

Implementarea sistemului informatic pentru agenții bursiere de mărfuri

Lect.dr. Daniel LUCA

Catedra de Informatică Economică, A.S.E., București

În procesele economice în care fluxul de date și complexitatea transformării acestora este mare necesitatea utilizării calculatorului este evidentă. Cum timpul de reacție ridicat reprezintă într-o afacere un atu foarte important, utilizarea calculatorului se impune din ce în ce mai mult. Deținerea informației într-un timp cât mai scurt este pentru un agent bursier cheia succesului. Construit după unul dintre cele mai la modă principii de programare, programarea orientată spre obiecte, soft-ul realizat are avantajele ce decurg din acest fapt: reutilizabilitatea, extensibilitatea, robustețea și programarea în mare. Software-ul propus în acest articol este un punct de plecare în dezvoltarea unei aplicații mai vaste. El tratează o mare parte a problemelor ce apar într-o agenție bursieră.

Cuvinte cheie: sistem informatic, agenție bursieră, SQL, Visual Basic, API, OLE.

1. Introducere

În software-ul elaborat sunt abordate câteva domenii mai interesante cum ar fi: legătura unui program executabil cu baze de date; interogarea bazei de date prin clauze SQL generate de program pe baza cerințelor utilizatorului; posibilitatea transmiterii datelor, informațiilor prin poștă electrică (E-mail); programarea prin controale de comunicație a modemului telefon/fax; generația help-ului sensitiv la context și a hypergraficelor cu puncte fierbinți; generarea rapoartelor de diverse tipuri și forme; integrarea în aplicație a obiectelor de tip OLE (Object Linking and Embedding); folosirea bibliotecilor DLL și a unor controale deosebite dezvoltate de firmele de software. Desigur, mai există multe posibilități de dezvoltare a sistemului propus, cum ar fi o analiză a eficienței și complexității metodelor propuse, implementarea formelor clasice într-o bază de date grafică cu posibilitatea recunoașterii formelor.

2. Implementarea sistemului informatic bursier

După terminarea fazei sau etapei de proiectare se trece efectiv la faza de implementare a sistemului într-un lim-

baj de programare adecvat cerințelor utilizatorului. Pentru realizarea sistemului informatic bursier am apelat în această fază la programarea sub Windows sub mediul Visual Basic 4.0. Avantajele utilizării unui mediu de programare orientat spre obiecte și dirijat de evenimente (OOED) sunt:

- se pot implementa cu ușurință cele trei ~~rule~~le construite cu ajutorul metodologiei OMT: al obiectelor, dinamic și funcțional;
- se poate utiliza o interfață grafică utilizator (GUI) cum este mediul Microsoft Windows 95 ce are avantajul de a fi accesibilă utilizatorilor dar și programatorilor;
- relațiile ce există între obiecte pot fi implementate ușor ca proceduri atașate interfeței grafice ce se declanșează la apariția unui eveniment dorit;
- se pot utiliza toate funcțiile și procedurile Windows API (Application Programming Interface) cu avantajul portabilității pe orice platforme Windows dar și cu economisirea unui timp important în programare (ceea ce de fapt constituie și esența programării orientate spre obiecte);
- se pot utiliza resurse deja existente în mediul Windows, cum ar fi, de exem-

plu, programele de calcul tabelar, sistemele de gestiune a bazelor de date, sistemele de creare a rapoartelor, procesoarele de texte, calculatorul de buzunar, calendarul, ceasul etc.

- existența bibliotecilor de legături dinamice (DLL), în care putem găsi o multitudine de funcții, proceduri, structuri de date, controale pot duce, prin utilizarea acestora, la sporirea complexității și fiabilității programului precum și la reducerea timpului de programare;
- existența unor controale de tip OLE (Object Linking and Embedding) dau programatorului posibilitatea de a crea interfețe în care utilizatorul să folosească date din diverse aplicații Windows dar și să modifice, să editeze aceste date cu înregistrarea automată a modificărilor și în programul care gestiona acele date.
- posibilitatea refacerii programului (reutilizabilitatea programelor) cu o mare ușurință la apariția unor noi cerințe ale utilizatorilor.

