

## 8. PROIECTAREA UNOR APLICAȚII COMPLEXE

### 8.1. Utilizarea prin program a obiectelor grafice standard

Folosind designer-ul se pot crea Form-uri care să conțină mai multe obiecte de tipuri diferite, care se utilizează în aplicație pentru a asigura o interfață flexibilă cu utilizatorul, dar se pot proiecta numai aplicații simple. Proiectarea clasică permite utilizarea **unui singur fișier** care apare ca parametru al obiectului Form pentru proprietatea View (this.View = "stud.dbf"). Deschiderea indexată a fișierului, sau deschiderea altor fișiere prin procedurile programului duce la anomalii. Nu se acceptă alte câmpuri decât cele din fișierul asociat Form-ului, în proprietățile DataLink din obiectele Entryfield și SpinBox, DataSource din ListBox și ComboBox, Fields din Browse.

**Designer-ul crează o singură clasă Form** cu obiectele și proprietățile încapsulate și fixe fără posibilitatea de a avea acces la ele din exterior. Prin crearea unui obiect Form din clasa creată se copiază proprietățile implicite ale clasei. Punerea în funcțiune a programului echivalează cu crearea unui obiect din clasa definită și deschiderea lui.

#### Definirea obiectelor dintr-o clasă existentă

Obiectele de interfață grafică standard existente în Windows pot fi create direct în programe sau proceduri folosind comanda DEFINE.

Orice obiect grafic dintr-o clasa Windows se definește în cadrul unui container care este de regula o fereastră (FORM). Comenzile pentru definire, afișare, modificare se pot da în linia de comandă, sau în program.

Proprietățile obiectelor standard sunt:

- Proprietăți structurale (dimensiune, nume, adresa, culoare, ..)
- Metode – funcții care acționează asupra obiectelor clasei
- Proprietăți comportamentale - subrutine asociate unor evenimente

Se recomandă folosirea documentației disponibile în HELP la comanda DEFINE

Definirea unui obiect dintr-o clasă existentă se face cu comanda DEFINE care are sintaxa

```
DEFINE <nume_clasa> <nume_obiect> [OF <obiect_container>]  
[FROM <linie, coloana> TO <linie, coloana> > | <AT <linie, coloana>]  
[PROPERTY <lista_proprietati>]  
[CUSTOM <custom property list>]
```

Unde <nume\_clasa> este clasa obiectului care se dorește a fi definit.

FROM sau AT se utilizează pentru a preciza poziția obiectului în fereastră.

Lista de proprietăți conține valorile proprietăților specifice obiectului.

Există proprietăți comune pentru obiecte și specifice unei clase.

Obiectele standard sunt următoarele:

<b>Browse</b>	<b>Checkbox</b>	<b>Combobox</b>
DDELink	DDETopic	<b>Editor</b>
<b>Entryfield</b>	<b>Form</b>	<b>Image</b>
Line	<b>Listbox</b>	<b>Menu</b>
Menubar	Object	Ole
Paintbox	<b>Popup</b>	<b>Pushbutton</b>
<b>Radiobutton</b>	Rectangle	Scrollbar
Shape	<b>Spinbox</b>	Tabbox
<b>Text</b>		

### *Definirea unui form*

**DEFINE FORM f1 PROPERTY TEXT ‘Titlul ferestrei’, height 20,width 80**

-- Creează obiectul in memorie

F1.OPEN() -- afișează fereastra deja definita

F1.CLOSE() -- închide fereastra

F1.RELEASE() -- șterge obiectul din memorie

### *Modificarea parametrilor unui obiect existent*

F1.TEXT=’Noul titlu al ferestrei’ --atributul titlului (textul afișat)

