

ΚΕΦΑΛΑΙΟ 11ο

ΣΥΓΧΡΟΝΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

11.1. Προσδοκώμενα αποτελέσματα



Μέχρι τώρα είχες εργαστεί σε ένα παραδοσιακό περιβάλλον προγραμματισμού, που κυρίως ενδιαφέρεται για το πρόγραμμα και όχι για το περιβάλλον εργασίας του χρήστη. Σε αυτό το κεφάλαιο έρχεσαι σε επαφή με σύγχρονα γραφικά περιβάλλοντα προγραμματισμού που παρέχουν ιδιαίτερες δυνατότητες επικοινωνίας του προγράμματος με το χρήστη αλλά και ειδικά εργαλεία προς το προγραμματιστή για την απλούστευση του προγραμματισμού.

Οι βασικές έννοιες προγραμματισμού που ήδη γνωρίζεις και χρησιμοποιείς, εξακολουθούν να εφαρμόζονται και σε ένα σύγχρονο προγραμματιστικό περιβάλλον, σε συνδυασμό όμως με τις τεχνικές του αντικειμενοστραφή και του οδηγούμενου από τα γεγονότα προγραμματισμού. Ο συνδυασμός αυτός έχει σαν στόχο την εκμετάλλευση όλων των δυνατοτήτων ενός σύγχρονου περιβάλλοντος με την προσομοίωση του χώρου εργασίας με τον πραγματικό φυσικό μας κόσμο, μέσα από την παραδοχή ότι κάθε γραφικό εργαλείο της διασύνδεσης του προγράμματος με το χρήστη αποτελεί ένα ανεξάρτητο αντικείμενο. Μέσα από τα παραδείγματα του κεφαλαίου, αναλύεται ο νέος τρόπος αντίληψης των εφαρμογών και επιδεικνύεται ο συνδυασμός των τεχνικών προγραμματισμού.

Σαν προγραμματιστικό περιβάλλον, για την παρουσίαση των λυμένων ασκήσεων, έχουμε επιλέξει τη Visual Basic for Windows, γιατί είναι απλό, εύχρηστο και δημοφιλές και κυρίως γιατί οι εντολές κώδικα είναι σχεδόν πανομοιότυπες με τη γλώσσα προγραμματισμού Basic που ήδη έχουμε χρησιμοποιήσει σε προηγούμενα παραδείγματα του τετραδίου. Εναλλακτικά, σε κάθε παράδειγμα συμπεριλαμβάνουμε και τον κώδι-

κα σε περιβάλλον Delphi. Στην υλοποίηση της διεπαφής χρήστη με το Delphi δεν αναφερόμαστε καθόλου, γιατί η φιλοσοφία και οι τεχνικές δημιουργίας του δεν διαφοροποιούνται από τις αντίστοιχες της Visual Basic.

11.2. Επιπλέον παραδείγματα



Παράδειγμα 1

Στο βιβλίο σου έχουμε ήδη παρουσιάσει το πρώτο παράδειγμα εφαρμογής σε υποθετικό περιβάλλον προγραμματισμού. Το παράδειγμα αναφέρεται σε μια παρουσίαση των τριών βημάτων ανάπτυξης μιας εφαρμογής σε σύγχρονο προγραμματιστικό περιβάλλον. Στο σημείο αυτό θεωρούμε ότι πρέπει να επιδείξουμε την υλοποίησή του σε πραγματικό περιβάλλον προγραμματισμού.

Περιβάλλον προγραμματισμού Visual Basic

Το πρώτο στοιχείο που πρέπει να σε απασχολήσει, είναι ο τρόπος επικοινωνίας του χρήστη με την εφαρμογή. Για το σχεδιασμό λοιπόν της διεπαφής χρήστη χρησιμοποιήσαμε τα παρακάτω γραφικά αντικείμενα:

- ⇒ μια **φόρμα** (form) η οποία θα αποτελέσει το παράθυρο μέσα στο οποίο θα εκτελείται η εφαρμογή,
- ⇒ τέσσερα **πλήκτρα εντολής** (command buttons) με τα οποία ο χρήστης θα καθοδηγεί και θα τερματίζει την εκτέλεσή της,
- ⇒ τέσσερις **ετικέτες** (labels) που θα εμφανίζουν τις πληροφορίες,
- ⇒ μια **γραμμή** (line) για να δώσουμε έμφαση στην ετικέτα με τον τίτλο της εφαρμογής.

Μόλις ξεκινήσουμε τη δημιουργία μιας εφαρμογής, η Visual Basic έχει ήδη δημιουργήσει μια φόρμα με το όνομα Form1. Επάνω σε αυτή τη φόρμα τοποθετούμε τα πλήκτρα εντολής, τις ετικέτες και το αντικείμενο γραμμή. Τα γραφικά αντικείμενα τα επιλέγουμε από την εργαλειοθήκη (toolbox) που μας παρέχει το περιβάλλον προγραμματισμού της Visual Basic.

Στις ιδιότητες των αντικειμένων της εφαρμογής αποδίδουμε τις επόμενες τιμές:

Αντικείμενο	Ιδιότητα	Τιμή
Φόρμα	Name	FrmMain
	Caption	Η πρώτη μας εφαρμογή σε σύγχρονο προγραμματιστικό περιβάλλον
Πλήκτρο εντολής 1	Name	CmdStep1
	Caption	1
Πλήκτρο εντολής 2	Name	CmdStep2
	Caption	2
Πλήκτρο εντολής 3	Name	CmdStep3
	Caption	3
Πλήκτρο εντολής 4	Name	CmdEnd
	Caption	(κενό)
	Style	1-Graphical
	Picture	\\vb\graphics\icons\traffic\Trffc14.ico
Ετικέτα 0	Name	LblTitle
	Visible	True
	Caption	Βήματα ανάπτυξης μιας εφαρμογής
	ForeColor	&H000000C0&
Ετικέτα 1	Name	LblStep1
	Visible	False
	Caption	Σχεδιασμός του τρόπου επικοινωνίας χρήστη-εφαρμογής, επιλέγοντας τα κατάλληλα αντικείμενα
	ForeColor	&H8000000D& (Highlight)
Ετικέτα 2	Name	LblStep2
	Visible	False
	Caption	Καθορισμός της αρχικής συμπεριφοράς των αντικειμένων μέσω των ιδιοτήτων που τα χαρακτηρίζουν
	ForeColor	&H8000000D& (Highlight)
Ετικέτα 3	Name	LblStep3
	Visible	False
	Caption	Δημιουργία και εκσφαλμάτωση του κώδικα
	ForeColor	&H8000000D& (Highlight)
Γραμμή	Name	Lntitle
	ForeColor	&H000000C0&

Το τελευταίο βήμα που πρέπει να εκτελέσουμε για να ολοκληρωθεί ο σχεδιασμός της εφαρμογής, είναι η προσθήκη των εντολών κώδικα.

Στη Visual Basic και στο Delphi, για να μετατρέψουμε την τιμή μιας ιδιότητας ενός αντικειμένου χρησιμοποιούμε εντολές της μορφής :

Αντικείμενο.Ιδιότητα = Τιμή

Για παράδειγμα αν θέλουμε να εμφανίσουμε μέσα σε ένα αντικείμενο πλαίσιο κειμένου, με όνομα Text1, το κείμενο “Καλώς ήλθατε στο κόσμο των αντικειμένων” πρέπει να χρησιμοποιήσουμε μια εντολή της μορφής :

Text1.Text= “Καλώς ήλθατε στο κόσμο των αντικειμένων”

Για την εκτέλεση μιας μεθόδου χρησιμοποιούμε εντολές της μορφής :

Αντικείμενο.Μέθοδος

Ο καθαρισμός των περιεχομένων ενός αντικειμένου λίστας με όνομα List1, πραγματοποιείται με τη μέθοδο Clear:

List1.Clear

Στην εντολή εκτέλεσης των περισσότερων μεθόδων είναι δυνατό να συμπεριλάβουμε και ορίσματα (arguments) με τα οποία επιτυγχάνουμε παραμετροποίηση του αποτελέσματος. Σε αυτή τη περίπτωση η εντολή εκτέλεσης μιας μεθόδου συντάσσεται ως εξής :

Αντικείμενο.Μέθοδος Λίστα ορισμάτων

Όταν θέλουμε να μετακινήσουμε επάνω σε μία φόρμα ένα αντικείμενο εικόνα με όνομα Image1 θα χρησιμοποιήσουμε την μέθοδο Move ως εξής:

Image1.Move 100, 200

Το αποτέλεσμα της εντολής, θα είναι η μετακίνηση του αντικειμένου Image1 στο σημείο της φόρμας με απόλυτες συντεταγμένες (100, 200).

Μέσα από το παράθυρο κώδικα που παρέχει η Visual Basic πληκτρολογούμε τις παρακάτω εντολές στις διαδικασίες γεγονότων:

```
Private Sub CmdStep1_Click()  
    ' ΔΕΡΕΘΗΙ ΑΙΘΥΙΕΟΦΘ ΘΗΡΘΙΘ ΑΡΙΑΘΙΘ  
    ' ΙΑ ΘΦΙ ΕΑΕΥΘΦΘΑ Visible ΑΙΘΑΙΒΑΙΘΙΑ Ρ ΑΘΙΕΗΥΘΘΙΘΙΑ ΑΘΥ ΘΦΙ ΙΕΥΙΦ  
    ' ΥΙΑ ΑΗΑΘΕΕΥ ΑΙΘΕΕΑΒΙΑΙΙ  
    ' ΙΕ ΑΘΙΑΘΥΘ ΘΕΙΥΘ ΘΦΘ ΕΑΕΥΘΦΘΑΘ Visible ΑΒΙΑΕ ΑΙΘΒΘΘΙΕ+Α True Ρ  
    ' False.  
    LblStep1.Visible = True  
    LblStep2.Visible = False  
    LblStep3.Visible = False  
End Sub  
Private Sub CmdStep2_Click()  
    ' ΔΕΡΕΘΗΙ ΑΙΘΥΙΕΟΦΘ ΑΑΥΘΑΗΙΘ ΑΡΙΑΘΙΘ
```

```

        LblStep1.Visible = False
        LblStep2.Visible = True
        LblStep3.Visible = False
    End Sub
Private Sub CmdStep3_Click()
    ' Δεξέομαι αιτιόσους ορισμούς απλά
    LblStep1.Visible = False
    LblStep2.Visible = False
    LblStep3.Visible = True
End Sub
Private Sub CmdEnd_Click()
    ' Δεξέομαι ορισμούς
    End
End Sub

```

Περιβάλλον προγραμματισμού Delphi

```

program Prj1;
uses
    Forms,
    Unit1 in 'Unit1.pas' {LblStep1};
{$R *.RES}
begin
    Application.Initialize;
    Application.CreateForm(TLblStep1, LblStep1);
    Application.Run;
end.
unit Unit1;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs,
    StdCtrls, Buttons;
type
    TLblStep1 = class(TForm)
        Title1: TLabel;
        CmdStep1: TButton;
        CmdStep2: TButton;
        CmdStep3: TButton;
        LblStep1: TLabel;
        LblStep2: TLabel;
        LblStep3: TLabel;
        CmdEnd: TBitBtn;
        procedure CmdStep1Click(Sender: TObject);
        procedure CmdStep2Click(Sender: TObject);
        procedure CmdStep3Click(Sender: TObject);
        procedure CmdEndClick(Sender: TObject);
    private
        { Private declarations }
    public

```

```

        { Public declarations }
    end;
var
    LblStep1: TLblStep1;
implementation
{$R *.DFM}
procedure TLblStep1.CmdStep1Click(Sender: TObject);
begin
    LblStep1.Visible := True;
    LblStep2.Visible := False;
    LblStep3.Visible := False;
end;
procedure TLblStep1.CmdStep2Click(Sender: TObject);
begin
    LblStep1.Visible := False;
    LblStep2.Visible := True;
    LblStep3.Visible := False;
end;
procedure TLblStep1.CmdStep3Click(Sender: TObject);
begin
    LblStep1.Visible := False;
    LblStep2.Visible := False;
    LblStep3.Visible := True;
end;
procedure TLblStep1.CmdEndClick(Sender: TObject);
begin
    close;
end;
end.

