

**Τεχνολογικό Εκπαιδευτικό  
Ίδρυμα Κρήτης  
Σχολή εφαρμοσμένων επιστημών  
Τμήμα Ηλεκτρονικών Μηχανικών**

**Πτυχιακή Εργασία**

**Ανάπτυξη εφαρμογής για την χρήση του eclass  
μέσω smartphones**

**Του φοιτητή  
Δάβιου Βασίλη  
Α. Μ. 3779**

**Επιβλέπων Καθηγητής  
Δρ. Κόκκινος Ευάγγελος**

**Χανιά 2014**

## Πρόλογος

Το 2008 πρωτοεμφανίζεται στην αγορά το πρώτο iPhone (3G) . Τρία χρόνια πριν, το 2005, κάνει την εμφάνιση του το Facebook ένα καινούργιο μέσο κοινωνικής δικτύωσης . Αυτές οι δύο εταιρίες και τα προϊόντα τους θα αλλάξουν για πάντα την κινητή τηλεφωνία και την χρησιμότητα των κινητών τηλεφώνων στην καθημερινότητα μας. Ο Steve jobs ο ιδρυτής της apple σε μια παρουσίαση των καινούργιων προϊόντων της εταιρίας του αναφερόμενος στο iPhone είπε : Με το iPhone φέραμε τον κόσμο του ίντερνετ στην τσέπη σας . Σαν συνέχεια αυτών το προϊόντων ήρθε η σημαντική βελτίωση και αύξηση της ταχύτητας που δύναται να συνδεθεί ένα χρήστης από το κινητό του στο ίντερνετ.

Σήμερα μας δίνεται η δυνατότητα μέσω των κινητών να κάνουμε σχεδόν ότι μπορούμε να κάνουμε μέσω των προσωπικών μας υπολογιστών. Αυτή η εξέλιξη της τεχνολογίας οδήγησε στην δημιουργία καινούργιων κλάδων σε πάρα πολλά επαγγέλματα, οι οποίοι σχετίζονται με την κινητή τηλεφωνία και συγκεκριμένα τα smartphones, όπως ονομάζονται το iPhone και άλλα αντίστοιχα κινητά. Σημαντικές αλλαγές, με τον ερχομό των smartphones και την ευρεία αποδοχή από το καταναλωτικό κοινό, υπέστησαν και οι ιστότοποι.

Ακολουθώντας αυτές τις εξελίξεις στον χώρο της τεχνολογίας επέλεξα σαν πτυχιακή μου εργασία να δημιουργήσω μια έκδοση για κινητά τηλέφωνα του ιστότοπου του e-class που χρησιμοποιούν οι φοιτητές της σχολής μας.



## Περίληψη

Για την δημιουργία ενός ιστότοπου για κινητά τηλέφωνα χρειάζονται κάποιες βασικές γνώσεις στις γλώσσες προγραμματισμού, στις βάσεις δεδομένων, αλλά και σε κάποια απαραίτητα προγραμματιστικά εργαλεία που θα χρησιμοποιηθούν. Επίσης χρειάζεται κατανόηση της λογικής που κρύβεται πίσω από την δημιουργία ενός ιστότοπου για κινητά τηλέφωνα.

Στο πρώτο κεφάλαιο θα αναλύσουμε τι είναι οι γλώσσες προγραμματισμού και πως αυτές χωρίζονται. Επίσης θα αναλύσουμε και κάποιες βασικές έννοιες για τις βάσεις δεδομένων.

Στο δεύτερο κεφάλαιο θα δούμε τις γλώσσες που χρησιμοποιήσαμε για την δημιουργία του eclass για κινητά τηλέφωνα και πώς μπορούμε να προγραμματίσουμε με αυτές.

Στο τρίτο κεφάλαιο θα δούμε τα προγραμματιστικά προγράμματα που χρησιμοποιήθηκαν και κάποιες βασικές τους χρήσεις τους.

Τέλος στο τέταρτο κεφάλαιο θα δούμε τον κώδικα του eclass και θα τον επεξηγήσουμε .

Λέξεις κλειδιά : κινητά τηλέφωνα, προγραμματισμός, ίντερνετ

## Abstract

To create a mobile form website one needs to have a basic knowledge of program languages, data bases, and certain program tools necessary to use. Furthermore one needs to understand the logic behind the creation of a website for mobiles.

In the first chapter we are going to analyse what exactly program languages are and how they differ from one another. In addition to this we will analyse some basic rules for data bases.

In the second chapter we are going to review the program languages we used to create eclass and how we can create programs with them.

In the third chapter we will review the program tools we used for the creation of eclass and features of these tools.

Finally in the fourth chapter we are going to explain the code behind eclass.

Key words: mobile phone, programming, internet

## Περιεχόμενα

Πρόλογος.....	1
Κεφάλαιο 1.....	8
Εισαγωγή στις γλώσσες προγραμματισμού και βάσεις δεδομένων : .....	8
Γλώσσες προγραμματισμού.....	9
Διαδικασία ανάπτυξης λογισμικού 10	
Ανάλυση απαιτήσεων .....	11
Σχεδιασμός λογισμικού .....	11
Αποσφαλμάτωση.....	11
Έλεγχος λογισμικού.....	12
Συντήρηση κώδικα .....	12
Πρότυπα Προγραμματισμού.....	13
Προστακτικός προγραμματισμός.....	13
Συναρτησιακός Προγραμματισμός.....	14
Λογικός Προγραμματισμός.....	14
Αντικειμενοστραφής Προγραμματισμός .....	15
Βάσεις Δεδομένων .....	18
Μοντέλα βάσεων δεδομένων .....	19
Κεφάλαιο 2.....	22
Πλατφόρμα E-class : .....	22
E-class .....	23
Γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την δημιουργία του ιστοτόπου e-class : .....	23
NET framework και C# .....	23
Χαρακτηριστικά του .NET Framework .....	24
Common Language Runtime .....	27
Η γλώσσα προγραμματισμού C#:.....	32
Χαρακτηριστικά της C# : .....	33
HTML: .....	34
Προέλευση.....	34
Βασικές γνώσεις της HTML που χρειάστηκαν για την δημιουργία του class για κινητά τηλέφωνα : .....	35
Css:.....	39
Διαφορά μεταξύ CSS και HTML : .....	39

Δυνατότητες της CSS :.....	39
Στοιχεία CSS :.....	40
Βάση δεδομένων και Queries :.....	42
JavaScript.....	47
Τρόποι εισαγωγής της JavaScript: .....	47
Κεφάλαιο 3.....	48
Προγραμματιστικά Εργαλεία :.....	48
Εργαλεία που χρησιμοποιήθηκαν για την δημιουργία του eclass .....	49
MySQL/phpMyAdmin: .....	49
Apache HTTP:.....	50
Visual Studio:.....	50
Εκδόσεις του Visual Studio : .....	50
Visual Studio Express.....	50
Visual Studio LightSwitch.....	50
Visual Studio Professional.....	51
Visual Studio Premium .....	51
Visual Studio Ultimate.....	51
Το περιβάλλον του Visual Studio Ultimate 2013.....	51
Ο Solution Explorer .....	52
Προσθήκη νέων Project και Items.....	54
Προσθήκη αναφορών .....	54
Τα παράθυρα εργαλείων .....	55
Ο επεξεργαστής κειμένου .....	57
Περιοχές .....	58
Outlining .....	58
Επισήμανση αναφορών .....	59
Code Zooming .....	59
Word Wrap .....	59
Αρίθμηση Γραμμών.....	60
Fonts και χρώματα .....	60
Full-Screen Mode.....	60
Παρακολούθηση Αλλαγών.....	60
Split View .....	61
Άλλες λειτουργίες .....	62

Δημιουργία ενός νέου ιστότοπου.....	63
Κεφάλαιο 4.....	72
Περιγραφή κώδικα του E-class :.....	72
Τροποποιώντας το e-class για κινητά τηλέφωνα : .....	73
Παρουσίαση και επεξήγηση του κώδικα του eclass :.....	75
1)Master Page .....	75
2)Το βασικό μενού.....	77
3)Τα μαθήματα που παρακολουθεί ο χρήστης.....	78
4)Κατάλογος μαθημάτων της σχολής.....	81
5)Μενού μαθήματος.....	83
6) Ανακοινώσεις .....	85
7) Ομάδες χρηστών .....	88
8) Επικοινωνία.....	91
Αναφορές.....	92



# Κεφάλαιο 1

Εισαγωγή στις γλώσσες προγραμματισμού και  
βάσεις δεδομένων

## Γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού είναι τυπικές γλώσσες εμπλουτισμένες με σημασιολογία που χρησιμοποιούνται για τον έλεγχο της συμπεριφοράς των ηλεκτρονικών υπολογιστών. Όλες οι γλώσσες προγραμματισμού έχουν κάποια θεμελιώδη δομικά στοιχεία για την περιγραφή των δεδομένων και των διεργασιών ή των μετασχηματισμών που εφαρμόζονται σε αυτά και ορίζονται χρησιμοποιώντας συντακτικούς και σημασιολογικούς κανόνες για τον προσδιορισμό της δομής και της σημασίας τους αντίστοιχα. Οι κανόνες αυτοί περιγράφονται σε μια προδιαγραφή γνωστή και ως programming language specification.

Τα κυριότερα δομικά στοιχεία των γλωσσών προγραμματισμού είναι το συντακτικό η σημασιολογία και η κύρια βιβλιοθήκη τους. Το συντακτικό μια γλώσσας περιγράφει τους πιθανούς συνδυασμούς των συμβόλων της γλώσσας που στοιχειοθετούν ορθές φράσεις σε ένα πρόγραμμα. Το νόημα ενός συγκεκριμένου συντακτικώς ορθού συνδυασμού λέξεων καθορίζεται στη σημασιολογία. Η κύρια βιβλιοθήκη μια γλώσσας, που συνήθως αποτελεί και μέρος της προδιαγραφής της, είναι ένα σύνολο κώδικα για την τέλεση βασικών λειτουργιών όπως το διάβασμα και το γράψιμο δεδομένων, ή παρέχει την υλοποίηση βασικών αλγορίθμων και δομών δεδομένων.

Οι περισσότερες γλώσσες προγραμματισμού είναι καθαρά κειμενικές : χρησιμοποιούν σειρές κειμένου που περιλαμβάνουν λέξεις, αριθμούς και σημεία στίξης. Υπάρχουν ωστόσο μερικές γλώσσες που είναι από τη φύση τους πιο οπτικές, αφού χρησιμοποιούν εικονικές σχέσεις μεταξύ συμβόλων για να καθορίσουν ένα πρόγραμμα.

Οι γλώσσες προγραμματισμού δεν είναι εύκολο να κατηγοριοποιηθούν πρωταρχικά, καθώς συνδυάζουν τα στοιχεία τους με νέες ιδέες, ορισμένες εκ των οποίων μπορεί να μην είναι ευδιάκριτες στις φαινομενικά όμοιες τους. Διακρίσεις των γλωσσών προγραμματισμού ωστόσο μπορούν να γίνουν ανάλογα με τον προγραμματιστικό πρότυπο που ακολουθούν και ενδεχομένως το πεδίο χρήσης για το οποίο προορίζονται. Επίσης οι γλώσσες προγραμματισμού χωρίζονται σε επίπεδα. Στο χαμηλότερο βρίσκεται η μοναδική γλώσσα την οποία «κατανοεί» ένας επεξεργαστής : ένα σύστημα εντολών που εκτελούνται κατευθείαν από αυτόν. Αυτό είναι το χαμηλότερο επίπεδο γλώσσας στο οποίο μπορεί να γραφτεί ένα πρόγραμμα και είναι γνωστό ως γλώσσα μηχανής. Το επίπεδο των υπόλοιπων γλωσσών εξαρτάται από το πόσο μοιάζουν με τον κώδικα μηχανής.

Τα προγράμματα σε γλώσσες χαμηλού επιπέδου έχουν το πλεονέκτημα ότι μπορούν να εκμεταλλευτούν όλα τα χαρακτηριστικά που μπορεί να έχει η αρχιτεκτονική όλα τα χαρακτηριστικά που μπορεί να έχει η αρχιτεκτονική του συγκεκριμένου επεξεργαστή όπου τρέχουν. Κατά συνέπεια, μπορεί να είναι πολύ αποδοτικά, αφού χρησιμοποιούν τον επεξεργαστή με το βέλτιστο τρόπο. Η δημιουργία τέτοιων προγραμμάτων απαιτεί αρκετό χρόνο τόσο για την εκμάθηση του συντακτικού τους όσο και για την ανάπτυξη τους, αλλά και πλήρη κατανόηση της λειτουργίας του επεξεργαστή. Για το λόγο αυτό ο προγραμματισμός σε χαμηλό επίπεδο γίνεται για πολύ μικρά προγράμματα ή για τμήματα κώδικα που

επιτελούν πολύ σημαντικές εργασίες και είναι απαραίτητο να τρέχουν όσο αποδοτικότερα γίνεται.

Οι γλώσσες υψηλού επιπέδου αναπαριστούν αφηρημένα τις λεπτομέρειες του υπολογιστή, χρησιμοποιούν εκτενώς φυσική γλώσσα, είναι ευκολότερες στη χρήση και συνεπώς επιτρέπουν τη γρήγορη δημιουργία μεγάλων προγραμμάτων. Το τελικό πρόγραμμα μπορεί να μην είναι όσο αποδοτικό όσο το αντίστοιχο σε γλώσσα χαμηλού επιπέδου, ωστόσο το κέρδος στο χρόνο ανάπτυξης υπερκεράζει την απόδοση των τελικών προγραμμάτων. Υπάρχουν δύο κύρια μοντέλα εκτέλεσης των υψηλών γλωσσών προγραμματισμού : α) οι διερμηνευμένες γλώσσες ( interpreted languages ) οι οποίες διαβάζονται από ένα ειδικό πρόγραμμα που καλείται διερμηνευτής (interpreter), μετατρέπονται σε γλώσσα μηχανής από αυτό και εκτελούνται άμεσα χωρίς φάση μεταγλώττισης (πχ. η PHP και η Python) και β) οι μεταγλωττισμένες γλώσσες ( compiled languages ) που πριν την εκτέλεση τους μετατρέπονται σε μια εκτελέσιμη μορφή: είτε κώδικα μηχανής ( πχ. η C και η C++) είτε ενδιάμεσης μορφής που χρειάζεται να διερμηνευτεί για να εκτελεστεί ( πχ. η java και η C#). Υπάρχουν επίσης και συνδυασμοί των παραπάνω μοντέλων όπως της java, που συνδυάζουν τη μεταγλώττιση με τη διερμηνεία, χρησιμοποιώντας ταυτόχρονα και τη μέθοδο just-in-time compilation για τη βελτίωση της ταχύτητας της διερμηνείας.

Σημαντικές γλώσσες στον προγραμματισμό τέλος είναι και οι γλώσσες σήμανσης (markup languages). Μια γλώσσα σήμανσης είναι ένα μοντέρνο σύστημα προσθήκης σημάνσεων σε κείμενο με τέτοιο τρόπο ώστε να είναι συντακτικά ευδιάκριτες από το κείμενο στο οποίο προστίθενται. Οι γλώσσες σήμανσης δεν είναι γλώσσες προγραμματισμού ωστόσο χρησιμοποιούνται ευρέως σε μια μεγάλη γκάμα εφαρμογών εξαιτίας της περιγραφικής δύναμης τους. Οι σημάνσεις που μπορούν να προστεθούν σε ένα κείμενο, μπορούν να είναι α) παρουσιαστικές (presentational markup) δηλαδή να χρησιμοποιούνται για να προσδιορίσουν το παρουσιαστικό του κειμένου στο οποίο αναφέρονται ( πχ. η HTML), β) διαδικαστικές (procedural markup) για να παρέχουν οδηγίες σε άλλα προγράμματα για το πώς να επεξεργάζονται το κείμενο που περιέχουν (πχ. η TEX) και γ) περιγραφικές (descriptive markup) που απλώς παρέχουν μια ετικέτα στο κείμενο στο οποίο αναφέρονται.

#### Διαδικασία ανάπτυξης λογισμικού

Ένας απλός ορισμός που αποδίδει ικανοποιητικά την έννοια του προγραμματισμού υπολογιστών είναι η ικανότητα της επικοινωνίας του προγραμματιστή με τον υπολογιστή σε μια αμοιβαίως αντιληπτή γλώσσα προγραμματισμού για την πραγματοποίηση χρήσιμων εργασιών. Ο προγραμματισμός ωστόσο αποτελεί μόνο ένα κομμάτι της ανάπτυξης λογισμικού. Κατά συνέπεια ένας προγραμματιστής είναι γενικά χρήσιμο να γνωρίζει τα στάδια που ακολουθούνται στην ανάπτυξη έργων λογισμικού. Η ανάπτυξη λογισμικού περιλαμβάνει τις διαδικασίες της ανάλυσης απαιτήσεων, του σχεδιασμού, της γραφής, της δοκιμής, της αποσφαλμάτωσης και της διατήρησης του πηγαίου κώδικα των προγραμμάτων γραμμένων σε μια γλώσσα ή και περισσότερες. Ένας προγραμματιστής είναι καλό να κατανοεί τις μεθόδους

και τις πρακτικές που εφαρμόζονται σε κάθε μια από τις παραπάνω διαδικασίες, δίνοντας έμφαση κατ' εξοχήν στη γραφή κώδικα. Ωστόσο ο προγραμματιστής δεν περιορίζεται μόνο στις παραπάνω διαδικασίες. Συχνά απαιτείται και η εξειδίκευση σε πολλαπλές διαφορετικές πρακτικές ή θεωρητικές έννοιες, όπως οι αλγόριθμοι και η λογική .

#### Ανάλυση απαιτήσεων

Η επιτυχία οποιουδήποτε έργου λογισμικού εξαρτάται σε μεγάλο βαθμό από τον αρχικό εντοπισμό των απαιτήσεων του υπό ανάπτυξη συστήματος τόσο μέσα από τη λήψη των ζητούμενων των πελατών, όσο και από την ανάδειξη των διαφόρων υπο-προβλημάτων που μπορεί να προκύψουν. Το πρώτο βήμα στην ανάπτυξη ενός έργου λογισμικού είναι η ανάλυση απαιτήσεων (requirement analysis). Το αποτέλεσμα του πρώτου αυτού βήματος είναι προδιαγραφές τεχνολογικές, επιχειρηματικές, λειτουργικές, κόστους και χρονικής διάρκειας.

#### Σχεδιασμός λογισμικού

Η διαδικασία της γραφής κώδικα, είναι η μετατροπή του σχεδιασμού και γενικότερα της λογικής του λογισμικού σε κώδικα μιας γλώσσας προγραμματισμού. Η γραφή κώδικα απαιτεί πλήρη γνώση της χρησιμοποιούμενης γλώσσας προγραμματισμού καθώς η ελλιπής γνώση μπορεί να οδηγήσει σε άσκοπο γράσιμο κώδικα που υπάρχει ήδη ή την επιλογή λανθασμένων μονοπατιών υλοποίησης. Σε μεγάλα έργα προγραμματισμού υιοθετούνται συχνά ένα σύνολο προτύπων, κατευθύνσεων ή συμβάσεων γραφής με σκοπό τη βελτίωση της ποιότητας, της αναγνωσιμότητας και της ευκολίας της συντήρησης του κώδικα. Τα πρότυπα αυτά περιλαμβάνουν κανόνες ονομασίας και οργάνωσης των αρχείων ή των λέξεων στον κώδικα, τη δόμηση του κώδικα σε μπλοκ με εσοχές για να τονίζουν το επίπεδο ένθεσης του ή τη μορφοποίηση των διαφόρων εντολών. Τα πρότυπα κωδικογράφησης GNU, το στυλ του πυρήνα BSD και οι συμβάσεις κώδικα στη Java.

#### Αποσφαλμάτωση

Η γραφή κώδικα είναι μια ανθρώπινη διαδικασία και κατά συνέπεια είναι υποκείμενη σε σφάλματα που μπορεί να οφείλονται σε παραλήψεις ή απρόβλεπτους παράγοντες. Η αποσφαλμάτωση (debugging) είναι η μεθοδική διαδικασία της εύρεσης και του περιορισμού των σφαλμάτων (bugs) και των ελαττωμάτων στον κώδικα ενός προγράμματος. Γενικά αποτελείται από τα βήματα της αναπαραγωγής του προβλήματος, της απομόνωσης του σημείου που εμφανίζεται το σφάλμα, της αναγνώρισης της αιτίας που το προκαλεί, της διόρθωσης του σφάλματος και της επιβεβαίωσης της απαλοιφής του προβλήματος. Η αποσφαλμάτωση είναι μια αρκετά επίπονη διαδικασία και για τ ο λόγο αυτό αναπτύχθηκαν ειδικά προγράμματα για το σκοπό αυτό : οι αποσφαλματωτές (debuggers).

## Έλεγχος λογισμικού

Στη διαδικασία του ελέγχου λογισμικού βεβαιώνεται ότι το αποτέλεσμα της συγγραφής πληροί τις προδιαγραφές όπως αυτές ορίστηκαν στην ανάλυση και το σχεδιασμό, ότι δουλεύει όπως αναμενόταν και ότι μπορεί να εγκατασταθεί για εφαρμογή με τα ίδια χαρακτηριστικά. Υπάρχουν διάφορες μέθοδοι ελέγχου που μπορούν να γίνουν σε διαφορετικά επίπεδα, ανάλογα με την πρόσβαση στο κάθε τμήμα κώδικα. Ο έλεγχος μπορεί να φτάσει στα θεμελιώδη τμήματα του κώδικα ή να αντιμετωπίσει την εφαρμογή στην ολότητα της. Ο έλεγχος μπορεί να πραγματοποιείται και για μεμονωμένα τμήματα κώδικα κατά την ανάπτυξη του αλλά παραδοσιακά γίνεται μετά την ολοκλήρωση της ανάπτυξης του.

## Συντήρηση κώδικα

Ακόμα και μετά την αποσφαλμάτωση και τον έλεγχο του λογισμικού μπορούν να εντοπιστούν προβλήματα στο σχεδιασμό ή την υλοποίηση ή ακόμα και να προστεθούν νέες απαιτήσεις. Επειδή η αλλαγή είναι έμφυτη στην ανάπτυξη λογισμικού, η ανάγκη του ελέγχου της οργάνωσης και της εξέλιξης της ανάπτυξης του κώδικα είναι επιτακτική, ιδιαίτερα για τα έργα μεγάλης κλίμακας. Η διαδικασία της συντήρησης κώδικα (code maintenance) είναι ουσιαστικά ο έλεγχος εκδόσεων (revision control) του πηγαίου κώδικα και περιλαμβάνει τον έλεγχο της εξέλιξης του μέσα στο χρόνο, κρατώντας ένα χρονικό για το σκεπτικό πίσω από τις υφιστάμενες αλλαγές. Υπάρχει μια πληθώρα συστημάτων ελέγχου εκδόσεων (revision control systems) όπως τα CVS και Subversion (SVN) που χρησιμοποιούνται κατά κόρον σε έργα ανοιχτού κώδικα και τον Team Foundation Server για έργα στην πλατφόρμα .NET.

