

**Τ.Ε.Ι. ΗΠΕΙΡΟΥ**

**T.E.I. OF EPIRUS**



ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ & ΟΙΚΟΝΟΜΙΑΣ (Σ.Δ.Ο)  
*ΤΜΗΜΑ ΤΗΛΕΠΛΗΡΟΦΟΡΙΚΗΣ & ΔΙΟΙΚΗΣΗΣ*

SCHOOL OF MANAGEMENT AND ECONOMICS  
*DEPARTMENT OF COMMUNICATIONS,  
INFORMATICS AND MANAGEMENT*

# Πτυχιακή Εργασία

Φοιτητές

**Μπίρμπο Μπλένταρ  
&  
Τζούμπα Πηνελόπη**

Επιβλέπων καθηγητής

**Κουρκουμέλης Νικόλαος**

ΘΕΜΑ

Online **Ενημέρωση Διδασκόμενων**  
**Εκπαιδευτικών Ιδρυμάτων**  
&  
**Συγχρονισμός δεδομένων**  
**με Ms Outlook και Mobile Devices**

Άρτα, Σεπτέμβριος 2005

## Περιεχόμενα

	σελίδα
<a href="#">Περιγραφή.....</a>	<a href="#">3</a>
<a href="#">Τεχνολογία .NET Framework.....</a>	<a href="#">4</a>
<a href="#">ProSync.....</a>	<a href="#">5</a>
<a href="#">    Δημιουργία προγράμματος.....</a>	<a href="#">6</a>
<a href="#">    Κώδικας.....</a>	<a href="#">7</a>
<a href="#">ProWeb.....</a>	<a href="#">26</a>
<a href="#">    Εμφάνιση του προγράμματος στο Outlook....</a>	<a href="#">27</a>
<a href="#">    Κώδικας.....</a>	<a href="#">30</a>

## ΠΕΡΙΓΡΑΦΗ

Οι κλασικοί τρόποι με τους οποίους οι φοιτητές και οι καθηγητές ενημερώνονται για το πρόγραμμα των μαθημάτων είναι , πρώτον - να πάνε στο ΤΕΙ και να αντιγράψουν το πρόγραμμα από τους πίνακες ανακοινώσεων, δεύτερων, να κατεβάσουν από το site του ΤΕΙ τα PDF αρχεία και πάλι να τα αντιγράψουν από εκεί. Μην ξεχνάμε ότι αυτό γινόταν κάθε φορά που άλλαζε το πρόγραμμα. Μετά, αν κάποιος ήθελε να τα έχει σε ηλεκτρονική μορφή έπρεπε να γράψει τα μαθήματα ένα ένα στο Excel, η στο Outlook κτλ.

Αυτή η πτυχιακή φτιάχτηκε για να διευκολύνει όσο γίνεται τους φοιτητές και τους καθηγητές πάνω σε αυτό το θέμα, και επίσης σιγά σιγά να χρησιμοποιήσουμε όλο και λιγότερο το μολύβι και το χαρτί.

Το ProSync και το ProWeb είναι δυο εφαρμογές που επιτρέπουν στους φοιτητές και στους καθηγητές να ενημερώνουν την ατζέντα του εβδομαδιαίου προγράμματος τους αυτόματα μέσω Internet. Το ProSync είναι η εφαρμογή που επιτρέπει στον υπεύθυνο του προγράμματος να το περάσει σε ηλεκτρονική μορφή, αναγνωρίσιμη από το ProWeb. Το τελευταίο αναλαμβάνει την απεικόνιση του προγράμματος ως ιστοσελίδα.

Τα εργαλεία που επιλέξαμε για την δημιουργία αυτών των δυο κομματιών της πτυχιακής είναι το Visual Studio 2005 Beta2. Για να μπορούν να τρέξουν αυτά τα προγράμματα πρέπει να είναι εγκατεστημένο το .NET Framework στον υπολογιστή. Για αυτή τη τεχνολογία θα μιλήσουμε παρακάτω.

## Τεχνολογία .NET Framework

Η τεχνολογία .net είναι το βασικό συστατικό της πλατφόρμας λογισμικού της Microsoft, για την ασφαλή ανάπτυξη, προμήθεια και χρήση διαλειτουργικών εφαρμογών, τόσο σε ανεξάρτητους υπολογιστές όσο και σε δίκτυα υπολογιστών ([Internet](#), [Intranet](#), κ.λπ.). Η τεχνολογία .net είναι στην πραγματικότητα μία δέσμη (σετ) λύσεων λογισμικού, στις οποίες κομβική θέση κατέχουν το **Microsoft .net Framework** και το **Microsoft Visual Studio .net**.

Χωρίς να υπεισέλθουμε σε τεχνικές λεπτομέρειες, το Microsoft .net Framework περιλαμβάνει τις λύσεις λογισμικού που θα επιτρέψουν την επικοινωνία χρήστη και εφαρμογών, ενώ το Microsoft Visual Studio .net. προσφέρει τα εργαλεία για τη δημιουργία αυτών των εφαρμογών. Υπάρχουν όμως και άλλα, ιδιαίτερα ενδιαφέροντα στοιχεία.

Το πιο ενδιαφέρον ίσως συστατικό της τεχνολογίας .net είναι τα Web services. Πρόκειται για μικρές εφαρμογές λογισμικού που βασίζονται στα πρωτόκολλα επικοινωνίας XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), και WSDL (Web Services Description Language) και οι οποίες επιτρέπουν την ανταλλαγή ή τη λήψη εφαρμογών (και δεδομένων) ανάμεσα σε υπολογιστές που τρέχουν διαφορετικά λειτουργικά συστήματα και χρησιμοποιούν διαφορετικές γλώσσες προγραμματισμού. Η ανταλλαγή δεδομένων μπορεί να πραγματοποιείται είτε μέσω του Διαδικτύου είτε μέσα σε κάποιο [Intranet](#) και μπορεί να πάρει 3 βασικές μορφές:

**α)** Η ανταλλαγή να γίνεται ανάμεσα σε υπολογιστές που παίζουν το ρόλο του "πελάτη" (client to client)

**β)** Η ανταλλαγή να γίνεται ανάμεσα στο διακομιστή και τους clients (server to client)

**γ)** Η ανταλλαγή να γίνεται μεταξύ διακομιστών (server to server)

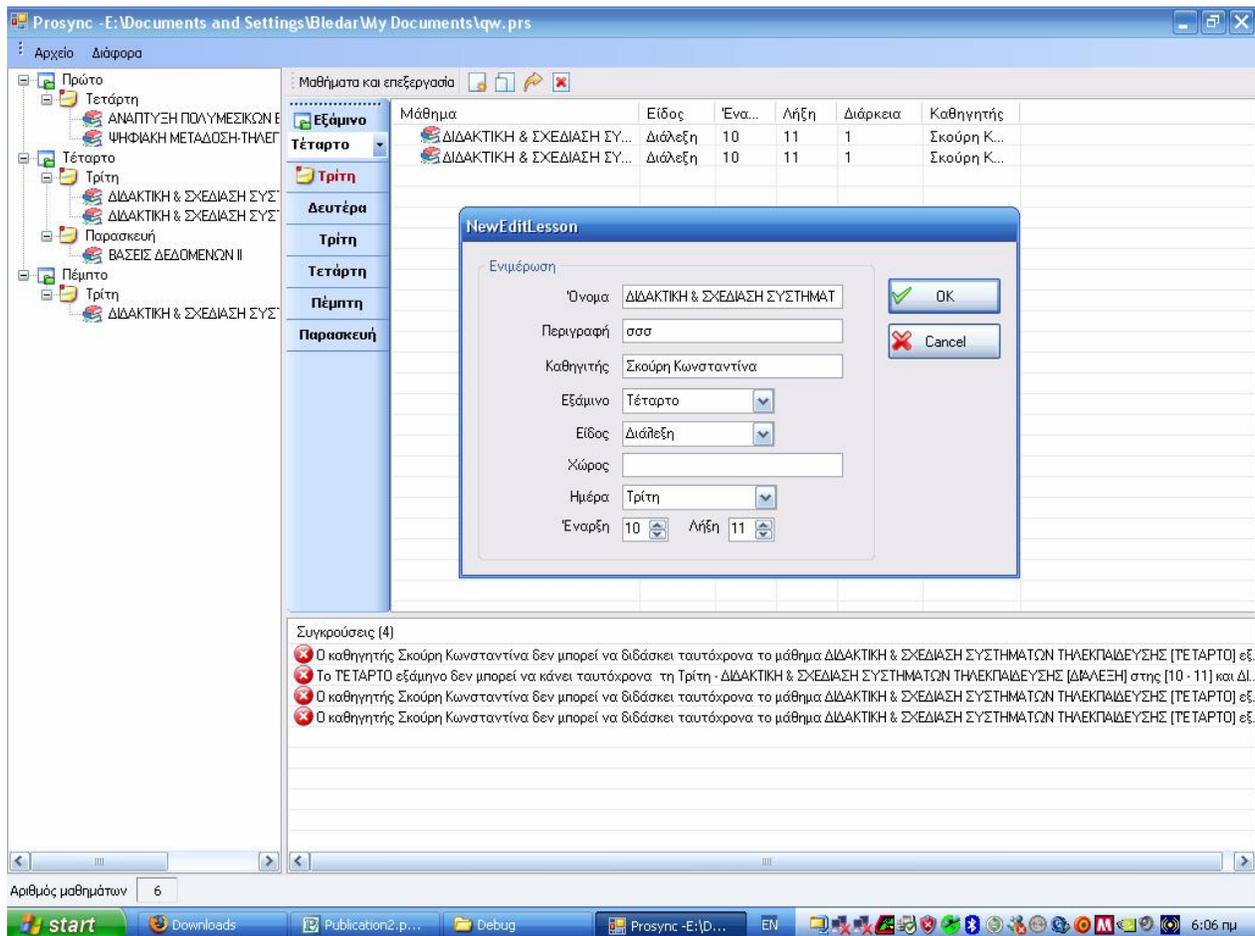
Έτσι, οι εν λόγω εφαρμογές μπορεί λ.χ. να βρίσκονται σε κάποιο διακομιστή, απ' όπου μπορούν να παραληφθούν, μέσω Internet, και να χρησιμοποιηθούν από κάθε ενδιαφερόμενο. Απλούστερα, τα Web services είναι εφαρμογές λογισμικού που αναπτύσσονται από κάποια εταιρία, "ανεβαίνουν" κάπου στο Διαδίκτυο (σ' ένα διακομιστή), και προσφέρονται για χρήση στις ενδιαφερόμενες επιχειρήσεις. Από την πλευρά τους, οι επιχειρήσεις/πελάτες καταβάλλουν ενδεχομένως κάποια συνδρομή για τη χρήση των εφαρμογών και αποκτούν το δικαίωμα χρήσης. Με το μοντέλο αυτό, το Internet γίνεται ο διάυλος και ταυτόχρονα η πηγή για την online άντληση εφαρμογών και τεχνολογικών λύσεων και όχι μόνο πληροφοριών.

