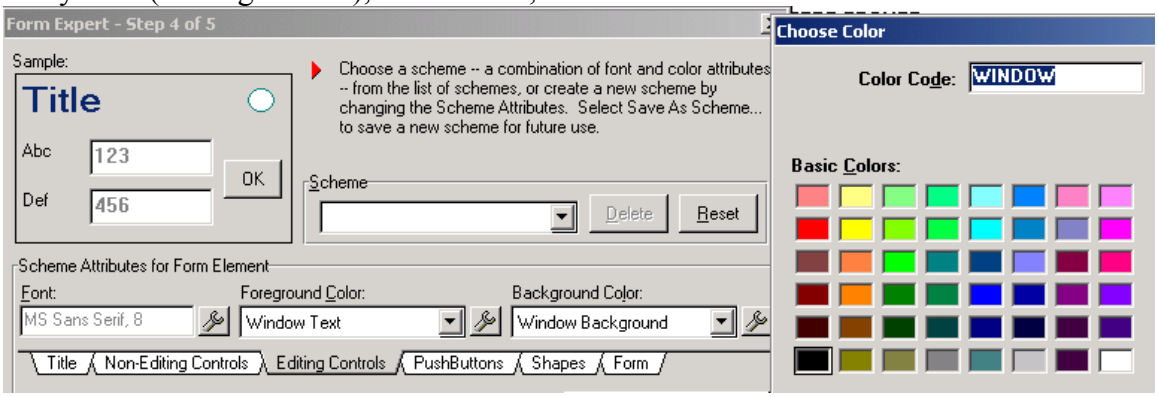
Pasul 2 permite selectarea cu butoane a câmpurilor care vor fi utilizate in aplicație şi vor fi trecute în **paleta de câmpuri**.



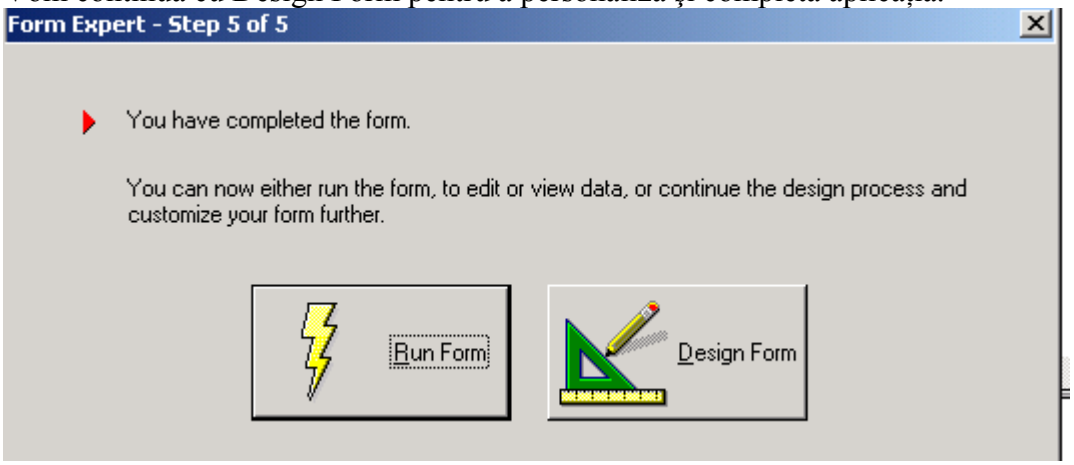
Pasul 3 permite alegerea modului de afişare a înregistrărilor din fişier pe Form.
S-a ales modul de afişare tabelar (Browse).



Pasul 4 permite alegerea culorilor şi fonturilor pentru Titlu, pentru Text(Non Editing), EntryField (Editing control), PushButton, Form.