Sistemul informatic bursier debutează cu un ecran de prezentare, asemănător cu al tuturor produselor sub Microsoft Windows. Această fereastră de prezentare este o instanță a clasei Form, se numește frmBegin și are următoarele atrbute:

```
Begin VB.Form frmBegin
Auto3D      = -1 'True
AutoRedraw   = -1 'True
BorderStyle  = 0 'None
Caption      = "Primul ecran"
ClientHeight = 6270
ClientLeft    = 2460
ClientTop     = 1365
ClientWidth   = 7095
ClipControls = 0 'False
ControlBox    = 0 'False
Height        = 6675
Left          = 2400
LinkTopic     = "Form2"
MaxButton     = 0 'False
MinButton     = 0 'False
ScaleHeight   = 6270
ScaleWidth    = 7095
Top           = 1020
Width         = 7215
```

În interiorul acestei ferestre am definit un obiect de tip SSSPanel (panou), trei

obiecte de tip Image (imagini) și mai multe etichete de text (obiecte de tip Label). Atributele obiectelor sunt legate de poziția acestora în fereastra frmBegin (Height, Left, Top, Width), de fonturi (Property Font), culori (ForeColor, BackColor), imagine (Picture, Stretch), text (Caption). În momentul execuției fereastra de prezentare arată ca în figura de mai jos:



Fig. 1. Fereastra de prezentare a aplicatiei frmBegin

Această fereastră este încărcată de un program principal Basic care are rolul de a face inițializările variabilelor globale (publice), de a afișa fereastra de prezentare timp de două secunde, de a închide această fereastră și de a deschide fereastra de acces în program prin introducerea unei parole. După aceste operații controlul este preluat de fereastra de acces, frmPass. După introducerea parolei corecte controlul programului este preluat de o fereastră principală frmMDI care arată ca în figura 2. Această fereastră este o interfață multidocument MDI (Multiple Document Interface), adică o formă părinte ce este container pentru alte forme numite MDI copil. Ea joacă rolul de fundal al aplicației și încadrează ferestrele copil. Acestea pot fi deschise, minimezate (sub forma unei pictograme), redimensionate, mutate și prelucrate ca și cum ar fi ferestre obișnuite. Totuși ferestrele copil rămân întotdeauna în interiorul ferestrei

părinte, ele neputând fi trase în afara acesteia. Când se minimizează fereastra părinte toate ferestrele copil dispar

în interiorul ei. O aplicație poate avea o singură fereastră MDI părinte dar ferestre MDI copil pot fi mai multe.



Fig. 2. Fereastra principală a aplicației frmMDI

În fereastra MDI a aplicației bursiere sunt plasate mai multe obiecte: un meniu ce poate fi accesat din toate ferestrele copil, o bară de unelte ce conține numele subsistemului accesat precum și butoane de tip SSRibbon pentru o defilare rapidă de la un subsistem la altul. În fereastra principală sunt plasate șase butoane și șase imagini corespunzătoare celor

șase subsisteme ale sistemului informatic bursier. Departamentul *Clienți* (relații cu publicul) este realizat ca o fereastră distinctă, de tip MDI copil, numită *frmClienti*. Ea conține mai multe butoane de comenzi, un ecran de vizualizare și introducere a clienților, precum și un cursor rapid de parcurgere a clientilor agenției bursiere (figura 3).