```

Παράδειγμα 2

Να κατασκευαστεί μια οθόνη εισαγωγής κωδικού πρόσβασης (password). Κατά την εισαγωγή του κωδικού στην οθόνη του υπολογιστή, για λόγους ασφάλειας, δεν πρέπει να εμφανίζονται οι χαρακτήρες που πληκτρολογεί ο χρήστης αλλά ο χαρακτήρας *. Όταν ο χρήστης πληκτρολογεί τον κωδικό να εμφανίζεται αντίστοιχα μήνυμα αποδοχής ή απόρριψης.

Περιβάλλον προγραμματισμού Visual Basic

Για την υλοποίηση της εφαρμογής προτείνουμε να χρησιμοποιηθούν τα παρακάτω αντικείμενα:

- ⇒ μια **φόρμα** η οποία είναι το παράθυρο μέσα στο οποίο θα εκτελείται η εφαρμογή,
- ⇒ ένα **πλαίσιο κειμένου** (text box) που θα εισάγετε ο κωδικός,
- ⇒ μια **ετικέτα** (label), για την προτροπή του χρήστη ώστε να πληκτρολογήσει τον κωδικό,

⇒ δύο **πλήκτρα εντολής** που θα χρησιμεύουν για την αποδοχή του κωδικού και τον τερματισμό της εφαρμογής.

Στις ιδιότητες των αντικειμένων της εφαρμογής αποδίδουμε τις παρακάτω τιμές:

Αντικείμενο	Ιδιότητα	Τιμή
Φόρμα	Name	Frmpwd
	Caption	Παράδειγμα πλαισίων διαλόγου
Ετικέτα 1	Name	Lblpwd
	Caption	Πληκτρολογήστε τον κωδικό :
Πλαίσιο κειμένου	Name	Txtpwd
	Passwordchar	*
	Text	(κενό)
Πλήκτρο εντολής 1	Name	CmdOK
	Caption	OK
Πλήκτρο εντολής 2	Name	CmdEnd
	Caption	Τέλος

Μετά την τοποθέτηση των αντικειμένων και την αντιστοίχιση των ιδιοτήτων έχουμε ολοκληρώσει την οθόνη εισαγωγής κωδικού (Σχήμα 11.1).



Σχ. 11.1. Η οθόνη της εφαρμογής εισαγωγής κωδικού.

Στο παράδειγμά μας θα χρησιμοποιήσουμε ως κωδικό την τρέχουσα ημερομηνία. Η ημερομηνία θα διαβάζεται από τον υπολογιστή με την συνάρτηση Date.

Τα μηνύματα προς το χρήστη θα γίνονται με τη χρήση προκαθορισμένων πλαισίων διαλόγου που προσφέρει η Visual Basic. Η εμφάνιση των προκαθορισμένων πλαισίων διαλόγου γίνεται με την συνάρτηση MsgBox, η οποία συντάσσεται ως εξής:

MsgBox(μήνυμα[, πλήκτρα εντολής] [, τίτλος πλαισίου] [, αρχείο βοήθειας, δείκτης αρχείου])

Τα περισσότερα ορίσματα της συνάρτησης MsgBox, όπως γίνεται αντιληπτό, είναι μη υποχρεωτικά και μπορούμε να τα παραλείψουμε. Όταν παραλείψουμε κάποιο όρισμα η Visual Basic χρησιμοποιεί τα δικά της εξ ορισμού ορίσματα, για παράδειγμα στον τίτλο πλαισίου τοποθετεί το όνομα της εφαρμογής.

Σε ένα προκαθορισμένο πλαίσιο διαλόγου έχουμε τη δυνατότητα να εμφανίσουμε πλήκτρα εντολής. Μόλις ο χρήστης πατήσει κάποιο από τα πλήκτρα εντολής του πλαισίου διαλόγου, αυτό φεύγει από την οθόνη και επιστρέφει στο πρόγραμμα μια τιμή που εκφράζει το πλήκτρο που επιλέχτηκε. Το όρισμα **πλήκτρα εντολής**, βασικά καθορίζει τον αριθμό και τον τύπο των πλήκτρων καθώς και το εικονίδιο που θα εμφανιστούν στο πλαίσιο διαλόγου. Οι τιμές του συνήθως ορίζονται με τη χρήση σταθερών και οι πιο βασικές είναι :



Σταθερά	Τιμή	Περιγραφή
vbOKOnly	0	Εμφάνιση OK πλήκτρου.
vbOKCancel	1	Εμφάνιση OK και Cancel πλήκτρων.
vbAbortRetryIgnore	2	Εμφάνιση Abort, Retry, και Ignore πλήκτρων.
vbYesNoCancel	3	Εμφάνιση Yes, No, και Cancel πλήκτρων.
vbYesNo	4	Εμφάνιση Yes και No πλήκτρων.
vbRetryCancel	5	Εμφάνιση Retry και Cancel πλήκτρων.
vbCritical	16	Εμφάνιση εικονίδιου κρίσιμης εργασίας.
vbQuestion	32	Εμφάνιση εικονίδιου ερώτησης.
vbExclamation	48	Εμφάνιση εικονίδιου προειδοποίησης.
vbInformation	64	Εμφάνιση εικονίδιου πληροφοριών.

Ένα όρισμα **πλήκτρα εντολής** μπορεί να οριστεί και με το άθροισμα δύο σταθερών. Για παράδειγμα δηλώνοντας στο όρισμα την τιμή vbOKCancel + vbExclamation θα εμφανιστούν στο πλαίσιο διαλόγου τα πλήκτρα OK και Cancel καθώς και το εικονίδιο προειδοποίησης.

Όταν ο χρήστης επιλέξει κάποιο από τα πλήκτρα του πλαισίου διαλόγου που εμφανίζεται με τη συνάρτηση MsgBox επιστρέφεται αντίστοιχα μια από τις παρακάτω τιμές :

Σταθερά	Τιμή	Περιγραφή
vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

Ο προγραμματιστής, αξιολογεί στο πρόγραμμα την τιμή που επιστράφηκε και εκτελεί την κατάλληλη εργασία. Ο έλεγχος του πλήκτρου που επιλέχτηκε, μπορεί να γίνει απ' ευθείας με την τιμή ή με τη χρήση της αντίστοιχης σταθεράς που επέστρεψε ή συνάρτηση MsgBox. Για παράδειγμα οι παρακάτω δομές ελέγχου είναι ισοδύναμες :

```
If Response = vbOK Then
    'ΑίδοιέΥδ έρääéää
End If
```

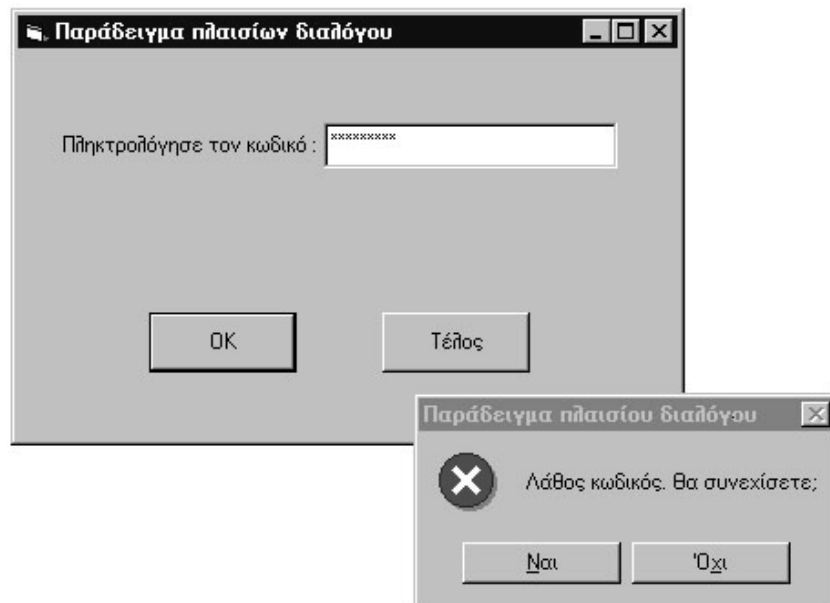
```
If Response = 1 Then
    'ΑίδοιέΥδ έρääéää
End If
```

Στη συνέχεια ακολουθεί ο κώδικας της εφαρμογής :

```
Private Sub CmdOK_Click()
    If Txtpwd.Text = Date Then
        MsgBox ("Όυόδύδ έüääéüδ !!!")
    Else
        Dim Response As String
        ' Δääßóéï äñρóρóçð ðñïð ôï ÷ñρóðç
        Response = MsgBox("Έüèïð έüääéüð. Έά όόíä÷ßóäðä;", vbYesNo +
            vbCritical, "ΔáñÛääéäíä ðääéóßïð äääéüäïð")
        If Response = vbYes Then
            ' Êäääñéóïüð éää äóðßáóç ôïð ðääéóßïð éääéýíïð
            Txtpwd.Text = ""
            Txtpwd.SetFocus
        Else
            ' ΆέóÝääóç ôçð äääääéääóßáð ðáñíäðéóïÿ
            CmdEnd_Click
        End If
    End If
End Sub
Private Sub CmdEnd_Click()
    End
End Sub
```



Με τη μέθοδο Setfocus, μπορούμε κάθε φορά να εστιάσουμε προγραμματιστικά ένα αντικείμενο στην οθόνη του προγράμματος.



Σχ. 11.2. Η εφαρμογή εισαγωγής κωδικού κατά την εκτέλεση.



Για να είναι κατανοητός ο κώδικας, φρόντισε να υπάρχει η εντολή τερματισμού του προγράμματος σε ένα μόνο σημείο. Δημιούργησε μια ρουτίνα τερματισμού του προγράμματος και κάλεσέ την, όπου χρειάζεται. Στο παραπάνω παράδειγμα, όταν ο χρήστης επιλέξει το πλήκτρο No στο πλαίσιο διαλόγου της συνάρτησης MsgBox, δεν τερματίζεται η εφαρμογή με την εντολή End, αλλά καλείται η ρουτίνα γεγονόςτος CmdEnd_Click που εκτελεί την εργασία. Σε μεγαλύτερα προγράμματα χρησιμοποίησε γενικές ρουτίνες τερματισμού του προγράμματος.

Περιβάλλον προγραμματισμού Delphi

```
program Prj2;
uses
  Forms,
  Unit2 in 'Unit2.pas' {Frmpwd};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TFrmpwd, Frmpwd);
  Application.Run;
end.
unit Unit2;
interface
```

```

uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
StdCtrls,
  Buttons, Dialogs;
type
  TFrmpwd = class(TForm)
    Label1: TLabel;
    Txtpwd: TEdit;
    CmdOK: TButton;
    CmdEnd: TButton;
    procedure CmdEndClick(Sender: TObject);
    procedure CmdOKClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Frmpwd: TFrmpwd;
implementation
{$R *.DFM}
procedure TFrmpwd.CmdEndClick(Sender: TObject);
begin
  close;
end;
procedure TFrmpwd.CmdOKClick(Sender: TObject);
begin
  if Txtpwd.Text = DateToStr(Date) then
    ShowMessage('Όυόόυò êùääéëüò !!!')
  else
    begin
      if MessageDlg( 'ËÛèìò êùääéëüò. Èá óðíâ÷ßóâðâ ?',
        mtConfirmation , [mbYes, mbNo], 0 ) = mrYES then
        begin
          Txtpwd.Text := '';
          Txtpwd.SetFocus;
        end
      else
        close;
      end;
end;
end;
end.