F1.HEIGHT=10 --atributul înălțime

F1.WIDTH=10 --atributul lățime

F1.LEFT=5 -- poziție fata de stânga ecran

F1.TOP=2 --număr linie fata de latura de sus ecran

F1.VISIBLE=.F. --afișează sau nu obiectul

F1.ESCEXIT=.T. --se poate închide fereastra cu ESC

### *Tratarea evenimentelor*

F1.ONMOVE={;?’ Se mișcă fereastra!’}

F1.OnLeftMouseDown={; Append}

F1.OnOpen= {; Use stud index inume alias St}

### *Afișarea atributelor*

? F1.TEXT, F1.WIDTH, F1.TOP -- returnează valoarea atributului indicat

### *Asocierea unui fișier la o fereastră*

F1.VIEW=’STUDENTI.DBF’ -- Asociază un fișier ferestrei.

-- care va fi deschis odată cu fereastra

**INSPECT(F1)** -- Afișează lista de proprietăți pentru obiectul F1

## Crearea de obiecte in containerul Form F1

### *Definire obiecte TEXT*

Obiectele Text înlocuiesc afișările normale cu SAY din DOS. Obiectele Text permit o scriere sofisticată, care folosește toate facilitățile sistemului Windows (tip și dimensiune de fonturi, culori) precizate prin proprietăți.

#### **DEFINE TEXT T1 OF F1 PROPERTY TEXT "Nume Student: "**

F1.T1.WIDTH=30            -- lățime text  
F1.T1.HEIGHT=2            -- înălțime text  
F1.T1.BORDER=.T.            -- bordura in jurul textului  
F1.T1.FONTNAME="ARIAL"    -- Nume font folosit  
F1.T1.FONTSIZE=14        -- dimensiune font  
F1.T1.FONTBOLD=.T.        -- utilizare fond Bold  
F1.T1.ColorNormal='GR/GR+' -- culoare verde/galben

### **Definire obiecte Button**

Obiectele Button sunt funcțional texte scrise pe un suport grafic de formă tri-dimensională și au toate proprietățile obiectelor Text. Ele se folosesc mai frecvent pentru comenzi fiindcă sugerează butoanele unui tablou de comandă. Trebuie să precizăm că **pentru orice obiect (Text, EntryField,...) se pot asocia proceduri atașate unor evenimente.**

#### **Define PushButton B1 At 8,5 Prop Text 'Adaugare'**

F1.B1>OnLeftMouseDown={;Append}

### *Definire obiecte ENTRYFIELD*

Obiectele EntryField înlocuiesc Get-urile din varianta DOS și permit o scriere cu fonturi.

#### **DEFINE ENTRYFIELD E1 OF F1**

F1.E1.DATALINK='ST->NUME'    -- CIMP ASOCIAT  
F1.E1.WIDTH=20  
F1.E1.TOP =5  
DEFINE ENTRYFIELD E2 OF F1  
F1.E2.DATALINK='ST->ADRESA'  
F1.E1.TOP =6

In proprietatea **DataLink** se poate da un **nume de câmp sau un nume de variabilă**, a cărei valoare se afișează și poate fi modificată. La modificarea valorii unui EntryField nu se **modifică valoarea câmpului** afișat, care se poate face folosind comanda Replace într-o procedură asociată evenimentului **On Change**.

### **Definire obiecte Combobox**

#### **Define COMBOBOX CB1 of F1 Prop Datasource 'Field STUD->Nume'**

F1.CB1.Datasource ='Field Stud->Adresa' -- schimbă câmpul afișat  
? F1.CB1.value            -- afișează valoarea curentă a obiectului CB1

La acționarea unui obiect Combobox se vor afișa într-o listă valorile câmpului asociat și va putea fi selectată una din valori, care va deveni valoarea curentă a obiectului și va putea fi apelată prin atributul ComboBox.Value (vezi selecția tipului și mărimea fontului în Word). Din acest punct de vedere este asemănător cu EntryField, mai ales că valoarea curentă poate fi dată prin tastare directă.