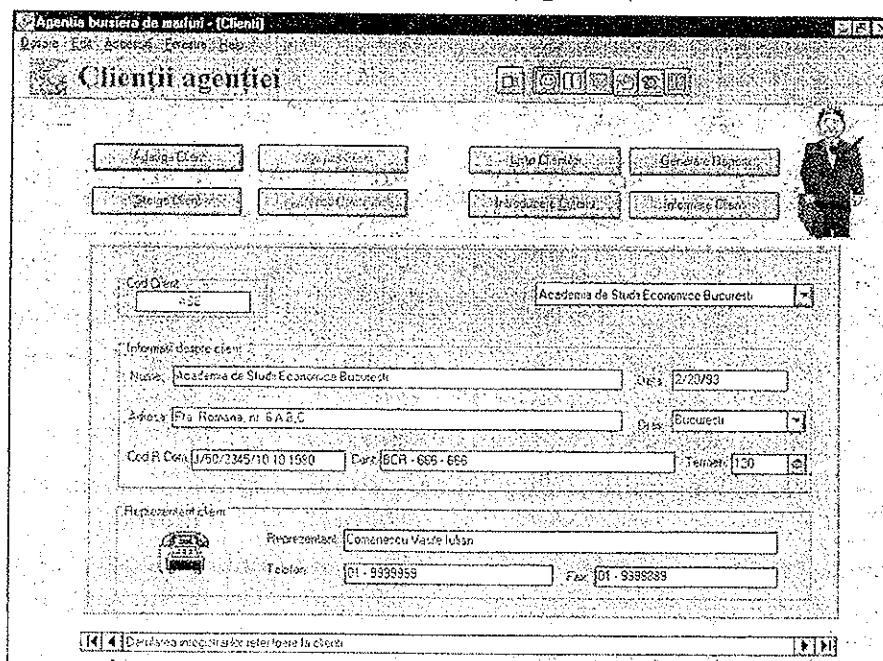


Fig. 3. Fereastra principală a departamentului *Clienți*

Acest departament realizează funcțiile compartimentului de relații cu publicul din cadrul agenției bursiere. Structura care stă la baza funcționării acestui departament este cea descrisă în partea de analiză și proiectare prin cele trei modele (al obiectelor, dinamic și funcțional). Datele referitoare la clienți se păstrează într-o tabelă a bazei de date, numită *Clienți*, care are o structură asemănătoare cu cea descrisă în modelul obiectelor. Pentru a controla datele din tabelă se folosesc în Visual Basic două controale de date (*Data Control*) și obiectele *Database* și *Recordset*. Controlul de date (*Data Control*) este un control predefinit utilizat pentru a conecta o aplicație Visual Basic cu sursa de date selectată. Controalele de date permit parcurgerea simplă a câmpurilor, a înregistrărilor și a tabelelor uneia sau a mai multor baze de date. Ele sunt utilizate și pentru a lega alte controale (casete de text, etichete, casete cu imagine etc.) cu câmpuri din tabelele selectate ale bazei de date. În momentul în care un control, cum ar fi o casetă de text, este legat de un anumit câmp, orice modificare a conținutului casetei de text este înregistrată automat în câmpul asociat. La încărcarea formei sunt setate baza de date a controalelor de date (*DatabaseName*), tabela din care se extrag, se modifică sau se adaugă date (*RecordSource*). În cazul celui de-al doilea control de date, care este folosit pentru afișarea orașelor în ordine alfabetică, se folosește pentru selectarea datelor o interogare SQL (*Standard Query Language*). Această interogare are rolul de a selecta toate orașele existente, o singură dată (*DISTINCT*), ordonate alfabetic (*ORDER BY*). Secvența de cod asociată procedurii *Form_Load* este:

```
Private Sub Form_Load()
    Data1.DatabaseName = BazaDeDate
    Data1.RecordSource = "Clienti"
    Data2.DatabaseName = BazaDeDate
    Data2.RecordSource = "SELECT DISTINCT &
```

```
[oras] " "FROM clienti ORDER BY [oras]"
    btnCautare.Caption = &
        "Introducere &Criteriu"
End Sub
```

Opțiunile pe care le are utilizatorul sunt: adăugarea unui client în baza de date, ștergerea unui client din baza de date, salvarea modificărilor, afișarea tuturor înregistrărilor, selectarea unui client prin căutare prin defilare sau prin selecție din listă, căutarea clienților după diverse criterii, generarea unor rapoarte și contactarea prin modemul de comunicație a clienților. Adăugarea unui client se face prin completarea câmpurilor din panoul *SSPanel* (fig. 3): *cod*, *nume*, *adresă*, *oraș*, *data înregistrării*, *cod Registrul Comerțului*, *cont bancar*, *termen*, *reprezentant*, *telefon* și *fax*. Pentru a putea fi înregistrat trebuie completat obligatoriu câmpul *Nume*, celelalte fiind opționale. După completare apăsarea butonului de salvare adaugă înregistrarea în baza de date. Apăsarea butonului de adăugare eliberează casetele corespunzătoare câmpurilor, pregătindu-le pentru o nouă înregistrare. Dacă utilizatorul dorește să anuleze introducerea clientului în baza de date are la dispoziție butonul de anulare. Pentru ștergerea unui client din baza de date se folosește butonul de ștergere. Înregistrarea care se șterge este cea curentă, care este afișată pe ecran.