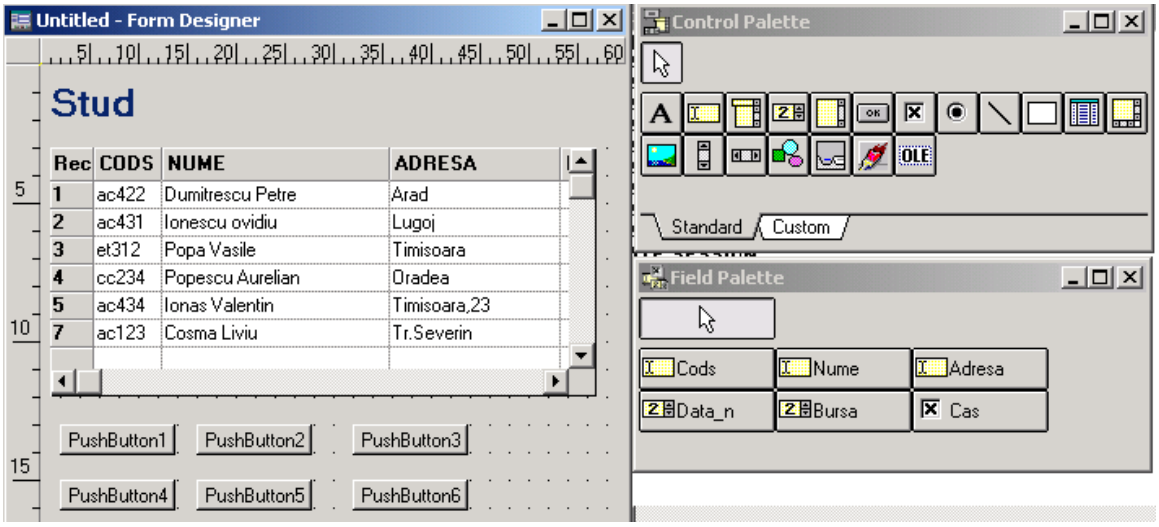


Pasul 5 permite generarea, compilarea şi rularea programului (Run Form) sau continuarea proiectării Form-ului pentru a plasa obiecte pe el(Design Form).
Vom continua cu Design Form pentru a personaliza şi completa aplicația.



Pe ecran apare macheta într-o forma generală cu afişarea conţinutului fişierului. Dimensiunile Form-ului şi tabelului de afişare pot fi modificate cu mouse-ul.

Lângă Form se găseşte paleta de obiecte de unde putem selecta cu mouse-ul obiecte pe care să le punem pe Form. S-au plasat pe Form câteva butoane care sunt de aceeaşi dimensiune şi automat denumite cu numele clasei şi un număr. Ele sunt simple figuri inerte care trebuie personalizate prin modificarea proprietăţilor.