```

Παράδειγμα 3

Να γραφεί πρόγραμμα με το οποίο θα γίνεται η γραφική προσομοίωση της προσγείωσης ενός αεροπλάνου. Η προσομοίωση της προσγείωσης μπορεί να γίνει με την μετακίνηση ενός εργαλείου εικόνας προς το κάτω μέρος της φόρμας με παράλληλη αύξηση του μεγέθους του.

Περιβάλλον προγραμματισμού Visual Basic

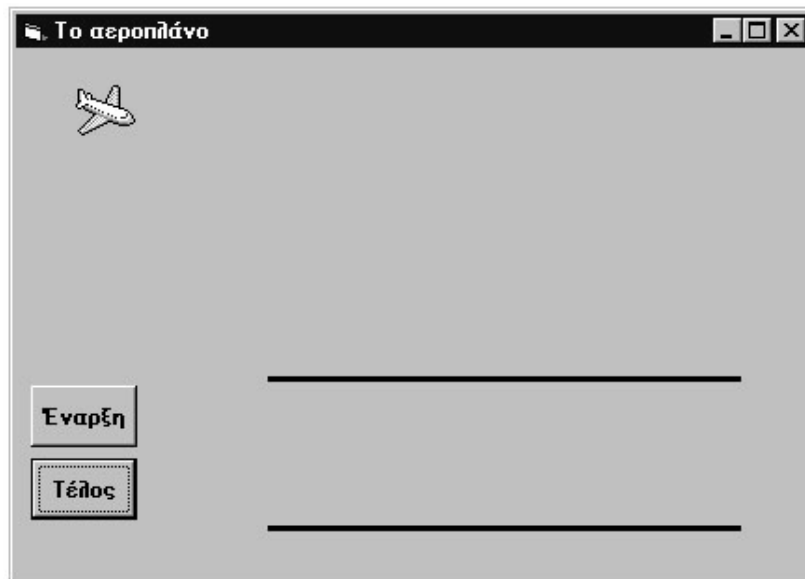
Για την υλοποίηση της εφαρμογής προτείνουμε να χρησιμοποιηθούν τα παρακάτω αντικείμενα:

- ⇒ μια **φόρμα** η οποία είναι το παράθυρο μέσα στο οποίο θα εκτελείται η εφαρμογή,
- ⇒ μια **εικόνα** (image) που θα περιέχει την εικόνα του αεροπλάνου,
- ⇒ δύο **γραμμές** με τις οποίες θα δημιουργήσουμε το διάδρομο προσγείωσης,
- ⇒ δύο **πλήκτρα εντολής** που θα χρησιμεύουν για την έναρξη και τον τερματισμό της εφαρμογής,
- ⇒ ένα εργαλείο **χρονομέτρη** (timer). Με το χρονομέτρη σε συγκεκριμένα χρονικά διαστήματα θα μετακινείται και θα αλλάζει το μέγεθος του αεροπλάνου.

Στις ιδιότητες των αντικειμένων της εφαρμογής αποδίδουμε τις παρακάτω τιμές:

Αντικείμενο	Ιδιότητα	Τιμή
Φόρμα	Name	FrmPlane
	Caption	Το αεροπλάνο
Εικόνα	Name	imgplane
	Height	480
	Picture	\\vb\graphics\icons\industry\plane.ico
	Stretch	True
	Width	480
Γραμμή1	Name	Line1
	BorderWidth	3
Γραμμή2	Name	Line2
	BorderWidth	3
Πλήκτρο εντολής 1	Name	CmdStart
	Caption	Έναρξη
Πλήκτρο εντολής 2	Name	CmdEnd
	Caption	Τέλος
Χρονομέτρης	Name	Tmrmove
	Interval	0

Μετά την τοποθέτηση των αντικειμένων και την αντιστοίχιση των ιδιοτήτων έχουμε ολοκληρώσει το τρόπο επικοινωνίας χρήστη εφαρμογής (Σχ. 11.3).



Σχ. 11.3. Η διασύνδεση του χρήστη με την εφαρμογή στο παράδειγμά μας.

Ένα εργαλείο χρονομέτρη υποστηρίζει το γεγονός Timer, το οποίο δεν προκαλείται από το χρήστη αλλά από το ρολόι του συστήματος. Η ιδιότητα Interval, καθορίζει σε χιλιοστά του δευτερολέπτου, το χρονικό διάστημα που το σύστημα θα προκαλέσει ένα γεγονός Timer. Η τιμή 0 που αποδώσαμε κατά το χρόνο σχεδιασμού αρχικά απενεργοποιεί το εργαλείο.

Το εργαλείο χρονομέτρη δεν εμφανίζεται κατά την εκτέλεση της εφαρμογής, γι αυτό το λόγο δεν έχει σημασία σε ποιο σημείο της φόρμας θα τοποθετηθεί. Φυσικά προσπαθούμε να το τοποθετήσουμε σε κάποιο σημείο που δεν θα καλύπτεται από κάποιο άλλο εργαλείο, για να γίνεται αντιληπτή κατά το χρόνο σχεδιασμού η ύπαρξη του.

Για να επιτύχουμε την προσομοίωση της προσγείωσης, όταν προκληθεί ένα γεγονός Timer με την ενέργεια της μεθόδου **Move**, μετακινούμε το αεροπλάνο, μετατρέποντας τις ιδιότητες Left και Top και αυξάνουμε το μέγεθός του μετατρέποντας τις ιδιότητες Width και Height της Εικόνας. Οι ιδιότητες αυτές εκφράζουν αντίστοιχα :

Left : την απόσταση της πάνω αριστερά γωνίας του αντικειμένου από την αριστερή πλευρά της φόρμας

Top : την απόσταση της πάνω αριστερά γωνίας του αντικειμένου από την κορυφή της φόρμας

Width: το πλάτος του εργαλείου

Height: το ύψος του εργαλείου

Η προσγείωση του αεροπλάνου θα πραγματοποιηθεί ανάμεσα στις δύο γραμμές που περιγράφουν το διάδρομο προσγείωσης. Κατά συνέπεια πρέπει να γνωρίζουμε τη θέση τους για να τερματίσουμε τη προσγείωση. Η θέση ενός αντικειμένου γραμμή καθορίζεται από τις παρακάτω ιδιότητες :



X1 : η x συντεταγμένη του σημείου έναρξης της γραμμής

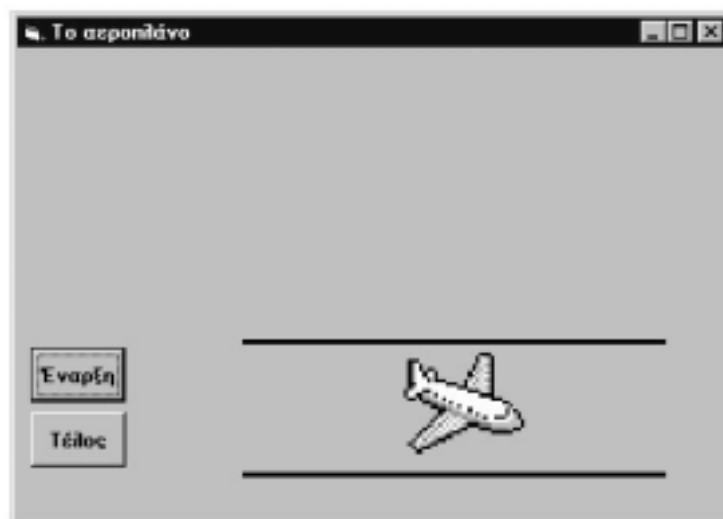
X2 : η x συντεταγμένη του σημείου τέλους της γραμμής

Y1 : η y συντεταγμένη του σημείου έναρξης της γραμμής

Y2 : η y συντεταγμένη του σημείου τέλους της γραμμής

Για την προσομοίωση της προσγείωσης του αεροπλάνου, πρέπει να συμπεριλάβουμε τις παρακάτω εντολές κώδικα στις αντίστοιχες διαδικασίες γεγονότων:

```
Private Sub CmdStart_Click()  
    ' ΔΕΡΕΘΗΙ ΥΙΑΗΙΖΘ  
    ' ΙΑ ΘΖΙ ΑΘΥΑΙΘ ΘΖΘ ΘΕΙΘ 5 ΘΘΖΙ ΕΑΕΥΘΖΘΑ Interval ΑΙΑΗΑΙΘΙΕΙΥΙΑ  
    ' ΘΙ ΑΗΑΑΕΑΒΙ +ΗΙΙΙΙΥΘΗΖ  
    ' ΕΥΘΑ ΘΥΙΘΑ +ΕΕΕΙΘΘΥ ΘΙΘ ΑΑΘΘΑΗΙΕΥΘΘΙΘ  
    Timer1.Interval = 5  
End Sub  
Private Sub Timer1_Timer()  
    ' ,ΕΑΑ+ΙΘ ΑΙ Ζ ΕΙΗΘΘΘ ΘΙΘ ΑΑΗΙΘΕΥΙΙΘ ΘΑΗΥΘΑΕ ΘΖΙ ΘΗΘΖ ΑΗΑΙΛΘ  
    ' ΘΙΘ ΑΕΑΑΗΥΙΙΘ ΘΗΙΘΑΑΒΥΘΖ.  
    ' Ζ ΕΑΕΥΘΖΘΑ ΘΙ ΘΙΘ ΑΗΑΑΕΑΒΙΘ ΑΗΑΙΛΘ ΑΙΑΘΥΗΑΘΑΕ ΘΘΖΙ Υ  
    ' ΘΘΙΘΑΑΑΑΙΥΙΖ ΘΙΘ ΘΖΙΑΒΙΘ ΥΙΑΗΙΘ ΘΖΘ.  
    If Imgplane.Top < Line1.Y1 Then  
        Imgplane.Move Imgplane.Left + 25, Imgplane.Top + 20,  
        Imgplane.Width + 5, Imgplane.Height + 5  
    Else  
        Timer1.Interval = 0  
        ' ΑΘΑΙΑΗΑΙΘΙΒΖΘΘ ΘΙΘ ΑΗΑΑΕΑΒΙΘ +ΗΙΙΙΙΥΘΗΖ  
    End If  
End Sub  
Private Sub CmdEnd_Click(Index As Integer)  
    ' ΔΕΡΕΘΗΙ ΘΑΗΙΑΘΕΘΙΥ  
    End  
End Sub
```



Σχ. 11.4. Η προσγείωση του αεροπλάνου μετά την εκτέλεση του παραδείγματος

Περιβάλλον προγραμματισμού Delphi

```
program Prj3;
uses
  Forms,
  Unit3 in 'Unit3.pas' {FrmPlane};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TFrmPlane, FrmPlane);
  Application.Run;
end.
unit Unit3;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  StdCtrls, Buttons, ExtCtrls;
type
  TFrmPlane = class(TForm)
    Imgplane: TImage;
    Panell1: TPanel;
    Panel2: TPanel;
    CmdStart: TBitBtn;
    CmdEnd: TBitBtn;
    Timer1: TTimer;
    procedure CmdStartClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure CmdEndClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  FrmPlane: TFrmPlane;
implementation
{$R *.DFM}
procedure TFrmPlane.CmdStartClick(Sender: TObject);
begin
  Timer1.Interval := 1000;
end;
procedure TFrmPlane.Timer1Timer(Sender: TObject);
begin
  if ImgPlane.Top < Panell1.Top then
  begin
    ImgPlane.Top := ImgPlane.Top + 20;
    ImgPlane.Left := ImgPlane.Left + 25;
    ImgPlane.Height := ImgPlane.Height + 5;
    ImgPlane.Width := ImgPlane.Width + 5;
```

```

        end;
    end;
    procedure TFrmPlane.CmdEndClick(Sender: TObject);
    begin
        close;
    end;
end.

```

Παράδειγμα 4

Να κατασκευαστεί μια εφαρμογή επεξεργασίας κειμένου στην οποία οι βασικές εργασίες θα πραγματοποιούνται μέσα από επιλογές ενός κεντρικού μενού επιλογών.