După apăsarea butonului de adăugare opțiunile utilizatorului sunt de salvare sau anulare, care inițial erau dezactivate (atributul *Enabled = False*). Opțiunea de ștergere este dezactivată în momentul adăugării deoarece nu mai există înregistrare curentă. După salvare sunt refăcute conținutul listei clienților și cel al listei orașelor (metoda *Refresh*). Același lucru se întâmplă și în momentul în care este șters un client. Procedurile de tratare a evenimentelor de apăsare a butoanelor de adăugare, ștergere, salvare sunt descrise în continuare:

```

Private Sub btnAdauga_Click()
    btnAdauga.Enabled = False
    btnSterge.Enabled = False
    btnSalveaza.Enabled = True
    btnAnuleaza.Enabled = True
    Data1.Recordset.AddNew
    txtCamp(7).Text = "0"
    txtCamp(0).SetFocus
End Sub
Private Sub btnSterge_Click()
    On Error Resume Next
    If MsgBox("Se dorește stergerea clientului: " & txtCamp(1) & "?", vbQuestion + vbYesNo, "Atenție se sterge un client!") = vbYes Then
        Data1.Recordset.Delete
        Data1.Recordset.MovePrevious
        Data1.Refresh: Data2.Refresh
        DBCombo1.Refresh: DBCombo2.Refresh
    End If
End Sub
Private Sub btnSalveaza_Click()
    If txtCamp(1) = "" Then
        MsgBox "Introduceti numele clientului obligatoriu!", vbCritical, "Nu ati introdus datele!"
    Else
        Data1.Recordset.Update
        btnAdauga.Enabled = True
        btnSterge.Enabled = True
        btnSalveaza.Enabled = False
        btnAnuleaza.Enabled = False
        Data1.Refresh: Data2.Refresh
        DBCombo1.Refresh:
        DBCombo2.Refresh
    End If
End Sub

```

Selectarea unei înregistrări poate fi făcută cu ajutorul controlului de date poziționat în partea inferioară a ecranului sau cu ajutorul unei liste a clienților. Cu ajutorul controlului de derulare se poate trece la înregistrarea următoare, precedentă, la începutul înregistrărilor sau la sfârșitul acestora. Lista clienților este obținută prin adăugarea unui control numit *DB ComboBox*. Pentru ca lista să fie ordonată alfabetic modalitatea folosită este de indexare a tabelei *Clienți* după câmpul *Nume*. În momentul selectării unui nume din listă se actualizează toate casetele de text din panoul de afișare, prin deplasarea pointerului de înregistrare la prima înregistrare cu numele identic cu cel selectat.

Pentru robustețea programului am analizat și valorile introduse în casetele de text de către utilizator. Dacă valorile introduse nu sunt în concordanță cu structura bazei de date, având lungimea mai mare sau fiind de alt tip, se

atenționează utilizatorul iar valoarea greșită este stearsă. Analiza datelor este făcută numai în cazul în care nu se dorește introducerea unui criteriu de căutare (butonul de căutare are numele *Introducere Criteriu*) ci se dorește adăugarea sau modificarea unei valori dintr-un câmp al înregistrării curente. La introducerea unui criteriu este anulată validarea deoarece se pot introduce secvențe de căutare (interrogări SQL) care au lungimea mai mare decât lungimea câmpului.

```

Private Sub txtCamp_KeyPress(Index As Integer, KeyAscii As Integer)
    Dim Baza As Database
    Dim Tabla As Table
    Dim Camp As Field
    If btnCautare.Caption = "&Cautare Clienti" Then Exit Sub
    If KeyAscii = vbKeyBack Then Exit Sub
    Set Baza = OpenDatabase(BazaDeDate)
    Set Tabla = Baza.OpenTable("Clienti")
    Set Camp = Tabla(Index)
    If Len(txtCamp(Index)) > Camp.Size Then
        MsgBox "Ati introdus o valoare incorecta!", vbExclamation, "Atenție, data incorecta!"
        txtCamp(Index) = ""
    End If
End Sub