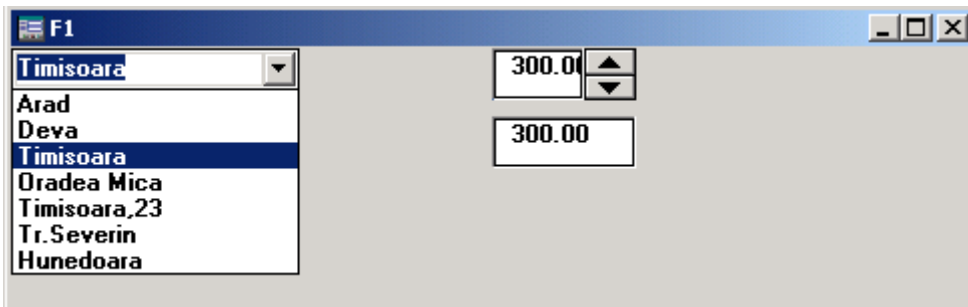
Prin Combobox se selectează o valoare și **nu se face poziționarea pe înregistrarea** care conține valoarea selectată. Este utilă la selectarea numelui complet al unei instituții. Poziționarea pe înregistrarea selectată prin valoarea unui câmp se face cu ListBox.



### Definire obiecte SpinBox

Obiectele SpinBox afișează valoarea curentă a unui câmp sau variabilă numerică, care poate fi modificată prin direct, sau prin butonul de incrementare / decrementare asociat. În rest se comportă ca un EntryField după cum se vede în figură.

**DEFINE SpinBox SB1 OF F1 Property DataLink " Stud->BURSA";  
LEFT 40, TOP 10, WIDTH 12, HEIGHT 1.5**



### Definire obiecte ListBox

Un obiect ListBox este o listă de valori ale unui **singur câmp** din fișierul curent deschis dacă se folosește în proprietatea DataSource argumentul FIELD, sau elementele unui tablou linear dacă argumentul este ARRAY. Nu se recomandă să fie folosit ListBox pentru afișarea unor rezultate.

**ListBox este folosit la selectarea unei înregistrări pe baza valorii unui câmp.**

### DEFINE LISTBOX LB1 OF F1

F1.LB1.DATASOURCE= "FIELD STUDENTI->NUME"

F1.LB1.TOP=3

Obiectele din form se sincronizează. La selecția unei înregistrări din Listbox valoarea obiectelor EntryField se va modifica corespunzător înregistrării pe care s- poziționat.

### Definire obiecte Browse

Obiectul Browse se folosește pentru afișarea sub formă de tabel a valorii unor câmpuri dintr-un fișier, specificate în atributul Fields. Intr-o aplicație în care se selectează un număr de înregistrări printr-o procedură, se recomandă ca înregistrările să fie memorate într-un fișier intermediar de rezultate care să se afișeze prin declararea unui Browse.

### Define BROWSE BR1 of F1 Prop Text 'Evidenta Studenti ',Height 20,; Width 40, Left 15,Top 2, Fields ' Nume, Adresa, Bursa'

Intr-un Browse se poate specifica o listă de câmpuri din mai multe fișiere deschise simultan în zone de lucru diferite, care sun legate între ele prin Set Relation. Dacă se consideră fișierul de studenți Stud și mașini cu aliasul MS putem defini un Browse care să afișeze atât datele personale cât și ale mașinii.

#### \* Programul Prel.wfm

Set Talk off

Clear all ; sterge toate obiectele și inchide fisierele

Use stud in 1

Use masini in 2 Index INrm alias MS

Set Relation to NRM Into MS

Defi form f1 prop width 120

F1.Open()

Define BROWSE BR2 of F1 Prop ;

Text "Evidenta Studenti " ,Height 20,;

Width 100, Left 15,Top 2,;

Fields ' Nume, Adresa, Bursa, MS->NRM, MS->TIP, MS->AN\_F,MS->putere'