Οι αρχές προγραμματισμού είναι ανεξάρτητες της γλώσσας που χρησιμοποιείται. Κάθε γλώσσα έχει τα δυνατά και αδύναμα σημεία της, καθιστώντας έτσι την ανάπτυξη κάποιων προγραμμάτων ευκολότερη ή δυσκολότερη σε αυτές. Ένας πεπειραμένος προγραμματιστής θα βρει σχετικά εύκολη η μετάβαση από μια γλώσσα σε μια άλλη καθώς το ουσιαστικότερο πράγμα που πρέπει να κάνει είναι να εντοπίσει πως υλοποιούνται οι προγραμματιστικές αρχές στη νέα γλώσσα. Τα περισσότερα μαθήματα προγραμματισμού επικεντρώνονται στο γράψιμο των προγραμμάτων από το μηδέν. Ωστόσο αρκετά μεγάλο ποσοστό της ανάπτυξης λογισμικού καταναλώνεται αφού έχει γραφεί ο κώδικας και πιο συγκεκριμένα στην ανάγνωση, την κατανόηση και την τροποποίηση του αρχικού κώδικα.

Η εκμάθηση της ανάγνωσης κώδικα είναι επίσης μια απαραίτητη ικανότητα για έναν προγραμματιστή ιδιαίτερα σήμερα, στην εποχή της επαναχρησιμοποίησης κώδικα και της συνεργατικής ανάπτυξης σε απαιτητικά εταιρικά περιβάλλοντα. Το λογισμικό ανοιχτού κώδικα παρέχει πρόσβαση σε εκατομμύρια γραμμές κώδικα για ανάγνωση, κριτική, τη δυνατότητα βελτίωσης από τον πεπειραμένο χρήστη και εκμάθησης για τον αρχάριο. Για το λόγο αυτό είναι καλή πρακτική η ανάγνωση ποιοτικού κώδικα που έχει γραφτεί για λογισμικό ανοιχτού κώδικα, καθώς έχει συζητηθεί, συγκεκριμενοποιηθεί, γενικευτεί και τεκμηριωθεί από αρκετούς και κυρίως έμπειρους προγραμματιστές. Η ανάγνωση κώδικα επομένως είναι πιθανό να ανεβάσει το αισθητικό κριτήριο

του προγραμματιστή για τον κώδικα που παράγει, όπως ακριβώς η ανάγνωση ποιοτικού πεζού λόγου εμπλουτίζει το λεξιλόγιο, ενεργοποιεί τη φαντασία και διευρύνει το νου του αναγνώστη.

### Πρότυπα Προγραμματισμού

Για να ξεκινήσει κανείς τη μελέτη της C# και του αντικειμενοστραφούς προγραμματισμού τον οποίο αυτή ακολουθεί, πρέπει να κατανοήσει τη γενικότερη φιλοσοφία και τα χαρακτηριστικά του. Ταυτόχρονα θα ήταν σκόπιμο να τον συγκρίνει με τα κυριότερα προγραμματιστικά πρότυπα που υπάρχουν.

Ένα προγραμματιστικό πρότυπο είναι μια προσέγγιση για τον προγραμματισμό υπολογιστών που βασίζεται σε κάποια μαθηματική θεωρία ή ένα αυστηρό σύνολο κανόνων. Κάθε πρότυπο περιέχει ένα σύνολο εννοιών που το καθιστούν ιδανικό για την επίλυση συγκεκριμένου είδους προβλημάτων. Για παράδειγμα, ο αντικειμενοστραφής προγραμματισμός είναι ιδανικός για προβλήματα με ένα μεγάλο πλήθος αφαιρέσεων δεδομένων, οργανωμένων σε μια ιεραρχία ενώ ο λογικός προγραμματισμός είναι ιδανικός για το μετασχηματισμό πολύπλοκων συμβολικών δομών σύμφωνα με κανόνες λογικής. Υπάρχουν περίπου 27 διαφορετικά προγραμματιστικά πρότυπα, αρκετά από τα οποία ωστόσο έχουν κοινά στοιχεία μεταξύ τους.

Τα προγραμματιστικά πρότυπα δεν πρέπει να συγχέονται με τις τεχνικές προγραμματισμού οι οποίες σχετίζονται με τις αλγοριθμικές ιδέες για την επίλυση προβλημάτων κάποιας συγκεκριμένης τάξης, ούτε και με το στυλ προγραμματισμού, δηλαδή την αποτύπωση της προσωπικής έκφρασης του προγραμματιστή μέσα στον κώδικα.

Η επίλυση ενός προγραμματιστικού προβλήματος απαιτεί σωστές προσεγγίσεις. Ακόμα και τα πιο απλά προβλήματα ενδέχεται να χρειάζονται διαφορετική προσέγγιση σε διάφορα μέρη τους. Για το λόγο αυτό, είναι ιδανικό μια γλώσσα προγραμματισμού να υποστηρίζει περισσότερα από ένα προγραμματιστικά πρότυπα. Κατά πολλούς είναι ατυχές το γεγονός ότι δημοφιλείς γλώσσες όπως η Java ή η C++ υποστηρίζουν ένα ή δύο πρότυπα μόνο. Η υποστήριξη πολλαπλών προτύπων σε μια γλώσσα επιτρέπει στους προγραμματιστές να επιλέξουν τις σωστές έννοιες όταν τις χρειάζονται άμεσα. Οι γλώσσες που υποστηρίζουν πολλαπλά πρότυπα συχνά καλούνται «πολυπρότυπες». Είναι δυνατόν ωστόσο να υλοποιηθεί, εν μέρει ή και πλήρως, ένα πρότυπο σε μια γλώσσα που δεν το υποστηρίζει, αφού αυτό έχει κατανοηθεί πλήρως. Για παράδειγμα ο συναρτησιακός προγραμματισμός είναι δυνατόν να υποστηριχθεί στην C#.

### Προστακτικός Προγραμματισμός

Στον προστακτικό προγραμματισμό ένα πρόβλημα επιλύεται υπολογιστικά μέσω προτάσεων που αλλάζουν την κατάσταση του προγράμματος. Τα προστακτικά προγράμματα ορίζουν μια ακολουθία εντολών προς εκτέλεση από τον υπολογιστή.

Εδώ ο προγραμματιστής γράφει κώδικα που περιγράφει τα ακριβή βήματα που πρέπει να ακολουθήσει ένα πρόγραμμα για να πετύχει το στόχο του. Επομένως, ο κυριότερος στόχος του προστακτικού προγραμματισμού είναι να

δεχθεί δεδομένα σαν είσοδο, να τα αλλάξει μέσω ενημερώσεων στη μνήμη και να παράγει έξοδο βασισμένη σε αυτά τα συνεχώς αλλασώμενα δεδομένα. Κάθε πρόγραμμα έχει μια αρχική κατάσταση, μια λίστα υπολογιστικών βημάτων που πρέπει να ολοκληρώσει σε μια σειρά που ελέγχεται από τις δομές ελέγχου και ένα σημείο εξόδου. Οι συνηθέστερες εντολές που προσφέρονται από τις προστακτικές γλώσσες προγραμματισμού αφορούν την ανάθεση τιμών και την είσοδο/έξοδο. Οι γνωστότερες γλώσσες που ακολουθούν αυτό το παράδειγμα είναι η C, η Basic, η Pascal και η Fortran.

### Συναρτησιακός Προγραμματισμός

Στο συναρτησιακό προγραμματισμό ο στόχος του προγράμματος είναι ελαφρώς διαφορετικός και αυτό έχει επίπτωση στον τρόπο προγραμματισμού : δίνεται έμφαση στη μορφή της πληροφορίας που είναι επιθυμητή και στους μετασχηματισμούς που χρειάζονται για να φτάσει το πρόγραμμα σε αυτή. Το επίκεντρο εδώ είναι οι κλήσεις συναρτήσεων. Ο κυριότερος τρόπος αλλαγής των δεδομένων είναι οι συναρτήσεις και το πέρασμα των παραμέτρων.

Τα δεδομένα δεν αλλάζουν στο συναρτησιακό προγραμματισμό και οι νέες τιμές δημιουργούνται εκ νέου από παλιότερες. Κατά συνέπεια δεν υπάρχει η κλασική κατάσταση προγράμματος. Επίσης η διαφορά μεταξύ του προγράμματος και των δεδομένων εξαφανίζεται. Μια συνάρτηση είναι δεδομένα όπως κάθε άλλο στοιχείο δεδομένων. Για παράδειγμα οι ακέραιοι αριθμοί και οι συναρτήσεις είναι το ίδιο. Τα στατικά δεδομένα μπορεί να υπάρχουν αλλά παίζουν μικρό ρόλο στο συναρτησιακό προγραμματισμό.

Εδώ η στοίβα αποκτά μεγαλύτερη σημασία καθώς η περισσότερη δουλειά πραγματοποιείται με την κλήση συναρτήσεων. Στις συναρτησιακές γλώσσες λαμβάνεται μεγαλύτερη μέριμνα στην απόκρυψη του σωρού από τον προγραμματιστή έτσι ώστε να μην είναι διαθέσιμος για επεμβάσεις. Τα δεδομένα είναι δυναμικά αλλά μόλις δημιουργούνται στο σωρό, αν ακολουθείται συναρτησιακό μοντέλο αυστηρά δεν μπορούν να αλλάξουν. Ωστόσο αρκετές γλώσσες παρακάμπτουν αυτόν τον κανόνα για να αντιμετωπίσουν θέματα απόδοσης. Μερικές γλώσσες που ακολουθούν αυτό το παράδειγμα είναι η Clojure, η Erlang, η Haskell και F#.

### Λογικός Προγραμματισμός

Ο λογικός προγραμματισμός είναι αρκετά διαφορετικός από τα προηγούμενα πρότυπα. Εδώ ο προγραμματιστής δεν γράφει ένα πρόγραμμα με την παραδοσιακή του έννοια αλλά ένα σύνολο από λογικούς κανόνες που πρέπει να ισχύουν. Από αυτό το σύνολο το πρόγραμμα προσπαθεί να απαντήσει ερωτήσεις με ένα ναι ή όχι. Κατά συνέπεια, η σειρά δήλωσης των κανόνων επηρεάζει την έξοδο του προγράμματος. Στο λογικό προγραμματισμό υπάρχει η έννοια του σωρού καθώς οι κανόνες αποτιμώνται κατά την εκτέλεση του προγράμματος και αυτή προϋποθέτει την δυναμική προσθήκη ή αφαίρεση κανόνων κατά την εκτέλεση.

Ο λογικός προγραμματισμός είναι ιδιαίτερα χρήσιμος στο πεδίο της Τεχνητής Νοημοσύνης όπου μπορεί να χρησιμοποιηθεί για την αναπαράσταση

γνώσης με τη μορφή ενός λογικού προγράμματος. Η γνωστότερη γλώσσα στο λογικό προγραμματισμό είναι η Prolog.

### Αντικειμενοστραφής Προγραμματισμός

Ο αντικειμενοστραφής προγραμματισμός καθιερώθηκε τη δεκαετία του 1990, η γλώσσα που συνεισέφερε αρκετά σε αυτό ήταν η Java, αντικαθιστώντας σε μεγάλο βαθμό το παραδοσιακό πρότυπο του προστακτικού προγραμματισμού. Εδώ το βασικό στοιχείο του προγράμματος είναι το αντικείμενο που συνδυάζει ιδιότητες και διαδικασίες που επενεργούν σε αυτές αλλάζοντας τα δεδομένα τους. Κάθε αντικείμενο αποτελεί δεσμευμένο κομμάτι της μνήμης (και συγκεκριμένα του σωρού) και καλείται και στιγμιότυπο, ενός σύνθετου τύπου δεδομένων που καλείται κλάση. Η κλάση είναι μια αυτοτελής και αφαιρετική αναπαράσταση κάποιας κατηγορίας αντικειμένων, είτε φυσικών αντικειμένων του προγραμματικού κόσμου είτε νοητών, εννοιολογικών αντικειμένων και στην ουσία αποτελεί την προδιαγραφή των δεδομένων και των διαδικασιών που επιδρούν πάνω σε αυτά.

Αν κανείς παρατηρήσει πως υλοποιούνται οι διάφορες εργασίες στον πραγματικό κόσμο, θα παρατηρήσει ότι αλληλοεπιδρά σε έναν κόσμο που στρέφεται γύρω από αντικείμενα. Αν θέλει κάποιος να πάει σε ένα μέρος για παράδειγμα μπορεί να χρησιμοποιήσει το αυτοκίνητο του. Το αντικείμενο αυτοκίνητο αποτελείται από άλλα αντικείμενα που αλληλοεπιδρούν μεταξύ τους και κάνουν εφικτή τη μεταφορά μας. Ο οδηγός απομονώνεται από τις διαδικασίες αυτές και το πώς αλληλοεπιδρούν για να ξεκινήσει το αυτοκίνητο. Το μόνο που περιμένει είναι το αυτοκίνητο να ξεκινήσει και να διανύσει την διαδρομή που ο ίδιος επιθυμεί.

Παρόμοια, αποκρύπτεται από τους τελικούς χρήστες ενός λογισμικού η λογική που χρειάζεται για να υλοποιηθεί μια διεργασία. Για παράδειγμα, όταν εκτυπώνει ένας χρήστης μια σελίδα από ένα πρόγραμμα επεξεργασίας κειμένου, ξεκινάει μια διαδικασία πατώντας το κουμπί της εκτύπωσης και χωρίς να ξέρει τι γίνεται εσωτερικά στο πρόγραμμα, περιμένει να δει μια εκτυπωμένη σελίδα ή ένα μήνυμα αποτυχίας.

Οι αιτίες που ώθησαν στην ανάπτυξη του αντικειμενοστραφούς προγραμματισμού είναι η ευκολία συντήρησης, οργάνωσης, χειρισμού και επαναχρησιμοποίησης κώδικα μεγάλων και πολύπλοκων εφαρμογών και επικράτησε επειδή μπορούσε να ανταπεξέλθει σε προγράμματα πολύ μεγαλύτερου όγκου και πολυπλοκότητας. Σε αυτό συνετέλεσαν τα ακόλουθα χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού :



## Ενθυλάκωση

Η ιδιότητα της ενθυλάκωσης αναφέρεται στην απόκρυψη των δεδομένων και της υλοποίησης ενός αντικειμένου και είναι γνωστή και ως απόκρυψη δεδομένων. Τα δεδομένα και ο κώδικας όταν ενθυλακώνονται αποκρύπτονται από την εξωτερική πρόσβαση. Όταν ένας εξωτερικός παρατηρητής βλέπει ένα αντικείμενο, μπορεί να το χρησιμοποιήσει μόνο μέσω της εξωτερικής διασύνδεσης του ( interface), δηλαδή ότι είναι διαθέσιμο προς χρήση, ενώ οι εσωτερικές λεπτομέρειες είναι κρυμμένες και δεν μπορούν να χρησιμοποιηθούν. Έτσι τα ενθυλακωμένα δεδομένα δεν μπορούν να αλλάξουν και πιο συγκεκριμένα να ενημερωθούν άμεσα από εξωτερικά αντικείμενα.

## Αφαίρεση δεδομένων

Η αφαίρεση δεδομένων είναι η αρχή που απαιτεί τα προγράμματα να μην λαμβάνουν υπόψη συγκεκριμένες υλοποιήσεις ή εσωτερικές αναπαραστάσεις αλλά να έχουν μια γενική αφηρημένη ιδέα των αντικειμένων που χρησιμοποιούν. Στον αντικειμενοστραφή προγραμματισμό κάθε αντικείμενο μπορεί να ζητηθεί να παρέχει δεδομένα ή υπηρεσίες λαμβάνοντας μηνύματα/αιτήσεις. Τον αιτούντα δεν τον απασχολεί πως το αντικείμενο θα παράγει τα δεδομένα αλλά τα δεδομένα καθαυτά. Το πώς δηλαδή η υλοποίηση ή αλλιώς η δημιουργία της πληροφορίας είναι μια εσωτερική για τον αντικείμενο υπόθεση. Επομένως δίνεται έμφαση στο τι αντί για το πώς κατά την αλληλεπίδραση μεταξύ δύο αντικειμένων. Η αφαίρεση δεδομένων προσπαθεί να μειώσει τις λεπτομέρειες και να τις ομαδοποιήσει σε επίπεδα μιας υλοποίησης έτσι ώστε ο προγραμματιστής να επικεντρωθεί στις έννοιες που τον ενδιαφέρουν για την επίλυση ενός προβλήματος. Ταυτόχρονα ο προγραμματιστής δεν απασχολείται με την εκμάθηση συγκεκριμένων υλοποιήσεων, αλλά επικεντρώνεται στην επίλυση του δεδομένου προβλήματος, γνωρίζοντας μόνο ότι χρειάζεται. Ένα σύστημα μπορεί να έχει πολλαπλά επίπεδα αφαίρεσης, σε καθένα από τα οποία φανερώνονται διαφορετικές λεπτομέρειες.

## Κληρονομικότητα

Τα αντικείμενα τείνουν να δομούνται βασισμένα σε άλλα αντικείμενα. Όταν δημιουργείται ένα νέο αντικείμενο (μέσα από μια κλάση), ουσιαστικά ορίζονται οι ιδιότητες που το κάνουν ξεχωριστό από τα υπόλοιπα αντικείμενα. Εξαιτίας της ομοιότητας πολλών αντικειμένων υπάρχει ένας μηχανισμός για τη μεταφορά των ιδιοτήτων κάποιων αντικειμένων σε άλλα αντικείμενα έτσι ώστε να αποφεύγεται ο άσκοπος ορισμός των χαρακτηριστικών σε κάθε αντικείμενο ξεχωριστά, αλλά και για να υπάρχει μια νοητή συγγένεια μεταξύ των αντικειμένων. Αυτός ο μηχανισμός καλείται κληρονομικότητα. Η κληρονομικότητα επιτρέπει στους προγραμματιστές την επαναχρησιμοποίηση των ορισμών δομών που έχουν οριστεί στο παρελθόν, μειώνοντας έτσι την εργασία κατά την ανάπτυξη προγραμμάτων.

## Πολυμορφισμός

Ο όρος πολυμορφισμός στην κυριολεξία αναφέρεται στην ιδιότητα του να έχει κάτι πολλαπλές μορφές. Στις γλώσσες προγραμματισμού τις περισσότερες φορές αναφέρεται στην ιδιότητα των μεθόδων να μπορούν να δέχονται και/ή να επιστρέφουν τιμές διαφορετικού τύπου. Για παράδειγμα μια μέθοδος που δέχεται ένα μόνο όρισμα είναι πολυμορφική αν μπορεί να δεχτεί παραμέτρους διαφορετικού τύπου. Ο πολυμορφισμός είναι διάχυτος στις αντικειμενοστραφείς γλώσσες προγραμματισμού και συνδέεται αρκετά με την κληρονομικότητα.

## Βάσεις Δεδομένων

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Για την διαχείριση αυτού του τεράστιου όγκου πληροφοριών χρησιμοποιούμε συστήματα βάσεων δεδομένων που μας επιτρέπουν την αποθήκευση την επεξεργασία αλλά και την αποδοτική εκμετάλλευση της πληροφορίας.

Μια βάση δεδομένων είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, έτσι ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες ταυτόχρονα. Τα δεδομένα που υπάρχουν στις βάσεις δεδομένων πρέπει να είναι :

- Ολοκληρωμένα, δηλαδή τα δεδομένα πρέπει να είναι αποθηκευμένα σε ομοιόμορφα οργανωμένα σύνολα αρχείων όπου δεν πρέπει να υπάρχει επανάληψη ή πλεονασμός των ίδιων στοιχείων.
- Καταμεριζόμενα. Να μπορούν δηλαδή περισσότεροι του ενός χρήστες να βλέπουν και να μοιράζονται τα ίδια δεδομένα την ίδια χρονική στιγμή.

Υπάρχει ένα ειδικό λογισμικό το οποίο μεσολαβεί ανάμεσα στα αρχεία δεδομένων και τις εφαρμογές που χρησιμοποιούν οι χρήστες και αποκαλείται Σύστημα Διαχείρισης Βάσης Δεδομένων ή DBMS (Data Base Management System). Το DBMS είναι στην ουσία ένα σύνολο από προγράμματα και υπορουτίνες που έχουν να κάνουν με τον χειρισμό της βάσης δεδομένων, όσον αφορά τη δημιουργία, τροποποίηση, διαγραφή στοιχείων, με ελέγχους ασφάλειας κ.λπ.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ αυτήν. Το DBMS παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του DBMS μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων.

Ένα Σύστημα Βάσης Δεδομένων αποτελείται από το υλικό, το λογισμικό τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά :

- Το υλικό αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους κ.α. , όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το λογισμικό είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων της βάσης.
- Η βάση δεδομένων αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως στον χρήστη δίνεται η εντύπωση ότι βρίσκονται

συγκεντρωμένα στο ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα και μερισμένα. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.

- Οι χρήστες μια βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :
- Τελικοί χρήστες. Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογη πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
- Προγραμματιστές εφαρμογών. Αναπτύσσουν τις εφαρμογές του DBMS σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
- Διαχειριστής δεδομένων. Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
- Διαχειριστής βάσης δεδομένων. Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του DBMS.

#### Μοντέλα βάσεων δεδομένων

Υπάρχουν τρία βασικά μοντέλα που έχουν επικρατήσει στις βάσεις δεδομένων, το ιεραρχικό, το δικτυωτό και το σχεσιακό, και τα οποία αναπτύχθηκαν με βάση αντίστοιχες δομές.

#### Το ιεραρχικό μοντέλο

Το ιεραρχικό μοντέλο έχει μια ιεραρχική δομή που θυμίζει δένδρο. Οι οντότητες μοιάζουν με απολήξεις από κλαδιά δένδρων και τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες.

Από μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο εκκινούν πολλά κλαδιά, καθένα από τα οποία καταλήγει σε μια οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο. Αλλά, σε κάθε οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο αντιστοιχεί μια και μόνο μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο. Το μοντέλο αυτό ήταν το πρώτο που εμφανίσθηκε αλλά σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.

Το δικτυωτό μοντέλο βάσεων δεδομένων

Και στο δικτυωτό μοντέλο, τα στοιχεία τοποθετούνται σε ένα επίπεδο ιεραρχίας, αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία είτε σε ένα κατώτερο ή σ' ένα ανώτερο επίπεδο.

Το σχεσιακό μοντέλο βάσεων δεδομένων

Το σχεσιακό μοντέλο έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα μ' αυτό αποκαλούνται σχεσιακές. Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Υστερούν μόνο σε ταχύτητα υπολογισμών και σε χώρο αποθήκευσης, αλλά μόνο όταν έχουμε να κάνουμε με πολύ μεγάλες βάσεις δεδομένων.

Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με την μορφή πινάκων και σχέσεων αντίστοιχα.

Ένας πίνακας αποτελείται από γραμμές και στήλες, όπου τοποθετούμε τα στοιχεία σε οριζόντια και κάθετη μορφή. Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται χαρακτηριστικό ή πεδίο, ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες που αφορούν ένα στοιχείο της οντότητας και αποκαλείται πλειάδα ή εγγραφή.

Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ, για παράδειγμα. Το σύνολο των αποδεκτών τιμών μια οντότητας αποκαλείται πεδίο ορισμού.