```

Afișarea și căutarea clienților se face utilizând un control de tip rețea (*Grid*). Acestea sunt realizate într-o formă nouă numită *frmGrid*. În cazul apăsării butonului de afișare a listei integrale a clienților interogarea SQL este *SELECT * FROM Clienti*. Dacă se dorește o selectare a clienților atunci se generează prin program interogarea SQL corespunzătoare. Acest lucru se realizează parcurgând casetele de text în care s-a introdus text. Utilizatorul poate căuta astfel toți clienții cu numele care începe cu o anumită literă sau secvență, introducând în caseta de nume litera dorită, sau toți clienții dintr-un anumit oraș introducând sau selectând în caseta de oraș numele orașului. Interogarea generată va fi de forma *SELECT * FROM Clienti WHERE nume LIKE 'A*' AND oras LIKE 'Bucuresti*'.* Procedurile care fac acest lucru sunt:

```

Private Sub btnLista_Click()
    sSQL = "Clienti"
    frmGrid.WindowState = vbNormal
    frmGrid.Left = 435
    frmGrid.Top = 1825
    frmGrid.Height = 5760
    frmGrid.Width = 11070
    frmGrid.Show
End Sub
Private Sub btnCautare_Click()
    Dim dbBursa As Database
    Dim rsBursa As Recordset
    Dim iContor As Integer
    Dim frmGridNou As New frmGrid
    If      btnCautare.Caption      =
    "Introducere &Criteriu" Then
        btnCautare.Caption = "&Cautare Clienti"
        btnAdauga.Enabled = False
        btnSterge.Enabled = False
        btnSalveaza.Enabled = False
        btnAnuleaza.Enabled = False
        DBCombo1.Enabled = True
        DBCombo2.Enabled = True
        Data1.Enabled = True
        Data1.Refresh
        Data2.Refresh
        DBCombo1.Refresh
        DBCombo2.Refresh
    End If
End Sub

```

```

    btnSterge.Enabled = True
    btnSalveaza.Enabled = False
    btnAnuleaza.Enabled = False
    DBCombo1.Enabled = True
    DBCombo2.Enabled = True
    Data1.Enabled = True
    Data1.Refresh
    Data2.Refresh
    DBCombo1.Refresh
    DBCombo2.Refresh
End If
End Sub

```

Pentru a realiza lista integrală sau selectivă a clienților se setează în forma *frmGrid* dimensiunile coloanelor, titlul lor, conținutul lor, se adaugă o coloană cu numărul înregistrărilor și o linie cu numele câmpurilor, ambele fixe (în sensul că la o parcurgere pe orizontală sau verticală acestea rămân ca puncte de reper). De asemenea tot în această formă se tratează și cazurile de excepție, când nu sunt înregistrări ce verifică un criteriu de căutare sau când criteriu introdus este eronat. Este analizată și situația redimensionării formei *frmGrid*, situație în care se redimensionează și rețea *grdBursa*, pentru a fi vizibile sau accesibile toate înregistrările prin parcurgere cu ajutorul butoanelor de scroll vertical sau orizontal asociate controlului rețea. Pentru exemplificare iată cum arată ecranul în situația apăsării butonului de afișare a tuturor clienților (figura 4).