Συμπερασματικά, η τεχνολογία .net είναι μία "πλατφόρμα διαλειτουργικότητας και ευελιξίας": Διαλειτουργικότητας, γιατί είναι η λογισμική βάση που επιτρέπει την ασφαλή επικοινωνία εφαρμογών και προγραμμάτων, μέσω του Internet ή άλλου είδους δικτύων. Ευελιξίας, γιατί δημιουργεί ένα ψηφιακό περιβάλλον που ενσωματώνει το Διαδίκτυο και τα χαρακτηριστικά του.

Σημειώνεται, ότι η τεχνολογία .net υποστηρίζει όλη την οικογένεια προϊόντων της Microsoft (λειτουργικά περιβάλλοντα, σουίτες γραφείου κ.λπ.), ενώ εκτιμάται ότι στο μέλλον θα διαδραματίσει ακόμη πιο ενεργό ρόλο στα προϊόντα της εταιρίας.

# PROSYNC

Η εφαρμογή αυτή βοηθά τον administrator να δημιουργεί ένα αντίγραφο του προγράμματος σε μια ηλεκτρονική μορφή αναγνωρίσιμη από το ProWeb. Υπάρχει δυνατότητα Save, Save As του προγράμματος ώστε να μπορεί να γίνει αρχειοθέτηση των προγραμμάτων.



Επίσης το ProSync είναι ένα ενεργό πρόγραμμα και μπορεί μέσω δυο αλγορίθμων να ανιχνεύει τυχόν συγκρούσεις μαθημάτων ή καθηγητών. Ο υπεύθυνος μπορεί να χρησιμοποιεί το ProSync ως κύριο εργαλείο για την δημιουργία του προγράμματος.

Η διεπαφή είναι σχεδιασμένη για να γίνουν όλα τα παραπάνω εύκολα και γρήγορα .

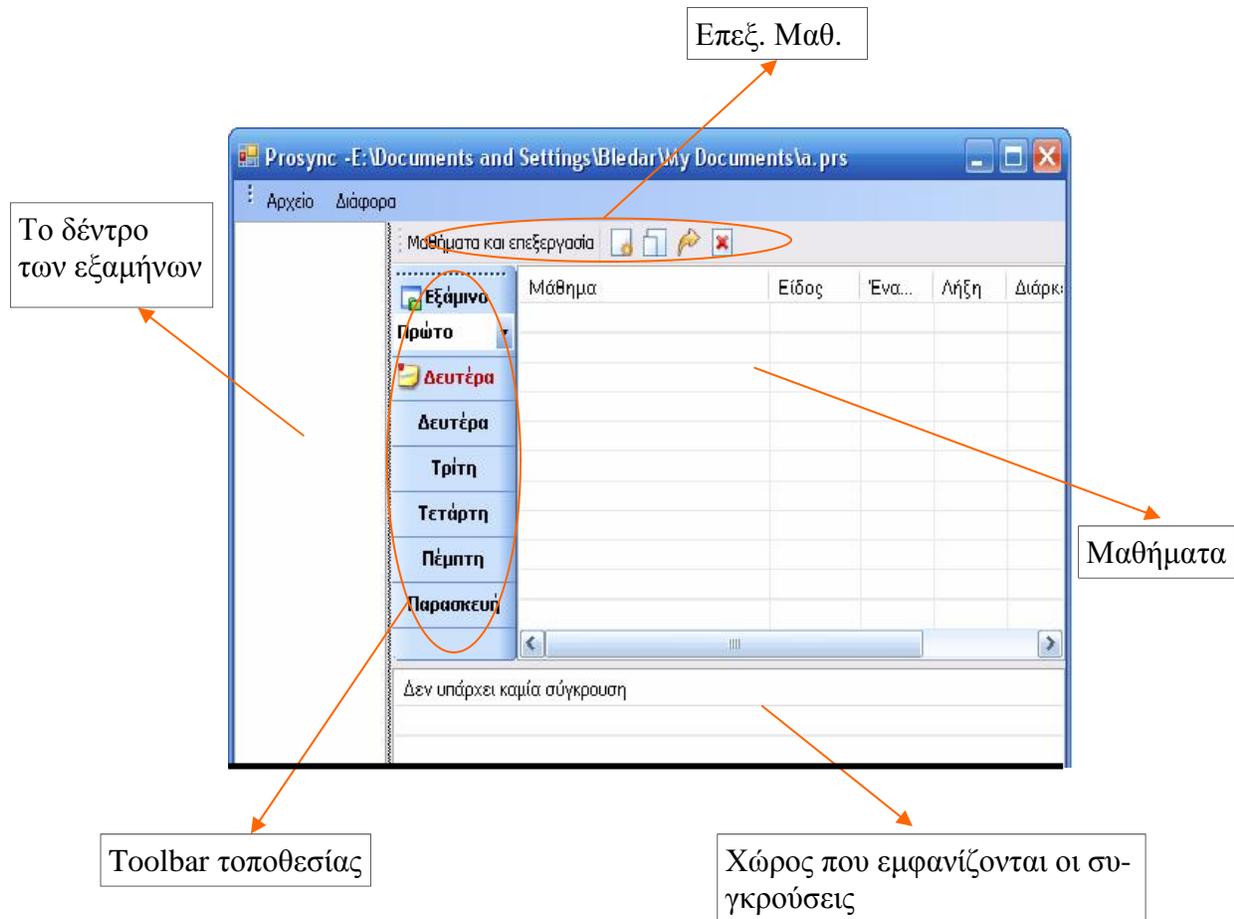
## Δημιουργία του προγράμματος

Από το μενού Αρχείο—> Νέο Πρόγραμμα.

Εισάγουμε το όνομα του προγράμματος.

Το ProSync δημιουργεί αυτόματα ένα αρχείο *όνομα.prs* και το ανοίγει για επεξεργασία.

Πάνω στο παράθυρο φαίνεται το αρχείο που είναι ανοικτό.



Για να εισάγουμε ένα καινούργιο μάθημα, επιλεγούμε πρώτα το εξάμηνο και την ημέρα στο Toolbar τοποθεσίας, μετά από την επεξεργασία μαθημάτων επιλέγουμε «Νέο Μάθημα». Εισάγουμε όσα μαθήματα χρειάζονται. Για οτιδήποτε αλλαγή ή λάθος, κάνοντας διπλό-κλικ στο μάθημα (όχι πάνω στο δέντρο) εμφανίζεται το παράθυρο για την επεξεργασία των μαθημάτων. Οι συγκρούσεις εμφανίζονται κατά την διάρκεια τις επεξεργασίας των μαθημάτων και εξαφανίζονται αυτόματα όταν όλα είναι εντάξει. Αφού τελειώσουν τα μαθήματα και θέλουμε το πρόγραμμα που μόλις φτιάξαμε να φαίνεται στο Site, τότε επιλέγουμε Αρχείο —> Export Program, εισάγουμε τον αριθμό των εβδομάδων που ισχύει και το ProSync μας λέει που υποθήκευσε το "Exported.CSV". Το Αρχείο αυτό το αντιγραφούμε (αντικαθιστώντας το υπάρχον) στον κατάλογο Root του site.

Το Export πρέπει να γίνει κάθε φορά που αλλάζει το πρόγραμμα.

## Κώδικας του PROSYNC

Το ProSync είναι γραμμένο με ένα από τα πιο σύγχρονα εργαλεία προγραμματισμού, με Visual Studio 2005 (beta 2). Η γλώσσα που χρησιμοποιήθηκε είναι η Visual Basic. Είναι γλώσσα τέταρτης γενιάς και έχει πάρα πολλές δυνατότητες. Το πρόγραμμα τρέχει πάνω από το DotNet Framework Version 2. Παρακάτω είναι ο κώδικας από τον οποίο αποτελείται το ProSync και μερικά σχόλια για καλύτερη κατανόηση του.

