

Τ.Ε.Ι. ΛΑΜΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ II
Visual Basic Net

ΣΗΜΕΙΩΣΕΙΣ

ΕΡΓΑΣΤΗΡΙΟΥ

Γρηγόρης Τζιάλλας

ΛΑΜΙΑ 2009

8 Σύνθετες δομές δεδομένων και Αντικείμενα

Συχνά, για την παράσταση και αποθήκευση των δεδομένων μιας εφαρμογής, χρειάζεται η ομαδοποίηση των δεδομένων.

Για παράδειγμα τα στοιχεία ενός προσώπου και η αναπαράσταση ενός σημείου στο επίπεδο (συντεταγμένες x και y) μπορούν να παρασταθούν και να διαχειρισθούν καλύτερα με την ομαδοποίηση των δεδομένων τους.

Υπάρχουν δύο εναλλακτικές λύσεις για την ομαδοποίηση δεδομένων: οι σύνθετες δομές δεδομένων (structures) και τα αντικείμενα (objects).

Η χρήση των σύνθετων δομών είναι πιο απλή και γρήγορη ενώ η χρήση αντικειμένων δίνει μεγαλύτερη ευελιξία και δυνατότητες.

Το περιβάλλον της Visual Basic διαθέτει ένα πολύ μεγάλο πλήθος αντικειμένων τα οποία μπορεί να επαναχρησιμοποιηθούν σε νέες εφαρμογές. Τα παράθυρα μιας εφαρμογής, τα στοιχεία ελέγχου, η οθόνη τερματικού, και σχεδόν οτιδήποτε χρησιμοποιείται από την Visual Basic είναι αντικείμενα.

8.1 Σύνθετες δομές δεδομένων

Οι σύνθετες δομές δεδομένων μπορούν να θεωρηθούν ως νέοι τύποι δεδομένων οι οποίοι συντίθενται από άλλους βασικούς ή απλούστερους τύπους δεδομένων. Στο παρακάτω παράδειγμα ορίζεται μια σύνθετη δομή δεδομένων με το όνομα `PersonData` η οποία αναπαριστά το όνομα, το επώνυμο και το ύψος ενός ατόμου:

```
'Ορισμός σύνθετης δομής με τα στοιχεία ενός ατόμου
Structure PersonData
    Public Name As String
    Public Surname As String
    Public Height As Double
End Structure
```

Ο ορισμός μιας σύνθετης δομής αρχίζει με την λέξη `Structure` ακολουθούμενη από το όνομα της δομής και τελειώνει με μία γραμμή με τις λέξεις `End Structure`.

Ο ορισμός μιας δομής περιέχει τον ορισμό ενός ή περισσότερων μεταβλητών (τα στοιχεία ή πεδία της δομής) όπως για παράδειγμα το όνομα και το επώνυμο. Ο τύπος δεδομένων των μεταβλητών μιας σύνθετης δομής μπορεί να είναι κάποιος από τους βασικούς τύπους δεδομένων που υποστηρίζει η Visual Basic, ή κάποιος άλλος οριζόμενος τύπος δεδομένων όπως για παράδειγμα κάποια άλλη σύνθετη δομή.

Η αποθήκευση δεδομένων σε μια σύνθετη δομή γίνεται με την χρήση μεταβλητών τύπου σύνθετης δομής. Εάν για παράδειγμα σε ένα πρόγραμμα επιθυμούμε να αναπαραστήσουμε στοιχεία ενός ατόμου, τότε πρέπει να ορίσουμε μια μεταβλητή `p` τύπου δεδομένων `PersonData`, όπως φαίνεται στο παρακάτω παράδειγμα:

```
Dim p As personData
```

Η αναφορά στα πεδία μιας σύνθετης δομής γίνεται με τον συνδυασμό του ονόματος της μεταβλητής και του ονόματος του πεδίου.

Η ανάθεση τιμής στα πεδία μιας μεταβλητής μπορεί να γίνει με τον τελεστή της ισότητας για το κάθε επιμέρους στοιχείο της μεταβλητής, όπως για παράδειγμα:

```
'Ανάθεση τιμών στα πεδία της μεταβλητής
p.Name = "Γεώργιος"
p.Surname = "Κωνσταντίνου"
p.Height = 1.85
```

ή εναλλακτικά με την εισαγωγή της τιμής από το πληκτρολόγιο:

```
'Εισαγωγή τιμής από το πληκτρολόγιο
Console.Write("Δώστε ύψος ")
p.Height = Console.ReadLine
```

Στο παρακάτω ολοκληρωμένο παράδειγμα ορίζεται μια σύνθετη δομή με το όνομα `PersonData`, ορίζεται μεταβλητή με το όνομα `p`, γίνεται ανάθεση τιμής στα στοιχεία της μεταβλητής και εκτυπώνονται τα στοιχεία της μεταβλητής `p`.

```
Module Module1

    'Ορισμός σύνθετης δομής με τα στοιχεία ενός ατόμου
    Structure PersonData
        Public Name As String
        Public Surname As String
        Public Height As Double
    End Structure

    Sub Main()
        'Ορισμός μεταβλητής
        Dim p As PersonData

        'Ανάθεση τιμών στα πεδία της μεταβλητής
        p.Name = "Γεώργιος"
        p.Surname = "Κωνσταντίνου"

        'Εισαγωγή τιμής από το πληκτρολόγιο
        Console.Write("Δώστε ύψος ")
        p.Height = Console.ReadLine

        'Εκτύπωση των στοιχείων του ατόμου
        Console.WriteLine("Τα στοιχεία το ατόμου είναι:")
        Console.WriteLine("Όνομα      : {0}", p.Name)
        Console.WriteLine("Επώνυμο   : {0}", p.Surname)
        Console.WriteLine("Υψος      : {0:N2}", p.Height)

        Console.ReadLine()

    End Sub

End Module
```

Παράδειγμα 8.1 Ορισμός και χρήση σύνθετης δομής

Μια σύνθετη δομή μπορεί να ορίζει διαδικασίες ή συναρτήσεις οι οποίες τροποποιούν ή επιστρέφουν τα στοιχεία της. Για παράδειγμα, εκτός των πεδίων της σύνθετης δομής `PersonData`, μπορούμε να ορίσουμε μία μέθοδο η οποία επιστρέφει το ονοματεπώνυμο του ατόμου.

```
'Ορισμός σύνθετης δομής με τα στοιχεία ενός ατόμου
Structure PersonData
    Public Name As String
    Public Surname As String
    Public Height As Double

    'Ορισμός συνάρτησης η οποία επιστρέφει το ονοματεπώνυμο
    Public Function fullname() As String
        Return Name + " " + Surname
    End Function
End Structure
```

8.2 Κλάσεις αντικειμένων και αντικείμενα

Αντί της σύνθετης δομής PersonData, μπορούμε να χρησιμοποιήσουμε μια κλάση αντικειμένων με το όνομα PersonClass. Στο παρακάτω παράδειγμα ορίζουμε και χρησιμοποιούμε την κλάση αντικειμένων PersonClass.

Module module1

```
'Ορισμός κλάσης αντικειμένων με τα στοιχεία ενός ατόμου
Class PersonClass
    Public Name As String
    Public Surname As String
    Public Height As Double

    'Ορισμός μεθόδου ή οποία επιστρέφει το ονοματεπώνυμο
    Public Function fullname() As String
        Return Name + " " + Surname
    End Function
End Class

Sub Main()
    'Ορισμός μεταβλητής τύπου PersonClass
    Dim p As PersonClass
    'Δημιουργία αντικειμένου της κλάσης PersonClass
    p = New PersonClass
    'Ανάθεση τιμών στα πεδία του αντικειμένου
    p.Name = "Γεώργιος"
    p.Surname = "Κωνσταντίνου"

    'Εισαγωγή τιμής από το πληκτρολόγιο
    Console.WriteLine("Δώστε ύψος ")
    p.Height = Console.ReadLine

    'Εκτύπωση των στοιχείων του αντικειμένου
    Console.WriteLine("Τα στοιχεία το ατόμου είναι:")
    Console.WriteLine("Όνομα      : {0}", p.Name)
    Console.WriteLine("Επώνυμο   : {0}", p.Surname)
    Console.WriteLine("Ύψος      : {0:N2}", p.Height)

    Console.ReadLine()
End Sub
End Module
```

Παράδειγμα 8.2 Ορισμός και χρήση κλάσης αντικειμένων

Ο ορισμός μιας κλάσης αντικειμένων γίνεται με την λέξη Class αντί της λέξης Structure και τελειώνει με τις λέξεις End Class αντί των λέξεων End Structure.

Ενώ τα παραδείγματα 8.1 και 8.2 είναι σχεδόν παρόμοια, σημαντική διαφορά υπάρχει στον τρόπο με τον οποία δημιουργούνται τα αντικείμενα των κλάσεων.

Κατά τον ορισμό μιας μεταβλητής ενός σύνθετου τύπου δεδομένων, δεσμεύεται ο απαραίτητος χώρος μνήμης για την αποθήκευση των δεδομένων, δημιουργείται η μεταβλητή και δίνεται αρχική τιμή στα δεδομένα (π.χ. 0 για αριθμητικές τιμές). Κατά τον ορισμό όμως μιας μεταβλητής μιας κλάσης αντικειμένων, δημιουργείται απλώς μια αναφορά για κάποιο αντικείμενο της κλάσης το οποίο θα πρέπει να δημιουργηθεί με την εντολή new και να συσχετισθεί με την μεταβλητή.

Για τον ορισμό μιας μεταβλητής και την συσχέτιση της με ένα αντικείμενο, όπως φαίνεται και από το παραπάνω παράδειγμα απαιτούνται οι παρακάτω εντολές:

```
'Ορισμός μεταβλητής τύπου PersonClass
Dim p As PersonClass
'Δημιουργία αντικειμένου της κλάσης PersonClass
και συσχέτιση της με την μεταβλητή p
p = New PersonClass
```

ή εναλλακτικά η παρακάτω εντολή:

```
'Ορισμός και δημιουργία μεταβλητής τύπου PersonClass
Dim p As PersonClass = New PersonClass
```

Για ακόμη μεγαλύτερη συντομία μπορούμε να χρησιμοποιήσουμε την εντολή:

```
'Ορισμός και δημιουργία μεταβλητής τύπου PersonClass
Dim p As New PersonClass
```

Σε αντίθεση επίσης με τις σύνθετες δομές, δύο μεταβλητές μπορούν να συσχετισθούν με το ίδιο αντικείμενο. Στο παρακάτω παράδειγμα οι μεταβλητές p και q με την εντολή p = q συσχετίζονται με το ίδιο αντικείμενο. Οποιαδήποτε αλλαγή γίνεται στα πεδία της μεταβλητής p, γίνεται επίσης και για τα πεδία της μεταβλητής q.

```
'Ορισμός μεταβλητών τύπου PersonClass
Dim p, q As PersonClass
'Δημιουργία αντικειμένου της κλάσης PersonClass
και συσχέτιση της με την μεταβλητή p
p = new PersonClass
'Συσχέτιση μεταβλητής q με το ίδιο αντικείμενο
q = p
```

Στο παράδειγμα 8.2 η κλάση αντικειμένων PersonClass ορίζεται εντός της λειτουργικής μονάδος Module1. Η συνηθισμένη πρακτική για λόγους καλύτερης οργάνωσης είναι οι κλάσεις αντικειμένων να ορίζονται σε ξεχωριστά αρχεία. Η δημιουργία ενός αρχείου μιας νέας κλάσης αντικειμένων και η πρόσθεση της στην εφαρμογή γίνεται με την επιλογή Add Class από το μενού Project.

8.3 Πίνακες σύνθετων δομών και αντικειμένων

Οι πίνακες της Visual Basic εκτός από το να έχουν ως στοιχεία τους βασικούς τύπους δεδομένων, μπορούν να έχουν ως στοιχεία και σύνθετες δομές δεδομένων ή αντικείμενα.

Ο ορισμός ενός πίνακα σύνθετων δομών δεδομένων PersonData με όνομα Persons γίνεται με την δήλωση:

```
Dim Persons(3) as PersonData
```

Αντίστοιχα, ο ορισμός ενός πίνακα αντικειμένων PersonClass με όνομα Persons γίνεται με την δήλωση

```
Dim Persons(3) as PersonClass
```

Η αναφορά στα πεδία ενός αντικειμένου του πίνακα γίνεται με την χρήση του ονόματος του πίνακα, του δείκτη του στοιχείου του πίνακα και του ονόματος του πεδίου.

Π.χ. ανάθεση ονόματος στο πρώτο στοιχείο του πίνακα

```
'Ανάθεση τιμής για το όνομα του πρώτου στοιχείου
Persons(0).Name = "Γεώργιος"
```

ή εάν η εισαγωγή γίνεται από το πληκτρολόγιο

```
'Εισαγωγή τιμής από το πληκτρολόγιο
Console.Write("Δώστε όνομα ")
Persons(i).Name = Console.ReadLine
```

Στο παρακάτω παράδειγμα, ορίζεται ένας πίνακας 4 αντικειμένων τύπου PersonClass, γίνεται η εισαγωγή των στοιχείων του πίνακα από το πληκτρολόγιο και εκτυπώνονται:

- Τα στοιχεία του πίνακα
- Ο μέσος όρος ύψους του πίνακα
- Το μικρότερο και το μεγαλύτερο ύψος του πίνακα
- Το πλήθος των ατόμων του πίνακα με ύψος > 1.80

Module Module1

```
'Ορισμός κλάσης αντικειμένων με τα στοιχεία ενός ατόμου
Class PersonClass
    Public Name As String
    Public Surname As String
    Public Height As Double

    'Ορισμός μεθόδου ή οποία επιστρέφει το ονοματεπώνυμο
    Public Function fullname() As String
        Return Name + " " + Surname
    End Function
End Class

'Ορισμός πίνακα ατόμων με 4 στοιχεία
Dim persons(3) As PersonClass

Public Sub main()

    'Διάβασμα στοιχείων πίνακα
    For i As Integer = 0 To 3
        persons(i) = New PersonClass
        Console.Write("Δώστε όνομα: ")
        persons(i).Name = Console.ReadLine
        Console.Write("Δώστε επίθετο: ")
        persons(i).Surname = Console.ReadLine
        Console.Write("Δώστε ύψος: ")
        persons(i).Height = Console.ReadLine
    Next

    'Εκτύπωση στοιχείων πίνακα
    Console.WriteLine("Τα στοιχεία το πίνακα είναι")
    For i As Integer = 0 To 3
        Console.WriteLine("Ονοματεπώνυμο {0,30} Ύψος {1:N2}", _
            persons(i).fullname, persons(i).Height)
    Next

    'Εύρεση μέσου όρου ύψους
    Dim sum As Double = 0
    For i As Integer = 0 To 3
        sum = sum + persons(i).Height
    Next
    Console.WriteLine("Ο Μέσος όρος ύψους = {0:N2}", sum / 4)

    'Εύρεση μικρότερου και μεγαλύτερου ύψους
    Dim min, max As Double
    min = persons(0).Height
    max = persons(0).Height
    For i As Integer = 0 To 3
        If persons(i).Height < min Then min = persons(i).Height
```

```

        If persons(i).Height > max Then max = persons(i).Height
    Next
    Console.WriteLine("Το μεγαλύτερο ύψος είναι {0:N2}", max)
    Console.WriteLine("Το μικρότερο ύψος είναι {0:N2}", min)

    'Εύρεση πλήθους ατόμων με ύψος > 1.80
    Dim n As Integer
    n = 0
    For i As Integer = 0 To 3
        If persons(i).Height > 1.8 Then n = n + 1
    Next
    Console.WriteLine("Πλήθος ατόμων με ύψος > 1.80 = {0}", n)

    Console.ReadLine()

End Sub

End Module

```

Παράδειγμα 8.3 Πίνακας αντικειμένων

8.4 Ιδιότητες σύνθετων δομών και αντικειμένων

Αντί των πεδίων μιας σύνθετη δομής δεδομένων ή μιας κλάσης αντικειμένων, συνηθίζεται για την καλύτερη προστασία και έλεγχο των δεδομένων να χρησιμοποιούνται οι ιδιότητες των σύνθετων δομών δεδομένων ή των κλάσεων αντικειμένων. Οι ιδιότητες επιτρέπουν την ελεγχόμενη πρόσβαση στα δεδομένα με την χρήση ξεχωριστών τμημάτων κώδικα (Get για ανάγνωση και Set για την αποθήκευση) σε αντίθεση με την χρήση πεδίων όπου η πρόσβαση στα δεδομένα είναι άμεση και μη ελεγχόμενη.

Για παράδειγμα αντί της δήλωσης της δημόσιας μεταβλητής name

```

Class PersonClass
    Public Name As String

```

μπορούμε να ορίσουμε την ιδιότητα name με τις παρακάτω δηλώσεις:

```

Class PersonClass
    'Ορισμός ιδιωτικής μεταβλητής για την ιδιότητα name
    Private _name As String
    'Ορισμός ιδιότητας name
    Public Property name() As String
        Get
            Return _name
        End Get
        Set(ByVal value As String)
            _name = value
        End Set
    End Property

```

Για τον ορισμό της ιδιότητας name χρησιμοποιείται η ιδιωτική μεταβλητή με το όνομα _name (συνηθίζεται να χρησιμοποιείται ή κάτω παύλα σε συνδυασμό με το όνομα της ιδιότητας ως το όνομα της ιδιωτικής μεταβλητής, αν και μπορεί να χρησιμοποιηθεί οποιοδήποτε άλλο όνομα). Η μεταβλητή δηλώνεται ως ιδιωτική ώστε να μην είναι προσβάσιμη από κάποιο άλλο σημείο της εφαρμογής παρά μόνο από το ίδιο το αντικείμενο.

Ο ορισμός της ιδιότητας γίνεται με το ορισμό του τμήματος Get .. End Get το οποίο λειτουργεί ως συνάρτηση η οποία επιστρέφει την τιμή της ιδιότητας και του τμήματος Set .. End Set το οποίο λειτουργεί ως διαδικασία η οποία αποθηκεύει την τιμή της ιδιότητας. Το τμήμα Set .. End Set έχει ως

παράμετρο εισόδου την μεταβλητή value η τιμή της οποίας στο παραπάνω παράδειγμα αποθηκεύεται στην ιδιωτική μεταβλητή _name.

Με την χρήση της δεσμευμένης λέξη readOnly μπορούμε να ορίσουμε ότι μια ιδιότητα μπορεί να χρησιμοποιηθεί μόνο για ανάγνωση. Στην περίπτωση αυτή δεν ορίζεται το τμήμα Set .. End Set της ιδιότητας. Μια ιδιότητα επίσης μπορεί να μην διαθέτει κάποια δική της ιδιωτική μεταβλητή για την αποθήκευση της τιμής της αλλά να υπολογίζεται από κάποιες άλλες ιδιότητες η πεδία του αντικειμένου.

Στο παρακάτω παράδειγμα η ιδιότητα fullName ορίζεται ως ιδιότητα η οποία μπορεί να χρησιμοποιηθεί μόνο για ανάγνωση και η τιμή της υπολογίζεται με την χρήση άλλων ιδιοτήτων.

```
Public ReadOnly Property fullname()
    Get
        Return name + " " + surname
    End Get
End Property
```

8.5 Συμβάντα κλάσεων σύνθετων δομών και κλάσεων αντικειμένων

Μια σύνθετη δομή ή μία κλάση αντικειμένων μπορεί να ορίσει και συμβάντα τα οποία μπορούν να συμβούν κατά την εκτέλεση των μεθόδων ή την αλλαγή των ιδιοτήτων της σύνθετης δομής ή της κλάσης αντικειμένου.

Ο ορισμός ενός συμβάντος με όνομα nameChanged για την κλάση PersonClass γίνεται με την παρακάτω δήλωση:

```
'Ορισμός συμβάντων
Public Event nameChanged(ByVal aName As String)
```

Ο ορισμός ενός συμβάντος μπορεί να ορίζει προαιρετικά μία ή περισσότερες παραμέτρους. Στο παραπάνω παράδειγμα το συμβάν nameChanged έχει την παράμετρο aName η οποία αποθηκεύει την τιμή της ιδιότητας name η οποία άλλαξε.

Η πρόκληση ενός συμβάντος γίνεται με την εντολή RaiseEvent. Στο παρακάτω παράδειγμα το συμβάν nameChanged προκαλείται στο τμήμα Set .. End Set της ιδιότητας name και γίνεται κάθε φορά που αλλάζει η τιμή της ιδιότητας name.

```
Public Property Name() As String
    Get
        Return _Name
    End Get
    Set(ByVal value As String)
        _Name = value
        'Δημιουργία συμβάντος
        RaiseEvent nameChanged(value)
    End Set
End Property
```

Με την χρήση της δεσμευμένης λέξη readOnly μπορούμε να ορίσουμε ότι μια ιδιότητα μπορεί να χρησιμοποιηθεί μόνο για ανάγνωση. Στην περίπτωση αυτή δεν ορίζεται το τμήμα Set .. End Set της ιδιότητας. Μια ιδιότητα επίσης μπορεί να μην διαθέτει κάποια δική της ιδιωτική μεταβλητή για την αποθήκευση της τιμής της αλλά να υπολογίζεται από κάποιες άλλες ιδιότητες η πεδία του αντικειμένου.

Για να γίνει χρήση των συμβάντων ενός αντικειμένου από μια εφαρμογή πρέπει η μεταβλητή του αντικειμένου να είναι καθολική μεταβλητή και να δηλωθεί με την δεσμευμένη λέξη WithEvents.

```
Dim WithEvents p As PersonClass
```


Για τον χειρισμό των συμβάντων ενός αντικειμένου πρέπει να ορισθεί κάποια διαδικασία η οποία θα καλείται κάθε φορά που προκαλείται το συμβάν. Η δήλωση του ονόματος της διαδικασίας γίνεται με την χρήση της δεσμευμένης λέξης `Handles` η οποία ακολουθείται από το όνομα της μεταβλητής του αντικειμένου, μία τελεία και το όνομα του συμβάντος. Το όνομα της διαδικασίας συνήθως είναι το όνομα της μεταβλητής ακολουθούμενο από κάτω παύλα και το όνομα του συμβάντος.

Για παράδειγμα, η διαδικασία για τον χειρισμό του συμβάντος `nameChanged` της μεταβλητής `p` ονομάζεται `p_nameChanged` και για τον ορισμό της χρησιμοποιείται η δήλωση `Handles p.nameChanged`. Η διαδικασία αυτή εκτυπώνει της ώρα του συμβάντος ή οποία δίνεται από την μέθοδο `now` της κλάσης αντικειμένων `Date` και την τιμή της ιδιότητας `name` η οποία άλλαξε.

```
Sub p_nameChanged(ByVal aName As String) Handles p.nameChanged
    Console.WriteLine("{0} Το όνομα άλλαξε σε {1}", _
        Date.Now.ToString("dd-MM-yy hh:mm:ss"), aName)
End Sub
```

Στο παρακάτω ολοκληρωμένο παράδειγμα ορίζεται κλάση `PersonClass` με ιδιότητα `name` και συμβάν `name changed`. Ορίζεται επίσης μεταβλητή `p` και διαδικασία `p_nameChanged` η οποία καλείται όταν αλλάζει η ιδιότητα `name` του αντικειμένου `p`.

```
Module Module1
    'Ορισμός κλάσης αντικειμένων με τα στοιχεία ενός ατόμου
    Class PersonClass
        'Ορισμός συμβάντος
        Public Event nameChanged(ByVal aName As String)
        'Ορισμός ιδιότητας name
        Public _Name As String
        Public Property Name() As String
            Get
                Return _Name
            End Get
            Set(ByVal value As String)
                _Name = value
                'Δημιουργία συμβάντος
                RaiseEvent nameChanged(value)
            End Set
        End Property
    End Class

    'Ορισμός καθολικής μεταβλητής p
    Dim WithEvents p As PersonClass

    Sub Main()
        'Δημιουργία αντικειμένου
        p = New PersonClass
        'Απόδοση τιμής στην ιδιότητα name
        p.Name = "Georgios"
        Console.ReadLine()
    End Sub

    'Διαδικασία για τον χειρισμό του συμβάντος nameChanged
    Sub p_nameChanged(ByVal aName As String) Handles p.nameChanged
        Console.WriteLine("{0} Το όνομα άλλαξε σε {1}", _
            Date.Now.ToString("dd-MM-yy hh:mm:ss"), aName)
    End Sub
End Module
```

Παράδειγμα 8.4 Συμβάντα κλάσης αντικειμένων

8.6 Κληρονομικότητα κλάσεων αντικειμένων

Με την δήλωση `inherits`, μια κλάση αντικειμένων μπορεί να κληρονομεί τα πεδία, τις ιδιότητες, τα συμβάντα και τις μεθόδους της σε κάποια άλλη κλάση αντικειμένων. Στο παρακάτω παράδειγμα ορίζεται μια κλάση `StudentClass` η οποία κληρονομεί τα χαρακτηριστικά της κλάσης `PersonClass` με την δήλωση. Η κλάση αντικειμένων `StudentClass` κληρονομεί τα πεδία `name`, `surname` και `height` και την μέθοδο `fullName` από την κλάση αντικειμένων `PersonClass` και προσθέτει τις ιδιότητες `department`, `semester` και την μέθοδο `allData`.

Module Module1

```
'Ορισμός κλάσης αντικειμένων με τα στοιχεία ενός ατόμου
Class PersonClass
    Public Name As String
    Public Surname As String
    Public Height As Double

    'Ορισμός μεθόδου ή οποία επιστρέφει το ονοματεπώνυμο
    Public Function fullname() As String
        Return Name + " " + Surname
    End Function
End Class

'Ορισμός κλάσης αντικειμένων με τα στοιχεία ενός σπουδαστή
Public Class StudentClass
    'Ορισμός κληρονομικότητας απο την κλάση PersonClass
    Inherits PersonClass

    'Ορισμός νέων πεδίων για την κλάση
    Public department As String
    Public semester As String

    'Ορισμός νέας μεθόδου
    Public Function allData() As String
        Return Name + " " + Surname + " " + _
            department + " " + semester
    End Function

End Class

Sub Main()
    Dim s As StudentClass = New StudentClass
    s.Name = "Γεώργιος"
    s.Surname = "Κωνσταντίνου"
    s.height = 1.85
    s.department = "Ηλεκτρονικής"
    s.semester = "Α"
    Console.WriteLine(s.fullname)
    Console.WriteLine(s.allData)
    Console.ReadLine()
End Sub

End Module
```

Παράδειγμα 8.5 Κληρονομικότητα κλάσεων αντικειμένων

Μια κλάση μπορεί να τροποποιήσει τα πεδία, τις ιδιότητες, και τις μεθόδους τις οποίες κληρονομεί με την δήλωση `Shadows`.

Στο παρακάτω παράδειγμα το πεδίο height και η μέθοδος fullname τροποποιούνται ώστε το πεδίο height να είναι τύπου Single και η μέθοδος fullName να επιστρέφει επιπλέον και το εξάμηνο του σπουδαστή.

```
'Τροποποίηση μεταβλητής η οποία κληρονομείται
Public Shadows height As Single

'Τροποποίηση μεθόδου η οποία κληρονομείται
Public Shadows Function fullname() As String
    Return Name + " " + Surname + " " + semester
End Function
```

Εναλλακτικά, αντί της δήλωσης Shadows για την τροποποίηση κάποιου χαρακτηριστικού μιας κλάσης, μπορεί να χρησιμοποιηθεί η δήλωση overrides. Η δήλωση overrides μπορεί να χρησιμοποιηθεί μόνον εάν το χαρακτηριστικό αυτό έχει δηλωθεί στην κλάση από την οποία κληρονομείται ως overridable. Η δήλωση overrides δεν μπορεί να χρησιμοποιηθεί για τα πεδία ενός αντικειμένου.

Στο παρακάτω παράδειγμα η μέθοδος fullname δηλώνεται ως overridable από την κλάση αντικειμένων PersonClass και τροποποιείται από την κλάση αντικειμένων StudentClass με την δήλωση overrides.

```
'Ορισμός μεθόδου fullname στην κλάση PersonClass
Public Overridable Function fullname() As String
    Return Name + " " + Surname
End Function
```

```
'Τροποποίηση μεθόδου από την κλάση StudentClass
Public Overrides Function fullname() As String
    Return Name + " " + Surname + " " + semester
End Function
```

Μια κλάση αντικειμένων μπορεί να καλέσει κάποιο χαρακτηριστικό της κλάσης από την οποία κληρονομεί, ακόμη κι εάν το έχει τροποποιήσει, με την χρήση της λέξης MyBase. Στο παρακάτω παράδειγμα η κλάση StudentClass ορίζει την τροποποιημένη μέθοδο fullName κάνοντας χρήση της μεθόδου fullName της προγονικής της κλάσης PersonClass.

```
'Τροποποίηση μεθόδου από την κλάση StudentClass
Public Overrides Function fullname() As String
    Return MyBase.fullname + " " + semester
End Function
```

8.7 Η μέθοδος new για την δημιουργία νέων αντικειμένων

Η μέθοδος new η οποία χρησιμοποιείται για την δημιουργία αντικειμένων δεν χρειάζεται να ορισθεί κατά τον ορισμό μιας νέας κλάσης αντικειμένων. Σε περίπτωση όμως που χρειάζεται να ορισθούν κάποιες συγκεκριμένες αρχικές τιμές για τα δεδομένα τότε αντικειμένου, τότε μπορούμε να την ορίσουμε ως μέθοδο του αντικειμένου με όνομα new.

Στο παρακάτω παράδειγμα ορίζεται η μέθοδος new για την κλάση αντικειμένων PersonClass ώστε να δίνει αρχικές τιμές «Άγνωστο» στο όνομα και το επώνυμο ενός ατόμου.

```
'Ορισμός μεθόδου new στην κλάση PersonClass
Public Sub New()
    Name = "Άγνωστο"
    Surname = "Άγνωστο"
End Sub
```

Μια κλάση αντικειμένων μπορεί να ορίζει περισσότερες από μία μεθόδους με το όνομα new οι οποίες έχουν διαφορετικές παραμέτρους εισόδου. Στο παρακάτω παράδειγμα ορίζεται μια δεύτερη μέθοδος new η οποία έχει ως παραμέτρους εισόδου το όνομα και το επώνυμο του ατόμου. Οι παράμετροι αυτοί χρησιμοποιούνται για τον καθορισμό των τιμών των πεδίων name και surname του αντικειμένου κατά την δημιουργία του.

```
Public Sub New(ByVal aName As String, ByVal aSurname As String)
    Name = aName
    Surname = aSurname
End Sub
```

8.8 Κοινόχρηστες μεταβλητές, ιδιότητες, συμβάντα και μέθοδοι κλάσεων αντικειμένων

Με την δεσμευμένη λέξη shared μπορούμε να καθορίσουμε κοινόχρηστες μεταβλητές, ιδιότητες, συμβάντα και μεθόδους μιας μεθόδου μιας κλάσης αντικειμένων.

Οι κοινόχρηστες μεταβλητές, ιδιότητες, συμβάντα και μέθοδοι ανήκουν στην κλάση αντικειμένων (και όχι στα αντικείμενα της κλάσης) και μπορούν να χρησιμοποιηθούν χωρίς αναγκαστικά να δημιουργηθεί κάποιο αντικείμενο της κλάσης. Η πρόσβαση στα κοινόχρηστα αυτά χαρακτηριστικά γίνεται κάνοντας χρήση του ονόματος της κλάσης των αντικειμένων αντί του ονόματος της μεταβλητής του αντικειμένων.

Στο παρακάτω παράδειγμα ορίζεται μια κοινόχρηστη μεταβλητή με όνομα PersonsCount η οποία χρησιμοποιείται για να αποθηκεύει το πλήθος των αντικειμένων της κλάσης PersonClass. Η μέθοδος new αυξάνει την τιμή της μεταβλητής κατά ένα κάθε φορά που δημιουργείται ένα νέο αντικείμενο και η κοινόχρηστη μέθοδος showPersonsCount εκτυπώνει το πλήθος των αντικειμένων της κλάσης.

```
Module Module1
    'Ορισμός κλάσης αντικειμένων με τα στοιχεία ενός ατόμου
    Class PersonClass
        Public Name As String
        Public Surname As String
        'Ορισμός κοινόχρηστης μεταβλητής
        Public Shared PersonsCount As Integer = 0
        'Ορισμός μεθόδου για την δημιουργία αντικειμένων
        Public Sub New(ByVal aName As String, _
            ByVal aSurname As String)
            Name = aName
            Surname = aSurname
            PersonsCount += 1
        End Sub
        'Ορισμός κοινόχρηστης μεθόδου
        Public Shared Sub showPersonsCount()
            Console.WriteLine("PersonsCount = {0}", PersonsCount)
        End Sub
    End Class

    Sub Main()
        'Δημιουργία δύο αντικειμένων
        Dim p1, p2 As PersonClass
        p1 = New PersonClass("Γιώργος", "Ιωάννου")
        p2 = New PersonClass("Πέτρος", "Ιωάννου")
        'Χρήση κοινόχρηστης μεθόδου
        PersonClass.showPersonsCount()
        Console.ReadLine()
    End Sub
End Module
```

Παράδειγμα 8.6 Κοινόχρηστες μεταβλητές και μέθοδοι

8.9 Ασκήσεις Κεφαλαίου 8

Ασκηση 8.1

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_8_ask_1, αντιγράψτε την λειτουργική μονάδα του παραδείγματος 8.1 και εκτελέστε την εφαρμογή.

Τροποποιήστε την εφαρμογή ώστε η εισαγωγή όλων των στοιχείων να γίνεται από το πληκτρολόγιο.

Προσθέστε δύο νέες μεταβλητές με ονόματα r και s τύπου PersonData. Το τροποποιημένο πρόγραμμα ζητά από τον χρήστη να εισάγει επιπλέον στοιχεία και για τις μεταβλητές r και s και εκτυπώνει:

- α) Τα στοιχεία και των τριών μεταβλητών
- β) Το μέσο όρο του ύψους

Ασκηση 8.2

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_8_ask_2, αντιγράψτε την λειτουργική μονάδα του παραδείγματος 8.2 και εκτελέστε την εφαρμογή.

Τροποποιήστε την εφαρμογή ώστε η κλάση αντικειμένων PersonClass να ορίζεται σε ξεχωριστό αρχείο με όνομα PersonClass. (Επιλέξτε Add Class από το μενού Project για την δημιουργία αρχείου κλάσης αντικειμένων και διαγράψτε τον ορισμό της κλάσης από την λειτουργική μονάδα module1)

Τροποποιήστε την εφαρμογή ώστε η εισαγωγή όλων των στοιχείων να γίνεται από το πληκτρολόγιο.

Προσθέστε δύο νέα αντικείμενα με ονόματα r και s τύπου PersonClass. Το τροποποιημένο πρόγραμμα ζητά από τον χρήστη να εισάγει επιπλέον στοιχεία και για τις μεταβλητές r και s και εκτυπώνει:

- α) Τα στοιχεία και των τριών αντικειμένων
- β) Το μεγαλύτερο, το μικρότερο ύψος και τα ονοματεπώνυμα των ατόμων με το μεγαλύτερο και το μικρότερο ύψος

Ασκηση 8.3

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_8_ask_3, αντιγράψτε την λειτουργική μονάδα του παραδείγματος 8.3 και εκτελέστε την εφαρμογή.

Τροποποιήστε την εφαρμογή ώστε αντί κλάσης αντικειμένων με το όνομα PersonClass να ορίζεται σύνθετη δομή με όνομα PersonData και ο πίνακας persons να είναι πίνακας σύνθετων δομών τύπου PersonData.

Ασκηση 8.4

Δημιουργήστε εφαρμογή με όνομα lab_8_ask_4 η οποία ορίζει κλάση αντικειμένων ProductClass με πεδία name, price, και remaining τα οποία αντιστοιχούν στο όνομα, την τιμή μονάδος και την διαθέσιμη ποσότητα. Η εφαρμογή ορίζει επίσης έναν μονοδιάστατο πίνακα 3 στοιχείων τύπου ProductClass με το όνομα products. Το πρόγραμμα ζητά από τον χρήστη να εισάγει τα στοιχεία του πίνακα products και στην συνέχεια εκτυπώνει τον μέσο όρο τιμής των προϊόντων και τα ονόματα των προϊόντων με διαθέσιμη ποσότητα μικρότερη του 2.

Άσκηση 8.5

Δημιουργήστε εφαρμογή τερματικού με όνομα lab_8_ask_5 η οποία ορίζει κλάση αντικειμένων ProductClass με ιδιότητες (προσοχή όχι πεδία) name, price, και remaining τα οποία αντιστοιχούν στο όνομα, την τιμή μονάδος και την διαθέσιμη ποσότητα. Η εφαρμογή δεν πρέπει να επιτρέπει την καταχώρησης αρνητική τιμής στις ιδιότητες price και remaining. Σε περίπτωση που δοθεί αρνητική τιμή καταχωρείται η τιμή 0 ως τιμή της ιδιότητας. Το πρόγραμμα ορίζει μια μεταβλητή με όνομα p τύπου ProductClass, δημιουργεί ένα νέο αντικείμενο το οποίο συσχετίζει με την μεταβλητή p, ζητά από τον χρήστη να εισάγει τις ιδιότητες του αντικειμένου και εκτυπώνει τα στοιχεία του στην οθόνη.