Περιβάλλον προγραμματισμού Visual Basic

Η υλοποίηση του επεξεργαστή κειμένου θα γίνει με τη χρήση των παρακάτω αντικείμενων:

- ⇒ μια **φόρμα** που θα είναι το παράθυρο μέσα στο οποίο θα εκτελείται η εφαρμογή,
- ⇒ ένα **πλαίσιο κειμένου** στο οποίο θα γίνεται η πληκτρολόγηση του κειμένου.

Στις ιδιότητες των αντικειμένων της εφαρμογής αποδίδουμε τις παρακάτω τιμές:

Αντικείμενο	Ιδιότητα	Τιμή
Φόρμα	Name	FrmMain
	Caption	Ο επεξεργαστής κειμένου
Πλαίσιο κειμένου	Name	TxtEdit
	Multiline	True
	Text	(κενό)

Το επόμενο βήμα είναι η δημιουργία του κεντρικού μενού επιλογών. Η Visual Basic μας παρέχει ένα εργαλείο σχεδιασμού μενού επιλογών που καλείται **επεξεργαστής μενού** (menu editor) με το οποίο σχεδιάζουμε τις επιλογές του μενού (Σχήμα 11.5). Κάθε επιλογή που δημιουργούμε αποτελεί ανεξάρτητο αντικείμενο που υποστηρίζει ένα καθορισμένο αριθμό ιδιοτήτων και επιπλέον το γεγονός κλικ. Μόλις ο χρήστης επιλέξει κάποια από τις επιλογές του μενού, ανιχνεύεται το γεγονός κλικ και εκτελείται ο κώδικας που έχουμε συμπεριλάβει στη αντίστοιχη διαδικασία γεγονότος του αντικειμένου μενού.

Μόλις ενεργοποιήσουμε το παράθυρο του επεξεργαστή μενού, δημιουργούμε τις παρακάτω επιλογές :



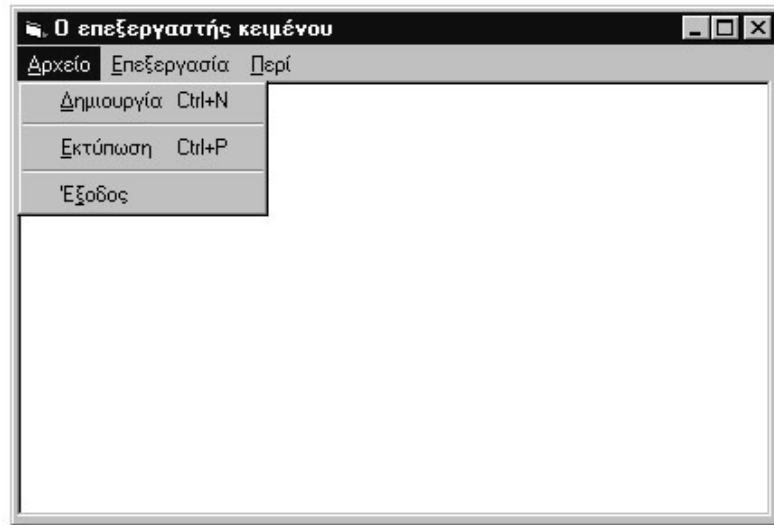
Σχ. 11.5. Ο επεξεργαστής μενού της Visual Basic.

Μενού	Επιλογή (Ιδιότητα Caption)	Όνομα (Ιδιότητα Name)	Πλήκτρο συντομίας (Ιδιότητα Shortcut)
&Αρχείο	&Δημιουργία	mnuNew	Ctrl+N
	-	separator1	
	&Εκτύπωση	mnuPrint	Ctrl+P
	-	separator2	
	Έ&ξοδος	mnuExit	
&Επεξεργασία	Απ&οκοπή	mnuCut	Ctrl+X
	Αντ&ιγραφή	mnuCopy	Ctrl+C
	&Επικόλληση	mnuPaste	Ctrl+V
	-	separator3	
	Ε&πιλογή όλων	mnuSelall	Ctrl+A
&Περί		mnuAbout	

Η ιδιότητα **Caption** περιέχει το λεκτικό με το οποίο θα εμφανίζεται η επιλογή στο μενού, η ιδιότητα **Name** είναι το όνομα που θα αναφερόμαστε από τον κώδικα στο αντικείμενο μενού και η ιδιότητα **Shortcut** εκφράζει τα πλήκτρα συντομίας με τα οποία

είναι δυνατόν να εκτελέσουμε άμεσα την επιλογή χωρίς την χρήση του μενού.

Μετά από τη δημιουργία και του κεντρικού μενού επιλογών έχει ολοκληρωθεί η δημιουργία της βασικής φόρμας της εφαρμογής (Σχήμα 11.6).



Σχ. 11.6. Η βασική φόρμα της διασύνδεσης του χρήστη με τον επεξεργαστή κειμένου

Με την επιλογή **Περί** του κεντρικού μενού επιλογών πρόκειται να εμφανίζουμε μια δεύτερη φόρμα που θα περιέχει πληροφοριακά στοιχεία για τη εφαρμογή. Αφού προσθέσουμε στο έργο (project) μια νέα φόρμα τοποθετούμε επάνω της τα παρακάτω αντικείμενα :

- ⇒ δύο **ετικέτες** που θα περιέχουν τα πληροφοριακά στοιχεία της εφαρμογής,
- ⇒ ένα **σχήμα** (shape) για την ομαδοποίηση των ετικετών,
- ⇒ μια **εικόνα** ως διακοσμητικό,
- ⇒ ένα **πλήκτρο εντολής** που θα κλείνει τη δεύτερη φόρμα.

Στις ιδιότητες των αντικειμένων της δεύτερης φόρμας αποδίδουμε τις παρακάτω τιμές:

Αντικείμενο	Ιδιότητα	Τιμή
Φόρμα	Caption	Πληροφορίες για το πρόγραμμα
	Name	FrmAbout
Ετικέτα1	Alignment	2-Center
	Caption	Κειμενογράφος
	ForeColor	&H8000000D& (Highlight)
	Name	Lblinfo1
Ετικέτα2	Alignment	2-Center
	Caption	Παράδειγμα μενού επιλογών και χρήση ειδικών αντικειμένων
	ForeColor	&H8000000D& (Highlight)
	Name	Lblinfo2
Σχήμα	ForeColor	&H8000000D& (Highlight)
	Name	Shape1
	Shape	0-Rectangle
Εικόνα	Name	Image1
	Picture	\\vb\graphics\icons\writing\note16.ico
	Stretch	True
Πλήκτρο εντολής	Caption	OK
	Name	CmdBack

Με τον σχεδιασμό της δεύτερης φόρμας (Σχήμα 11.7) ολοκληρώθηκε η διεπαφή του χρήστη με τον επεξεργαστή κειμένου.



Σχ. 11.7. Η δεύτερη φόρμα του επεξεργαστή κειμένου

Στον κώδικα του συγκεκριμένου παραδείγματος πρέπει να χρησιμοποιήσουμε δύο ειδικά αντικείμενα που υποστηρίζει η Visual Basic, το **Printer** που εκφράζει τον εκτυπωτή και το **Clipboard** (Πρόχειρο) που εκφράζει μια περιοχή μνήμης, η οποία είναι προσπελάσιμη από όλες τις εφαρμογές του συστήματος.

Το ειδικό αντικείμενο Printer δέχεται δεδομένα με τη χρήση της μεθόδου **Print**. Όταν ολοκληρώσουμε την τοποθέτηση των δεδομένων στο ειδικό αντικείμενο Printer, στέλνουμε τα περιεχόμενά του για εκτύπωση με τη μέθοδο **EndDoc**.

Με τη χρήση του Clipboard μπορούμε να χειριστούμε δεδομένα κειμένου ή γραφικών. Όταν πρόκειται για αποστολή κειμένου από την εφαρμογή προς το Clipboard πρέπει να χρησιμοποιήσουμε τη μέθοδο **SetText** και για λήψη τη μέθοδο **GetText**, ενώ όταν πρόκειται για γραφικά χρησιμοποιούμε αντίστοιχα τις μεθόδους **SetData** και **GetData**.

Ακόμη για την επιλογή **Επιλογή όλων** του μενού θα χρησιμοποιήσουμε τη συνάρτηση **Len**, η οποία υπολογίζει τον αριθμό χαρακτήρων ενός αλφαριθμητικού και επιπλέον τις παρακάτω ιδιότητες που υποστηρίζει το πλαίσιο κειμένου :

- ⇒ **SellLength** : επιστρέφει τον αριθμό των επιλεγμένων χαρακτήρων.
- ⇒ **SelStart** : επιστρέφει το σημείο έναρξης ενός επιλεγμένου κειμένου ή τοποθετεί το σημείο εισαγωγής σε συγκεκριμένο σημείο.
- ⇒ **SelText** : περιέχει το κείμενο που έχει επιλεγεί ή το μηδενικού μήκους αλφαριθμητικό ("") όταν δεν έχει επιλεγεί τίποτα.

Τα βασικά τμήματα του κώδικα της εφαρμογής πρέπει να τοποθετηθούν στις διαδικασίες γεγονότων κλικ των αντικειμένων μενού.

```
Private Sub mnunew_Click()
    ' Άεά öíí έάέάñέöíü öíö ðέάέóßíö έάέíÝííö áðíäßäíöíä öí
    ' ìçääíέêíÝ ìßêíöð áέöáñέêìçðέêü öðçí έάέüðçöä Text.
    Text1.Text = ""
End Sub
Private Sub mnuprint_Click()
    ' Άέöçðüðç öüí ðáñέä+ìíÝíüí öíö ðέάέóßíö έάέíÝííö TxtEdit
    ' ìä öç +ñßóç öíö áέäέêíÝ áíöέέάέíÝííö Printer
    Printer.Print TxtEdit.Text
    Printer.EndDoc
End Sub
Private Sub mnuexit_Click()
    End
End Sub
Private Sub mnucut_Click()
    ' Άðíöðíêß öüí ðáñέä+ìíÝíüí öíö ðέάέóßíö έάέíÝííö TxtEdit öðí
    ' áέäέêü áíöέέäßíäíí Clipboard.
    Clipboard.SetText TxtEdit.SelText
    ' Έάέáñέöíüð öüí ðáñέä+ìíÝíüí öíö ðέάέóßíö έάέíÝííö TxtEdit
    TxtEdit.SelText = ""
End Sub
Private Sub mnucopy_Click()
```

```

        ' Άδιόδοιέβ' ουί δάνεά+ιιΎιιι όιό δέάέόβιό έάέιΎιιό TxtEdit όδι
                                άέάέέü άίόέέάβιιί Clipboard.
Clipboard.SetText TxtEdit.SelText
End Sub
Private Sub mnupaste_Click()
    ' ιάόάδιñÜ ουί δάνεά+ιιΎιιι όιό Clipboard όδι δέάέόβιό έάέιΎιιό
                                                TxtEdit

    TxtEdit.SelText = Clipboard.GetText
End Sub
Private Sub mnuSelall_Click()
    ' ΌιόιέΎόζόζ όιό όζιιβιό άέόάüñάβδ έάέιΎιιό όόζι άñ+β όιό
                                δέάέόβιό έάέιΎιιό TxtEdit.

    TxtEdit.SelStart = 0
    ' Άόέέιιιü üέüί ουί δάνεά+ιιΎιιι όιό δέάέόβιό έάέιΎιιό TxtEdit.
    TxtEdit.SelLength = Len(TxtEdit.Text)
End Sub
Private Sub mnuAbout_Click()
    ' ιά όζ ιΎέιιι Show άιόάίβæάόάέ ζ ääγδάνζ öüñιι.
    FrmAbout.Show
End Sub

```

Στον κώδικα της δεύτερης φόρμας θα συμπεριλάβουμε στη διαδικασία γεγονότος κλικ του πλήκτρου εντολής OK τον κώδικα με τον οποίο θα αποκρύπτεται η φόρμα.

```

Private Sub CmndBack_Click()
    ' ιά όζ ιΎέιιι Hide έñγääόάέ ζ ääγδάνζ öüñιι.
    FrmAbout.Hide
End Sub

```



Στο πλαίσιο κειμένου έχουμε αποδώσει την τιμή True στην ιδιότητα Multiline. Έτσι όταν γεμίσει μια γραμμή του πλαισίου αυτόματα μεταφέρεται το σημείο εισαγωγής κειμένου στην επόμενη γραμμή. Αυτό ίσως μας δημιουργήσει προβλήματα κατά την εκτύπωση, γιατί η μέθοδος Print δεν στέλνει στο ειδικό αντικείμενο Printer κανένα χαρακτήρα αλλαγής γραμμής. Η λύση είναι να μεταφέρει ο χρήστης το σημείο εισαγωγής στην επόμενη γραμμή πατώντας το πλήκτρο Enter.