*Η κλάση αυτή έχει τις ιδιότητες που πρέπει να έχει ένα μάθημα. Χρησιμοποιείτε αρκετές φορές στον κώδικα και στο WebSite. Το <Serializable()> μας επιτρέπει να την αποθηκεύουμε στο δισκο.*

```
<Serializable()> Public Class GRecord
    Public Eksamino As String
    Public Onoma As String
    Public Perigrافي As String
    Public Kathigitis As String
    Public Ethusa As String
    Public Mera As String
    Public StartTime As Short
    Public StopTime As Short
    Public Idos As String
End Class
```

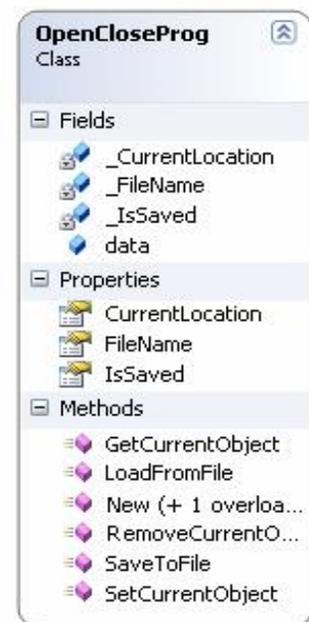


*Είναι μια πολύ σημαντική κλάση η οποία ασχολείται με την αποθήκευση, το άνοιγμα, και τις ιδιότητες των μαθημάτων οι οποίες παριστάνονται από το data, τύπου ArrayList. Το CurrentLocation είναι μια μεταβλητή που κρατά τον δείκτη για να διαβάζουμε ή να γράψουμε στο ArrayList.*

```
Public Class OpenCloseProg

    Public data As ArrayList
    Private _IsSaved As Boolean
    Private _FileName As String
    Private _CurrentLocation As Int16

    Public Property CurrentLocation() As Int16
        Get
            Return _CurrentLocation
        End Get
        Set(ByVal value As Int16)
            If value >= 0 And value < data.Count Then
                _CurrentLocation = value
            Else
                MsgBox("Could not set value for CurrentLocation")
            End If
        End Set
    End Property
End Class
```



```
End Set
End Property

Public Property FileName() As String
Get
    Return _FileName
End Get
Set(ByVal value As String)
    _FileName = value
End Set
End Property

Public Property IsSaved() As Boolean
Get
    Return _IsSaved
End Get
Set(ByVal value As Boolean)
    _IsSaved = value
End Set
End Property

Public Function GetCurrentObject() As GRecord
    Dim tmp As GRecord = CType(data.Item(_CurrentLocation), GRecord)
    Return tmp
End Function

Public Sub SetCurrentObject(ByVal newObject As GRecord)
    'MsgBox("removing at:" + _CurrentLocation.ToString, MsgBoxStyle.Information)
    RemoveCurrentObject()
    data.Insert(_CurrentLocation, newObject)
End Sub

Public Sub RemoveCurrentObject()
    data.RemoveAt(_CurrentLocation)
End Sub

Sub New()
    'data = New ArrayList
End Sub

Sub New(ByVal Vfilename As String, Optional ByVal CreateNew As Boolean = False)
    _FileName = Vfilename
    If CreateNew Then
        data = New ArrayList
        SaveToFile()
    Else
```

```
LoadFromFile()
End If
End Sub

Public Function LoadFromFile() As Boolean
    If File.Exists(_FileName) Then
        Try
            Dim FS As New System.IO.FileStream(_FileName, IO.FileMode.Open)
            Dim BinFormatter As New Binary.BinaryFormatter()
            data = CType(BinFormatter.Deserialize(FS), ArrayList)
            _CurrentLocation = data.Count - 1
            _IsSaved = True
            FS.Close()
        Catch ex As Exception
            MsgBox(ex.Message)
            Return False
        End Try
    End If
    Return True
End Function

Public Function SaveToFile() As Boolean
    Try
        Dim FS As New System.IO.FileStream(_FileName, IO.FileMode.Create)
        Dim BinFormatter As New Binary.BinaryFormatter()
        BinFormatter.Serialize(FS, data)
        FS.Close()
        _IsSaved = True
    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    End Try
    Return True
End Function
End Class
```

*Μόνο με τέσσερις γραμμές μπορούμε να διαβάσουμε ή να γράψουμε όλες τις κλάσης *Grecord* που είναι αποθηκευμένες στο *data* (οι γραμμές με έντονο χρώμα)*

Ακολουθεί ο κώδικας της κύριας φόρμας του προγράμματος. *MainFrm*  
Υπάρχουν εξηγήσεις στις γραμμές με έντονο χρώμα.

**Imports System.IO**

**Imports System.Runtime.Serialization.Formatters**

**Imports System.Collections.Specialized**

**Imports System.Windows.Forms**

*Ορισμός των βασικών κλάσεων που θα χρησιμοποιηθούν.*

**Public Class** MainFrm

**Dim** autoCompleteStringCollection1 **As New** AutoCompleteStringCollection()

**Private Enum** StateModes

NothingOpened = 0

Opened = 1

**End Enum**

**Private Enum** PosModes

PositionEksamino = 0

PositionMera = 1

**End Enum**

**Private Structure** LesLocation

**Public** Eksamino **As String**

**Public** Mera **As String**

**End Structure**

**Private** Position **As** LesLocation

**Private** dataList **As** OpenCloseProg

**Private** tmpSelected **As** ArrayList

**Private** mMonth() **As String** = {"Πρώτο", "Δεύτερο", "Τρίτο", "Τέταρτο", "Πέμπτο", "Έκτο", "Εβδομο" }

**Private** mDays() **As String** = {"Δευτέρα", "Τρίτη", "Τετάρτη", "Πέμπτη", "Παρασκευή" }

**Private** ImpactList **As** StringCollection

*Η συνάρτηση αυτή ελέγχει τη λίστα μαθημάτων για τυχόν μαθήματα που έχουν πρόβλημα, και ενημερώνει αυτόματα τη λίστα με τις συγκρούσεις*

**Private Function** CollisionDetection() **As Boolean**

**If** dataList.data.Count < 2 **Then** **Exit Function**

ErrorList.Items.Clear()

**Dim** i, j **As Integer**

**For** i = 0 **To** dataList.data.Count - 2

```

For j = i + 1 To dataList.data.Count - 1
  Dim ob1 As GRecord = CType(dataList.data(i), GRecord)
  Dim ob2 As GRecord = CType(dataList.data(j), GRecord)
  'elenxos gia tixon sugrousis kathigiton
  If ob1.Kathigitis = ob2.Kathigitis Then
    If ob1.Mera = ob2.Mera Then
      If (ob2.StartTime >= ob1.StartTime And ob2.StartTime < ob1.StopTime) _
        Or _
        (ob1.StartTime >= ob2.StartTime And ob1.StartTime < ob2.StopTime) Then
        Dim tmpSTR = "Ο καθηγητής " + ob1.Kathigitis + " δεν μπορεί να διδάσκει
ταυτόχρονα το μάθημα " _
          + ob1.Onoma.ToUpper + " [" + ob1.Eksamino.ToUpper + "] εξάμηνο
[" + ob1.StartTime.ToString + " - " + ob1.StopTime.ToString + "] [" + ob1.Mera + "]" _
          + " και το μάθημα " + ob2.Onoma.ToUpper + " [" +
ob2.Eksamino.ToUpper + "] εξάμηνο [" + ob2.StartTime.ToString + " - " +
ob2.StopTime.ToString + "] [" + ob2.Mera + "]"
        Dim tmpItem As New ListViewItem
        tmpItem.Text = tmpSTR
        tmpItem.ImageIndex = 3
        ErrorList.Items.Add(tmpItem)
      End If
    End If
  End If
  'elenxo gia tixon sigrouseis mathimaton toy idiou eksaminou
  If ob1.Eksamino = ob2.Eksamino Then
    If ob1.Mera = ob2.Mera Then
      If ob1.Idos = "Διάλεξη" Or ob2.Idos = "Διάλεξη" Then
        If (ob2.StartTime >= ob1.StartTime And ob2.StartTime < ob1.StopTime) _
          Or _
          (ob1.StartTime >= ob2.StartTime And ob1.StartTime < ob2.StopTime) Then
          Dim tmpSTR = "Το " + ob1.Eksamino.ToUpper + " εξάμηνο δεν μπορεί να
κάνει ταυτόχρονα τη " + ob1.Mera + " - " _
            + ob1.Onoma.ToUpper + " [" + ob1.Idos.ToUpper + "] στις [" +
ob1.StartTime.ToString + " - " + ob1.StopTime.ToString _
            + "] και " + ob2.Onoma.ToUpper + " [" + ob2.Idos.ToUpper +
            "] στις [" + ob2.StartTime.ToString + " - " + ob2.StopTime.ToString + "]"

          Dim tmpItem As New ListViewItem
          tmpItem.Text = tmpSTR
          tmpItem.ImageIndex = 3
          ErrorList.Items.Add(tmpItem)
        End If
      End If
    End If
  End If
Next

```

```

Next
If Me.ErrorList.Items.Count > 0 Then
    Me.ErrorList.Columns(0).Text = "Συγκρούσεις (" + Me.ErrorList.Items.Count.ToString
+ ")"
Else
    Me.ErrorList.Columns(0).Text = "Δεν υπάρχει καμία σύγκρουση"
End If
End Function