Άσκηση 8.6

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_8_ask_6, αντιγράψτε την λειτουργική μονάδα του παραδείγματος 8.4 και εκτελέστε την εφαρμογή.

Τροποποιήστε την εφαρμογή ώστε η κλάση PersonClass να ορίζει επιπλέον την ιδιότητα surname και το συμβάν surnameChanged. Ορίστε επίσης διαδικασία p_surnameChanged η οποία χειρίζεται το συμβάν p.surnameChanged.

Άσκηση 8.7

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_8_ask_7, αντιγράψτε την λειτουργική μονάδα του παραδείγματος 8.5 και εκτελέστε την εφαρμογή.

Τροποποιήστε την εφαρμογή ώστε η κλάση StudentClass να ορίζει επιπλέον το πεδίο School (τύπου String) το οποίο αντιστοιχεί στην σχολή του σπουδαστή.

Τροποποιήστε την μέθοδο main ώστε να δίνει την τιμή “ΣΤΕΦ” στο πεδίο school.

Τροποποιήστε την μέθοδο allData ώστε να εκτυπώνει και την σχολή του σπουδαστή.

Ορίστε μέθοδο fullName για την κλάση αντικειμένων StudentClass η οποία τροποποιεί την μέθοδο fullName η οποία κληρονομείται από την κλάση PersonClass ώστε να εκτυπώνει πρώτα το επίθετο και στην συνέχεια το όνομα του σπουδαστή. Η μέθοδος fullName για την κλάση PersonClass θα πρέπει να παραμείνει ή ίδια.

Μετατρέψτε το πεδίο semester σε ιδιότητα η οποία να μπορεί να δεχθεί μόνο τις τιμές “Α”, “Β”, “Γ”, “Δ”, “Ε”, “Ζ”, “ΣΤ”, “Η” και “ΠΤΥΧΙΟ”. Σε περίπτωση που δοθεί κάποια άλλη τιμή, ως τιμή της ιδιότητας αποθηκεύεται ή λέξη “Άγνωστο”.

Άσκηση 8.8

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_8_ask_7, αντιγράψτε την λειτουργική μονάδα του παραδείγματος 8.6 και εκτελέστε την εφαρμογή.

Προσθέστε ιδιότητα telephone τύπου String στην κλάση PersonClass.

Προσθέστε μέθοδο new στην κλάση PersonClass η οποία δέχεται ως παραμέτρους το όνομα, το επίθετο και το τηλέφωνο του ατόμου.

Τροποποιήστε την διαδικασία main ώστε να δίνει τιμές και στην ιδιότητα telephone κατά την δημιουργία των αντικειμένων.

9 Απαριθμητοί τύποι δεδομένων - Συλλογές

9.1 Ορισμός απαριθμητών τύπων δεδομένων

Οι απαριθμητοί τύποι δεδομένων (enumerations) χρησιμοποιούνται σε περιπτώσεις όπου μια μεταβλητή μπορεί να πάρει μόνον κάποιες συγκεκριμένες διακριτές τιμές, όπως για παράδειγμα ο τύπος δεδομένων `consoleColor` ο οποίος ορίζει τις δυνατές τιμές χρωμάτων για την οθόνη τερματικού (Console) οι οποίες είναι 16 και αντιστοιχούν στους ακέραιους αριθμούς από 0 έως 15.

Οι δυνατοί τύποι τιμών ενός απαριθμητού τύπου δεδομένων είναι `Byte`, `Integer`, `Long`, `SByte`, `Short`, `UInteger`, `ULong`, ή `UShort` και για την κάθε δυνατή τιμή αντιστοιχεί και ένα συμβολικό όνομα. Για παράδειγμα ο τύπος δεδομένων `ConsoleColor` ορίζει δεκαέξι ακέραιες τιμές από 0 έως 15 και αντίστοιχα συμβολικά ονόματα:

Black	0
DarkBlue	1
DarkGreen	2
DarkCyan	3
DarkRed	4
DarkMagenta	5
DarkYellow	6
Gray	7
DarkGray	8
Blue	9
Green	10
Cyan	11
Red	12
Magenta	13
Yellow	14
White	15

Εκτός από τους ήδη προκαθορισμένους τύπους απαριθμητών δεδομένων, μπορούμε να καθορίσουμε και νέους τύπους. Στο παρακάτω παράδειγμα ορίζεται ένας νέος τύπος δεδομένων με όνομα `DaysEnum` και τιμές τις ημέρες της εβδομάδας.

```
Enum DaysEnum As Integer
    Κυριακή
    Δευτέρα
    Τρίτη
    Τετάρτη
    Πέμπτη
    Παρασκευή
    Σάββατο
    Αγνωστη
End Enum
```

Ο ορισμός ενός απαριθμητού τύπου δεδομένων αρχίζει με την λέξη `Enum`, το όνομα του τύπου δεδομένων, την λέξη `As` και τον τύπο της αριθμητικής τιμής των δεδομένων η οποία μπορεί να είναι `Byte`, `Integer`, `Long`, `SByte`, `Short`, `UInteger`, `ULong`, ή `UShort`. Η δήλωση συνεχίζει με τα συμβολικά ονόματα των τιμών και τελειώνει με μία γραμμή με τις λέξεις `End Enum`.

Η απόδοση αριθμητικών τιμών γίνεται αυτόματα με αρχική τιμή 0 για την Κυριακή και τιμές οι οποίες αυξάνονται κατά ένα για τις επόμενες ημέρες.

Στον ορισμό ενός απαριθμητού τύπου δεδομένων μπορούμε να καθορίσουμε τις αριθμητικές τιμές οι οποίες αντιστοιχούν στα διάφορα συμβολικά ονόματα όπως στο παρακάτω παράδειγμα:

```
Enum DaysEnum As Integer
```

```

Κυριακή = 1
Δευτέρα = 2
Τρίτη = 3
Τετάρτη = 4
Πέμπτη = 5
Παρασκευή = 6
Σάββατο = 7
Αγνωστή = 8
End Enum

```

9.2 Απόδοση τιμής σε μεταβλητές

Ο ορισμός μιας μεταβλητής τύπου απαριθμητών τύπων δεδομένων γίνεται όπως και για κάθε άλλο τύπο μεταβλητής.

Π.χ.

```
Dim day as DaysEnum
```

Η απόδοση τιμής μπορεί να γίνει είτε με την χρήση της συμβολικής τιμής είτε με την χρήση της αριθμητικής τιμής.

Παράδειγμα απόδοσης τιμής με την χρήση συμβολικής τιμής:

```
day = DaysEnum.Κυριακή
```

Παράδειγμα απόδοσης τιμής με την χρήση αριθμητικής τιμής:

```
day = 0
```

Για την απόδοση τιμής σε μια μεταβλητή τύπου απαριθμητής μεταβλητής, μπορούμε να χρησιμοποιήσουμε την αριθμητική τιμή την οποία εισάγει ο χρήστης:

Π.χ.

```

'Απόδοση τιμής με διάβασμα από το πληκτρολόγιο
Console.WriteLine("Δώστε αριθμό ημέρας (από 0 έως 6) ")
day = Console.ReadLine

```

ή να χρησιμοποιήσουμε αφού γίνει η κατάλληλη μετατροπή το κείμενο το οποίο αντιστοιχεί στην συμβολική ονομασία της τιμής με την χρήση της μεθόδου parse η οποία δέχεται ως παραμέτρους το τύπο δεδομένων ο οποίος δίνεται με την συνάρτηση GetType(DaysEnum) όπου DaysEnum είναι το όνομα του απαριθμητού τύπου δεδομένων και το κείμενο της συμβολικής ονομασίας της τιμής.

Π.χ.

```

'Απόδοση τιμής με διάβασμα από το πληκτρολόγιο της
ονομασίας του συμβόλου
Console.WriteLine("Δώστε όνομα ημέρας ")
Dim xx As String
xx = Console.ReadLine
day = DaysEnum.Parse(GetType(DaysEnum), xx)

```

Η εκτύπωση της αριθμητικής τιμής μια μεταβλητής απαριθμητού τύπου δεδομένων γίνεται με τις παρακάτω εντολή:

```

'Εκτύπωση αριθμητικής τιμής
Console.WriteLine(day)

```

ή εναλλακτικά με την εντολή

```
Console.WriteLine("Η αριθμητική τιμή είναι {0:D}", day)
```


Η εκτύπωση της ονομασίας της συμβολικής τιμής γίνεται με την εντολή:

```
'Εκτύπωση τιμής
Console.WriteLine(day.ToString)
```

ή εναλλακτικά με την εντολή

```
Console.WriteLine("Η συμβολική ονομασία είναι {0}", day)
```

Ο έλεγχος εάν μία μεταβλητή τύπου απαριθμητού τύπου δεδομένων έχει κάποια αποδεκτή ή καθορισμένη τιμή γίνεται με την μέθοδο `isDefined`. Παράμετροι εισόδου της μεθόδου είναι ο τύπος δεδομένων `ων` και η μεταβλητή η οποία θέλουμε να ελέγξουμε εάν έχει αποδεκτή τιμή.

Π.χ.

```
'Έλεγχος για την ύπαρξη τιμής
If DaysEnum.IsDefined(GetType(DaysEnum), day) Then
    Console.WriteLine("Η τιμή υπάρχει")
Else
    Console.WriteLine("Δεν υπάρχει η τιμή")
End If
```

Η εκτύπωση των τιμών και των αντιστοιχιών τιμών ενός απαριθμητού τύπου δεδομένων γίνεται με την χρήση της μεθόδου `GetValues`, όπως φαίνεται και στο παρακάτω παράδειγμα:

```
'Εκτύπωση τιμών και συμβόλων απαριθμητού τύπου
For Each d As DaysEnum In _
    DaysEnum.GetValues(GetType(DaysEnum))
    Console.WriteLine("{0:D} {0:G}", d)
Next
```

Η μέθοδος `GetValues` ενός απαριθμητού τύπου δεδομένων επιστρέφει όλες τις δυνατές τιμές του τύπου δεδομένων. Παράμετρος εισόδου της συνάρτησης είναι ο τύπος δεδομένων ο οποίος δίνεται με την συνάρτηση `GetType(DaysEnum)`.

9.3 Παράδειγμα απαριθμητού τύπου δεδομένων

Το παρακάτω παράδειγμα ορίζει έναν απαριθμητό τύπο δεδομένων `DaysEnum` ο οποίος αντιστοιχεί στις ημέρες τις, εβδομάδες, ορίζει μεταβλητή με το όνομα `day` τύπου `DaysEnum`, αποδίδει τιμή στην μεταβλητή με διάφορους τρόπους, ελέγχει την τιμή της μεταβλητής και εκτυπώνει την τιμή της μεταβλητής καθώς και όλες τις δυνατές τιμές της μεταβλητής.

```
Module Module1

    Enum DaysEnum As Integer
        Κυριακή
        Δευτέρα
        Τρίτη
        Τετάρτη
        Πέμπτη
        Παρασκευή
        Σάββατο
        Αγνωστη
    End Enum

    Sub Main()
        'Ορισμός μεταβλητής και απόδοση τιμής
        Dim day As DaysEnum
        day = DaysEnum.Παρασκευή
        Console.WriteLine(day)
        Console.WriteLine(day.ToString)
```

```

'Απόδοση τιμής με διάβασμα από το πληκτρολόγιο
Console.Write("Δώστε αριθμό ημέρας (από 0 έως 6) ")
day = Console.ReadLine
'Εκτύπωση τιμής
Console.WriteLine(day)
Console.WriteLine(day.ToString)
Console.WriteLine("{0:D} {0:G}", day)
' D Decimal, G General (default)
'Έλεγχος για την ύπαρξη τιμής
If DaysEnum.IsDefined(GetType(DaysEnum), day) Then
    Console.WriteLine("Η τιμή υπάρχει")
Else
    Console.WriteLine("Δεν υπάρχει η τιμή")
End If
'Έλεγχος τιμής
If day = DaysEnum.Κυριακή Then
    Console.WriteLine("Αργία")
Else
    Console.WriteLine("Εργάσιμη Ημέρα")
End If
'Απόδοση τιμής με διάβασμα από το πληκτρολόγιο του συμβόλου
Console.Write("Δώστε όνομα ημέρας ")
Dim xx As String
xx = Console.ReadLine
If DaysEnum.IsDefined(GetType(DaysEnum), xx) Then
    day = DaysEnum.Parse(GetType(DaysEnum), xx)
Else
    day = DaysEnum.Άγνωστη
End If
Console.WriteLine(day)
Console.WriteLine(day.ToString)
'Functions GetType, GetValues, GetNames
'Εκτύπωση τιμών και συμβόλων απαριθμητού τύπου
For Each d As DaysEnum In _
    DaysEnum.GetValues(GetType(DaysEnum))
    Console.WriteLine("{0:D} {0:G}", d)
Next
'Εκτύπωση συμβόλων απαριθμητού τύπου
For Each s As String In DaysEnum.GetNames(GetType(DaysEnum))
    Console.WriteLine(s)
Next
Console.ReadLine()
End Sub

End Module

```

Παράδειγμα 9.1 Χρήση απαριθμητού τύπου δεδομένων

9.4 Συλλογές

Η κλάση αντικειμένων Collection μπορεί να χρησιμοποιηθεί για να δημιουργήσουμε συλλογές δεδομένων ή αντικειμένων. Μια συλλογή δεδομένων μπορεί να θεωρηθεί ως ένας δυναμικός πίνακας δεδομένων με την δυνατότητα πρόσθεσης νέων δεδομένων σε συγκεκριμένες θέσεις, διαγραφής δεδομένων και δυνατότητας αναφοράς στα στοιχεία της συλλογής με την χρήση είτε του δείκτη του στοιχείου, είτε με την χρήση λέξεων κλειδιών.

Η δημιουργία μιας συλλογής γίνεται με την δημιουργία ενός αντικειμένου της κλάσης αντικειμένων Collection.

Π.χ.

```
Dim colors As New Collection
```

Η προσθήκη αντικειμένων στην συλλογή γίνεται με την μέθοδο add. Η μέθοδος add δέχεται ως παράμετρο το στοιχείο το οποίο προστίθεται στην συλλογή.

Π.χ.

```
colors.Add("Λευκό")
```

Η μέθοδος add δέχεται προαιρετικά ως παράμετρο μια λέξη κλειδί (η οποία πρέπει να είναι μοναδική), τον δείκτη της θέσης πριν από την οποία θα προστεθεί το στοιχείο και τον δείκτη της θέσης μετά από τον οποίο θα προστεθεί το στοιχείο. Κατά τον ορισμό των προαιρετικών παραμέτρων δεν είναι απαραίτητο να ορισθούν όλες οι παράμετροι.

Για παράδειγμα η παρακάτω δήλωση προσθέτει ένα στοιχείο πριν την θέση 2.

```
colors.Add("Καφέ", , 2)
```

Επίσης η παρακάτω δήλωση προσθέτει ένα στοιχείο μετά την θέση 4.

```
colors.Add("Μαύρο", , , 4)
```

Το πλήθος των στοιχείων μιας συλλογής επιστρέφεται με την ιδιότητα Count.

Η μέθοδος remove διαγράφει ένα στοιχείο μιας συλλογής και δέχεται ως παράμετρο την θέση ή το κλειδί του στοιχείου.

Η μέθοδος clear διαγράφει όλα τα στοιχεία μιας συλλογής.

Η μέθοδος contains ελέγχει εάν η συλλογή περιλαμβάνει μια λέξη κλειδί.

Η απαρίθμηση όλων των στοιχείων μιας συλλογής είτε με την χρήση των δεικτών της συλλογής είτε με την χρήση της εντολής επανάληψης for .. each. Η εκτύπωση των στοιχείων μιας συλλογής με την χρήση των δεικτών γίνεται με το παρακάτω παράδειγμα. Οι δείκτες των στοιχείων μιας συλλογής ξεκινούν από τον αριθμό 1 σε αντίθεση με τους πίνακες οι οποίοι ξεκινούν από τον αριθμό 0.

```
Dim i As Integer  
For i = 1 To colors.Count  
    Console.WriteLine(colors.Item(i))  
Next
```

Εναλλακτικά μπορεί να χρησιμοποιηθεί η εντολή επανάληψης for .. each όπως φαίνεται στο παρακάτω παράδειγμα.

```
For Each c As String In colors  
    Console.WriteLine(c)  
Next
```

Στο παρακάτω παράδειγμα δημιουργείται μια συλλογή χρωμάτων με το όνομα colors, και δίνονται παραδείγματα χρήσης των κυριοτέρων ιδιοτήτων και μεθόδων της κλάσης Collection.

```
Sub Main()
    'Δημιουργία συλλογής δεδομένων
    Dim colors As New Collection
    'Πρόσθεση στοιχείων στην συλλογή
    colors.Add("Λευκό")
    colors.Add("Κόκκινο")
    colors.Add("Κίτρινο")
    colors.Add("Πράσινο")
    colors.Add("Μπλέ")
    'Προσθήκη στοιχείου πριν την θέση 2
    colors.Add("Καφέ", , 2)
    'Προσθήκη στοιχείου μετά την θέση 4
    colors.Add("Μαύρο", , , 4)
    'Πλήθος στοιχείων της συλλογής
    Console.WriteLine("Ο αριθμός των χρωμάτων είναι : " & _
        colors.Count)
    'Εκτύπωση των στοιχείων της συλλογής
    Console.WriteLine("Τα χρώματα είναι :")
    Dim i As Integer
    For i = 1 To colors.Count
        Console.WriteLine(colors.Item(i))
    Next
    'Διαγραφή στοιχείου συλλογής
    colors.Remove(3)
    'Εκτύπωση των στοιχείων της συλλογής
    Console.WriteLine("Τα χρώματα μετά την αφαίρεση είναι :")
    For Each c As String In colors
        Console.WriteLine(c)
    Next
    Console.ReadLine()
End Sub
```

Παράδειγμα 9.2 Χρήση συλλογής δεδομένων

Μια συλλογή μπορεί να περιλαμβάνει διάφορους τύπους δεδομένων ή αντικείμενων. Στο παρακάτω παράδειγμα ορίζεται η κλάση αντικείμενων PersonClass και μια συλλογή με το όνομα persons η οποία έχει ως στοιχεία της αντικείμενα τύπου PersonClass.

```
Class personClass
    Public id As String
    Public name As String
    Public surname As String

    Public Sub New(ByVal anId As String, _
        ByVal aName As String, ByVal aSurname As String)
        id = anId
        name = aName
        surname = aSurname
    End Sub
    'Επανακαθορισμός της μεθόδου toString
    Public Overrides Function toString() As String
        Return id + " " + name + " " + surname
    End Function
End Class
```

```

Sub Main()
    'Δημιουργία συλλογής αντικειμένων Persons
    Dim persons As New Collection
    'Δημιουργία και πρόσθεση αντικειμένων με λέξεις κλειδιά
    Dim p As personClass
    p = New personClass("111", "Γεώργιος", "Ιωάννου")
    persons.Add(p, p.id)
    p = New personClass("115", "Πέτρος", "Αναγνώστου")
    persons.Add(p, p.id)
    Console.WriteLine("Ο αριθμός ατόμων στην συλλογή είναι:" & persons.Count)
    'Εκτύπωση συλλογής
    For Each c As personClass In persons
        Console.WriteLine(c.toString)
    Next
    'Πρόσθεση στοιχείου πριν από λέξη κλειδί
    p = New personClass("112", "Δημήτρης", "Απέργης")
    persons.Add(p, p.id, "115")
    Dim i As Integer
    Console.WriteLine("Αριθμός ατόμων μετά την πρόσθεση:" & persons.Count)
    'Εκτύπωση συλλογής
    For i = 1 To persons.Count
        Dim c As personClass = persons(i)
        Console.WriteLine(c.toString())
    Next
    'Αναζήτηση στοιχείου με λέξη κλειδί
    Console.WriteLine("Αναζήτηση ατόμου με κωδικό 112")
    If persons.Contains("112") Then
        Console.WriteLine("Το άτομο βρέθηκε")
        Dim c As personClass = persons(p.id)
        Console.WriteLine(c.toString)
    Else
        Console.WriteLine("Το άτομο δεν βρέθηκε")
    End If
    'Αφαίρεση στοιχείου με λέξη κλειδί
    persons.Remove("112")
    Console.WriteLine("Αριθμός ατόμων μετά την αφαίρεση:" & persons.Count)
    'Εκτύπωση στοιχείων
    For i = 1 To persons.Count
        Dim c As personClass = persons(i)
        Console.WriteLine(c.toString())
    Next
    'Αναζήτηση στοιχείου
    Console.WriteLine("Αναζήτηση ατόμου με κωδικό 112")
    If persons.Contains(p.id) Then
        Console.WriteLine("Το άτομο βρέθηκε")
    Else
        Console.WriteLine("Το άτομο δεν βρέθηκε")
    End If
    'Διαγραφή όλων των στοιχείων
    persons.Clear()
    Console.WriteLine("Αριθμός ατόμων μετά τον καθαρισμό:" & persons.Count)
    Console.ReadLine()
End Sub

```

Παράδειγμα 9.3 Χρήση συλλογής αντικειμένων

Το παράδειγμα χρησιμοποιεί ως λέξη κλειδί για το κάθε στοιχείο της συλλογής των κωδικό του (id) και η προσθήκη στοιχείων γίνεται με παραμέτρους το αντικείμενο και τον κωδικό του:

```
persons.Add(p, p.id)
```

Επίσης, κατά την πρόσθεση ενός στοιχείου πριν από κάποια θέση, χρησιμοποιείται εναλλακτικά η λέξη κλειδί του στοιχείου.

```
'Πρόσθεση στοιχείου πριν από λέξη κλειδί  
p = New personClass("112", "Δημήτρης", "Απέργης")  
persons.Add(p, p.id, "115")
```

Σε περίπτωση που μια συλλογή σχετίζεται με λέξεις κλειδιά μπορούμε να χρησιμοποιήσουμε την μέθοδο contains για να ελέγξουμε ένα κάποια λέξη κλειδί υπάρχει στην συλλογή. Μπορούμε επίσης να χρησιμοποιήσουμε την λέξη κλειδί αντί του δείκτη για να αποκτήσουμε πρόσβαση στο αντικείμενο.

```
'Αναζήτηση στοιχείου με λέξη κλειδί  
If persons.Contains("112") Then  
    Console.WriteLine("Το άτομο βρέθηκε")  
    Dim c As personClass = persons(p.id)  
    Console.WriteLine(c.toString)  
Else  
    Console.WriteLine("Το άτομο δεν βρέθηκε")  
End If
```

9.5 Ασκήσεις Κεφαλαίου 9

Ασκηση 9.1

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_9_ask_1, αντιγράψτε την λειτουργική μονάδα του παραδείγματος 9.1 και εκτελέστε την εφαρμογή.

Τροποποιήστε τον ορισμό των ημερών όπως φαίνεται παρακάτω.

```
Enum DaysEnum As Integer
    Κυριακή
    Δευτέρα
    Τρίτη
    Τετάρτη = 40
    Πέμπτη
    Παρασκευή
    Σάββατο
    Αγνωστη
End Enum
```

Ποιες είναι οι τιμές οι οποίες αντιστοιχούν στις ημέρες τις εβδομάδας;

Ασκηση 9.2

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_9_ask_2 η οποία εκτυπώνει το τρέχον χρώμα υπόβαθρου της οθόνης τερματικού (Console.BackgroundColor).

Το πρόγραμμα στη συνέχεια εκτυπώνει τα ονόματα των δυνατών τιμών του τύπου δεδομένων ConsoleColor και ζητά από τον χρήστη να εισάγει έναν ακέραιο αριθμό. Εάν ο αριθμός αυτός αντιστοιχεί σε κάποιο υπαρκτό χρώμα (χρησιμοποιήστε την μέθοδο isDefined), τότε το χρώμα αυτό ορίζεται ως το χρώμα υπόβαθρου της οθόνης τερματικού.

Ασκηση 9.3

Δημιουργήστε εφαρμογή τερματικού (console) lab_9_ask_3 η οποία ορίζει απαριθμητό τύπο δεδομένων με όνομα monthsEnum και συμβολικές τιμές τα ονόματα των μηνών του έτους. Η εφαρμογή:

- Εκτυπώνει όλες τις συμβολικές τιμές του τύπου δεδομένων monthsEnum και τις αντίστοιχες αριθμητικές τους τιμές.
- Ορίζει μεταβλητή m τύπου monthsEnum, ζητά από το χρήστη να εισάγει το όνομα ενός μήνα, και αποθηκεύει την τιμή στην μεταβλητή m.
- Σε περίπτωση που το όνομα που εισάγει ο χρήστης δεν αντιστοιχεί σε κάποιον μήνα, τότε το πρόγραμμα εκτυπώνει όλα τα ονόματα των μηνών και ζητά από τον χρήστη να εισάγει ξανά το όνομα ενός μήνα έως ότου ο χρήστης πληκτρολογήσει το όνομα ενός υπαρκτού μήνα.

Ασκηση 9.4

Δημιουργήστε επίσης εφαρμογή τερματικού (console) lab_9_ask_4 αντιγράψτε την λειτουργική μονάδα του παραδείγματος 9.3 και εκτελέστε την εφαρμογή.

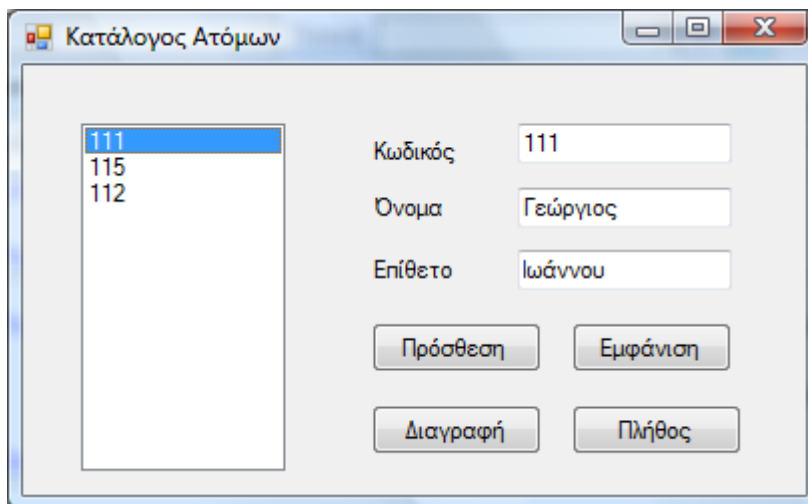
Τροποποιήστε την εφαρμογή ώστε να ορίζει απαριθμητό τύπο δεδομένων cityEnum με δυνατές τιμές Athens, Thessaloniki, Lamia, Patra και όποιες άλλες πόλεις επιθυμείτε.

Τροποποιήστε την κλάση αντικειμένων personClass ώστε να ορίζει πεδίο με όνομα city τύπου cityEnum. Τροποποιήστε την μέθοδο new ώστε να δέχεται ως παράμετρο το όνομα της πόλης ως κείμενο και την μέθοδο toString ώστε να εκτυπώνει και το όνομα της πόλης.

Τροποποιήστε επίσης την διαδικασία main ώστε να ορίζει και το όνομα της πόλης κατά την δημιουργία αντικειμένων τύπου personClass.

Άσκηση 9.5

Δημιουργήστε παραθυρική εφαρμογή lab_9_ask_5 η οποία εμφανίζει το παρακάτω παράθυρο με όνομα FormPerson και τίτλο Κατάλογος Ατόμων.



Η εφαρμογή ορίζει επίσης την κλάση αντικειμένων PersonClass του παραδείγματος 9.3 σε ξεχωριστό αρχείο με όνομα PersonClass.vb.

Η κλάση αντικειμένων FormPerson (το παράθυρο της εφαρμογής) καθορίζει και δημιουργεί καθολική μεταβλητή persons τύπου συλλογής με την δήλωση:

Public persons as new Collection.

Τα τρία πλαίσια κειμένου της εφαρμογής με ονόματα textBoxId, textBoxName και textBoxSurname χρησιμοποιούνται για την εισαγωγή και εμφάνιση των στοιχείων ενός ατόμου.

Με το πάτημα του πλήκτρου «Πρόσθεση» προστίθεται ένα νέο αντικείμενο τύπου personClass στην συλλογή με τιμές στα πεδία id, name και surname οι οποίες δίνονται από τις αντίστοιχες τιμές των πλαισίων κειμένου. Ως κλειδί του αντικειμένου που εισάγεται στην συλλογή δίνεται ο κωδικός του ατόμου (η ιδιότητα text του πλαισίου κειμένου textBoxId).

Με το πάτημα του πλήκτρου «Εμφάνιση» εμφανίζονται στην λίστα επιλογής οι κωδικοί των ατόμων της συλλογής. Με την επιλογή κάποιου κωδικού από την λίστα επιλογής, εμφανίζονται τα στοιχεία του ατόμου που αντιστοιχούν στον επιλεγμένο κωδικό.

Με το πάτημα του πλήκτρου «Διαγραφή» διαγράφεται το στοιχείο της συλλογής με τον επιλεγμένο κωδικό στην λίστα επιλογής και με το πάτημα του πλήκτρου «Πλήθος» εμφανίζεται μήνυμα με το πλήθος των στοιχείων της συλλογής. (π.χ. Πλήθος Ατόμων = 3)

10 Αρχεία

10.1 Εγγραφή και ανάγνωση δεδομένων σε αρχεία.

Υπάρχουν δύο είδη αρχείων, τα αρχεία κειμένου και τα δυαδικά αρχεία. Τα αρχεία κειμένου περιέχουν κείμενο και διαβάζονται συνήθως γραμμή προς γραμμή. Τα δυαδικά αρχεία περιέχουν αριθμητικά δεδομένα ή συνδυασμό κειμένου και αριθμητικών δεδομένων και διαβάζονται ανάλογα με τον τύπο και είδος δεδομένων τα οποία περιέχουν.

Για την προσπέλαση αρχείων πρέπει να ορισθεί ένα αντικείμενο της κλάσης `FileStream` και αναλόγως της επιθυμητής προσπέλασης (εγγραφή ή ανάγνωση) και του τύπου του αρχείου (αρχείο κειμένου ή δυαδικών δεδομένων) πρέπει να ορισθούν αντικείμενα των κλάσεων:

- `StreamReader` (για ανάγνωση αρχείου κειμένου)
- `StreamWriter` (για εγγραφή αρχείου κειμένου)
- `BinaryReader` (για ανάγνωση δυαδικού αρχείου)
- `BinaryWriter` (για εγγραφή δυαδικού αρχείου).

Η κλάση `FileStream` αναπαριστά μια ροή ή ακολουθία από δεδομένα και χρησιμοποιείται για την δημιουργία, και το άνοιγμα για εγγραφή ή ανάγνωση ενός αρχείου.

Η χρήση κλάσεων αντικειμένων για την ανάγνωση και εγγραφή σε αρχεία προϋποθέτει την χρήση της παρακάτω δήλωσης στην αρχή της λειτουργικής μονάδος ή της κλάσης αντικειμένων.

Imports System.IO

Η δημιουργία ενός αρχείου γίνεται με την δημιουργία ενός αντικειμένου της κλάσης `FileStream` με παραμέτρους το όνομα του αρχείου και την τιμή `FileMode.Create` για την παράμετρο `filemode`. Η παρακάτω εντολή δημιουργεί το αρχείο `data.txt` και συσχετίζει το αρχείο με την μεταβλητή `f`. Σε περίπτωση που το αρχείο υπάρχει, το αρχείο διαγράφεται και δημιουργείται ένα νέο.

'Δημιουργία αρχείου

```
Dim f As FileStream = New FileStream("data.txt", FileMode.Create)
```

Η ανάγνωση των δεδομένων ενός αρχείου το οποίο ήδη υπάρχει γίνεται με την δημιουργία ενός αντικειμένου της κλάσης `FileStream` με παραμέτρους το όνομα του αρχείου και την τιμή `FileMode.Open` για την παράμετρο `filemode`.

'Άνοιγμα αρχείου

```
Dim f As FileStream = New FileStream("data.txt", FileMode.Open)
```

Ο απαριθμητός τύπος δεδομένων `Filemode` ο οποίος δίνεται ως παράμετρος κατά την δημιουργία ενός αντικειμένου τύπου `FileStream` μπορεί να έχει τις παρακάτω τιμές:

- `Create`: Δημιουργεί ένα νέο αρχείο. Εάν το αρχείο ήδη υπάρχει το αρχείο διαγράφεται.
- `CreateNew`: Δημιουργεί ένα νέο αρχείο εφόσον το αρχείο δεν υπάρχει.
- `Open`: Ανοίγει ένα αρχείο το οποίο ήδη υπάρχει.
- `OpenOrCreate`: Ανοίγει ένα αρχείο. Εάν το αρχείο δεν υπάρχει δημιουργεί ένα νέο αρχείο.
- `Append`: Ανοίγει ένα αρχείο το οποίο υπάρχει για προσθήκη δεδομένων στο τέλος του αρχείου
- `Truncate`: Ανοίγει ένα αρχείο το οποίο ήδη υπάρχει και διαγράφει όλα τα δεδομένα του αρχείου.

10.2 Εγγραφή και ανάγνωση αρχείων κειμένου

Η εγγραφή δεδομένων σε ένα αρχείο κειμένου γίνεται με την δημιουργία αντικειμένων της κλάσης `StreamWriter`. Στο παρακάτω παράδειγμα δημιουργείται ένα αντικείμενο της κλάσης `StreamWriter` για εγγραφή στο αρχείο `data.txt` το οποίο σχετίζεται με την μεταβλητή `f`.

```
'Δημιουργία αντικειμένου StreamWriter για την εγγραφή στο αρχείο
Dim sw As StreamWriter = New StreamWriter(f)
```

Η εγγραφή κειμένου σε ένα αρχείο γίνεται με τις μεθόδους write και writeline οι οποίες είναι παρόμοιες με τις μεθόδους write και WriteLine της κλάσης αντικειμένων Console.

```
'Εγγραφή γραμμής κειμένου σε αρχείο
sw.WriteLine("Hello")
```

Μετά το τέλος της εγγραφής δεδομένων σε ένα αρχείο κειμένου, το αντικείμενο της κλάσης StreamWriter και το αντικείμενο της κλάσης FileStream πρέπει να κλείσουν με την μέθοδο close ώστε να αποθηκευθούν οι αλλαγές οι οποίες έγιναν στο αρχείο και να αποδεσμευτεί το αρχείο ώστε να είναι προσβάσιμο από άλλα προγράμματα.

```
'Κλείσιμο αντικειμένου StreamWriter
sw.Close()
'Κλείσιμο αρχείου
f.Close()
```

Το παρακάτω παράδειγμα δημιουργεί ένα αρχείο με όνομα data.txt, γράφει την λέξη “Hello” στο αρχείο και κλείνει το αρχείο.

```
Imports System.IO
Module Module1
    Sub Main()
        'Δημιουργία αρχείου κειμένου
        Dim f As FileStream = New FileStream("data.txt", _
                                           FileMode.Create)

        'Δημιουργία αντικειμένου StreamWriter για εγγραφή στο αρχείο
        Dim sw As StreamWriter = New StreamWriter(f)
        sw.WriteLine("Hello")
        'Κλείσιμο αντικειμένου StreamWriter
        sw.Close()
        'Κλείσιμο αρχείου
        f.Close()
    End Sub
End Module
```

Παράδειγμα 10.1 Εγγραφή σε αρχείο κειμένου

Το παρακάτω παράδειγμα ζητά από το χρήστη να εισάγει διαδοχικά κείμενο και το αποθηκεύει σε αρχείο κειμένου με το όνομα “data.txt” έως ότου ο χρήστης πληκτρολογήσει μια κενή γραμμή ως κείμενο.

```
Imports System.IO
Module Module1
    Sub Main()
        'Δημιουργία αρχείου κειμένου
        Dim f As FileStream = New FileStream("data.txt", _
                                           FileMode.Create)

        'Δημιουργία αντικειμένου StreamWriter για εγγραφή στο αρχείο
        Dim sw As StreamWriter = New StreamWriter(f)
        Dim s As String
        'Διάβασμα κειμένου από το πληκτρολόγιο και εγγραφή
        'σε αρχείο έως ότου πληκτρολογηθεί κενή γραμμή
        Do
            s = Console.ReadLine
            If s = "" Then Exit Do
            sw.WriteLine(s)
        Loop
        'Κλείσιμο αντικειμένου StreamWriter
```

```

        sw.Close()
        'Κλείσιμο αρχείου
        f.Close()
    End Sub
End Module

```

Παράδειγμα 10.2 Εγγραφή κειμένου από το πληκτρολόγιο σε αρχείο

Αντίστοιχα η ανάγνωση από αρχείο κειμένου γίνεται με αντικείμενα της κλάσης StreamReader και την μέθοδο readline. Το παρακάτω παράδειγμα ανοίγει ένα αρχείο με όνομα “data.txt” το οποίο υπάρχει ήδη, διαβάζει και εκτυπώνει την πρώτη γραμμή του αρχείου και κλείνει το αρχείο.

```

Imports System.IO
Module Module1
    Sub Main()
        'Ανοιγμα αρχείου κειμένου
        Dim f As FileStream = New FileStream("data.txt", _
                                            FileMode.Open)

        'Δημιουργία αντικειμένου StreamReader
        Dim sr As StreamReader = New StreamReader(f)
        Dim s As String
        s = sr.ReadLine
        Console.WriteLine(s)
        'Κλείσιμο αντικειμένου StreamReader
        sr.Close()
        'Κλείσιμο αρχείου
        f.Close()
    End Sub
End Module

```

Παράδειγμα 10.3 Ανάγνωση της πρώτης γραμμής από αρχείο κειμένου

Κάθε φορά γράφουμε μια γραμμή κειμένου σε κάποιο αρχείο, μετακινούμαστε στο τέλος του αρχείου, έτσι ώστε η επόμενη γραμμή να προστεθεί στο τέλος του αρχείου. Αντίστοιχα όταν διαβάζουμε μια γραμμή κειμένου, μετακινούμαστε στην αρχή της επόμενης γραμμής του αρχείου έως ότου φτάσουμε στο τέλος του αρχείου. Ο έλεγχος εάν βρισκόμαστε στο τέλος του αρχείου γίνεται με την μέθοδο EndOfStream του αντικειμένου της κλάσης StreamReader που χρησιμοποιούμε για την ανάγνωση από το αρχείο.