\* wait

Return

Rec	NUME	ADRESA	BURSA	NRM	TIP	AN_F	PUTERE
1	Dumitrescu Petre	Arad	500.00	ar1	bmw	2000	110
2	Ionescu ovidiu	Deva	300.00	tm1	dacia	1996	60
3	Popa Vasile	Timisoara	200.00	tm3	LOGAN	2004	75
4	Popescu Aurelian	Oradea Mica	140.00	tm2	audi	2001	85
5	Ionas Valentin Gh	Timisoara,23	.	tm5	OPEL	1993	85
7	Cosma Liviu	Tr.Severin	230.00	ar2	mercedes	2001	120
8	Alexandru Dan	Hunedoara	154.00	ar3	dacia	2002	70

### Proprietati comune:

Property	Event	Method	
before	id	OnLostFocus	release()
borderStyle	left	OnLeftMouseDown	setFocus()
enabled	mousePointer	onLeftDblClick	Open()
fontBold	name	onRightMouseDown	Close()
fontItalic	pageno	onLeftMouseUp	Release()
fontName	parent	onMouseMove	Copy()
fontSize	printable	onOpen	Paste()
fontStrikeout	statusMessage	onRender	Cut()
fontUnderline	tabStop	onRightDblClick	Undo()
form	text	onRightMouseDown	Print()
height	top	onRightMouseUp	
ColorNormal 'r/b '	visible	when move()	
ColorHighlight	value	onGotFocus	
helpFile	valid conditie	onLostFocus	
helpId hWnd	ValidErrorMsg	OnChange	
movable			
sizeable			

### 8.3. Crearea unor aplicații complexe folosind Designer-ul

Dacă se folosește în exclusivitate **Designer**-ul se pot realiza doar **aplicații simple cu un singur fișier** deschis specificat în atributul View din Form. Scrierea manuală a programelor prin definirea directă a tuturor obiectelor de interfață grafică este laborioasă. La aplicații complexe se poate combina utilizarea Designer-ului cu definirea și manipularea directă dinamică a obiectelor în proceduri. Se evită astfel operațiile de rutină și se acționează direct asupra obiectelor pentru a realiza operații mai complicate.

În continuare se va prezenta o metodă prin care se pot utiliza dinamic mai multe fișiere folosind Designer-ul, dar fără a folosi proprietatea View din Form.