Περιβάλλον προγραμματισμού Delphi

```

program Prj4;
uses
    Forms,
    Unit4 in 'Unit4.pas' {Form1},
    Unit41 in 'Unit41.pas' {FrmAbout};
{$R *.RES}
begin
    Application.Initialize;
    Application.CreateForm(TForm1, Form1);
    Application.CreateForm(TFrmAbout, FrmAbout);
    Application.Run;
end.
unit Unit4;

```

```

interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs,
    Menus, StdCtrls, Printers;
type
    TForm1 = class(TForm)
        Memo1: TMemo;
        MainMenu1: TMainMenu;
        mnuEdit: TMenuItem;
        mnuFile: TMenuItem;
        mnuAbout: TMenuItem;
        mnuNew: TMenuItem;
        mnuPrint: TMenuItem;
        mnuExit: TMenuItem;
        mnuCut: TMenuItem;
        mnuPaste: TMenuItem;
        mnuCopy: TMenuItem;
        mnuSelAll: TMenuItem;
        procedure mnuExitClick(Sender: TObject);
        procedure mnuCutClick(Sender: TObject);
        procedure mnuCopyClick(Sender: TObject);
        procedure mnuPasteClick(Sender: TObject);
        procedure mnuNewClick(Sender: TObject);
        procedure mnuSelAllClick(Sender: TObject);
        procedure mnuAboutClick(Sender: TObject);
        procedure mnuPrintClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    Form1: TForm1;
implementation
uses Unit41;
{$R *.DFM}
procedure TForm1.mnuExitClick(Sender: TObject);
begin
    close;
end;
procedure TForm1.mnuCutClick(Sender: TObject);
begin
    Memo1.CutToClipboard;
end;
procedure TForm1.mnuCopyClick(Sender: TObject);
begin
    Memo1.CopyToClipboard;
end;
procedure TForm1.mnuPasteClick(Sender: TObject);

```

```
begin
    Memol.PasteFromClipboard;
end;
procedure TForm1.mnuNewClick(Sender: TObject);
begin
    Memol.Clear;
end;
procedure TForm1.mnuSelAllClick(Sender: TObject);
begin
    Memol.SelStart := 0;
    Memol.Sellength := 32767;
end;
procedure TForm1.mnuAboutClick(Sender: TObject);
begin
    FrmAbout.Show;
end;
procedure TForm1.mnuPrintClick(Sender: TObject);
var
    index : Integer;
begin
    Printer.BeginDoc;
    for index := 0 to Memol.Lines.Count-1 do
        Printer.Canvas.TextOut(100,10+index*50,Memol.Lines[index]);
    Printer.EndDoc;
end;
end
unit Unit41;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs,
    StdCtrls, ExtCtrls;
type
    TFrmAbout = class(TForm)
        CmndBack: TButton;
        Panell1: TPanel;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Image1: TImage;
        procedure CmndBackClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    FrmAbout: TFrmAbout;
implementation
```

```
{ $R *.DFM }
procedure TFrmAbout.CmdnBackClick(Sender: TObject);
begin
    FrmAbout.Hide;
end;
end.
```

Παράδειγμα 5

Να κατασκευάσετε μια εφαρμογή παρουσίασης των ελληνικών μεταλλίων στους Ολυμπιακούς Αγώνες της Ατλάντα. Δημιουργήστε ολοκληρωμένη παρουσίαση χρησιμοποιώντας και αρχεία βίντεο.

Περιβάλλον προγραμματισμού Visual Basic

Για την υλοποίηση της παρουσίασης θα χρησιμοποιήσουμε τα παρακάτω αντικείμενα:

- ⇒ μια **φόρμα** που θα είναι το παράθυρο μέσα στο οποίο θα εκτελείται η εφαρμογή,
- ⇒ τεσσάρων **πλήκτρων επιλογής** (option buttons) για την επιλογή του αθλήματος,
- ⇒ ένα **πλαίσιο** (frame) στο οποίο θα γίνεται η ομαδοποίηση των πλήκτρων επιλογής,
- ⇒ έξι **ετικετών** που θα χρησιμεύουν για την εμφάνιση των πληροφοριών της εφαρμογής,
- ⇒ δύο **πλήκτρων εντολής** για την έναρξη του βίντεο και τον τερματισμό της εφαρμογής.

Στις ιδιότητες των αντικειμένων της εφαρμογής αποδίδουμε τις παρακάτω τιμές:

Αντικείμενο	Ιδιότητα	Τιμή
Φόρμα	Name	FrmMain
	Caption	Παραδείγματα video
Πλήκτρο επιλογής 1	Name	Option1
	Caption	Άρση βαρών
Πλήκτρο επιλογής 2	Name	Option2
	Caption	Γυμναστική
Πλήκτρο επιλογής 3	Name	Option3
	Caption	Ιστιοπλοΐα
Πλήκτρο επιλογής 4	Name	Option4
	Caption	Στίβος

Αντικείμενο	Ιδιότητα	Τιμή
Πλαίσιο	Name	FrOptions
	Caption	ΑΘΛΗΜΑΤΑ
Ετικέτα0	Name	LblTitle0
	Caption	Ελληνικά μετάλλια στην Ατλάντα
Ετικέτα1	Name	LblTitle1
	Caption	Αθλητής:
Ετικέτα2	Name	LblTitle2
	Caption	Αγώνισμα:
Ετικέτα3	Name	LblTitle3
	Caption	Μετάλλιο:
Ετικέτα4	Name	LblFullName
	BorderStyle	1-Fixed Single
	Caption	(κενό)
Ετικέτα5	Name	LblSport
	BorderStyle	1-Fixed Single
	Caption	(κενό)
Ετικέτα6	Name	LblMedal
	BorderStyle	1-Fixed Single
	Caption	(κενό)
Πλήκτρο εντολής 1	Name	CmdStart
	Caption	Έναρξη
Πλήκτρο εντολής 2	Name	CmdEnd
	Caption	Τέλος

Στη συνέχεια πρέπει να δώσουμε στην εφαρμογή μας και δυνατότητα χρήσης βίντεο. Για να το επιτύχουμε πρέπει να συνεργαστεί η εφαρμογή μας με κάποια εφαρμογή εκτέλεσης αρχείων βίντεο. Η συνεργασία των εφαρμογών επιτυγχάνεται με την αξιοποίηση της τεχνολογίας **OLE** (Object Linking and Embedding) που υποστηρίζει η Visual Basic και γενικότερα οι εφαρμογές που εργάζονται στο περιβάλλον των Windows.

Στο παράδειγμά μας θα επιδείξουμε μόνο το βίντεο του αθλήματος της ιστιοσανίδας και με τον ίδιο τρόπο μπορείς να υλοποιήσεις τα υπόλοιπα OLE εργαλεία. Για τη σύνδεση των εφαρμογών, επιλέγουμε από την εργαλειοθήκη και τοποθετούμε ένα αντικείμενο OLE επάνω στη φόρμα της εφαρμογής μας. Αμέσως εμφανίζεται το παράθυρο διαλόγου **Εισαγωγή αντικειμένου**. Στο παράθυρο διαλόγου επιλέγουμε **Δη-**

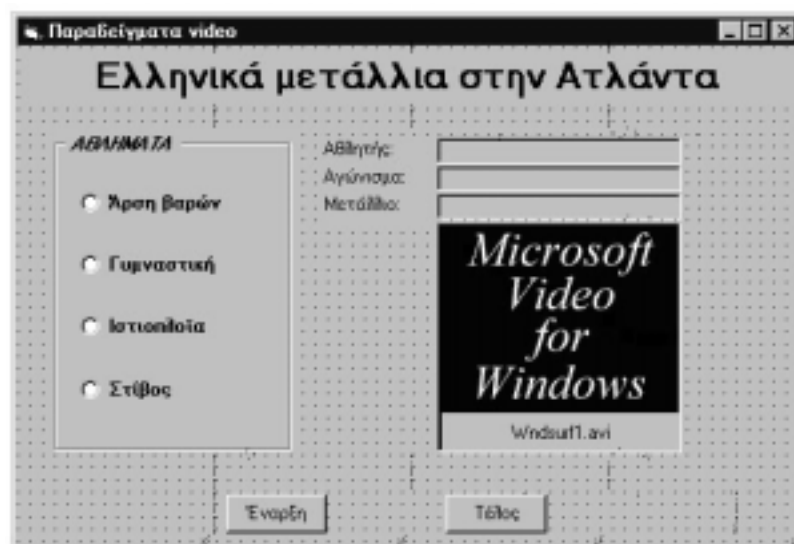
μιουργία από αρχείο και πατώντας το πλήκτρο εντολής **Αναζήτηση** εντοπίζουμε από το δίσκο του συστήματος το αρχείο .avi που πρόκειται να χρησιμοποιήσουμε. Στο παράδειγμά μας θα χρησιμοποιήσουμε το αρχείο windsurf2.avi. Εκτελώντας τις παραπάνω ενέργειες θα επιτύχουμε τη διασύνδεση των περιεχομένων του αρχείου σαν αντικείμενο της εφαρμογής μας. Όταν ενεργοποιήσουμε το αντικείμενο OLE το βίντεο θα εκτελεστεί μέσα από το πρόγραμμα που έχει δημιουργηθεί. Στο αντικείμενο OLE αντιστοιχούμε ακόμη τις παρακάτω ιδιότητες:

Αντικείμενο	Ιδιότητα	Τιμή
OLE	Name	OLE3
	Sizemode	2-Autosize
	Visible	False



Μετά τα δημιουργία του αντικειμένου OLE στην ιδιότητα **Class** έχει τοποθετηθεί η τιμή **aviFile** και στην ιδιότητα **SourceDoc** έχει τοποθετηθεί η διαδρομή και το όνομα του αρχείου βίντεο που διασυνδέσαμε.

Μετά την αντιστοίχιση των ιδιοτήτων των αντικειμένων έχουμε ολοκληρώσει το τρόπο επικοινωνίας της εφαρμογής με το χρήστη.



