

Κων/νου Στυλιάδη – Χρήστου Ιωάννου

# Εφαρμογή Αποθήκης σε Pascal

Φλώρινα, Φεβρουάριος 1995

Παρουσιάζεται μια ολοκληρωμένη εφαρμογή αποθήκης σε Pascal με τη χρήση δυαδικών αρχείων και μενού. Διδάχθηκε στο ΙΕΚ Φλώρινας την περίοδο 1994 – 1995.

### Program App\_Apothikh;

{ Απλή Εφαρμογή Αποθήκης – ΙΕΚ Φλώρινας 20-2-1995 }

Uses

Crt;

Const

MaxRec = 100;      { Μέγιστος αριθμός εγγραφών στο αρχείο }

KHome = 79;      { Κωδικοί πλήκτρων }

KEnd = 71;

KIns = 82;

KDel = 83;

KBack = 08;

KUp = 72;

KDown = 80;

KLeft = 75;

KRight = 77;

KEsc = 27;

KEnter = 13;

KSpace = 32;

Menu : Array[1..6] OF String[21] =

(' Νέα Εγγραφή ',

' Ανεύρεση ',

' Μεταβολή ',

' Διαγραφή ',

' Λίστα ',

' Τέλος ');

Type

ApoRecord = Record      { ορισμός της εγγραφής }

    Kwdikos      : Integer;

    Perigrafh    : String[25];

    PFPA         : Byte;

    PKerdos     : Byte;

    Paragelia   : Integer;      { πεδία της εγγραφής }

    Safe\_Stock   : Integer;

    Ypoloipo    : Integer;

    Xondrikh    : LongInt;

    Lianikh     : LongInt;

End; { Record }

{ το αρχείο Aprofile θα είναι ένας πίνακας με MAXREC εγγραφές }

ApoFile = Array[1..MAXREC] of ApoRecord;

Var

AFile : ApoFile;

Apothikh : File OF ApoRecord;

{ ορισμός διαδικασιών }

**Procedure Edit**(Var St : String; IsNum : Boolean; x,y,Len : Byte;  
Var Up, Down, Finish : Boolean);

Var

```

Ch, Idx : Byte;
Procedure SetDisplay;
Begin
  GotoXY(X, y); Write(' ' : Len);
  If IsNum Then Begin
    GotoXY((X+Len)-Length(st), Y); Write(St);
  End
  Else Begin
    GotoXY(x, Y); Write(St);
  End; { Else }
End; { SetDisplay }

```

Begin

```

Finish := False;
Up := False;
Down := False;
Idx := Length(St);      { IDX = μέγεθος της λέξης }
TextAttr := 112;
SetDisplay;

```

Repeat

```

ch := Ord(ReadKey);
If KeyPressed Then Begin
  { αν πατήθηκε ειδικό πλήκτρο, π.χ. βελάκι }
  Ch := Ord(ReadKey);
  Case Ch of
    KDel, KBack : Begin
      If Ord(St[0])>0 Then
        { μειώνει το μέγεθος του string }
        Dec(St[0]);
      End; { If }
    KUp : Up := True;
    KDown, KEnter : Down := True;
  End; { Case }

```

End

Else

Begin

```

Case Ch of
  KBack : Begin
    If Ord(St[0])>0 Then Dec(St[0]);
  End; { KBack }
  KEnter : Down := True;
  KEsc : Finish := True;

```

```

Else
  If Length(St) >= Len Then Begin
    { αν υπερβούμε το μέγεθος του len }
    Write(#7); { τότε έχουμε ηχητική ενδειξη }
  End
Else Begin
  If IsNum Then Begin      { μόνο αριθμοί }
    If Chr(Ch) In ['0'..'9',','] Then
      St := St + Chr(Ch);
    End
  Else Begin { όλα τα γράμματα και τα σύμβολα }
    St := St + Chr(Ch);
  End; { If-2 }
End; { If-1 }
End; { Case }
End; { If-0 }
SetDisplay;
Until Finish or (Up=True) or (Down=True) or (Ch=KEnter);
TextAttr := 07; SetDisplay;
End;

```

```

Function Center(S : String; W : Byte) : Byte;
Begin
  Center := (W - Length(S)) Div 2;
End;