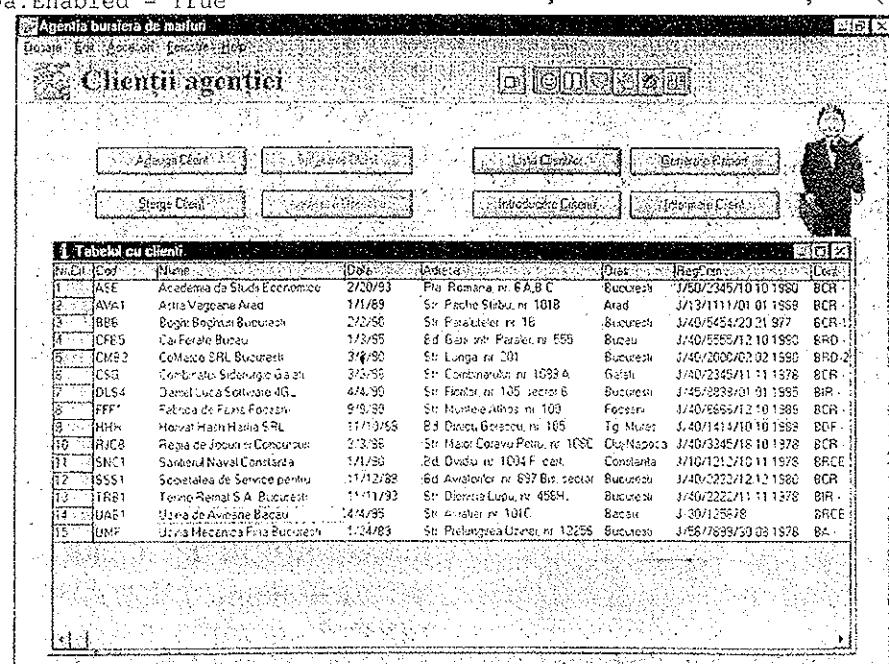


Fig. 4. Afisarea tuturor clientilor agenției

O altă opțiune pe care o are utilizatorul în cadrul departamentul *Clienți* este realizarea legăturii telefonice cu un client din baza de date. Această opțiune este necesară pentru diversele informări ale clienților despre stadiul tranzacțiilor sau pentru obținerea unor informații suplimentare despre clienți. La apăsarea butonului *Informare Clienți* este deschisă o nouă formă care controlează portul de comunicații și transferul datelor prin modem. La lansare se extrage numărul de telefon din baza de date și se transferă acestei forme. Lansarea formei se poate face și prin dublu clic pe imaginea telefonului. Forma *frmTelefon* (figura 5) folosește pentru realizarea legăturii telefonice cu clienții un control special Visual Basic, numit *control de comunicații*. Aceasta

gestionează comunicațiile seriale permitând transmiterea și recepționarea datelor prin intermediul unui port serial. Câteva din proprietățile controlului de comunicații utilizate în program sunt:

- *CommPort* - configerează și returnează numărul portului de comunicații;
- *Settings* - configerează și returnează rata de transfer (bps - biți/secundă), paritatea, biții de date și biții de stop sub forma unui sir;
- *PortOpen* - deschide sau închide un port serial pentru comunicații;
- *Input* - returnează sau elimină caractere din zona tampon de recepție;
- *Output* - scrie un sir de caractere în zona tampon de transmisie;
- *InBufferCount* - numără caracterele recepționate în zona tampon de intrare.

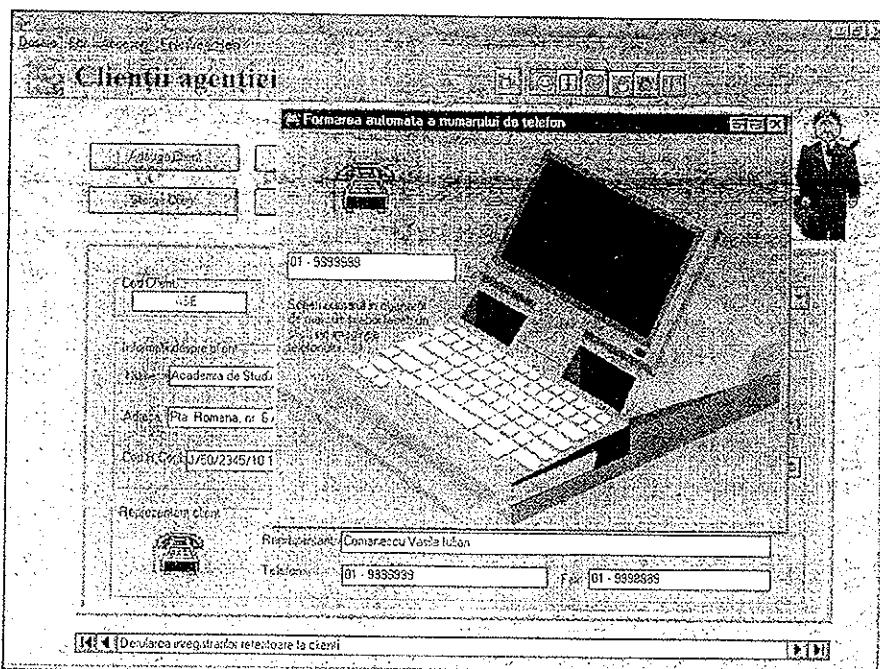


Fig. 5. Realizarea legăturii telefonice cu un client

Procedura *DialOut* primește ca parametru numărul care trebuie format și deschide portul de comunicații Comm 2 la o rată de transfer de 9600 așteptând primirea unui răspuns din partea modemului (tonul telefonului). După primirea tonului formează numărul și apoi închide portul serial.