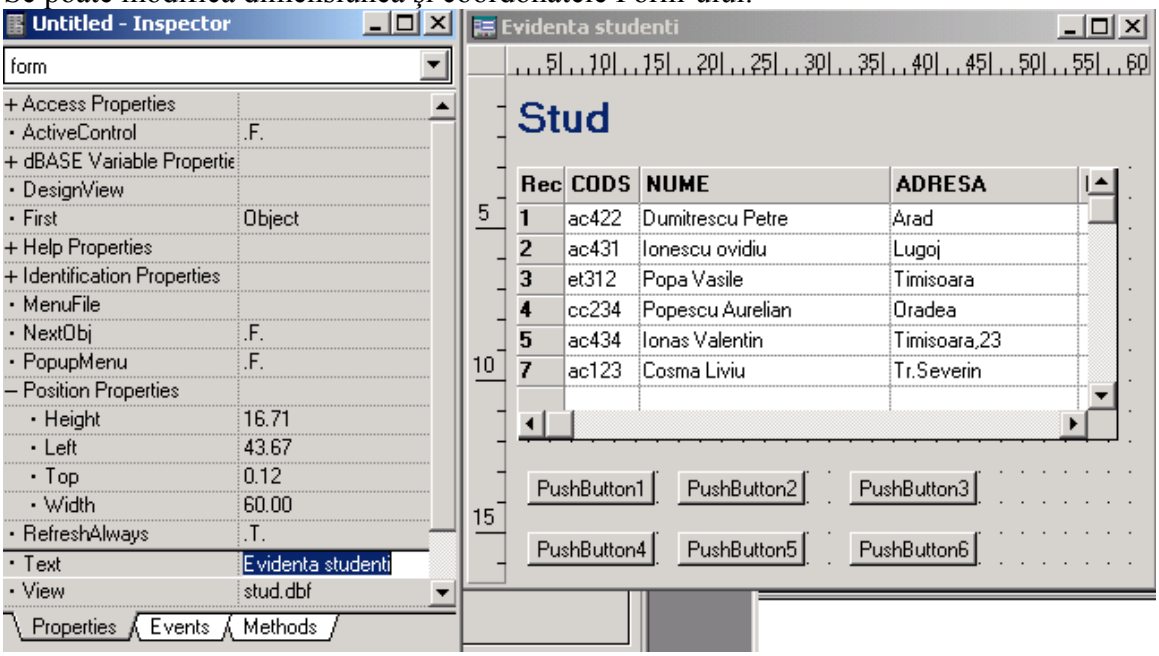


7.2. Modificare proprietăţi obiecte

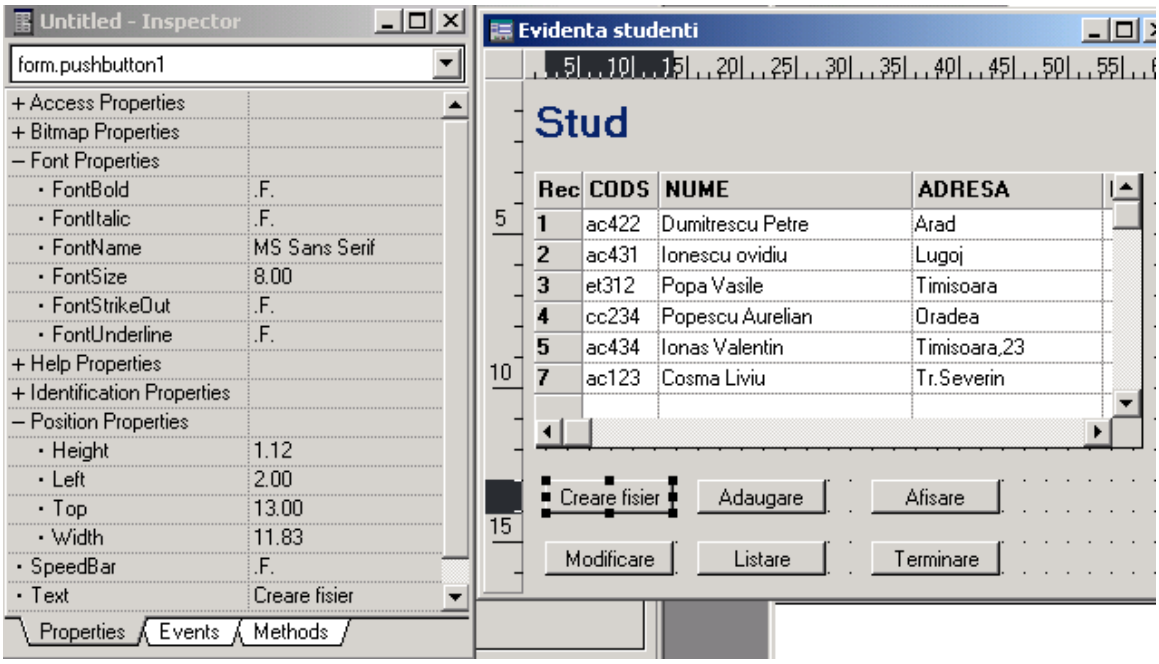
Modificarea obiectelor se face prin selectarea lor cu mouse-ul şi afişarea paletei de proprietăţi folosind unealta **Inspector** care se activează cu **click dreapta** pe Form.

S-a modificat Titlul Form-ului (proprietatea Text) unde s-a introdus Evidenta studenti.

Se poate modifica dimensiunea şi coordonatele Form-ului.



Selecția unui alt obiect de pe Form se poate face prin Click cu mouse-ul pe obiect sau din ComboBoxul din partea de sus din Inspector. Selectăm PushButon1 îi modificăm dimensiunea cu mouse-ul şi textul afişat pe el, culoarea tipul şi dimensiunea font-ului folosit din Inspector. **Grupele de proprietăți se afişează cu + şi se închid cu -.** Dimensiunea şi poziția pot fi modificate şi în paleta de proprietăți din Inspector. La fel se procedează şi cu celelalte butoane.



7.3. Modificare proprietăți comportamentale

În programarea orientată pe obiecte, obiectele au proprietăți care sunt încapsulate şi se pot moşteni la obiectele din subclasele derivate. Proprietățile obiectelor sunt:

- **Proprietăți structurale** (Properties) care descriu obiectul (nume, adresa, data naşterii, salar, culoare, tip font, înălțime, lățime, greutate, distanță,..)
- **Proprietăți comportamentale** care caracterizează **reacția obiectelor la evenimente**. Ele se precizează prin asocierea unei comenzi sau proceduri unui eveniment din tabela Events din Inspector. Evenimentele pot fi click stânga sau dreapta pe mouse (OnLeftMouseDown), deschidere sau închidere Form (OnOpen, OnClose), mişcare mouse pe Form (OnNavigate), la mişcare obiect (OnMove), sau selectare sau deselectare obiect (OnGotFocus, OnLostFocus), schimbare dimensiune (OnSize), la schimbarea unor valori dintr-un EntryField (OnChange).
- **Metodele** care sun proceduri interne ale obiectului care pot fi utilizate în prelucrare. Pentru Form sunt deschidere, închidere sau stergere Form (F1.Open(), F1.Close(), F1.Release()), reafişare Form (F1.Refreash())

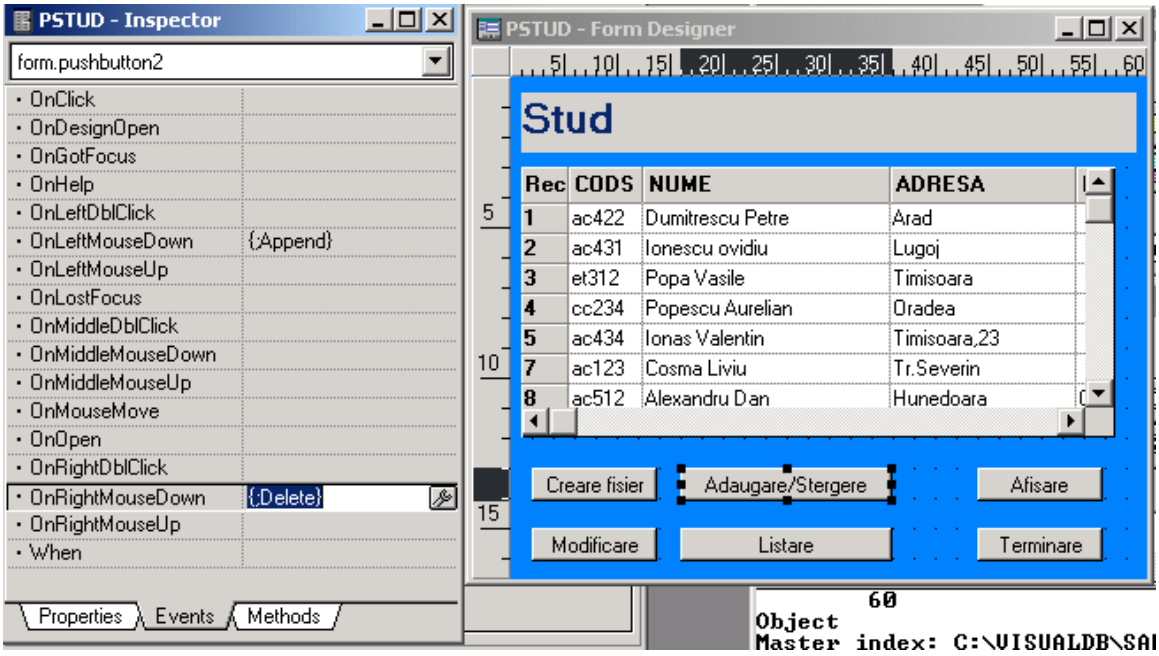
Programarea bazată pe evenimente presupune ataşarea unor proceduri la evenimente. Procedurile se lansează astfel asincron la apariția evenimentului asociat obiectului. Un