```

*{ η διαδικασία αυτή γράφει άμεσα έναν χαρακτήρα στη μνήμη οθόνης στη θέση X, Y }*

```

Procedure SetScreenCh(x, y : Byte; Ch : Char);
Var
  Position : Word;
Begin
  Position := ((y-1)*80 + x - 1) * 2;
  Mem[$B800 : Position] := Ord(Ch);
End;

```

*{ η διαδικασία αυτή τοποθετεί άμεσα έναν χαρακτήρα ιδιοτήτων (χρώμα) στη μνήμη οθόνης στη θέση X, Y }*

```

Procedure SetScreenColor(x, y, Col : Byte);
Var
  Position : Word;
Begin
  Position := ((y-1) * 80 + x - 1) * 2;
  Mem[$B800 : Position+1] := Col;
End;

```

{ η διαδικασία αυτή υπολογίζει το foreground μαζί με το background και τα κάνει ένα byte }

```
Function CAttr(F, B : Byte) : Byte;
Begin
  CAttr := (F Shl 4) or B
End;
```

```
Procedure Frame(X, Y, X1, Y1, clr : Byte);
Var
  Hor, Ver : Byte;
Begin
  For Hor := X TO X1 Do Begin
    SetScreenCh(Hor, Y, '-');
    SetScreenColor(Hor, y, Clr);
    SetScreenCh(Hor, Y1, '-');
    SetScreenColor(Hor, y1, Clr);
  End; { For }
  For Ver := Y TO Y1 Do Begin
    SetScreenCh(x, Ver, '|');
    SetScreenColor(x, Ver,Clr);
    SetScreenCh(x1, Ver, '|');
    SetScreenColor(x1, Ver,Clr);
  End; { For }
  SetScreenCh(x, y, '+');
  SetScreenCh(x, y1, '+');
  SetScreenCh(x1, y, '+');
  SetScreenCh(x1, y1, '+');
End;
```

```
Function ComputePoso(Poso : LongInt; Percent : LongInt) : LongInt;
Begin
  ComputePoso := Poso * Percent Div 100;
End;
```

```
Procedure Message(Msg : String);
Begin
  TextAttr := 168;           { αντίστροφο χρώμα }
  GotoXY(1, 25); Write(Msg);
  { τυπώνει το μήνυμα MSG στην 25η γραμμή }
  ClrEol;
  TextAttr := 07;           { κανονικό χρώμα }
  Repeat Until KeyPressed;
  GotoXY(1,25); ClrEol;     { καθαρίζει την 25η γραμμή }
End;
```

**Procedure PressKey;**

```

Begin
    Message('Πάτα ένα πλήκτρο...');
End;
```

**Function Choice(Msg : String) : Boolean;**

```

Var
    Ch : Char;
Begin
    GotoXY(1,25); Write(Msg);
    Repeat
        Ch := Uppcase(ReadKey);
    Until Ch IN ['N', 'n', 'Y', 'y', 'O', 'O', 'o'];
    Choice := Ch IN ['N', 'n', 'Y', 'y'];
End;
```

*{ η διαδικασία αυτή βρίσκει εγγραφές }*

**Function FindRec(Perigr : String; Var Idx : LongInt; Var Arec : ApoRecord)**

```

:
    Boolean;
Var
    Find : Boolean;
Begin
    Idx := 0;
    Find := False;
    Repeat
        Inc(Idx);
        With AFile[Idx] Do Begin
            { ψάχνουμε βάσει της περιγραφής ή μέρους αυτής }
            If Perigr = Copy(Perigraph, 1, Length(Perigr)) Then Begin
                Find := True;
                { αν βρεθεί, τότε η find γίνεται true }
                ARec := AFile[Idx];
                { διαβάζουμε την εγγραφή από το αρχείο }
            End; { If }
        End; { With }
    Until (Find=True) Or (Idx >= MaxRec);
    FindRec := Find;
End;
```

*{ η διαδικασία αυτή καταχωρεί εγγραφές στο αρχείο }*

**Procedure AddRec(Idx : LongInt; ARec : ApoRecord);**

```

Begin
    AFile[Idx] := ARec;
    { καταχωρούμε την εγγραφή στη θέση IDX }
End;
```