Όπως είναι εύκολα κατανοητό, η βασικότερη εργασία που έχουμε να κάνουμε κατά τον σχεδιασμό μια σχεσιακής βάσης δεδομένων είναι να ορίσουμε τους πίνακες που θα χρησιμοποιήσουμε καθώς και τα πεδία που θα περιέχει ο καθένας απ' αυτούς. Η διαδικασία αυτή αποκαλείται κατασκευή του σχήματος μιας βάσης δεδομένων.

Οι κανόνες που πρέπει να ακολουθήσουμε πιστά κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι οι εξής :

- Η κάθε οντότητα πρέπει να παριστάνεται ως ένας ξεχωριστός πίνακας
- Η κάθε στήλη του πίνακα αντιστοιχεί σε μια ιδιότητα της οντότητας.
- Η κάθε γραμμή του πίνακα αντιστοιχεί σε μια εμφάνιση της οντότητας.
- Η κάθε γραμμή πρέπει να είναι μοναδική, δηλαδή αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν τα ίδια ακριβώς στοιχεία.
- Η σειρά εμφάνισης των γραμμών δεν έχει καμία σημασία.
- Η κάθε στήλη έχει μια δική της μοναδική ονομασία.

- Οι τιμές που ανήκουν στην ίδια στήλη πρέπει να είναι του ίδιου τύπου δηλαδή ή όλες αριθμοί ή όλες αλφαριθμητικές.
- Η στήλη που αποτελεί το πρωτεύον κλειδί μιας οντότητας, δεν πρέπει να είναι ποτέ κενή.
- Αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν την ίδια τιμή στο πρωτεύον κλειδί.
- Το πρωτεύον κλειδί μιας οντότητας αποκαλείται ξένο κλειδί σε μια άλλη οντότητα, με την οποία υπάρχει συσχετισμός.
- Μπορεί να υπάρχουν πολλές γραμμές που να έχουν την ίδια τιμή στο ξένο κλειδί.

Οι γλώσσες προγραμματισμού και οι βάσεις δεδομένων αποτελούν ένα μεγάλο κομμάτι της πτυχιακής μου εργασίας. Αφού κατανοήσαμε τις βασικές έννοιες και τους τρόπους λειτουργίας τους, στο επόμενο κεφάλαιο θα μιλήσουμε συγκεκριμένα για το ποιες γλώσσες προγραμματισμού χρειαστήκαμε για να δημιουργήσουμε το eclass για κινητά τηλέφωνα τη βάση δεδομένων χρησιμοποιήσαμε και άλλα σημαντικά εργαλεία που χρειάστηκαν για την υλοποίηση της πτυχιακής.

## Κεφάλαιο 2

### Πλατφόρμα E-class

## E-class

Ένα site σαν το e-class αποτελείται από διάφορα στοιχεία. Πρώτο στοιχείο είναι η πρόσοψη του site αυτό δηλαδή που εμφανίζεται στο πρόγραμμα πλοήγησης(browser) όταν πληκτρολογήσουμε την διεύθυνση του ιστότοπου. Το e-class χρησιμοποιεί επίσης μια βάση δεδομένων, η οποία δεν είναι ορατή, ώστε να αντλεί πληροφορίες για τον χρήστη όταν αυτός συνδέεται στο site και να αποθηκεύει τυχόν τροποποιήσεις που κάνει στο προφίλ του. Για κάθε χρήστη, είτε αυτός είναι φοιτητής είτε είναι καθηγητής είτε είναι ο διαχειριστής του ιστότοπου, υπάρχει ένα ξεχωριστό προφίλ στην βάση δεδομένων. Επίσης υπάρχει ένας διακομιστής-εξυπηρετητής του οποίου η δουλειά είναι να παράγει τις ιστοσελίδες και να τις αποστέλλει στο πρόγραμμα πλοήγησης.

Για την δημιουργία site υπάρχουν διαθέσιμα στο ίντερνετ διάφορα εργαλεία που σου επιτρέπουν να δημιουργήσεις ιστότοπους για προγράμματα πλοήγησης υπολογιστών αλλά και για κινητά τηλέφωνα. Για την ολοκλήρωση της πτυχιακής εργασίας επέλεξα να χρησιμοποιήσω την εφαρμογή Visual Studio της Microsoft. Σαν διακομιστή-εξυπηρετητή χρησιμοποίησα τον Apache, η βάση δεδομένων είναι MySql και για την OnLine διαχείριση της βάσης δεδομένων χρησιμοποίησα το phpMyAdmin . Αυτά είναι τα εργαλεία που χρησιμοποίησαν και οι προγραμματιστές που δημιούργησαν το e-class.

Γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την δημιουργία του ιστότοπου e-class

### NET Framework και C#

Το .NET Framework είναι ένα πλαίσιο λογισμικού που προορίζεται για την πλατφόρμα των Windows. Αποτελείται από μια μεγάλη βιβλιοθήκη κλάσεων και υποστηρίζει μια πλειάδα γλωσσών προγραμματισμού με την δυνατότητα η μια να μπορεί να χρησιμοποιηθεί από την άλλη. Τα προγράμματα που γράφονται για το .NET Framework εκτελούνται σε ένα περιβάλλον εκτέλεσης γνωστό ως Common Language Runtime (CLR), ενός ειδικού λογισμικού, σχεδιασμένου να υποστηρίζει την εκτέλεση προγραμμάτων και την απρόσκοπτη συνεργασία με το λειτουργικό σύστημα. Το CLR περιέχει μια εικονική μηχανή (virtual machine) που διαχειρίζεται την εκτέλεση ενός προγράμματος και παρέχει μια σειρά σημαντικών υπηρεσιών όπως ασφάλεια, διαχείριση μνήμης και διαχείριση εξαιρέσεων.

Τα προγράμματα που γράφονται για το .NET χρησιμοποιούν τη βιβλιοθήκη κλάσεων του .NET, η οποία δίνει πρόσβαση στο περιβάλλον εκτέλεσης του. Βασικές λειτουργίες όπως οι γραφικές διεπαφές χρηστών, η επικοινωνία με



βάσεις δεδομένων, η κρυπτογραφία, η ανάπτυξη Web εφαρμογών και οι δικτυακές επικοινωνίες παρέχονται μέσω του Application Programming Interface (API) του .NET και μπορούν να συνδυαστούν με κώδικα από του προγραμματιστές για τη δημιουργία ολοκληρωμένων εφαρμογών. Όσο ένα πρόγραμμα περιορίζεται στη χρήση της βιβλιοθήκης κλάσεων του .NET, μπορεί να τρέχει οπουδήποτε υπάρχει εγκατεστημένο το περιβάλλον εκτέλεσης του .NET.

## Χαρακτηριστικά του .NET Framework

### Υποστήριξη πολλαπλών γλωσσών

Στο .NET, η δια-λειτουργικότητα μεταξύ των γλωσσών επιτυγχάνεται μέσω της δια-γλωσσικής κληρονομικότητας. Μαζί με ένα ενοποιημένο σύστημα τύπων, η συνένωση κώδικα γραμμένου σε διαφορετικές γλώσσες είναι πλέον εύκολη υπόθεση. Αυτό διευκολύνει και άλλα προγραμματιστικά πρότυπα όπως το συναρτησιακό της F#

### Ανάπτυξη βασισμένη σε components

Η τέχνη της δημιουργίας ανεξάρτητων κομματιών λογισμικού που περιέχουν βιβλιοθήκες κλάσεων και διαμοιράσιμα τμήματα λειτουργικότητας μιας εφαρμογής έχει γίνει πολύ πιο εύκολη από παλιότερα. Η διαμοιράσιμη μονάδα κώδικα στο .NET βασίζεται στην έννοια του assembly, το οποίο μεταφέρει πληροφορίες που απαιτούνται για τη χρήση της όπως η έκδοση της.

### Μεταφερσιμότητα

Η μεταφερσιμότητα επιτυγχάνεται μέσω της εικονικής μηχανής στο CLR. Η εικονική μηχανή αυτή είναι η προδιαγραφή μιας αφηρημένης εικονικής μηχανής για την οποία κώδικας γραμμένος σε οποιαδήποτε από τις .NET γλώσσες μεταγλωττίζεται σε μια ενδιάμεση γλώσσα γνωστή ως Common Intermediate Language (CIL) και περιέχεται σε ένα assembly. Ένα assembly είναι είτε ένα εκτελέσιμο αρχείο (.exe) είτε ένα dll. Η μεταγλώττιση σε αυτήν την ενδιάμεση γλώσσα είναι που εξασφαλίζει και την ανεξαρτησία από την πλατφόρμα, η οποία ωστόσο μένει μόνο στη θεωρία, αφού η μοναδική πλήρης υλοποίηση του .NET είναι διαθέσιμη μόνο για την πλατφόρμα των Windows. Όμως είναι διαθέσιμη μια μερική υλοποίηση ανοιχτού κώδικα με το όνομα Mono.

### Απλή εγκατάσταση εφαρμογών

Αντίθετα με τα components που βασίζονται στο COM, δεν χρειάζεται καμία καταχώρηση στο registry του συστήματος για την εγκατάσταση των assemblies καθώς μια απλή αντιγραφή αρκεί για τη χρήση τους. Επιπρόσθετα τα πιθανά προβλήματα της χρήσης των Dynamic Link Libraries γνωστά και ως "DLL Hell" απαλείφθηκαν με την υποστήριξη της ύπαρξης πολλαπλών εκδόσεων του ίδιου component δίπλα δίπλα. Το ίδιο το .NET Framework είναι καλό παράδειγμα αυτής της δυνατότητας, αφού είναι δυνατόν να υπάρχουν πολλαπλές εκδόσεις του Framework εγκατεστημένες στο ίδιο μηχάνημα.

Επικοινωνία με υπάρχων κώδικα

Παρόλο που η πλατφόρμα .NET προορίζεται να είναι ένα υποκατάστατο των παλαιότερων τεχνολογιών, υποστηρίζει τη χρήση κομματιών λογισμικού που έχουν αναπτυχθεί σε παλαιότερες τεχνολογίες όπως της COM και παρέχει πρόσβαση σε λειτουργίες χαμηλού επιπέδου του λειτουργικού συστήματος μέσω του μηχανισμού P/Invoke.

Αξιοπιστία

Το .NET είναι σχεδιασμένο για τη δημιουργία λογισμικού υψηλής αξιοπιστίας (robustness). Πραγματοποιεί εκτεταμένους ελέγχους κατά τη μεταγλώττιση αλλά και κατά την εκτέλεση των προγραμμάτων. Τα χαρακτηριστικά του .NET από μόνα τους ωθούν τους προγραμματιστές στην απόκτηση αξιόπιστων προγραμματιστικών συνηθειών. Το μοντέλο διαχείρισης της μνήμης είναι απλό : η δέσμευση μνήμης γίνεται με μια λέξη, δεν υπάρχουν δείκτες προς θέσεις μνήμης σαν τύποι δεδομένων, ούτε αριθμητική δεικτών ενώ η αποδέσμευση μνήμης γίνεται αυτόματα μέσω του μηχανισμού garbage collection. Αυτό το απλό μοντέλο είναι διαχείρισης μνήμης απαλείφει ολόκληρες κατηγορίες λαθών που απασχολούν τους προγραμματιστές σε C C++. Η ανάπτυξη προγραμμάτων γίνεται με την πεποίθηση ότι αρκετά λάθη θα εντοπιστούν από το σύστημα πολύ πριν την ολοκλήρωση της.

Αυτόματη διαχείριση μνήμης

Άλλο ένα σημαντικό χαρακτηριστικό του .NET Framework είναι η δυνατότητα αυτόματης διαχείρισης της μνήμης μέσω του Garbage Collector. Ιστορικά η διαχείριση της μνήμης ήταν από τις δυσκολότερες εργασίες που είχε να κάνει ένας προγραμματιστής. Οι παλαιότερες προσεγγίσεις στη διαχείριση μνήμης ήταν είτε χαμηλού επιπέδου (όπως η malloc/free στη C και η new/delete στη C++), προκαλώντας διάφορα bugs, είτε αρκετά πολύπλοκη (όπως στην περίπτωση του COM). Ένας από τους στόχους του .NET και του CLR πιο συγκεκριμένα ήταν να απαλείψει από τους προγραμματιστές την ανάγκη διαχείρισης της μνήμης.

Το CLR με το μηχανισμό Garbage Collector διαχειρίζεται τη μνήμη ερευνώντας τότε να ελευθερωθεί. Η μνήμη δεσμεύεται με την αρχικοποίηση των διαφόρων τύπων (αντικειμένων) από ένα διαθέσιμο τμήμα της μνήμης που διαχειρίζεται το CLR γνωστό ως heap. Όσο υπάρχει κάποια άμεση ή έμμεση ( μέσω του γράφου των αντικειμένων) αναφορά προς ένα αντικείμενο, τότε αυτό θεωρείται πως χρησιμοποιείται. Στην αντίθετη περίπτωση, όταν δεν μπορεί να αναφερθεί από κάποιο άλλο τμήμα του κώδικα τότε είναι διαθέσιμο για καταστροφή. Ο garbage collector τρέχει ανά τακτά χρονικά διαστήματα σε ένα ξεχωριστό thread, εντοπίζει ποια αντικείμενα είναι έτοιμα προς καταστροφή και αποδεσμεύει τη μνήμη που αυτά χρησιμοποιούσαν.

Ασφάλεια

Το .NET είναι σχεδιασμένο να λειτουργεί σε κατανεμημένα περιβάλλοντα, η ασφάλεια των οποίων είναι ύψιστη σημασίας. Με μέτρα που είναι ενσωματωμένα

στις γλώσσες αλλά και κατά την εκτέλεση των προγραμμάτων, το .NET επιτρέπει τη δημιουργία προγραμμάτων που δεν μπορούν να προσβληθούν από έξω. Σε δικτυακά περιβάλλοντα οι εφαρμογές του .NET είναι ασφαλείς από εισβολές μη εξουσιοδοτημένου κώδικα που επιχειρεί να δράσει στο παρασκήνιο και να εγκαταστήσει κακόβουλο λογισμικό ή να εισβάλει σε συστήματα αρχείων. Το περιβάλλον εκτέλεσης του .NET έχει ενσωματωμένο έναν μηχανισμό ασφαλείας με το όνομα Code Access Security που απομονώνει τον κώδικα ανάλογα με την προέλευσή του, τον δημιουργό του ή άλλες πολιτικές. Το μοντέλο ασφαλείας συμβαδίζει με τους μηχανισμούς ασφαλείας που παρέχει το λειτουργικό σύστημα όπως οι Access Control List και τα Windows security tokens.

#### Απόδοση

Η ενδιάμεση γλώσσα στην οποία μεταγλωττίζονται τα προγράμματα στο .NET, μεταγλωττίζεται πάντα με τη μέθοδο Just-in-Time. Ο μεταγλωττιστής JIT μεταγλωττίζει κάθε τμήμα του κώδικα στην ενδιάμεση γλώσσα όταν αυτό καλείται. Όταν έχει ήδη μεταγλωττιστεί κάποιο τμήμα κώδικα, το αποτέλεσμα αποθηκεύεται μέχρι η εφαρμογή να τερματίσει έτσι ώστε να μην χρειάζεται να ξαναμεταγλωττιστεί όταν ζητηθεί το συγκεκριμένο τμήμα. Η διαδικασία αυτή είναι αποδοτικότερη σε σύγκριση με την απευθείας μεταγλώττιση ολόκληρου του κώδικα, γιατί υπάρχει περίπτωση μεγάλα τμήματα κώδικα να μην εκτελεστούν ποτέ στην πραγματικότητα. Με τον μεταγλωττιστή JIT τέτοιος κώδικας δεν θα μεταγλωττίζοταν ποτέ.

#### Υποστήριξη με επαγγελματικά εργαλεία

Το .NET Framework συνοδεύεται από ένα σύνολο εργαλείων για την υποστήριξη της ανάπτυξης λογισμικού σε αυτό, όπως το σύστημα σχεδιασμού, τα Frameworks για unit testing, την πλατφόρμα για build MSBuild και το σύστημα debugging, μέσα από το Visual Studio. Το Visual Studio είναι ένα από τα καλύτερα Integrated Development Environments. Χρησιμοποιείται για την ανάπτυξη όλων των ειδών εφαρμογών που μπορούν να αναπτυχθούν στο .NET : από console εφαρμογές και εφαρμογές με γραφικές διεπαφές μέχρι Web εφαρμογές και Web Services σε managed αλλά και εγγενή κώδικα. Οι Premium και Ultimate εκδόσεις του Visual Studio στοχεύουν στην ομαδική ανάπτυξη μαζί με τον Team Foundation Server που χρησιμοποιείται για την παρακολούθηση ενός έργου .NET, τη διαχείριση του κώδικα, τη διαχείριση των αναφορών αλλά και το διαμοιρασμό κώδικα, τεκμηριώσεων ή άλλων εγγράφων.

## Αρχιτεκτονική του .NET Framework

Ένας προγραμματιστής στην πλατφόρμα Windows έχει πρωταρχικά δυο μονοπάτια που μπορεί να ακολουθήσει κατά την ανάπτυξη εφαρμογών. Το πρώτο είναι η ανάπτυξη εφαρμογών μη διαχειρισμένου κώδικα και περιλαμβάνει τον προγραμματισμό κατευθείαν στο λειτουργικό σύστημα, χρησιμοποιώντας το Windows Application Programming Interface συνήθως με τη γλώσσα C++. Στόχος του Windows API είναι να δώσει τη δυνατότητα στους προγραμματιστές να αναπτύξουν προγράμματα με το λειτουργικό σύστημα για να αποφύγουν το γράψιμο κώδικα για τη δημιουργία components των Windows όπως φόρμες, κουμπιά μενού και άλλα. Με το Windows API ένας προγραμματιστής καλεί τις συναρτήσεις και αφήνει το λειτουργικό σύστημα να δημιουργήσει αυτά τα components.

Ο κύριος λόγος χρήσης του Windows API είναι η απουσία υποστήριξης διαφόρων λειτουργιών από τη βασική έκδοση της χρησιμοποιούμενης γλώσσας προγραμματισμού. Η χρήση μιας λειτουργίας του Windows API από μια γλώσσα γίνεται με την κλήση της κατάλληλης συνάρτησης. Ωστόσο, οι κλήσεις τέτοιων συναρτήσεων είναι συνήθως μια πολύπλοκη διαδικασία που μπορεί να προκαλέσει απρόβλεπτες συνέπειες αν δεν γίνει σωστά. Η πολυπλοκότητα αυτή είναι ένας από τους λόγους που οδήγησε στη δημιουργία του δεύτερου μονοπατιού στην ανάπτυξη εφαρμογών, δηλαδή την ανάπτυξη εφαρμογών διαχειρισμένου κώδικα.

Διαχειρισμένος κώδικας είναι ουσιαστικά κώδικας που τρέχει στην πλατφόρμα .NET. Το .NET αποτελείται από ένα περιβάλλον εκτέλεσης το οποίο διαχειρίζεται την εκτέλεση κώδικα μέσω κανόνων ασφάλειας, αυτόματης διαχείρισης της μνήμης και αυστηρότητας τύπων, μειώνοντας έτσι σε μεγάλο βαθμό την πιθανότητα πρόκλησης λαθών από αυτούς τους παράγοντες.

Η πλατφόρμα αποτελείται από διάφορα μέρη καθένα από τα οποία παρέχει ένα συγκεκριμένο κομμάτι των δυνατοτήτων της. Αυτά μπορεί να είναι αυτόνομα προγράμματα βιβλιοθήκες ή βοηθητικά αρχεία.

### Common Language Runtime

Το Common Language Runtime είναι μια υλοποίηση του προτύπου Common Language Infrastructure από τη Microsoft. Το CLI προσδιορίζει την υποδομή εκτέλεσης που απαιτείται για τη χρήση managed κώδικα, έτσι ώστε διαφορετικές βιβλιοθήκες και γλώσσες να συνεργάζονται απρόσκοπτα. Υπάρχουν και άλλες υλοποιήσεις του CLI από τρίτους παρόχους όπως το Mono και το DotGNU Portable.NET αλλά και της Microsoft με το όνομα Shared Source CLI γνωστό και ως Rotor, το οποίο τρέχει σε πλατφόρμες όπως το FreeBSD. Αυτό θεωρητικά σημαίνει ότι κώδικα που στοχεύει στο .NET όχι μόνο επιτρέπει τη διαλειτουργικότητα μεταξύ διαφορετικών γλωσσών προγραμματισμού αλλά και την εκτέλεση σε διαφορετικές πλατφόρμες.

Η προδιαγραφή CLI και κατά συνέπεια το CLR αποτελείται από τρία κύρια στοιχεία

- Το Virtual Execution System (VES)
- Το Common Type System (CTS)
- Το Common Language Specification (CLS)

Η VES είναι η καρδιά του CLR και είναι αυτή που εξασφαλίζει τη μεταφραστικότητα των προγραμμάτων σε .NET. Η VES αντιλαμβάνεται το format της ενδιάμεσης γλώσσας Common Intermediate Language (CIL) και τα μεταδεδομένα που περιγράφουν τον κώδικα σε αυτήν την ενδιάμεση μορφή. Ο κώδικας σε CIL μορφή είναι ίδιος ανεξάρτητα από το υλικό ή την πλατφόρμα όπου εκτελείται. Η VES είναι ουσιαστικά μια εικονική μηχανή διεργασίας (process virtual machine). Τέτοιες μηχανές τρέχουν σαν κανονική εφαρμογή μέσα στο λειτουργικό σύστημα στο οποίο φιλοξενούνται και υποστηρίζονται από μια διεργασία. Η λειτουργία τους ξεκινά με την έναρξη της διεργασίας και παύει με τον τερματισμό της. Ο σκοπός των εικονικών μηχανών είναι να παρέχουν ένα περιβάλλον εκτέλεσης προγραμμάτων ανεξάρτητο συνήθως από την πλατφόρμα που τρέχουν, η οποία κρύβει τις λεπτομέρειές του υποκείμενου υλικού ή λειτουργικού συστήματος, επιτρέποντας έτσι στα προγράμματα να τρέχουν με τον ίδιο τρόπο σε κάθε πλατφόρμα. Δεν πρέπει να συγχέονται με τις εικονικές μηχανές συστήματος που παρέχουν μια πλήρη πλατφόρμα για την εκτέλεση ενός ολόκληρου λειτουργικού συστήματος όπως η Virtual Box, VMWare ή η Hyper-V.

Για να εκτελεστεί ο κώδικας μιας γλώσσας στο CLR, πρέπει πρώτα να μεταγλωττιστεί σε μια μορφή η οποία μπορεί να αναγνωριστεί κατά την εκτέλεση. Αυτή η μορφή είναι η Common Intermediate Language. Ο κώδικας που εκτελείται στο CLR είναι γνωστός ως managed κώδικας, σε αντίθεση με τον εγγενή κώδικα που εκτελείται κατευθείαν στον επεξεργαστή της μηχανής χωρίς επιπρόσθετη υποδομή εκτέλεσης. Εκτός από τις εντολές σε IL για την εκτέλεση μιας εφαρμογής το CLR χρειάζεται μεταδεδομένα που περιγράφουν τη δομή και την οργάνωση των τύπων δεδομένων. Η χρήση των μεταδεδομένων απαλείφει την ανάγκη ύπαρξης header files και βοηθά διάφορα εργαλεία να επιθεωρήσουν τους τύπους. Ένα τέτοιο παράδειγμα είναι το Intellisense του Visual Studio. Η ενδιάμεση γλώσσα CIL περιέχεται μέσα στα assemblies τα οποία, εκτός από το ρόλο τους στην εκτέλεση ενός προγράμματος, αποτελούν και τις βασικές μονάδες εγκατάστασης στο .NET. Εξαιτίας της αυτοπεριγραφικής φύσης τους που παρέχεται από τα μεταδεδομένα, τα assemblies δεν χρειάζονται την βοήθεια πολύπλοκων οδηγιών εγκατάστασης για την εγγραφή τους στο registry, αφού μια απλή αντιγραφή τους αρκεί, ενώ είναι δυνατή η ύπαρξη πολλαπλών εκδόσεων των assemblies αφού ενσωματώνουν τον αριθμό έκδοσής τους.