Σχ. 11.8. Η φόρμα της εφαρμογής παρουσίασης

Στη συνέχεια πρέπει να γράψουμε το κώδικα της εφαρμογής:

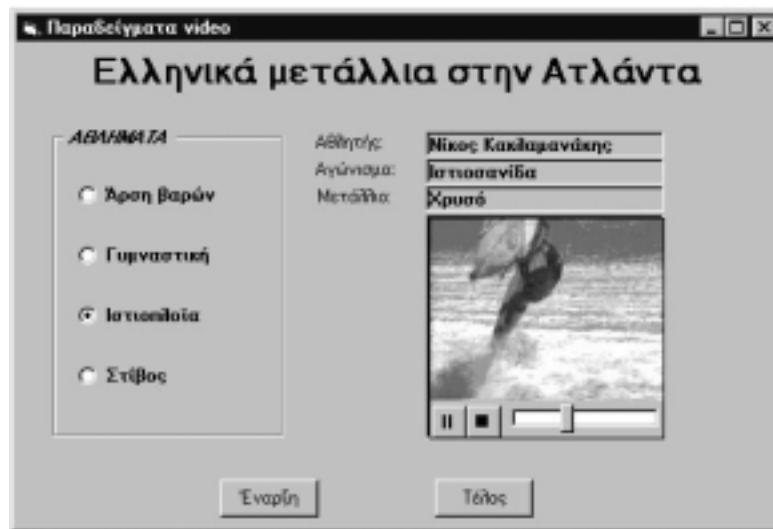
```

Private Sub CmdStart_Click()
    ' ,έάã+ιò äðέääĩÝíiö äèèΠιάöiö
    If Option3.Value = True Then
        ' ÄðέéÝ+ðçêä ôï Ûèèçìá ðçð Éóðéiöáíßääð
        ' ÄíçìÝñùòç äðééääòbí
        lblFullName.Caption = "Íßéið ÊáééáíáíÛêçð"
        lblSport.Caption = "Éóðéiöáíßää"
        lblMedal.Caption = "×ñðóü"
        'Äíáñãäiðißçóç ôiö áiðééääéiÝíiö OLE
        OLE3.Action = 7
    End If
End Sub
Private Sub CmdEnd_Click()
    End
End Sub

```



Στον κώδικα της διαδικασίας CmdStart_Click() δεν χρησιμοποιούμε την ιδιότητα Visible για να εμφανίσουμε ή να αποκρύψουμε το εργαλείο OLE, γιατί επιτυγχάνεται αυτόματα με την ενεργοποίηση της εφαρμογής βίντεο.



Σχ. 11.9. Η εφαρμογή παρουσίασης κατά το χρόνο εκτέλεσης



Κατά την εκτέλεση της εφαρμογής μαζί με το πλαίσιο προβολής βίντεο, εμφανίζονται δύο πλήκτρα ελέγχου που επιτρέπουν το πρώτο τη παύση ή τη συνέχιση της προβολής βίντεο και το δεύτερο τη διακοπή της.

Περιβάλλον προγραμματισμού Delphi

```

program Prj5;
uses
  Forms,
  Unit5 in 'Unit5.pas' {FrmMain};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TFrmMain, FrmMain);
  Application.Run;
end.
unit Unit5;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  StdCtrls, ExtCtrls, MPlayer;
type
  TFrmMain = class(TForm)
    OptionGroup: TRadioGroup;
    LblTitle0: TLabel;
    LblTitle1: TLabel;
    LblTitle2: TLabel;
    LblTitle3: TLabel;
    FullName: TEdit;
    Sport: TEdit;
    Medal: TEdit;
    CmdStart: TButton;
    MediaPlayer1: TMediaPlayer;
    Panell: TPanel;
    CmdEnd: TButton;
    procedure CmdStartClick(Sender: TObject);
    procedure CmdEndClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  FrmMain: TFrmMain;
implementation
{$R *.DFM}
procedure TFrmMain.CmdStartClick(Sender: TObject);
begin
  if OptionGroup.ItemIndex = 2 then
  begin
    Fullname.Text := ' Íßêïð ÊáêêáíáíÛêçð;
    Sport.Text := ' Êóóéïóáíßää';
    Medal.Text := '×ñóóù';
  end;
end;

```

```
MediaPlayer1.Open;  
MediaPlayer1.Rewind;  
MediaPlayer1.Play;  
end;  
  
end;  
  
procedure TFrmMain.CmdEndClick(Sender: TObject);  
begin  
    close;  
end;  
  
end.
```

Παράδειγμα 6

Υλοποιήστε κατά το χρόνο σχεδιασμού μιας Visual Basic εφαρμογής τη σύνδεσή της με μια άλλη εφαρμογή του περιβάλλοντος των Windows, όπως το Microsoft Excel.

Περιβάλλον προγραμματισμού Visual Basic

Εάν θέλουμε μια εφαρμογή μας της Visual Basic να χρησιμοποιήσει δεδομένα που ανανεώνονται συνέχεια και προέρχονται από κάποια άλλη εφαρμογή, τότε θα πρέπει να χρησιμοποιήσουμε τις δυνατότητες που μας προσφέρει η Dynamic Data Exchange (DDE). Μα τι είναι η DDE; Είναι ένας μηχανισμός που υποστηρίζεται από τα Windows και επιτρέπει σε δύο εφαρμογές να ανταλλάσσουν αυτόματα μεταξύ τους δεδομένα. Οι δύο εφαρμογές που ανταλλάσσουν πληροφορίες εμπλέκονται σε μία DDE “συνομιλία”. Η ανταλλαγή των δεδομένων προϋποθέτει κατ’ αρχήν τη δημιουργία συνδέσμου. Η εφαρμογή η οποία αρχίζει τη “συνομιλία” ονομάζεται προορισμός (destination) και η εφαρμογή που ανταποκρίνεται ονομάζεται πηγή (source).

Κατά τη διάρκεια μιας DDE “συνομιλίας” πρέπει να προσδιοριστεί το θέμα της (topic). Κατά τη διάρκεια της “συνομιλίας” ο προορισμός και η πηγή μπορούν να ανταλλάσσουν πληροφορίες που αφορούν ένα ή περισσότερα άρθρα (items) σχετικά με το θέμα, τα οποία δεν είναι παρά αναφορές σε δεδομένα που ενδιαφέρουν και τις δύο εφαρμογές. Το τι ακριβώς μπορεί να είναι ένα άρθρο εξαρτάται από την εφαρμογή. Για παράδειγμα, το Excel αναγνωρίζει σαν άρθρο σε μία “συνομιλία”, τις συντεταγμένες των κελιών (cells) ενός λογιστικού φύλλου.

Η Visual Basic υποστηρίζει διαδικασίες DDE και κάθε αντικείμενο πλαίσιο κειμένου, πλαίσιο εικόνας και ετικέτα, μπορεί να θεωρηθεί σαν προορισμός, ενώ τα προαναφερθέντα αντικείμενα και κάθε φόρμα θεωρείται σαν πηγή. Στην περίπτωση που μία φόρμα σε μία εφαρμογή της Visual Basic, είναι η πηγή μιας “συνομιλίας”, το όνομα καθενός από τα αντικείμενα πλαίσιο κειμένου, πλαίσιο εικόνας και ετικέτα επάνω στη φόρμα, μπορεί να αποτελεί άρθρο για την DDE “συνομιλία”. Σημειώνεται ότι μία εφαρμογή μπορεί να εμπλακεί ταυτόχρονα σε πολλές “συνομιλίες” λειτουργώντας σε άλλες σαν προορισμός και σε άλλες σαν πηγή.

Η δημιουργία συνδέσμου, μεταξύ της εφαρμογής της Visual Basic και της άλλης εφαρμογής με την οποία θα ανταλλάσσουν δεδομένα, μπορεί να γίνει είτε στη φάση σχεδίασης της εφαρμογής, είτε στη φάση της εκτέλεσής της. Οι σύνδεσμοι που δη-

μιουργούνται στη φάση της σχεδίασης είναι απλούστεροι, αλλά θέτουν κάποιους περιορισμούς :

- ✓ Κατ' αρχήν η άλλη εφαρμογή θα πρέπει να περιέχει στο μενού επιλογών Edit την επιλογή Paste Link.
- ✓ Επιπλέον, όταν εκτελείται η εφαρμογή της Visual Basic, η άλλη εφαρμογή θα πρέπει να είναι διαθέσιμη εκείνη τη στιγμή.
- ✓ Τέλος, επειδή η ενημέρωση του αντικείμενου της εφαρμογής της Visual Basic που περιέχει τα δεδομένα που προέρχονται από την άλλη εφαρμογή, γίνεται αυτόματα όποτε σημειώνεται κάποια αλλαγή, αυτό μπορεί να έχει σαν αποτέλεσμα την αποτυχία της ενημέρωσης (κάτω από ορισμένες συνθήκες), αν η άλλη εφαρμογή δεν έχει δυνατότητα κάτω από αυτές τις συνθήκες να ανανεώσει το περιεχόμενο του συνδέσμου.

Σε μία εφαρμογή της Visual Basic όταν ένα αντικείμενο της είναι ο προορισμός μιας “συνομιλίας”, η Visual Basic φροντίζει να καθορίσει το άρθρο της συνομιλίας θέτοντας ανάλογη τιμή στην ιδιότητα LinkItem αυτού του αντικείμενου.

Μία DDE “συνομιλία” πολύ συχνά καλείται σύνδεσμος (link) επειδή οι δύο εφαρμογές συνδέονται μέσω των δεδομένων που ανταλλάσσουν. Υπάρχουν τρία είδη συνδέσμων, με σημείο διαφοροποίησης τον τρόπο με τον οποίο ενημερώνεται ο προορισμός όταν αλλάζουν τα δεδομένα στην πηγή:

- ✓ Αυτόματος σύνδεσμος (automatic link): Η πηγή προμηθεύει με δεδομένα τον προορισμό κάθε φορά που αλλάζουν τα δεδομένα που προσδιορίζονται από το άρθρο σύνδεσης (link item).
- ✓ Χειροκίνητος σύνδεσμος (manual link): Η πηγή παρέχει πληροφορίες για τις τυχόν αλλαγές που έχουν επέλθει στα δεδομένα του άρθρου μόνο όταν το ζητήσει ο προορισμός.
- ✓ Σύνδεσμος γνωστοποίησης (notify link): Η πηγή γνωστοποιεί στον προορισμό ότι τα δεδομένα άλλαξαν, αλλά παρέχει αυτές τις αλλαγές των δεδομένων μόνο όταν της το ζητήσει ο προορισμός.

Ακριβώς αυτή η δυνατότητα που παρουσιάζει η Visual Basic, για δημιουργία συνδέσμων μεταξύ διαφορετικών εφαρμογών που υποστηρίζουν διαδικασίες DDE, προσδίδει δυνατότητες πλοήγησης και χαρακτηριστικά Hypermedia σε καλά οργανωμένες και δομημένες εφαρμογές της.

Έστω ότι θέλουμε να δημιουργήσουμε έναν σύνδεσμο ανάμεσα σε μία εφαρμογή της Visual Basic και σε ένα βιβλίο εργασίας του Excel από το οποίο παίρνει στοιχεία και ενημερώνεται για τις τιμές υλικού και λογισμικού πολυμέσων. Έχουμε ήδη ανοίξει και τις δύο εφαρμογές μας (της Visual Basic και του Excel), και βρισκόμαστε στην εφαρμογή του Excel, στο βιβλίο εργασίας prices.xls. Επιλέγουμε εκείνα τα δεδομένα που θέλουμε να συνδέσουμε με την εφαρμογή μας της Visual Basic (Σχήμα 11.10).