Το παρακάτω παράδειγμα διαβάζει και εκτυπώνει όλες τις γραμμές ενός αρχείου κειμένου κάνοντας χρήση της μεθόδου EndOfStream ως συνθήκη τερματισμού των επαναλήψεων για την ανάγνωση των γραμμών του αρχείου.

```

Imports System.IO
Module Module1
    Sub Main()
        'Ανοιγμα αρχείου κειμένου
        Dim f As FileStream = New FileStream("data.txt", _
                                            FileMode.Open)

        'Δημιουργία αντικειμένου StreamReader για
        'ανάγνωση από αρχείο
        Dim sr As StreamReader = New StreamReader(f)
        'Διάβασμα διαδοχικά γραμμών κειμένου και
        'εκτύπωση μέχρι το τέλος του αρχείου
        Dim s As String
        Do While Not sr.EndOfStream
            s = sr.ReadLine
            Console.WriteLine(s)
        Loop
        'Κλείσιμο αντικειμένου StreamReader
        sr.Close()
    End Sub
End Module

```

```

        'Κλείσιμο αρχείου
        f.Close()
        Console.ReadLine()
    End Sub
End Module

```

Παράδειγμα 10.4 Ανάγνωση και εκτύπωση αρχείου κειμένου

10.3 Εγγραφή και ανάγνωση δυαδικών αρχείων

Η εγγραφή δεδομένων σε δυαδικό αρχείο γίνεται με την χρήση της κλάσης αντικειμένων BinaryWriter αντί της κλάσης StreamWriter και της μεθόδου write αντί της μεθόδου writeLine. Το παρακάτω παράδειγμα γράφει τους ακεραίους από 1 έως 20 στο δυαδικό αρχείο 'data.txt'.

```

Imports System.IO
Module Module1
    Sub Main()
        'Δημιουργία δυαδικού αρχείου δεδομένων
        Dim f As FileStream = New FileStream("data.txt", _
                                           FileMode.Create)
        'Δημιουργία αντικειμένου StreamWriter για εγγραφή στο αρχείο
        Dim bw As BinaryWriter = New BinaryWriter(f)
        'Εγγραφή ακεραίων 1 έως 20 στο αρχείο
        For i As Integer = 1 To 20
            bw.Write(i)
        Next
        'Κλείσιμο αντικειμένου BinaryWriter
        bw.Close()
        'Κλείσιμο αρχείου
        f.Close()
    End Sub
End Module

```

Παράδειγμα 10.5 Εγγραφή ακεραίων σε αρχείο κειμένου

Η ανάγνωση από δυαδικό αρχείο δεδομένων γίνεται με την χρήση της κλάσης αντικειμένων BinaryReader και των μεθόδων read αναλόγως του τύπου δεδομένων (π.χ. readBoolean, readInt32, readSingle, readDouble, ReadString για μεταβλητές τύπου Boolean, Integer, Single, Double και String αντίστοιχα)

Ο έλεγχος εάν έχουν διαβασθεί όλα τα στοιχεία ενός αρχείου γίνεται με την χρήση των ιδιοτήτων position και length της κλάσης αντικειμένων FileStream οι οποίες επιστρέφουν την τρέχουσα θέση και το μήκος του αρχείου.

```

Imports System.IO
Module Module1
    Sub Main()
        'Άνοιγμα δυαδικού αρχείου
        Dim f FileStream = New FileStream("data.txt", FileMode.Open)
        'Δημιουργία αντικειμένου BinaryReader για ανάγνωση
        Dim br As BinaryReader = New BinaryReader(f)
        'Διάβασμα όσο ή τρέχουσα θέση του αρχείου
        'είναι μικρότερη από το μήκος του αρχείου
        Do While f.Position < f.Length
            'Διάβασμα ακεραίου 32 Bit με την μέθοδο readInt32
            Dim i As Integer = br.ReadInt32()
            Console.WriteLine(i)
        Loop
        'Κλείσιμο αντικειμένου BinaryReader
        br.Close()
    End Sub
End Module

```

```

        'Κλείσιμο αρχείου
        f.Close()
        Console.ReadLine()
    End Sub
End Module

```

Παράδειγμα 10.6 Ανάγνωση αρχείου ακεραίων

Η ιδιότητα position μπορεί επίσης να χρησιμοποιηθεί και για την μετακίνηση και την ανάγνωση ή τροποποίηση μιας προκαθορισμένης θέσης του αρχείου. Το παρακάτω παράδειγμα ανοίγει ένα υπάρχον δυαδικό αρχείο ακεραίων, μετακινείται στο τρίτο ακέραιο του αρχείου, αλλάζει την τιμή του, στην συνέχεια μετακινείται στην αρχή του αρχείου και εκτυπώνει τα περιεχόμενα του. Η θέση του τρίτου ακεραίου είναι η θέση 8 (Ο πρώτος ακέραιος βρίσκεται στην θέση 0, ο δεύτερος στην θέση 4 γιατί ο κάθε ακέραιος καταλαμβάνει 32 bits ή 4 bytes, ο τρίτος στην θέση 8, κ.λ.π.).

```

Imports System.IO
Module Module1
    Sub Main()
        'Ανοιγμα αρχείου
        Dim f As New FileStream("data.txt", FileMode.Open)
        'Δημιουργία αντικειμένου BinaryWriter για τροποποίηση
        Dim bw As BinaryWriter = New BinaryWriter(f)
        'Μετακίνηση στον τρίτο ακέραιο του αρχείου
        f.Position = 8
        'Αλλαγή της τιμής του σε 45
        bw.Write(45)
        'Μετακίνηση στην αρχή του αρχείου
        f.Position = 0
        'Εκτύπωση των περιεχομένων του αρχείου
        Dim br As BinaryReader = New BinaryReader(f)
        Do While f.Position < f.Length
            Dim i As Integer = br.ReadInt32()
            Console.WriteLine(i)
        Loop
        'Κλείσιμο του αρχείου
        bw.Close()
        br.Close()
        f.Close()
        Console.ReadLine()
    End Sub
End Module

```

Παράδειγμα 10.7 Τροποποίηση αρχείου ακεραίων

10.4 Εγγραφή και ανάγνωση αρχείων αντικειμένων

Η εγγραφή και ανάγνωση αρχείων αντικειμένων μπορεί να γίνει με την χρήση δυαδικών αρχείων δεδομένων.

Το παρακάτω παράδειγμα ορίζει μια κλάση αντικειμένων με κατάλληλες μεθόδους για εγγραφή και ανάγνωση από αρχείο, δημιουργεί αντικείμενα, τα αποθηκεύει σε αρχείο, τα ανακτά από το αρχείο και τα εκτυπώνει.

```

Imports System.IO

'Ορισμός κλάσης αντικειμένων
Public Class personClasss
    Public name As String
    Public surname As String

```

```

Public height As Double
Public Overrides Function ToString() As String
    Return name & " " & surname & " : " & height.ToString
End Function
'Ορισμός μεθόδου για εγγραφή σε αρχείο
Public Sub saveObject(ByVal bw As BinaryWriter)
    bw.Write(name)
    bw.Write(surname)
    bw.Write(height)
End Sub
'Ορισμός μεθόδου για ανάγνωση από αρχείο
Public Sub readObject(ByVal br As BinaryReader)
    name = br.ReadString()
    surname = br.ReadString()
    height = br.ReadDouble()
End Sub
End Class

Module module1
    Sub Main()
        'Δημιουργία αντικειμένων
        Dim p1 As New personClassss
        p1.name = "Γιώργος"
        p1.surname = "Ιωάννου"
        p1.height = 1.81

        Dim p2 As New personClassss
        p2.name = "Πέτρος"
        p2.surname = "Γεωργίου"
        p2.height = 1.82

        Dim p3 As New personClassss
        p3.name = "Νίκος"
        p3.surname = "Πέτρου"
        p3.height = 1.83

        'Αποθήκευση αντικειμένων σε αρχείο
        Dim f = New FileStream("data.txt", FileMode.Create)
        Dim bw As BinaryWriter = New BinaryWriter(f)
        p1.saveObject(bw)
        p2.saveObject(bw)
        p3.saveObject(bw)
        bw.Close()
        f.Close()

        'Ανάκτηση αντικειμένων σε αρχείο
        f = New FileStream("data.txt", FileMode.Open)
        Dim br As BinaryReader = New BinaryReader(f)
        Do While f.Position < f.Length
            Dim p As New personClassss
            p.readObject(br)
            Console.WriteLine(p.ToString)
        Loop
        br.Close()
        f.Close()
        Console.ReadLine()
    End Sub
End Module

```

Παράδειγμα 10.8 Εγγραφή και ανάγνωση αντικειμένων σε αρχείο

Στο παραπάνω παράδειγμα ορίζεται αντικείμενο τύπου `PersonClass` με πεδία `name`, `surname` και `height`. Η εγγραφή των στοιχείων ενός αντικειμένου γίνει με τον ορισμό μεθόδου `SaveObject` και η ανάγνωση των στοιχείων ενός αντικειμένου γίνεται με τον ορισμό μεθόδου `ReadObject`. Οι μέθοδοι `SaveObject` και `ReadObject` δέχονται ως παράμετρο αντικείμενα τύπου `BinaryWriter` και `BinaryReader` τα οποία χρησιμοποιούνται για την εγγραφή και ανάγνωση αντίστοιχα. Οι παραπάνω μέθοδοι γράφουν και διαβάζουν τα πεδία των αντικειμένων με την ίδια σειρά.

Η διαδικασία `main` ορίζει τρία αντικείμενα τύπου `PersonClass`, δημιουργεί αρχείο δυαδικών δεδομένων με το όνομα `'data.txt'`, δημιουργεί αντικείμενο `BinaryWriter` για το συγκεκριμένο αρχείο και κάνει χρήση της μεθόδου `SaveObject` των αντικειμένων για την αποθήκευση των στοιχείων τους στο αρχείο.

Στην συνέχεια η διαδικασία `main` ανοίγει το αρχείο `'data.txt'` για ανάγνωση, δημιουργεί αντικείμενο τύπου `BinaryReader` και χρησιμοποιεί την διαδικασία `ReadObject` για να διαβάσει τα δεδομένα ενός προσωρινού αντικειμένου `p` και να τα εκτυπώσει.

10.5 Εγγραφή και ανάγνωση δυαδικών αρχείων με την κλάση *BinaryFormatter*

Αντί των κλάσεων `BinaryReader` και `BinaryWriter` για την εγγραφή σε δυαδικά αρχεία, μπορεί να χρησιμοποιηθεί η κλάση `BinaryFormatter` η οποία έχει την δυνατότητα να διαβάζει δεδομένα, αντικείμενα, πίνακες και συλλογές αντικειμένων ή δεδομένων.

Για να υπάρχει πρόσβαση στην κλάση αυτή, θα πρέπει να χρησιμοποιηθεί ή παρακάτω δήλωση στην αρχή της λειτουργικής μονάδας ή της κλάσης αντικειμένων:

```
Imports System.Runtime.Serialization.Formatters.Binary
```

Η εγγραφή και ανάγνωση δεδομένων απαιτεί την δημιουργία ενός αντικειμένου της κλάσης `BinaryFormatter`.

```
'Δημιουργία αντικειμένου BinaryFormatter  
Dim bf As New BinaryFormatter()
```

Η εγγραφή δεδομένων γίνεται με την χρήση της μεθόδου `serialize` της κλάσης `BinaryFormatter` και η ανάγνωση τους με την μέθοδο `deserialize` οι οποίες δέχονται ως πρώτη παράμετρο το αντικείμενο τύπου `FileStream`.

```
'Εγγραφή ακεραίου σε αρχείο  
bf.Serialize(f, 13)
```

```
'Ανάγνωση ακεραίου από αρχείο  
dim n as integer  
n = bf.Deserialize(f)
```

Εγγραφή και ανάγνωση σε αρχείο με την κλάση BinaryFormatter

Το παρακάτω παράδειγμα γράφει και διαβάζει από δυαδικό αρχείο ακεραίους, κείμενο, δισδιάστατο πίνακα ακεραίων και μία συλλογή ονομάτων.

```
Imports System.IO  
Imports System.Runtime.Serialization.Formatters.Binary  
Module Module1  
    Sub main()  
        Dim bf As New BinaryFormatter()  
        'Δημιουργία δυαδικού αρχείου  
        Dim f As FileStream = New FileStream("data.txt", _  
                                           FileMode.Create)  
    End Sub  
End Module
```

```

'Αποθήκευση ακεραίου στο αρχείο
bf.Serialize(f, 12)
'Αποθήκευση κειμένου στο αρχείο
bf.Serialize(f, "Hello")
'Αποθήκευση πίνακα στο αρχείο
Dim m1(,) As Integer = {{1, 45}, {3, 56}, {2, 7}}
bf.Serialize(f, m1)
'Αποθήκευση συλλογής στο αρχείο
Dim names As New Collection
names.Add("Γιάννης")
names.Add("Γιώργος")
names.Add("Πέτρος")
bf.Serialize(f, names)
f.Close()

'Ανάγνωση δεδομένων από αρχείο
f = New FileStream("data.txt", FileMode.Open)
'Ανάγνωση ακεραίου από αρχείο
Dim n As Integer
n = bf.Deserialize(f)
Console.WriteLine(n)
'Ανάγνωση κειμένου από αρχείο
Dim s As String
s = bf.Deserialize(f)
Console.WriteLine(s)
'Ανάγνωση πίνακα από αρχείο
Dim m2(,) As Integer
m2 = bf.Deserialize(f)
For i = 0 To m2.GetUpperBound(0)
    For j = 0 To m2.GetUpperBound(1)
        Console.Write("{0,4}", m2(i, j))
    Next
    Console.WriteLine()
Next
'Ανάγνωση συλλογής από αρχείο
names.Clear()
names = bf.Deserialize(f)
For Each aName As String In names
    Console.WriteLine(aName)
Next
Console.ReadLine()
End Sub

```

Παράδειγμα 10.9 Εγγραφή και ανάγνωση σε αρχείο με την κλάση BinaryFormatter

Εγγραφή και ανάγνωση αντικειμένων σε αρχείο με την κλάση BinaryFormatter

Η κλάση BinaryFormatter μπορεί να χρησιμοποιηθεί για την εγγραφή και ανάγνωση αντικειμένων σε αρχεία. Στην περίπτωση αυτή, πρέπει η κλάση αντικειμένων θα πρέπει να δηλωθεί ως Serializable π.χ.

```

'Ορισμός κλάσης αντικειμένων
<Serializable()> Public Class person

```

Η εγγραφή αντικειμένων γίνεται με την χρήση της μεθόδου serialize της κλάσης BinaryFormatter και η ανάγνωση τους με την μέθοδο deserialize.

Το παρακάτω παράδειγμα αποθηκεύει και διαβάζει αντικείμενα τύπου PersonClass με την χρήση της κλάσης BinaryFormatter. Σε αντίθεση με το παράδειγμα 10.8 δεν χρειάζεται να ορισθούν οι μέθοδοι

SaveObject και ReadObject. Επίσης δεν χρειάζεται η δημιουργία αντικειμένων των κλάσεων BinaryReader και BinaryWriter.

```
Imports System.IO
Imports System.Runtime.Serialization.Formatters.Binary

'Ορισμός κλάσης αντικειμένων
<Serializable()> Public Class personClass
    Public name As String
    Public surname As String
    Public height As Double
    Public Overrides Function ToString() As String
        Return name & ": " & surname & " " & height.ToString
    End Function
End Class

Module Module1
    Sub Main()
        'Δημιουργία αντικειμένων
        Dim p1 As New person
        p1.name = "Γιώργος"
        p1.surname = "Ιωάννου"
        p1.height = 1.81

        Dim p2 As New person
        p2.name = "Πέτρος"
        p2.surname = "Γεωργίου"
        p2.height = 1.82

        Dim p3 As New person
        p3.name = "Νίκος"
        p3.surname = "Πέτρου"
        p3.height = 1.83

        'Αποθήκευση αντικειμένων σε αρχείο
        'Δημιουργία αντικειμένου BinaryFormatter
        Dim bf As New BinaryFormatter()
        'Δημιουργία δυαδικού αρχείου
        Dim f As FileStream = New FileStream("data.txt", _
                                           FileMode.Create)

        'Αποθήκευση αντικειμένων σε αρχείο
        bf.Serialize(f, p1)
        bf.Serialize(f, p2)
        bf.Serialize(f, p3)
        f.Close()

        'Ανάκτηση Αντικειμένων
        f = New FileStream("data.txt", FileMode.Open)
        Do While f.Position < f.Length
            Dim p As person
            'Ανάκτηση αντικειμένων από αρχείο
            p = bf.Deserialize(f)
            Console.WriteLine(p.ToString)
        Loop
        f.Close()
        Console.ReadLine()
    End Sub
End Module
```

Παράδειγμα 10.10 Εγγραφή και ανάγνωση αντικειμένων σε αρχείο με την κλάση BinaryFormatter

Εγγραφή και ανάγνωση συλλογής αντικειμένων από αρχείο

Το παρακάτω παράδειγμα χρησιμοποιεί μια συλλογή αντικειμένων της κλάσης PersonClass με το όνομα persons την οποία αποθηκεύει σε αρχείο, διαγράφει τα στοιχεία της συλλογής και στην συνέχεια διαβάζει τα στοιχεία της από αρχείο. Το παράδειγμα αυτό είναι αντίστοιχο με τα παραδείγματα 10.8 και 10.10 με την διαφορά ότι χρησιμοποιεί συλλογή αντικειμένων για να αποθηκεύσει και να διαβάσει τα αντικείμενα της κλάσης PersonClass.

```
Imports System.IO
Imports System.Runtime.Serialization.Formatters.Binary

'Ορισμός κλάσης αντικειμένων
<Serializable()> Public Class personClass
    Public name As String
    Public surname As String
    Public height As Double
    Public Overrides Function ToString() As String
        Return name & ": " & surname & " " & height.ToString
    End Function
End Class

Module Module1
    Sub Main()
        'Δημιουργία συλλογής αντικειμένων
        Dim persons As New Collection
        Dim p As personClass

        p = New personClass
        p.name = "Γιώργος"
        p.surname = "Ιωάννου"
        p.height = 1.81
        persons.Add(p)

        p = New personClass
        p.name = "Πέτρος"
        p.surname = "Γεωργίου"
        p.height = 1.82
        persons.Add(p)

        p = New personClass
        p.name = "Νίκος"
        p.surname = "Πέτρου"
        p.height = 1.83
        persons.Add(p)

        'Αποθήκευση αντικειμένων σε αρχείο
        'Δημιουργία αντικειμένου BinaryFormatter
        Dim bf As New BinaryFormatter()
        'Δημιουργία δυαδικού αρχείου
        Dim f As FileStream = New FileStream("data.txt", _
                                            FileMode.Create)

        'Αποθήκευση συλλογής σε αρχείο
        bf.Serialize(f, persons)
        f.Close()
```

```

'Ανάκτηση Αντικειμένων
f = New FileStream("data.txt", FileMode.Open)
persons.Clear()
persons = bf.Deserialize(f)
For Each p In persons
    Console.WriteLine(p.ToString)
Next
f.Close()
Console.ReadLine()
End Sub
End Module

```

Παράδειγμα 10.11 Εγγραφή και ανάγνωση συλλογής αντικειμένων

10.6 Εγγραφή και ανάγνωση αντικειμένων και συλλογών σε αρχεία XML

Η εγγραφή αντικειμένων και πινάκων αντικειμένων σε αρχεία XML μπορεί να γίνει με την χρήση της κλάσης `XmlSerializer` αντί της κλάσης `BinaryFormatter`. Η διαφορά τους είναι ότι η κλάση `XmlSerializer` μπορεί να γράψει και διαβάσει μόνο ένα αντικείμενο, ή πίνακα (όχι συλλογές) αντικειμένων σε αρχείο τύπου XML.

Για να υπάρχει πρόσβαση στην κλάση αυτή θα πρέπει να χρησιμοποιηθεί ή παρακάτω δήλωση στην αρχή της λειτουργικής μονάδας ή της κλάσης αντικειμένων:

```
Imports System.Xml.Serialization
```

Η εγγραφή και ανάγνωση δεδομένων απαιτεί την δημιουργία ενός αντικειμένου της κλάσης `XmlSerializer`. Η μέθοδος `new` απαιτεί ως παράμετρο τον τύπο των δεδομένων ο οποίος δίνεται με την μέθοδο `GetType`.

```

'Δημιουργία αντικειμένου XmlSerializer
Dim xs As New XmlSerializer(m1.GetType)

```

Η εγγραφή δεδομένων γίνεται με την χρήση της μεθόδου `serialize` της κλάσης `XmlSerializer` και η ανάγνωση τους με την μέθοδο `deserialize`.

```

'Αποθήκευση πίνακα αντικειμένων σε αρχείο
xs.Serialize(f, m1)

```

```

'Ανάγνωση πίνακα αντικειμένων
m2 = xs.Deserialize(f)

```

Το παρακάτω παράδειγμα ορίζει πίνακες `m1` και `m2` αντικειμένων `personClass`, ορίζει τα στοιχεία του πίνακα `m1`, αποθηκεύει τα στοιχεία του πίνακα `m1` σε αρχείο XML, διαβάζει τα στοιχεία του πίνακα `m2` από αρχείο XML και εκτυπώνει τα στοιχεία.

```

Imports System.IO
Imports System.Xml.Serialization

'Ορισμός κλάσης αντικειμένων
<Serializable()> Public Class personClass
    Public name As String
    Public surname As String
    Public height As Double
    Public Overrides Function ToString() As String
        Return name & ": " & surname & " " & height.ToString
    End Function
End Class

```

```

Module Module1
    Sub Main()
        'Ορισμός πινάκων αντικειμένων m1 και m2
        Dim m1(2) As personClass
        Dim m2(2) As personClass
        Dim p As personClass

        p = New personClass
        p.name = "Γιώργος"
        p.surname = "Ιωάννου"
        p.height = 1.81
        m1(0) = p

        p = New personClass
        p.name = "Πέτρος"
        p.surname = "Γεωργίου"
        p.height = 1.82
        m1(1) = p

        p = New personClass
        p.name = "Νίκος"
        p.surname = "Πέτρου"
        p.height = 1.83
        m1(2) = p

        'Δημιουργία αντικειμένου XmlSerializer
        Dim xs As New XmlSerializer(m1.GetType)
        'Δημιουργία δυαδικού αρχείου
        Dim f As FileStream = New FileStream("data.txt", _
                                            FileMode.Create)

        'Αποθήκευση πίνακα αντικειμένων σε αρχείο
        xs.Serialize(f, m1)
        f.Close()

        'Άνοιγμα αρχείου
        f = New FileStream("data.txt", FileMode.Open)
        'Ανάγνωση πίνακα αντικειμένων
        m2 = xs.Deserialize(f)
        For Each p In m2
            Console.WriteLine(p.ToString)
        Next
        f.Close()
        Console.ReadLine()
    End Sub
End Module

```

Παράδειγμα 10.12 Εγγραφή και ανάγνωση πίνακα αντικειμένων σε αρχείο XML

10.7 Ασκήσεις Κεφαλαίου 10

Ασκηση 10.1

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_10_ask_1a, η οποία ζητά από τον χρήστη να εισάγει διαδοχικά κείμενο έως ότου να πληκτρολογηθεί μια κενή γραμμή. Το κείμενο το οποίο εισάγεται από τον χρήστη αποθηκεύεται σε **αρχείο κειμένου** με το όνομα 'data.txt'.

Δημιουργήστε εφαρμογή τερματικού με όνομα lab_10_ask_1b, η οποία διαβάζει τα περιεχόμενα του αρχείου κειμένου που δημιουργήθηκε με την εφαρμογή lab_10_ask_1a, εκτυπώνει τα περιεχόμενα του αρχείου και εκτυπώνει το πλήθος των γραμμών του αρχείου.

Υπόδειξη: Χρησιμοποιήστε τα παραδείγματα 10.2 και 10.4 και μετρήστε το πλήθος των επαναλήψεων για να βρείτε το πλήθος των γραμμών του αρχείου.

Ασκηση 10.2

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_10_ask_2a, η οποία ζητά από τον χρήστη να εισάγει διαδοχικά πραγματικούς αριθμούς έως ότου να πληκτρολογηθεί ο αριθμός 0 και τους αποθηκεύει σε **δυναμικό αρχείο** με το όνομα 'data.txt'.

Δημιουργήστε εφαρμογή τερματικού με όνομα lab_10_ask_2b, η οποία διαβάζει τα περιεχόμενα του δυναμικού αρχείου που δημιουργήθηκε με την εφαρμογή lab_10_ask_2a και εκτυπώνει τους πραγματικούς αριθμούς που περιέχει το αρχείο και το άθροισμα τους.

Υπόδειξη: Χρησιμοποιήστε τα παραδείγματα 10.5 και 10.6 και τις κλάσεις αντικειμένων BinaryReader και BinaryWriter

Ασκηση 10.3

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_10_ask_3a, η οποία ορίζει πίνακα πραγματικών αριθμών 3X4, διαβάζει τα στοιχεία του πίνακα και τα αποθηκεύει σε δυναμικό αρχείο με όνομα 'data.txt'.

Δημιουργήστε εφαρμογή τερματικού με όνομα lab_10_ask_3b, η οποία ορίζει πίνακα πραγματικών αριθμών 3X4, διαβάζει τα στοιχεία του πίνακα από το δυναμικό αρχείο που δημιουργήθηκε με την εφαρμογή lab_10_ask_3a, και εκτυπώνει τον πίνακα, το μέσο όρο του και το μεγαλύτερο στοιχείο του.

Υπόδειξη: Χρησιμοποιήστε το παράδειγμα 10.9 και την κλάση αντικειμένων BinaryFormatter

Ασκηση 10.4

Δημιουργήστε εφαρμογή τερματικού (console application) με όνομα lab_10_ask_4, η οποία τροποποιεί την άσκηση 9.5. Η εφαρμογή εμφανίζει παράθυρο με όνομα FormPerson και τίτλο Κατάλογος Ατόμων.

Η εφαρμογή ορίζει επίσης την κλάση αντικειμένων PersonClass του παραδείγματος 9.3 σε ξεχωριστό αρχείο με όνομα PersonClass.vb.

Τα τρία πλαίσια κειμένου της εφαρμογής με ονόματα textBoxId, textBoxName και textBoxSurname χρησιμοποιούνται για την

εισαγωγή και εμφάνιση των στοιχείων ενός ατόμου. Η λίστα επιλογής με όνομα listBoxPersons εμφανίζει τους κωδικούς των ατόμων.

Τα πλήκτρα «Πρόσθεση», «Διαγραφή», «Εμφάνιση», «Πλήθος», «Αποθήκευση» και «Ανάγνωση» έχουν αντίστοιχα τα ονόματα ButtonAdd, ButtonDelete, ButtonShow, ButtonCount, ButtonSave και ButtonRead.

Αντιγράψτε αρχικά τον παρακάτω κώδικα για το παράθυρο με όνομα FormPerson.

```
Public Class FormPerson

    Public persons As New Collection()

    Private Sub ButtonAdd_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles ButtonAdd.Click
        Dim p As PersonClass
        p = New PersonClass(TextBoxId.Text, TextBoxName.Text, TextBoxSurname.Text)
        persons.Add(p, p.id)
    End Sub

    Private Sub ButtonShow_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles ButtonShow.Click
        ListBoxPersons.Items.Clear()
        For Each c As PersonClass In persons
            ListBoxPersons.Items.Add(c.id)
        Next
    End Sub

    Private Sub ListBoxPersons_SelectedIndexChanged(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles ListBoxPersons.SelectedIndexChanged
        If ListBoxPersons.SelectedIndex > -1 Then
            TextBoxId.Text = persons(ListBoxPersons.SelectedItem).id
            TextBoxName.Text = persons(ListBoxPersons.SelectedItem).name
            TextBoxSurname.Text = persons(ListBoxPersons.SelectedItem).surname
        End If
    End Sub

    Private Sub ButtonDelete_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles ButtonDelete.Click
        If ListBoxPersons.SelectedIndex > -1 Then
            persons.Remove(ListBoxPersons.SelectedItem)
        End If
    End Sub

    Private Sub ButtonCount_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles ButtonCount.Click
        MessageBox.Show("Πλήθος = " + persons.Count.ToString)
    End Sub
End Class
```

Πρόσθεστε στην συνέχεια τις κατάλληλες μεθόδους ώστε με το πάτημα του πλήκτρου ButtonSave να αποθηκεύεται η συλλογή persons σε αρχείο με όνομα 'data.txt' και με το πάτημα του πλήκτρου ButtonRead να διαβάζεται από το αρχείο 'data.txt'.

Υπόδειξη: Χρησιμοποιήστε το παραδείγματα 10.11 και την κλάση αντικειμένων BinaryFormatter για την εγγραφή και ανάγνωση της συλλογής αντικειμένων. Προσθέστε επίσης την δήλωση <serializable> στην δήλωση της κλάσης αντικειμένων PersonClass.

11 Γραφικά

11.1 Η κλάση Bitmap

Τα γραφικά των Windows βασίζονται σε ένα περιβάλλον σχεδίασης γραφικών το οποίο ονομάζεται GDI (Graphics Design Interface). Το GDI είναι μια συλλογή κλάσεων για την σχεδίαση γραφικών, κειμένων και εικόνων.

Η βασικότερη από τις κλάσεις του GDI είναι η κλάση Bitmap η οποία αναπαριστά μια εικόνα του H/Y. Οι εικόνες είναι διδιάστατοι πίνακες οι οποίες αποθηκεύουν τις χρωματικές τιμές των pixels (φωτεινών σημείων) από τα οποία απαρτίζεται μια εικόνα.

Η σχεδίαση γραφικών γίνεται συνήθως με την χρήση του στοιχείου ελέγχου PictureBox (πλαίσιο εικόνας). Τα αντικείμενα της κλάσης PictureBox έχουν την ιδιότητα Image η οποία μπορεί να πάρει ως τιμή ένα αντικείμενο της κλάσης Bitmap. Εκτός του στοιχείου ελέγχου PictureBox, υπάρχουν και άλλα στοιχεία ελέγχου (πχ. Button, Label, κλπ) τα οποία έχουν την ιδιότητα Image και μπορούν να χρησιμοποιηθούν για την σχεδίαση γραφικών. Σχεδίαση γραφικών μπορεί να γίνει επίσης και πάνω στα παράθυρα των εφαρμογών με την χρήση της ιδιότητας BackgroundImage.

Οι ιδιότητες image και backgroundImage μπορούν να καθορισθούν κατά την σχεδίαση μιας εφαρμογής ή να ορισθούν και τροποποιηθούν δυναμικά με εντολές σχεδίασης κατά την εκτέλεση της εφαρμογής.

Ένα αντικείμενο της κλάσης Bitmap μπορεί να δημιουργηθεί με την εντολή new η οποία δέχεται ως παραμέτρους το πλάτος και το ύψος της εικόνας σε pixels.

Όρισμός και δημιουργία ενός Bitmap
Dim bmp As Bitmap = New Bitmap(200, 300)

Ένα αντικείμενο Bitmap μπορεί να δημιουργηθεί επίσης από κάποια εικόνα η οποία είναι αποθηκευμένη σε κάποιο αρχείο.

Διάβασμα εικόνας από αρχείο
Dim bmp As Bitmap = Bitmap.FromFile("c:\dock.jpg")

Η απεικόνιση ενός αντικειμένου Bitmap σε κάποιο στοιχείο ελέγχου ή παράθυρο γίνεται με την απόδοση του αντικειμένου ως τιμής της ιδιότητας image ή backgroundImage αντίστοιχα.

Συσχέτιση Bitmap με την εικόνα στοιχείου ελέγχου
bmp = PictureBox1.Image

Επίσης, σε περίπτωση που επιθυμούμε να συνεχίσουμε την σχεδίαση γραφικών και όχι να δημιουργήσουμε ένα νέο αντικείμενο σχεδίασης, μπορούμε να ορίσουμε ένα αντικείμενο Bitmap και να το συσχετίσουμε (χωρίς να το δημιουργήσουμε) με την ιδιότητα Image του στοιχείου ελέγχου όπου γίνεται η σχεδίαση ή την ιδιότητα BackgroundImage του παραθύρου της εφαρμογής.

Σχεδίαση πάνω σε υπάρχουσα εικόνα στοιχείου ελέγχου
Dim bmp As Bitmap = PictureBox1.Image

Ο τρόπος με τον οποίο εμφανίζεται μια εικόνα σε ένα στοιχείο ελέγχου PictureBox καθορίζεται από την τιμή της ιδιότηταςSizeMode η οποία μπορεί να πάρει τις παρακάτω τιμές:

- **AutoSize:** Οι διαστάσεις του PictureBox προσαρμόζονται αυτόματα στο μέγεθος της εικόνας
- **CenterImage:** Η εικόνα τοποθετείται στο κέντρο (εφόσον είναι μικρότερη από το PictureBox)
- **NormalImage:** Δεν γίνεται καμία προσαρμογή μεγεθών
- **StretchImage:** Η εικόνα τοποθετείται έτσι ώστε να καλύψει ολόκληρο το PictureBox
- **Zoom:** Η εικόνα μεγεθύνεται χωρίς να αλλάξει η αναλογία πλάτους προς ύψος

Η περιστροφή και αναστροφή ενός Bitmap γίνεται με την μέθοδο RotateFlip. Η μέθοδος αυτή δέχεται ως παράμετρο τον τύπο της περιστροφής. Η περιστροφή μπορεί να είναι 90, 180 και 270 μοιρών. Η αναστροφή της εικόνας μπορεί να γίνει ως προς τον οριζόντιο ή τον κάθετο άξονα ή και ως προς τους δύο άξονες ταυτόχρονα. Ο απαριθμητής τύπος δεδομένων RotateFlipType καθορίζει τους τύπους περιστροφής, αναστροφής και τους δυνατούς συνδυασμούς τους.

Το παρακάτω παράδειγμα χρησιμοποιεί την μέθοδο RotateFlip για περιστρέψει μια εικόνα κατά 90 μοίρες.

```
Dim bmp As Bitmap = PictureBox1.Image
bmp.RotateFlip(RotateFlipType.Rotate90FlipNone)
PictureBox1.Image = bmp
```

Παράδειγμα 11.3 Περιστροφή εικόνας κατά 90 μοίρες

Η περιστροφή μπορεί να γίνει εναλλακτικά με την χρήση ως παραμέτρου της μεθόδου RotateFlip της ιδιότητας Image του PictureBox1. Για την απεικόνιση της εικόνας μετά την περιστροφή πρέπει να χρησιμοποιηθεί η μέθοδος Invalidate όπως φαίνεται στο παρακάτω παράδειγμα.

```
PictureBox1.Image.RotateFlip(RotateFlipType.Rotate90FlipNone)
PictureBox1.Invalidate()
```

Η αποθήκευση ενός Bitmap σε αρχείο γίνεται με την μέθοδο Save της κλάσης Bitmap. Η μέθοδος αυτή δέχεται ως παράμετρο το όνομα του αρχείου και τον τύπο του (π.χ. jpeg, tiff, gif, κλπ).

```
Dim bmp As Bitmap = PictureBox1.Image
bmp.Save("tmpimage.jpg", Imaging.ImageFormat.Jpeg)
```

Παράδειγμα 11.4 Αποθήκευση εικόνας σε αρχείο

11.2 Σχεδίαση γραφικών με την κλάση Graphics

Η κλάση Graphics χρησιμοποιείται για τον σχεδιασμό γραφικών σε εικόνες (αντικείμενα της κλάσης Bitmap). Ο σχεδιασμός γίνεται με την δημιουργία ενός αντικειμένου της κλάσης Graphics και την χρήση των μεθόδων σχεδίασης που διαθέτει το αντικείμενο Graphics.