obiect poate reacționa diferit la diferite evenimente. Fiecărui eveniment care acționează asupra obiectului (ex.buton) i se poate atașa o procedură distinctă. Deci procedura nu i se atașează obiectului ci evenimentului care acționează asupra acelui obiect.

La proiectarea unei aplicații se recomandă:

- La **OnOpen** pentru Form să se asocieze o procedură în care **să se deschidă toate fişierele** (tabelele) bazei de date.
- La **OnClose** pentru Form **să se închidă toate fişierele** bazei de date (Close Database, sau Clear All)

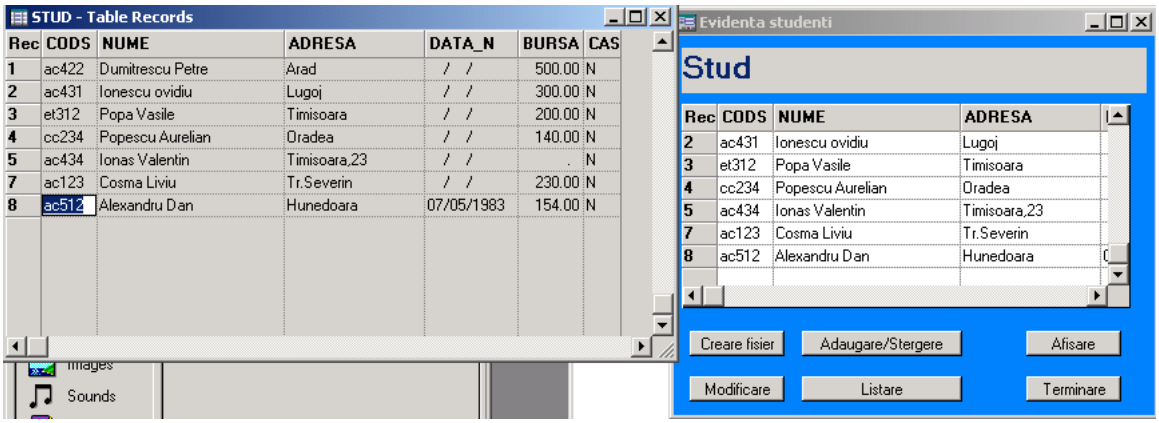
In exemplul prezentat s-au asociat comenzi simple butoanelor Adăugare - Append Afisare - Browse, Modificare – Edit, Listare – List, Terminare – Close database. S-a modificat butonul Adaugare in Adaugare/Stergere şi s-a asociat evenimentului **Click stânga** comanda **Append** iar pentru **click dreapta** comanda **Delete.** Dacă evenimentului i se asociază o singură comandă ea se pune direct în tabelul Inspector sub forma **{;Append}.** Dacă evenimentului i se asociază o procedură, se va da un click pe unealta din dreapta rândului, care va deschide o procedură care va fi scrisă de programator.