```

*Το PopulateTree διαβάζει την λίστα μαθημάτων και φτιάχνει ένα δέντρο σύμφωνα με το εξάμηνο και την ημέρα.*

```

Private Function PopulateTree() As Boolean
    MainTree.Nodes.Clear()
    Dim i, j, ind As Integer
    Dim foundDAY As Boolean = False
    Dim foundMonth As Boolean = False
    Dim mNode As TreeNode
    Dim dNode As TreeNode
    For i = 0 To mMonth.Length - 1
        foundMonth = False
        For j = 0 To mDays.Length - 1
            foundDAY = False
            For ind = 0 To dataList.data.Count - 1
                Dim tmp As GRecord
                tmp = CType(dataList.data(ind), GRecord)
                If tmp.Eksamino = mMonth(i) And tmp.Mera = mDays(j) Then
                    If foundMonth = False Then
                        MainTree.SelectedNode = Nothing
                        mNode = New TreeNode(mMonth(i), 0, 0)
                        MainTree.Nodes.Add(mNode)
                        'MainTree.SelectedNode = mNode
                        foundMonth = True
                    End If
                    If foundDAY = False Then
                        MainTree.SelectedNode = mNode
                        'Dim newNode As TreeNode
                        dNode = New TreeNode(mDays(j), 1, 1)
                        MainTree.SelectedNode.Nodes.Add(dNode)
                        MainTree.SelectedNode = dNode
                        Dim lNode As TreeNode
                        lNode = New TreeNode(tmp.Onoma, 2, 2)
                        MainTree.SelectedNode.Nodes.Add(lNode)
                        foundDAY = True
                    Else
                        MainTree.SelectedNode = dNode
                    End If
                End If
            End For
        End For
    End For
End Function

```

```

    Dim lnode As TreeNode
    lnode = New TreeNode(tmp.Onoma, 2, 2)
    MainTree.SelectedNode.Nodes.Add(lnode)
    foundDAY = True
  End If
End If
tmp = Nothing
Next
Next
MainTree.ExpandAll()
End Function

```

```

Private Function FillLessonGridRow(ByVal tmpRec As GRecord, ByVal index As Short) As Boolean

```

```

  If tmpRec Is Nothing Then
    Return False
  End If
  Me.LessonList.Items.Add(tmpRec.Onoma)
  Me.LessonList.Items(index).SubItems.Add(tmpRec.Idos)
  Me.LessonList.Items(index).SubItems.Add(tmpRec.StartTime.ToString)
  Me.LessonList.Items(index).SubItems.Add(tmpRec.StopTime.ToString)
  Me.LessonList.Items(index).SubItems.Add((tmpRec.StopTime -
tmpRec.StartTime).ToString)
  Me.LessonList.Items(index).SubItems.Add(tmpRec.Kathigitis)
  Me.LessonList.Items(index).ImageIndex = 2
End Function

```

*Η συνάρτηση αυτή εμφανίζει στον χώρο των μαθημάτων όλα εκείνα τα μαθήματα που έχουν εξάμηνο και ημέρα σύμφωνα με την επιλογή στο Toolbar τοποθεσίας.*

```

Private Sub drawLessons()
  tmpSelected = New ArrayList
  Me.LessonList.Items.Clear()
  Me.MainTree.Nodes.Clear()
  If dataList Is Nothing Then Exit Sub
  If dataList.data.Count < 1 Then
    Exit Sub
  End If
  Dim i As Int16
  statusLessons.Text = dataList.data.Count.ToString
  PopulateTree()
  CollisionDetection()
  For i = 0 To dataList.data.Count - 1
    'MsgBox(dataList.GetCurrentObject.Mera)
  Next i
End Sub

```

```
If dataList.data(i).Mera = Position.Mera And dataList.data(i).Eksamino = Position.Eksamino Then
    tmpSelected.Add(i)
    dataList.CurrentLocation = i
    Dim tmp As GRecord = dataList.GetCurrentObject
    FillLessonGridRow(tmp, tmpSelected.Count - 1)
End If
Next
' set current selected = array 0
'If tmpSelected.Count > 0 Then
' i = CType(tmpSelected.Item(0), Int16)
' dataList.CurrentLocation = i
' Me.LessonList.Items(0).Selected = True
' End If
End Sub
```

```
Private Sub newLesson()
    Dim tmpBox As New NewEditLesson
    EditLesson = New GRecord
    EditLesson.Eksamino = Position.Eksamino
    EditLesson.Mera = Position.Mera
    Dim res = tmpBox.ShowDialog
    ' If res = Windows.Forms.DialogResult.Cancel Then Exit Sub
    If EditLesson Is Nothing Then Exit Sub
    dataList.data.Add(EditLesson)
    Unsaved()
    drawLessons()
End Sub
```

```
Private Function PositionTo(ByVal strToPos As String, ByVal Mode As PosModes) As Boolean
```

```
    If Mode = PosModes.PositionEksamino Then
        Position.Eksamino = strToPos
        EksaminaDrop.Text = strToPos
    Else
        Position.Mera = strToPos
        ToolDAY.Text = Position.Mera
    End If
```

```
    drawLessons()
    Return True
End Function
```

```
Private Function enableDisableAll(ByVal state As Boolean) As Boolean
    Me.MainTree.Enabled = state
    Me.LessonList.Enabled = state
    Me.DaysToolbar.Enabled = state
```

```

Me.ErrorList.Enabled = state
If state = False Then Me.Text = "ProSync"
End Function

```

*Σύμφωνα με την παράμετρο State η συνάρτηση αυτή ενεργοποιεί ή απενεργοποιεί τα controls της φόρμας. Καλείται όταν ανοίγουμε ή δημιουργούμε ένα καινούργιο πρόγραμμα.*

```

Private Function SetStateMode(ByVal state As StateModes) As Boolean

```

```

If state = StateModes.NothingOpened Then

```

```

    enableDisableAll(False)

```

```

    Me.MnuSaveProg.Enabled = False

```

```

    Me.MnuSaveAsProg.Enabled = False

```

```

    Me.LTool.Enabled = False

```

```

    statusLessons.Text = ""

```

```

    Me.ErrorList.Items.Clear()

```

```

    Me.LessonList.Items.Clear()

```

```

    Me.MainTree.Nodes.Clear()

```

```

    Exit Function

```

```

End If

```

```

If state = StateModes.Opened Then

```

```

    enableDisableAll(True)

```

```

    Me.MnuSaveAsProg.Enabled = True

```

```

    Me.LTool.Enabled = True

```

```

    Exit Function

```

```

End If

```

```

End Function

```

```

Private Function CloseProgram() As Boolean

```

```

If dataList Is Nothing Then Return True

```

```

If dataList.IsSaved = False Then

```

```

    Dim resp As MsgBoxResult = MsgBox("Το πρόγραμμα έχει αλλάξει. Θέλετε να αποθηκευτούν οι αλλαγές;", MsgBoxStyle.YesNoCancel, "ProSync")

```

```

If resp = MsgBoxResult.Yes Then

```

```

    dataList.SaveToFile()

```

```

    dataList = Nothing

```

```

    MainTree.Nodes.Clear()

```

```

Else

```

```

    If resp = MsgBoxResult.No Then

```

```

        dataList = Nothing

```

```

    Else

```

```

        Return False

```

```

    End If

```

```

End If

```

```

End Function

```

```

SetStateMode(StateModes.NothingOpened)

```

```
Me.Text = "Prosync"  
Return True  
End Function
```

```
Private Function NewProgram() As Boolean  
Dim resp As Windows.Forms.DialogResult = SaveFile.ShowDialog  
Dim tmpfilename As String  
tmpfilename = SaveFile.FileName  
If resp = Windows.Forms.DialogResult.OK And tmpfilename.Trim <> "" Then  
If Not CloseProgram() Then Return False  
dataList = Nothing  
dataList = New OpenCloseProg(tmpfilename, True)  
'dataList.IsSaved = False  
Me.Text = "Prosync -" + tmpfilename  
SetStateMode(StateModes.Opened)  
drawLessons()  
Return True  
End If  
Return False  
'  
End Function
```

```
Private Function OpenProgram() As Boolean  
Dim resp As Windows.Forms.DialogResult = OpenFileDialog.ShowDialog  
Dim tmpfilename As String  
tmpfilename = OpenFileDialog.FileName  
If resp = Windows.Forms.DialogResult.OK And tmpfilename.Trim <> "" Then  
If CloseProgram() = False Then Return False  
dataList = New OpenCloseProg(tmpfilename)  
Me.Text = "Prosync -" + tmpfilename  
SetStateMode(StateModes.Opened)  
drawLessons()  
Return True  
End If  
Return True  
'  
End Function
```

```
Private Sub MainFrm_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load  
SetStateMode(StateModes.NothingOpened)  
Me.EksaminaDrop.SelectedIndex = 0  
PositionTo("Πρώτο", PosModes.PositionEksamino)  
PositionTo("Δευτέρα", PosModes.PositionMera)  
  
AutoCompOBJ.Mathimata = ReadAutoFiles(Application.StartupPath + "\automath.txt")
```

```

AutoCompOBJ.Kathigites = ReadAutoFiles(Application.StartupPath + "\autokath.txt")
'AutoCompOBJ.Xoros = ReadAutoFiles("autoxoro.txt")
End Sub

```

*Διαβάζει τα αρχεία που περιέχουν σε απλό κείμενο την λίστα με τα ονόματα των καθηγητών ή μαθημάτων για autocomplete.*

```

Private Function ReadAutoFiles(ByVal fn As String) As AutoCompleteStringCollection
    Dim tmpc As New AutoCompleteStringCollection
    Try
        Dim sr As StreamReader = New StreamReader(fn)
        Dim line As String
        Do
            line = sr.ReadLine()
            If Not line Is Nothing Then
                If line.Trim <> "" Then
                    tmpc.Add(line)
                End If
            End If
        Loop Until line Is Nothing
        sr.Close()
    Catch E As Exception
        MsgBox(E.Message + " FROM AUTOCOMplete")
        Return Nothing
    End Try
    Return tmpc
End Function