Αναλόγως της μεθόδου σχεδίασης, απαιτείται και η χρήση μεταβλητών των σύνθετων δομών Point, Size και Rectangle και αντικειμένων των κλάσεων Pen και Brush.

Η σύνθετη δομή Point

Η σύνθετη δομή Point αναπαριστά ένα σημείο της επιφάνειας σχεδίασης. Μια μεταβλητή της δομής αυτής δημιουργείται με παραμέτρους τις συντεταγμένες x και y του σημείου.

```
'Δημιουργία σημείου με συντεταγμένες (100,100)
Dim p1 As Point = New Point(100, 100)
```

Η σύνθετη δομή Size

Η σύνθετη δομή Size αναπαριστά τις διαστάσεις μια περιοχής της επιφάνειας σχεδίασης (συνήθως ορθογωνίου). Μια μεταβλητή της δομής αυτής δημιουργείται με παραμέτρους το πλάτος και το ύψος του σχήματος.

```
'Δημιουργία διαστάσεων σχήματος
Dim s1 As Size = New Size(100, 100)
```

Η σύνθετη δομή Rectangle

Η σύνθετη δομή Rectangle αναπαριστά μια ορθογώνια περιοχή της επιφάνειας σχεδίασης. Η δημιουργία μιας μεταβλητής Rectangle γίνεται με παραμέτρους τις συντεταγμένες x και y της πάνω αριστερής γωνίας του ορθογωνίου και του πλάτους και ύψους του ορθογωνίου.

```
'Δημιουργία ορθογωνίου με χρήση συντεταγμένων
Dim r1 As Rectangle = New Rectangle(100, 100, 20, 10)
```

Εναλλακτικά η δημιουργία μιας μεταβλητής τύπου Rectangle γίνεται με παραμέτρους το πάνω αριστερό σημείο του ορθογωνίου και των διαστάσεων του.

```
'Δημιουργία ορθογωνίου με χρήση σημείου και διαστάσης
Dim p1 as New Point(100,100)
Dim s1 as New Size(20,10)
Dim r1 As Rectangle = New Rectangle(p1, s1)
```

Η κλάση Pen

Η κλάση Pen αναπαριστά ένα εικονικό μολύβι ή πένα. Η δημιουργία ενός αντικειμένου τύπου Pen γίνεται με τον καθορισμό του χρώματος και του πάχους του.

```
'Δημιουργία αντικειμένου Pen
Dim p As Pen = New Pen(Color.Blue, 2)
```

Η visual basic διαθέτει μια μεγάλη συλλογή με όνομα Pens η οποία περιέχει εικονικά μολύβια διαφόρων χρωμάτων και πάχους 1 pixel. (π.χ. Pens.Blue, Pens.Black, κλπ) τα οποία μπορούν να χρησιμοποιηθούν, χωρίς να δημιουργηθεί ένα νέο αντικείμενο.

Ένα αντικείμενο τύπου Pen έχει τις παρακάτω βασικές ιδιότητες:

- **StartCap** και **EndCap**: Καθορίζουν το σχήμα των άκρων αρχής και τέλους των γραμμών σχεδίασης (π.χ. Round, Square, Flat, Diamond, κλπ).
- **DashStyle**: Καθορίζει τον τύπο της διακεκομμένης γραμμής (Solid, Dash, DashDot, DashDotDot, Dot και Custom)
- **DashCap**: Καθορίζει το σχήμα της διακεκομμένης γραμμής (Flat, Round, και Triangle)

Το παρακάτω παράδειγμα ορίζει μια εικονική πένα με μαύρο χρώμα, πάχος γραμμής 5 pixels, καθορίζει τα σχήμα αρχής και τέλους των γραμμών, και καθορίζει επίσης τον τύπο και το σχήμα της διακεκομμένης γραμμής.

```
'Ορισμός εικονικής διακεκομμένης πένας μαύρου χρώματος
'με καθορισμό πάχους, και σχήματος αρχής, τέλους
'και σχήματος των διακεκομμένων σημείων της γραμμής
Dim p As Pen = New Pen(Color.Black, 5)
'Ορισμός σχήματος αρχής
p.StartCap = Drawing2D.LineCap.RoundAnchor
'Ορισμός σχήματος τέλους
p.EndCap = Drawing2D.LineCap.ArrowAnchor
'Ορισμός τύπου διακεκομμένης γραμμής
p.DashStyle = Drawing2D.DashStyle.Dot
'Ορισμός σχήματος διακεκομμένης γραμμής
p.DashCap = Drawing2D.DashCap.Round
```

Οι κλάσεις Brush

Οι κλάσεις SolidBrush, HatchBrush, LinearGradientBrush, PathGradientBrush και TextureBrush είναι κλάσεις οι οποίες χρησιμοποιούνται για το γέμισμα (χρωματισμό) σχημάτων και αναπαριστούν εικονικά πινέλα.

Για την χρήση των κλάσεων Brush απαιτείται η χρήση της βιβλιοθήκης Drawing2D με την παρακάτω δήλωση η οποία πρέπει να γίνει στην αρχή της λειτουργικής μονάδας ή της κλάσης αντικειμένων.

Imports System.Drawing.Drawing2D

Η κλάση SolidBrush

Η κλάση SolidBrush γεμίζει ένα σχήμα με συμπαγές χρώμα. Ένα αντικείμενο της κλάσης αυτής δημιουργείται με παράμετρο το χρώμα γεμίσματος.

```
'Δημιουργία συμπαγούς πινέλου με κόκκινο χρώμα
Dim redBrush As SolidBrush
redBrush = New SolidBrush(Color.Red)
```

Εναλλακτικά μπορεί να χρησιμοποιηθεί κάποιο πινέλο από την συλλογή Brushes της visual basic (π.χ. Brushes.Red).

Η κλάση HatchBrush

Η κλάση HatchBrush γεμίζει ένα σχήμα με διαγραμμισμένο μοτίβο. Η δημιουργία ενός αντικειμένου της κλάσης αυτής δέχεται ως παραμέτρους τον τύπο της διαγράμμισης, το χρώμα διαγράμμισης και το χρώμα υπόβαθρου.

```
'Δημιουργία πινέλου διαγράμμισης
Dim hBrush As HatchBrush
hBrush = New HatchBrush(HatchStyle.ForwardDiagonal, _
    Color.Black, Color.White)
```

Ο απαριθμητός τύπος δεδομένων HatchStyle παρέχει ένα μεγάλο πλήθος τύπων διαγράμμισης όπως Vertical, Horizontal, Cross, DiagonalCross, ForwardDiagonal, BackwardDiagonal, Percent20, κλπ.

Η κλάση LinearGradientBrush

Η κλάση LinearGradientBrush χρησιμοποιείται για το γέμισμα σχημάτων με γραμμικό ντεγκραντέ. Για την δημιουργία ενός αντικειμένου της κλάσης αυτής απαιτείται ο καθορισμός του ορθογωνίου όπου θα εφαρμοσθεί ντεγκραντέ, το αρχικό χρώμα, το τελικό χρώμα και ο τύπος γραμμικού ντεγκραντέ. Ο τύπος γραμμικού ντεγκραντέ μπορεί να έχει τις τιμές horizontal, vertical, BackwardDiagonal και ForwardDiagonal οι οποίες αντιστοιχούν σε οριζόντιο, κάθετο, διαγώνιο από την πάνω δεξιά γωνία και διαγώνιο από την πάνω αριστερή γωνία γέμισμα ντεγκραντέ.

```
'Δημιουργία πινέλου γραμμικού ντεγκραντέ γεμίσματος
Dim rect As Rectangle = New Rectangle(250, 30, 300, 100)
Dim lgBrush As LinearGradientBrush
lgBrush = New LinearGradientBrush(rect, Color.Blue, _
    Color.Red, LinearGradientMode.ForwardDiagonal)
```

Η κλάση PathGradientBrush

Η κλάση PathGradientBrush χρησιμοποιείται για το γέμισμα σχημάτων με ντεγκραντέ διαδρομής. Το ντεγκραντέ διαδρομής ξεκινά από ένα σημείο και διαβαθμίζεται σε πολλά διαφορετικά χρώματα προς διαφορετικές κατευθύνσεις. Η δημιουργία ενός αντικειμένου της κλάσης αυτής δέχεται ως παράμετρο ενός πίνακα σημείων οι οποίες είναι οι κατευθύνσεις προς τις οποίες θα γίνει το ντεγκραντέ διαδρομής. Στην συνέχεια θα πρέπει να καθοριστεί η ιδιότητα SurroundColors ως ένας πίνακας χρωμάτων ο οποίος αντιστοιχεί στο χρώμα των σημείων. Επίσης πρέπει να δοθούν τιμές στις ιδιότητες CenterColor και CenterPoint οι οποίες καθορίζουν το χρώμα και τις συντεταγμένες του κεντρικού σημείου του ντεγκραντέ διαδρομής.

```

'Δημιουργία ντεγκραντέ διαδρομής
'Ορισμός πίνακα με 3 σημεία
Dim gPoints() As Point = {New Point(500, 200), _
                           New Point(550, 300), _
                           New Point(450, 300)}
'Ορισμός και δημιουργία αντικειμένου ντεγκραντέ διαδρομής
Dim pgBrush As PathGradientBrush
pgBrush = New PathGradientBrush(gPoints)
'Δημιουργία πίνακα τριών χρωμάτων
Dim sColors() As Color = {Color.Red, _
                           Color.Green, Color.Blue}
pgBrush.SurroundColors = sColors
'Καθορισμός χρώματος και συντεταγμένων κεντρικού σημείου
pgBrush.CenterColor = Color.White
pgBrush.CenterPoint = New Point(500, 266)

```

Η κλάση TextureBrush

Η κλάση TextureBrush χρησιμοποιείται για το γέμισμα σχημάτων με μια συγκεκριμένη υφή (εικόνα). Η δημιουργία ενός αντικειμένου της κλάσης αυτής δέχεται ως παράμετρο ένα αντικείμενο τύπου Bitmap.

Το παρακάτω παράδειγμα δημιουργεί ένα πινέλο υφής με βάση την εικόνα η οποία είναι αποθηκευμένη στο αρχείο “c:\dock.jpg”

```

'Δημιουργία πινέλου υφής
Dim tBmp As Bitmap = Bitmap.FromFile("c:\dock.jpg")
Dim tBrush As TextureBrush
tBrush = New TextureBrush(tBmp)

```

Μέθοδοι σχεδίασης της κλάσης Graphics

Η σχεδίαση γραφικών γίνεται με την δημιουργία ενός αντικειμένου Bitmap και ενός αντικειμένου graphics για την σχεδίαση πάνω στο αντικείμενο Bitmap. Η δημιουργία του αντικειμένου Graphics γίνεται με την μέθοδο FromImage της κλάσης Graphics με παράμετρο το αντικείμενο Bitmap. Μετά το τέλος των εντολών σχεδίασης, πρέπει να δοθεί το αντικείμενο Bitmap ως τιμή της ιδιότητας image του στοιχείου ελέγχου ή της ιδιότητας backgroundImage του παραθύρου όπου γίνεται η σχεδίαση ώστε να εμφανισθεί η εικόνα.

Το παρακάτω παράδειγμα δημιουργεί ένα αντικείμενο Bitmap με πλάτος και ύψος το πλάτος και ύψος ενός PictureBox με όνομα PictureBox1, δημιουργεί ένα αντικείμενο Graphics για την σχεδίαση στο αντικείμενο Bitmap και τέλος συσχετίζει το αντικείμενο Bitmap με την ιδιότητα image του PictureBox1.

```

'Δημιουργία αντικειμένου Bitmap με διαστάσεις
'το πλάτος και το ύψος ενός PictureBox
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
'Δημιουργία αντικειμένου της κλάσης Graphics
Dim g As Graphics = Graphics.FromImage(bmp)

'Εντολές σχεδίασης

'Συσχέτιση αντικειμένου Bitmap με PictureBox1
'για την εμφάνιση της εικόνας
PictureBox1.Image = bmp

```

Οι βασικότερες από τις μεθόδους για την σχεδίαση σχημάτων είναι οι παρακάτω:

- Clear: Καθαρίζει και ορίζει το χρώμα υπόβαθρου
- DrawLine: Σχεδιάζει μία γραμμή
- DrawLines: Σχεδιάζει πολλές γραμμές
- DrawPolygon και FillPolygon: Σχεδιάζουν και γεμίζουν πολύγωνα αντίστοιχα
- DrawRectangle και FillRectangle: Σχεδιάζουν και γεμίζουν ορθογώνια αντίστοιχα
- DrawEllipse και FillEllipse: Σχεδιάζουν και γεμίζουν ελλείψεις αντίστοιχα
- DrawCurve: Σχεδιάζει μια καμπύλη
- DrawClosedCurve και FillClosedCurve: Σχεδιάζουν και γεμίζουν κλειστές καμπύλες
- DrawBezier: Σχεδιάζει μια καμπύλη Bezier
- DrawArc: Σχεδιάζει ένα τόξο
- DrawPie και FillPie: Σχεδιάζουν και γεμίζουν πίτες (για χρήση σε pie charts)
- DrawString: Εκτυπώνει κείμενο
- DrawImage: Εκτυπώνει μια εικόνα ή τμήμα της σε σημείο ή περιοχή της επιφάνειας σχεδίασης

Η μέθοδος Clear

Η μέθοδος Clear της κλάσης Graphics καθαρίζει ένα αντικείμενο σχεδίασης και ορίζει το χρώμα υπόβαθρου του αντικειμένου. Η μέθοδος δέχεται ως παράμετρο το χρώμα υπόβαθρου.

```
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Καθαρισμός Bitmap με χρώμα υπόβαθρου λευκό
g.Clear(Color.White)
PictureBox1.Image = bmp
```

Παράδειγμα 11.5 Η μέθοδος Clear

Η μέθοδος DrawLine

Η μέθοδος DrawLine χρησιμοποιείται για τον σχεδιασμό ευθύγραμμων τμημάτων μεταξύ 2 σημείων. Η μέθοδος έχει τις παρακάτω μορφές:

A) g.DrawLine(pen, x1, y1, x2, y2)

όπου	
g	αντικείμενο Graphics
pen	εικονική πένα
x1, y1	συντεταγμένες πρώτου σημείου
x2, y2	συντεταγμένες δεύτερου σημείου

B) g.DrawLine(pen, point1, point2)

όπου	
g	αντικείμενο Graphics
pen	εικονική πένα
point1	πρώτο σημείο
point2	δεύτερο σημείο

Το παρακάτω παράδειγμα σχεδιάζει ευθύγραμμα τμήματα κάνοντας χρήση των διαφορετικών μορφών σύνταξης της μεθόδου DrawLine και διαφορετικών τύπων εικονικών πενών.

```

'Δημιουργία αντικειμένου Bitmap με διαστάσεις
'το πλάτος και το ύψος ενός PictureBox
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)

'Δημιουργία αντικειμένου της κλάσης Graphics
Dim g As Graphics = Graphics.FromImage(bmp)
'Σχεδίαση γραμμής με χρήση συντεταγμένων
g.DrawLine(Pens.Black, 100, 50, 200, 50)
'Σχεδίαση γραμμής με χρήση σημείων
Dim p1 As Point = New Point(100, 75)
Dim p2 As Point = New Point(200, 75)
g.DrawLine(Pens.Red, p1, p2)
'Σχεδίαση διακεκομμένης γραμμής μαύρου χρώματος
'με καθορισμό πάχους, και σχήματος αρχής, τέλους
'και σχήματος των διακεκομμένων σημείων της γραμμής

'Δημιουργία εικονικής πέννας με μαύρο χρώμα
'και πάχος 5 pixels
Dim p As Pen = New Pen(Color.Black, 5)
'Ορισμός σχήματος αρχής
p.StartCap = Drawing2D.LineCap.RoundAnchor
'Ορισμός σχήματος τέλους
p.EndCap = Drawing2D.LineCap.ArrowAnchor
'Ορισμός τύπου διακεκομμένης γραμμής
p.DashStyle = Drawing2D.DashStyle.Dot
'Ορισμός σχήματος διακεκομμένης γραμμής
p.DashCap = Drawing2D.DashCap.Round
g.DrawLine(p, 100, 100, 200, 100)
'Ευσχέτιση αντικειμένου Bitmap με PictureBox1
PictureBox1.Image = bmp

```

Παράδειγμα 11.6 Σχεδίαση γραμμών

Η μέθοδος DrawLines

Η μέθοδος DrawLines χρησιμοποιείται για την σχεδίαση γραμμών μεταξύ διαφόρων σημείων της επιφάνειας σχεδίασης. Η μέθοδος αυτή δέχεται ως παράμετρο ένα πίνακα με τα σημεία σχεδίασης.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)

PictureBox1.Image = bmp
'Ορισμός πίνακα 5 σημείων
Dim points() As Point = {New Point(100, 100), _
                          New Point(150, 100), _
                          New Point(150, 150), _
                          New Point(200, 150), _
                          New Point(200, 200)}
'Σχεδίαση γραμμών μεταξύ των 5 σημείων
g.DrawLines(Pens.Black, points)
Dim g As Graphics = Graphics.FromImage(bmp)

```

Παράδειγμα 11.7 Σχεδίαση πολλαπλών γραμμών με την μέθοδο DrawLines

Οι μέθοδοι DrawPolygon και FillPolygon

Η μέθοδος DrawPolygon χρησιμοποιείται για την σχεδίαση του περιγράμματος ενός πολυγώνου. Η μέθοδος αυτή δέχεται ως παράμετρο μία εικονική πένα και ένα πίνακα σημείων τα οποία ορίζουν το πολύγωνο.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων του πολυγώνου
Dim points() As Point = {New Point(100, 200), _
                           New Point(250, 120), _
                           New Point(440, 250), _
                           New Point(80, 270)}

'Σχεδίαση πολύγωνου
g.DrawPolygon(Pens.Black, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.8 Σχεδίαση περιγράμματος πολυγώνου

Η μέθοδος FillPolygon χρησιμοποιείται για το γέμισμα με χρώμα ενός πολυγώνου. Η μέθοδος αυτή δέχεται ως παράμετρο ένα εικονικό πινέλο και ένα πίνακα σημείων τα οποία ορίζουν το πολύγωνο. Αναλόγως του τύπου του εικονικού πινέλου το οποίο χρησιμοποιείται, γίνεται και το γέμισμα του πολυγώνου. Τα παρακάτω παραδείγματα χρησιμοποιούν συμπαγές πινέλο, διακεκομμένο πινέλο, γραμμικό ντεγκραντέ πινέλο, ντεγκραντέ πινέλο διαδρομής και πινέλο υφής.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων του πολυγώνου
Dim points() As Point = {New Point(100, 200), _
                           New Point(250, 120), _
                           New Point(440, 250), _
                           New Point(80, 270)}

'Σχεδίαση περιγράμματος πολυγώνου
g.DrawPolygon(Pens.Black, points)
'Γέμισμα πολυγώνου με συμπαγές χρώμα
g.FillPolygon(Brushes.Red, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.9 Σχεδίαση πολυγώνου με συμπαγές χρώμα γεμίματος

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων του πολυγώνου
Dim points() As Point = {New Point(100, 200), _
                           New Point(250, 120), _
                           New Point(440, 250), _
                           New Point(80, 270)}

'Δημιουργία πινέλου διαγράμμισης
Dim hBrush As HatchBrush
hBrush = New HatchBrush(HatchStyle.ForwardDiagonal, _
                        Color.Black, Color.White)

'Σχεδίαση περιγράμματος πολυγώνου
g.DrawPolygon(Pens.Black, points)
'Γέμισμα πολυγώνου με διαγράμμιση
g.FillPolygon(hBrush, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.10 Σχεδίαση πολυγώνου με διαγραμμισμένο χρώμα γεμίματος

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων του πολυγώνου
Dim points() As Point = {New Point(100, 200), _

```

```

New Point(250, 120), _
New Point(440, 250), _
New Point(80, 270)}

'Δημιουργία πινέλου γραμμικού ντεγκραντέ γεμίσματος
Dim rect As Rectangle = New Rectangle(80, 120, 440, 270)
Dim lgBrush As LinearGradientBrush
lgBrush = New LinearGradientBrush(rect, Color.Blue, _
    Color.Red, LinearGradientMode.BackwardDiagonal)
'Σχεδίαση περιγράμματος πολύγωνου
g.DrawPolygon(Pens.Black, points)
'Γέμισμα πολυγώνου με γραμμικό ντεγκραντέ
g.FillPolygon(lgBrush, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.11 Σχεδίαση πολυγώνου με γραμμικό ντεγκραντέ χρώμα γεμίσματος

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων του πολυγώνου
Dim points() As Point = {New Point(100, 200), _
    New Point(250, 120), _
    New Point(440, 250), _
    New Point(80, 270)}
'Ορισμός και δημιουργία αντικειμένου ντεγκραντέ διαδρομής
Dim pgBrush As PathGradientBrush
pgBrush = New PathGradientBrush(points)
'Δημιουργία πίνακα τεσσάρων χρωμάτων
Dim sColors() As Color = {Color.Red, _
    Color.Green, Color.Blue, Color.Yellow}
pgBrush.SurroundColors = sColors
'Καθορισμός χρώματος και συντεταγμένων κεντρικού σημείου
pgBrush.CenterColor = Color.White
pgBrush.CenterPoint = New Point(200, 200)
'Σχεδίαση περιγράμματος πολύγωνου
g.DrawPolygon(Pens.Black, points)
'Γέμισμα πολυγώνου με ντεγκραντέ διαδρομής
g.FillPolygon(pgBrush, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.12 Γέμισμα πολυγώνου με ντεγκραντέ διαδρομής

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων του πολυγώνου
Dim points() As Point = {New Point(100, 200), _
    New Point(250, 120), _
    New Point(440, 250), _
    New Point(80, 270)}
'Δημιουργία πινέλου υφής
Dim tBmp As Bitmap = Bitmap.FromFile("c:\dock.jpg")
Dim tBrush As TextureBrush
tBrush = New TextureBrush(tBmp)
'Σχεδίαση περιγράμματος πολύγωνου
g.DrawPolygon(Pens.Black, points)
'Γέμισμα πολυγώνου με πινέλο υφής
g.FillPolygon(tBrush, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.13 Σχεδίαση πολυγώνου με πινέλο υφής

Οι μέθοδοι DrawRectangle και FillRectangle

Οι μέθοδοι DrawRectangle και FillRectangle χρησιμοποιούνται για τον σχεδιασμό και το γέμισμα ορθογωνίων.

Η μέθοδος DrawRectangle δέχεται ως παραμέτρους μία εικονική πένα και μια μεταβλητή τύπου Rectangle ή εναλλακτικά μια εικονική πένα, τις συντεταγμένες του πάνω αριστερού σημείου του ορθογωνίου και το πλάτος και ύψος του ορθογωνίου.

Το παρακάτω παράδειγμα σχεδιάζει δύο ορθογώνια, το πρώτο με χρήση μεταβλητής τύπου Rectangle και το δεύτερο με την χρήση συντεταγμένων και διαστάσεων.

```
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός ορθογωνίου
Dim rect1 As Rectangle
rect1 = New Rectangle(100, 100, 50, 50)
'Σχεδιασμός περιγράμματος ορθογωνίου
g.DrawRectangle(Pens.Blue, rect1)

'Σχεδιασμός ορθογωνίου με χρήση συντεταγμένων
g.DrawRectangle(Pens.Red, 200, 100, 50, 50)
PictureBox1.Image = bmp
```

Παράδειγμα 11.14 Σχεδίαση περιγράμματος ορθογωνίου

Η μέθοδος FillRectangle δέχεται τις ίδιες παραμέτρους με την μέθοδο DrawRectangle με την διαφορά ότι αντί της εικονικής πένας χρησιμοποιείται εικονικό πινέλο.

```
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός ορθογωνίου
Dim rect1 As Rectangle
rect1 = New Rectangle(100, 100, 50, 50)
'Γέμισμα ορθογωνίου
g.FillRectangle(Brushes.Red, rect1)
PictureBox1.Image = bmp
```

Παράδειγμα 11.15 Γέμισμα ορθογωνίου με συμπαγές χρώμα

Οι μέθοδοι DrawEllipse και FillEllipse

Οι μέθοδοι DrawEllipse και FillEllipse χρησιμοποιούνται για τον σχεδιασμό και το γέμισμα κύκλων και ελλείψεων. Οι μέθοδοι δέχονται αντίστοιχες παραμέτρους με τις μεθόδους DrawRectangle και FillRectangle, διότι χρησιμοποιούν ως ορισμό για την έλλειψη, αντί του κέντρου και της κύριας και δευτερεύουσας ακτίνας της έλλειψης, το ορθογώνιο το οποίο περιβάλλει την έλλειψη.

Η μέθοδος DrawEllipse δέχεται ως παραμέτρους μία εικονική πένα και μια μεταβλητή τύπου Rectangle ή εναλλακτικά μια εικονική πένα, τις συντεταγμένες του πάνω αριστερού σημείου και το πλάτος και ύψος του ορθογωνίου που περιβάλλει την έλλειψη.

Το παρακάτω παράδειγμα σχεδιάζει ένα κύκλο και μία έλλειψη. Ο κύκλος σχεδιάζεται με την χρήση μεταβλητής τύπου Rectangle όπου το πλάτος είναι ίσο με το ύψος του ορθογωνίου. Η έλλειψη σχεδιάζεται με την χρήση συντεταγμένων και διαστάσεων.

```
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
```

```

'Ορισμός τετραγώνου που περιβάλλει τον κύκλο
Dim rect1 As Rectangle
rect1 = New Rectangle(100, 100, 50, 50)
'Σχεδιασμός κύκλου
g.DrawEllipse(Pens.Blue, rect1)
'Σχεδιασμός έλλειψης με χρήση συντεταγμένων
g.DrawEllipse(Pens.Red, 200, 100, 150, 50)
PictureBox1.Image = bmp

```

Παράδειγμα 11.16 Σχεδίαση κύκλου και έλλειψης

Το γέμισμα με χρώμα μιας έλλειψης γίνεται με μέθοδο FillEllipse η οποία δέχεται τις ίδιες παραμέτρους με την μέθοδο DrawEllipse με την διαφορά ότι αντί της εικονικής πένας χρησιμοποιείται εικονικό πινέλο.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός ορθογώνιου που περικλείει την έλλειψη
Dim rect1 As Rectangle
rect1 = New Rectangle(100, 100, 150, 50)
'Γέμισμα Έλλειψης με συμπαγές χρώμα
g.FillEllipse(Brushes.Red, rect1)
PictureBox1.Image = bmp

```

Παράδειγμα 11.17 Γέμισμα έλλειψης με συμπαγές χρώμα

Η μέθοδος DrawCurve

Η μέθοδος DrawCurve χρησιμοποιείται για την σχεδιασμό καμπυλών. Η μέθοδος αυτή δέχεται ως παράμετρο μία εικονική πένα και ένα πίνακα σημείων τα οποία ορίζουν τα σημεία από τα οποία διέρχεται η καμπύλη.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων της καμπύλης
Dim points() As Point = {New Point(100, 200), _
    New Point(250, 120), _
    New Point(440, 250), _
    New Point(80, 270)}
'Σχεδίαση καμπύλης
g.DrawCurve(Pens.Black, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.18 Σχεδιασμός καμπύλης

Οι μέθοδοι DrawClosedCurve και FillClosedCurve

Οι μέθοδοι DrawClosedCurve και FillClosedCurve χρησιμοποιούνται για τον σχεδιασμό και το γέμισμα κλειστών καμπυλών.

Η μέθοδος DrawClosedCurve δέχεται ως παραμέτρους μια εικονική πένα σχεδίασης και ένα πίνακα σημείων από όπου διέρχεται η καμπύλη.

Το γέμισμα με χρώμα μιας κλειστής καμπύλης γίνεται με την μέθοδο FillClosedCurve η οποία δέχεται τις ίδιες παραμέτρους με την μέθοδο DrawClosedCurve με την διαφορά ότι αντί της εικονικής πένας χρησιμοποιείται εικονικό πινέλο.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων της καμπύλης
Dim points() As Point = {New Point(100, 200), _
                           New Point(250, 120), _
                           New Point(440, 250), _
                           New Point(80, 270)}
'Σχεδίαση κλειστής καμπύλης
g.DrawClosedCurve(Pens.Black, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.19 Σχεδιασμός κλειστής καμπύλης

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Ορισμός πίνακα σημείων της καμπύλης
Dim points() As Point = {New Point(100, 200), _
                           New Point(250, 120), _
                           New Point(440, 250), _
                           New Point(80, 270)}
'Σχεδίαση κλειστής καμπύλης
g.DrawClosedCurve(Pens.Black, points)
'Γέμισμα κλειστής καμπύλης
g.FillClosedCurve(Brushes.Green, points)
PictureBox1.Image = bmp

```

Παράδειγμα 11.20 Σχεδιασμός και γέμισμα κλειστής καμπύλης

Η μέθοδος DrawBezier

Η μέθοδος DrawBezier σχεδιάζει καμπύλες Bezier οι οποίες ορίζονται από τα δύο άκρα τους και δύο σημεία ελέγχου τα οποία ελκύουν την καμπύλη καθώς κινείται από το ένα άκρο στο άλλο.

Η μέθοδος DrawBezier δέχεται ως παραμέτρους μία εικονική πένα και τα τέσσερα σημεία τα οποία ορίζουν την καμπύλη. Τα σημεία δίνονται είτε ως μεταβλητές τύπου Point είτε ως συντεταγμένες x, y. Πρώτα δίνεται το σημείο αρχής, στην συνέχεια τα δύο σημεία ελέγχου και τέλος το σημείο τέλους της καμπύλης.

Το παρακάτω παράδειγμα σχεδιάζει καμπύλες Bezier κάνοντας εναλλακτικά χρήση μεταβλητών τύπου Point και συντεταγμένων.

```

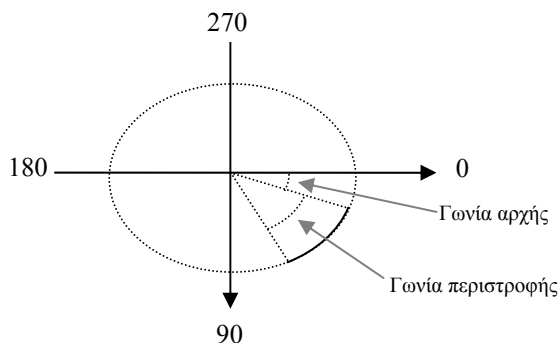
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Σχεδίαση καμπύλης Bezier με σημεία
g.DrawBezier(Pens.Black, New Point(100, 200), _
              New Point(150, 250), _
              New Point(250, 100), _
              New Point(300, 200))
'Σχεδίαση καμπύλης Bezier με συντεταγμένες
g.DrawBezier(Pens.Black, 100, 300, _
              250, 200, 150, 350, 300, 300)
PictureBox1.Image = bmp

```

Παράδειγμα 11.21 Σχεδιασμός καμπύλης Bezier

Η μέθοδος DrawArc

Η μέθοδος DrawArc σχεδιάζει ένα τόξο μιας έλλειψης. Οι παράμετροι της μεθόδου είναι παρόμοιες με την μέθοδο DrawEllipse με την διαφορά ότι στο τέλος προστίθενται ως παράμετροι η γωνία αρχής και η γωνία περιστροφής του τόξου. Οι γωνίες μετρώνται σε μοίρες και αυξάνονται σε δεξιόστροφη κατεύθυνση όπως φαίνεται στο παρακάτω σχήμα.



Η μέθοδος DrawArc έχει ως παραμέτρους μία εικονική πένα, μια μεταβλητή τύπου Rectangle ή εναλλακτικά τις συντεταγμένες του πάνω αριστερού σημείου και το πλάτος και ύψος του ορθογωνίου που περιβάλλει την έλλειψη, την γωνία αρχής του τόξου και την γωνία περιστροφής.

Το παρακάτω παράδειγμα σχεδιάζει τόξα με την χρήση παραμέτρων:

- εικονική πένα, αντικείμενο Rectangle, γωνία αρχής και γωνία περιστροφής
- εικονική πένα, συντεταγμένες και διαστάσεις ορθογωνίου και γωνία αρχής και περιστροφής.

```
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Σχεδίαση τόξου με χρήση Rectangle
Dim rect As New Rectangle(100, 100, 50, 50)
g.DrawArc(Pens.Green, rect, 0, 45)
'Σχεδίαση τόξου με χρήση συντεταγμένων
g.DrawArc(Pens.Red, 100, 100, 50, 50, 45, 90)
PictureBox1.Image = bmp
```

Παράδειγμα 11.22 Σχεδιασμός τόξων

Οι μέθοδοι DrawPie και FillPie

Οι μέθοδοι DrawPie και FillPie σχεδιάζουν και γεμίζουν αντίστοιχα πίτες οι οποίες χρησιμοποιούνται σε γραφήματα πίτας.

Ο ορισμός των παραμέτρων της μεθόδου DrawPie είναι παρόμοιος με τον ορισμό των παραμέτρων της μεθόδου DrawArc. Η μέθοδος DrawPie έχει ως παραμέτρους μία εικονική πένα, μια μεταβλητή τύπου Rectangle ή εναλλακτικά τις συντεταγμένες του πάνω αριστερού σημείου και το πλάτος και ύψος του ορθογωνίου που περιβάλλει την έλλειψη, την γωνία αρχής του τόξου και την γωνία περιστροφής.

Η μέθοδος FillPie γεμίζει μία πίτα και χρησιμοποιεί τις ίδιες παραμέτρους με την μέθοδο DrawPie με την διαφορά ότι αντί της εικονικής πέννας χρησιμοποιεί εικονικό πινέλο.

Το παρακάτω παράδειγμα σχεδιάζει και γεμίζει πίτες με την χρήση εναλλακτικά αντικειμένων Rectangle και συντεταγμένων για το ορθογωνίου το οποίο περιβάλλει την έλλειψη της οποίας τμήματα είναι οι πίτες.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
'Σχεδίαση και γέμισμα πίτας με χρήση Rectangle
Dim rect As New Rectangle(100, 100, 50, 50)
g.DrawPie(Pens.Black, rect, 0, 45)
g.FillPie(Brushes.Green, rect, 0, 45)
'Σχεδίαση και γέμισμα πίτας με χρήση συντεταγμένων
g.DrawPie(Pens.Black, 100, 100, 50, 50, 45, 90)
g.FillPie(Brushes.Red, 100, 100, 50, 50, 45, 90)
PictureBox1.Image = bmp

```

Παράδειγμα 11.23 Σχεδιασμός και γέμισμα πίτας

Η μέθοδος DrawString

Η μέθοδος DrawString χρησιμοποιείται για την εκτύπωση κειμένου. Η μέθοδος DrawString δέχεται ως παραμέτρους το κείμενο εκτύπωσης, την γραμματοσειρά εκτύπωσης, το πινέλο με το οποίο θα γίνει η εκτύπωση και τις συντεταγμένες του σημείου εκτύπωσης. Προαιρετικά μπορεί να ορισθεί και ένα αντικείμενο StringFormat το οποίο καθορίζει διάφορα χαρακτηριστικά για την εκτύπωση, όπως για παράδειγμα την στοίχιση και την κατεύθυνση εκτύπωσης.

Για τον ορισμό της γραμματοσειράς εκτύπωσης χρησιμοποιείται ένα αντικείμενο της κλάσης Font. Η δημιουργία ενός αντικειμένου της κλάσης Font γίνεται με τον καθορισμό του ονόματος της γραμματοσειράς και του μεγέθους της.

```

'Δημιουργία αντικειμένου γραμματοσειράς
Dim f As Font
f = New Font("Arial", 26)

```

Προαιρετικά κατά την δημιουργία ενός αντικειμένου Font μπορούμε να ορίσουμε και την ιδιότητα Style της γραμματοσειράς με την χρήση του απαριθμητού τύπου δεδομένων FontStyle. Οι τιμές της ιδιότητας Style μπορεί να είναι:

- FontStyle.FontBold,
- FontStyle.Italic,
- FontStyle.Regular,
- FontStyle.StrikeOut και
- FontStyle.Underline.