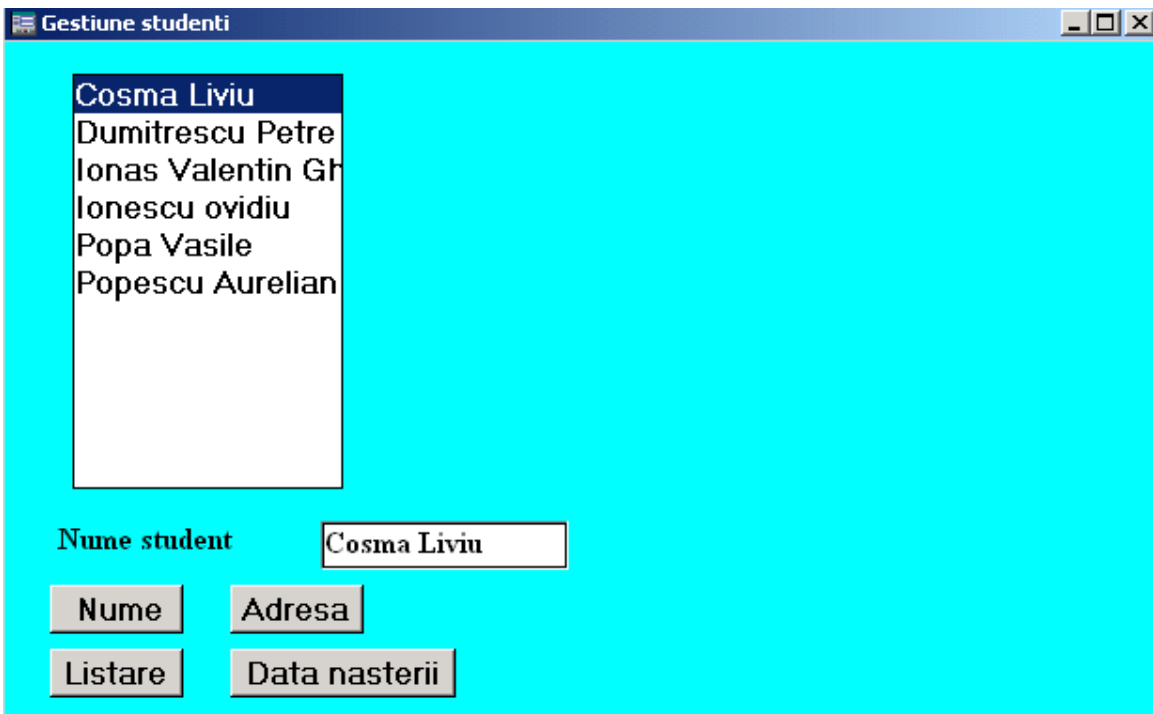
- Se va genera un Form care să conțină mai multe obiecte
- În procedura asociată evenimentului OnOpen se vor deschide fișierele de date și index și se vor face setări (Set Talk off, Clear, Set Date to DMY, Set Decimal To) și se vor defini relațiile dintre fișiere (Set Relation)
- Obiectelor li se vor atașa proceduri prin evenimente.
- În cadrul procedurilor se vor atașa proprietăți de legătură cu fișierele pentru obiectele de pe Form (DataLink din obiectele Entryfield și SpinBox, DataSource din ListBox și ComboBox, Fields din Browse)
- În proceduri se vor defini și șterge obiecte de tip ListBox și Browse
- În proceduri se pot chema alte programe care se găsesc pe alte Form-uri

Se va exemplifica în continuare cu o aplicație, în care pe un Form la care nu s-a asociat nici un fișier se pun cu Designer-ul Texte, Butoane, Entryfield și ListBox pentru

care nu se precizează atributele DataLink și respectiv DataSource. La deschiderea Form-ului pe OnOpen se deschide fișierul STUD cu indexul Inume, încât câmpurile acestuia vor fi recunoscute în continuare și se poate consulta în acces direct. Închiderea fișierului se va face pe evenimentul OnClose la închiderea Form-ului. Atributele DataLink și DataSource se vor adăuga prin procedurile asociate obiectelor pe evenimentul Click stânga din EntryField și respectiv ListBox. La deschiderea Form-ului Entryfield-ul și ListBoxul rămân albe și se vor completa dacă dăm click pe ele. În acest fel nu trebuie intervenit în codul generat de Designer ci numai în procedurile asociate evenimentelor atașate obiectelor.

```
Procedure ENTRYFIELD1_OnLeftMouseDown(flags, col, row)  
form.entryfield1.datalink='stud->nume'  
return
```

```
Procedure LISTBOX1_OnLeftMouseDown(flags, col, row)  
form.listbox1.datasource='field stud->nume'  
return
```