{ η διαδικασία αυτή τυπώνει τις επιγραφές των πεδίων }

**Procedure Forma**(Header : String);

Begin

```

    TextAttr := 112;
    GotoXY(2,2);ClrEol;
    GotoXY(Center(Header, 80), 2); Write(Header);
    TextAttr := 07;
    Frame(1, 1, 80, 03, 7);
    Frame(1, 1, 80, 09, 7);
    Frame(1, 1, 80, 22, 7);
    TextAttr := 15;
    GotoXY(02,4); Write('Κωδικός :');
    GotoXY(02,5); Write('Περιγραφή :');
    GotoXY(02,6); Write('Φ.Π.Α : %');
    GotoXY(02,7); Write('Κέρδος : %');
    GotoXY(42,4); Write('Υπόλοιπο Αποθήκης :');
    GotoXY(42,5); Write('Στοκ Ασφαλείας :');
    GotoXY(42,6); Write('Προς Παραγγελία :');
    GotoXY(42,7); Write('Τιμή Αγοράς :');
    GotoXY(42,8); Write('Τιμή Λιανικής :');
    TextAttr := 07;

```

End;

{ η διαδικασία αυτή δημιουργεί μια νέα εγγραφή }

**Procedure Eisagwgh**(Var Idx : LongInt; Var ApoRec : ApoRecord);

Type

```

    FieldRec = Record
        x, y, Len : Byte;
        Num : Boolean;
    End; { Record }

```

Const

```

    XY : Array[1..7] of FieldRec =
    ((X:13; Y:4; Len:5; Num :True),(X:13;Y:5; Len:20; Num:False),
    (X:13; Y:6; Len:4; Num :True),(X:13;Y:7; Len:4; Num :True),
    (X:61; Y:4; Len:6; Num :True),(X:61;Y:5; Len:6; Num :True),
    (X:61; Y:7; Len:8; Num :True));

```

Var

```

    St : String;
    Err : Integer;
    Kerdos, Ajia, FPA, Paragel : LongInt;
    IsUp, IsDown, Finish : Boolean;
    Count : Byte;

```

```

Begin
  With ApoRec Do Begin Count := 1;
  Repeat
    St := ”;
    Case Count of
      1 : If Kwdikos > 0 Then Str(Kwdikos : 5, St);
      2 : St := Perigraph;
      3 : If PFPA>0 Then Str(PFPA : 2, St);
      4 : IF PKerdos > 0 Then Str(Pkerdos : 2, St);
      5 : IF Ypoloipo > 0 Then Str(Ypoloipo : 6, St);
      6 : IF Safe_Stock > 0 Then Str(Safe_Stock : 6, st);
      7 : IF Xondrikh > 0 Then Str(Xondrikh : 8, St);
    End; { Case }
    With XY[Count] Do
      Edit(St, Num, X, Y, Len, IsUp, IsDown, Finish);
      Case Count of
        1 : Val(St, Kwdikos, Err);
        2 : Perigraph := St;
        3 : Val(St, PFPA, Err);
        4 : Val(St, PKerdos, Err);
        5 : Val(St, Ypoloipo, Err);
        6 : Val(St, Safe_Stock, Err);
        7 : Val(St, Xondrikh, Err);
      End; { Case }
      If IsUp Then Dec(Count);
      If IsDown Then Inc(Count);
      If Count < 1 Then Count := 7;
      If Count > 7 Then Count := 1;
      Idx := Kwdikos; Paragel := Ypoloipo - Safe_Stock;
      If Paragel < 0 Then TextAttr := 130;
      GotoXY(61,6); Write(Paragel : 7);
      TextAttr := 07;
      Kerdos := ComputePoso(Xondrikh, PKerdos);
      Ajia := Xondrikh + Kerdos;
      FPA := ComputePoso(Ajia, PFPA);
      { συνάρτηση υπολογισμού με ποσοστό }
      Lianikh := Ajia + Fpa;
      TextAttr := 12;
      GotoXY(61, 8); Write(Lianikh : 10); TextAttr := 07;
    Until Finish;
  End; { With }
End;

```



**Procedure NewRecord;**

```

Var
  ARec: ApoRecord;
  Idx : LongInt;
Begin
  ClrScr;
  { τυπώνουμε τη φόρμα εισαγωγής }
  Forma('Νέα Εγγραφή');
  FillChar(ARec, SizeOF(ARec), 0);
  { εισάγουμε τα δεδομένα της εγγραφής }
  Eisagwgh(Idk, ARec);
  If Choice('Να καταχωρηθεί η εγγραφή (N/O)') Then
    AddRec(Idk, ARec);
    { και τα καταχωρούμε σε μια θέση στο αρχείο }
End;