Μπορούμε επίσης να χρησιμοποιήσουμε ως τιμή και συνδυασμούς των παραπάνω τιμών με τον τελεστή or (π.χ. FontStyle.Bold or FontStyle.Italic).

```

'Δημιουργία αντικειμένου γραμματοσειράς
'με καθορισμό Style
Dim f2 As Font
f2 = New Font("Arial", 26, _
    FontStyle.Bold Or FontStyle.Italic)

```

Το παρακάτω παράδειγμα ορίζει χρώμα υπόβαθρου λευκό και εκτυπώνει κείμενο σε διάφορα σημεία ενός PictureBox.

```

Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
    PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
g.Clear(Color.White)
'Δημιουργία αντικειμένου γραμματοσειράς
Dim f1 As Font
f1 = New Font("Arial", 26)
'Εκτύπωση κειμένου
g.DrawString("Hello", f1, Brushes.Green, 100, 50)

```

```
'Δημιουργία αντικειμένου γραμματοσειράς
'με καθορισμό Style
Dim f2 As Font
f2 = New Font("Arial", 26, _
               FontStyle.Bold Or FontStyle.Italic)
'Εκτύπωση κειμένου
g.DrawString("Hello", f2, Brushes.Blue, 100, 100)
PictureBox1.Image = bmp
```

Παράδειγμα 11.24 Εκτύπωση κειμένου

Η εκτύπωση κειμένου με την μέθοδο DrawString μπορεί να γίνει και σε ένα προκαθορισμένο ορθογώνιο πλαίσιο. Στην περίπτωση αυτή αντί των συντεταγμένων x και y του σημείου εκτύπωσης χρησιμοποιείται ως παράμετρος ένα αντικείμενο Rectangle.

```
'Εκτύπωση κειμένου σε πλαίσιο
Dim rect As Rectangle = New Rectangle(0, 0, 100, 250)
g.DrawString("1234567890", f1, Brushes.Blue, rect)
```

Ο ορισμός ενός αντικειμένου StringFormat γίνεται με την δημιουργία ενός νέου αντικειμένου της κλάσης StringFormat και τον ορισμό των ιδιοτήτων του Alignment, Trimming και FormatFlags. Η ιδιότητα Alignment καθορίζει την στοίχιση του κειμένου όταν η εκτύπωση γίνεται σε ορθογώνιο πλαίσιο. Η ιδιότητα Trimming καθορίζει με ποιόν τρόπο περικόπτεται το κείμενο όταν εκτυπώνεται σε ορθογώνιο πλαίσιο και μέρος του κειμένου δεν χωρά στο πλαίσιο. Η ιδιότητα FormatFlags καθορίζει διάφορα χαρακτηριστικά της εκτύπωσης όπως την κατεύθυνση εκτύπωσης (DirectionVertical).

Το παρακάτω παράδειγμα εκτυπώνει κείμενο σε ορθογώνια πλαίσια και κάνει χρήση αντικειμένων StringFormat για τον καθορισμό διάφορων χαρακτηριστικών εκτύπωσης.

```
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)
Dim g As Graphics = Graphics.FromImage(bmp)
g.Clear(Color.White)

'Δημιουργία αντικειμένου γραμματοσειράς
Dim f1 As Font
f1 = New Font("Arial", 18)
'Εκτύπωση κειμένου με ορισμό κατεύθυνσης εκτύπωσης
Dim sf1 As StringFormat = New StringFormat()
sf1.FormatFlags = StringFormatFlags.DirectionVertical
g.DrawString("Κάθετο κείμενο", f1, Brushes.Red, 10, 10, sf1)

'Εκτύπωση κειμένου σε πλαίσιο
Dim rect2 As Rectangle = New Rectangle(50, 10, 400, 50)
Dim sf2 As StringFormat = New StringFormat()
'Στοίχιση στο κέντρο του πλαισίου
sf2.Alignment = StringAlignment.Center
g.DrawRectangle(Pens.Black, rect2)
g.DrawString("Κείμενο σε πλαίσιο", f1, _
              Brushes.Blue, rect2, sf2)

'Εκτύπωση κειμένου σε πλαίσιο
Dim rect3 As Rectangle = New Rectangle(50, 100, 170, 80)
Dim sf3 As StringFormat = New StringFormat()
'Στοίχιση στα δεξιά του πλαισίου
sf3.Alignment = StringAlignment.Far
g.DrawRectangle(Pens.Black, rect3)
g.DrawString("Εκτύπωση κειμένου σε πολλές γραμμές", _
              f1, Brushes.Blue, rect3, sf3)
```

```

'Εκτύπωση κειμένου σε πλαίσιο
Dim rect4 As Rectangle = New Rectangle(280, 100, 170, 80)
Dim sf4 As StringFormat = New StringFormat()
'Στοίχιση στα αριστερά του πλαισίου
sf4.Alignment = StringAlignment.Near
'Καθορισμός τρόπου αποκοπής κειμένου
sf4.Trimming = StringTrimming.EllipsisWord
g.DrawRectangle(Pens.Black, rect4)
g.DrawString("Εκτύπωση κειμένου σε πολλές γραμμές", _
    f1, Brushes.Blue, rect4, sf4)

PictureBox1.Image = bmp

```

Παράδειγμα 11.25 Εκτύπωση κειμένου σε ορθογώνιο πλαίσιο

Η μέθοδος DrawImage

Η μέθοδος DrawImage χρησιμοποιείται για τον σχεδιασμό μιας εικόνας ή ενός τμήματος της πάνω σε μία άλλη εικόνα.

Η μέθοδος αυτή συντάσσεται με πολλούς διαφορετικούς τρόπους. Οι βασικότερες μορφές σύνταξης της μεθόδου είναι:

Σχεδίασης εικόνας σε σημείο

Η σχεδίαση εικόνας σε σημείο είναι η πιο απλή μορφή σύνταξης της μεθόδου DrawImage. Η μορφή αυτή χρησιμοποιεί ως παραμέτρους την εικόνα και τις συντεταγμένες του σημείου όπου θα γίνει η εκτύπωση της εικόνας.

```

'Εκτύπωση εικόνας σε σημείο
g.DrawImage(img, 300, 10)

```

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε ως παραμέτρους την εικόνα και μια μεταβλητή σημείου

```

'Εκτύπωση εικόνας σε σημείο
Dim p As Point = New Point(300, 20)
g.DrawImage(img, p)

```

ή χωρίς την χρήση μεταβλητής

```

'Εκτύπωση εικόνας σε σημείο
g.DrawImage(img, New Point(300, 20))

```

Σχεδίασης εικόνας σε ορθογώνιο πλαίσιο

Η μορφή αυτής σχεδίασης προσαρμόζει μια εικόνα ώστε να καλύψει ένα ορθογώνιο πλαίσιο της επιφάνειας σχεδίασης. Η μέθοδος αυτή δέχεται ως παραμέτρους την εικόνα και τις συντεταγμένες και διαστάσεις του ορθογωνίου. Εναλλακτικά μπορεί να χρησιμοποιηθούν ως παράμετροι η εικόνα και μια μεταβλητή τύπου Rectangle.

```

'Εκτύπωση εικόνας σε ορθογώνιο με χρήση συντεταγμένων
g.DrawImage(img, 150, 0, 50, 50)
'Εκτύπωση εικόνας με χρήση Rectangle
Dim rect As Rectangle
rect = New Rectangle(0, 0, 100, 100)
g.DrawImage(img, rect)

```

Σχεδίασης τμήματος εικόνας σε ορθογώνιο πλαίσιο

Η μέθοδος αυτή δέχεται ως παραμέτρους την εικόνα η οποία θα εκτυπωθεί, μια μεταβλητή τύπου Rectangle η οποία καθορίζει το ορθογώνιο πλαίσιο όπου θα εκτυπωθεί η εικόνα, μια δεύτερη μεταβλητή τύπου Rectangle η οποία καθορίζει το τμήμα της εικόνας το οποίο θα εκτυπωθεί και τέλος μια μεταβλητή η οποία καθορίζει την μονάδα μέτρησης των γραφικών διαστάσεων (π.χ. GraphicsUnit.Pixel). Το τμήμα της εικόνας το οποίο σχεδιάζεται, προσαρμόζεται ώστε να καλύψει ολόκληρη την ορθογώνια περιοχή εκτύπωσης.

```
'Εκτύπωση τμήματος εικόνας σε ορθογώνιο
'Ορισμός ορθογωνίου στο οποίο γίνεται η εκτύπωση
Dim rect1 As New Rectangle(180, 100, 100, 100)
'Ορισμός τμήματος εικόνας η οποία θα εκτυπωθεί
Dim rect2 As New Rectangle(500, 500, 400, 300)
g.DrawImage(img, rect1, rect2, GraphicsUnit.Pixel)
```

Σχεδίασης εικόνας σε παραλληλόγραμμο

Η μέθοδος αυτή σχεδιάζει μία εικόνα σε μια οποιοδήποτε παραλληλόγραμμη περιοχή της επιφάνειας σχεδίασης (όχι απαραίτητα ορθογώνια). Η μέθοδος αυτή προσαρμόζει τις διαστάσεις και περιστρέφει την αρχική εικόνα ώστε να εκτυπωθεί στην καθορισμένη περιοχή εκτύπωσης. Το παραλληλόγραμμο ορίζεται από ένα πίνακα τριών σημείων τα οποία ορίζουν μοναδικά το παραλληλόγραμμο (το τέταρτο σημείο προκύπτει από τα άλλα τρία σημεία). Η μέθοδος δέχεται ως παραμέτρους την εικόνα, και ένα πίνακα τριών σημείων ο οποίος καθορίζει το παραλληλόγραμμο όπου θα γίνει η εκτύπωση.

```
'Εκτύπωση εικόνας σε παραλληλόγραμμο
'Ορισμός πίνακα 3 σημείων
Dim points1() As Point = {New Point(100, 100), _
                           New Point(150, 200), _
                           New Point(50, 200)}
g.DrawImage(img, points1)
```

Σχεδίασης τμήματος εικόνας σε παραλληλόγραμμο

Η σχεδίαση τμήματος εικόνας σε παραλληλόγραμμη περιοχή (όχι απαραίτητα ορθογώνια) γίνεται με την καθορισμό της εικόνας, ενός πίνακα τριών σημείων ο οποίος ορίζει την παραλληλόγραμμη περιοχή εκτύπωσης (όπως και στην εκτύπωση εικόνας σε παραλληλόγραμμο), μια μεταβλητή τύπου Rectangle η οποία καθορίζει το τμήμα της εικόνας το οποίο θα εκτυπωθεί και τέλος μια μεταβλητή η οποία καθορίζει την μονάδα μέτρησης των γραφικών διαστάσεων (π.χ. GraphicsUnit.Pixel).

```
'Εκτύπωση τμήματος εικόνας σε παραλληλόγραμμο
'Ορισμός πίνακα 3 σημείων
Dim points3() As Point = {New Point(200, 280), _
                           New Point(225, 200), _
                           New Point(250, 300)}
'Ορισμός τμήματος εικόνας που θα εκτυπωθεί
Dim rect3 As New Rectangle(500, 500, 400, 300)
g.DrawImage(img, points3, rect3, GraphicsUnit.Pixel)
```

Το παρακάτω παράδειγμα χρησιμοποιεί όλες τις βασικές μορφές σχεδίασης εικόνων με την μέθοδο DrawImage.

```
Dim bmp As Bitmap
bmp = New Bitmap(PictureBox1.Width, _
                 PictureBox1.Height)

Dim g As Graphics
g = Graphics.FromImage(bmp)
'Δημιουργία Bitmap από αρχείο
Dim img As Bitmap = Bitmap.FromFile("c:\dock.jpg")

'Εκτύπωση εικόνας σε σημείο
```



```

g.DrawImage(img, 300, 10)
Dim p As Point = New Point(350, 20)
g.DrawImage(img, p)

'Εκτύπωση εικόνας σε ορθογώνιο
g.DrawImage(img, 150, 0, 50, 50)
'Εκτύπωση με χρήση Rectangle
Dim rect As Rectangle
rect = New Rectangle(0, 0, 100, 100)
g.DrawImage(img, rect)

'Εκτύπωση τμήματος εικόνας σε ορθογώνιο
Dim rect1 As New Rectangle(180, 100, 100, 100)
Dim rect2 As New Rectangle(500, 500, 400, 300)
g.DrawImage(img, rect1, rect2, GraphicsUnit.Pixel)

'Εκτύπωση εικόνας σε παραλληλόγραμμο
'Ορισμός πίνακα 3 σημείων
Dim points1() As Point = {New Point(100, 100), _
                           New Point(150, 200), _
                           New Point(50, 200)}
g.DrawImage(img, points1)

'Εκτύπωση τμήματος εικόνας σε παραλληλόγραμμο
'Ορισμός πίνακα 3 σημείων
Dim points3() As Point = {New Point(200, 280), _
                           New Point(225, 200), _
                           New Point(250, 300)}
'Ορισμός τμήματος εικόνας που θα εκτυπωθεί
Dim rect3 As New Rectangle(500, 500, 400, 300)
g.DrawImage(img, points3, rect3, GraphicsUnit.Pixel)
PictureBox1.Image = bmp

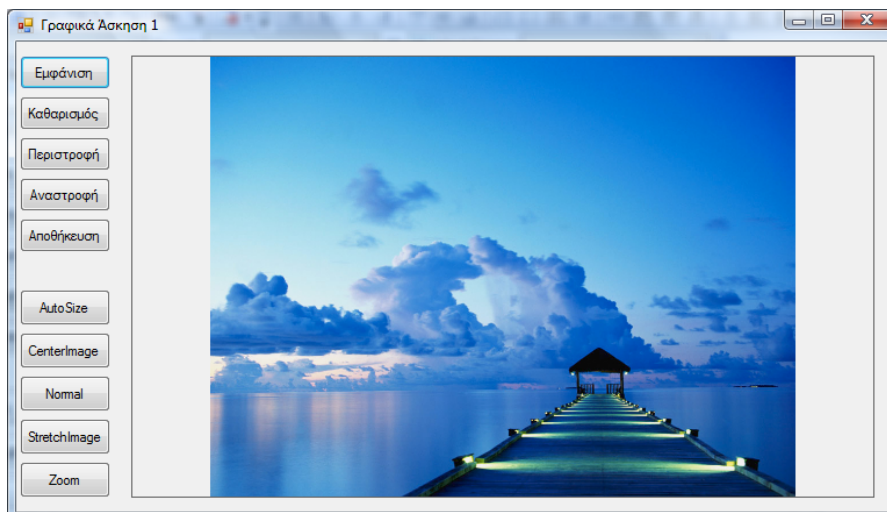
```

Παράδειγμα 11.26 Εκτύπωση εικόνας με την μέθοδο DrawImage

11.3 Ασκήσεις Κεφαλαίου 11

Ασκηση 11.1

Δημιουργήστε παραθυρική εφαρμογή με όνομα lab_11_ask_1, η οποία εμφανίζει το παρακάτω παράθυρο. Το παράθυρο περιέχει στοιχείο ελέγχου PictureBox με όνομα PictureBox1. Ορίστε την ιδιότητα BorderStyle σε FixedSingle, την ιδιότητα Size σε 640;380 (ή width 640 και height 380) και την ιδιότητα SizeMode σε Zoom.



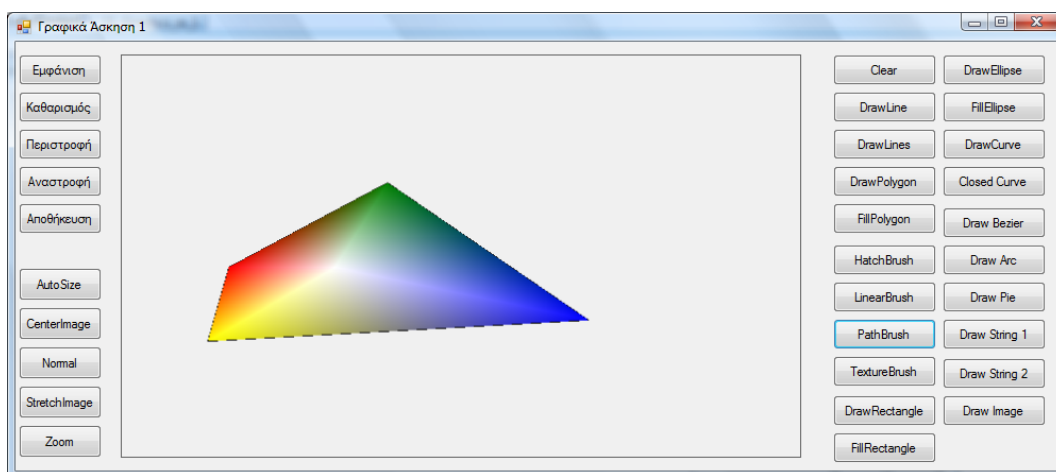
Τα πλήκτρα «Εμφάνιση», «Καθαρισμός», «Περιστροφή» και «Αποθήκευση» εκτελούν τις εντολές των παραδειγμάτων 11.1, 11.2, 11.3 και 11.4 αντίστοιχα.

Το πλήκτρο «Αναστροφή» αναστρέφει ως προς τον οριζόντιο άξονα την εμφανιζόμενη εικόνα (Χρησιμοποιήστε το παράδειγμα 11.3 με παράμετρο RotateFlipType.RotateNoneFlipX).

Τα πλήκτρα «AutoSize», «CenterImage», «Normal», «StretchImage» και «Zoom» αλλάζουν την τιμή της ιδιότητας SizeMode του PictureBox1 σε AutoSize, CenterImage, Normal, StretchImage και Zoom.

Ασκηση 11.2

Τροποποιήστε την παραθυρική εφαρμογή με όνομα lab_11_ask_1, ώστε να εμφανίζει το παρακάτω παράθυρο.



Τα πλήκτρα «Clear», «DrawLine» .. «Draw Image» αντιστοιχούν στα παραδείγματα 11.5 έως 11.25

Άσκηση 11.3

Δημιουργήστε παραθυρική εφαρμογή με όνομα lab_11_ask_3. Το παράθυρο της εφαρμογής περιέχει στοιχείο ελέγχου PictureBox με όνομα PictureBox1. Ορίστε την ιδιότητα BorderStyle σε FixedSingle, την ιδιότητα Size σε 300;300 (ή width 300 και height 300) και την ιδιότητα SizeMode σε Zoom.

Το πλήκτρο «Σχεδίαση τετραγώνων» ορίζει χρώμα υπόβαθρου λευκό και σχεδιάζει με εντολή επανάληψης τετράγωνα στις θέσεις (1,1), (3,3), .. (100,100) με πλάτος και ύψος 1,3, .. , 100.

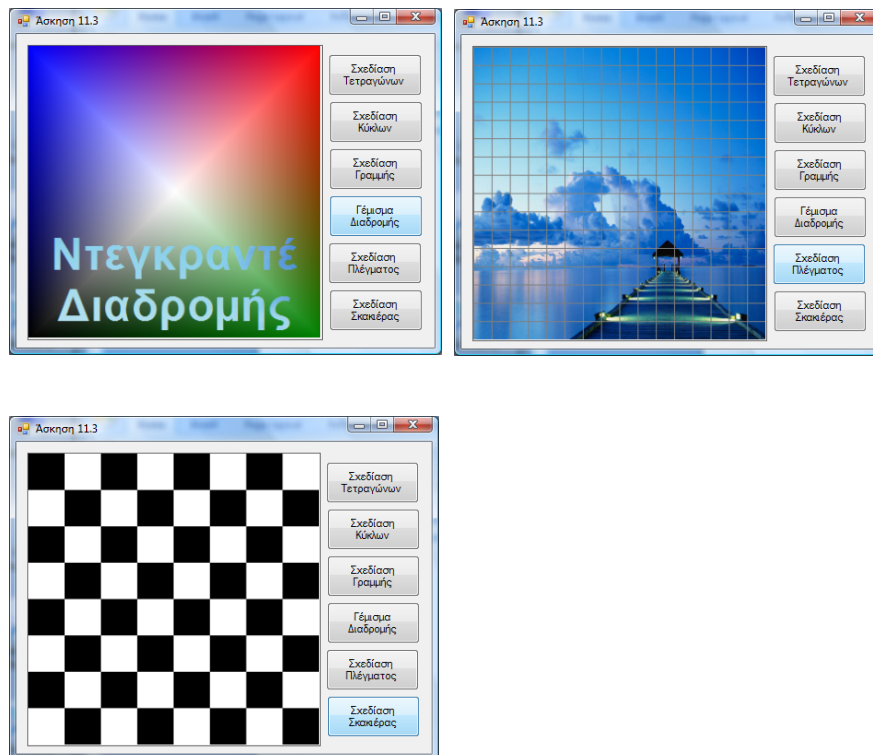
Το πλήκτρο «Σχεδίαση κύκλων» σχεδιάζει με εντολή επανάληψης κύκλους με κέντρο το σημείο (160, 160) και ακτίνες 10, 20, .. 100.

Το πλήκτρο «Σχεδίαση γραμμής» σχεδιάζει μια γραμμή με αρχή το σημείο (0,160) και τέλος το σημείο (160,160) κάνοντας χρήση της μεθόδου SetPixel. (να μην χρησιμοποιήσετε αντικείμενο της κλάσης Graphics).

Το πλήκτρο «Γέμισμα διαδρομής» γεμίζει με ντεγκραντέ διαδρομής ολόκληρο το PictureBox1 με σημεία (0,0), (319,0), (319,319) και (0,319), με χρώματα μπλε, κόκκινο, πράσινο και κίτρινο, με κεντρικό σημείο το σημείο (160,160) και κεντρικό χρώμα λευκό. Στην συνέχεια εκτυπώνει το κείμενο «Ντεγκραντέ διαδρομής» στο τετράγωνο με πάνω αριστερό σημείο (0,200), πλάτος 320 και ύψος 160. Η εκτύπωση γίνεται με στοίχιση στο κέντρο του τετράγωνου και χρησιμοποιείται η γραμματοσειρά Arial με μεγέθους 36 και μορφή (Style) Bold. Για την εκτύπωση ορίζεται και χρησιμοποιείται πινέλο υφής με χρήση της εικόνας “c:\dock.jpg”.

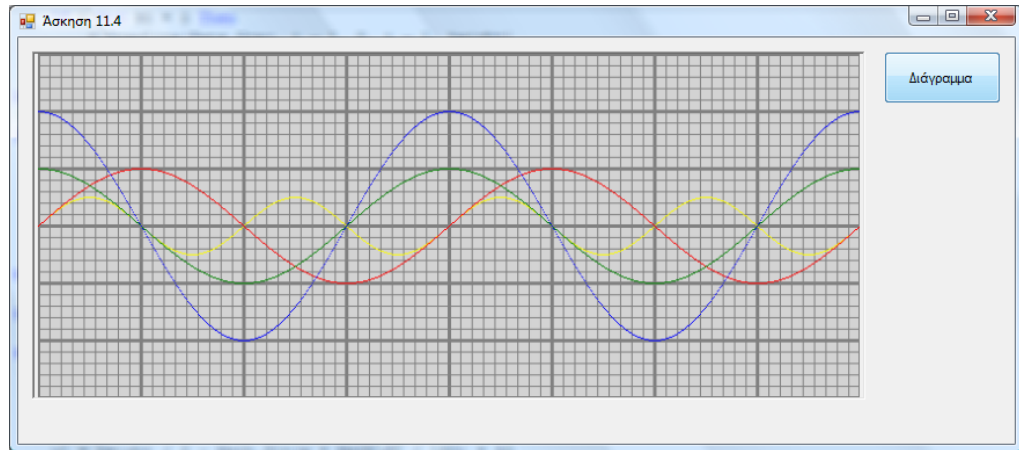
Το πλήκτρο «Σχεδίαση πλέγματος» εκτυπώνει την εικόνα “c:\dock.jpg” στο τετράγωνο του PictureBox1 (χρησιμοποιήστε την μέθοδο drawImage για να σχεδιάσετε εικόνα σε τετράγωνο) και στην συνέχεια εκτυπώνει οριζόντιες και κάθετες γραμμές με απόσταση μεταξύ τους 20 pixels με χρώμα γκρι (Colors.Gray).

Το πλήκτρο «Σχεδίαση σκακιέρας» εκτυπώνει μια σκακιέρα.



Άσκηση 11.4

Δημιουργήστε παραθυρική εφαρμογή με όνομα lab_11_ask_4. Το παράθυρο της εφαρμογής περιέχει στοιχείο ελέγχου PictureBox με όνομα PictureBox1. Ορίστε την ιδιότητα BorderStyle σε FixedSingle, την ιδιότητα Size σε 720;300 (ή width 720 και height 300) και την ιδιότητα SizeMode σε Zoom.



Το πλήκτρο «Διάγραμμα» εκτελεί το παρακάτω πρόγραμμα το οποίο σχεδιάζει ένα διάγραμμα της συνάρτησης ημίτονο(χ) για τιμές του χ από 0 έως 719.

```
Dim bmp As Bitmap = New Bitmap(PictureBox1.Width, _
                                PictureBox1.Height)

Dim g As Graphics
g = Graphics.FromImage(bmp)

g.Clear(Color.LightGray)
Dim width As Integer = PictureBox1.Width
Dim height As Integer = PictureBox1.Height
'Σχεδιασμός κάθετου πλέγματος
For i As Integer = 0 To width Step 10
    g.DrawLine(Pens.Gray, i, 0, i, height)
    'Σχεδιασμός κύριων γραμμών πλέγματος ανά 90 μοίρες
    If i Mod 90 = 0 Then
        g.DrawLine(Pens.Gray, i - 1, 0, i - 1, height)
        g.DrawLine(Pens.Gray, i + 1, 0, i + 1, height)
    End If
Next
'Σχεδιασμός οριζοντίου πλέγματος
For i As Integer = 0 To height Step 10
    g.DrawLine(Pens.Gray, 0, i, width, i)
    If i Mod 50 = 0 Then
        'Σχεδιασμός κύριων γραμμών οριζοντίου πλέγματος
        g.DrawLine(Pens.Gray, 0, i + 1, width, i + 1)
        g.DrawLine(Pens.Gray, 0, i - 1, width, i - 1)
    End If
Next

Dim y1 As Integer

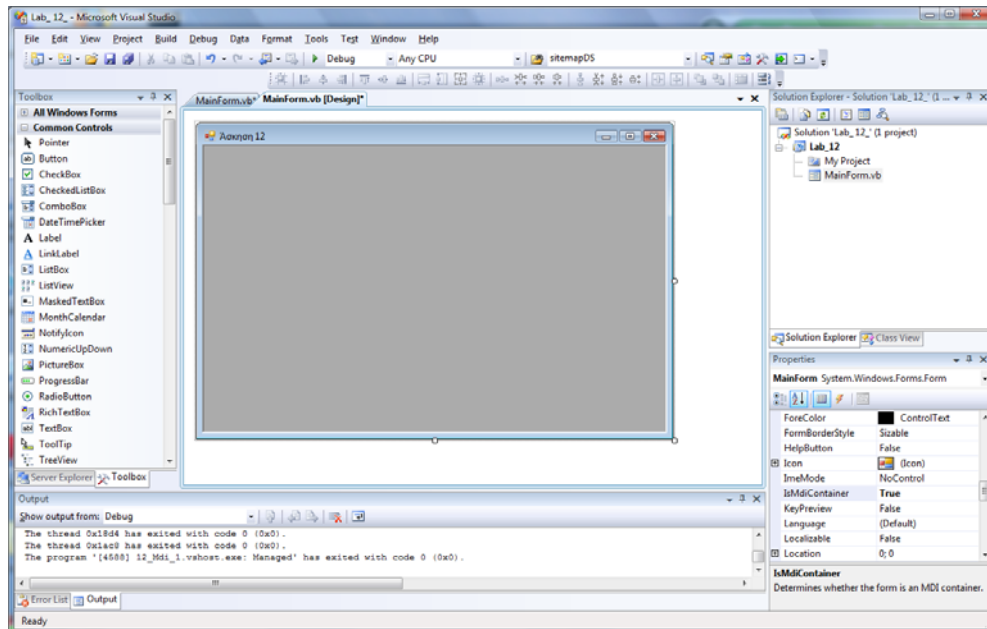
For x As Integer = 0 To width - 1
    'Υπολογισμός ημιτόνου( $\chi$ ), μεγέθυνση επί 50
    'και αφαίρεση του από το μέσο του ύψους ώστε να
    'εκτυπωθεί στο σωστό σημείο
    y1 = height / 2 - Math.Sin(x * Math.PI / 180) * 50
    bmp.SetPixel(x, y1, Color.Red)
Next
PictureBox1.Image = bmp
```

Τροποποιήστε το παραπάνω πρόγραμμα ώστε να εκτυπώνει επίσης τα διαγράμματα των συναρτήσεων $\sin(\chi)$, $\sin(\chi) + \sin(\chi)$ και $\sin(\chi) * \sin(\chi)$

12 Παράθυρα MDI - Μενού – Γραμμές Εργαλείων

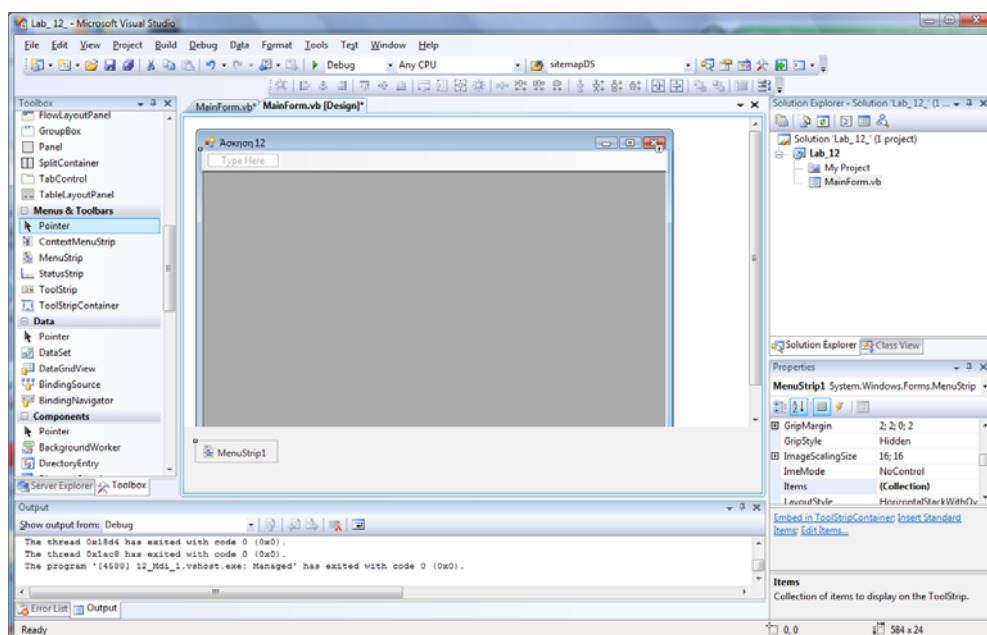
12.1 Παράθυρα MDI

Μια παραθυρική εφαρμογή μπορεί να έχει ένα κεντρικό παράθυρο το οποίο εμπεριέχει τα υπόλοιπα παράθυρα της εφαρμογής. Το παράθυρο αυτό λέγεται MDI (Main Document Interface) και μπορεί να δημιουργηθεί ορίζοντας την ιδιότητα `isMdiContainer` σε `true`.

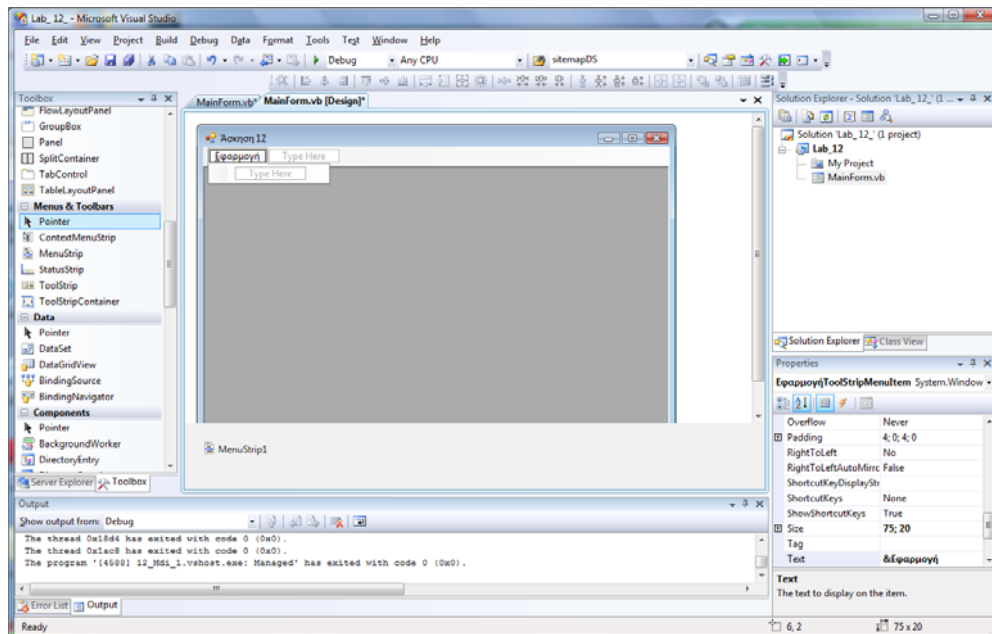


12.2 Μενού επιλογών

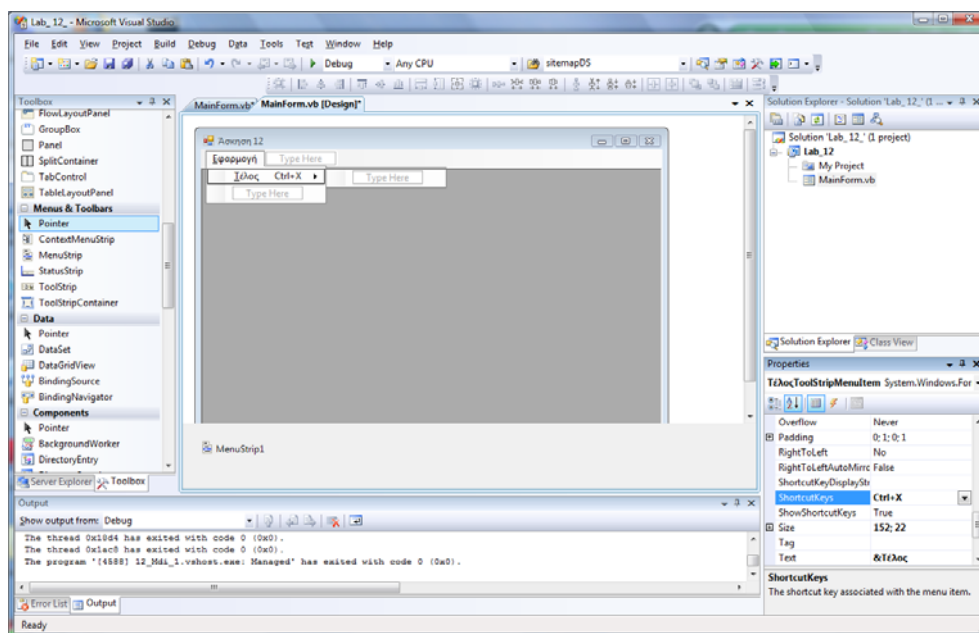
Ένα παράθυρο MDI σε τυπικές παραθυρικές εφαρμογές, αλλά και απλά παράθυρα εφαρμογών έχουν συνήθως στο πάνω μέρος του μια γραμμή μενού. Η πρόσθεση της γραμμής μενού γίνεται με την τοποθέτηση του στοιχείου ελέγχου `MenuStrip` πάνω στο παράθυρο της εφαρμογής όπως φαίνεται στην παρακάτω εικόνα.



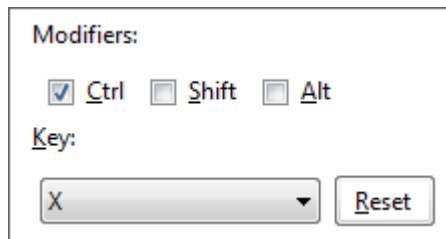
Ο σχεδιασμός της γραμμής μενού γίνεται με τον ορισμό των τίτλων των επιλογών των μενού και των υπομενού. Ο ορισμός των τίτλων γίνεται με την χρήση των πλαισίων κειμένου του οποία εμφανίζονται με αρχικό τίτλο «Type Here» σε γκρι απόχρωση. Με την επιλογή και συμπλήρωση του τίτλου μιας θέσης του μενού, εμφανίζεται ένα πλευρικό μενού (αριστερά) και ένα υπομενού (κάτω) τα οποία σε περίπτωση που συμπληρωθούν ορίζουν ένα νέο πλευρικό μενού ή ένα υπομενού αντίστοιχα, όπως φαίνεται στην παρακάτω εικόνα.