```

```

Public Sub Unsaved()
    If dataList Is Nothing Then Exit Sub
    dataList.IsSaved = False
    Me.MnuSaveProg.Enabled = Not dataList.IsSaved
End Sub

```

**#Region "ToolBarDays"**

```

Private Sub MondayToolStripButton_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MondayToolStripButton.Click
    PositionTo("Δευτέρα", PosModes.PositionMera)
End Sub

```

```

Private Sub TuesdayToolStripButton_Click(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles TuesdayToolStripButton.Click
    PositionTo("Τρίτη", PosModes.PositionMera)
End Sub

```

```

Private Sub WednesdayToolStripButton_Click(ByVal sender As System.Object, ByVal e As Sys-

```

```
tem.EventArgs) Handles WendsToolStripButton.Click
    PositionTo("Τετάρτη", PosModes.PositionMera)
End Sub
```

```
Private Sub ThurstdayToolStripButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ThurstdayToolStripButton.Click
    PositionTo("Πέμπτη", PosModes.PositionMera)
End Sub
```

```
Private Sub ΠαρασκευήToolStripButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ΠαρασκευήToolStripButton.Click
    PositionTo("Παρασκευή", PosModes.PositionMera)
End Sub
#End Region
```

```
'Private Sub AboutToolStripMenuItem_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs)
' Dim abt As New AboutBox1
' abt.ShowDialog()
' End Sub
```

```
Private Sub mnuDELL_Click_1(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles mnuDELL.Click
    Dim listIndexes As ListView.SelectedIndexCollection
    listIndexes = Me.LessonList.SelectedIndices
    If listIndexes.Count = 1 Then
        Dim ListIND As Integer = listIndexes.Item(0)
        'Dim tmpBox As New NewEditLesson
        dataList.CurrentLocation = CType(tmpSelected(ListIND), Int16)
        ' EditLesson = dataList.GetCurrentObject
        'Dim res = tmpBox.ShowDialog
        ' If res = Windows.Forms.DialogResult.Cancel Then Exit Sub
        'If EditLesson Is Nothing Then Exit Sub
        dataList.RemoveCurrentObject()
        drawLessons()
        Unsaved()
    Else
        MsgBox("Παρακαλούμε επιλέξτε ένα μάθημα πρώτα !", MsgBoxStyle.Exclamation,
"Prosync")
    End If
End Sub
```

```
Private Sub mnuCOPYL_Click_1(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles mnuCOPYL.Click
    Dim listIndexes As ListView.SelectedIndexCollection
    listIndexes = Me.LessonList.SelectedIndices
```

```
If listIndexes.Count = 1 Then
    Dim ListIND As Integer = listIndexes.Item(0)
    dataList.CurrentLocation = CType(tmpSelected(ListIND), Int16)
    Dim tmp As New GRecord
    tmp.Onoma = dataList.GetCurrentObject().Onoma
    tmp.Kathigitis = dataList.GetCurrentObject().Kathigitis
    tmp.Ethusa = dataList.GetCurrentObject().Ethusa
    tmp.Eksamino = dataList.GetCurrentObject().Eksamino
    tmp.Mera = dataList.GetCurrentObject().Mera
    tmp.Perigrafi = dataList.GetCurrentObject().Perigrafi
    tmp.Idos = dataList.GetCurrentObject().Idos
    tmp.StartTime = dataList.GetCurrentObject().StartTime
    tmp.StopTime = dataList.GetCurrentObject().StopTime
    dataList.data.Add(tmp)
    Unsaved()
    drawLessons()
Else
    MsgBox("Παρακαλούμε επιλέξτε ένα μάθημα πρώτα !", MsgBoxStyle.Exclamation,
"Prosync")
End If
End Sub

Private Sub MnuNewProg_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuNewProg.Click
    NewProgram()
End Sub

Private Sub MnuExitProg_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuExitProg.Click
    If CloseProgram() = True Then
        End
    End If
End Sub

Private Sub MnuSaveProg_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuSaveProg.Click
    dataList.SaveToFile()
    Me.MnuSaveProg.Enabled = Not dataList.IsSaved
End Sub

Private Sub mnuNEWL_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuNEWL.Click
    newLesson()
End Sub

Private Sub EksaminaDrop_SelectedIndexChanged1(ByVal sender As Object, ByVal e As
```

```
System.EventArgs) Handles EksaminaDrop.SelectedIndexChanged  
    PositionTo(Me.EksaminaDrop.Text, PosModes.PositionEksamino)  
End Sub
```

```
Private Sub MnuOpenProg_Click_1(ByVal sender As System.Object, ByVal e As Sys-  
tem.EventArgs) Handles MnuOpenProg.Click  
    OpenProgram()  
End Sub
```

```
Private Sub expandall_Click(ByVal sender As System.Object, ByVal e As Sys-  
tem.EventArgs) Handles expandall.Click  
    MainTree.ExpandAll()  
End Sub
```

```
Private Sub collapseall_Click(ByVal sender As System.Object, ByVal e As Sys-  
tem.EventArgs) Handles collapseall.Click  
    MainTree.CollapseAll()  
End Sub
```

```
Private Sub mnuMOVEL_Click(ByVal sender As System.Object, ByVal e As Sys-  
tem.EventArgs) Handles mnuMOVEL.Click  
    Dim listIndexes As ListView.SelectedIndexCollection  
    listIndexes = Me.LessonList.SelectedIndices  
    If listIndexes.Count > 0 Then  
        Dim ListIND As Integer = listIndexes.Item(0)  
        Dim tmpBox As New NewEditLesson  
        dataList.CurrentLocation = CType(tmpSelected(ListIND), Int16)  
        EditLesson = dataList.GetCurrentObject  
        Dim res As New frmMove  
        res.ShowDialog()  
        'If res = Windows.Forms.DialogResult.Cancel Then Exit Sub  
        If EditLesson Is Nothing Then Exit Sub  
        dataList.SetCurrentObject(EditLesson)  
        Unsaved()  
        drawLessons()  
    Else  
        MsgBox("Παρακαλούμε επιλέξτε ένα μάθημα πρώτα !", MsgBoxStyle.Exclamation,  
"Prosync")  
    End If  
End Sub
```

```
Private Sub LessonList_DoubleClick(ByVal sender As Object, ByVal e As Sys-  
tem.EventArgs) Handles LessonList.DoubleClick  
    Dim listIndexes As ListView.SelectedIndexCollection  
    listIndexes = Me.LessonList.SelectedIndices  
    If listIndexes.Count > 0 Then
```

```

Dim ListIND As Integer = listIndexes.Item(0)
Dim tmpBox As New NewEditLesson
dataList.CurrentLocation = CType(tmpSelected(ListIND), Int16)
EditLesson = Nothing
EditLesson = dataList.GetCurrentObject
Dim res = tmpBox.ShowDialog
' If res = Windows.Forms.DialogResult.Cancel Then Exit Sub
If EditLesson Is Nothing Then Exit Sub
dataList.SetCurrentObject(EditLesson)
Unsaved()
drawLessons()
End If
End Sub

```

```

Private Sub MainTree_NodeMouseDoubleClick(ByVal sender As Object, ByVal e As System.Windows.Forms.TreeNodeMouseClickEventArgs) Handles MainTree.NodeMouseDoubleClick
If e.Node.Level = 1 Then
PositionTo(e.Node.Text, PosModes.PositionMera)
PositionTo(e.Node.Parent.Text, PosModes.PositionEksamino)
Else
If e.Node.Level = 0 Then
PositionTo(e.Node.Text, PosModes.PositionEksamino)
End If
End If
End Sub

```

```

Private Sub MnuSaveAsProg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuSaveAsProg.Click
Dim resp As Windows.Forms.DialogResult = SaveFile.ShowDialog
Dim tmpfilename As String
tmpfilename = SaveFile.FileName
If resp = Windows.Forms.DialogResult.OK And tmpfilename.Trim <> "" Then
'ataList = Nothing
'dataList = New OpenCloseProg(tmpfilename, True)
'dataList.IsSaved = False
dataList.FileName = tmpfilename
Me.Text = "Prosync -" + tmpfilename
dataList.SaveToFile()
Me.MnuSaveProg.Enabled = Not dataList.IsSaved
'SetStateMode(StateModes.Opened)
'drawLessons()
'Return True
End If
'Return False
End Sub

```

```

Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AboutToolStripMenuItem.Click
    Dim ab As AboutBox1
    ab = New AboutBox1
    ab.ShowDialog()
End Sub

```

*Η συνάρτηση αυτή μετατρέπει την λίστα με τα μαθήματα σε Comma Separated Values και την αποθήκευση με το όνομα «Export.CSV». Η πρώτη γραμμή του αρχείου περιέχει τον αριθμό των βδομάδων που θα ισχύει το πρόγραμμα.*

```

Private Sub ExportProgramToolStripMenuItem_Click(ByVal sender As System.Object, By-
Val e As System.EventArgs) Handles ExportProgramToolStripMenuItem.Click
    If dataList.data.Count > 0 Then
        Dim fsE As New StreamWriter(Application.StartupPath + "\Exported.CSV", False)
        Dim weeks As String
        weeks = InputBox("Για πόσες εβδομάδες θα ισχύει το πρόγραμμα αυτό ;", "ProWEB",
"1")
        fsE.WriteLine(weeks)
        Dim i
        For i = 0 To dataList.data.Count - 1
            MsgBox(dataList.GetCurrentObject.Mera)
            dataList.CurrentLocation = i
            Dim tmp As GRecord = dataList.GetCurrentObject
            Dim StrToWrite As String

            StrToWrite = tmp.Eksamino + "#" + _
                tmp.Kathigitis + "#" + _
                tmp.Ethusa + "#" + _
                tmp.Idos + "#" + _
                tmp.Mera + "#" + _
                tmp.Onoma + "#" + _
                tmp.Perigrafi + "#" + _
                tmp.StartTime.ToString + "#" + _
                tmp.StopTime.ToString
            fsE.WriteLine(StrToWrite)
        Next
        fsE.Close()
        MsgBox("Exported To :" + Application.StartupPath + "\Exported.CSV", MsgBox-
Style.Information, "Prosync")
    End If
End Sub
End Class