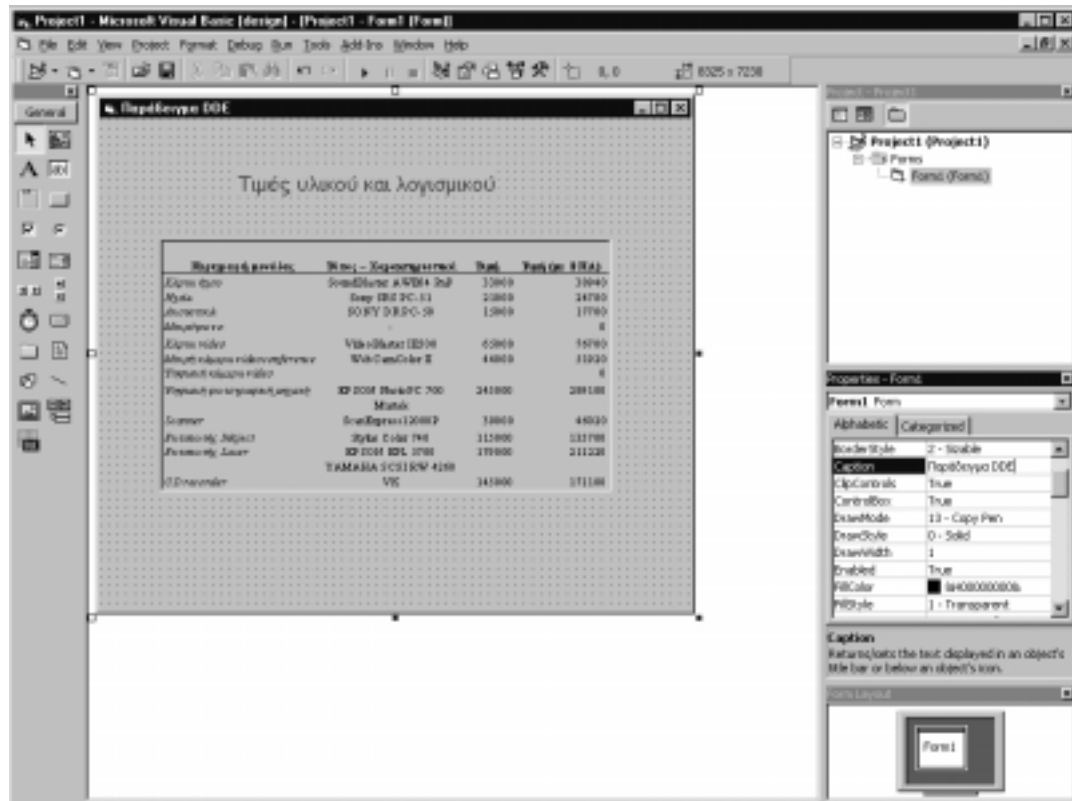
	A	B	C	D	E
1	ΕΞΟΠΛΙΣΜΟΣ ΠΟΛΥΜΕΣΩΝ				
2					
3	HARDWARE				
4					
5	Περιγραφή μονάδας	Τίτλος - Χαρακτηριστικά	Τιμή	Τιμή με ΦΠΑ	
6	Κάρτα ήχου	SoundBlaster AWE64 Pro	33000	38940	
7	Ηχεία	Sony SRS PC-51	21000	24780	
8	Δορυφόροι	SONY DR-PC-50	15000	17700	
9	Μικροφώνια	-		0	
10	Κάρτα βίντεο	VideoBlaster IE500	65000	76700	
11	Μικρή κάμερα τηλεοράσεων	Web-CamColor II	44000	51920	
12	Μηχανή κάμερα βίντεο			0	
13	Φθιστική φωτογραφική μηχανή	EPSON PhotoPC 700	245000	289100	
14	Scanner	Mastek ScanExpress1200IP	39000	46020	
15	Εκτυπωτής Inkjet	Stylus Color 740	115000	135700	
16	Εκτυπωτής Laser	EPSON EPL 5700	179000	211220	
17	CD recorder	YAMAHA SCSI RW 4260	145000	171100	
18	Fast drive / SCSI int	VK	109000	128620	
19	Fast drive / SCSI ext		191000	225380	
20	Fast diskette / IOB		34000	40120	
21	Fast diskette / IOB		43000	50740	
22	Zip drive ext parallel		46000	54280	
23	Zip drive ext SCSI		54000	63720	

Σχ. 11.10. Επιλογή δεδομένων από την εφαρμογή του Excel

Στη συνέχεια από τη γραμμή μενού του Excel επιλέγουμε Edit, μετά Copy. Στη συνέχεια επιστρέφουμε στην εφαρμογή μας της Visual Basic χωρίς να κλείσουμε την εφαρμογή στο Excel. Επιλέγουμε ένα εργαλείο, για παράδειγμα ένα πλαίσιο εικόνας, το οποίο μπορεί να χρησιμοποιηθεί σαν αυτό που θα λάβει τα δεδομένα και σχηματίζουμε το αντίστοιχο αντικείμενο. Από τη γραμμή μενού της Visual Basic επιλέγουμε Edit και μετά Paste Link. Εάν ο σύνδεσμος έχει πραγματοποιηθεί με επιτυχία, τα δεδομένα που είχαμε επιλέξει από το Excel, αντιγράφονται στην εφαρμογή μας της Visual Basic (Σχήμα 11.11).

Είναι αξιοσημείωτο πάντως, πως παρ' όλο που στην εφαρμογή μας στο Excel επιλέξαμε δύο ασυνεχή τμήματα, στην εφαρμογή μας της Visual Basic αντιγράφηκε ένα ενιαίο τμήμα, στο οποίο συμπεριλαμβάνεται και το ενδιάμεσο μεταξύ των δύο επιλεγμένων τμημάτων τμήμα, που δεν είχαμε επιλέξει.

Παρατηρούμε τις τιμές που έχουν πάρει πλέον οι ιδιότητες του πλαισίου εικόνας το οποίο χρησιμοποιήσαμε για να πάρουμε τα δεδομένα από το Excel. Στην επιλογή LinkItem έχουν παρουσιαστεί οι συντεταγμένες που προσδιορίζουν το τμήμα του λογιστικού φύλλου που συνδέθηκε με την εφαρμογή μας της Visual Basic. Ακόμα, πάντα στο πλαίσιο ιδιοτήτων, στην επιλογή LinkMode εμφανίζεται η τιμή Automatic, στην επιλογή LinkTopic εμφανίζεται μαζί με την πλήρη διαδρομή και το όνομα της εφαρμογής του Excel και τέλος στην επιλογή Picture εμφανίζεται ο χαρακτηρισμός Metafile, που φανερώει ακριβώς τη μορφή με την οποία απεικονίζονται τα δεδομένα από το Excel, στο πλαίσιο εικόνας που δημιουργήσαμε στην εφαρμογή της Visual Basic.



Σχήμα 11.11. Μεταφορά των δεδομένων του Excel στην εφαρμογή της Visual Basic

Έχοντας πάντοτε ανοικτές και τις δύο εφαρμογές, της Visual Basic και του Excel, μπορούμε να κάνουμε κάποιες αλλαγές στο Excel και να διαπιστώσουμε, γυρνώντας αμέσως μετά στην εφαρμογή της Visual Basic, ότι οι αλλαγές αυτές μεταφέρονται και σε στην εφαρμογή μας της Visual Basic.

Πειραματιστείτε αλλάζοντας στην εφαρμογή του Excel την τιμή κάποιων από τα υλικά που έχουμε απεικονίσει και στην εφαρμογή της Visual Basic και διαπιστώστε μόνοι σας ότι οι αλλαγές αυτές αμέσως στη συνέχεια αντικατοπτρίζονται και στην εφαρμογή της Visual Basic.

Η υλοποίηση αυτού του παραδείγματος δεν παρουσιάζεται και σε περιβάλλον Delphi, γιατί η τεχνική δημιουργίας DDE συνδέσεων κατά το σχεδιασμό μιας Delphi εφαρμογής με κάποια άλλη εφαρμογή στο περιβάλλον των Windows πραγματοποιείται επίσης μέσω του Clipboard. Στο περιβάλλον προγραμματισμού Delphi για την DDE συνομιλία χρειαζόμαστε τα DDEClientConv και DDEClientItem components.

11.3. Συμβουλές - υποδείξεις

- ⇒ Κατά, το σχεδιασμό και την υλοποίηση των εφαρμογών είναι σκόπιμο να εκμεταλλευτείς όλες τις δυνατότητες που σου παρέχει ένα σύγχρονο περιβάλλον. Χρησιμοποίησε τα κατάλληλα εργαλεία και δημιούργησε ένα φιλικό, ευχάριστο και δη-

μιουργικό περιβάλλον εργασίας. Ακόμη μάθε να αξιοποιείς τις δυνατότητες που σου παρέχουν οι καινούργιες τεχνικές προγραμματισμού όπως ο αντικειμενοστραφής και οδηγούμενος από γεγονότα προγραμματισμός.

- ⇒ Δημιούργησε ομοιόμορφες και εύχρηστες εφαρμογές. Φρόντισε ο τρόπος επικοινωνίας των εφαρμογών σου με το χρήστη να ακολουθεί τις γενικές αρχές εφαρμογών που γνωρίζουν οι χρήστες από εφαρμογές που ήδη χρησιμοποιούν (επεξεργαστές κειμένου, λογιστικά φύλλα κ.ά). Χρησιμοποίησε στις εφαρμογές σου μενού εντολών, κλασικά πλαίσια διαλόγου, εργαλειοθήκες και ότι νομίζεις ότι θα διευκολύνει το χρήστη.
- ⇒ Κάθε εφαρμογή σου, κατά το χρόνο εκτέλεσης, πρέπει να συνοδεύεται από αρχεία βοήθειας και τεχνικές καθοδήγησης του χρήστη. Εμφάνιζε συχνά όπου είναι εφικτό επεξηγηματικά πλαίσια (tooltips), γραμμές εργαλείων (toolbars), πτυσσόμενα μενού επιλογών (popup menus) και γραμμές κατάστασης (status lines).
- ⇒ Σε μια εφαρμογή μπορείς να χρησιμοποιήσεις μια ή περισσότερες φόρμες. Κάθε φόρμα αποτελεί ένα ξεχωριστό παράθυρο επικοινωνίας της εφαρμογής. Κατά το χρόνο εκτέλεσης όταν κλείσεις και την τελευταία φόρμα τερματίζεται η εκτέλεση της εφαρμογής. Για να αποφύγεις τη πολυπλοκότητα μιας εφαρμογής, μη χρησιμοποιείς πολλές φόρμες, εκτός αν είναι απαραίτητο.
- ⇒ Ένα γεγονός μπορεί να προκληθεί και από το σύστημα. Στο παράδειγμα του αεροπλάνου η προσγείωση πραγματοποιείται με τη χρήση ενός γεγονότος που προκαλείται από το ρολόι του συστήματος. Εκμεταλλεύσου τα συμβάντα του συστήματος ή τις δυνατότητες του κώδικα για την αυτοματοποίηση διαδικασιών.
- ⇒ Κάθε εξωτερικό μήνυμα είναι δυνατό να προκαλέσει περισσότερα του ενός γεγονότων στην εφαρμογή. Όταν ο χρήστης πατήσει ένα πλήκτρο από το πληκτρολόγιο καθώς το σημείο εισαγωγής βρίσκεται μέσα σε ένα πλαίσιο κειμένου τότε διαδοχικά προκαλούνται τα γεγονότα πάτημα πλήκτρου (keypress) και αλλαγή (change).
- ⇒ Ένα γεγονός είναι δυνατό να προκληθεί και μέσα από εντολές κώδικα. Για παράδειγμα σε ένα αντικείμενο πλήκτρο εντολής της εφαρμογής μπορούμε να συμπεριλάβουμε μια εντολή με την οποία όταν ο χρήστης πατήσει με το αριστερό πλήκτρο του ποντικιού (γεγονός Κλικ) το πλήκτρο να αλλάζει το περιεχόμενο ενός πλαισίου κειμένου. Για να αντικαταστήσουμε κατά την εκτέλεση της εφαρμογής, το περιεχόμενο του πλαισίου κειμένου πρέπει να αποδώσουμε καινούργια τιμή στην ιδιότητά του **κείμενο**. Η διαδικασία γεγονότος θα έχει τη μορφή :

ΔΙΑΔΙΚΑΣΙΑ Δεβρεδñi-Άiδiεβδl_êéê

Δεάβδέi-ÊâéiÝiïðl.Êâβiâiï = "ÊâéiÝñéi êâβiâiï"

ΤΕΛΟΣ ΔΙΑΔΙΚΑΣΙΑΣ

Μετά την εκτέλεση της παραπάνω εντολής η μετατροπή του περιεχομένου του πλαισίου κειμένου, θα προκαλέσει διαδοχικά το γεγονός Αλλαγή του πλαισίου κειμένου το οποίο θα οδηγήσει στην εκτέλεση της επόμενης διαδικασίας γεγονότος:

ΔΙΑΔΙΚΑΣΙΑ Δεάβόεί-ΈάείΎίί01_Άέέάά

...

Εντολές κώδικα

...