Ένα από τα σημαντικότερα κομμάτια του CLR είναι το CTS γιατί εξασφαλίζει τη δια λειτουργικότητα μεταξύ των γλωσσών. Το CTS ορίζει τους προκαθορισμένους τύπους δεδομένων που είναι διαθέσιμες στην ενδιάμεση

γλώσσα έτσι ώστε όλες οι γλώσσες του .NET να παράγουν μεταγλωττισμένο κώδικα που βασίζεται σε αυτούς τους τύπους.

Το τελευταίο σημαντικότερο κομμάτι του CLR είναι το Common Language Specification. Το CLS είναι ένα σύνολο κανόνων στο οποίο βασίζονται όλες οι γλώσσες του .NET για να εξασφαλίσουν ότι δεν τίθενται προς χρήση τύποι δεδομένων τους οποίους δεν μπορούν να χειριστούν μερικές γλώσσες. Αν τηρούνται όλοι οι κανόνες ένας τύπος ορίζεται ως συμβατός με το CLS και κατά συνέπεια μπορεί να χρησιμοποιηθεί από όλες τις γλώσσες. Ένα παράδειγμα ενός κανόνα είναι ένας τύπος δεν μπορεί να έχει public μέλη που διακρίνονται μόνο από το αν οι χαρακτήρες τους είναι πεζοί ή κεφαλαίοι. Μερικές γλώσσες, όπως η Visual Basic, δεν κάνουν διάκριση σε πεζούς και κεφαλαίους χαρακτήρες και δεν θα μπορούσαν να αντιληφθούν τη διαφορά.

#### Βιβλιοθήκες κλάσεων

Τα περισσότερα σύγχρονα λειτουργικά συστήματα παρέχουν ένα σύνολο επαναχρησιμοποιήσιμου κώδικα για τη διευκόλυνση της ανάπτυξης προγραμμάτων σε αυτά. Ο κώδικας αυτός παρέχεται συνήθως σαν ένα σύνολο βιβλιοθηκών που μπορούν να καλούν οι εφαρμογές κατά την εκτέλεσή τους, όπως οι Dynamic Link Libraries στα Windows και τα Dynamic Shared Objects στα συστήματα Unix. Τα DLLs αυτά, που βρίσκονται στο φάκελο Windows\System 32, περιέχουν συναρτήσεις που χρησιμοποιούνται για την απρόσκοπτη λειτουργία του λειτουργικού συστήματος και τη διατήρηση ενός συνεπούς παραθυρικού περιβάλλοντος. Αυτά τα αρχεία αποτελούν το Windows API.

Προφανώς, το περιβάλλον εκτέλεσης συνετέλεσε αρκετά στην επιτυχία του .NET αλλά πολύ σημαντικότερη για τους προγραμματιστές είναι η πλούσια βιβλιοθήκη κλάσεων η οποία δίνει πρόσβαση σε έναν πλούτο λειτουργικότητας που μπορεί να χρησιμοποιηθεί από τις εφαρμογές. Οι βιβλιοθήκες κλάσεων του .NET υπηρετούν κυρίως δύο σκοπούς : α) όπως άλλες βιβλιοθήκες παρέχον ένα καλά ορισμένο σύνολο κώδικα για την πραγματοποίηση τετριμμένων εργασιών, όπως η διατήρηση λιστών αντικειμένων ή η επεξεργασία συμβολοσειρών και β) παρέχουν μια αφηρημένη υλοποίηση για εργασίες που κανονικά θα απαιτούσαν πρόσβαση στο υλικό και το λειτουργικό σύστημα, όπως η χρήση του συστήματος αρχείων ή η πρόσβαση στο δίκτυο.

Η βιβλιοθήκη κλάσεων του .NET έχει μεγαλώσει αρκετά από την πρώτη έκδοση του περίπου το 2000. Σήμερα, είναι σχεδόν αδύνατο να δοθεί πλήρης κάλυψη του framework ακόμα και σε ένα βιβλίο. Αντίθετα είναι πολύ πιο σημαντικό να κατανοηθεί η βασική δομή του έτσι ώστε η εξερεύνηση του να γίνεται εύκολα. Σπάνια κανείς θα έχει την πλήρη λύση ενός προβλήματος όταν ξεκινάει την ανάπτυξη μιας εφαρμογής, αλλά έχοντας κατανοήσει τη δομή του, οτιδήποτε στη βιβλιοθήκη μπορεί να μαθευτεί και να χρησιμοποιηθεί.

Δεδομένων των τεράστιων διαστάσεων της βιβλιοθήκης, είναι απαραίτητη η οργάνωση της για να διευκολυνθεί ο εντοπισμός των διαφόρων λειτουργιών της. Η προσέγγιση που ακολουθείται στο .NET είναι η ιεραρχική οργάνωση των παρεχόμενων τύπων με τη χρήση των namespaces, σε αντίθεση με το Win32

API όπου η οργάνωση είναι επίπεδη. Ακολουθούν ορισμένα γνωστά namespaces και η λειτουργικότητα που παρέχουν:

- Το system περιέχει του primitive τύπους δεδομένων, όπως τους Int32, Double, String, Booleans.
- Το System.Collections παρέχει πρόσβαση σε χρήσιμους τύπους συλλογών δεδομένων όπως λίστες, σύνολα και λεξικά.
- Το System.Data που παρέχει λειτουργικότητα για την πρόσβαση σε OnLine ή offline δεδομένα με κύριο στόχο τις βάσεις δεδομένων.
- Το System.IO που παρέχει τις απαραίτητες δομές για τη διαχείριση της Εισόδου/Εξόδου (input/output) με streams, αρχεία, pipes κ.α.
- Το System.Net που περιέχει κλάσεις για τη μετάδοση δεδομένων μέσω δικτύου που υποστηρίζουν γνωστά πρωτόκολλα όπως το TCP,UDP και το HTTP.
- Το System.Security που ενσωματώνει όλη τη λειτουργικότητα σχετική με την ασφάλεια μια εφαρμογής, όπως permissions και κρυπτογραφία.
- Το System.Text που χρησιμεύει για τη διαχείριση κειμένου, υποστηρίζοντας διάφορες κωδικοποιήσεις αλλά και τη χρήση regular expressions.
- Το System.Windows που φιλοξενεί διάφορες τεχνολογίες για γραφικά περιβάλλοντα όπως τις Windows Forms και WPF.
- Το System.XML που υποστηρίζει την επεξεργασία εγγράφων XML.

#### Γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού είναι θεωρητικά εκτός την έννοιας μια πλατφόρμας λογισμικού, ωστόσο η σημασία τους στο .NET Framework είναι αρκετά μεγάλη. Το .NET υποστηρίζει τη χρήση πολλαπλών γλωσσών για τη δημιουργία εφαρμογών και βιβλιοθηκών κλάσεων. Από την πρώτη έκδοση του .NET, τόσο η Microsoft όσο και τρίτοι πάροχοι προσέφεραν ένα πλούσιο σύνολο γλωσσών συμβατών με το περιβάλλον εκτέλεσης. Οι κύριες γλώσσες που παρέχονται από τη Microsoft σαν κομμάτι του .NET Framework SDK ωστόσο είναι τέσσερις :

- Η Visual Basic .NET : Μια γλώσσα με την οποία εξοικειώνονται αμέσως όσοι έχουν γράψει κλασικά προγράμματα Visual Basic στο παρελθόν και που τώρα στοχεύει στο .NET Framework με πλήρη υποστήριξη αντικειμενοστρέφιας.
- Η C# : Η γλώσσα που αναπτύχθηκε παράλληλα με τη δημιουργία του περιβάλλοντος εκτέλεσης του .NET και την πρώτη έκδοση των βιβλιοθηκών κλάσεων. Βασίζεται στις αρχές του αντικειμενοστραφούς προγραμματισμού και η φύση της είναι πολύ πρακτική.
- Η C++/CLI: αρχικά γνωστή ως Managed Extension for C++, επιτρέπει την εκτέλεση C++ κώδικα πάνω στο περιβάλλον εκτέλεσης του .NET παρέχοντας ένα πολύ καλό εργαλείο για την αντιμετώπιση δια λειτουργικών προβλημάτων.
- Η F# : η τελευταία προσθήκη στον πυρήνα γλωσσών του .NET που στοχεύει στην επιστημονική υπολογιστική και συνδυάζει τις αρχές του

συναρτησιακού προγραμματισμού με τον πλούτο της βιβλιοθήκης κλάσεων του .NET

Εκτός των επίσημων γλωσσών της Microsoft, διάφορες άλλες γλώσσες έχουν εισαχθεί στο .NET Framework. Παραδείγματα αποτελούν η COBOL, η Smalltalk, η Delphi, η APL, η Pascal και άλλες.

Το τοπίο των γλωσσών προγραμματισμού είναι πάντα ρευστό και η ενοποιημένη υποδομή του .NET παρέχει έναν ασφαλή τρόπο για την εξυπηρέτηση των αναγκών των διαφορετικών αυτών γλωσσών, δίνοντας τους τη δυνατότητα να χρησιμοποιήσουν βιβλιοθήκες μεταξύ τους και να χρησιμοποιήσουν τα έτοιμα εργαλεία. Όλα αυτά επιτρέπουν στους προγραμματιστές με διαφορετικές προτιμήσεις σε γλώσσες προγραμματισμού να δουλέψουν μαζί και να επιλέξουν τα σωστά εργαλεία για κάθε έργο. Για παράδειγμα, ενώ η C# είναι μια πολύ καλή γλώσσα γενικού σκοπού για τη δημιουργία εταιρικών εφαρμογών, η F# είναι πιο ελκυστική για την ανάπτυξη επιστημονικών εφαρμογών.

#### Εργαλεία του .NET Framework

Το .NET περιλαμβάνει ένα σύνολο εργαλείων για τη διευκόλυνση της δημιουργίας, εγκατάστασης και διαχείρισης εφαρμογών και components σε αυτό. Τα περισσότερα εργαλεία εγκαθίστανται με την εγκατάσταση του Visual Studio. Όλα τα εργαλεία είναι δυνατόν να εκτελεστούν από τη γραμμή εντολών των Windows εκτός από το εργαλείο Assembly Cache Viewer που πρέπει να χρησιμοποιηθεί από τον Windows Explorer. Ο καλύτερος τρόπος εκτέλεσης των command-line εργαλείων είναι μέσα από το Visual Studio Command Prompt (γνωστό και ως CMD Shell) που παρέχεται σε κάθε Visual Studio από την έκδοση 2005 και μετά. Ακολουθεί μια αναφορά σε επιλεγμένα εργαλεία του .NET και μια κατηγοριοποίηση τους:

- Εργαλεία μεταγλώττισης :
- Csc.exe (C sharp Compiler) : ο μεταγλωττιστής της C# λαμβάνει αρχεία που περιέχουν κώδικα C# και παράγει εκτελέσιμα αρχεία (exe), dynamic link libraries ή modules κώδικα
- Εργαλεία Build, Deployment και Configuration
  - Ildasm.exe (MSIL Assembler) : λαμβάνει ένα portable executable αρχείο που περιέχει κώδικα στην ενδιάμεση γλώσσα MSIL και παράγει ένα text αρχείο που μπορεί να προωθηθεί στο εργαλείο MSIL Assembler
  - Mage.exe: Επιτρέπει τη δημιουργία, επεξεργασία και υπογραφή των manifest μιας εφαρμογής.
- Εργαλεία αποσφαλμάτωσης( Debugging)
  - MDbg.exe: Βοηθάει τους προγραμματιστές να εντοπίζουν και να διορθώνουν bugs στα προγράμματα που τρέχουν στο CLR. Αυτό το εργαλείο χρησιμοποιεί το runtime debugging API.
- Εργαλεία ασφάλειας
  - Peverify.exe: βοηθάει στην επιβεβαίωση ότι ο ενδιάμεσος κώδικας MSIL και τα αντίστοιχα μεταδεδομένα καλύπτουν τις απαιτήσεις ασφαλείας.
- Εργαλεία διαλειτουργικότητας



- Regasm.exe: διαβάζει τα μεταδεδομένα σε ένα assembly και προσθέτει τις απαραίτητες εγγραφές στο registry. Αυτό επιτρέπει στους COM clients να εμφανίζονται σαν κλάσεις του .NET
- Διάφορα άλλα εργαλεία:
  - SqlMetal.exe: παράγει κώδικα και αντιστοιχίες για το LINQ το SQL component του .NET.
  - Wincl.exe: εμφανίζει πληροφορίες για μια κλάση ή σύνολο κλάσεων χρησιμοποιώντας το reflection API του CLR.

## Η γλώσσα προγραμματισμού C#

Η C# είναι κατά κύριο λόγο μια αντικειμενοστραφής γλώσσα προγραμματισμού που ενσωματώνει ωστόσο μερικά χαρακτηριστικά διαφόρων προγραμματιστικών προτύπων. Αναπτύχθηκε στη Microsoft, από μια ομάδα κάτω από την ηγεσία του Anders Hejberg, σαν μέρος του .NET Framework.

Η C# είναι μια συνεχώς εξελισσόμενη γλώσσα και με κάθε νέα της έκδοση προστίθενται νέα χαρακτηριστικά και συντακτικό, με μόνιμο στόχο να κάνει τα απλά πράγματα εύκολα, τα δύσκολα πράγματα εφικτά και άσχημα πράγματα δύσκολα. Ακολουθεί μια σύντομη επεξήγηση των χαρακτηριστικών της C# .

## Χαρακτηριστικά της C#

### Απλότητα

Η C# είναι μια απλή γλώσσα που μπορεί να χρησιμοποιηθεί χωρίς εντατική εκμάθηση, ενώ ταυτόχρονα είναι εναρμονισμένη με σύγχρονες προγραμματιστικές πρακτικές. Οι θεμελιώδεις αρχές της γλώσσας μπορούν να κατανοηθούν γρήγορα κάτι που σημαίνει ότι οι προγραμματιστές θα είναι παραγωγικοί σε σύντομο χρονικό διάστημα. Η C# έχει σχεδιαστεί έτσι ώστε να μειώνεται η πιθανότητα πρόκλησης λαθών από την πολυπλοκότητα του κώδικα, αφού τη μειώνει σε μεγάλο βαθμό με το απλουστευμένο συντακτικό της και την οργάνωση κώδικά της.

### Αντικειμενοστρέφεια

Η C# από τα θεμέλια της σχεδιάστηκε να είναι αντικειμενοστραφής. Ο αντικειμενοστραφής προγραμματισμός επικράτησε σαν προγραμματιστικό πρότυπο την προηγούμενη δεκαετία και παραμένει στις πρώτες προτιμήσεις των προγραμματιστών. Οι ανάγκες για κατανεμημένα συστήματα πελάτη-εξυπηρέτηση συμπίπτουν με την ενθουσία και την ανταλλαγή μηνυμάτων που είναι βασικά χαρακτηριστικά του αντικειμενοστρεφούς προγραμματισμού. Κατά πολλούς ειδικούς στις γλώσσες προγραμματισμού, η επιτυχής λειτουργία των προγραμματιστικών συστημάτων σε δικτυακά περιβάλλοντα αυξανόμενης πολυπλοκότητας βασίζεται στην αντικειμενοστρέφεια. Η C# παρέχει μια ξεκάθαρη και αποδοτική αντικειμενοστραφή πλατφόρμα παρέχοντας στους προγραμματιστές μια συλλογή βιβλιοθηκών δοκιμασμένων αντικειμένων που παρέχουν λειτουργικότητα που ποικίλει από απλούς τύπους δεδομένων, σε διεπαφές εισόδου/εξόδου ή δικτυακές και εργαλεία για τη δημιουργία παραθυρικών εφαρμογών. Αυτές οι βιβλιοθήκες μπορούν να προσαρμοστούν στις ανάγκες του προγραμματιστή.

Επιπρόσθετα η C# υποστηρίζει και τον προγραμματισμό βασισμένο σε components ο οποίος επιτρέπει τον προσδιορισμό αυτόνομων μονάδων λειτουργικότητας που είναι απομονωμένα και τεκμηριωμένα, παρουσιάζοντας ένα μοντέλο με ιδιότητες, μεθόδους, events και μεταδεδομένα για το component. Η C# υποστηρίζει αυτά τα χαρακτηριστικά άμεσα κάνοντας έτσι τη διαδικασία δημιουργίας και χρήσης των components πολύ εύκολη.

# HTML

## Προέλευση

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο Cern, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε το πρόγραμμα πλοήγησης και το λογισμικό του εξυπηρετητή.

Η HTML ( ακρωνύμιο του αγγλικού Hyper Text Markup Language) είναι η βασική γλώσσα σήμανσης που χρησιμοποιούμε για την δημιουργία ιστοσελίδων, και τα στοιχεία της αποτελούν τα δομικά στοιχεία των ιστοσελίδων. Τα στοιχεία HTML αποτελούνται από ετικέτες που λειτουργούν ανά ζεύγη οι οποίες περικλείονται μέσα σε σύμβολα, «μεγαλύτερο από» και «μικρότερο από», για παράδειγμα `<h1>` και `</h1>` . Ανάμεσα στις ετικέτες, υπάρχει η δυνατότητα να τοποθετήσουμε κείμενο, πίνακες, εικόνες κλπ.

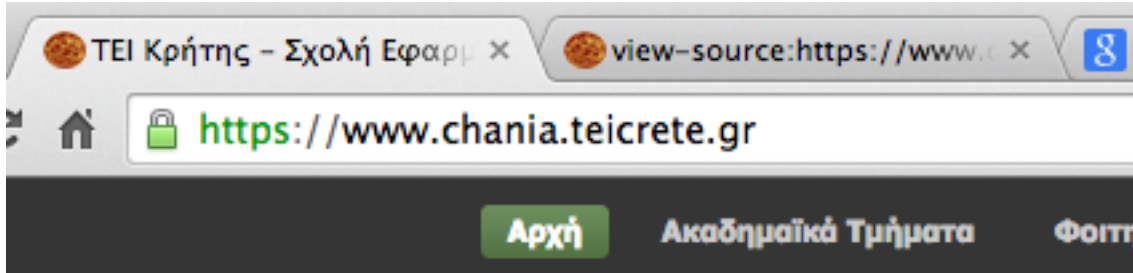
Ο σκοπός των προγραμμάτων πλοήγησης είναι να διαβάζουν τα έγγραφα HTML και να τα συνθέτουν σε σελίδες που μπορεί κανείς να τα διαβάσει ή να τα ακούσει. Το πρόγραμμα πλοήγησης δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους τους ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει δια δραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων ( εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται με σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.



Το κεφάλι του κώδικα HTML

Το κεφάλι περιέχει πληροφορίες για το HTML αρχείο μας όπως τον τίτλο. Ο τίτλος είναι αυτό που βλέπουμε στην σελίδα του προγράμματος πλοήγησης όπως φαίνεται στην παρακάτω φωτογραφία.



εικ 2) Ο τίτλος της ιστοσελίδας του TEI που εμφανίζεται στο πρόγραμμα πλοήγησης «ΤΕΙ Κρήτης-Σχολή Εφαρμοσμένων Επιστημών»

Η παρακάτω εικόνα μας δείχνει τον κώδικα που γράφτηκε ώστε να εμφανίζεται αυτός ο τίτλος στο πρόγραμμα πλοήγησης. Ο κώδικας είναι γραμμένος μέσα στο κομμάτι <head> </head> του html αρχείου και πιο συγκεκριμένα στο : <title> TEI Κρήτης-Σχολή Εφαρμοσμένων Επιστημών</title>.