```

*Κώδικας της φόρμας NewEditLesson.*

```
Public Class NewEditLesson
    Sub New()
        ' This call is required by the Windows Form Designer.
        InitializeComponent()
    End Sub

    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
        ReturnValues()
        Me.Close()
    End Sub

    Private Sub ReturnValues()
        EditLesson.Onoma = Me.txtOnoma.Text
        EditLesson.Perigrafi = Me.txtPerigrafi.Text
        EditLesson.Kathigitis = Me.txtKathigitis.Text
        EditLesson.Ethusa = Me.CmbXoros.Text
        EditLesson.Mera = Me.CmbImera.Text
        EditLesson.Idos = Me.cmbIdos.Text
        EditLesson.Eksamino = Me.cmbEksamino.Text
        EditLesson.StartTime = Me.nmStart.Value
        EditLesson.StopTime = Me.nmStop.Value
    End Sub

    Private Sub NewEditLesson_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Me.txtOnoma.AutoCompleteMode = AutoCompleteMode.Suggest
        Me.txtOnoma.AutoCompleteSource = AutoCompleteSource.CustomSource
        Me.txtOnoma.AutoCompleteCustomSource = AutoCompOBJ.Mathimata
        Me.txtKathigitis.AutoCompleteMode = AutoCompleteMode.Suggest
        Me.txtKathigitis.AutoCompleteSource = AutoCompleteSource.CustomSource
        Me.txtKathigitis.AutoCompleteCustomSource = AutoCompOBJ.Kathigites
        If Not EditLesson Is Nothing Then
            Me.txtOnoma.Text = EditLesson.Onoma
            Me.txtPerigrafi.Text = EditLesson.Perigrafi
            Me.txtKathigitis.Text = EditLesson.Kathigitis
            Me.CmbXoros.SelectedText = EditLesson.Ethusa
            Me.CmbImera.SelectedText = EditLesson.Mera
            If EditLesson.StartTime < 8 Or EditLesson.StartTime > 19 Then
                Me.nmStart.Value = 8
            Else
```

```
    Me.nmStart.Value = EditLesson.StartTime
End If
If EditLesson.StopTime < 9 Or EditLesson.StopTime > 20 Then
    Me.nmStop.Value = 9
Else
    Me.nmStop.Value = EditLesson.StopTime
End If
Me.cmbIdos.SelectedText = EditLesson.Idos
Me.cmbEksamino.SelectedText = EditLesson.Eksamino
End If
End Sub
```

```
Private Sub nmStart_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles nmStart.ValueChanged
    Me.nmStop.Value = Me.nmStart.Value + 1
End Sub
```

```
Private Sub Cancel_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel_Button.Click
    EditLesson = Nothing
    Me.Close()
End Sub
```

End Class

*Κώδικας της φόρμας μετακίνησης μαθημάτων. frmMove*

```
Public Class frmMove
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        ReturnValues()
        Me.Close()
    End Sub

    Private Sub ReturnValues()
        EditLesson.Mera = Me.CmbImera.Text
        EditLesson.Eksamino = Me.cmbEksamino.Text
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        EditLesson = Nothing
        Me.Close()
    End Sub
End Class
```

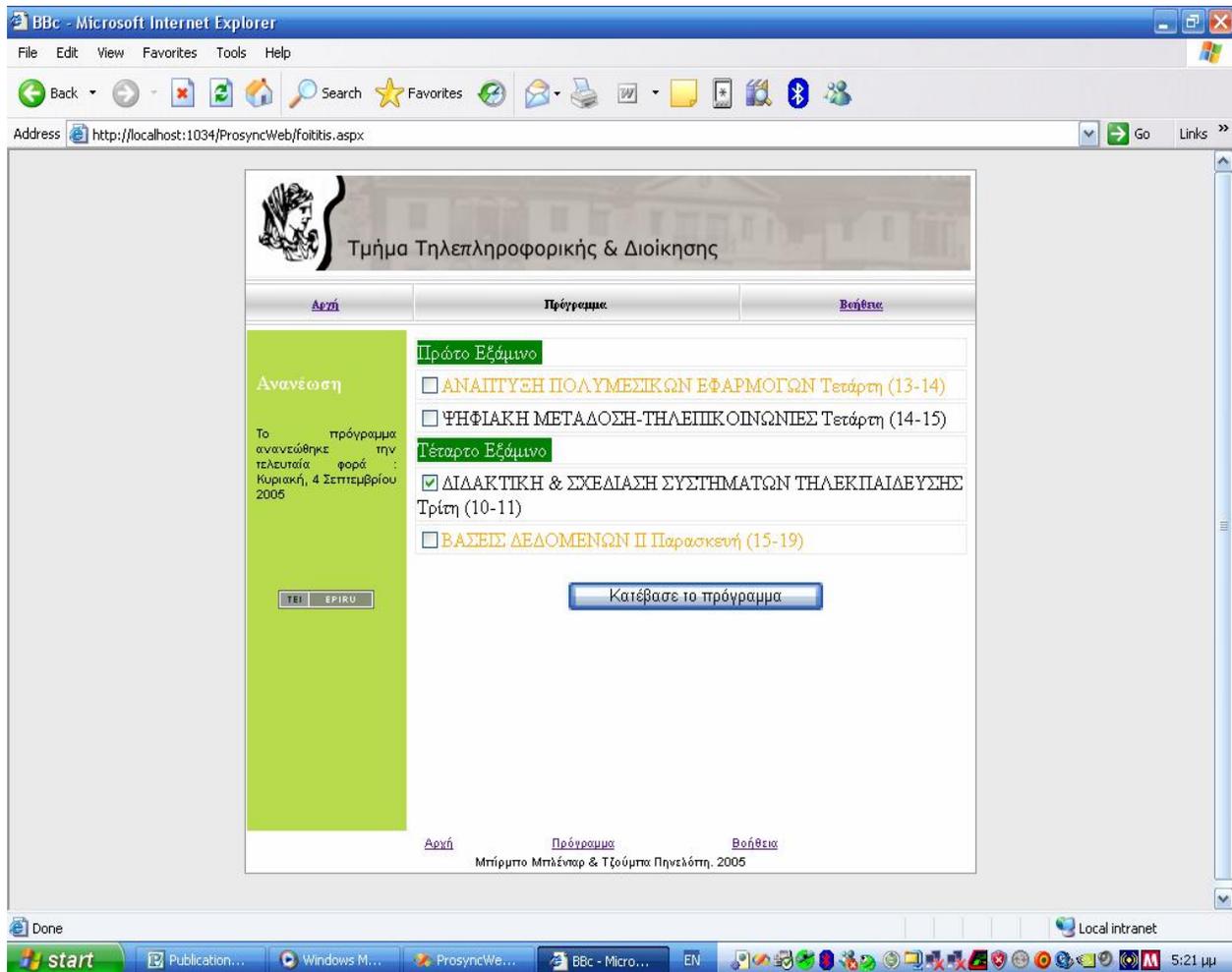
```
Private Sub frmMove_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    If Not EditLesson Is Nothing Then
        Me.CmbImera.SelectedText = EditLesson.Mera
        Me.cmbEksamino.SelectedText = EditLesson.Eksamino
    End If
End Sub
End Class
```

*Κώδικας της φόρμας About.*

```
Public Class AboutBox1
    Private Sub AboutBox1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ' Set the title of the form.
        Dim ApplicationTitle As String
        If My.Application.Info.Title <> "" Then
            ApplicationTitle = My.Application.Info.Title
        Else
            ApplicationTitle = System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If
        ApplicationTitle += " Έκδοση " + My.Application.Info.Version.ToString
        Me.Text = String.Format("About {0}", ApplicationTitle)
    End Sub
    Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.Close()
    End Sub
    Private Sub TableLayoutPanel_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs)
    End Sub
End Class
```

# PROWEB

Το ProWeb είναι ένα μικρό WebSite και αποτελείται από τρεις σελίδες. « ΑΡΧΗ» «ΠΡΟΓΡΑΜΜΑ» και «ΒΟΗΘΕΙΑ». Όταν κάνουμε κλικ στο πρόγραμμα ανοίγει η σελίδα «fititis.aspx». Το Πρώτο πράγμα που κάνει είναι να κοιτάει εάν υπάρχει το αρχείο «exported.csv», αν υπάρχει εμφανίζει όλα τα μαθήματα που περιέχει, ομαδοποιημένα ανά εξάμηνο. Αν δεν βρεθεί το «exported.csv» τότε εμφανίζεται σχετικό μήνυμα.