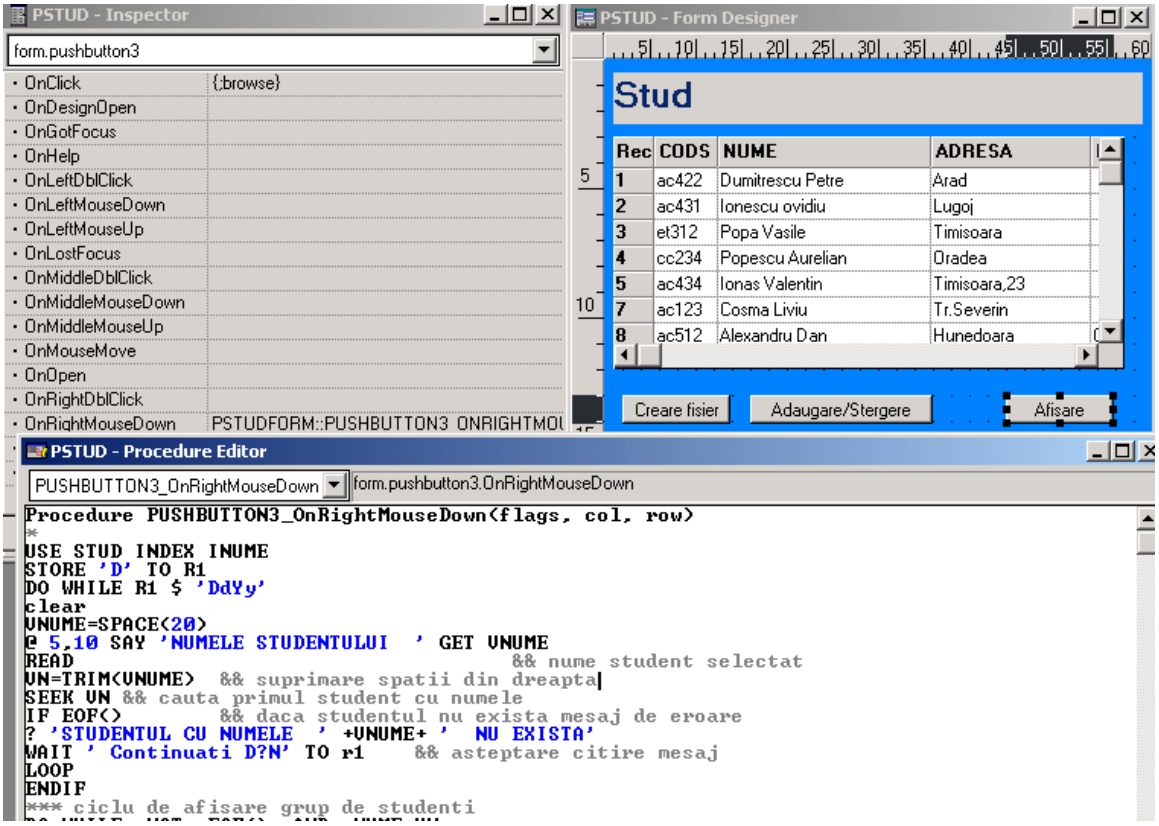
Macheta proiectată cu Design-erul va servi la generarea unei clase obiect Form care conține toate obiectele adăugate. Trecerea din modul Design in **modul Run** se face dând click în colțul din stânga sus şi alegând opțiunea Run Form (sau F2) şi precizând apoi numele programului care va avea extensia **WFM (Windows Form).** Programul va fi generat, compilat şi pus în execuție prin deschiderea Form-ului (Form.Open()). Un Form în execuție poate fi comutat în modul design pentru completare sau corectare dând click în colțul din stânga sus a ferestrei şi alegând Design Form (sau SH+F2).

Pentru butonul Listare vom lăsa List pentru Click stânga şi vom introduce procedura de listare cu cap de tabel din cursul trecut. La fel se poate proceda cu butonul Afişare unde pe click dreapta putem afişa datele unui student dat prin nume şi căutat în

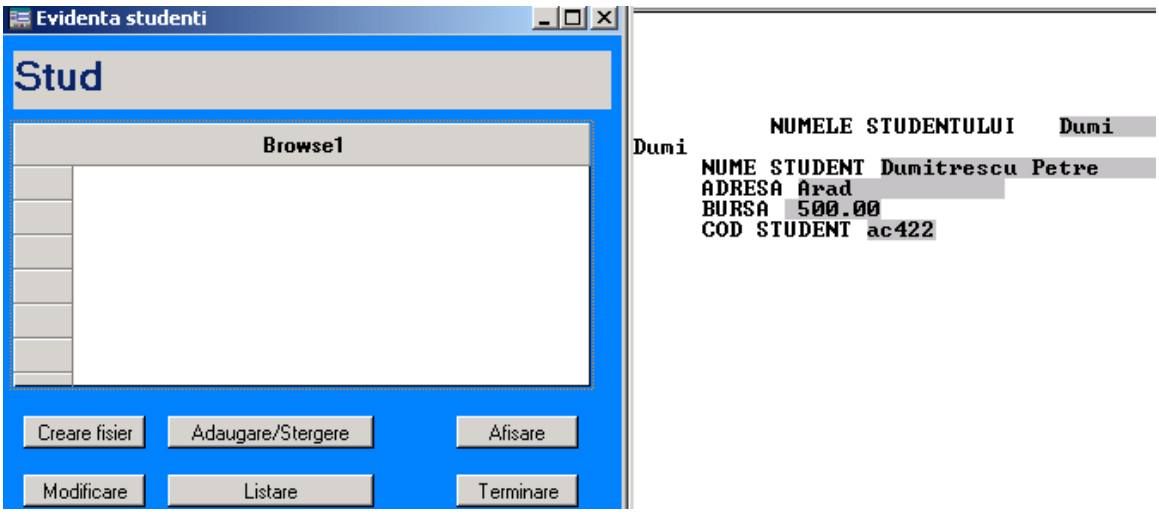
acces direct (prin index Inume). Butonul Creare fişier poate să aibă pe click dreapta procedura de creare care copiază structura fişierului de referinţă Rstud.