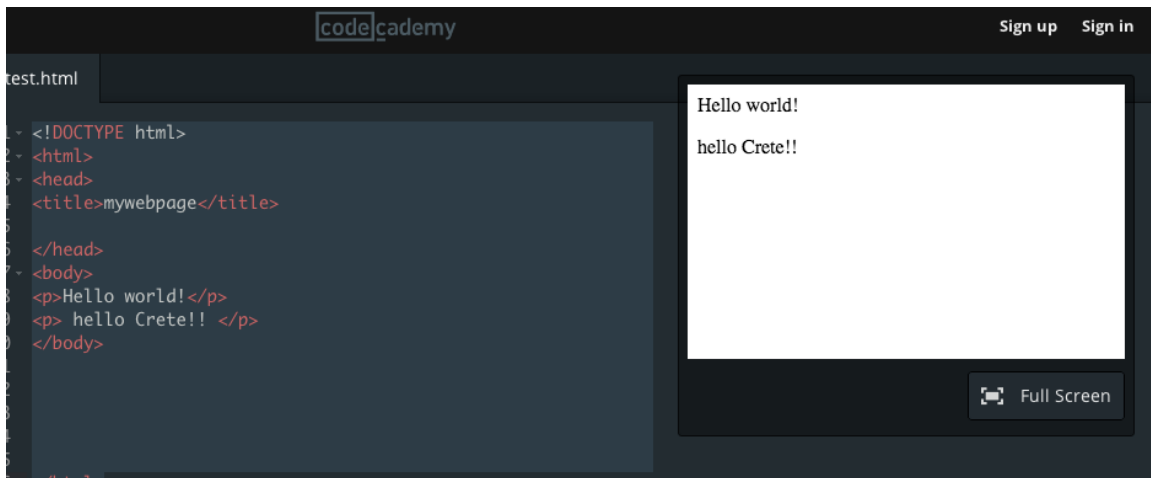
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="el-gr" lang="el-gr">
<head>
  <base href="https://www.chania.teicrete.gr/" />
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta name="robots" content="index, follow" />
  <meta name="keywords" content="TEI, Κρήτης, Σχολή, Χανιά, τμήμα, ηλεκτρονικής, φυσι
Εφαρμοσμένων Επιστημών, Σχολή Εφαρμοσμένων Επιστημών Χανιά" />
  <meta name="description" content="TEI Κρήτης - Σχολή Εφαρμοσμένων Επιστημών με έδρ
Φυσικών Πόρων και Περιβάλλοντος Τ.Ε." />
  <meta name="generator" content="Joomla! 1.5 - Open Source Content Management" />
  <title>TEI Κρήτης - Σχολή Εφαρμοσμένων Επιστημών</title>
```

εικ 3) Ο κώδικας HTML της ιστοσελίδας του TEI που μας δίνει αυτόν τον τίτλο

Το σώμα του κώδικα HTML

Στο σώμα είναι το κομμάτι που θα βάλουμε το περιεχόμενο της ιστοσελίδας, όπως κείμενα, εικόνες και συνδέσμους. Το περιεχόμενο του σώματος είναι αυτό που βλέπουμε στην ιστοσελίδα μας.

Για παράδειγμα έστω ότι θέλω να φτιάξω μια ιστοσελίδα η οποία θα μου εμφανίζει hello world και hello Crete. Τότε ο κώδικας που χρειάζεται να γράψω είναι ο εξής :

A screenshot of a code editor interface. The top left corner shows the file name 'test.html'. The top right corner has 'codecademy' and 'Sign up Sign in' links. The main area is split into two panes. The left pane shows the following HTML code:

```
1 - <!DOCTYPE html>
2 - <html>
3 - <head>
4 - <title>mywebpage</title>
5
6 </head>
7 - <body>
8 <p>Hello world!</p>
9 <p> hello Crete!! </p>
10 </body>
```

The right pane shows the rendered output of the code, which is a white box containing the text:

```
Hello world!
hello Crete!!
```

At the bottom right of the right pane, there is a 'Full Screen' button.

εικ 4) Ο κώδικας HTML και το αποτέλεσμα του

Στο αριστερό μέρος της οθόνης μέσα στο κορμί του κώδικα ανοίξαμε δύο παραγράφους με τις ετικέτες `<p></p>` και γράψαμε το κείμενο της επιλογής μας. Στο δεξί μέρος της οθόνης εμφανίζεται το επιθυμητό αποτέλεσμα.

## Δημιουργία υπερσυνδέσεων

Πολλές φορές κομμάτια του κειμένου στα site που διαβάζουμε είναι συνδέσεις σε άλλες ιστοσελίδες. Το παρακάτω παράδειγμα μας δείχνει τον τρόπο για να κάνουμε μια τέτοια σύνδεση.



The screenshot shows a code editor interface with a dark theme. At the top, there is a logo for 'codecademy' and links for 'Sign up' and 'Sign in'. The editor is open to a file named 'index.html'. The code is as follows:

```
1 - <!DOCTYPE html>
2 - <html>
3 -   <head>
4 -     <title></title>
5 -   </head>
6 -   <body>
7 -     <a href = "https://www.chania.teicrete.gr/"> TEI </a>
8 -   </body>
9 - </html>
```

To the right of the code editor, there is a preview window showing the rendered output of the code. It displays the text 'TEI' as a blue, underlined hyperlink. Below the preview window, there is a 'Full Screen' button.

εικ 5) Ο κώδικας που μας δίνει την δυνατότητα δημιουργίας υπερσυνδέσεων (hyperlink)

Για την δημιουργία υπερσυνδέσεων ο κώδικας είναι ο εξής . Αρχικά ανοίγουμε μια ετικέτα, μέσα στο σώμα του κώδικα html, <a η οποία περιέχει ένα χαρακτηριστικό το href το οποίο μας στέλνει στην σελίδα που επιθυμούμε. Τοποθετούμε δηλαδή την διεύθυνση της σελίδας που επιθυμούμε να μας πάει ο υπερσύνδεσμος με τον εξής τρόπο

<a href = " https://www.examplepage.gr/">. Στην συνέχεια γράφουμε το κείμενο ή την λέξη με την οποία θέλουμε να εμφανίζεται αυτός ο υπερσύνδεσμος στην ιστοσελίδα μας και κλείνουμε την ετικέτα </a>.

<a href = " https://www.examplepage.gr/">Παράδειγμα </a>

## Css

Με τον όρο CSS αναφερόμαστε σε μια γλώσσα μορφοποίησης σελίδων HTML. Για παράδειγμα, ένα φύλλο CSS μπορεί να ρυθμίσει την εμφάνιση των γραμματοσειρών, τα χρώματα, τα περιθώρια, τις γραμμές, το ύψος, το πλάτος, τις εικόνες φόντου μιας ιστοσελίδας και πολλά άλλα πράγματα.

### Διαφορά μεταξύ CSS και HTML

Οι γλώσσες HTML και XHTML μπορεί να χρησιμοποιούνται για φτιάξουμε δομή στο περιεχόμενο και να ορίσουμε την διάταξη των αντικειμένων μιας ιστοσελίδας. Οι εντολές και τα φύλλα CSS προσφέρουν μορφοποίηση σε ήδη δομημένο περιεχόμενο, έχουν περισσότερες επιλογές εμφάνισης και καλύτερη ακρίβεια. Σήμερα τα CSS φύλλα υποστηρίζονται από όλα τα προγράμματα πλοήγησης.

Αρχικά η γλώσσα HTML χρησιμοποιήθηκε μόνο για να προσδώσει δομή στο κείμενο. Ο προγραμματιστής θα μπορούσε να βάλει σήμανση σ' ένα κείμενο του με την χρήση των HTML tags, όριζε τον τίτλο με <h1> και με <p> μια παράγραφο κλπ.

Αυτό οδήγησε σε μια κατάσταση όπου οι αρχικές ετικέτες δομής, χρησιμοποιήθηκαν περισσότερο στις ιστοσελίδες προκειμένου να προσδιοριστεί η εμφάνιση του ιστότοπου παρά για να προσδεθεί δομή στο κείμενο. Επίσης υπήρχαν πολλές περιπτώσεις που ετικέτες διάταξης δεν υποστηριζόντουσαν από όλα τα προγράμματα πλοήγησης.

Η γλώσσα CSS εφευρέθηκε για να διορθωθεί αυτή η κατάσταση, παρέχοντας στους σχεδιαστές ιστοσελίδων δυνατότητες οι οποίες να υποστηρίζονται από όλους τους browser. Ακόμη, ο διαχωρισμός της μορφής παρουσίασης των εγγράφων από το περιεχόμενο των εγγράφων, κάνει την συντήρηση του δικτυακού τύπου να είναι πολύ πιο εύκολη.

### Δυνατότητες της CSS

Η CSS ήταν μια επανάσταση στον κόσμο του σχεδιασμού ιστοσελίδων. Παραθέτουμε κάποιες από τις δυνατότητες που μας δίνονται μέσω της CSS Ένα μόνο φύλλο CSS μπορεί να προσφέρει μορφοποίησης σε πολλά HTML και XHTML έγγραφα. Κάνει τον έλεγχο της διάταξης μιας ιστοσελίδας με μεγαλύτερη ακρίβεια. Παρουσιάζουν διαφορετική διάταξη σε διαφορετικούς τύπους μέσων πχ σε οθόνη, σε εκτύπωση, κλπ. Χρησιμοποιούν πολλές νεότερες, προηγμένες και εξελιγμένες τεχνικές.

Στη συνέχεια θα δούμε πως μπορούμε να χρησιμοποιήσουμε την CSS ώστε να μορφοποιήσουμε την ιστοσελίδα μας, όπως για παράδειγμα να αλλάξουμε τα χρώματα το μέγεθος των γραμματοσειρών, το πλάτος και το χρώματα των γραμμών κ.α.

Για παράδειγμα, σε μια σελίδα html οι απαιτήσεις μορφοποίησης μπορεί να είναι ως εξής : να καθορίσετε ότι το φόντο της σελίδας είναι ένα 'κρεμ' χρώμα, το περιεχόμενο όλων των στοιχείων των διαφόρων παραγράφων (<p>) να εμφανίζεται με γκρι χρώμα και με γραμματοσειρά Arial, και όλοι οι τίτλοι πρώτου



επιπέδου (<h1>) θα πρέπει να είναι με κόκκινο χρώμα και γραμμένοι με την γραμματοσειρά Times New ROMAN.

Για να πετύχουμε τέτοιες απαιτήσεις θα πρέπει να ορίσουμε ανάλογους κανόνες κάθε φορά για κάθε στοιχείο που θέλουμε να μορφοποιήσουμε.

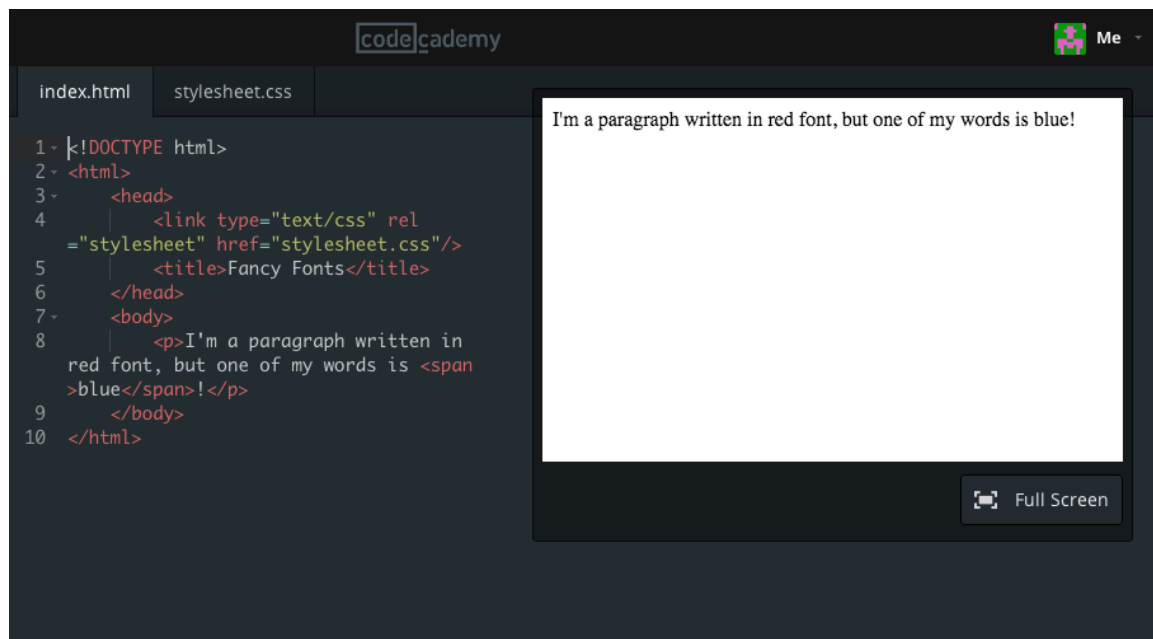
## Στοιχεία CSS

Για αυτό είναι απαραίτητο να γνωρίζουμε τα παρακάτω :

- Τι κάνει ένας κανόνας CSS
- Πώς τοποθετούμε CSS κανόνες μέσα σε μια ιστοσελίδα ή πώς συνδέουμε την ιστοσελίδα με ένα εξωτερικό έγγραφο CSS
- Πώς ελέγχονται οι ιδιότητες παρουσίασης και οι αντίστοιχες τιμές τους των διαφόρων στοιχείων
- Πώς ελέγχεται η παρουσίαση του κειμένου με τη χρήση CSS
- Πώς τα φύλλα CSS βασίζονται σε ένα μοντέλο κουτί, και πώς ορίζονται διαφορετικές ιδιότητες για αυτά τα κουτιά.

Στη συνέχεια θα δούμε κάποια βασικά παραδείγματα της CSS και των δυνατοτήτων που μας προσφέρει για την μορφοποίηση της ιστοσελίδα μας.

Στο πρώτο μάθημα CSS που κάναμε στο codecademy μας δίνεται μια παράγραφος η οποία μέσα στο κείμενο της έχει μια λέξη μέσα στα tags <span>. Σ αυτό το Project στόχος μας είναι να αλλάξουμε το χρώμα της της παραγράφου σε κόκκινο και της λέξης που βρίσκεται μέσα στα tags <span> σε μπλε.



```
codecademy Me
index.html stylesheet.css
1 - <!DOCTYPE html>
2 - <html>
3 -   <head>
4 -     <link type="text/css" rel
   = "stylesheet" href="stylesheet.css"/>
5 -     <title>Fancy Fonts</title>
6 -   </head>
7 -   <body>
8 -     <p>I'm a paragraph written in
   red font, but one of my words is <span
   >blue</span>!</p>
9 -   </body>
10 </html>
```

I'm a paragraph written in red font, but one of my words is blue!

Full Screen

εικ 6) Ο html κώδικας στο αριστερό μέρος της οθόνης και στα δεξιά μια προεπισκόπηση σε ένα εικονικό πρόγραμμα πλοήγησης

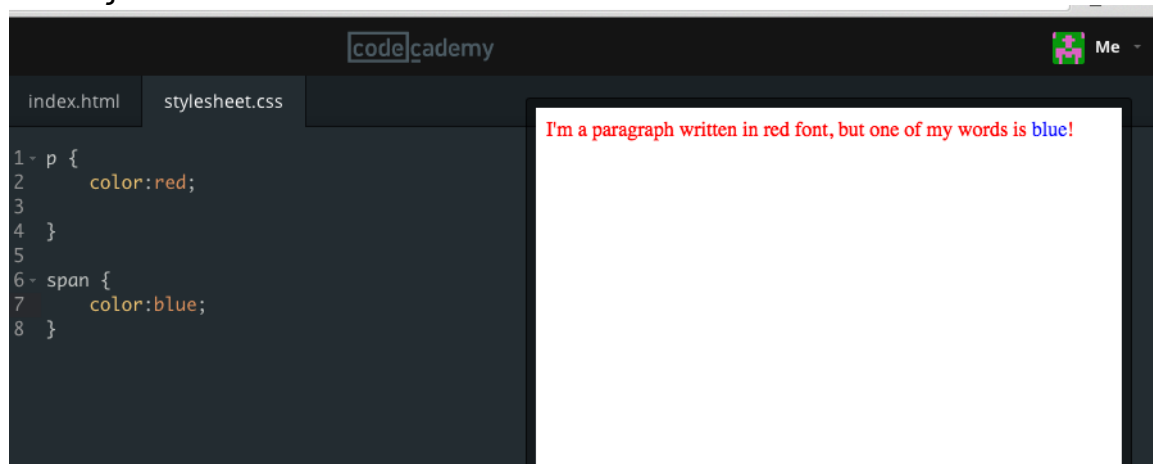
Το πρώτο και απαραίτητο βήμα που πρέπει να κάνουμε ώστε να διαμορφώσουμε την ιστοσελίδα μας με CSS είναι να συνδέσουμε με ένα link την html σελίδα με τον CSS κώδικα. Αυτό γίνεται βάζοντας αυτές τις δύο γραμμές κώδικα στο head της html σελίδας:

```
<link type="text/css" rel="stylesheet" href="stylesheet.css"/>
```

εικ 7) Το link που συνδέει την html σελίδα μας με την CSS σελίδα.

Για να συνδέσουμε τις δύο σελίδες τοποθετούμε ένα link στο head της html σελίδας το οποίο έχει τρία χαρακτηριστικά. Το πρώτο είναι ο τύπος type που πρέπει πάντα να ισούται ="text/css" το δεύτερο ένα rel το οποίο πρέπει πάντα να ισούται με ="style sheet" και τέλος το τρίτο χαρακτηριστικό είναι ένα href το οποίο πρέπει να δείχνει την διεύθυνση του CCS αρχείου, όπου αυτό στη περίπτωση μας είναι href="stylesheet.css"

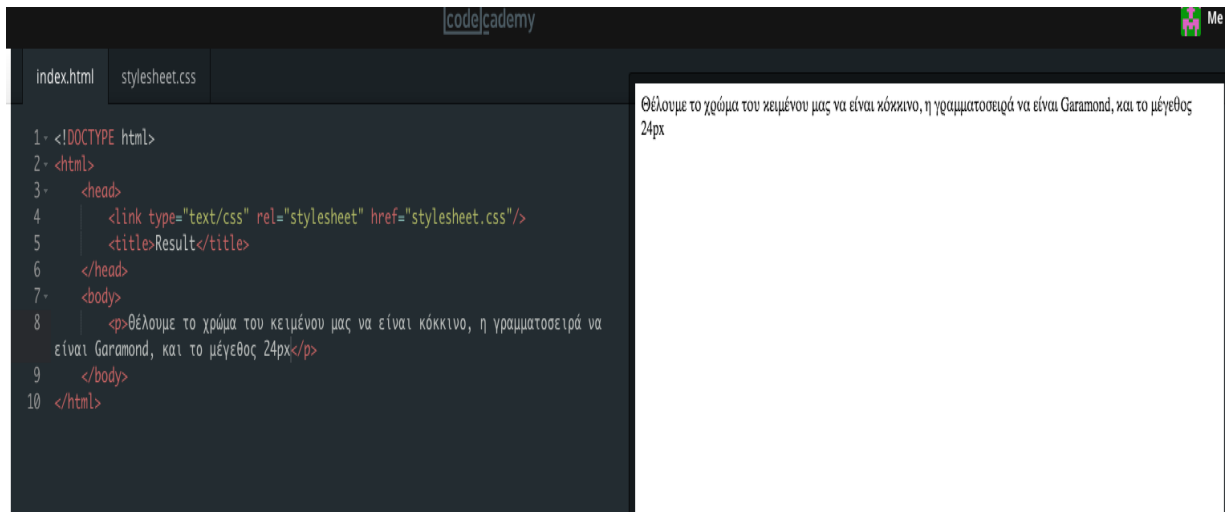
Επιστρέφοντας στο Project μας γράφουμε το CSS αρχείο και επιλέγουμε όπως μας ζητήθηκε όλοι οι παράγραφοι να είναι με κόκκινο χρώμα και η λέξη μέσα στα tags <span> να είναι μπλε. Αυτό γίνεται εφικτό με τις παρακάτω πολύ απλές εντολές.



εικ 8) Ο CSS κώδικας που χρειάστηκε για το επιθυμητό αποτέλεσμα στο δεξί μέρος της οθόνης

Αυτό γίνεται με τον εξής απλό τρόπο. Επιλέγουμε της παραγράφους (p) και μέσα στις αγκύλες επιλέγουμε το χρώμα που θέλουμε να έχουν, στο δικό μας παράδειγμα color: red; . Με τον ίδιο τρόπο επιλέγουμε τις λέξεις που βρίσκονται μέσα σε span tags και βάζουμε το χρώμα τους.

Στο επόμενο παράδειγμα που ακολουθεί μας ζητάτε να αλλάξουμε και το χρώμα του κειμένου και την γραμματοσειρά μας και το μέγεθος της γραμματοσειράς.

The image shows a code editor window with a dark theme. At the top, there are tabs for 'index.html' and 'stylesheet.css'. The 'index.html' tab is active, showing the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link type="text/css" rel="stylesheet" href="stylesheet.css"/>
5     <title>Result</title>
6   </head>
7   <body>
8     <p>Θέλουμε το χρώμα του κειμένου μας να είναι κόκκινο, η γραμματοσειρά να
είναι Garamond, και το μέγεθος 24px</p>
9   </body>
10 </html>
```

On the right side, there is a preview window showing the rendered HTML. It contains the text: 'Θέλουμε το χρώμα του κειμένου μας να είναι κόκκινο, η γραμματοσειρά να είναι Garamond, και το μέγεθος 24px'.

εικ 9) Η παράγραφος μας και η προεπισκόπηση της στο πρόγραμμα πλοήγησης

Για να γίνει αυτό εφικτό γράψαμε τον ακόλουθο κώδικα CSS.

The image shows a code editor window with a dark theme. It contains the following CSS code:

```
1 p {
2   font-family: Garamond;
3   color: red;
4   font-size: 24px;
5   |
6 }
```

εικ 10) Κώδικα CSS

Επιλέξαμε την παράγραφο του κειμένου μας και ανοίγοντας αγκύλες αλλάξαμε την γραμματοσειρά σε `font-family: Garamond;` το χρώμα της παραγράφου σε `color:red;` και το μέγεθος των γραμμάτων σε `font-size:24px;`

## Βάση δεδομένων και Queries

Τα site που απαιτούν από τους χρήστες να δημιουργήσουν προσωπικό προφίλ χρησιμοποιούν, όπως αναφέραμε και στο προηγούμενο κεφάλαιο, βάσεις δεδομένων για την αποθήκευση όλων αυτών των πληροφοριών. Οι πληροφορίες στις βάσεις δεδομένων αποθηκεύονται σε πίνακες. Για την δημιουργία του eclass για κινητά χρησιμοποιήσαμε MySql βάση δεδομένων. Στην εικόνα παρακάτω βλέπουμε την βάση δεδομένων που τρέχει πίσω από το eclass.

Δημιουργία προβλή [ Εσωτερικό ] [ Επεξεργασία ]

Αριθμός εγγραφών: 25

Ταξινόμηση ανά κλειδί: PRIMARY (Αύξουσα)

+ Επιλογές

	cours_id	code	language	Course	intitule	description	course_keywords	course_addon	course_license	visible	titulaires	fake_code
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	1	TMA100	greek		Εργαστήριο Εφαρμοσμένου Ηλεκτρομαγνητισμού		NULL	NULL	NULL	0	Κουριδάκης Στυλιανός Τζιανόλα Ειρήνη	TMA100
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	2	TMA101	greek		Εφαρμοσμένος Ηλεκτρομαγνητισμός		NULL	NULL	NULL	0	Πετρίδης Κωνσταντίνος	TMA101
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	3	TMA102	greek		Ψηφιακά Συστήματα Επικοινωνιών		NULL	NULL	NULL	0	Ευάγγελος Κάκκινος	TMA102
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	4	TMA103	greek		CAD και Κατασκευή		NULL	NULL	NULL	0	Χατζάκης Ιωάννης	TMA103
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	5	TMA104	greek		ΜΑΘΗΜΑΤΙΚΑ ΜΕ ΧΡΗΣΗ ΗΥ I		NULL	NULL	NULL	0	Πλάτης Δημήτριος	TMA104
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	6	TMA105	greek		Εφαρμογές των Μικροελεγκτών στις Τηλεπικοινωνίες...		NULL	NULL	NULL	0	Μπαρμπούνης Ιωάννης	TMA105
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	7	TMA106	greek		Ψηφιακά Συστήματα Επικοινωνιών (Εργ.)		NULL	NULL	NULL	0	Κάκκινος Ευάγγελος	TMA106
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	8	TMA107	greek		Ανάλυση & Σύνθεση Κυκλωμάτων με ΗΥ		NULL	NULL	NULL	0	Δρ. Αντώνιος Ι. Κωνσταντίνος, Επίκουρος Καθηγητής	TMA107
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	9	TMA108	greek		Αρχιτεκτονική ΗΥ (Εργαστήριο)		NULL	NULL	NULL	0	Παράσης Νικόλαος Ρηγάκης Ηρακλής	TMA108
<input type="checkbox"/> Επεξεργασία <input type="checkbox"/> Αντιγραφή <input type="checkbox"/> Διαγραφή	10	TMA109	greek		Αρχιτεκτονική ΗΥ		NULL	NULL	NULL	0	Πετράκης	TMA109

εικ 11) Η βάση δεδομένων του eclass.