Prin butoanele **Nume**, **Adresa**, **Data nașterii** s-au asociat proceduri care schimbă câmpul asociat obiectului Entryfield prin DataLink. La selecția unui student din ListBox se va afișa în EntryField câmpul Nume, Adresa sau Data nasterii.

```
Procedure PUSHBUTTON1_OnLeftMouseDown(flags, col, row)  
Form.text1.text='Nume student'  
Form.Entryfield1.datalink=' stud->Nume'  
Return
```

```

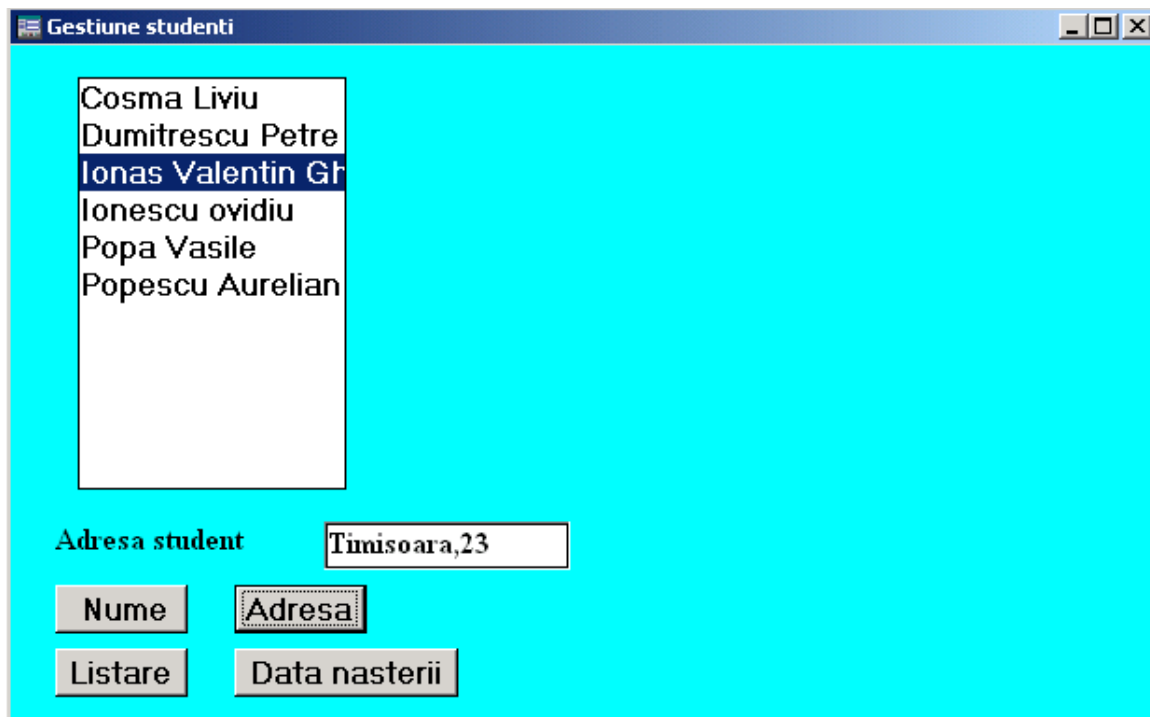
Procedure PUSHBUTTON2_OnLeftMouseDown(flags, col, row)
  Form.text1.text='Adresa student'
  Form.Entryfield1.datalink= ' stud->Adresa'
  Return

```

```

Procedure PUSHBUTTON3_OnLeftMouseDown(flags, col, row)
  Form.text1.text='Data nasterii'
  Form.Entryfield1.datalink= ' stud->Data_n'
  Return