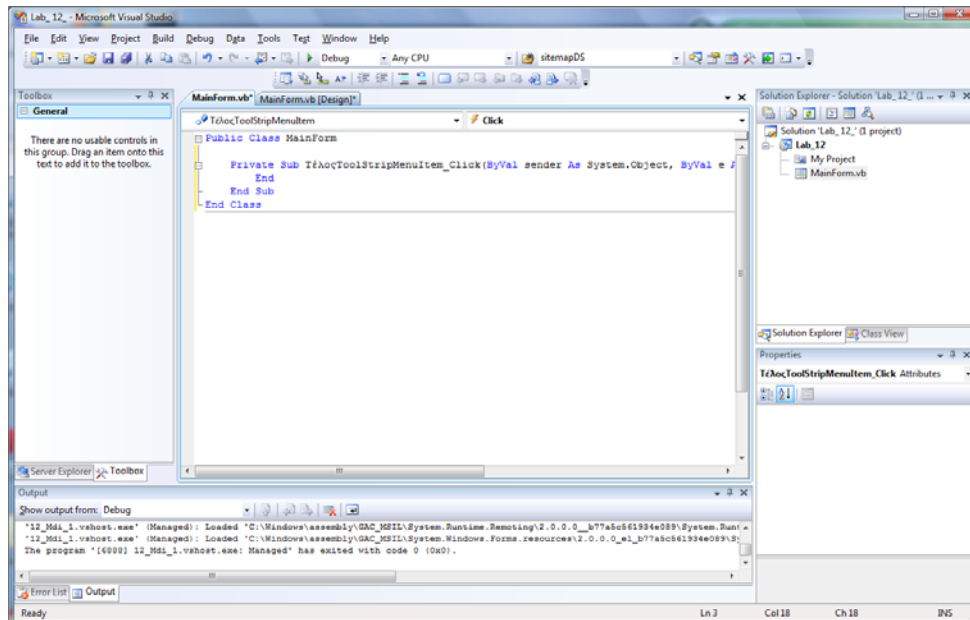
Ένα μενού μπορεί να συσχετισθεί με πλήκτρα πρόσβασης τα οποία επιτρέπουν στον χρήστη να επιλέξει το μενού με την χρήση του πλήκτρου Alt και ενός γράμματος και στην συνέχεια να μετακινηθεί σε ένα υπομενού με την χρήση των γραμμάτων. Ο ορισμός ενός πλήκτρου πρόσβασης γίνεται με την χρήση του ειδικού χαρακτήρα & στον τίτλο του μενού (ιδιότητα text). Ο ειδικός χαρακτήρας & τοποθετείται πριν από τον χαρακτήρα του τίτλου ο οποίος ορίζεται ως πλήκτρο πρόσβασης. Στην παραπάνω εικόνα ο τίτλος του μενού ορίζεται ως «&Εφαρμογή» και έχει ως αποτέλεσμα να χρησιμοποιείται ο χαρακτήρας «Ε» ως πλήκτρο συντόμευσης. Αυτό φαίνεται με την υπογράμμιση του χαρακτήρα αυτού στην γραμμή του μενού.



Ένα μενού ή υπομενού μπορεί να συσχετισθεί επίσης με πλήκτρα συντόμευσης τα οποία είναι ειδικά πλήκτρα ή συνδυασμός πλήκτρων για άμεση πρόσβαση. Τα πλήκτρα αυτά ορίζονται με την ιδιότητα `ShortcutKeys`. Στην παρακάτω εικόνα ορίζεται ως πλήκτρο συντόμευσης ο συνδυασμός των πλήκτρων `Ctrl X`.

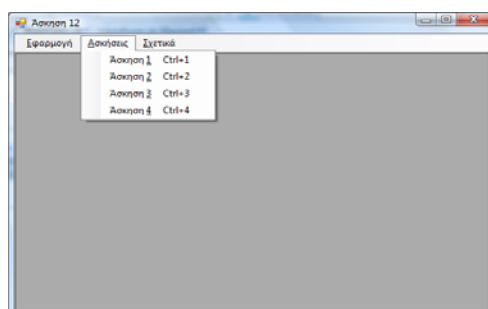


Τα μενού και υπομενού συσχετίζονται συνήθως με μία μέθοδο η οποία εκτελείται όταν επιλεγεί το μενού από τον χρήστη. Η συσχέτιση γίνεται κατά την σχεδίαση της εφαρμογής με το διπλό πάτημα του αριστερού πλήκτρου του ποντικιού πάνω στο όνομα του μενού ή υπομενού. Για παράδειγμα το υπομενού τέλος του μενού Εφαρμογή εκτελεί την μέθοδο `ΤέλοςToolStripMenuItem_Click` η οποία τελειώνει (κλείνει) την εφαρμογή με την εντολή `End`.



12.3 Δημιουργία παραθύρων που εμπεριέχονται σε MDI παράθυρο

Η δημιουργία ενός παραθύρου το οποίο εμπεριέχεται στο κύριο παράθυρο της εφαρμογής γίνεται με την προσθήκη ενός νέου παραθύρου (επιλογή `Project – Add Windows Form`) και τον ορισμό της ιδιότητας `MdiParent`. Στο παρακάτω παράδειγμα η επιλογή του μενού Άσκηση 1 ορίζει ότι το παράθυρο με όνομα `FormAskisi1` εμπεριέχεται στο κύριο παράθυρο και στην συνέχεια το δημιουργεί και το εμφανίζει.



Ο ορισμός της ιδιότητας MDIParent γίνεται πριν την εντολή για το άνοιγμα του παραθύρου όπως φαίνεται και στην παρακάτω μέθοδο η οποία καλείται με την επιλογή του μενού «Άσκηση 1»:

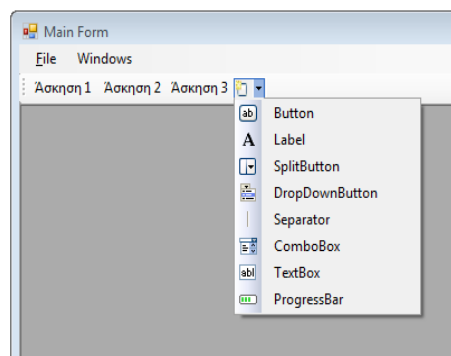
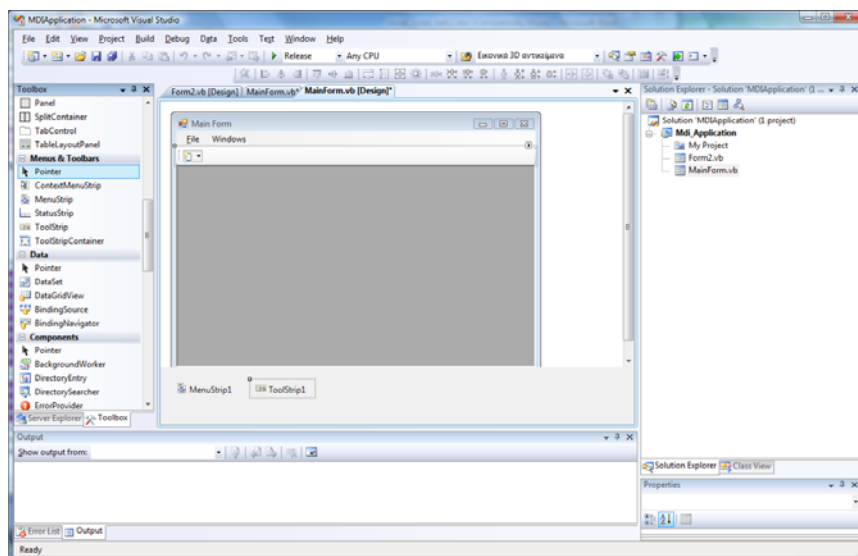
```
FormAskisi1.MdiParent = Me
FormAskisi1.Show()
```

Όταν ένα παράθυρο εμπεριέχεται σε ένα άλλο παράθυρο, τότε το μέγεθος του και η θέση του προσαρμόζονται αυτόματα στο παράθυρο το οποίο τα εμπεριέχει. Η θέση και το μέγεθος ενός παραθύρου μπορούν να τροποποιηθούν με την χρήση των ιδιοτήτων Left, Top, Width και Height τα οποία καθορίζουν τις συντεταγμένες του πάνω αριστερού σημείου (οι συντεταγμένες είναι σχετικές ως προς την αρχή του κυρίως παραθύρου) και τις διαστάσεις του, όπως φαίνεται στο παρακάτω παράδειγμα.

```
FormAskisi1.MdiParent = Me
FormAskisi1.Show()
FormAskisi1.Left = 50
FormAskisi1.Top = 50
FormAskisi1.Width = 350
FormAskisi1.Height = 250
```

12.4 Γραμμές εργαλείων

Οι γραμμές εργαλείων μπορούν να χρησιμοποιηθούν παράλληλα ή εναλλακτικά με τα μενού επιλογών. Η προσθήκη μιας γραμμής εργαλείων γίνεται με την επιλογή του στοιχείου ελέγχου ToolStrip από την εργαλειοθήκη (Toolbox) και την τοποθέτηση του πάνω στο παράθυρο.



Η προσθήκη αντικειμένων (πλήκτρων ή εικονιδίων) στην γραμμή εργαλείων γίνεται με το πάτημα του πλήκτρου του ποντικιού πάνω στο πλαίσιο επιλογής (comboBox) της γραμμής εργαλείων και την επιλογή του αντικειμένου το οποίο θα προστεθεί. Με την επιλογή του αντικειμένου Button, προστίθεται ένα πλήκτρο στην γραμμή εργαλείων το οποίο μπορεί να συσχετισθεί με μία εικόνα (ιδιότητα Image) την οποία θα απεικονίζει, ή ένα αντικείμενο τύπου label το οποίο θα εμφανίζει κείμενο (ιδιότητα text).

Με διπλό πάτημα του αριστερού πλήκτρου του ποντικιού πάνω σε ένα αντικείμενο της γραμμής εργαλείων μπορούμε να ορίσουμε ένα συμβάν το οποίο προκαλείται με την επιλογή του αντικειμένου και να τα συσχετίσουμε με μία μέθοδο.

12.5 Παράθυρα διαλόγου

Η Visual Basic παρέχει έτοιμα παράθυρα διαλόγου για την χρήση σε διάφορες εφαρμογές. Τα κυριότερα από τα παράθυρα αυτά είναι:

- OpenFileDialog για την επιλογή του ονόματος και της διαδρομής ενός αρχείου για ανάγνωση
- SaveFileDialog για την επιλογή του ονόματος και της διαδρομής ενός αρχείου για αποθήκευση
- FolderBrowserDialog για την επιλογή μιας διαδρομής (καταλόγου)
- FontDialog για την επιλογή μιας γραμματοσειράς
- ColorDialog για την επιλογή ενός χρώματος

Για την δημιουργία ενός παραθύρου διαλόγου και την εμφάνιση του στη οθόνη πρέπει να δημιουργηθεί ένα νέο αντικείμενο της κλάσης που αντιστοιχεί στο παράθυρο διαλόγου και να εμφανισθεί με την μέθοδο showDialog.

```
Dim f As New OpenFileDialog  
f.ShowDialog()
```

Ο χρήστης, ανάλογα με τον τύπο του παραθύρου επιλέγει το όνομα ενός αρχείου, την γραμματοσειρά ή το χρώμα και πατά το πλήκτρο OK ή Ακυρο για να επιβεβαιώσει ή ακυρώσει την επιλογή του. Στην συνέχεια, θα πρέπει να χρησιμοποιηθούν οι παρακάτω ιδιότητες του παραθύρου διαλόγου ώστε να διαβασθεί η επιλογή του χρήστη.

- OpenFileDialog: Η ιδιότητα filename επιστρέφει το όνομα του αρχείου που επέλεξε ο χρήστης
- SaveFileDialog: Η ιδιότητα filename επιστρέφει το όνομα του αρχείου που επέλεξε ο χρήστης
- FolderBrowserDialog: : Η ιδιότητα selectedPath επιστρέφει το όνομα του καταλόγου που επέλεξε ο χρήστης για την επιλογή μιας διαδρομής (καταλόγου)
- FontDialog: Η ιδιότητα Font επιστρέφει την γραμματοσειρά που επέλεξε ο χρήστης
- ColorDialog: Η ιδιότητα Color επιστρέφει το χρώμα που επέλεξε ο χρήστης

Η μέθοδος showResult επιστρέφει επίσης μια τιμή η οποία δείχνει εάν ο χρήστης επιβεβαίωσε ή ακύρωσε την επιλογή του (εάν πατήθηκε το πλήκτρο OK ή Cancel). Σε περίπτωση την οποία ο χρήστης πατήσει τι πλήκτρο OK του παραθύρου διαλόγου, η τιμή αυτή είναι ίση με Windows.Forms.DialogResult.OK.

Το παρακάτω παράδειγμα ζητά από τον χρήστη να επιλέξει το όνομα ενός αρχείου και εφόσον ο χρήστης πατήσει το πλήκτρο OK στο παράθυρο διαλόγου, τότε εμφανίζεται ένα μήνυμα με το όνομα του αρχείου το οποίο επιλέχθηκε.

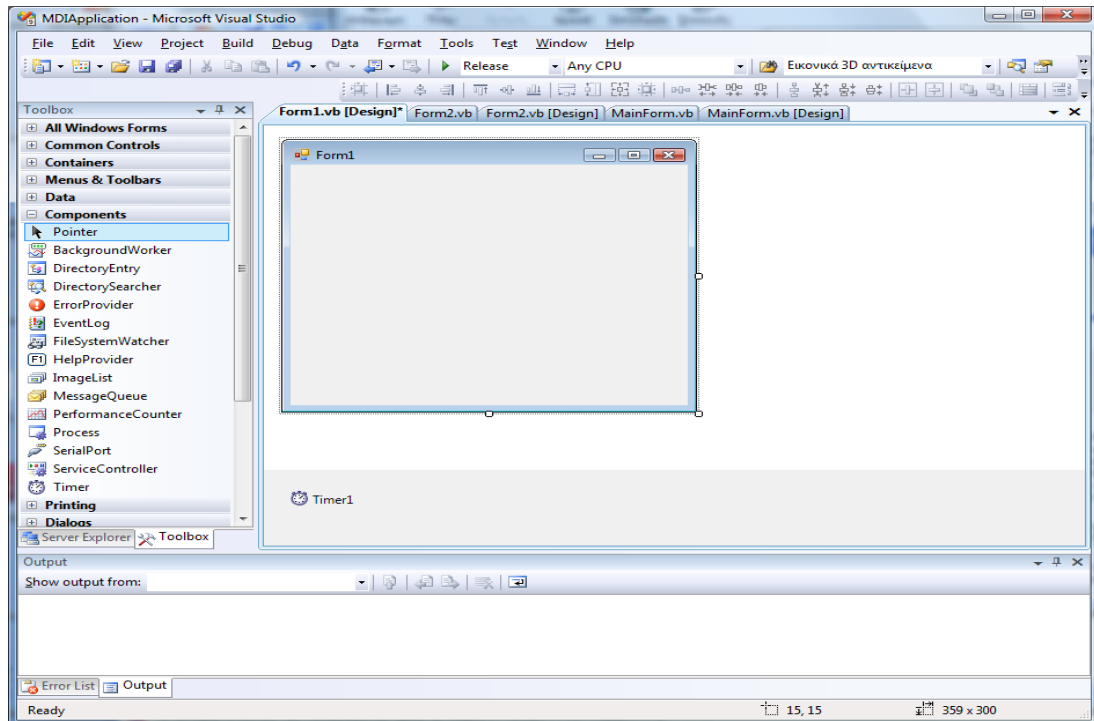
```
Dim f As New OpenFileDialog  
If f.ShowDialog() = Windows.Forms.DialogResult.OK Then  
    f.ShowDialog()  
End If  
MessageBox.Show(f.FileName)
```

Το παρακάτω παράδειγμα επίσης ζητά από τον χρήστη να καθορίσει ένα χρώμα και εάν πατήσει το πλήκτρο OK, τότε καθορίζει το χρώμα υπόβαθρου του παραθύρου στο χρώμα το οποίο επέλεξε ο χρήστης.

```
Dim f As New ColorDialog  
If f.ShowDialog() = Windows.Forms.DialogResult.OK Then  
    Me.BackColor = f.Color  
End If
```

12.6 Χρονόμετρα (Timers)

Το στοιχείο ελέγχου Timer λειτουργεί ως ένα χρονόμετρο το οποίο ενεργοποιεί περιοδικά συμβάντα. Η προσθήκη ενός στοιχείου ελέγχου Timer σε ένα παράθυρο γίνεται με την επιλογή του από την εργαλειοθήκη και την τοποθέτησή του πάνω στο παράθυρο.

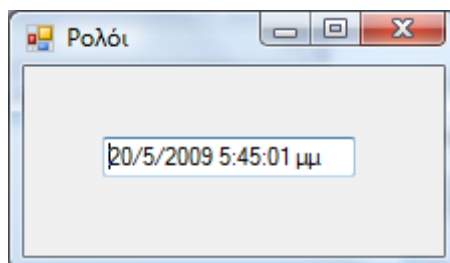


Οι κυριότερες ιδιότητες ενός Timer είναι η ιδιότητα `enabled` η οποία καθορίζει εάν το στοιχείο ελέγχου είναι ενεργοποιημένο ή όχι (τιμή `true` ή `false` αντίστοιχα) και η ιδιότητα `interval` η οποία καθορίζει τον χρόνο ενεργοποίησης (περίοδο) σε χιλιοστά του δευτερολέπτου.

Με το πάτημα του αριστερού πλήκτρου του ποντικιού πάνω στο στοιχείο ελέγχου μπορούμε να ορίσουμε το συμβάν και την μέθοδο η οποία θα καλείται περιοδικά.

Το παρακάτω παράδειγμα χρησιμοποιεί ένα πλαίσιο κειμένου με όνομα `textBox1` και ένα χρονόμετρο με όνομα `Timer1`. Η περίοδος του χρονομέτρου (`Interval`) έχει ορισθεί σε 500. Η μέθοδος η οποία έχει συσχετισθεί με το χρονόμετρο εμφανίζει την τρέχουσα ημέρα και ώρα στο πλαίσιο κειμένου `textBox1`.

```
TextBox1.Text = Date.Now()
```



12.7 Ασκήσεις Κεφαλαίου 12

Άσκηση 12

Α Μέρος

Δημιουργήστε παραθυρική εφαρμογή με όνομα lab_12_ask. Ορίστε το όνομα του αρχικού παραθύρου που εμφανίζεται σε MainForm, τον τίτλο του σε Άσκηση 12, την ιδιότητα isMdiForm σε true και το μέγεθος του (ιδιότητα size) σε 600;400.

Το παράθυρο έχει τρεις κύριες επιλογές μενού με όνομα «Εφαρμογή», «Ασκήσεις» και «Σχετικά» με πλήκτρα πρόσβασης το πρώτο γράμμα του κάθε ονόματος.

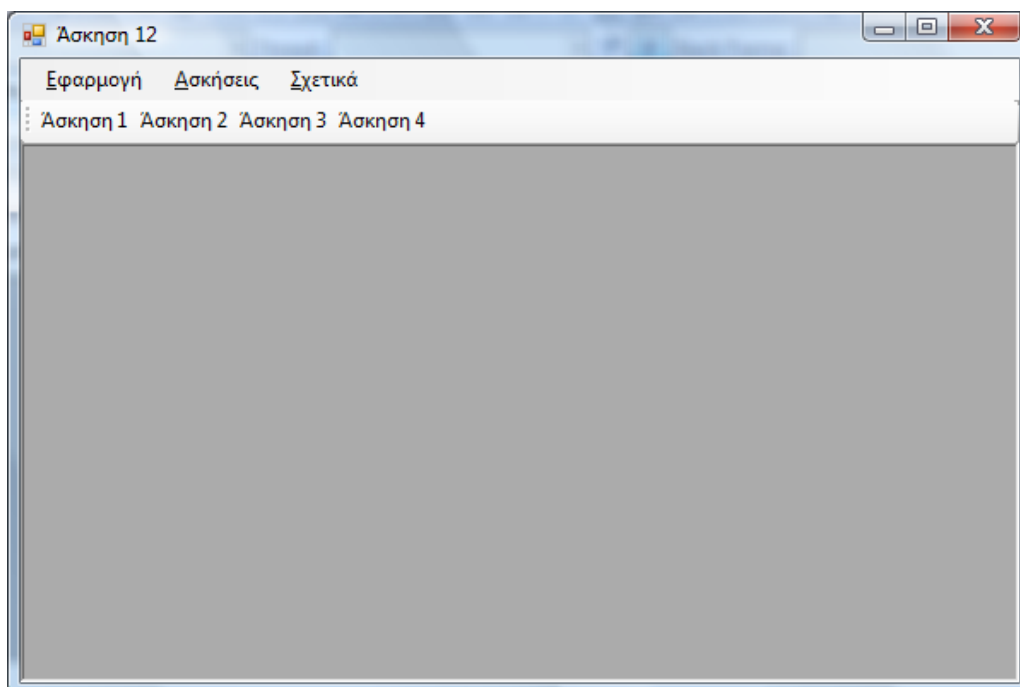
Η επιλογή μενού «Εφαρμογή» έχει ως υπομενού την επιλογή «Τέλος» με πλήκτρο πρόσβασης τον πρώτο χαρακτήρα του και πλήκτρο συντόμευσης Ctrl-X.

Η επιλογή μενού «Ασκήσεις» έχει ως υπομενού τις επιλογές «Άσκηση 1», «Άσκηση 2», «Άσκηση 3» και «Άσκηση 4» με πλήκτρα πρόσβασης 1, 2, 3, και 4 και πλήκτρα συντόμευσης Ctrl-1, Ctrl-2, Ctrl-3, Ctrl-4.

Η επιλογή μενού «Σχετικά» έχει πλήκτρο συντόμευσης F1 και παρουσιάζει ένα παράθυρο με το όνομα σας και την ημερομηνία δημιουργίας της εφαρμογής.

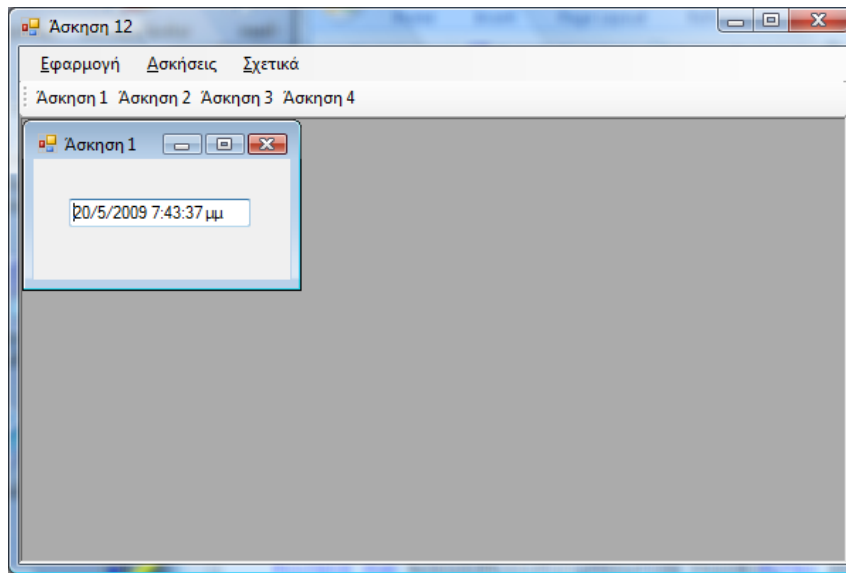
Δημιουργήστε γραμμή εργαλείων όπως φαίνεται στο παρακάτω παράθυρο.

Προσθέστε 4 παράθυρα στην εφαρμογή με τίτλους «Άσκηση 1», «Άσκηση 2», «Άσκηση 3» και «Άσκηση 4» και όνομα FormAskisi1, FormAskisi2, FormAskisi3 και FormAskisi4. Συσχετίστε τις επιλογές του μενού και της γραμμής εργαλείων με το άνοιγμα των παραθύρων με αντίστοιχο τίτλο. Τα παράθυρα θα πρέπει να εμφανίζονται ως παράθυρα που εμπεριέχονται στο MainForm.



Β Μέρος

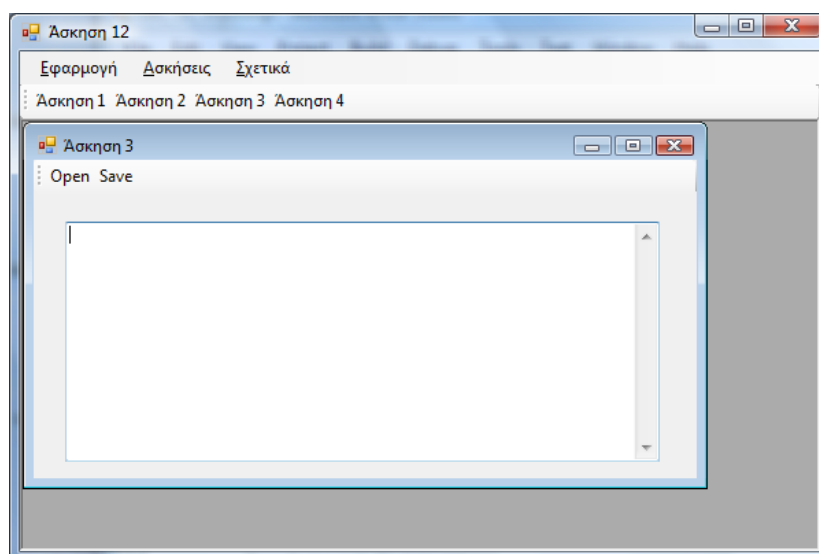
Προσθέστε στοιχείο ελέγχου TextBox με όνομα textbox1 και στοιχείο ελέγχου Timer με όνομα timer1 στο παράθυρο FormAskisi1. Το στοιχείο ελέγχου timer1 έχει ιδιότητα enabled = true και interval=500 και σχετίζεται με μέθοδο η οποία εκτελεί την εντολή `textBox1.text=Date.now`

**Γ Μέρος**

Ορίστε μενού για το παράθυρο FormAskisi2 με επιλογές «Γραμματοσειρά» και «Χρώμα». Προσθέστε στοιχείο ελέγχου TextBox με όνομα textbox1 και ιδιότητα Text = «Κείμενο». Η επιλογή «Γραμματοσειρά» ζητά από τον χρήστη να καθορίσει μια γραμματοσειρά με το παράθυρο διαλόγου FontDialog και εφόσον ο χρήστης πατήσει το πλήκτρο OK στο παράθυρο διαλόγου, τότε αλλάζει την γραμματοσειρά του textbox1. Η επιλογή «Χρώμα» ζητά από τον χρήστη να καθορίσει ένα χρώμα με το παράθυρο διαλόγου colorDialog και εφόσον ο χρήστης πατήσει το πλήκτρο OK στο παράθυρο διαλόγου, τότε ορίζει ως χρώμα υπόβαθρου του παραθύρου (ιδιότητα BackColor) το χρώμα το οποίο επέλεξε ο χρήστης.

Δ Μέρος

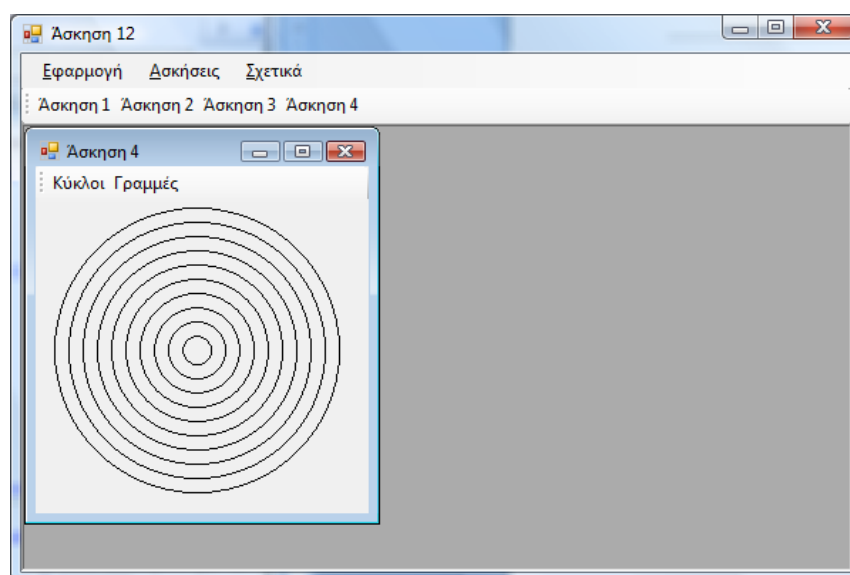
Ορίστε γραμμή εργαλείων για το παράθυρο FormAskisi3 με πλήκτρα «Open» και «Save».



Προσθέστε στοιχείο ελέγχου TextBox με όνομα textbox1, την ιδιότητα multiline= true και την ιδιότητα ScrollBars=Both. Η επιλογή «Save» ζητά από τον χρήστη να δώσει το όνομα ενός αρχείου με την χρήση του SaveFileDialog και αποθηκεύει τα περιεχόμενα του textbox1 στο αρχείο. Η επιλογή «Open» ζητά από τον χρήστη να δώσει το όνομα ενός αρχείου με την χρήση του OpenFileDialog, διαβάζει το αρχείο και εμφανίζει τα περιεχόμενα του αρχείου στο textbox1. Γράψτε κείμενο και αποθηκεύστε το σε αρχείο, κλείστε την εφαρμογή και στην συνέχεια εκτελέστε ξανά την εφαρμογή και διαβάστε το κείμενο από το αρχείο το οποίο δημιουργήσατε.

Ε Μέρος

Ορίστε γραμμή εργαλείων για το παράθυρο FormAskisi4 με πλήκτρα «Κύκλου» και «Γραμμές». Το παράθυρο εμφανίζεται με πλάτος 250, ύψος 280 και περιέχει PictureBox με πλάτος 210 και ύψος 210. Η επιλογή «Κύκλου» σχεδιάζει κύκλους στο PictureBox με κέντρο το σημείο 100,100 και ακτίνες από 10 έως 100 με βήμα 10. Η επιλογή «Γραμμές» σχεδιάζει οριζόντιες γραμμές PictureBox με απόσταση μεταξύ τους 20 pixels.



13 Βάσεις δεδομένων

13.1 Σύνδεση με πίνακα βάσης δεδομένων

Η σύνδεση και το διάβασμα των στοιχείων ενός πίνακα βάσης δεδομένων, γίνεται με την χρήση των κλάσεων DataTable και DataAdapter.

Στο παρακάτω παράδειγμα φαίνεται ο πίνακας Persons μιας βάσης δεδομένων Access, ο οποίος έχει 4 στήλες για την αποθήκευση του κωδικού, του ονόματος, του επώνυμου και του ύψους ατόμων.

aa	onoma	eponymo	ypsos
1	Γεώργιος	Γεωργίου	1,78
2	Πέτρος	Ιωάννου	1,82
3	Μαρία	Παπαδοπούλου	1,65
4	Πέτρος	Πέτρου	1,771

Τα αντικείμενα της κλάσης DataTable μπορούν να χρησιμοποιηθούν για την προσωρινή αποθήκευση και επεξεργασία πινάκων βάσεων δεδομένων στην μνήμη του Η/Υ.

Η κλάση DataAdapter χρησιμοποιείται για το διάβασμα των στοιχείων ενός αντικειμένου DataTable από μια βάση δεδομένων ή για την αποθήκευση των στοιχείων ενός αντικειμένου DataTable στην βάση δεδομένων. Αναλόγως του τύπου της βάσης, χρησιμοποιείται και διαφορετική υποκλάση της κλάσης DataAdapter. Για σύνδεση με βάσεις δεδομένων Access μπορεί να χρησιμοποιηθεί η υποκλάση OleDbDataAdapter.

Ο ορισμός ενός αντικειμένου DataTable για την αποθήκευση των στοιχείων του πίνακα persons γίνεται με την παρακάτω δήλωση. Η δήλωση αυτή ορίζει μια μεταβλητή με όνομα personsTable η οποία αντιστοιχεί σε ένα πίνακα δεδομένων με όνομα "Persons" (ιδιότητα tableName).

```
Dim personsTable As DataTable = new DataTable("Persons")
```

Το διάβασμα των στοιχείων ενός πίνακα γίνεται με την χρήση των παρακάτω εντολών:

```
'Ορισμός παραμέτρων για την σύνδεση με την βάση δεδομένων
Dim connStr As String = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\app\data.mdb"

'Εκφραση SQL για την ανάκτηση δεδομένων
Dim sqlStr As String = "SELECT * FROM Persons"

'Δημιουργία αντικειμένου DataAdapter για την μεταφορά δεδομένων
Dim personsDataAdapter = New _
    System.Data.OleDb.OleDbDataAdapter(sqlStr, connStr)

'Μεταφορά δεδομένων σε αντικείμενο dataTable
personsDataAdapter.Fill(personsTable)

'Διαγραφή αντικειμένου DataAdapter
personsDataAdapter.Dispose()
```

Παράδειγμα 13.1 Διάβασμα των στοιχείων ενός πίνακα

Οι παράμετροι σύνδεσης με την βάση δεδομένων (μεταβλητή `connStr`) ορίζουν τον οδηγό (Provider) ο οποίος θα χρησιμοποιηθεί και το αρχείο (DataSource) της βάσης. Στο συγκεκριμένο παράδειγμα η βάση βρίσκεται στην διαδρομή “c:\app\data.mdb”.

Για το διάβασμα στοιχείων δεδομένων από μια βάση απαιτείται επίσης η χρήση μιας έκφρασης SQL (Structured Query Language) η οποία καθορίζει πια δεδομένα θα διαβασθούν. Στο παράδειγμα 13.1 διαβάζονται όλα τα στοιχεία του πίνακα Persons με την έκφραση “SELECT * FROM Persons” όπου Persons είναι το όνομα του πίνακα του οποίου θέλουμε να διαβάσουμε τα στοιχεία.

Η έκφραση SQL μπορεί να ορίσει συνθήκες τις οποίες πρέπει να ικανοποιούν τα δεδομένα όπως για παράδειγμα:

- Η έκφραση “SELECT * FROM persons where onoma ='Μαρία' ” επιστρέφει όλες τις εγγραφές με όνομα «Μαρία».
- Η έκφραση “SELECT * FROM persons where ypsos >= 1.70 and ypsos <= 1.80” επιστρέφει όλες τις εγγραφές με ύψος μεταξύ 1.70 και 1.80.

Επίσης η έκφραση SQL μπορεί να καθορίσει την ταξινόμηση των εγγραφών όπως για παράδειγμα:

- Η έκφραση “SELECT * FROM persons order by eronymo, onoma” ταξινομεί πρώτα κατά επίθετο και στην συνέχεια κατά όνομα
- Η έκφραση “SELECT * FROM persons order by ypsos desc” ταξινομεί ως προς το ύψος κατά φθίνουσα σειρά.

Η δημιουργία ενός αντικειμένου OleDbDataAdapter γίνεται με την μέθοδο `new` η οποία έχει ως παραμέτρους τα στοιχεία σύνδεσης και την έκφραση SQL. Η μεταφορά των στοιχείων στο αντικείμενο `personTable` γίνεται με την μέθοδο `Fill` του αντικειμένου `PersonsDataAdapter` το οποίο στην συνέχεια το διαγράφουμε με την μέθοδο `dispose`.

Με την χρήση του παραδείγματος 13.1 μπορούμε να ανοίγουμε οποιονδήποτε πίνακα βάσης δεδομένων Access αφού προηγουμένως αλλάξουμε το όνομα της βάσης και το όνομα του πίνακα.

Τα αντικείμενα της κλάσης `DataTable`, μετά το διάβασμα των στοιχείων τους από την βάση δεδομένων μπορεί να χρησιμοποιηθούν για την πρόσβαση, μετατροπή και προσθήκη στοιχείων.

Ένα αντικείμενο της κλάσης `DataTable` είναι ένας πίνακας ο οποίος αποτελείται από γραμμές και στήλες στις οποίες μπορούμε να έχουμε πρόσβαση με τις ιδιότητες `Rows` και `Columns`.

Η ιδιότητα `Rows` μας επιστρέφει μια συλλογή γραμμών τα οποία είναι αντικείμενα της κλάσης `DataRow`.

Η ιδιότητα `Count` της συλλογής `Columns` ενός πίνακα μας επιστρέφει το πλήθος των στηλών ενός πίνακα. Η ιδιότητα `ColumnName` μιας στήλης του πίνακα μας επιστρέφει το όνομα της στήλης και η ιδιότητα `DataType` τον τύπο δεδομένων της στήλης.

Το παρακάτω παράδειγμα εκτυπώνει το πλήθος των στηλών, το όνομα και τον τύπο δεδομένων της πρώτης στήλης του πίνακα.

```
Console.WriteLine("Πλήθος στηλών {0} ", _
    personsTable.Columns.Count)
Console.WriteLine(personsTable.Columns(0).ColumnName)
Console.WriteLine(personsTable.Columns(0).DataType.ToString)
```

Η ιδιότητα `Count` της συλλογής `Rows` μας επιστρέφει το πλήθος των γραμμών.

```
MessageBox.Show("Πλήθος γραμμών ", _
    personsTable.Rows.Count.ToString)
```

Η πρόσβαση στα στοιχεία μιας γραμμής γίνεται με τη χρήση της ιδιότητας `items` ή οποία δέχεται ως παράμετρο τον αριθμό ή το όνομα της στήλης. Μπορεί επίσης να παραλειφθεί η λέξη `items`, όπως φαίνεται στα παρακάτω παραδείγματα, τα οποία αναφέρονται στο ίδιο στοιχείο του πίνακα.

```

MessageBox.Show(personsTable.Rows(0).Item(1))
MessageBox.Show(personsTable.Rows(0).Item("onoma"))
MessageBox.Show(personsTable.Rows(0)(1))
MessageBox.Show(personsTable.Rows(0)("onoma"))
MessageBox.Show(personsTable.Rows(0)!onoma)

```

Η πρόσθεση γραμμών σε ένα πίνακα δεδομένων γίνεται με την χρήση ενός αντικειμένου DataRow το οποίο δημιουργείται του με την μέθοδο NewRow του πίνακα στον οποίο θα προστεθεί, τον καθορισμό των στοιχείων του και την προσθήκη του στις γραμμές του πίνακα, όπως φαίνεται στο παρακάτω παράδειγμα.

```

'Πρόσθεση γραμμής σε πίνακα δεδομένων
Dim aRow As DataRow = personsTable.NewRow
aRow!aa = 10
aRow!onoma = "Νίκος"
aRow!eponymo = "Παπανικολάου"
aRow!ypsos = 1.72
personsTable.Rows.Add(aRow)

```

Η διαγραφή γραμμών του πίνακα δεδομένων γίνεται με την μέθοδο delete της γραμμής.

```

'Διαγραφή της τρίτης γραμμής του πίνακα δεδομένων
personsTable.Rows(2).Delete()

```

Η αποθήκευση των αλλαγών σε ένα πίνακα δεδομένων γίνεται με την μέθοδο update της κλάσης DataAdapter.