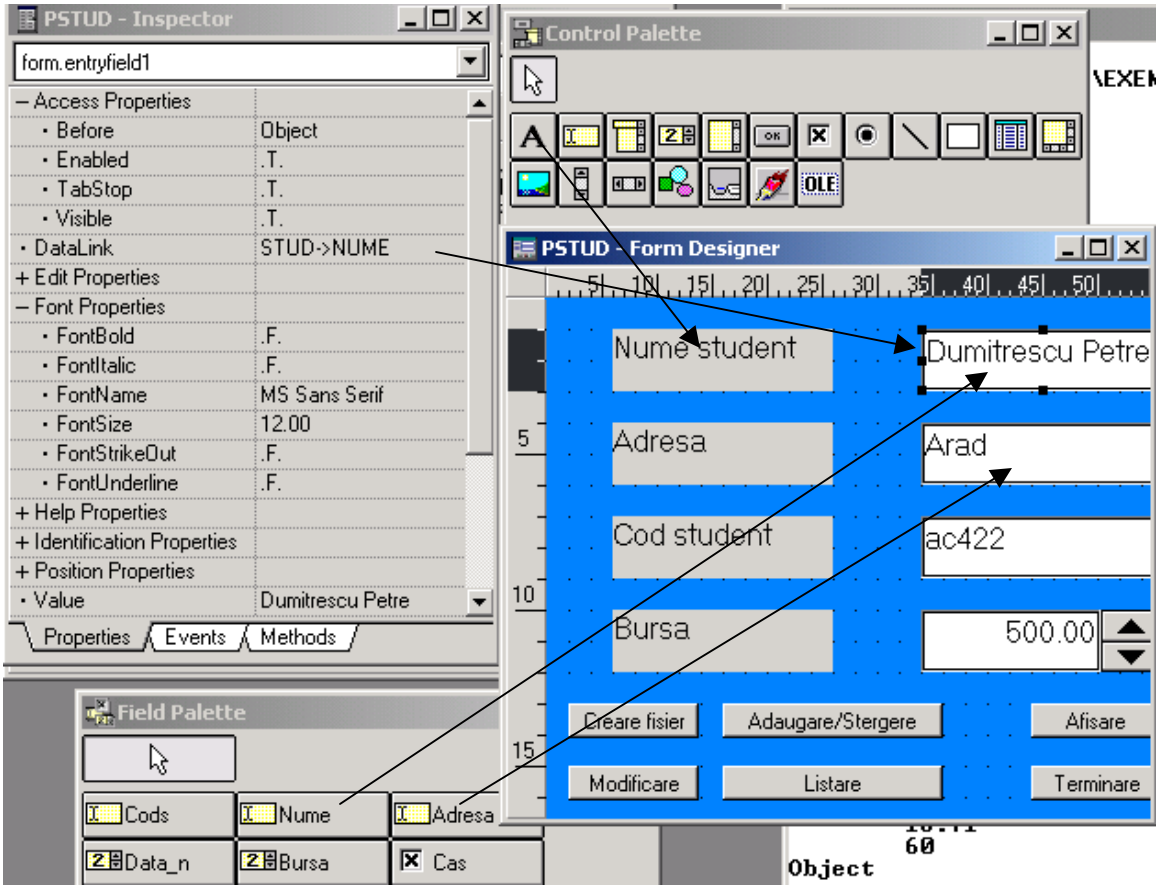
Următoarea figură exemplifica adăugarea unei proceduri de afişare pentru un student dat prin nume pe OnRightMouse Down pentru butonul Afisare.



Din cauză că s-a deschis prin program un fişier Browse-ul din Form nu mai afişează nimic. Din acest motiv nu se recomandă utilizarea variantei EXPERT a Designer-ului, care în plus acceptă un singur fişier deschis, care este dat ca o proprietate **Form.View**. Este preferabilă proiectarea completă a Form-ului de către programator.
Programul generat poate fi afişat şi modificat selectând **Edit Form** pe click dreapta.



Vom modifica programul eliminând browse-ul din Form (selectare şi Del).
Vom afişa datele studentului curent folosind obiecte Text şi EntryField, corespunzătoare clauzelor Say şi Get din Dos. Obiectele text le luăm de pe paleta de obiecte şi le modificăm dimensiunea şi proprietăţile (Text, FontSize, FontName). Obiectele EntryField se iau din paleta de câmpuri care le sunt asociate şi se afişează conţinutul lor. Numele câmpului asociat se trece în proprietatea DataLink.



Putem suprima butonul Terminare, deoarece închiderea fişierelor se poate face pe evenimentul OnClose din Form, iar oprirea programului se face prin închidere Form.