```



Prin butonul **Listare** se asociază o procedură care definește un Browse pe Form și se afișează câmpurile Nume, Adresa, Data\_n din fișierul Stud. Dând click dreapta pe același buton se șterge obiectul Browse. Obiectele create în afara clasei trebuie șterse înainte de relansarea programului sau cu Clear All în procedura asociată la OnOpen.

```

Procedure PUSHBUTTON4_OnLeftMouseDown(flags, col, row)
  Form.whidth =120
  Defi browse br1 of form property Fields 'nume, adresa, bursa.data_n',;
  height 20, width 80,left 35, top 0,FontSize 11
  * Form.br1.OnrightMouseDown =form.br1.release()
  Return

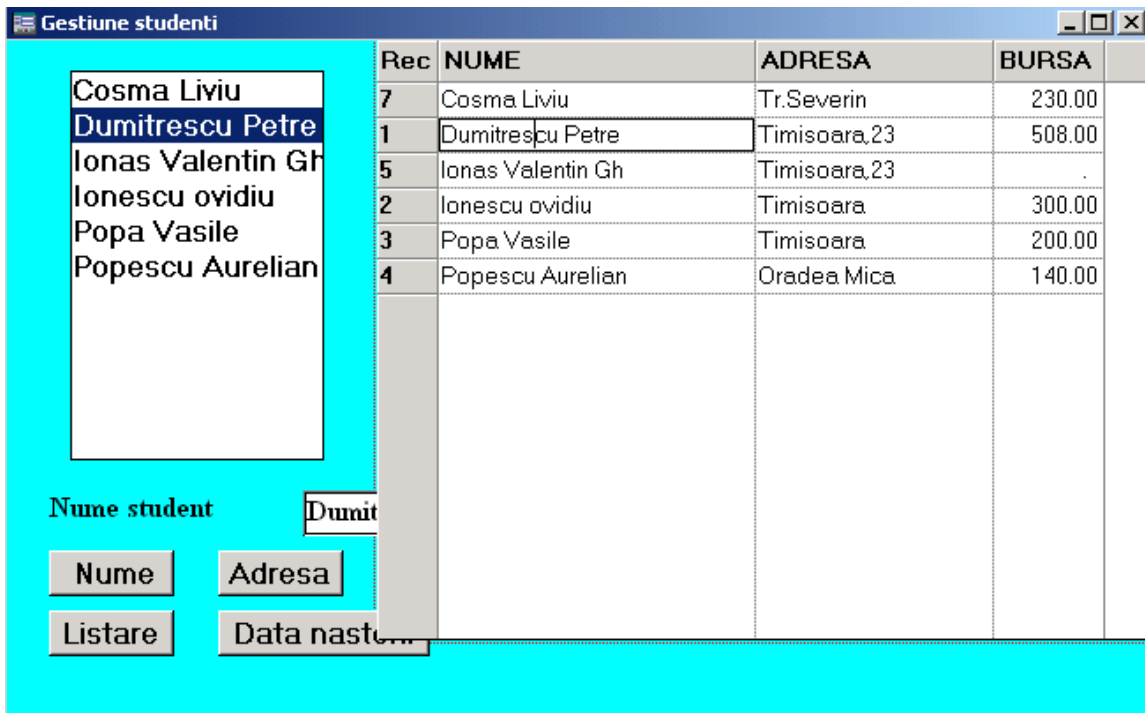
```

```

Procedure PUSHBUTTON4_OnRightMouseDown(flags, col, row)
  Form.br1.release()
  return

```





Se observă că toate obiectele s-au poziționat pe același student indiferent de modul în care s-a făcut poziționarea (prin ListBox, Browse, Go, Seek, Locate).

**Definirea și deschiderea unei noi ferestre** dintr-o procedură în care să se afișeze datele despre mașina studentului selectat.

Vom defini un nou buton Masina la care vom atașa procedura, care definește o nouă fereastră F2 pe care vom afișa datele mașinii. Mașina va fi identificată printr-o legătură de Set Relation între fișierul Stud și Masini în procedura atașată la OnOpen unde se deschid și cele 2 fișiere indexate. Dacă dorim ca datele mașinii să fie permanent afișate vom atașa la OnOpen din Form-ul principal procedura următoare, care definește și Formul F2 ce conține obiecte Text și EntryField referitoare la mașina studentului curent selectat.