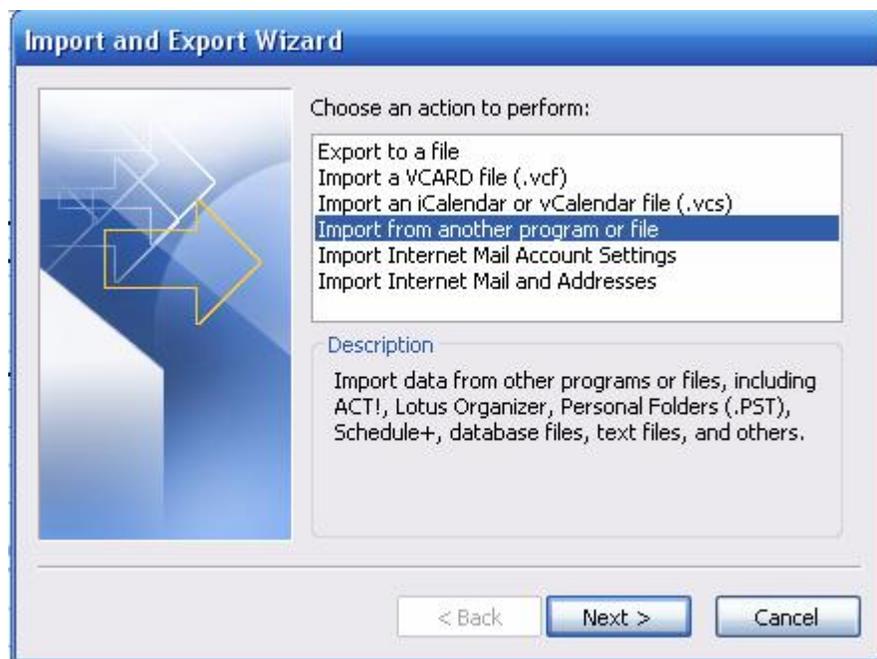
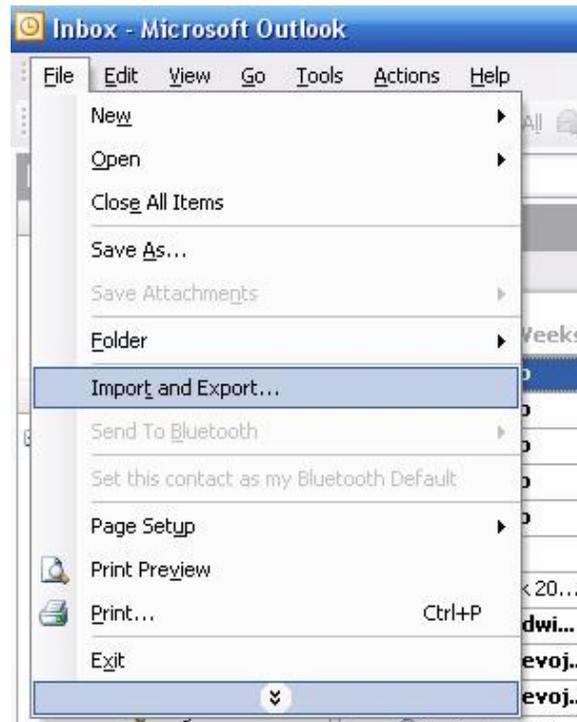


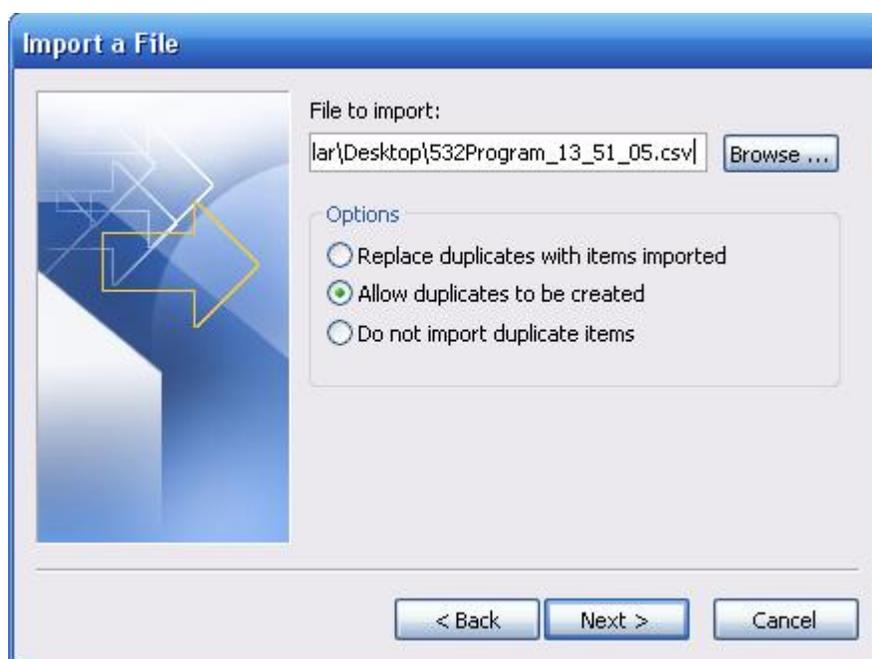
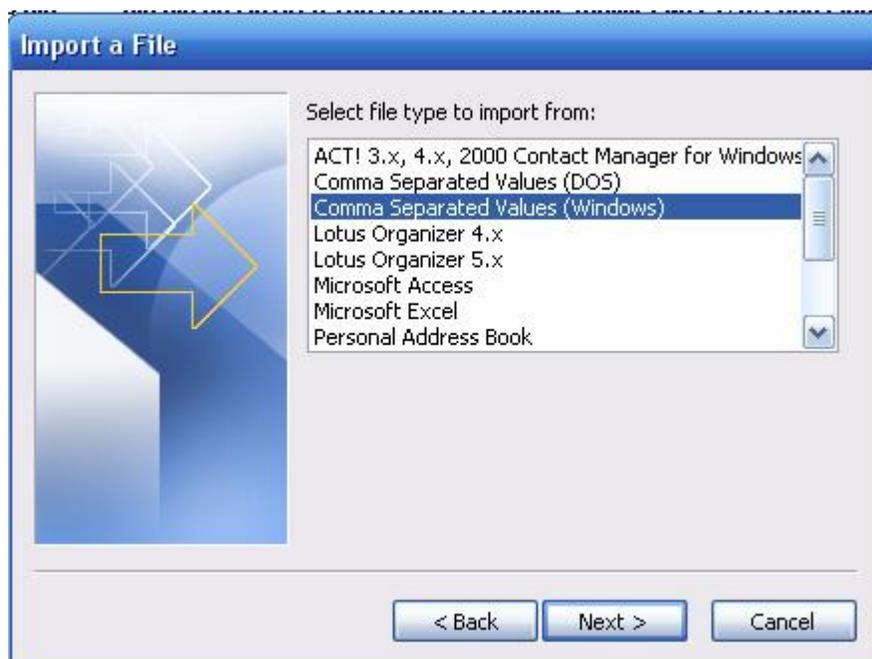
Στη λίστα των μαθημάτων τα εργαστήρια είναι με πορτοκαλί χρώμα. Αφού επιλέξουμε όλα τα μαθήματα που θέλουμε πατάμε το κουμπί «Κατέβασε το πρόγραμμα». Θα δημιουργηθεί ένα αρχείο .CSV στον ίδιο κατάλογο που είναι και οι σελίδες και θα εμφανιστεί το παράθυρο διαλόγου για να το κατεβάσουμε.

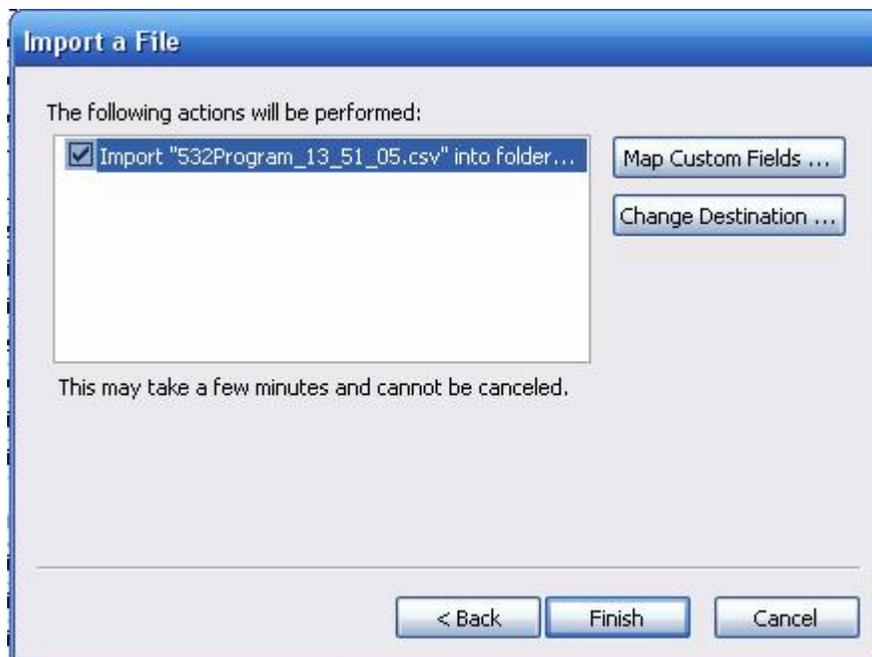
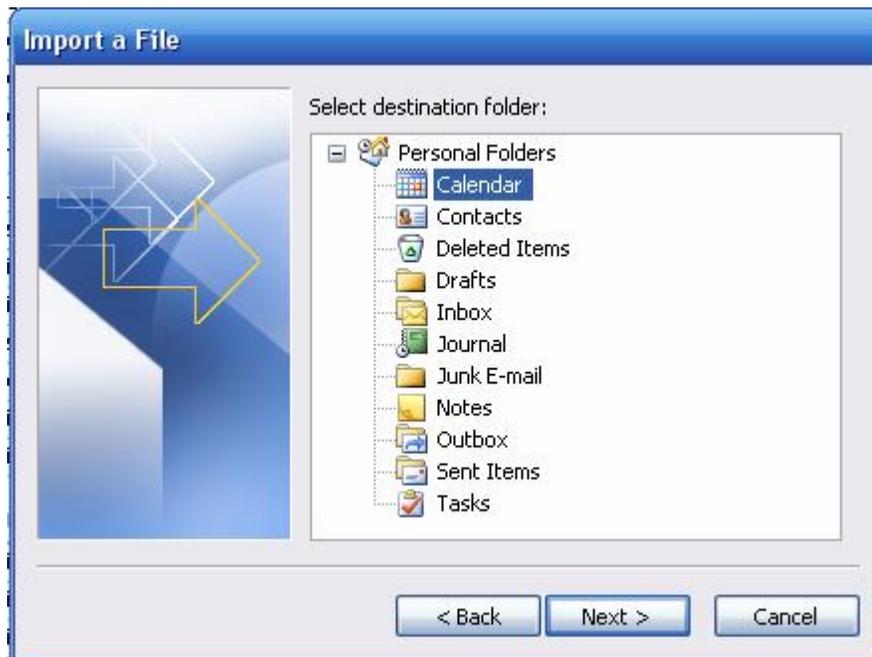
Το αρχεία αυτά δεν διαγράφονται μετά το κατέβασμα. Επειδή για κάθε σπουδαστή το πρόγραμμα δημιουργεί καινούργιο αρχείο, συνιστάται τακτικό σβήσιμο αυτών των αρχείων από τον υπεύθυνο που έχει πρόσβαση και στο Server.