ΤΕΛΟΣ ΔΙΑΔΙΚΑΣΙΑΣ

- ⇒ Τα ειδικά αντικείμενα, δεν έχουν γραφικό ενδιαμέσο και συνήθως εκφράζουν εργαλεία του συστήματος. Ως ειδικά αντικείμενα στη Visual Basic μπορούμε να αναφέρουμε τον **εκτυπωτή** (Printer), το **πρόχειρο** (Clipboard), την **οθόνη** (Screen), την **εφαρμογή** (App) ή τα **αντικείμενα πρόσβασης δεδομένων** (Recordset).
- ⇒ Οι επιλογές ενός μενού είναι ανεξάρτητα αντικείμενα και μπορούμε να μετατρέψουμε τα χαρακτηριστικά τους με την αλλαγή των τιμών των ιδιοτήτων τους. Έτσι για μια επιλογή μενού μπορούμε να ορίσουμε ένα αρχείο βοήθειας ή να την ενεργοποιήσουμε, να την μαρκάρουμε και να την αποκρύψουμε.
- ⇒ Η εμφάνιση ενός αναδιπλωμένου μενού, γίνεται με την χρήση του δεξιού πλήκτρου του ποντικιού. Συνεπώς στο σημείο που θέλουμε να εμφανιστεί το αναδιπλωμένο μενού επιλογών πρέπει να ελέγχουμε αν ο χρήστης έχει πατήσει το δεξί πλήκτρο. Όταν η εφαρμογή ανιχνεύσει ότι ο χρήστης πάτησε και απελευθέρωσε το δεξί πλήκτρο πρέπει μέσα από την αντίστοιχη διαδικασία γεγονόςτος με την κατάλληλη μέθοδο να εμφανίσουμε το ανάλογο μενού. Στη Visual Basic η μέθοδος εμφάνισης αναδιπλωμένων μενού είναι η `PopupMenu`.
- ⇒ Ένα πλαίσιο διαλόγου μπορεί να εμφανιστεί ως **υποχρεωτικό** (modal) ή **μη υποχρεωτικό** (modeless). Τα υποχρεωτικά πλαίσια διαλόγου παραμένουν σαν ενεργά αντικείμενα της εφαρμογής μέχρι να τα κλείσει ο χρήστης. Με την τεχνική αυτή εξασφαλίζουμε ότι ο χρήστης θα αντιληφθεί και θα απκριθεί στο μήνυμα που περιέχουν. Τα μη υποχρεωτικά πλαίσια μπορούν να μεταφερθούν ανοικτά στο παρασκήνιο της εφαρμογής και χρησιμοποιούνται κυρίως για την εμφάνιση βοηθητικών μηνυμάτων. Τα μη υποχρεωτικά πλαίσια διαλόγου δεν χρησιμοποιούνται ποτέ για την εκτέλεση κρίσιμων εργασιών. Ο ορισμός ενός προκαθορισμένου πλαισίου διαλόγου ως υποχρεωτικό γίνεται με την αντιστοίχιση των σταθερών `vbApplicationModal` ή `vbSystemModal` στο όρισμα πλήκτρα εντολής της συνάρτησης `Msgbox`. Όταν οριστεί το πλαίσιο διαλόγου ως `SystemModal` όλες οι εφαρμογές του συστήματος απενεργοποιούνται μέχρι να απαντήσει ο χρήστης στο πλαίσιο διαλόγου, ενώ όταν οριστεί ως `ApplicationModal` απενεργοποιείται μόνο η τρέχουσα εφαρμογή.
- ⇒ Μια φόρμα της Visual Basic είναι δυνατόν να εμφανιστεί ως υποχρεωτικό πλαίσιο διαλόγου. Για να το επιτύχουμε πρέπει να συμπεριλάβουμε στην εντολή εμφάνισης της φόρμας το όρισμα 0 (μηδέν) :

```
MyForm.Show 0
```

Η εμφάνιση μιας φόρμας, ως υποχρεωτικό πλαίσιο διαλόγου δεν απενεργοποιεί τις υπόλοιπες εφαρμογές του συστήματος.

- ⇒ Με τη χρήση των ιδιοτήτων Height, και Width μπορούμε να δημιουργήσουμε ισομεγέθη αντικείμενα και με τη χρήση των ιδιοτήτων Left και Top μπορούμε να τα στοιχίσουμε.

11.4. Δραστηριότητες - ασκήσεις

Στην τάξη



ΔΤ1. Αναφέρατε αντικείμενα από το περιβάλλον σας.

ΔΤ2. Ποιες ιδιότητες και μεθόδους υποστηρίζουν τα αντικείμενα που περιγράψατε;

ΔΤ3. Περιγράψτε τα αντικείμενα που θα χρησιμοποιούσατε για το σχεδιασμό και την υλοποίηση του τρόπου επικοινωνίας:

- α) ενός συστήματος διαχείρισης της προσωπικής σου βιβλιοθήκης.
- β) μιας εφαρμογής ρολοι-ξυπνητήρι.

ΔΤ4. Ποια αντικείμενα και γεγονότα θα εκμεταλλευόσασταν από τα περιβάλλοντα που περιγράψατε στην προηγούμενη δραστηριότητα για την συγγραφή του κώδικα;

Στο εργαστήριο



Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σου:

ΔΕ1. Υλοποίησε σε γραφικό περιβάλλον το πρόγραμμα, που περιγράψαμε στο κεφάλαιο 7, για τη μετατροπή της θερμοκρασίας από τη κλίμακα Fahrenheit σε κλίμακα Celsius. Υπενθυμίζεται ότι η μετατροπή της θερμοκρασίας από βαθμούς Celsius σε Fahrenheit δίνεται από τον τύπο: $F = 9C/5 + 32$.

ΔΕ2. Στο παράδειγμα της βολής, που περιγράψαμε στο κεφάλαιο 7, υλοποίησε τη λύση της άσκησης σε γραφικό προγραμματιστικό περιβάλλον. Να επεκτείνετε το παράδειγμα με την προσθήκη και της γραφικής προσομοίωσης της βολής.

ΔΕ3. Να γράψεις πρόγραμμα με το οποίο ο χρήστης θα έχει τη δυνατότητα να επιλέγει και να εκτελεί από μια λίστα αρχείων μουσικής κάποιο μουσικό κομμάτι.



Στο σπίτι

Στο τετράδιο σας αντιμετωπίστε τα παρακάτω προβλήματα :

ΔΣ1. Περιέγραψε αντικείμενα από ένα χώρο που περνάς τον ελεύθερο χρόνο σου. Για κάθε αντικείμενο ανάφερε ιδιότητες και μεθόδους που υποστηρίζει. Ποια γεγονότα είναι δυνατόν να συμβούν σε αυτό το περιβάλλον, από εσάς και ποια από άλλες αιτίες;

ΔΣ2. Σχεδίασε ένα πρόγραμμα για την επίλυση τριωνύμου. Ο χρήστης θα πληκτρολογεί τους τρεις συντελεστές και μόλις πατήσει κάποιο πλήκτρο εντολής θα εμφανίζεται η λύση.

ΔΣ3. Δημιούργησε μια εφαρμογή για τη μετατροπή αριθμών σε διαφορετικά αριθμητικά συστήματα. Χρησιμοποίησε το οκταδικό, δεκαδικό και δεκαεξαδικό σύστημα.

ΔΣ4. Στο παράδειγμα του επεξεργαστή κειμένου (παράδειγμα 4) που έχουμε περιγράψει παραπάνω, πρόσθεσε ένα επιπλέον μενού επιλογών, με τις επιλογές του οποίου θα διαμορφώνεις το κείμενο του πλαισίου κειμένου (γραμματοσειρά, μέγεθος και χρώμα). Στη συνέχεια πρόσθεσε και μια γραμμή εργαλείων μέσω της οποίας θα εκτελούνται άμεσα οι βασικές εργασίες του μενού. Η γραμμή εργαλείων μπορεί να δημιουργηθεί με εργαλεία πλήκτρων εντολής ή εικόνας.

11.4. Τεστ αυτοαξιολόγησης



1. Δίνονται οι παρακάτω λέξεις. Ποιες από αυτές εκφράζουν αντικείμενα σύμφωνα με αυτά που αναφέρονται στο βιβλίο σου.

- | | |
|---------------|------------|
| A. αυτοκίνητο | Z. πράσινο |
| B. αγάπη | H. βιβλίο |
| Γ. λουλούδι | Θ. σελίδα |
| Δ. μεταφορά | I. ανάβαση |
| E. σκέψη | K. δίσκος |

Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος

- Ο δομημένος προγραμματισμός καταργείται με την εμφάνιση νέων τεχνικών προγραμματισμού.
- Τα γεγονότα προκαλούνται μετά από κάποια ενέργεια του χρήστη.
- Σε ένα σύγχρονο περιβάλλον εργασίας μπορούμε να επιτύχουμε τη συνεργασία διαφορετικών εφαρμογών.
- Σε μια εφαρμογή που έχει υλοποιηθεί με την τεχνική του οδηγούμενου από τα γεγονότα προγραμματισμού υπάρχουν μόνο διαδικασίες γεγονότων.

6. Μια μέθοδος δεν είναι συγκεκριμένη για ένα αντικείμενο και είναι δυνατό να εφαρμοστεί και σε κάποιο άλλο.
7. Δίνονται οι παρακάτω προτάσεις. Βάλτε τις προτάσεις σε σωστή σειρά ώστε να περιγράφουν τα βήματα ανάπτυξης μιας εφαρμογής σε σύγχρονο προγραμματιστικό περιβάλλον.
 - A. Δημιουργία και εκσφαλμάτωση του κώδικα.
 - B. Καθορισμός της αρχικής συμπεριφοράς των αντικειμένων μέσω των ιδιοτήτων που τα χαρακτηρίζουν.
 - Γ. Σχεδιασμός του τρόπου επικοινωνίας χρήστη-εφαρμογής, επιλέγοντας τα κατάλληλα αντικείμενα.

Διάλεξε όλα όσα χρειάζεται μεταξύ των προτεινόμενων

8. Ποια από τα παρακάτω είναι ιδιότητες:

A. χρώμα	Z. στοίχιση
B. αριθμός	H. ανανέωση
Γ. πλάτος	Θ. ενεργό
Δ. γραμματοσειρά	I. μεταβλητή
E. βαλβίδα	K. σκάλα
9. Τα παρακάτω χαρακτηριστικά συναντώνται μόνο σε ένα αντικειμενοστραφές περιβάλλοντος ανάπτυξης :

A. γραφικά	Δ. κληρονομικότητα
B. πολυμορφισμός	E. μενού επιλογών
Γ. κλάσεις	
10. Ένα γεγονός :
 - A. προκαλείται μόνο από το σύστημα
 - B. είναι δυνατό να προκαλέσει άλλα γεγονότα
 - Γ. μπορεί να μην επιδράσει σε κανένα αντικείμενο
 - Δ. είναι επακόλουθο της συνεργασίας διαφορετικών εφαρμογών
 - E. προκαλεί την εκτέλεση γενικών διαδικασιών κώδικα

Συμπλήρωσε τα κενά με το σωστή λέξη που λείπει

11. Ο σχεδιασμός του τρόπου επικοινωνίας χρήστη-εφαρμογής επιτυγχάνεται με την επιλογή των κατάλληλων _____

12. Μια _____ είναι η εφαρμογή μιας ενέργειας επάνω σε ένα αντικείμενο.
13. Τα χαρακτηριστικά ενός αντικειμένου καλούνται _____
14. Κάθε αντικείμενο κληρονομεί τις ιδιότητες και της μεθόδους από την _____
15. Κατά την συνεργασία διαφορετικών εφαρμογών με την _____ τα δεδομένα αποθηκεύονται στην δική μας εφαρμογή.

Συνδυάστε τα περιεχόμενα που καταγράφονται στις παρακάτω λίστες :

- | | |
|-----------------------|----------------------|
| 16. γεγονός | A. διαγραφή |
| 17. ιδιότητα | B. πάτημα πλήκτρου |
| 18. μέθοδος | Γ. λίστα |
| 19. αντικείμενο | Δ. πλήκτρο συντομίας |
| 20. αντικείμενο μενού | E. ορατό |