#### **Procedure Form\_OnOpen**

**Set Talk off**

**Set Date to DMY**

**Clear**

**Use Stud in 1 Index Inume**

**Use Masini in 2 Index INrm alias MS**

**Set Relation to NRM Into MS**

**Defi Form F2 Prop width 40, Text " Masini",left 50**

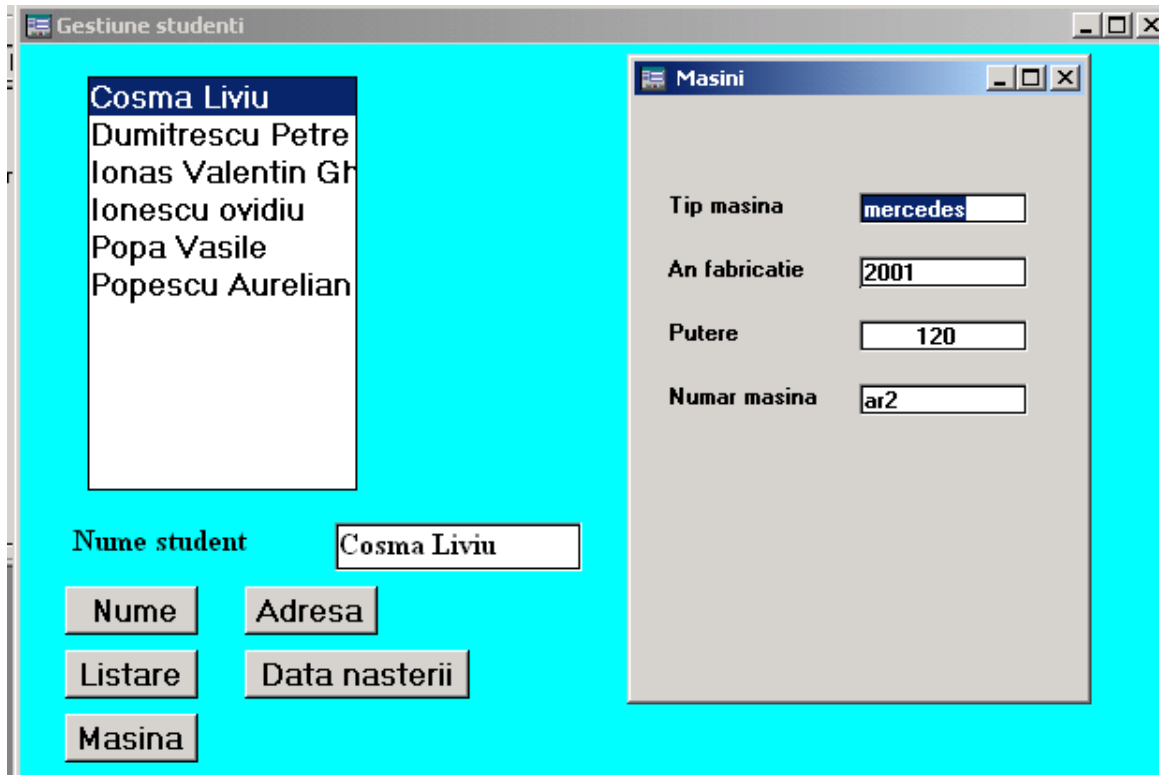
**F2.OnClose = {;f2.release()}**

**Defi TEXT Tip of F2 Prop Text 'Tip masina ',left 3, top 3,width 15**

**Defi TEXT AnFabr of F2 Prop Text 'An fabricatie ',left 3, top 5,width 15**

**Defi TEXT Putere of F2 Prop Text 'Putere ',left 3, top 7,width 15**  
**Defi TEXT Numar of F2 Prop Text 'Numar masina ',left 3, top 9,width 15**  
**Defi EntryField Etip of F2 Prop left 20, top 3, width 15, DataLink 'MS->TIP'**  
**Defi EntryField EAnf of F2 Prop left 20, top 5, width 15, DataLink 'MS->An\_F'**  
**Defi EntryField Etip1 of F2 Prop left 20, top 7, width 15, DataLink 'MS->Putere'**  
**Defi EntryField ENrm of F2 Prop left 20, top 9, width 15, DataLink 'MS->Nrm'**

**F2.Open()**  
**Return**



Folosind această metodă de lucru se poate realiza proiectarea grafică a Form-urilor, care este o muncă de uzură cu Designer-ul, iar în procedurile asociate evenimentelor pentru diferite obiecte se vor preciza elementele de legătură cu fișierele bazei de date.