Mai jos se prezintă un exemplu de Form realizat direct cu Designer-ul care generează o clasă Form pe care sunt plasate obiectele Texte, EntryField şi Pushbutton. Programul principal conține numai comanda de creare a unui obiect F şi deschiderea lui.

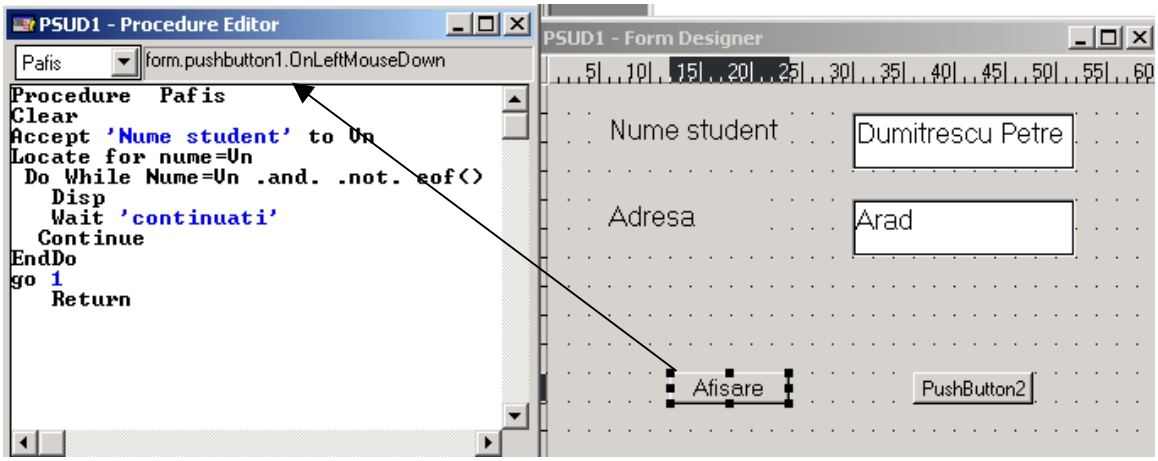
```
local f
Use stud
f = new PSUD1FORM()
  f.Open()
```

In EntryField-uri se va afişa valoarea câmpului asociat din înregistrarea curentă. Pentru aceasta s-a modificat programul generat:

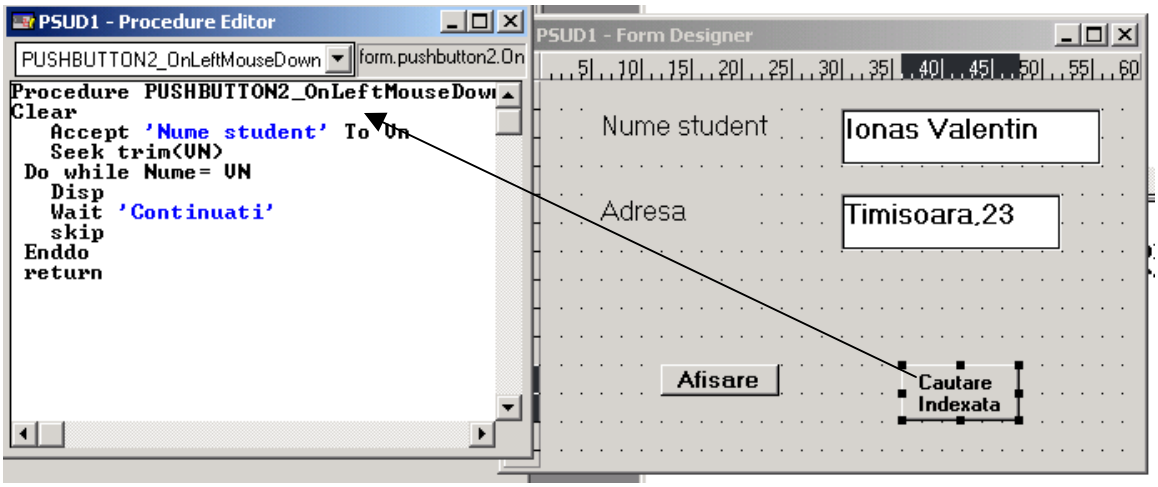
- S-a adăugat la începutul programului principal deschiderea fişierului (Use Stud)
- La descrierea fiecărui EntryField s-a adăugat proprietatea DataLink care îi asociază un cîmp:
  (**DataLink "stud->Nume",;** şi **DataLink "stud->Adresa",;)**
- Pentru butonul afişare se introduce proprietatea:
  OnLeftMouseDown CLASS::PAFIS,;

Parcurgerea secvențială a fişierului se poate face prin butoanele din bara unelte

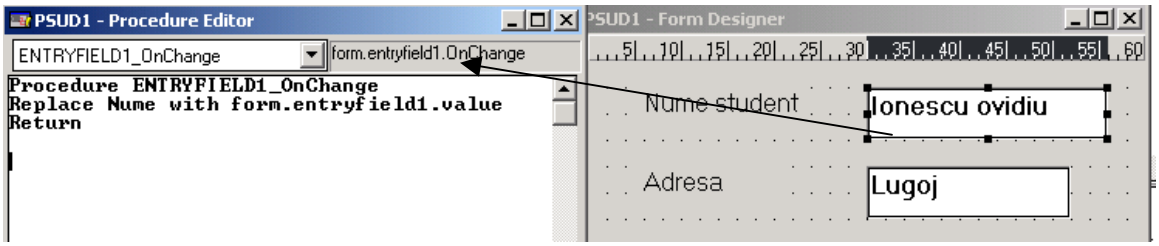
In fiecare moment se va afişa în Form Numele şi Adresa studentului curent selectat. Vom observa că la acționarea butonului Afisare se va afişa şi în Form numele studentului selectat.