<input type="checkbox"/> admin_announcements
<input type="checkbox"/> agenda
<input type="checkbox"/> annonces
<input type="checkbox"/> auth
<input type="checkbox"/> config
<input type="checkbox"/> cours
<input type="checkbox"/> course_review
<input type="checkbox"/> course_units
<input type="checkbox"/> course_user
<input type="checkbox"/> document
<input type="checkbox"/> ebook
<input type="checkbox"/> ebook_section
<input type="checkbox"/> ebook_subsection
<input type="checkbox"/> faculte
<input type="checkbox"/> forum_notify
<input type="checkbox"/> glossary
<input type="checkbox"/> glossary_category
<input type="checkbox"/> group
<input type="checkbox"/> group_members
<input type="checkbox"/> group_properties
<input type="checkbox"/> link
<input type="checkbox"/> link_category
<input type="checkbox"/> logout
<input type="checkbox"/> logout_summary
<input type="checkbox"/> login_failure
<input type="checkbox"/> monthly_summary
<input type="checkbox"/> parents_announcements
<input type="checkbox"/> pma_bookmark
<input type="checkbox"/> pma_column_comments
<input type="checkbox"/> pma_pdf_pages
<input type="checkbox"/> pma_relation

εικ 12) data tables

Στην παραπάνω εικόνα βλέπουμε τους πίνακες μέσα στην βάση δεδομένων μας. Αυτοί οι πίνακες περιέχουν όλες τις πληροφορίες του site μας. Τα μαθήματα τις λεπτομέρειές των μαθημάτων, τους φοιτητές, πληροφορίες για τους φοιτητές κ.α. Στην εικόνα 13 θα δούμε το εσωτερικό ενός πίνακα, πως φαίνονται δηλαδή οι αποθηκευμένες πληροφορίες σε ένα πίνακα.

[ Δημιουργία προφίλ ] [ Εσωτερικό ] [ Επεξεργασία ]

Αριθμός εγγραφών: 25

Ταξινόμηση ανά κλειδί: PRIMARY (Αόριστα)

+ Επιλογές

				cours_id	code	languageCourse	intitule	description	course_keywords	course_addon	course_license	visible	titulaires	fake_code
<input type="checkbox"/>				1	TMA100	greek	Εργαστήριο Εφαρμοσμένου Ηλεκτρομαγνητισμού	NULL		NULL		0	1 Κουρδός Στυλιανός Τζανιάκη Ειρήνη	TMA100
<input type="checkbox"/>				2	TMA101	greek	Εφαρμοσμένος Ηλεκτρομαγνητισμός	NULL	NULL	NULL		0	1 Πετρίδης Κωνσταντίνος	TMA101
<input type="checkbox"/>				3	TMA102	greek	Ψηφιακά Συστήματα Επικοινωνιών	NULL	NULL	NULL		0	1 Ευάγγελος Κάκκινος	TMA102
<input type="checkbox"/>				4	TMA103	greek	CAD και Κατασκευή	NULL	NULL	NULL		0	1 Χατζόκης Ιωάννης	TMA103
<input type="checkbox"/>				5	TMA104	greek	ΜΑΘΗΜΑΤΙΚΑ ΜΕ ΧΡΗΣΗ ΗΥ I	NULL	NULL	NULL		0	2 Πιλάκης Δημήτριος	TMA104
<input type="checkbox"/>				6	TMA105	greek	Εφαρμογές των Μικροϋπολογιστών στις Τηλεπικοινωνίες...	NULL	NULL	NULL		0	1 Μπαρμπουνάκης Ιωάννης	TMA105
<input type="checkbox"/>				7	TMA106	greek	Ψηφιακά Συστήματα Επικοινωνιών (Εργ.)	NULL	NULL	NULL		0	1 Κάκκινος Ευάγγελος	TMA106
<input type="checkbox"/>				8	TMA107	greek	Ανάλυση & Σύνθεση Κυκλωμάτων με ΗΥ	NULL	NULL	NULL		0	1 Δρ. Αντώνιος Ι. Κωνσταντάρης, Επίκουρος Καθηγητής	TMA107
<input type="checkbox"/>				9	TMA108	greek	Αρχιτεκτονική ΗΥ (Εργαστήριο)	NULL	NULL	NULL		0	1 Πετράκης Νικόλαος, Ρηγάκης Ηρακλής	TMA108
<input type="checkbox"/>				10	TMA109	greek	Αρχιτεκτονική ΗΥ	NULL	NULL	NULL		0	1 Πετράκης	TMA109

εικ 13) Οι αποθηκευμένες πληροφορίες μέσα σε ένα πίνακα της βάσης δεδομένων

Τα αιτήματα που γίνονται προς την βάση δεδομένων για συγκεκριμένες πληροφορίες ονομάζονται Queries. Πολλές φορές οι ερωτήσεις που κάνουμε προς τις βάσεις δεδομένων απαιτούν τον συνδυασμό πινάκων.

Για παράδειγμα στο eclass οι φοιτητές μπορούν να δουν σε ποια μαθήματα είναι εγγεγραμμένοι. Αυτό για να το δούνε χρειάζεται ο συνδυασμός πινάκων. Την ίδια δυνατότητα θα έχουν και οι φοιτητές που επισκέπτονται το eclass από το κινητό τους. Για να γίνει εφικτό αυτό το αίτημα-query που στείλαμε στην βάση δεδομένων είναι το εξής :

```
SELECT intitule, cours.cours_id FROM `cours` join cours_user on cours.cours_id = cours_user.cours_id where user_id = 2
```

Μέσα στη βάση δεδομένων υπάρχει ένας πίνακας με την ονομασία cours(εικ 13) ο οποίος έχει όλα τα μαθήματα της σχολής μας. Μέσα σ αυτό το πίνακα υπάρχει μια στήλη στην οποία έχουμε αποθήκευση τα μαθήματα με την ονομασία τους την intitule. Έχουμε δώσει επίσης και μια ταυτότητα σε κάθε μάθημα και αυτή η ταυτότητα βρίσκεται στη στήλη cours\_id. Αφού πάρουμε αυτές τις πληροφορίες από αυτόν το πίνακα του ζητάμε να τις συνδυάσει με τις πληροφορίες που θα πάρουμε από ένα άλλο πίνακα με την ονομασία cours\_user(εικ 14) όπου εκεί έχουμε αποθήκευση τις ταυτότητες (id) των χρηστών αλλά και τις ταυτότητες των μαθημάτων ( cours\_id) και φαίνεται ποιοι χρήστες παρακολουθούν το κάθε μάθημα.

		cours_id	user_id	statut	team	tutor	editor	reviewer	reg_date	receive_mail
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	1	1	1	0	1	0	0	2014-07-19	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	5	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	2	1	1	0	1	0	0	2014-07-19	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	3	1	1	0	1	0	0	2014-07-19	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	4	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	4	2	5	0	1	0	0	2014-09-27	0
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	6	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	7	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	8	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	9	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	10	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	11	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	12	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	13	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	14	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	15	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	16	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	17	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	18	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	19	1	1	0	1	0	0	2014-09-16	1
<input type="checkbox"/>	Επεξεργασία  Αντιγραφή  Διαγραφή	20	1	1	0	1	0	0	2014-09-16	1

εικ 14) Ο cours\_user πίνακας.

Μέσα από αυτόν τον συνδυασμό πινάκων ζητάμε στην βάση να μας επιστρέψει σαν αποτέλεσμα τα μαθήματα που παρακολουθεί ο χρήστης με cours\_id= 2. Βλέπουμε πως αν τρέξουμε αυτό το αίτημα η βάση δεδομένων μας δίνει την επιθυμητή απάντηση στην παρακάτω εικόνα.

```
SELECT intitule,cours.cours_id FROM `cours` join cours_user on cours.cours_id = cours_user.cours_id where user_id = 2
```

Αριθμός εγγραφών: 25

πιλογές	intitule	cours_id
Επεξεργασία  Αντιγραφή  Διαγραφή	Εργαστήριο Εφαρμοσμένου Ηλεκτρομαγνητισμού	1
Επεξεργασία  Αντιγραφή  Διαγραφή	Εφαρμοσμένος Ηλεκτρομαγνητισμός	2
Επεξεργασία  Αντιγραφή  Διαγραφή	Ψηφιακά Συστήματα Επικοινωνιών	3
Επεξεργασία  Αντιγραφή  Διαγραφή	CAD και Κατασκευή	4

Επιλογή όλων    Με τους επιλεγμένους: Αλλαγή Διαγραφή Εξαγωγή

Αριθμός εγγραφών: 25

εικ 15) Τα μαθήματα που παρακολουθεί ο χρήστης με user\_id= 2

Τα queries πολλές φορές μπορεί να είναι μια δύσκολη και σύνθετη διαδικασία. Ο μόνος τρόπος να κάνουμε πιο εύκολη αυτή τη διαδικασία είναι κατά την δημιουργία της βάσης δεδομένων. Ο προγραμματιστής καλείται να έχει στο νου

του συνέχεια ότι μια βάση δεδομένων πρέπει να είναι καλά οργανωμένη και να είναι εύκολη η πρόσβαση σε βασικές πληροφορίες.

Η html η CSS και η C# είναι οι τρεις γλώσσες που χρειαστήκαμε για την δημιουργία του eclass για κινητά. Το eclass το δημιουργήσαμε στο περιβάλλον του Visual Studio και χρησιμοποιήσαμε και κάποια άλλα βασικά εργαλεία. Στο παρακάτω κεφάλαιο θα δούμε τα εργαλεία που χρησιμοποιήθηκαν και το περιβάλλον στο οποίο δημιουργήθηκε το eclass.

# JavaScript

Μια άλλη γλώσσα που χρησιμοποιήσαμε για την δημιουργία του mobile eclass είναι η JavaScript.

Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού των ηλεκτρονικών υπολογιστών. Χρησιμοποιείται συχνά για την δημιουργία δια δραστικών ιστοσελίδων. Χρησιμοποιείται επίσης για την δημιουργία παιχνιδιών αλλά και για την δημιουργία εφαρμογών της επιφάνειας εργασίας και κινητών.

Η JavaScript χαρακτηρίζεται ως μια πρωτότυπη γλώσσα σεναρίου (scripting) με δυναμικούς τύπους και πολλά χαρακτηριστικά από τις γλώσσες πρώτης κατηγορίας. Αυτά τα χαρακτηριστικά της δίνουν την δυνατότητα να υποστηρίζει διάφορα στυλ προγραμματισμού όπως αντικειμενοστραφή(object-oriented), επιτακτικό(imperative), και λειτουργικό(functional).

Εκτός από τις ομοιότητες στην ονομασία, στην σύνταξη και σε κάποιες βασικές βιβλιοθήκες, η JavaScript έχει πολύ διαφορετική σημασιολογία με την Java και είναι δύο εντελώς διαφορετικές γλώσσες. Στην πραγματικότητα ή σύνταξη της JavaScript προέρχεται από την C.

Η JavaScript είναι μια ερμηνευόμενη γλώσσα ( interpreted language), δηλαδή τα σενάρια (scripts) εκτελούνται χωρίς να προηγηθεί μεταγλώττιση του κώδικα. Η JavaScript υποστηρίζεται από όλα τα δημοφιλή προγράμματα πλοήγησης, όπως ο Internet Explorer, ο Google Chrome, ο Firefox κ.λπ. .

## Τρόποι εισαγωγής της JavaScript

Υπάρχουν δύο τρόποι να εισάγουμε τον κώδικα JavaScript μέσα σε ένα Html αρχείο, εσωτερικά (Internal) και εξωτερικά (External).

### 1.Εσωτερικά

Εσωτερικά του html αρχείου υπάρχουν δύο θέσεις που μπορούμε να γράψουμε τον JavaScript κώδικα.

Η μια θέση είναι μέσα στο σώμα του html αρχείου δηλαδή στο <body>js Code</body>. Η άλλη θέση είναι μέσα στο κεφάλι του html αρχείου. <head> js Code </head>.

### 2. Εξωτερικά

Ο κώδικας JavaScript μπορεί να γραφτεί και ξεχωριστά από html αρχείο. Συνήθως αυτά τα αρχεία περιέχουν κώδικα που μπορεί να χρησιμοποιηθεί σε διαφορετικές σελίδες. Τα εξωτερικά αρχεία JavaScript έχουν κατάληξη .js .



## Κεφάλαιο 3

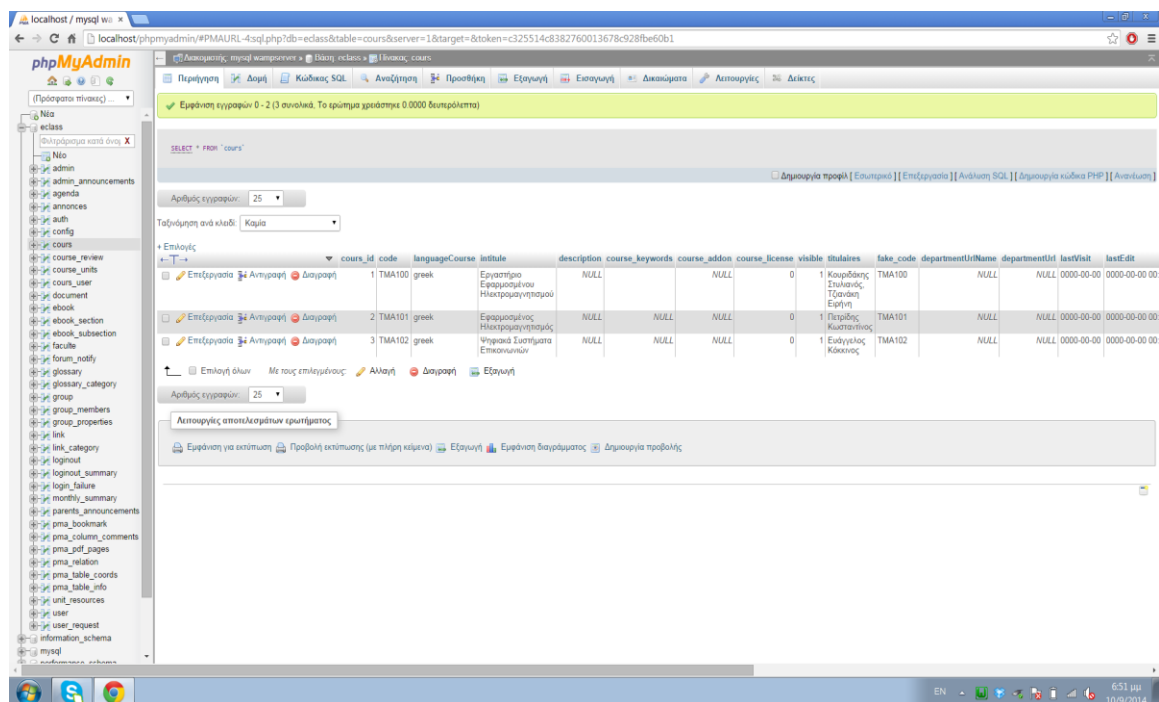
### Προγραμματιστικά Εργαλεία

## Εργαλεία που χρησιμοποιήθηκαν για την δημιουργία του eclass

### MySQL/phpMyAdmin

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Το πρόγραμμα τρέχει έναν εξυπηρετητή παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, το Google, το Facebook και το Twitter.

Για την διαχείριση της MySQL μέσω ενός προγράμματος πλοήγησης χρησιμοποιήσαμε το phpMyAdmin ένα εργαλείο που διατίθεται δωρεάν στο ίντερνετ. Με το phpMyAdmin μπορούμε να κάνουμε διάφορες εργασίες όπως, δημιουργία τροποποίηση ή διαγραφή δεδομένων από τη βάση, δημιουργία πινάκων πεδίων ή σειρών. Διαχείριση των χρηστών και των δικαιωμάτων τους.



The screenshot shows the phpMyAdmin web interface. The left sidebar contains a tree view of the database structure, including tables like 'course', 'course\_review', 'course\_units', etc. The main area displays a table with the following data:

Επιλογές	course_id	code	language	Course	title	description	course_keywords	course_license	course_license	visible	titulaires	fake_code	department	litName	department	lastVisit	lastEdit
<input type="checkbox"/>	1	TMA100	greek		Εργαστήριο Εφαρμοσμένα Ηλεκτρονικά			NULL	NULL	0	1	Κουρδής	TMA100	NULL	NULL	0000-00-00	0000-00-00 001
<input type="checkbox"/>	2	TMA101	greek		Εφαρμοσμένος Ηλεκτρονικός			NULL	NULL	0	1	Παπαδόπουλος	TMA101	NULL	NULL	0000-00-00	0000-00-00 001
<input type="checkbox"/>	3	TMA102	greek		Ψηφιακό Σύστημα Επικοινωνιών			NULL	NULL	0	1	Ευαγγελικός Κόκορας	TMA102	NULL	NULL	0000-00-00	0000-00-00 001

εικ 6) Η βάση δεδομένων όπως φαίνεται στο πρόγραμμα πλοήγησης μέσω του phpMyAdmin

## Apache HTTP

Ο apache είναι ένας εξυπηρετητής του διαδικτύου. Όταν ένας χρήστης επισκέπτεται μια ιστοσελίδα το πρόγραμμα πλοήγησης επικοινωνεί με ένα διακομιστή ο οποίος παράγει την επιθυμητή ιστοσελίδα και τις αποστέλλει πίσω στο πρόγραμμα πλοήγησης. Ένας από τους λόγους που οι προγραμματιστές χρησιμοποιούν συχνά τον apache είναι γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, Mac OS, Linux και Unix. Επίσης χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης βάσης δεδομένων όπως MySql και Oracle.

## Visual Studio

Το Visual Studio είναι μια ολοκληρωμένη συλλογή εργαλείων και υπηρεσιών (Integrated Development Environment) που προσφέρει η Microsoft για την ανάπτυξη εφαρμογών της επιφάνειας εργασίας, καθώς και δικτυακών ιστότοπων, δικτυακών εφαρμογών και δικτυακών υπηρεσιών. Το Visual Studio υποστηρίζει διάφορες γλώσσες προγραμματισμού όπως C/C++, VB.NET ( Visual basic.net), C# και F#. Για δικτυακές εφαρμογές και ιστότοπους υποστηρίζει XML/XSLT, HTML/XHTML, JavaScript και CSS.

### Εκδόσεις του Visual Studio

Υπάρχουν πολλές εκδόσεις του Visual Studio. Για την υλοποίηση του eclass σε mobile form χρησιμοποίησα το Visual Studio Ultimate 2013.

### Visual Studio Express

Οι εκδόσεις Visual Studio Express είναι ένα σύνολο ελαφρών ανεξάρτητων IDEs που παρέχονται σαν απογυμνωμένες εκδόσεις του Visual Studio δωρεάν, ανά πλατφόρμα και ανά γλώσσα. Πιο συγκεκριμένα οι εκδόσεις περιλαμβάνουν τα εργαλεία ανάπτυξης για τις αντίστοιχες πλατφόρμες (web, windows, phone) ή για τις αντίστοιχες γλώσσες (VB, C#). Το Visual Studio Express περιλαμβάνει μόνο ένα υποσύνολο των εργαλείων που παρέχονται στις άλλες εκδόσεις. Επίσης δεν υποστηρίζει plug-ins. Οι μεταγλωττιστές για 64-bit αρχιτεκτονικές δεν συμπεριλαμβάνονται στις εκδόσεις Express, ωστόσο μπορούν να εγκατασταθούν ξεχωριστά. Οι εκδόσεις Express στοχεύουν στην εκπαίδευση και στους χομπίστες. Τέλος οι Express εκδόσεις δεν χρησιμοποιούν την πλήρη MSDN library αλλά την MSDN Essentials Library.

### Visual Studio LightSwitch

Το Visual Studio LightSwitch είναι ένα απλοποιημένο IDE που επιτρέπει τη δημιουργία εταιρικών εφαρμογών γρήγορα και εύκολα για το desktop ή το cloud. Οι εφαρμογές που παράγονται είναι αρχιτεκτονικά 3 επιπέδων : η διεπαφή χρήστη τρέχει σε Microsoft Silver light, η λογική και το επίπεδο πρόσβασης βασίζεται σε WCF RIA Services και το Entity Framework ενώ φιλοξενείται στην ASP.NET. Επίσης υποστηρίζει και τον SQL Server αλλά και το Microsoft SQL

Azure για την αποθήκευση δεδομένων. Περιλαμβάνει περιβάλλοντα σχεδίασης για το σχεδιασμό οντοτήτων και συσχετίσεων αλλά και γραφικών διεπαφών. Η λογική της εφαρμογής μπορεί να γραφτεί σε Visual Basic ή C#. Η έκδοση αυτή μπορεί να εγκατασταθεί αυτόματα ή μέσα στο Visual Studio Professional.

#### Visual Studio Professional

Το Visual Studio Professional παρέχει ένα IDE για όλες τις υποστηριζόμενες γλώσσες στο .NET. Απλουστεύει τη δημιουργία, την αποσφαλμάτωση και την εγκατάσταση εφαρμογών σε πολλαπλές πλατφόρμες, που συμπεριλαμβάνουν το SharePoint και το Cloud. Η υποστήριξη στο MSDN διατίθεται στην έκδοση Essentials ή στην πλήρη έκδοση, ανάλογα με την άδεια που αγοράστηκε. Υποστηρίζει την επεξεργασία XML και XSLT και μπορεί να δημιουργήσει πακέτα εγκατάστασης που χρησιμοποιούν τις τεχνολογίες Click Once και MSI. Συμπεριλαμβάνει εργαλεία όπως τον Server Explorer για την ενοποίηση με τον SQL Server.

#### Visual Studio Premium

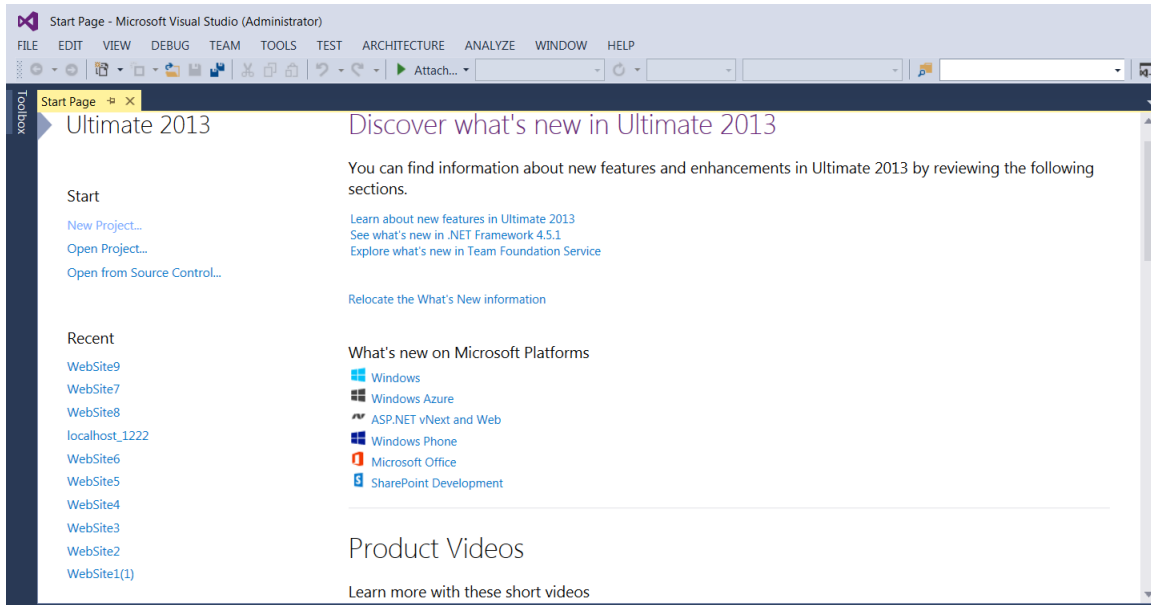
Η έκδοση Premium είναι ένα πλήρες σύνολο εργαλείων που απλουστεύει την ανάπτυξη εφαρμογών ατομικά αλλά και στα πλαίσια ομάδων ανάπτυξης. Ουσιαστικά περιλαμβάνει όλα τα εργαλεία της έκδοσης Professional αλλά και εργαλεία για την ανάλυση του κώδικα, εργαλεία για profiling αλλά και για database unit testing.