Το παρακάτω παράδειγμα δημιουργεί ένα αντικείμενο της κλάσης DataAdapter αντίστοιχο του παραδείγματος 13.1 και χρησιμοποιεί την μέθοδο update για να ενημερώσει τον πίνακα personsTable. Η αποθήκευση των αλλαγών απαιτεί την χρήση ενός αντικειμένου της κλάσης OleDbCommandBuilder το οποίο χρησιμοποιείται για την παραγωγή των κατάλληλων εκφράσεων SQL για την ενημέρωση των στοιχείων. Η μέθοδος update επιστρέφει έναν ακέραιο αριθμό ο οποίος δηλώνει το πλήθος των αλλαγών που έγιναν.

```

Dim connStr As String = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\app\data.mdb"
Dim sqlStr As String = "SELECT * FROM persons"
Dim personsDataAdapter = New
    System.Data.OleDb.OleDbDataAdapter(sqlStr, connStr)
Dim cmdBuilder As New
    System.Data.OleDb.OleDbCommandBuilder(personsDataAdapter)
personsDataAdapter.Update(personsTable)
personsDataAdapter.Dispose()

```

Παράδειγμα 13.2 Αποθήκευση αλλαγών σε βάση δεδομένων

13.2 Ταξινόμηση και αναζήτηση στοιχείων σε πίνακες

Η ταξινόμηση και αναζήτηση στοιχείων σε πίνακες γίνεται με την χρήση της κλάσης DataView. Η κλάση αντικειμένων DataView διαθέτει την μέθοδο Sort για την ταξινόμηση δεδομένων και την μέθοδο RowFilter για την εμφάνιση των εγγραφών οι οποίες ικανοποιούν κάποια συνθήκη.

Το παρακάτω παράδειγμα δημιουργεί ένα αντικείμενο DataView με παράμετρο τον πίνακα personsTable και ορίζει την ταξινόμηση του κατά επώνυμο και όνομα. Η επιθυμητή ταξινόμηση ορίζεται με την παράθεση των επιθυμητών πεδίων (στηλών) του πίνακα για ταξινόμηση. Η ταξινόμηση γίνεται σε αύξουσα σειρά, εκτός αν το όνομα του πεδίου συνοδεύεται από την λέξη Desc όποτε γίνεται σε φθίνουσα σειρά (πχ . personsView.Sort = "ypsos desc").


```

Dim personsView As DataView = New DataView(personsTable)
personsView.Sort = "eponymo,onoma"
Console.WriteLine("*****")
Console.WriteLine("Ταξινομήση δεδομένων")
Console.WriteLine("*****")
For i = 0 To personsView.Count - 1
    Console.WriteLine("{0} {1} {2} {3} ", _
        personsView.Item(i)!aa, _
        personsView.Item(i)!onoma, _
        personsView.Item(i)!Eponymo, _
        personsView.Item(i)!ypsos)
Next

```

Παράδειγμα 13.3 Ταξινόμηση και εκτύπωση στοιχείων πίνακα

Η εύρεση στοιχείων του πίνακα γίνεται με την RowFilter όπως φαίνεται στο παρακάτω παράδειγμα το οποίο ορίζει ως κριτήριο επιλογής των εγγράφων ότι το επώνυμο πρέπει να αρχίζει με το γράμμα Π. Τα κριτήρια επιλογής σχηματίζονται με την χρήση των ονομάτων των πεδίων και συνθηκών με παρόμοια σύνταξη με τις συνθήκες της Visual Basic. Σε περίπτωση που ορίζονται αλφαριθμητικές τιμές όπως στο παρακάτω παράδειγμα, πρέπει να χρησιμοποιούνται αντί των διπλών εισαγωγικών, μονά εισαγωγικά.

```

Dim personsView As DataView = New DataView(personsTable)
personsView.RowFilter = "eponymo like 'Π*'"
Console.WriteLine("*****")
Console.WriteLine("Αναζήτηση δεδομένων")
Console.WriteLine("*****")
For i = 0 To personsView.Count - 1
    Console.WriteLine("{0} {1} {2} {3} ", _
        personsView.Item(i)!aa, _
        personsView.Item(i)!onoma, _
        personsView.Item(i)!Eponymo, _
        personsView.Item(i)!ypsos)
Next

```

Παράδειγμα 13.4 Αναζήτηση και εκτύπωση στοιχείων πίνακα

Όπως φαίνεται και από τα παραπάνω παραδείγματα, η ιδιότητα Count μας επιστρέφει τον αριθμό των γραμμών. Η πρόσβαση στις γραμμές ενός αντικειμένου DataView γίνεται με την χρήση της ιδιότητας item(i) η οποία επιστρέφει την γραμμή με δείκτη i.

Η κλάση αντικειμένων DataTable διαθέτει ένα αντικείμενο της κλάσης DataView στο οποίο μπορούμε να έχουμε πρόσβαση με την μέθοδο DefaultView. Στα παραπάνω παραδείγματα θα μπορούσαμε εναλλακτικά να χρησιμοποιήσουμε την δήλωση:

```
Dim personsView As DataView = PersonsTable.DefaultView
```

Μπορούμε επίσης να χρησιμοποιήσουμε αντί κάποιας μεταβλητής την ιδιότητα PersonsTable.DefaultView (π.χ. PersonsTable.DefaultView.RowFilter="eponymo like 'Π*']").

13.3 Αποθήκευση και ανάγνωση πινάκων από αρχεία XML

Τα στοιχεία ενός πίνακα δεδομένων μπορούν να αποθηκευθούν ή διαβασθούν από αρχεία XML. Η αποθήκευση δεδομένων γίνεται με την μέθοδο writeXML της κλάσης αντικειμένων DataTable όπως φαίνεται στο παρακάτω παράδειγμα.

```

personsTable.WriteXml("c:\app\persons.xml", _
    XmlWriteMode.WriteSchema)

```

Το διάβασμα των στοιχείων ενός πίνακα από αρχείο δεδομένων γίνεται με την μέθοδο ReadXML της κλάσης αντικειμένων DataTable.

```
personsTable = New DataTable("Persons")
personsTable.ReadXml("c:\app\persons.xml")
```

13.4 Χρήση πινάκων χωρίς σύνδεση σε βάση δεδομένων

Ένας πίνακας δεδομένων μπορεί να δημιουργηθεί και χρησιμοποιηθεί χωρίς απαραίτητα να σχετίζεται με κάποια βάση δεδομένων.

Η δημιουργία ενός πίνακα μπορεί να γίνει με την μέθοδο new της κλάσης αντικειμένων DataTable η οποία δέχεται ως παράμετρο το όνομα του πίνακα.

```
'Δημιουργία πίνακα για την αποθήκευση ρίψεων ζαριών
Dim zariData As New DataTable("zariData")
```

Ο ορισμός των στηλών του πίνακα μπορεί να γίνει με την δημιουργία αντικειμένων της κλάσης DataColumn και την πρόσθεση τους στις στήλες (ιδιότητα Columns) του πίνακα. Η δημιουργία ενός αντικειμένου DataColumn γίνεται με παραμέτρους το όνομα της στήλης (του πεδίου του πίνακα) και του τύπου δεδομένων της στήλης του πίνακα.

Το παρακάτω παράδειγμα ορίζει δύο στήλες για τον πίνακα zariData.

```
'Δημιουργία και προσθήκη στηλών στον πίνακα
Dim col1 As New DataColumn("id", GetType(Integer))
zariData.Columns.Add(col1)
Dim col2 As New DataColumn("zari", GetType(Integer))
zariData.Columns.Add(col2)
```

Η πρόσθεση στοιχείων στον πίνακα γίνεται με την δημιουργία γραμμών, τον καθορισμό των στοιχείων τους και την πρόσθεση τους στον πίνακα. Το παρακάτω παράδειγμα προσθέτει στον πίνακα zariData τα αποτελέσματα 6000 ρίψεων του ζαριού.

```
'Αποθήκευση των ρίψεων ζαριών στον πίνακα
For i As Integer = 1 To 6000
    Dim aRow As DataRow = zariData.NewRow
    aRow("id") = i
    Randomize()
    aRow("zari") = CInt(Int(6 * Rnd()) + 1)
    zariData.Rows.Add(aRow)
Next
```

Παράδειγμα 13.5 Πρόσθεση στοιχείων σε πίνακα

Η συνάρτηση Rnd επιστρέφει έναν τυχαίο αριθμό ο οποίος είναι μεγαλύτερος ή ίσος με του μηδενός και μικρότερος της μονάδας. Η έκφραση CInt(Int(6 * Rnd()) + 1) επιστρέφει έναν τυχαίο ακέραιο από το 1 έως το 6. Η συνάρτηση Randomize() χρησιμοποιείται σε συνδυασμό με την συνάρτηση Rnd για να αρχικοποιήσει την γεννήτρια τυχαίων αριθμών.

13.5 Διασύνδεση πινάκων δεδομένων με στοιχεία ελέγχου

Τα στοιχεία ενός πίνακα δεδομένων μπορούν να διασυνδεθούν με τα στοιχεία ελέγχου παραθυρικών εφαρμογών. Στο παρακάτω παράδειγμα, η παραθυρική εφαρμογή χρησιμοποιεί πλαίσια κειμένου με ονόματα textBoxAA, textBoxOnoma, textBoxEponymo, και textBoxYpsos τα οποία εμφανίζουν τα στοιχεία του personsTable. Επίσης το παράθυρο χρησιμοποιεί στοιχείο ελέγχου DataGridView με όνομα gridPersons.

aa	onoma	eponymo	ypsos
1	Γεώργιος	Γεωργίου	1.78
2	Πέτρος	Ιωάννου	1.82
3	Μαρία	Παπαδοπούλου	1.651
4	Πέτρος	Αναγνώστου	1.771
5	Αντώνης	Δημητρίου	1.7845
6	Νίκος	Παπανικολάου	1.7212
7	Ελένη	Γεωργίου	1.68

Η διασύνδεση των πλαισίων κειμένου με τον πίνακα personsTable γίνεται με τις παρακάτω εντολές οι οποίες καθορίζουν ότι η ιδιότητα Text των πλαισίων κειμένου διασυνδέονται με τον πίνακα personsTable στα αντίστοιχα πεδία.

```

TextBoxAA.DataBindings.Add("Text", personsTable, "aa")
TextBoxOnoma.DataBindings.Add("Text", personsTable, "onoma")
TextBoxEponymo.DataBindings.Add("Text", personsTable, "eponymo")
TextBoxYpsos.DataBindings.Add("Text", personsTable, "ypsos")

```

Η διασύνδεση του gridPersons γίνεται με την παρακάτω εντολή η οποία καθορίζει ότι το στοιχείο ελέγχου τροφοδοτείται με δεδομένα από τον πίνακα personsTable.

```

gridPersons.DataSource = personsTable

```

Η διασύνδεση του πίνακα δεδομένων με τα στοιχεία ελέγχου πρέπει να γίνει μετά από το διάβασμα των στοιχείων του πίνακα από την βάση δεδομένων ή από αρχείο XML.

Το πλαίσιο ελέγχου DataGridView επιτρέπει στον χρήστη να τροποποιήσει τα δεδομένα, να προσθέσει νέες εγγραφές πληκτρολογώντας τα στοιχεία της νέας εγγραφής στην τελευταία γραμμή του DataGridView και να διαγράψει μια εγγραφή επιλέγοντας πρώτα ολόκληρη την γραμμή της εγγραφής και πατώντας στην συνέχεια το πλήκτρο Del.

Ο χρήστης μπορεί με το πάτημα του πλήκτρου του ποντικιού πάνω στο όνομα κάποιας στήλης να ταξινομήσει τα δεδομένα ως προς την στήλη αυτή.

Η λειτουργικότητα ενός αντικειμένου DataGridView μπορεί να καθορισθεί από τις παρακάτω ιδιότητες του:

- AllowUserToAddRows: Καθορίζει εάν ο χρήστης μπορεί να προσθέσει νέες γραμμές
- AllowUserToDeleteRows: Καθορίζει εάν ο χρήστης μπορεί να διαγράψει γραμμές
- ReadOnly: Καθορίζει εάν ο χρήστης δεν μπορεί να τροποποιήσει τα δεδομένα.

Η πρόσθεση γραμμών με εντολές (προγραμματιστικά) σε ένα αντικείμενο DataGridView το οποίο είναι διασυνδεδεμένο με ένα πίνακα δεδομένων μπορεί να γίνει με την πρόσθεση μιας νέας γραμμής στο αντικείμενο DataTable με το οποίο είναι διασυνδεδεμένο.

```

'Πρόσθεση νέας εγγραφής
Dim aRow As DataRow = personsTable.NewRow
personsTable.Rows.Add(aRow)

```

Η αφαίρεση γραμμών σε ένα αντικείμενο DataGridView μπορεί να γίνει με την μέθοδο remove της ιδιότητας Rows του αντικειμένου DataGridView.

```

'Διαγραφή τρέχουσας γραμμής DataGridView
gridPersons.Rows.Remove(gridPersons.CurrentRow)

```

13.6 Ασκήσεις Κεφαλαίου 13

Ασκηση 13.1

Δημιουργήστε εφαρμογή τερματικού με όνομα lab_13_ask_1. Το πρόγραμμα:

- Διαβάζει τον πίνακα persons από την βάση c:\app\data.mdb
- Ταξινομεί τα στοιχεία του πίνακα ως προς το επώνυμο και όνομα και τα εκτυπώνει.
- Εκτυπώνει τις εγγραφές του πίνακα των οποίων το επώνυμο αρχίζει με το γράμμα 'Π'.
- Εκτυπώνει:
 - Το πλήθος των γραμμών
 - Το πλήθος των στηλών
 - Το όνομα και τον τύπο δεδομένων των στηλών του πίνακα
 - Τον μέσο όρο του ύψους του πίνακα

Τροποποιήστε το πρόγραμμα ώστε να:

- Διαβάζει τον εκτυπώνει τα στοιχεία ταξινομημένα κατά ύψος σε φθίνουσα σειρά.
- Ζητά από τον χρήστη να εισάγει τα στοιχεία ενός νέου ατόμου και τα προσθέτει στην βάση δεδομένων και αποθηκεύει τις αλλαγές

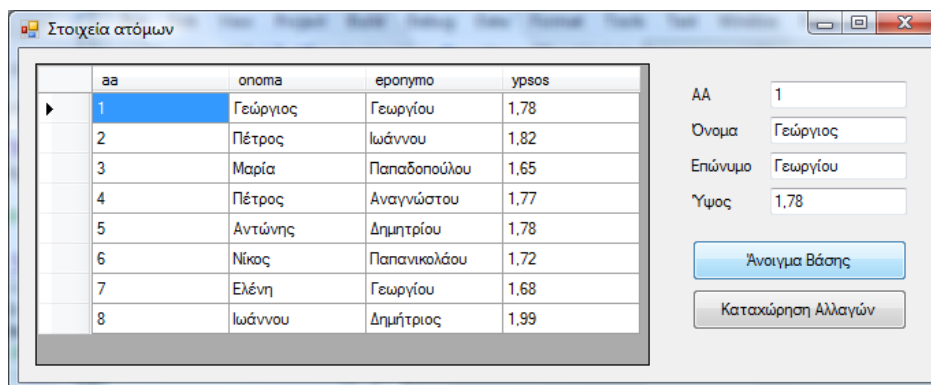
Ασκηση 13.2

Δημιουργήστε εφαρμογή τερματικού όνομα lab_13_ask_2 η οποία ορίζει πίνακα δεδομένων DataTable με όνομα zariData και αποθηκεύει 6000 διαδοχικές ρίψεις ζαριού ως δεδομένα στο πίνακα. (Παράδειγμα 13.5).

Χρησιμοποιήστε την μέθοδο RowFilter και την ιδιότητα Count της κλάσης DataView για να εκτυπώσετε το πλήθος των εμφανίσεων των αριθμών από 1 έως 6 στα δεδομένα του πίνακα.

Ασκηση 13.3

Δημιουργήστε παραθυρική εφαρμογή με όνομα lab_13_ask_3.



Δώστε ονόματα textBoxAA, textBoxOnoma, textBoxEponymo, και textBoxYpsos στα πλαίσια ελέγχου τα οποία εμφανίζουν τα στοιχεία του ατόμου.

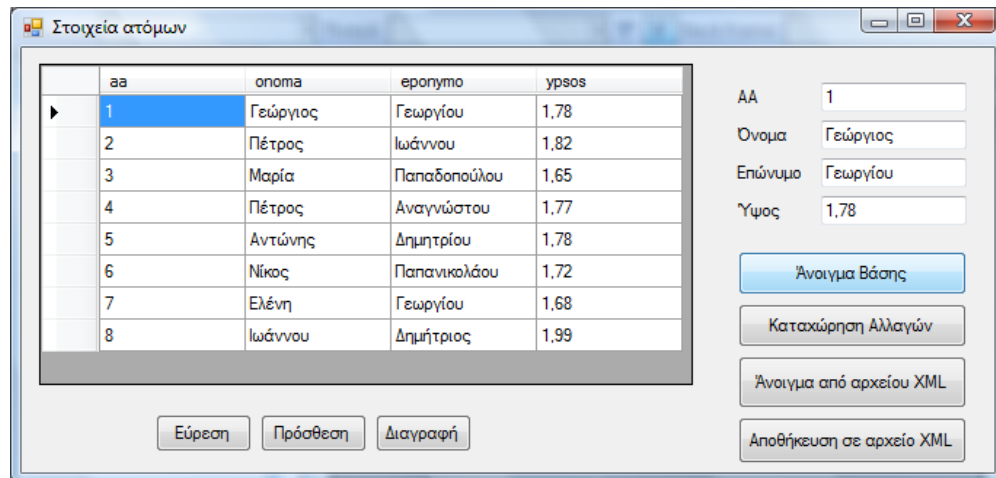
Προσθέστε ένα αντικείμενο DataGridView στο παράθυρο και ορίστε το όνομά του σε personsGrid.

Ορίστε μεταβλητή personsData για το παράθυρο της εφαρμογής η οποία είναι ένα νέο αντικείμενο τύπου DataTable με όνομα πίνακα "Persons".

Το πλήκτρο «Άνοιγμα Βάσης» διαβάζει τα στοιχεία του πίνακα persons από την βάση δεδομένων (Παράδειγμα 13.1) και διασυνδέει τον πίνακα δεδομένων με τα στοιχεία ελέγχου του παραθύρου. Το πλήκτρο «Καταχώρηση αλλαγών» αποθηκεύει τις αλλαγές οι οποίες έγιναν στην βάση δεδομένων (Παράδειγμα 13.2).

Άσκηση 13.4

Τροποποιήστε την παραθυρική lab_13_ask_3 όπως φαίνεται στο παρακάτω παράθυρο.



Ορίστε ότι το αντικείμενο personsGrid δεν επιτρέπει στον χρήστη να προσθέσει νέες γραμμές, να αφαιρέσει γραμμές και να προσθέσει δεδομένα.

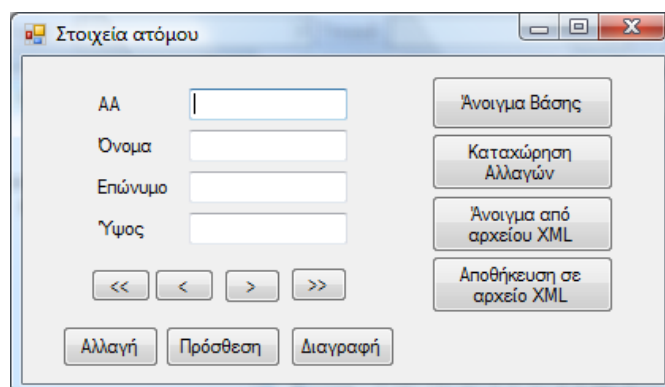
Τα πλήκτρα «Πρόσθεση» και «Διαγραφή» προσθέτουν μια νέα εγγραφή και διαγράφουν την τρέχουσα εγγραφή. Η διαγραφή της τρέχουσας εγγραφής γίνεται αφού πρώτα ο χρήστης ερωτηθεί εάν είναι σίγουρος ότι θέλει να διαγράψει την εγγραφή με κατάλληλο μήνυμα MessageBox.

Το πλήκτρο εύρεση ζητά από την χρήστη να ορίσει μια συνθήκη για τα δεδομένα (π.χ. "eronymo like 'Π*'" ή "ypsos > 1.70") και στην συνέχεια εμφανίζει τις εγγραφές οι οποίες ικανοποιούν την συνθήκη. (Η επανεμφάνιση όλων των εγγραφών γίνεται όταν ο χρήστης δεν ορίσει κάποιο κείμενο για την συνθήκη)

Τα πλήκτρα «Άνοιγμα Αρχείου XML» διαβάζει τα στοιχεία του πίνακα από αρχείο XML και διασυνδέει τον πίνακα δεδομένων με τα στοιχεία ελέγχου του παραθύρου. Το πλήκτρο «Αποθήκευση σε αρχείο XML» αποθηκεύει τα στοιχεία του πίνακα δεδομένων σε αρχείο XML. Το όνομα του αρχείου είναι "c:\app\persons.xml"

Άσκηση 13.2

Δημιουργήστε παραθυρική εφαρμογή με όνομα lab_13_ask_2.



Δώστε ονόματα textBoxAA, textBoxOnoma, textBoxEponymo, και textBoxYpsos στα πλαίσια ελέγχου τα οποία εμφανίζουν τα στοιχεία του ατόμου.

Ορίστε μεταβλητή personsData για το παράθυρο της εφαρμογής η οποία είναι ένα νέο αντικείμενο τύπου DataTable με όνομα πίνακα “Persons”.

Το πλήκτρο «Άνοιγμα Βάσης» διαβάζει τα στοιχεία του πίνακα persons από την βάση δεδομένων (Παράδειγμα 13.1). Το πλήκτρο «Καταχώρηση αλλαγών» αποθηκεύει τις αλλαγές οι οποίες έγιναν στην βάση δεδομένων (Παράδειγμα 13.2).

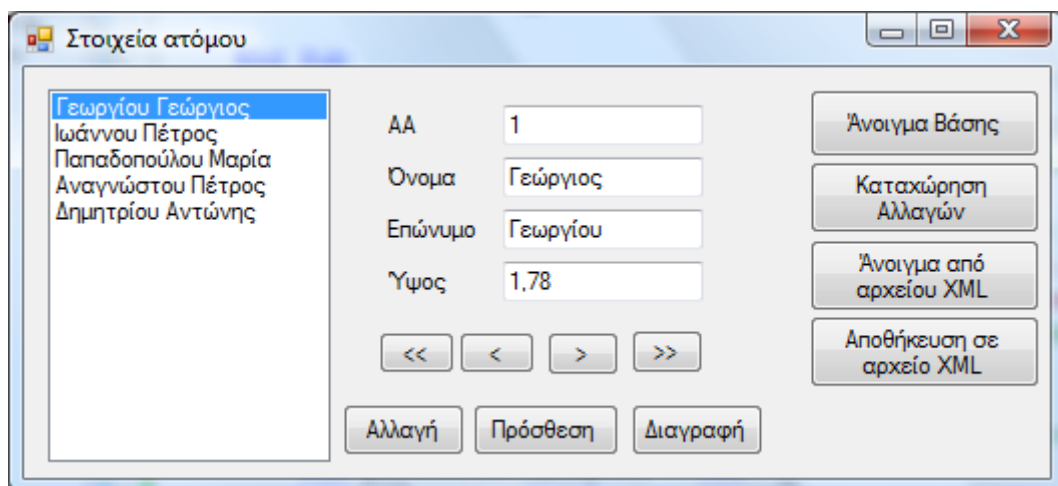
Τα πλήκτρα “<<”, “<”, “>” και “>>” εμφανίζουν τα στοιχεία του πρώτου, του προηγούμενου, του επόμενου και του τελευταίου ατόμου.

Τα πλήκτρα «Αλλαγή», «Πρόσθεση» και «Διαγραφή» αλλάζουν τα στοιχεία της τρέχουσας εγγραφής, προσθέτουν μια νέα εγγραφή και διαγράφουν την τρέχουσα εγγραφή.

Τα πλήκτρα «Άνοιγμα Αρχείου XML» και αποθήκευση αρχείου XML διαβάζουν τα στοιχεία του πίνακα από αρχείο XML και αποθηκεύουν τα στοιχεία του πίνακα σε αρχείο XML. τα στοιχεία του πίνακα persons από την βάση δεδομένων. Το όνομα του αρχείου είναι “c:\app\persons.xml”

Άσκηση 13.2

Τροποποιήστε την παραθυρική εφαρμογή με όνομα lab_13_ask_2 ώστε να εμφανίζει κατάλογο επιλογής (ListBox) με όνομα ListBoxNames ο οποίος εμφανίζει το επίθετο και όνομα των ατόμων. Ο χρήστης μπορεί να χρησιμοποιήσει τον κατάλογο για να επιλέξει κάποιο άτομο και να εμφανίσει τα στοιχεία του.



Το πρόγραμμα διαβάζει τον πίνακα persons από την βάση c:\app\data

14 Εφαρμογές WEB

14.1 Δημιουργία εφαρμογών WEB

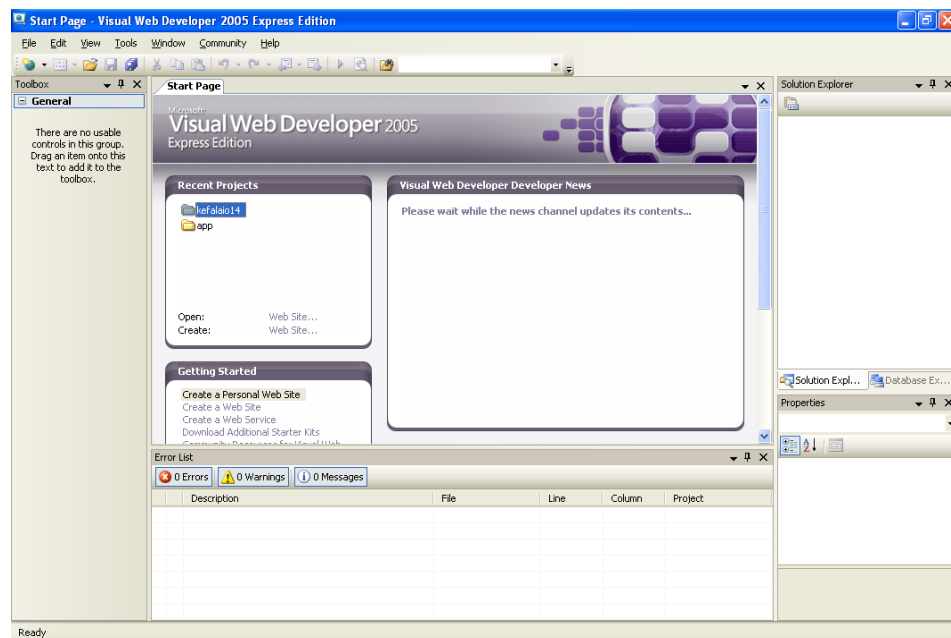
Η δημιουργία εφαρμογών WEB με την γλώσσα προγραμματισμού Visual Basic μπορεί να γίνει με την χρήση του περιβάλλοντος προγραμματισμού Microsoft Visual Web Developer 2005 Express Edition το οποίο είναι διαθέσιμο ελεύθερα στην τοποθεσία:

<http://microsoft-visual-web-developer-2005-express-edition.en.softonic.com/>

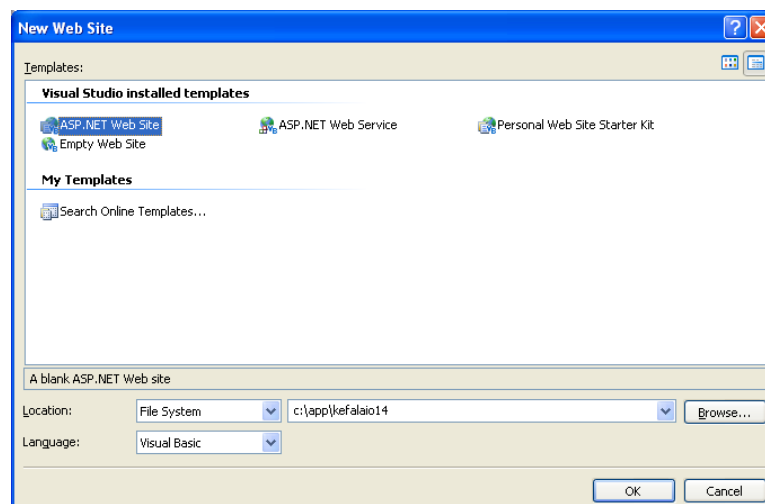
ή μπορεί εναλλακτικά να χρησιμοποιηθεί η τρέχουσα έκδοση του 2008 από την τοποθεσία:

<http://www.microsoft.com/express/vwd/>.

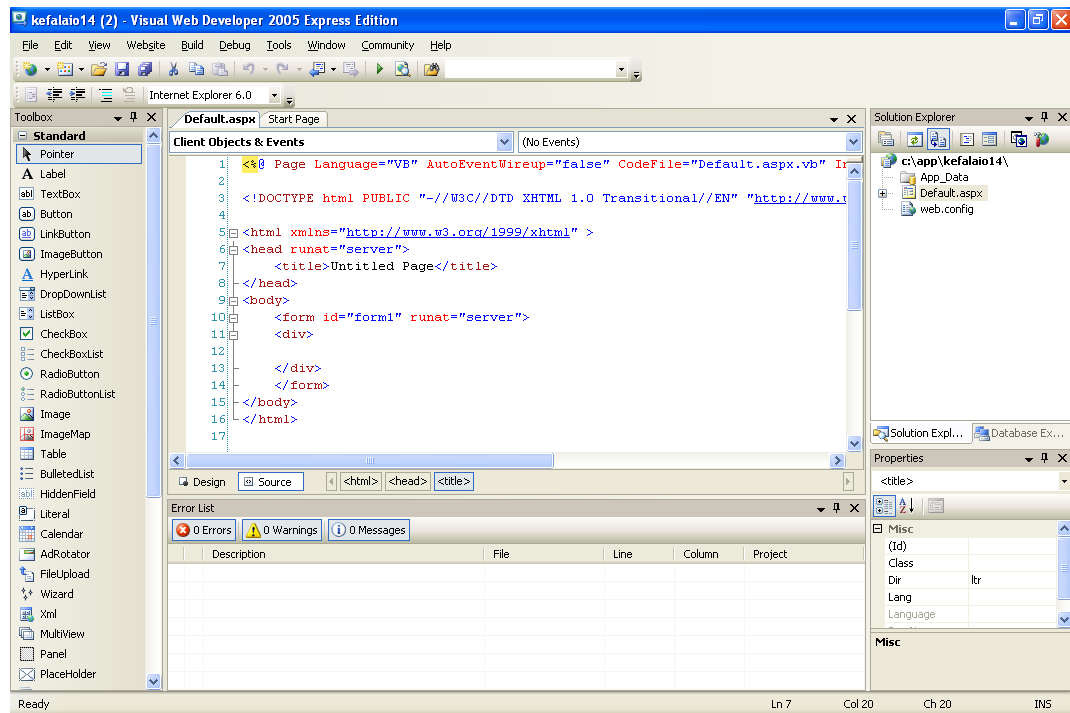
Το περιβάλλον προγραμματισμού Visual Web Developer ξεκινά με το πάτημα του αντίστοιχου εικονιδίου της επιφάνειας εργασίας ή την επιλογή μενού από τα προγράμματα του Η/Υ όπως φαίνεται στην παρακάτω εικόνα.



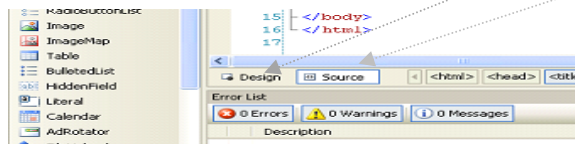
Η δημιουργία εφαρμογών WEB με την Visual Basic γίνεται με την επιλογή New Web Site του μενού File, οπότε και εμφανίζεται το παρακάτω παράθυρο διαλόγου.



Στο παράθυρο διαλόγου επιλέξτε ως Template την επιλογή ASP NET Web Site, Location: File System, Language: Visual Basic και κατάλογο c:\app\kefalaio14. Με το πάτημα του πλήκτρου OK δημιουργείται μια νέα τοποθεσία WEB για προγραμματισμό σε Visual Basic, όπως φαίνεται στην παρακάτω εικόνα.



Στην συνέχεια εμφανίζεται στην οθόνη μια ιστοσελίδα με όνομα default.aspx η οποία παρουσιάζεται αρχικά σε προβολή κώδικα. Με την χρήση των πλήκτρων «Design» και «Source» όπως φαίνονται στην παρακάτω εικόνα μπορούμε να επιλέξουμε την προβολή σχεδίασης ή την προβολή κώδικα αντίστοιχα.



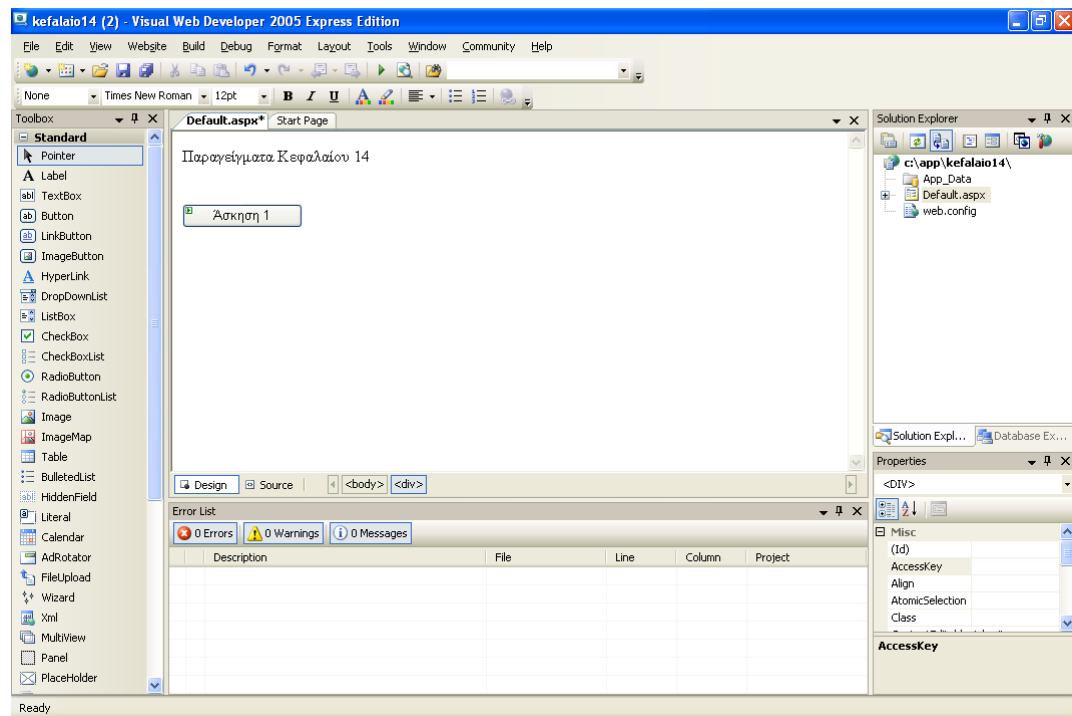
Η προβολή κώδικα εμφανίζει τον κώδικα HTML ο οποίος αντιστοιχεί στην ιστοσελίδα default.aspx. Μπορούμε να ορίσουμε την ιστοσελίδα χρησιμοποιώντας κώδικα HTML και δηλώσεις asp όπως φαίνεται στην παρακάτω εικόνα όπου ορίζεται ο τίτλος της ιστοσελίδας και ένα αρχικό κείμενο για την ιστοσελίδα.

```

4
5 <html xmlns="http://www.w3.org/1999/xhtml" >
6 <head runat="server">
7     <title>Κεφάλαιο 14</title>
8 </head>
9 <body>
10     <form id="form1" runat="server">
11         <div>
12             Παραδείγματα Κεφαλαίου 14
13         </div>
14     </form>
15 </body>
16 </html>

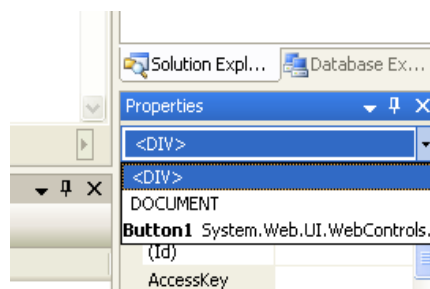
```


Με την προβολή σχεδίασης ή οποία φαίνεται στην παρακάτω εικόνα, μπορούμε να προεπισκοπήσουμε την ιστοσελίδα αλλά και να την σχεδιάσουμε, όπως σχεδιάζουμε και τις παραθυρικές εφαρμογές.