Sub DialOut(sCallNumber As String)

```

On Error GoTo Tratare
MSComm1.CommPort = 2
MSComm1.Settings = "9600,N,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
MSComm1.Output = "AT" + Chr$(13)
Do
    Dummy = DoEvents()
Loop Until MSComm1.InBufferCount >= 2
MSComm1.InBufferCount = 0
MSComm1.Output = "ATDT" + sCallNumber +
Chr$(13)
MsgBox "Ridicati receptorul si executati Click pe OK.", vbExclamation,
"Formeaza: " & sCallNumber & "..."
```

```

MsgBox "Ridicati receptorul si
executati Click pe OK.", vbExclamation,
"Formeaza: " & sCallNumber & "..."
MSCOM1.PortOpen = False
Exit Sub
Tratare:
MsgBox Error$(Err), vbCritical, "Eroare
de comunicatie prin modem"
Exit Sub
End Sub

```

În program este realizată comunicația cu clienții într-un singur sens, de la agenția bursieră către client. Realizarea legăturii inverse este similară. Proprietățile care se utilizează în acest caz sunt *Output*, *InBufferCount* și *InBufferSize*. Proprietatea *InBufferSize* configuraază dimensiunea totală a zonei tampon de recepție. Cu ajutorul acestor proprietăți pot fi scrise proceduri care să receptioneze informații de la client cu ajutorul modemului de comunicații.

Departamentele *Tranzacții*, *Ordine*, *Brokeri*, *Lichidare tranzacții* sunt realizate în mod similar. Se utilizează controale de date legate la tabelele *Tranzacții*, *Ordine*, *Brokeri* din baza de date *Bursa*. Această bază de date este o bază *Microsoft Access*, dar se pot utiliza și alte tipuri de baze de date. Ea a fost construită în mediul de programare *Visual Basic*, care dispune de un gestionar de date, numit *Data Manager*, ce poate importa sau exporta diverse tipuri de fișiere bază de date (*Microsoft Access*, *dBase III și IV*, *Paradox 3.x și 4.x*, *Btrieve*, *Fox 2.0 și 2.5*, *ODBC*). La realizarea kit-ului de instalare se pot integra și driverele *ODBC* de conversie corespunzătoare. Aceste drivere sunt cele pentru: *Btrieve*, *dBase III și IV*, *Paradox 3.x și 4.x*, *Fox 2.0, 2.5, 2.6*, *Excel 3, 4, 5*, *Text*, *Word Merge* și *SQL*.

Cel de-al șaselea departament al agenției bursiere este cel de analiză și prognoză a prețurilor, care are alte principii de funcționare. Acest departament va fi descris într-unul din numerele următoare.

Bibliografie

- [1] Bellentante B. - *La Bourse, temple de la speculation ou marche financier*, Ed. Hatier, Paris, 1987;
- [2] Bellentante B. - *La Bourse. Son fonctionnement et son role dans la vie économique*, Ed. Hatier, Paris, 1992;
- [3] Date, C. J. - *An Introduction to Database Systems*, Vol. I, Fourth Edition, Addison-Wesley Publishing, 1983;
- [4] Eiteman W.J., Dice Ch. A., Eiteman D.K. - *The Stock Market*, Mc Graw-Hill Book, New York, 1986;
- [5] Heyman Mark-Steven - *Bazele Visual Basic 4*, Editura TEORA, București, 1996;
- [6] Lesson Majorie, Rowe H. Standford, Pointdexter E. Sandra - *Systems Analysis and Design*, 2nd Edition, Science Research Associates Inc, 1995;
- [7] Luca Daniel - *Sistem informatic pentru agenții bursiere de mărfuri*, revista Informatica Economică, vol. I, nr. 1, ian. 1997, pag. 56-61;
- [8] Odăgescu Ioan, Luca Daniel, Furtună Felix, Smeureanu Ion - *Metode și tehnici de programare - Enunțuri, soluții, probleme propuse*, Ed. Intact, București, 1994;
- [9] Popa Ioan - *Bursa, vol. I - Organizare, funcții, experiențe*, Ed. Adevărul, Colecția Bursa, București, 1993;
- [10] Rumbaugh James, Blaha Michael, Premerlani William, Eddy Frederick, Lorenzen William - *Object-Oriented Modelling and Design*, Prentice Hall Int, 1991;
- [11] Sabău Gheorghe, Lungu Ion, Surcel Traian, Bodea Constanța - *Sisteme informaticice pentru conducere*, Editura SIAJ, București, 1994;
- [12] Teasley Mynatt Barbee - *Software Engineering with Student Project Guidance*, Prentice Hall International, 1991;
- [13] Teweles Richard, Bradley Edward - *The Stock Market*, John Willy & Sons, New York, 1987.
- [14] Yourdon Edward - *Modern Structured Analysis*, Yourdon Press, 1992.