#### Visual Studio Ultimate

Η έκδοση Ultimate αποτελεί μετονομασία της έκδοσης Team Suite. Η έκδοση αυτή παρέχει ένα σύνολο εργαλείων για την υποστήριξη της συνεργατικής ανάπτυξης έργων στο .NET. Επιπρόσθετα των χαρακτηριστικών του Visual Studio Premium, συμπεριλαμβάνονται και εργαλεία για την ανάπτυξη βάσεων δεδομένων, την ανάλυση του κώδικα, την αρχιτεκτονική μιας εφαρμογής, τη δοκιμή αλλά και εργαλεία reporting.

#### Το περιβάλλον του Visual Studio Ultimate 2013

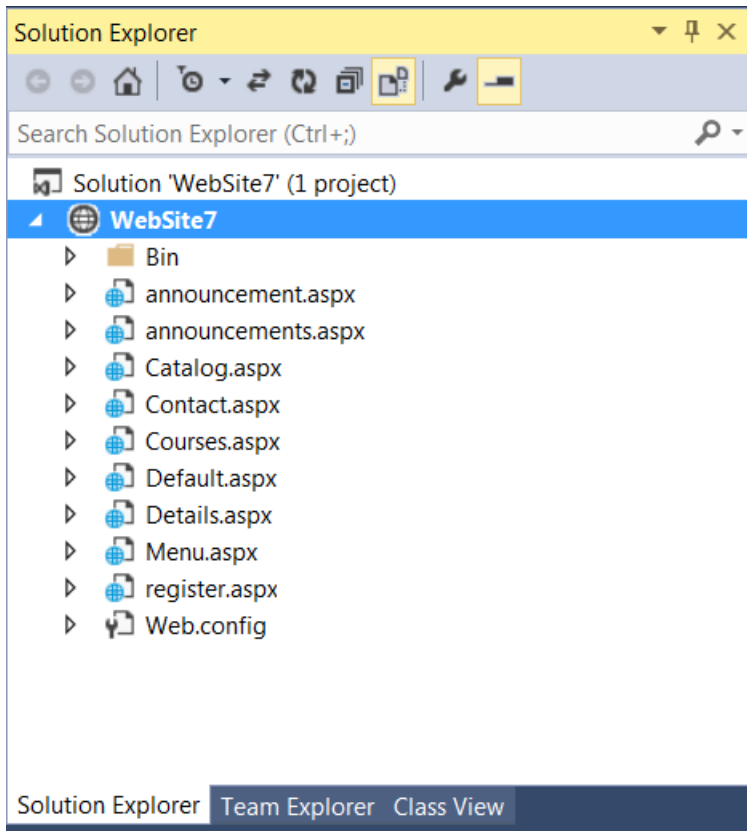
Προεπιλεγμένα όταν ξεκινάμε το Visual Studio εμφανίζεται μια σελίδα γνωστή ως Start Page. Σε αυτήν εμφανίζονται μηνύματα καλωσορίσματος, βίντεο με οδηγούς και πληροφορίες όπως τα project στα οποία δουλεύαμε τελευταία αλλά και μια επιλογή για τη λήψη των τελευταίων νέων μέσα από RSS Feeds.



εικ 7) Η αρχική σε σελίδα του Visual Studio και οι επιλογές που μας δίνονται

## Ο Solution Explorer

Όταν δημιουργούμε μια νέα εφαρμογή, ανοίγουμε μια ήδη υπάρχουσα ή ακόμα και ένα μόνο αρχείο, το Visual Studio 2013 χρησιμοποιεί την έννοια του solution για να οργανώσει τα πάντα. Συνήθως ένα solution αποτελείται από ένα ή περισσότερα projects, καθένα από τα οποία περιλαμβάνει πολλαπλά άλλα αρχεία. Το παράθυρο εργαλείου του Solution Explorer παρέχει μια βολική οπτική αναπαράσταση του solution, των Project και των αρχείων, όπως αυτή της εικόνας παρακάτω. Κάθε project συνοδεύεται από ένα εικονίδιο που συνήθως υποδεικνύει το είδος του project. Στην δική μας περίπτωση αυτό το Project είναι μια ιστοσελίδα.



εικ 8 ) Το παράθυρο του solution explorer

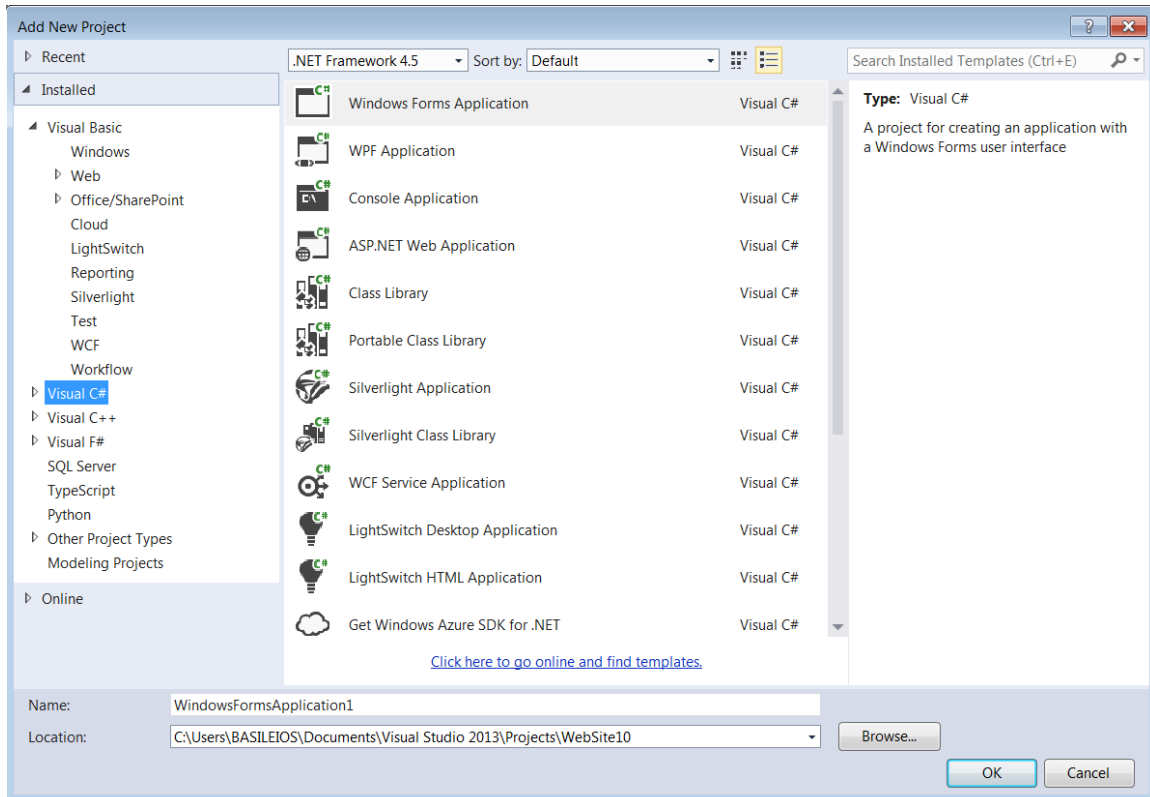
Ένας κόμβος ξεχωρίζει, γιατί το front του είναι bold. Αυτό υποδεικνύει ότι είναι η αρχική μας σελίδα, δηλαδή αυτό που θα τρέξει αν επιλέξουμε debug -> start Debugging ή πιέσουμε το F5. Για να αλλάξουμε την αρχική σελίδα, κάνουμε δεξί κλικ στη σελίδα που θέλουμε να θέσουμε σαν αρχική μας σελίδα και επιλέγουμε set as start page.

Η γραμμή εργαλείων στην κορυφή του Solution Explorer επιτρέπει την προσαρμογή της εμφάνισης των περιεχομένων του παραθύρου και δίνει συντομεύσεις στις διάφορες όψεις των περιεχομένων. Για παράδειγμα το κουμπί nest related files, επεκτείνει το solution εμφανίζοντας επιπρόσθετα αρχεία και φακέλους που δεν εμφανίζονται προεπιλεγμένα. Ένα πράγμα που πρέπει να θυμόμαστε είναι ότι αν αλλάξει κάτι από το file system, οι αλλαγές δεν θα φαίνονται άμεσα στον Solution Explorer. Το κουμπί Refresh ανανεώνει τα αρχεία που φαίνονται έτσι ώστε να είμαστε σίγουροι ότι βλέπουμε τα σωστά αρχεία.

Εκτός από έναν βολικό τρόπο εμφάνισης των Projects και των διαφόρων items ο Solution Explorer έχει ένα δυναμικό μενού που προσαρμόζεται ανάλογα με το είδος του item που επιλέγεται για να μας δώσει την πρόσβαση στις πιο συχνές εργασίες, όπως το να κάνουμε build to solution ή ένα μεμονωμένο Project, και να ανοίγουμε αρχεία.

## Προσθήκη νέων Project και Items

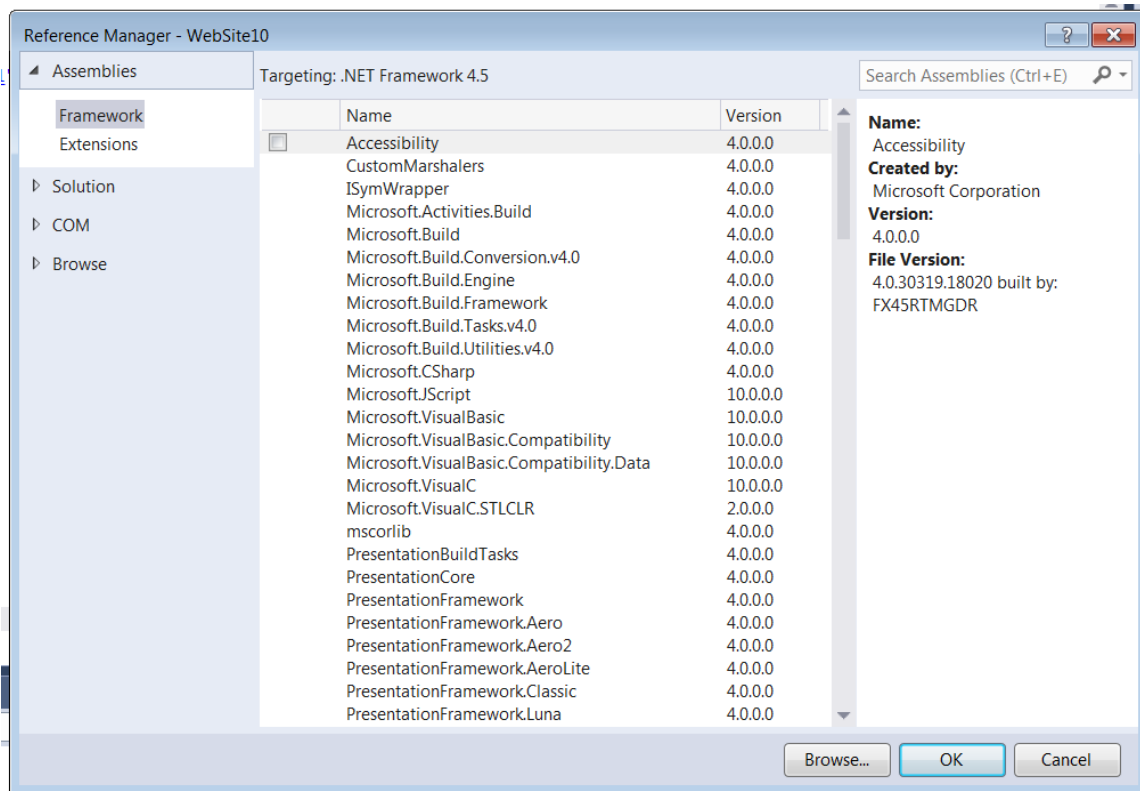
Μια από τις πιο συνηθισμένες ενέργειες που γίνονται με τον Solution Explorer είναι η προσθήκη, η αφαίρεση και η μετονομασία των Projects και των items. Για την προσθήκη ενός νέου Project σε μια υπάρχουσα solution επιλέγουμε Add->New Project από το μενού που εμφανίζεται με δεξί κλικ στον κόμβο Solution. Αυτό εμφανίζει το παρακάτω παράθυρο διαλόγου. Οι επιλογές μπορούν να ταξινομηθούν ως προς το όνομα. Ο χώρος στα δεξιά εμφανίζει πληροφορίες για το επιλεγμένο Project, όπως τον τύπο του και την περιγραφή του.



εικ 9) Το παράθυρο επιλογών για τη δημιουργία ενός νέου Project

## Προσθήκη αναφορών

Ένας από τους τρόπους με τους οποίους το Visual Studio υποστηρίζει την επαναχρησιμοποίηση των components είναι με τον μηχανισμό της προσθήκης αναφορών. Μια αναφορά, ουσιαστικά επιτρέπει στον compiler να αναγνωρίσει τύπους δεδομένων, ιδιότητες, πεδία και μεθόδους έτσι ώστε να γίνει υλοποιηθεί το Project μας. Αν χρειαστεί να χρησιμοποιήσουμε μια κλάση ενός τρίτου παρόχου, ή ακόμα και ένα δικό μας .NET assembly θα χρειαστεί να προσθέσουμε μια αναφορά προς αυτό. Αυτό γίνεται από το μενού που εμφανίζεται με δεξί κλικ σε κάποιο κόμβο Project και την επιλογή add -> Reference. Η επιλογή αυτή εμφανίζει το παρακάτω παράθυρο.



εικ 10) Το παράθυρο που μας εμφανίζεται για την προσθήκη μια αναφοράς

Κατά το άνοιγμα του παραθύρου αυτού, το Visual Studio αναζητά στον τοπικό υπολογιστή, την Global Assembly Cache και το ανοιχτό solution και κατασκευάζει μια λίστα με τις γνωστές βιβλιοθήκες που μπορούν να προστεθούν. Αυτές περιλαμβάνουν και τις αναφορές του COM αλλά και των πρόσφατα χρησιμοποιημένων αναφορών. Αν το component που θέλουμε να χρησιμοποιήσουμε δεν βρίσκεται σε κάποια από αυτές τις λίστες τότε μπορούμε να το εντοπίσουμε μέσω του Browse tab κατευθείαν από το file system.

Είναι δυνατό να προσθέσουμε αναφορά σε κανονικά Project μέσα στο τρέχον solution, αντί να αναφερθούμε στα μεταγλωττισμένα binary components. Μια τέτοια προσθήκη έχει το πλεονέκτημα ότι μπορούμε να κάνουμε ευκολότερα debug και σιγουρευόμαστε ότι χρησιμοποιούμε την τελευταία έκδοση όλων των components.

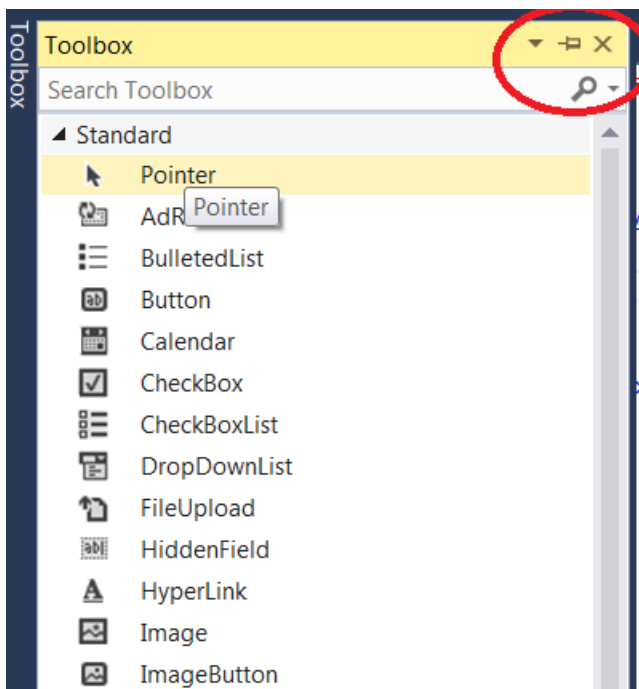
#### Τα παράθυρα εργαλείων

Πριν ξεκινήσει κανείς τη δημιουργία μιας εφαρμογής, είναι σημαντικό να εξοικειωθεί με τα αποτελούμενα μέρη του Visual Studio. Τα μενού και οι γραμμές εργαλείων είναι τοποθετημένες στο πάνω μέρος του παραθύρου και ένα σύνολο από μικρότερα παράθυρα ή χώρους εμφανίζεται στα αριστερά, δεξιά και κάτω από την κύρια περιοχή του παραθύρου. Στο κέντρο υπάρχει ο κύριος χώρος για τον επεξεργαστή κειμένου : όποτε ανοίγουμε ένα αρχείο πηγαίου κώδικα, ένα έγγραφο XML, μια φόρμα ή οποιοδήποτε άλλο αρχείο, θα εμφανίζεται σε αυτόν τον χώρο για επεξεργασία. Με κάθε νέο αρχείο που ανοίγει, δημιουργείται ένα



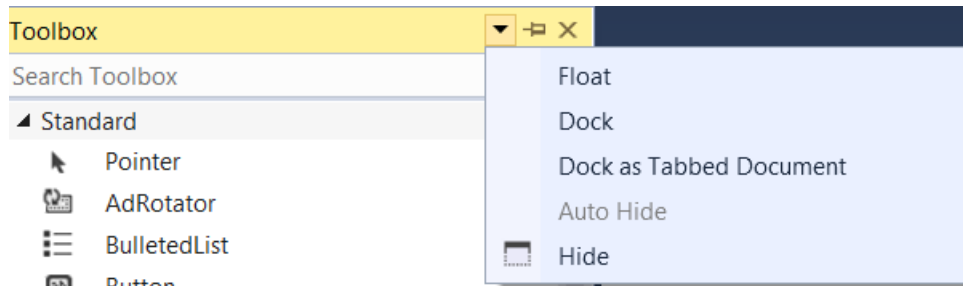
νέο tab για να μπορούμε να επιλέγουμε ποιο θα δούμε μεταξύ των ανοιχτών αρχείων.

Σε κάθε πλευρά του χώρου για τον επεξεργαστή κειμένου υπάρχει ένα σύνολο παραθύρων που αντιστοιχούν σε εργαλεία. Αυτά τα παράθυρα παρέχουν επιπρόσθετες πληροφορίες ανάλογα με την λειτουργικότητα που τους έχει προσδοθεί. Η προεπιλεγμένη τοποθέτηση των παραθύρων περιλαμβάνει τον Solution Explorer στα δεξιά, ενώ στα αριστερά περιλαμβάνει το Toolbox. Τα εργαλεία στα αριστερά εμφανίζονται στην συρτημένη μορφή τους. Αν επιλέξουμε τον τίτλο ενός παραθύρου, εμφανίζεται σε εκτεταμένη μορφή, αλλά κλείνει ξανά αν μετακινήσουμε τον κέρσορα αλλού. Όταν ένα παράθυρο είναι στην εκτεταμένη μορφή του, εμφανίζονται τρία εικονίδια όπως φαίνονται στην παρακάτω εικόνα :



εικ 11) Οι γενικές επιλογές ενός παραθύρου εργαλείων

Αν θέλουμε το παράθυρο να παραμείνει στην εκτεταμένη μορφή του γνωστή και ως *ripped*, πατάμε το μεσαίο κουμπί που μοιάζει με καρφίτσα. Η καρφίτσα περιστρέφεται κατά 90 μοίρες για να δείξει ότι το παράθυρο είναι καρφισωμένο. Το πάτημα του τρίτου κουμπιού κλείνει το παράθυρο. Αν αργότερα θελήσουμε να ξαναεμφανίσουμε το παράθυρο αυτό μπορούμε να το επιλέξουμε από το μενού View.



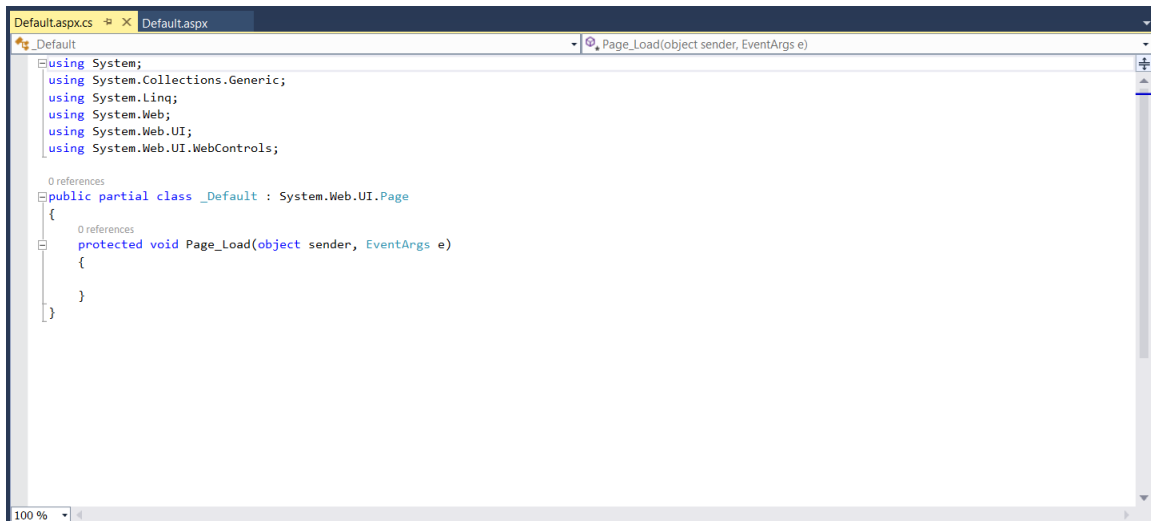
εικ 12) Λίστα με τους διαφορετικούς τρόπου τοθέτησης ενός παραθύρου εργαλείων

Αν επιλέξουμε το πρώτο εικονίδιο (το κάτω βελάκι), εμφανίζει μια λίστα που αναπαριστά έναν διαφορετικό τρόπο τοποθέτησης του παραθύρου. Η επιλογή Float επιτρέπει την τοποθέτηση του παραθύρου οπουδήποτε στην οθόνη ανεξαρτήτως του κυρίου παραθύρου. Κάτι τέτοιο είναι χρήσιμο όταν έχουμε πολλαπλές οθόνες, γιατί μπορούμε να μεταφέρουμε τα εργαλεία στη δεύτερη οθόνη και να αφήσουμε την πρώτη μόνο για το κύριο παράθυρο, μεγιστοποιώντας έτσι το διαθέσιμο χώρο. Η επιλογή Dock as Tabbed Document τοποθετεί το παράθυρο σε ένα επιπρόσθετο tab στον κεντρικό χώρο. Τέλος η επιλογή Hide, όπως δηλώνει το όνομα της, αποκρύπτει το παράθυρο.