## Εμφάνιση του προγράμματος στο Outlook

Παρακάτω είναι τα βήματα που πρέπει να ακολουθήσει κανείς για να ενσωματώσει το πρόγραμμα που μόλις κατέβασε στο Outlook







Πατώντας «Finish» θα δούμε το πρόγραμμα στο Calendar του Outlook. Έχοντας το πρόγραμμα στο Outlook μπορούμε να το κάνουμε Synchronize με το κινητό μας τηλέφωνο ή με τον υπολογιστή τσέπης.

## Κώδικας του PROWEB

Fititis.aspx.vb

`Imports System.IO``Imports System.Runtime.Serialization.Formatters``Imports System.Collections.Specialized``Imports System.Windows.Forms``Partial Class foititis` `Inherits System.Web.UI.Page` `Dim weeks As Integer` `Dim Data As ArrayList` `Dim contr As ArrayList` `Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load` `Try` `Dim str As String = File.GetCreationTime(Server.MapPath("Exported.CSV")).ToLongDateString()` `Label1.Text = "Το πρόγραμμα ανανεώθηκε την τελευταία φορά : " + str` `Catch q As Exception` `Label1.Text = "EXPORTED.CSV was not found!"` `Exit Sub` `End Try` `Dim currentEksamino As String = ""` `contr = New ArrayList` `contr.Clear()` `Page.EnableViewState = True` `If LoadFile() Then` `Dim tmpREC As GRecord` `Dim i As Integer` `For i = 0 To Data.Count - 1` `Dim tmpRow As New TableRow()` `Dim tmpCell As New TableCell()` `Dim tmpCheck As New CheckBox()` `tmpREC = CType(Data.Item(i), GRecord)` `If tmpREC.Eksamino <> currentEksamino Then` `Dim tmpLabel As New Label` `Dim tmpCellLabel As New TableCell` `Dim tmpRowLabel As New TableRow`

```
tmpLabel.Text = tmpREC.Eksamino + " Εξάμηνο "  
tmpLabel.ID = tmpREC.Eksamino  
tmpLabel.EnableViewState = True  
tmpLabel.BackColor = Drawing.Color.Green  
tmpLabel.ForeColor = Drawing.Color.White  
tmpCellLabel.Controls.Add(tmpLabel)  
tmpCellLabel.EnableViewState = True  
tmpCellLabel.BorderWidth = 1  
tmpRowLabel.Controls.Add(tmpCellLabel)  
tblFoitis.Rows.Add(tmpRowLabel)  
currentEksamino = tmpREC.Eksamino  
End If  
  
tmpCheck.Text = tmpREC.Onoma + " " + tmpREC.Mera + " (" +  
tmpREC.StartTime.ToString + "-" + tmpREC.StopTime.ToString + ")"  
tmpCheck.ID = tmpREC.Onoma  
tmpCheck.Checked = False  
tmpCheck.EnableViewState = True  
contr.Add(tmpCheck)  
  
tmpCell.Controls.Add(tmpCheck)  
tmpCell.EnableViewState = True  
tmpCell.BorderWidth = 1  
tmpCell.BackColor = Drawing.Color.White  
If tmpREC.Idos = "Εργαστήριο" Then  
tmpCell.ForeColor = Drawing.Color.Orange  
End If  
  
tmpRow.Cells.Add(tmpCell)  
tblFoitis.Rows.Add(tmpRow)  
Next  
End If  
Me.tblFoitis.Width = WebControls.Unit.Percentage(100)  
End Sub  
  
Private Function LoadFile() As Boolean  
Data = New ArrayList  
Try  
Dim fsE As New StreamReader(Server.MapPath("Exported.CSV"), False)  
Dim pr As String = fsE.ReadLine  
weeks = Integer.Parse(pr)  
While fsE.EndOfStream = False  
Dim tmp As GRecord  
Dim StrToRead As String  
StrToRead = fsE.ReadLine()  
Dim fragments() As String
```

```

    fragments = StrToRead.Split(CType("#", Char))
    tmp = New GRecord
    tmp.Eksamino = fragments(0)
    tmp.Kathigitis = fragments(1)
    tmp.Ethusa = fragments(2)
    tmp.Idos = fragments(3)
    tmp.Mera = fragments(4)
    tmp.Onoma = fragments(5)
    tmp.Perigrafi = fragments(6)
    tmp.StartTime = Short.Parse(fragments(7))
    tmp.StopTime = Short.Parse(fragments(8))
    Data.Add(tmp)
End While
'MsgBox(dataList.GetCurrentObject.Mera)
fsE.Close()
Return True
Catch e As IOException
    MsgBox("Error:" + e.Message)
End Try
Return False
End Function

```

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click  
'generate

```

Dim rndStr As String = (Int(999 * Rnd(Now.Ticks))).ToString
Dim FS As New System.IO.StreamWriter(Server.MapPath(rndStr + "Program_" + Format
(Now, "dd_mm_yy") + ".csv"), False)
FS.WriteLine(Q("Subject") + "," + Q("Start Date") + "," + _
Q("Start Time") + "," + Q("End Date") + "," + Q("End Time") + "," + _
Q("All day event") + "," + Q("Reminder on/off") + "," + _
Q("Reminder Date") + "," + Q("Reminder Time") + "," + _
Q("Meeting Organizer") + "," + Q("Required Attendees") + "," + _
Q("Optional Attendees") + "," + Q("Meeting Resources") + "," + _
Q("Billing Information") + "," + Q("Categories") + "," + _
Q("Description") + "," + Q("Location") + "," + _
Q("Mileage") + "," + Q("Priority") + "," + Q("Private") + "," + _
Q("Sensitivity") + "," + Q("Show time as"))
Dim pu As CheckBox
Dim dat As GRecord
Dim i As Integer
Dim baseDate As Date = Now
Dim overDate As Date = Date.op_Addition(Now, New TimeSpan(7 * weeks, 0, 0, 0, 0))
Dim DayWeek As Integer
For i = 0 To contr.Count - 1

```

```

pu = CType(contr.Item(i), CheckBox)
dat = CType(Data.Item(i), GRecord)
If pu.Checked Then
  Select Case dat.Mera
    Case "Δευτέρα" : DayWeek = 1
    Case "Τρίτη" : DayWeek = 2
    Case "Τετάρτη" : DayWeek = 3
    Case "Πέμπτη" : DayWeek = 4
    Case "Παρασκευή" : DayWeek = 5
  End Select
  Dim dateI As Date = baseDate
  While Date.op_LessThanOrEqual(dateI, overDate)
    If dateI.DayOfWeek = DayWeek Then
      Dim eventString As String
      eventString = Q(dat.Onoma) + "," + _
        Q(Format(dateI, "mm/dd/yyyy")) + "," + _
        Q(dat.StartTime.ToString + ":00:00") + "," + _
        Q(Format(dateI, "mm/dd/yyyy")) + "," + _
        Q(dat.StopTime.ToString + ":00:00") + "," + _
        Q("FALSE") + "," + _
        Q("FALSE") + "," + _
        Q(Format(dateI, "dd/mm/yyyy")) + "," + _
        Q(dat.StartTime.ToString + ":00:00") + "," + _
        Q(dat.Kathigitis) + "," + _
        Q("") + "," + _
        Q(dat.Ethusa) + "," + _
        Q("") + "," + _
        Q("Normal") + "," + _
        Q("FALSE") + "," + _
        Q("Normal") + "," + _
        Q("2")
      FS.WriteLine(eventString)
    End If
    dateI = Date.op_Addition(dateI, New TimeSpan(1, 0, 0, 0, 0))
  End While
End If
Next
FS.Close()
Response.Redirect(rndStr + "Program_" + Format(Now, "dd_mm_yy") + ".csv")
End Sub

```

```
Function Q(ByVal value As String) As String
    Return Chr(34) + value + Chr(34)
End Function
End Class
```