```

{ απεικόνιση εγγραφής }

**Procedure DisplayFields(ARec : ApoRecord);**

{ τυπώνει το περιεχόμενο της εγγραφής AREC }

```

Begin
  With ARec Do Begin
    GotoXY(13, 4); Write(Kwdikos);
    GotoXY(13, 5); Write(Perigraph);
    GotoXY(13, 6); Write(PFPA : 2);
    GotoXY(13, 7); Write(PKerdos : 2);
    GotoXY(61, 4); Write(Ypoloipo);
    GotoXY(61, 5); Write(Safe_Stock);
    GotoXY(61, 6); Write(Paragelia);
    GotoXY(61, 7); Write(Xondrikh : 8);
    GotoXY(61, 8); Write(Lianikh : 10);
  End; { With }
End;

```

**Procedure DisplayRec;**

```

Var
  ARec : ApoRecord;
  Perigr : String;
  Idx : LongInt;
Begin
  ClrScr;
  GotoXY(1, 1); Write('Δώσε την περιγραφή...');
  ReadLN(Perigr);
  { αν δεν βρεθεί η εγγραφή }
  If Not FindRec(Perigr, Idx, ARec) Then
    Message('Δεν βρέθηκε η εγγραφή...')
  Else Begin
    Forma('Προβολή εγγραφής');
    DisplayFields(ARec);
    PressKey;
  End; { Else }
End;

```

{ μεταβολή εγγραφής }

**Procedure Metabolh;**

```

Var
  ARec : ApoRecord;
  Perigr : String;
  Idx : LongInt;
Begin
  ClrScr;
  GotoXY(1, 1); Write('Δώσε την περιγραφή...');
  ReadLN(Perigr);
  { αν δεν βρεθεί η εγγραφή }
  If Not FindRec(Perigr, Idx, ARec) Then
    Message('Δεν βρέθηκε η εγγραφή...')
  Else Begin
    Forma('Μεταβολή εγγραφής');
    DisplayFields(ARec);
    Eisagwgh(Idx, ARec);
    If Choice('Θέλεις να καταχωρήσεις τις αλλαγές (N/O)') Then
      Begin
        AddRec(idx, ARec);
      End; { If-2 }
    PressKey;
  End; { Else }
End;

```

{ διαγραφή εγγραφών }

**Procedure Diagraph;**

Var

  ARec : ApoRecord;

  Perigr : String;

  Idx : LongInt;

  A : Word;

Begin

  ClrScr;

  GotoXY(1, 1); Write('Δώσε την περιγραφή...');

  ReadLN(Perigr);

  If Not FindRec(Perigr,Idx,AREC) Then

    Message('Δεν βρέθηκε η εγγραφή...')

  Else Begin

    Forma('Διαγραφή εγγραφής');

    DisplayFields(ARec);

    If Choice('Θέλεις να διαγράψεις αυτή την εγγραφή (N/O)')

      Then Begin

        FillChar(AFile[Idx], SizeOF(AFile[Idx]), 0);

      End; { If-2 }

    PressKey;

    End; { Else }

End;

{ λίστα εγγραφών }

**Procedure List;**

Var

  Idx : LongInt;

  Grammh : Byte;

  Procedure Header;           { επικεφαλίδα της λίστας }

  Begin

    TextAttr := 112;

    GotoXY(1, 1);

    Write('Κωδ | Περιγραφή | Υπόλοιπο | Χονδρική | Αξία Υπολ');

    ClrEOL;

  TextAttr := 07;

  End; { Header }

Begin

  ClrScr;

  Header;

  Idx := 1;

  Grammh := 1;

  For Idx := 1 to MaxRec Do Begin

    With AFile[Idx] Do Begin

      If Perigraph <>'' Then Begin

{ αν η περιγραφή δεν είναι κενή, εδώ τυπώνονται τα πεδία κάθε εγγραφής }

```

        GotoXY(1, 1+grammh);
        Write(Kwdikos : 3, '|', Perigraph, '': 25-
Length(Perigraph), '|', Υπολοιπο : 10, '|', Xondrikh : 10, '|',
Υπολοιπο * Xondrikh : 10);
        Inc(Grammh);      { μετρητής γραμμών }
End; { If }
End; { With }
If Grammh >= 23 Then Begin
    { αν περάσουμε την 23η γραμμή }
    Grammh := 1;
    { μηδενίζουμε ξανά τον μετρητή }
    PressKey;
End; { If }
End; { For }
PressKey;
End;

{ μενού εφαρμογής }
Procedure DisplayMenu;
Var
    X : Byte;
Begin
    Frame(24, 3, 41, 05, 7);
    Frame(24, 3, 41, 12, 7);
    TextAttr := 113;
    GotoXY(25, 4); Write(' ____ Επιλογές ____ ');
    TextAttr := 07;
    For x:= 1 to 6 do Begin
        GotoXY(25, 5+x); Write(Menu[x]);
    End; { For }
    GotoXY(20, 25); Write(' Επιλογή με βελάκια και [Insert] ');
End;