In continuare modificăm butonul 2 pentru ai asocia o procedură de căutare în acces direct folosind un index. Vom cere de la consolă numele unui student sau grup de studenți. Vom căuta primul student îl afişăm şi continuăm ciclul cu toți studenții care fac parte din grup (ex.Pop). Fiecare student selectat în acces direct va fi afişat şi în EntryFieldur-ile din Form.



Menționăm că **valorile din obiectele EntryField se pot modifica, dar nu se modifică direct valoarea câmpului asociat din fişier**. Acesta se poate modifica cel mai simplu folosind **evenimentul OnChange** din obiectul EntryField căruia îi asociem o procedură, care înlocuieşte valoarea câmpului cu proprietatea obiectului EntryField.Value, care este specifică obiectelor EntryField.



Asemănător se scrie o procedură pentru modificarea câmpului Adresa dacă se modifică pe ecran valoarea obiectului EntryField Adresa:

Procedure ENTRYFIELD2_OnLeftMouseDown
Replace Adresa whith form.entryfield2.Value
Return

Se observă ca obiectul Text înlocuieşte clauza Say iar obiectul EntryField Get-ul.

In continuare se prezintă programul generat pentru cel de al 2-lea Form.

Se observă definirea **CLASS PSUD1FORM OF FORM** cu proprietățile structurale şi apoi definirea obiectelor ce se plasează pe acest Form.

Fiecare obiect are proprietăți structurale şi comportamentale reacționând prin acțiuni la evenimente prin proceduri ataşate evenimentelor.

Procedurile se plasează în ultima parte a clasei.

Programul principal se găseşte la început şi constă în principiu din 2 instrucțiuni, care creează un obiect Form F1 din clasa definită şi îl deschide.

f = new PSUD1FORM()
f.Open()

```
* Generated on 11/08/2005
*
parameter bModal
local f
Use stud index inume
f = new PSUD1FORM()
if (bModal)
  f.mdi = .F. && ensure not MDI
  f.ReadModal()
else
  f.Open()
endif
CLASS PSUD1FORM OF FORM
  this.OnLeftMouseDown = CLASS::FORM_ONLEFTMOUSEDOWN
  this.Left = 56.666
  this.Text = "Form"
  this.Top = 13.0586
  this.Height = 13.6465
  this.Width = 60

  DEFINE TEXT TEXT1 OF THIS PROPERTY;
       Left 6,   Text "Nume student", Top 1, FontSize 12,;
       FontBold .F., Height 2, Width 18

  DEFINE TEXT TEXT2 OF THIS PROPERTY;
      Left 6,  Text "Adresa", Top 4, FontSize 12,;
 FontBold .F., Height 2, Width 16

  DEFINE ENTRYFIELD ENTRYFIELD1 OF THIS PROPERTY;
      Left 30,  Top 1,   FontSize 12,  Height 2, Width 26;
DataLink "stud->Nume",;
 OnChange CLASS::ENTRYFIELD1_ONCHANGE,;

  DEFINE ENTRYFIELD ENTRYFIELD2 OF THIS PROPERTY;
       , DataLink "stud->Adresa", Left 30, Top 4,;
          FontSize 12, Height 2, Width 22

  DEFINE PUSHBUTTON PUSHBUTTON1 OF THIS PROPERTY;
     OnLeftMouseDown CLASS::PAFIS,;
       Left 12, Text "Afisare", Top 10,  FontSize 11,;
       Height 1.1172,  Group .T., Width 11

  DEFINE PUSHBUTTON PUSHBUTTON2 OF THIS  PROPERTY;
 OnLeftMouseDown CLASS::PUSHBUTTON2_ONLEFTMOUSEDOWN,;
 Left 36,  Text "Cautare Indexata", Top 10, Height 2, Group .T., Width 11
```

```
Procedure  Pafis
Clear
Accept 'Nume student' to Vn
Locate for nume=Vn
 Do While Nume=Vn .and. .not. eof()
   Disp
   Wait 'continuati'
  Continue
EndDo
go 1
   Return

Procedure PUSHBUTTON2_OnLeftMouseDown(flags, col, row)
Clear
   Accept 'Nume student' To Vn
   Seek trim(VN)
 Do while Nume= VN
   Disp
   Wait 'Continuati'
   skip
 Enddo
 return

Procedure ENTRYFIELD1_OnChange
Replace Nume with form.entryfield1.value
Return

ENDCLASS
```