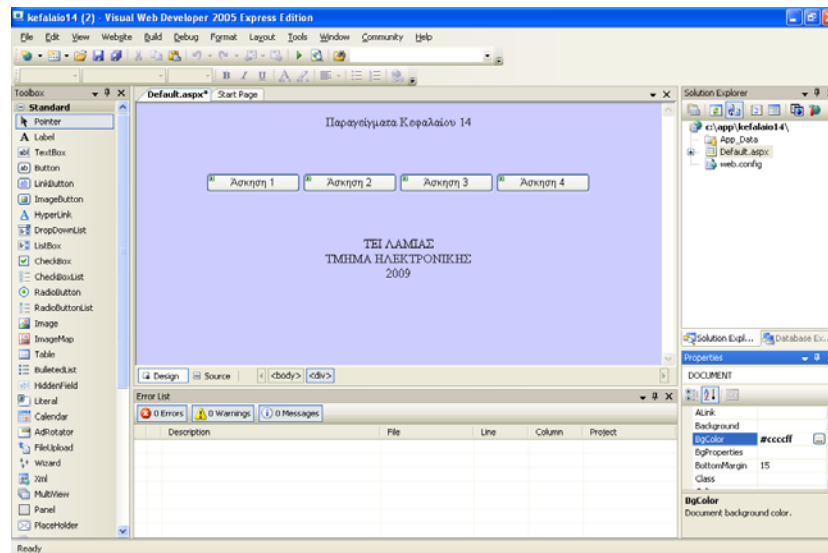


Σε μια ιστοσελίδα μπορούμε να εισάγουμε και να μορφοποιήσουμε κείμενο, όπως γίνεται και με έναν επεξεργαστή κειμένου. Μπορούμε επίσης, κάνοντας χρήση της εργαλειοθήκης να προσθέσουμε διάφορα στοιχεία ελέγχου στην ιστοσελίδα, όπως για παράδειγμα πλήκτρα, πλαίσια κειμένου, ετικέτες κ.λπ.. Τα στοιχεία ελέγχου τα οποία μπορούμε να χρησιμοποιήσουμε για να σχεδιάσουμε μια ιστοσελίδα είναι παρόμοια με τα στοιχεία ελέγχου των παραθυρικών εφαρμογών.

Μπορούμε κάνοντας χρήση του παραθύρου των ιδιοτήτων να ορίσουμε διάφορες ιδιότητες για τα πλαίσια ελέγχου, όπως για παράδειγμα την ιδιότητα text και την ιδιότητα width ενός πλήκτρου. Μπορούμε επίσης να ορίσουμε ιδιότητες και για διάφορα άλλα στοιχεία της ιστοσελίδας, όπως για παράδειγμα το στοιχείο <DIV> το οποίο ορίζει τις ιδιότητες της τρέχουσας παραγράφου (τμήματος ιστοσελίδας) και του στοιχείου DOCUMENT το οποίο καθορίζει τις ιδιότητες της ιστοσελίδας (title – τίτλος, bgcolor – χρώμα υπόβαθρου, background – εικόνα υπόβαθρου, κ.α.)



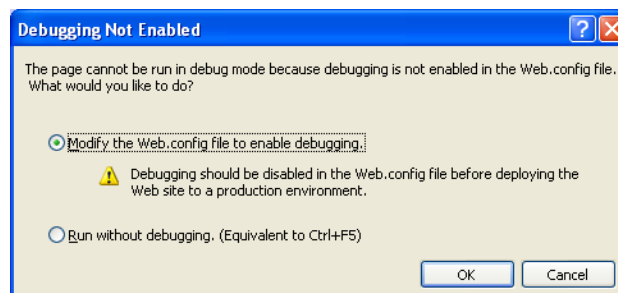
Στο παρακάτω εικόνα όπου φαίνεται η ιστοσελίδα default.aspx έχουν προστεθεί 4 πλήκτρα με ονόματα Button1, Button2, Button3 και Button4, έχει καθορισθεί η ιδιότητα width για το κάθε πλήκτρο σε 120px και έχει καθορισθεί η ιδιότητα text σε «Άσκηση 1», «Άσκηση 2», «Άσκηση 3» και «Άσκηση 4» αντίστοιχα. Επίσης πληκτρολογήθηκαν τρεις σειρές κειμένου στην προβολή σχεδίασης στο τέλος της ιστοσελίδας («ΤΕΙ ΛΑΜΙΑΣ ... 2009») και ορίσθηκε η ιδιότητα align = center για το στοιχείο <DIV> και η ιδιότητα bgcolor = #ccccff για το στοιχείο DOCUMENT.



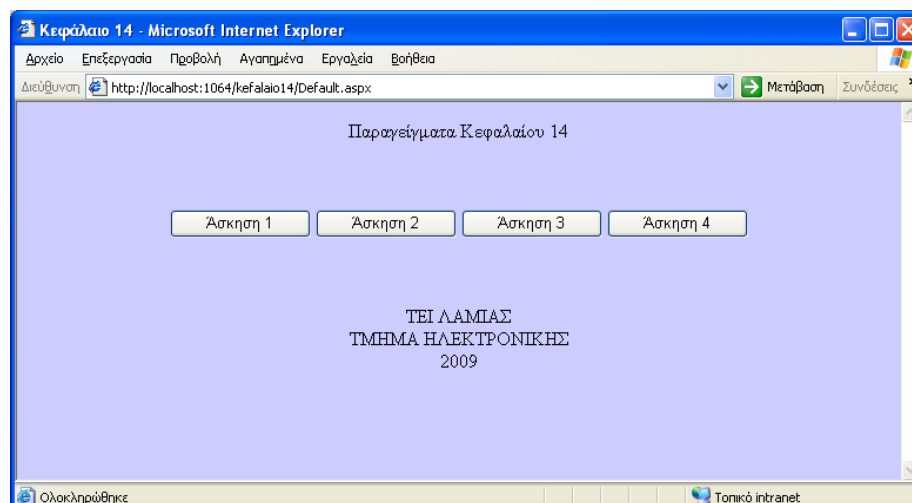
14.2 Εκτέλεση εφαρμογών WEB

Η εκτέλεση εφαρμογών WEB από το περιβάλλον ανάπτυξης γίνεται όπως και για τις εφαρμογές της Visual Basic με την επιλογή Start Debugging από το μενού Debug ή την συντόμευση F5.

Κατά την πρώτη εκτέλεση της εφαρμογής μπορεί να εμφανισθεί το παρακάτω παράθυρο διαλόγου, το οποίο μας ζητά την τροποποίηση των παραμέτρων της εφαρμογής για την εκτέλεση της και αποσφαλμάτωση της (debug).



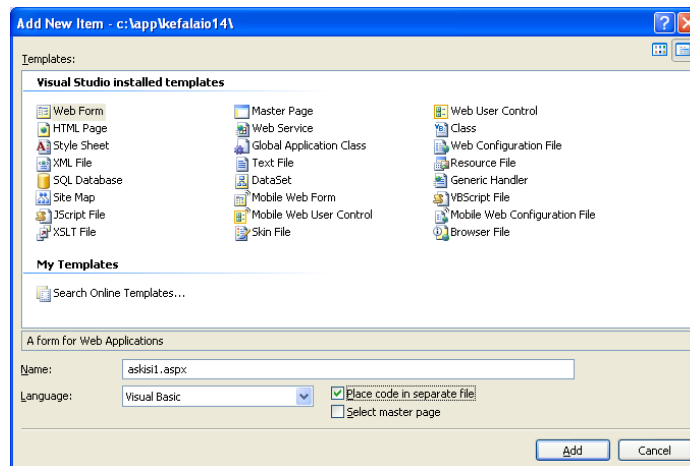
Πατώντας το πλήκτρο «OK» εμφανίζεται ένα παράθυρο του Internet Explorer ο οποίος προβάλλει την ιστοσελίδα όπως θα εμφανιζόταν εάν είχαμε δημοσιεύσαμε την ιστοσελίδα στο διαδίκτυο και την προβάλλαμε με έναν Web Browser.



14.3 Προσθήκη Ιστοσελίδων σε μια εφαρμογή

Με την επιλογή 'Add New Item' του μενού 'Web Site' μπορούμε να προσθέσουμε ιστοσελίδες σε μια εφαρμογή.

Αρχικά εμφανίζεται το παρακάτω παράθυρο διαλόγου στο οποίο πρέπει να επιλέξουμε ως template την επιλογή Web Form, ως όνομα το όνομα της νέας ιστοσελίδας, ως Language Visual Basic και να επιλέξουμε ότι ο κώδικας της εφαρμογής τοποθετείται σε ξεχωριστό αρχείο (Place code in separate file).

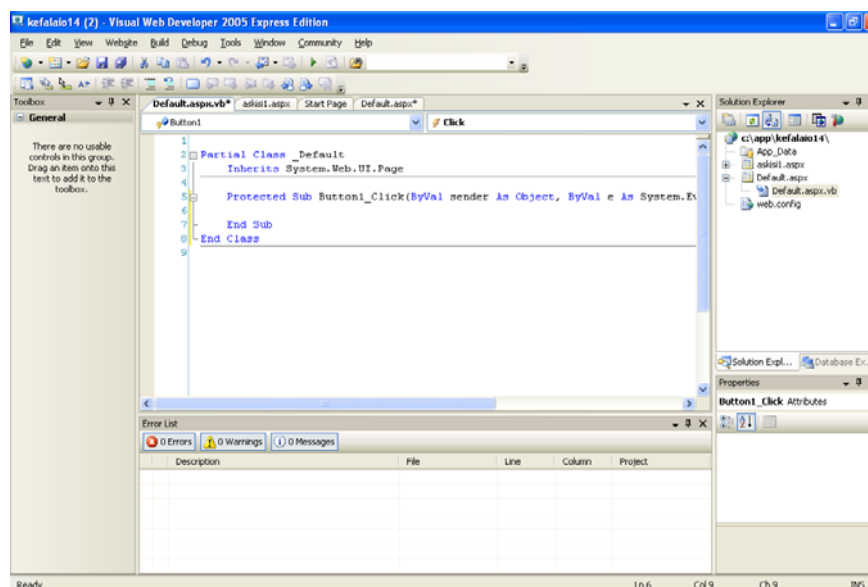


Πατώντας το πλήκτρο «Add», δημιουργείται μια νέα ιστοσελίδα με το όνομα που επιλέξαμε και προστίθεται στην εφαρμογή.

14.4 Προγραμματισμός Ιστοσελίδων

Όπως γίνεται και με τις παραθυρικές εφαρμογές, μπορούμε να διασυνδέσουμε τα στοιχεία ελέγχου μιας ιστοσελίδας αλλά και την ίδια την ιστοσελίδα με υποπρογράμματα Visual Basic τα οποία καλούνται και εκτελούνται μέσω διαφόρων συμβάντων τα οποία ορίζουμε (πάτημα ενός πλήκτρου, άνοιγμα της ιστοσελίδας, επιλογή ενός αντικειμένου από μια λίστα επιλογής κ.λπ.).

Με το διπλό πάτημα του αριστερού πλήκτρου του ποντικιού πάνω στο πλήκτρο «Άσκηση 1» της ιστοσελίδας default.aspx, ορίζεται ένα νέο συμβάν και μέθοδος για το πάτημα του πλήκτρου και εμφανίζεται το αρχείο κώδικα της Visual Basic με όνομα default.aspx.vb.

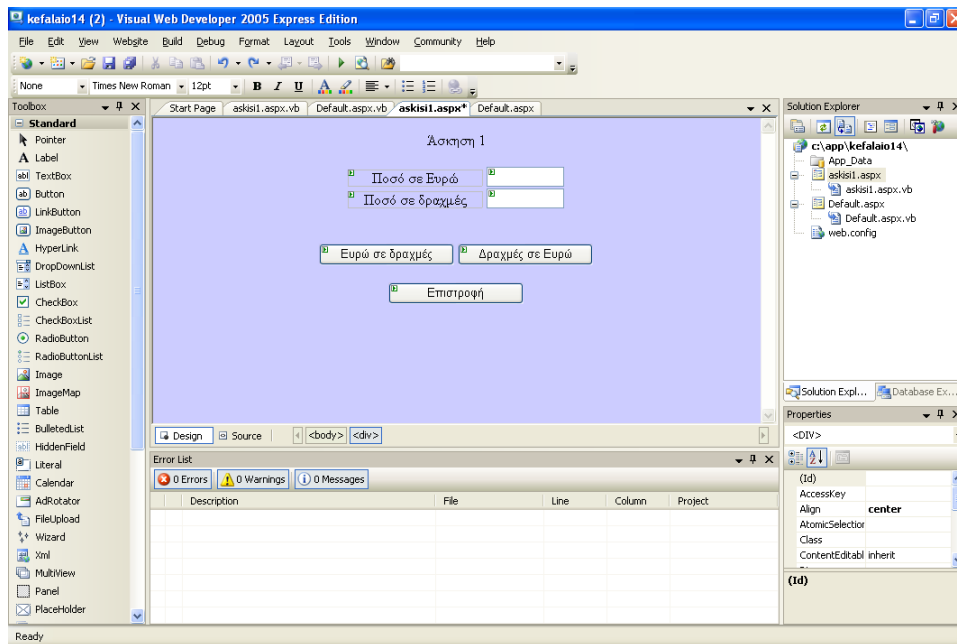


Η εμφάνιση μίας ιστοσελίδας από κάποια άλλη μπορεί να γίνει με την μέθοδο `redirect` της κλάσης αντικειμένων `Response`.

Εάν για παράδειγμα θέλουμε να εμφανίσουμε την ιστοσελίδα `askisil.aspx`, αντιστοιχούμε την παρακάτω εντολή στο πλήκτρο «Άσκηση 1»:

```
Response.Redirect("askisil.aspx")
```

Στο παρακάτω παράδειγμα, η ιστοσελίδα `askisil.aspx` έχει δύο ετικέτες (labels) με την ιδιότητα `width = 150` και δύο πλαίσια κειμένου με ονόματα `textBoxEuro` και `textBoxDrachmas` με την ιδιότητα `width=80`. Επίσης ορίσθηκε η ιδιότητα `align = center` για το στοιχείο `<DIV>` και η ιδιότητα `bgcolor = #ccccff` για το στοιχείο `DOCUMENT`.



Η ιστοσελίδα `askisil.aspx` μπορεί να χρησιμοποιηθεί για την μετατροπή ευρώ σε δραχμές και αντίστροφα.

Για την μετατροπή ευρώ σε δραχμές έχει αντιστοιχισθεί η παρακάτω εντολή για το πάτημα του πλήκτρου «Ευρώ σε δραχμές»:

```
textBoxdrachmas.Text = TextBoxEuro.Text * 340.75
```

Για την μετατροπή δραχμών σε ευρώ έχει αντιστοιχισθεί η παρακάτω εντολή για το πάτημα του πλήκτρου «Δραχμές σε ευρώ»:

```
TextBoxEuro.Text = TextBoxDrachmas.Text / 340.75
```

Το πλήκτρο «Επιστροφή» εμφανίζει την αρχική σελίδα `default.aspx` με την παρακάτω εντολή:

```
Response.Redirect("default.aspx")
```

14.5 Εμφάνιση μηνυμάτων

Η εμφάνιση μηνυμάτων στον χρήστη σε εφαρμογές WEB δεν μπορεί να γίνει με την χρήση της κλάσης `MessageBox`. Επίσης δεν είναι διαθέσιμη ούτε η μέθοδος `InputBox` για την είσοδο δεδομένων.

Η εμφάνιση ενός μηνύματος μπορεί να γίνει μέσω της μεθόδου `window.alert` ενός javascript το οποίο δημιουργείται δυναμικά με την μέθοδο `Response.write`.

Στο παρακάτω παράδειγμα, γίνεται έλεγχος κατά το πάτημα του πλήκτρο «Ευρώ σε δραχμές» εάν έχει γίνει η εισαγωγή κάποιου ποσού στο πλαίσιο κειμένου `TextBoxEuro` και σε περίπτωση που δεν έχει γίνει εισαγωγή ποσού, τότε εμφανίζεται το μήνυμα «Δεν έχει καθορισθεί ποσό ευρώ».

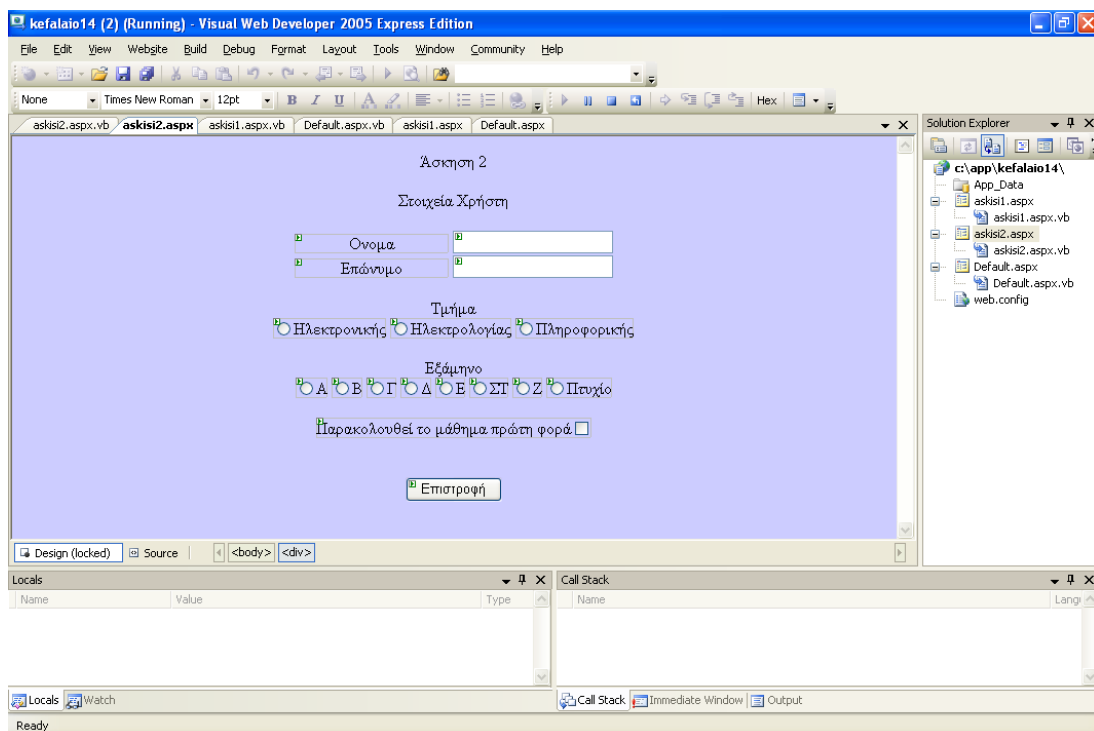
```
If TextBoxEuro.Text = "" Then
    Response.Write("<script language='javascript'>" + _
        "window.alert('Δεν έχει καθορισθεί ποσό ευρώ');" + _
        "</script>")
Else
    TextBoxDrachmas.Text = TextBoxEuro.Text * 340.75
End If
```

14.6 Στοιχεία ελέγχου

Τα στοιχεία ελέγχου πλαίσιο επιλογής, πλήκτρο επιλογής, λίστα επιλογής και αναδιπλούμενος κατάλογος επιλογής, τα οποία είναι διαθέσιμα για παραθυρικές εφαρμογές, είναι επίσης διαθέσιμα και για τις ιστοσελίδες με κάποιες όμως διαφορές.

Η κύρια διαφορά για το πλήκτρο επιλογής (`RadioButton`) είναι ότι η ομαδοποίηση, ώστε μόνο ένα πλήκτρο της ομάδας να μπορεί να επιλεγεί, γίνεται με την ιδιότητα `GroupName` αντί της τοποθέτησης του πλήκτρου επιλογής σε ένα ένα panel.

Στο παρακάτω παράδειγμα φαίνεται η ιστοσελίδα `askisi2.aspx` η οποία χρησιμοποιείται για τον καθορισμό των στοιχείων του χρήστη. Τα τρία πρώτα πλήκτρα επιλογής τα οποία χρησιμοποιούνται για την επιλογή τμήματος έχουν την ιδιότητα `GroupName = Τμήμα` και τα υπόλοιπα πλήκτρα τα οποία χρησιμοποιούνται για την επιλογή του εξαμήνου έχουν την ιδιότητα `GroupName= Εξάμηνο`.



Το πλαίσιο επιλογής (checkbox) δεν έχει την δυνατότητα τριών καταστάσεων (ιδιότητες `ThreeState` και `CheckedState`).

Επίσης, η λίστα επιλογής και ο αναδιπλούμενος κατάλογος επιλογής για την σωστή λειτουργία τους πρέπει να έχουν την ιδιότητα τους `AutoPostBack = true`.

14.7 Κοινές μεταβλητές ιστοσελίδων

Η χρήση κοινών μεταβλητών μεταξύ διάφορων ιστοσελίδων σε εφαρμογές WEB μπορεί να γίνει με την χρήση του αντικειμένου Session. Το αντικείμενο Session είναι μια συλλογή η οποία αποθηκεύει διάφορα στοιχεία στα οποία μπορούν να έχουν πρόσβαση οι ιστοσελίδες.

Αν για παράδειγμα επιθυμούμε τα στοιχεία χρήστη τα οποία εισάγονται στην ιστοσελίδα `default.aspx` να εμφανίζονται σε μία ετικέτα της ιστοσελίδας `default.aspx`, τότε ορίζουμε δύο νέα στοιχεία της συλλογής session κατά το πάτημα του πλήκτρου «Επιστροφή» της ιστοσελίδας `askisi2.aspx`

```
Session("Name") = TextBoxName.Text
Session("Surname") = TextBoxName.Text
Response.Redirect("default.aspx")
```

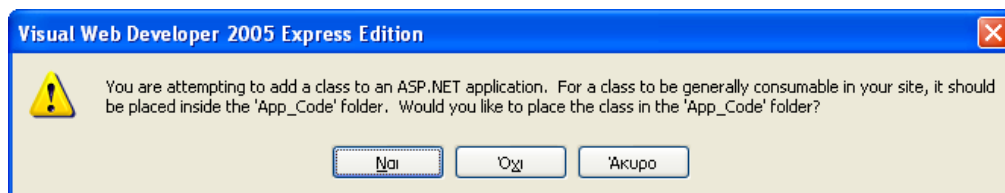
Για την εμφάνιση των στοιχείων του χρήστη στην ιστοσελίδα `default.aspx`, τοποθετούμε μια νέα ετικέτα με όνομα `labelUserName` και ορίζουμε ένα νέο συμβάν κατά το άνοιγμα της ιστοσελίδας (με διπλό πάτημα του πλήκτρου του ποντικιού πάνω στην ιστοσελίδα) το οποίο εκτελεί την παρακάτω εντολή:

```
LabelUserName.Text = Session("name") + " " + Session("surname")
```

14.8 Πρόσθεση κλάσεων αντικειμένων σε εφαρμογές WEB

Η προσθήκη κλάσεων αντικειμένων σε εφαρμογές WEB μπορεί να γίνει με τον ίδιο τρόπο με τον οποίο προσθέτουμε κλάσεις και σε μια παραθυρική εφαρμογή, με την διαφορά ότι η κλάση πρέπει να τοποθετηθεί σε ξεχωριστό κατάλογο με όνομα `App_Code`.

Η δημιουργία μιας νέας κλάσης αντικειμένων μπορεί να γίνει με την επιλογή `Add new Item` από το μενού `Web Site` και στην συνέχεια με την επιλογή `Class` ως `Template`. Στην συνέχεια εμφανίζεται ένα παράθυρο διαλόγου το οποίο προτρέπει να τοποθετηθεί η νέα κλάση στον κατάλογο `App_Code`. Για να είναι προσβάσιμο το αντικείμενο από την εφαρμογή, πρέπει να επιλεγεί το πλήκτρο «Ναι».

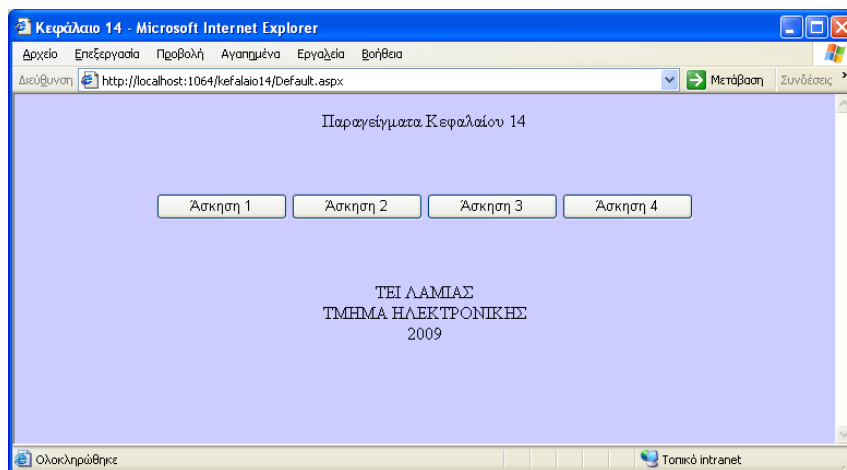


14.9 Ασκήσεις Κεφαλαίου 14

Άσκηση 14

Τμήμα Α

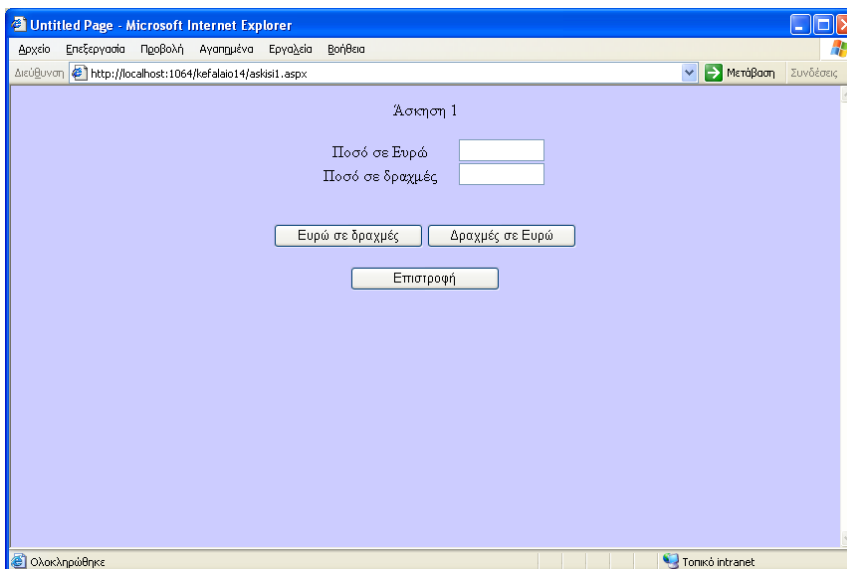
Δημιουργήστε εφαρμογή WEB με αρχική ιστοσελίδα default.aspx όπως φαίνεται στην παρακάτω εικόνα:



Προσθέστε στην εφαρμογή ιστοσελίδες με ονόματα askisi1.aspx, askisi2.aspx, askisi3.aspx και askisi4.aspx, οι οποίες αρχικά είναι κενές. Τα πλήκτρα «Άσκηση 1», «Άσκηση 2», «Άσκηση 3», και «Άσκηση 4» εμφανίζουν τις αντίστοιχες ιστοσελίδες.

Η ιστοσελίδα askisi1.aspx, όπως φαίνεται στην παρακάτω εικόνα, ορίζει πλαίσια κειμένου με ονόματα textBoxEuro και textBoxDrachmas. Με το πάτημα των πλήκτρων «Ευρώ σε δραχμές» και «Δραχμές σε Ευρώ» γίνεται η μετατροπή του ποσού από ευρώ σε δραχμές και αντίστροφα.

Σε περίπτωση μετατροπής κάποιου ποσού, εάν δεν έχει ορισθεί το ποσό το οποίο θα μετατραπεί, τότε εμφανίζεται μήνυμα ότι δεν έχει καθορισθεί το ποσό και η μετατροπή δεν γίνεται.



Το πλήκτρο «Επιστροφή» εμφανίζει την αρχική ιστοσελίδα με όνομα “default.aspx”.

Τμήμα Β

Η ιστοσελίδα askisi2.aspx, η οποία φαίνεται στην παρακάτω εικόνα, χρησιμοποιείται για την εισαγωγή των στοιχείων του σπουδαστή.

Με το πάτημα του πλήκτρου «Επιστροφή», τα στοιχεία τα οποία εισήγαγε ο χρήστης εμφανίζονται σε ετικέτα (label) με όνομα labelUserName του αρχικού παραθύρου default.aspx, όπως φαίνεται στην παρακάτω εικόνα.

Χρησιμοποιήστε αρχικά τα παραδείγματα της θεωρίας για την κοινή χρήση μεταβλητών και την μεταφορά του ονόματος και του επιθέτου.

Τροποποίηση:

Τροποποιήστε την εφαρμογή ώστε τα στοιχεία της ιστοσελίδας askisi2.aspx να αποθηκεύονται σε αντικείμενο μιας νέας κλάσης Person την οποία θα προσθέσετε στην εφαρμογή. Η κλάση Person έχει τα κατάλληλα πεδία ώστε να αποθηκεύει το όνομα, το επίθετο, την Σχολή, το εξάμηνο και το εάν ο σπουδαστής παρακολουθεί το μάθημα για πρώτη φορά.

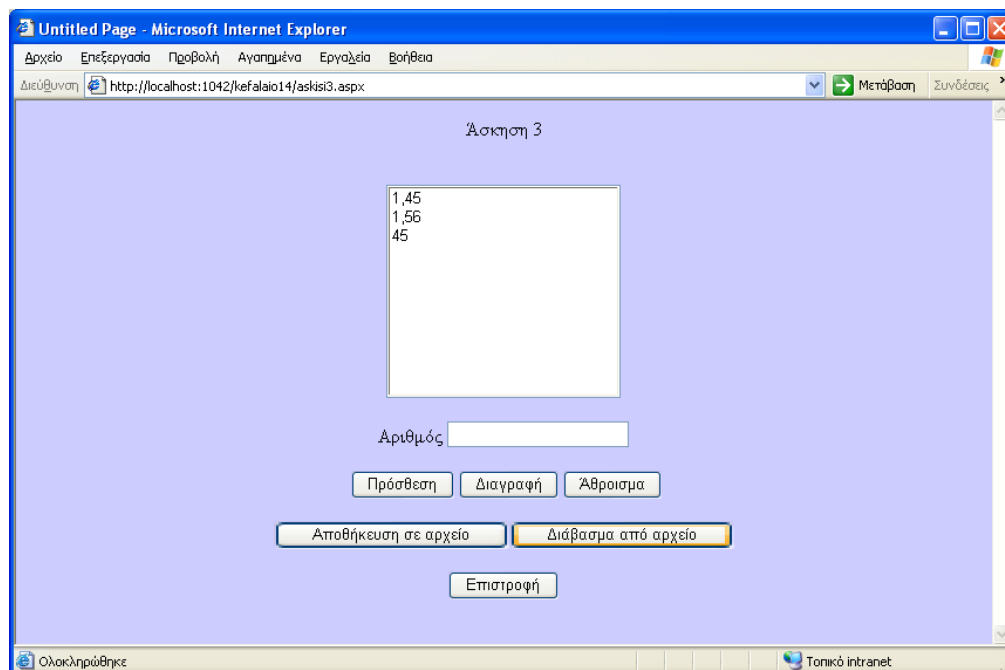
Υπόδειξη: Χρησιμοποιήστε κοινή μεταβλητή Session("Person") η οποία αποθηκεύει το αντικείμενο Person. Κατά την φόρτωση της ιστοσελίδας default.aspx ελέγξτε εάν δεν έχει καθορισθεί η μεταβλητή Session("Person") με την εντολή επιλογής

If Not Session("person") Is Nothing Then ...

Εάν δεν γίνει ο έλεγχος αυτός, τότε θα εμφανίζεται μήνυμα λάθους.

Τμήμα Γ

Η ιστοσελίδα askisi3.aspx, η οποία φαίνεται στην παρακάτω εικόνα, χρησιμοποιείται για την εισαγωγή των πραγματικών αριθμών σε λίστα επιλογής με όνομα ListBoxNumbers.



Το πλήκτρο «Πρόσθεση», προσθέτει ως αντικείμενο της λίστας τον αριθμό ο οποίος έχει πληκτρολογηθεί στο πλαίσιο κειμένου με όνομα textBoxNumber, το οποίο εμφανίζεται κάτω από την λίστα επιλογής.

Το πλήκτρο «Διαγραφή» διαγράφει το επιλεγμένο στοιχείο της λίστας.

Το πλήκτρο «Άθροισμα» εμφανίζει μήνυμα με το άθροισμα των στοιχείων του πίνακα.

Το πλήκτρο «Αποθήκευση σε αρχείο» αποθηκεύει τα στοιχεία της λίστας σε αρχείο με όνομα "c:\app\numbers.dat".

Το πλήκτρο «Διάβασμα από αρχείο» διαβάζει τα στοιχεία της λίστας από αρχείο με όνομα "c:\app\numbers.dat".

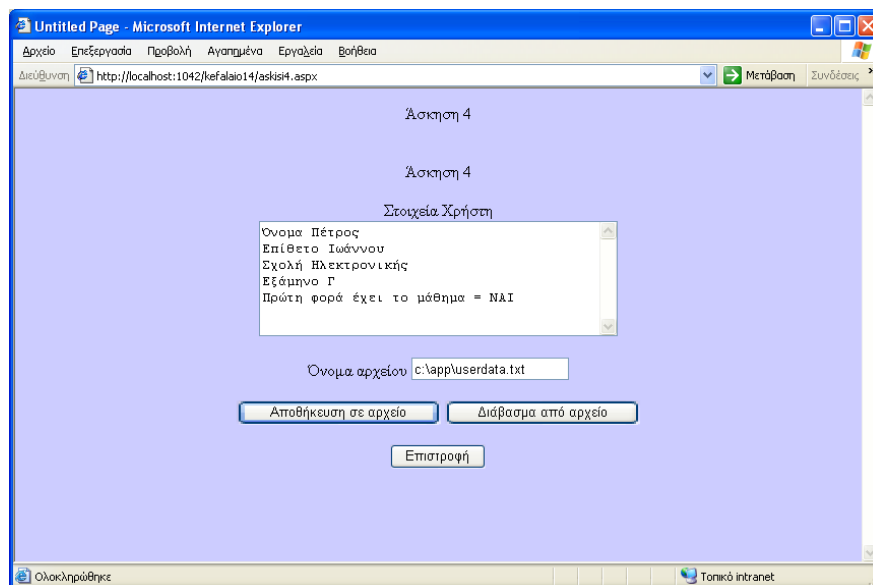
Το πλήκτρο «Επιστροφή» εμφανίζει την αρχική ιστοσελίδα με όνομα "default.aspx".

Τμήμα Δ

Η ιστοσελίδα askisi4.aspx, η οποία φαίνεται στην παρακάτω εικόνα, εμφανίζει τα στοιχεία τα οποία εισήγαγε ο χρήστης με την ιστοσελίδα askisi2.aspx.

Τα στοιχεία εμφανίζονται σε πλαίσιο κειμένου με όνομα TextBoxUserData το οποίο έχει την ιδιότητα TextMode=Multiline ώστε τα στοιχεία του χρήστη να εμφανίζονται σε πολλές γραμμές και να εκτυπώνονται όπως φαίνεται στην παρακάτω εικόνα.

Για την προσθήκη μιας γραμμής μπορείτε να χρησιμοποιήσετε την σταθερά vbCrLf. (Παράδειγμα η εντολή `Dim s as String="1" + vbCrLf + "2" + vbCrLf + "3"` ορίζει μια μεταβλητή κειμένου με τρεις γραμμές)



Το πλήκτρο «Αποθήκευση σε αρχείο» αποθηκεύει τα στοιχεία του χρήστη σε αρχείο με όνομα το οποίο ορίζεται από πλαίσιο κειμένου textBoxFilename στο οποίο ο χρήστης πληκτρολογεί το όνομα του αρχείου.

Το πλήκτρο «Διάβασμα από αρχείο» διαβάζει τα στοιχεία του χρήστη από αρχείο με όνομα το οποίο ορίζεται από πλαίσιο κειμένου textBoxFilename στο οποίο ο χρήστης πληκτρολογεί το όνομα του αρχείου.

Το πλήκτρο «Επιστροφή» εμφανίζει την αρχική ιστοσελίδα με όνομα "default.aspx".