```

**Procedure DoMenu;**

Var

Select : Byte;

Ch : Byte;

Uesh : Byte;

Begin

DisplayMenu;

Uesh := 1;

TextAttr := 112;

GotoXY(25, 5+Uesh); Write(Menu[Uesh]);

Repeat

Select := 0;

Ch := Ord(ReadKey);

TextAttr := 07;

GotoXY(25, 5+Uesh); Write(Menu[Uesh]);

If KeyPressed Then Begin

Ch := Ord(ReadKey);

Case Ch of

KUp : Dec(Uesh);

KDown : Inc(Uesh);

KIns : Select := Uesh;

End; { Case }

End; { If }

If Ch=KEsc Then Select := 6;

If Uesh &gt; 6 Then Uesh := 1;

If Uesh &lt; 1 Then Uesh := 6;

TextAttr := 112;

GotoXY(25,5+Uesh); Write(Menu[Uesh]);

TextAttr := 07;

Case Select of

1 : NewRecord;

2 : DisplayRec;

3 : Metabolh;

4 : Diagraph;

5 : List;

6 : Begin

GotoXY(1, 25); ClrEol;

If NOT Choice('Θέλεις να βγεις απο το  
πρόγραμμα (N/O)')

Then Select := 0;

GotoXY(1, 25); ClrEol;

End; { 6 }

End; { Case }

If Select &gt; 0 Then Begin

ClrScr;

```

        DisplayMenu;
        TextAttr := 112;
        GotoXY(25, 5+Uesh); Write(Menu[Uesh]);
    End;{if}
Until (Select = 6);
End;

```

*{ διαδικασίες αρχείων }*

**Function Exist**(FName : String): Boolean;

Var

F : File;

Begin

{ \$i- }

Assign(F, FName);

Reset(F);

If IOresult <> 0 Then

*{ αν το αρχείο δεν υπάρχει, η IORESULT <> 0 }*

Exist := False

Else

Exist := True;

*{ αν η IORESULT=0, το αρχείο υπάρχει }*

{ \$i+ }

End;

**Procedure ClearRec**(Var ARec : ApoRecord);

Begin

FillChar(ARec, SizeOF(ARec), 0);

*{ μηδενίζει την εγγραφή }*

End;

**Procedure Openfile;**

Var

ARec : ApoRecord;

Idx : LongInt;

Begin

Assign(Apothikh, 'C:\TP\Apothik.dat');

ClearRec(ARec);

If Not Exist('C:\TP\Apothik.dat') Then Begin

Rewrite(Apothikh);

For Idx := 1 to MaxRec Do Begin

Seek(Apothikh, Idx);

Write(Apothikh, ARec);

End; { For }

End Else

Reset(Apothikh);

End;

```
{ καταχωρεί τις εγγραφές στον δίσκο }  
Procedure WriteApothikh;  
Var  
    Idx : LongInt;  
Begin  
    For Idx := 1 to MaxRec Do Begin  
        Seek(Apothikh, Idx);  
        Write(Apothikh, AFile[Idx]);  
    End; { For }  
End;
```

```
{ διαβάζει τις εγγραφές από τον δίσκο }  
Procedure ReadApothikh;  
Var  
    Idx : LongInt;  
Begin  
    For Idx := 1 to MaxRec Do Begin  
        Seek(Apothikh, Idx);  
        Read(Apothikh, AFile[Idx]);  
    End; { For }  
End;
```

```
Begin    { κυρίως πρόγραμμα }  
    FillChar(AFile, SizeOF(AFile), 0);  
    OpenFile;  
    ReadApothikh;  
    TextAttr := 07;  
    ClrScr;  
    DoMenu;  
    WriteApothikh;  
    Close(Apothikh);  
End.
```