#### Ο επεξεργαστής κειμένου

Όπως τα περισσότερα IDEs, το Visual Studio έχει στο επίκεντρο του ένα παράθυρο για την επεξεργασία κώδικα. Ο επεξεργαστής αυτός έχει εξελιχτεί μέσα στα χρόνια και είναι πολλά περισσότερα από ένας απλός επεξεργαστής κειμένου. Παρόλο που υπάρχουν διάφοροι designers για την πραγματοποίηση εργασιών όπως τη δημιουργία φορμών, την αλλαγή των ρυθμίσεων των Projects και την επεξεργασία των διαφόρων πόρων των εφαρμογών, οι περισσότεροι προγραμματιστές ξοδεύουν τον περισσότερο χρόνο τους στη γραφή κώδικα με τον επεξεργαστή αυτό. Για το λόγο αυτό αξίζει να ρίξουμε μια ματιά στο πώς να ρυθμίζουμε την εμφάνιση του για να δουλεύουμε πιο αποδοτικά με αυτόν.

Όταν ανοίγουμε ένα αρχείο κώδικα για επεξεργασία, δουλεύουμε με το παράθυρο επεξεργασίας κώδικα (Code editor window). Το παράθυρο αυτό φαίνεται ως εξής :



εικ 13) Το Code editor window

Πάνω από το χώρο του κώδικα υπάρχουν δύο λίστες drop-down που βοηθούν στην περιήγηση σε μεγάλα αρχεία κώδικα. Η πρώτη drop-down λίστα εμφανίζει τις κλάσεις στο τρέχον αρχείο κώδικα και η δεύτερη drop-down λίστα εμφανίζει τα μέλη της επιλεγμένης κλάσης από την πρώτη λίστα. Εμφανίζονται σε αλφαβητική σειρά για να είναι πιο εύκολο να βρούμε μια μέθοδο ή μέλος μέσα στο αρχείο.

### Περιοχές

Η αποτελεσματική σχεδίαση κλάσεων συνήθως έχει σαν αποτέλεσμα τη δημιουργία κλάσεων που υπηρετούν έναν μόνο σκοπό και δεν είναι ιδιαίτερα πολύπλοκες ή μακροσκελείς. Υπάρχουν όμως περιπτώσεις που αναγκαστικά μια κλάση πρέπει να είναι μεγάλη και κατά συνέπεια δύσχρηστη. Σε τέτοιες περιπτώσεις είναι καλό να σπάμε των κώδικα σε πολλαπλά αρχεία ή να χρησιμοποιούμε περιοχές κώδικα(regions), κάνοντας έτσι ευκολότερη την περιήγηση στον κώδικα.

Η εισαγωγή των μερικών κλάσεων, όπου ο ορισμός μια κλάσης μπορεί να διαιρεθεί σε πολλαπλά αρχεία, σημαίνει ότι κατά τη σχεδίαση μια κλάσης μπορούμε να τοποθετήσουμε κώδικα σε διαφορετικά αρχεία αναπαριστώντας έτσι μια λογική κλάση. Το πλεονέκτημα χρήσης διαφορετικών αρχείων είναι ότι μπορούμε να ομαδοποιήσουμε όλες τις μεθόδους που έχουν κάποια σχέση μεταξύ τους, όπως για παράδειγμα όταν υλοποιούμε μια διασύνδεση. Το πρόβλημα που δημιουργείται με αυτήν την προσέγγιση όμως είναι ότι πρέπει να διαχειριζόμαστε πολλά αρχεία και να εναλλάσσουμε συνεχώς μεταξύ τους.

### Outlining

Το Visual Studio έχει επίσης τη δυνατότητα αυτόματης σύμπτυξης και εμφάνισης των τμημάτων ενός αρχείου κώδικα, όπως μεθόδων, σχολίων και κλάσεων. Μπορούμε να χειριστούμε το outlining μέσα από το μενού Edit -> Outlining, το οποίο παρέχει ένα σύνολο εντολών για την απόκρυψη/εμφάνιση των τμημάτων κώδικα.

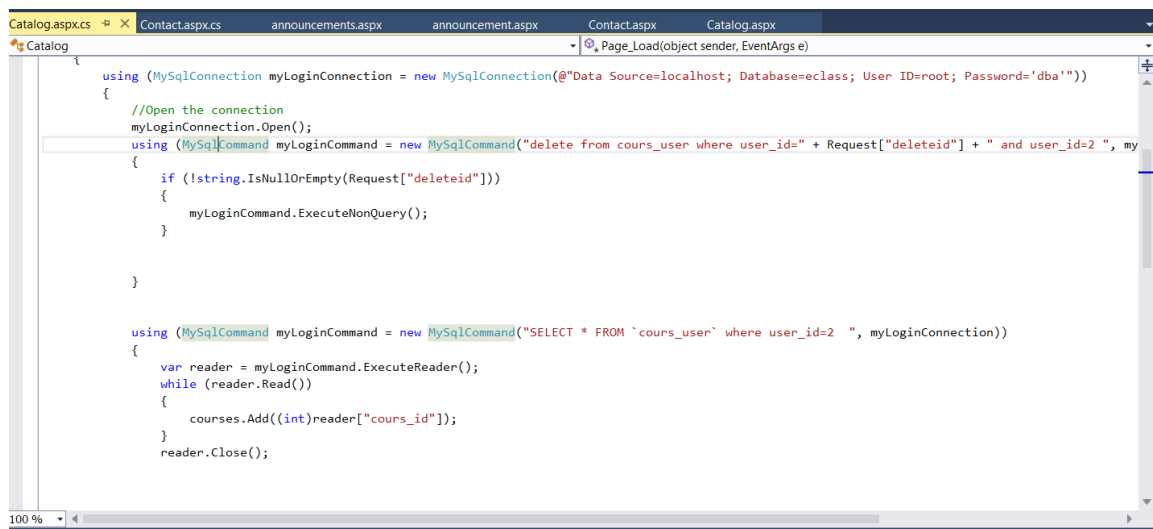
## Μορφοποίηση κώδικα

Το Visual Studio βοηθάει προεπιλεγμένα τον προγραμματιστή να δημιουργεί δομημένο κώδικα, προσθέτοντας αυτόματα εσοχές και κάνοντας αυτόματη στοίχιση. Όπως τα περισσότερα χαρακτηριστικά του Visual Studio, και η μορφοποίηση μπορεί να ρυθμιστεί. Από το μενού Tools -> Options επιλέγουμε Text Editor -> C# -> Formatting.

Το χαρακτηριστικό της προσθήκης εσοχών στη C# καλείται Smart indenting. Μεταξύ άλλων προσθέτει αυτόματα εσοχές κατά την προσθήκη μπλοκ κώδικα μέσα σε ένα αρχείο. Αν για οποιοδήποτε λόγο δεν μορφοποιήσει σωστά τον κώδικα, υπάρχει και η δυνατότητα επαναμορφοποίησης ολόκληρου του αρχείου κώδικα ή τμήματος του.

## Επισήμανση αναφορών

Ένα ακόμα χαρακτηριστικό του Visual Studio είναι η επισήμανση των αναφορών. Όλες οι χρήσεις του τρέχοντος επιλεγμένου προσδιοριστή ( όπως πχ. μιας μεθόδου ή μιας Property) μέσα σε μια εμβέλεια επισημαίνονται. Έτσι γίνεται πιο εύκολο να εντοπίσουμε που αλλού χρησιμοποιείται αυτός ο προσδιοριστής μέσα στον κώδικα.



```
using (MySqlConnection myLoginConnection = new MySqlConnection(@"Data Source=localhost; Database=eclass; User ID=root; Password='dba'"))
{
    //Open the connection
    myLoginConnection.Open();
    using (MySqlCommand myLoginCommand = new MySqlCommand("delete from cours_user where user_id=" + Request["deleteid"] + " and user_id=2 ", my
    {
        if (!string.IsNullOrEmpty(Request["deleteid"]))
        {
            myLoginCommand.ExecuteNonQuery();
        }
    }
}

using (MySqlCommand myLoginCommand = new MySqlCommand("SELECT * FROM `cours_user` where user_id=2 ", myLoginConnection))
{
    var reader = myLoginCommand.ExecuteReader();
    while (reader.Read())
    {
        courses.Add((int)reader["cours_id"]);
    }
    reader.Close();
}
```

εικ 14) επισήμανση αναφορών στον κώδικα με γκρι χρώμα

## Code Zooming

Χρησιμοποιώντας τον συνδυασμό Ctrl + Mouse Wheel, μπορούμε να μεγαλώσουμε ή να μικρύνουμε το μέγεθος του κώδικα. Αυτό μπορεί να φανεί χρήσιμο όταν παρουσιάζουμε τον κώδικα σε τρίτους. Το zoom μπορεί να οριστεί και από την λίστα της γωνίας κάτω αριστερά στον επεξεργαστή.

## Word Wrap

Με τον όρο Word Wrap εννοούμε την αυτόματη διαχείριση γραμμών έτσι ώστε ο κώδικας να χωράει στο προκαθορισμένο πλαίσιο του επεξεργαστή κειμένου. Πηγαίνοντας από το μενού Tools -> Options, επιλέγουμε Text Editor -> C# ->

Formatting -> Wrapping και μπορούμε να αλλάξουμε τις επιλογές για το Wrapping.

### Αρίθμηση Γραμμών

Ένα πολύ χρήσιμο εργαλείο που δεν είναι ενεργοποιημένο από την αρχή, είναι η εμφάνιση του αριθμού των γραμμών σε κάθε αρχείο κώδικα. Για να εμφανιστούν οι γραμμές αριθμημένες από το μενού Tools -> Options, επιλέγουμε αρχικά στο παράθυρο επιλογών που εμφανίζεται Show all settings κάτω αριστερά. Στη συνέχεια, κάτω από την επιλογή Text Editor -> All Languages -> General επιλέγουμε Line numbers κάτω από την ομάδα Display. Τώρα τα αρχεία του κώδικα θα εμφανίζονται αριθμημένα.

### Fonts και χρώματα

Ένα από τα πρώτα πράγματα που ίσως χρειαστεί να αλλάξει κανείς στο Visual Studio είναι τα Fonts και τα χρώματα του κύριου επεξεργαστή κειμένου. Η επιλογή του κατάλληλου Font και των χρωμάτων που είναι ευκολοδιάβαστα για τον καθένα μπορεί να τον κάνει πιο παραγωγικό και να γράφει κώδικα για περισσότερο χρόνο χωρίς να κουράζεται. Μπορούμε να αλλάξουμε τα Fonts και τα χρώματα από το μενού Tools -> Options και από την επιλογή Environment επιλέγουμε Fonts and Colors. Από το ίδιο παράθυρο μπορούμε να αλλάξουμε τις επιλογές για οποιοδήποτε άλλο παράθυρο κειμένου του Visual Studio επιλέγοντας το από τη λίστα Show settings for. Κατά την επιλογή ενός font πρέπει να θυμόμαστε ότι περισσότερο αποτελεσματικά είναι τα Fonts σταθερού μήκους και για το λόγο αυτό εμφανίζονται με έμφαση στη λίστα των διαθέσιμων Fonts.

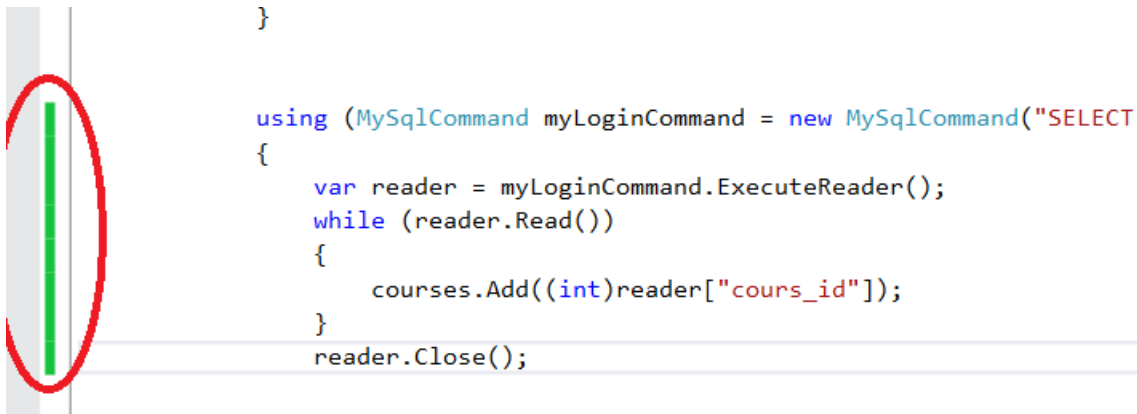
### Full-Screen Mode

Αν έχουμε ενεργά πολλαπλά παράθυρα εργαλείων, τότε μπορεί να ξεμεινουμε από χώρο για τον επεξεργαστή κειμένου. Για το λόγο αυτό το Visual Studio έχει διαθέσιμη μια full-screen Mode που μπορούμε να χρησιμοποιούμε από το μενού View -> Full Screen. Εναλλακτικά μπορούμε να επιλέξουμε Shift+Alt+Enter για να εισέλθουμε και να εξέλθουμε από αυτό το Mode. Όπως παρατηρούμε, δεν εμφανίζονται οι γραμμές εργαλείων ή τα παράθυρα των εργαλείων, ενώ το παράθυρο είναι πλήρως μεγιστοποιημένο. Το full-screen Mode μπορεί να φανεί ιδιαίτερα χρήσιμο, αν χρησιμοποιούμε πολλαπλές οθόνες.

### Παρακολούθηση Αλλαγών

Το Visual Studio έχει την επιλογή της παρακολούθησης των αλλαγών στον κώδικα ανά γραμμή για να υποδεικνύει ποιες γραμμές κώδικα έχουν αλλάξει κατά τη γραφή ενός αρχείου. Όταν ανοίγει ένα αρχείο για επεξεργασία, δεν θα υπάρχουν χρωματισμένες γραμμές. Ωστόσο, όταν ξεκινάμε το γράψιμο θα παρατηρήσουμε ότι εμφανίζεται μια κίτρινη ένδειξη δίπλα από κάθε γραμμή που έχει αλλάξει. Όταν το αρχείο αποθηκεύεται, οι γραμμές αλλάζουν χρώμα : από κίτρινο γίνεται πράσινο δείχνοντας με αυτόν τον τρόπο ότι έχουν γίνει αλλαγές από το άνοιγμα του αρχείου και ότι αυτές οι αλλαγές έχουν αποθηκευτεί. Η

επιλογή παρακολούθησης των αλλαγών μπορεί να αλλάξει από το μενού Tools -> Options -> Text Editor -> General και από την ομάδα Settings ενεργοποιούμε ή απενεργοποιούμε το check BOX "Track Changes".

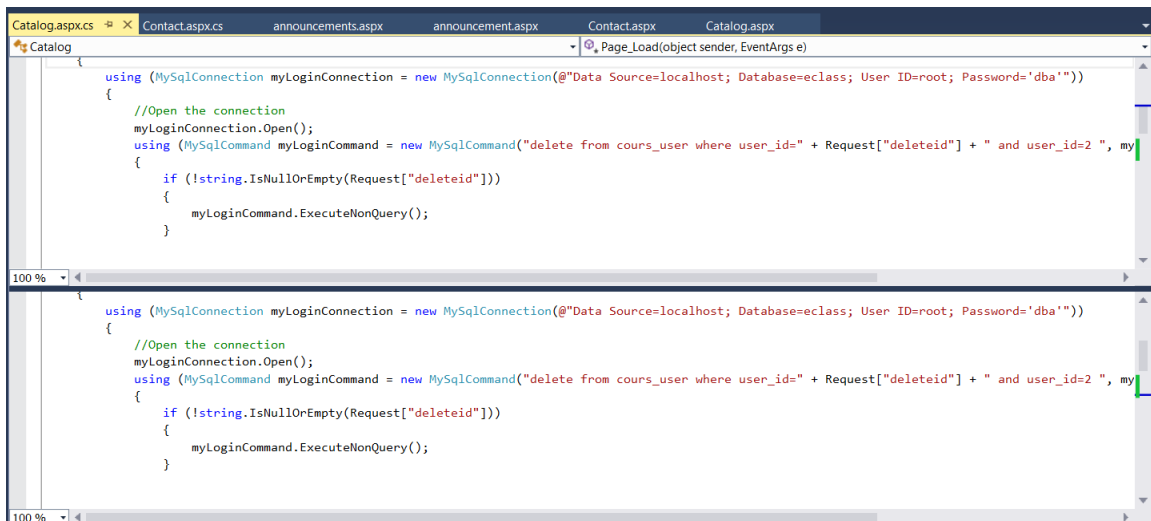


εικ 15) Πως εμφανίζονται οι αλλαγές με την παρακολούθηση αλλαγών ενεργοποιημένη

## Split View

Μερικές φορές είναι χρήσιμο να βλέπουμε διαφορετικά κομμάτια του ίδιου αρχείου ταυτόχρονα. Αυτό μπορεί να γίνει με τη Split View, η οποία επιτρέπει τη διαίρεση του ενεργού παραθύρου του επεξεργαστή κειμένου σε δύο οριζόντιους χώρους που χωρίζονται από μια μπάρα. Αυτοί οι χώροι μπορούν να γίνουν scrolled ξεχωριστά και να εμφανίζουν διαφορετικά κομμάτια του ίδιου αρχείου ταυτόχρονα.

Η διαίρεση του κύριου παραθύρου μπορεί να γίνει από το μενού Window -> Split. Η μπάρα διαχωρισμού μπορεί να ρυθμιστεί, σύροντας την πάνω η κάτω.



εικ 16) Split View

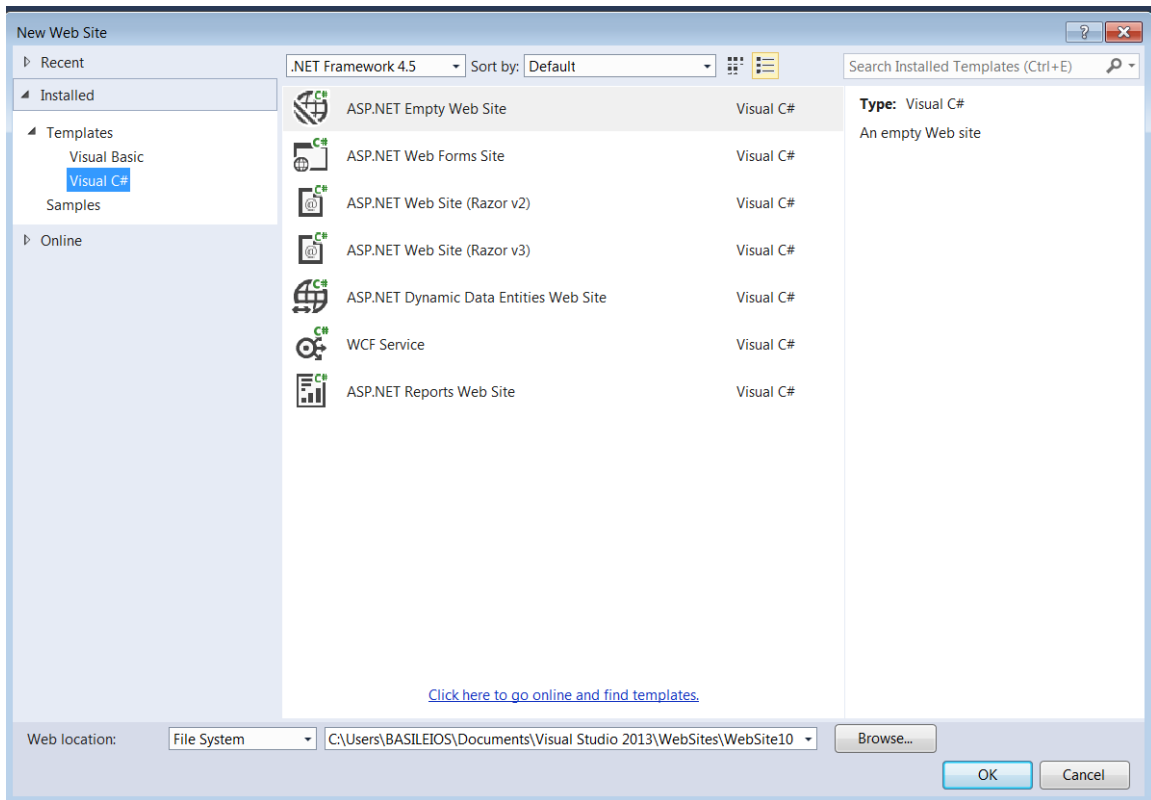
## Άλλες λειτουργίες

Για να βελτιώσει κανείς την παραγωγικότητα του είναι χρήσιμο να γνωρίζει κάποιες προχωρημένες λειτουργίες του επεξεργαστή κειμένου που δεν είναι ευδιάκριτες αλλά μπορούν να εξοικονομήσουν πολύ χρόνο κατά την ανάπτυξη.

- Σχολιασμός / από-σχολιασμός ενός μπλοκ κώδικα : Συχνά χρειάζεται να τοποθετήσουμε σε σχόλια, κώδικα που εκτείνεται σε πολλές γραμμές. Για να αποφύγουμε την τοποθέτηση σχολίων γραμμής χειροκίνητα και την αποφυγή της πιθανότητας εισαγωγής πολλαπλών σχολίων πολλαπλών γραμμών, χρησιμοποιούμε τον αυτόματο σχολιασμό του Visual Studio. Από το μενού Edit επιλέγουμε Advanced -> Comment Selection για το σχολιασμό και Advanced -> Uncomment για να αφαιρέσουμε τα σχόλια.
- Επιλογή μπλοκ : γνωστή και ως επιλογή ορθογωνίου, είναι η δυνατότητα επιλογής κειμένου σε μπλοκ αντί της κανονικής επιλογής γραμμών. Για να επιλέξουμε μπλοκ κειμένου, απλώς πιέζουμε το πλήκτρο alt ενώ επιλέγουμε το κείμενο με το ποντίκι.
- Επεξεργασία πολλαπλών γραμμών : ένα άλλο χαρακτηριστικό του Visual Studio είναι η δυνατότητα μετά την επιλογή ενός μπλοκ με τον παραπάνω τρόπο να επιτρέπεται η εισαγωγή νέου κειμένου που θα αντικαταστήσει το επιλεγμένο. Αυτό μπορεί να φανεί χρήσιμο όταν θέλουμε να αλλάξουμε για παράδειγμα τους προσδιοριστές πρόσβασης μιας ομάδας μεταβλητών.
- Διαχείριση clipboard: το Visual Studio διατηρεί τα τελευταία 20 snippets κειμένου που έχουν τοποθετηθεί στο clipboard. Κρατώντας πατημένα Ctrl+Shift και πατώντας V διαδοχικά εμφανίζονται τα snippets, ένα κάθε φορά.
- Go to Definition: για να πάμε γρήγορα στον ορισμό μιας κλάσης, μεθόδου ή μέλους κλάσης που υποδεικνύεται από τον κέρσορα πατάμε δεξί κλικ και go to definition, ή απλώς πατάμε F12
- Find all References: Μπορούμε να βρούμε από πού καλείται μια μέθοδος ή property με το να πατήσουμε δεξί κλικ στο ορισμό της και από το μενού επιλέγοντας Find All References ή πατώντας Shift+F12. Αυτό ενεργοποιεί το εργαλείο Find Symbol Results και εμφανίζει τις τοποθεσίες όπου υπάρχουν αναφορές της μεθόδου ή property. Μπορούμε να επιλέξουμε κάποια από αυτές τις τοποθεσίες και να εμφανίσουμε κατευθείαν τη γραμμή όπου εμφανίζεται η αναφορά.

## Δημιουργία ενός νέου ιστότοπου

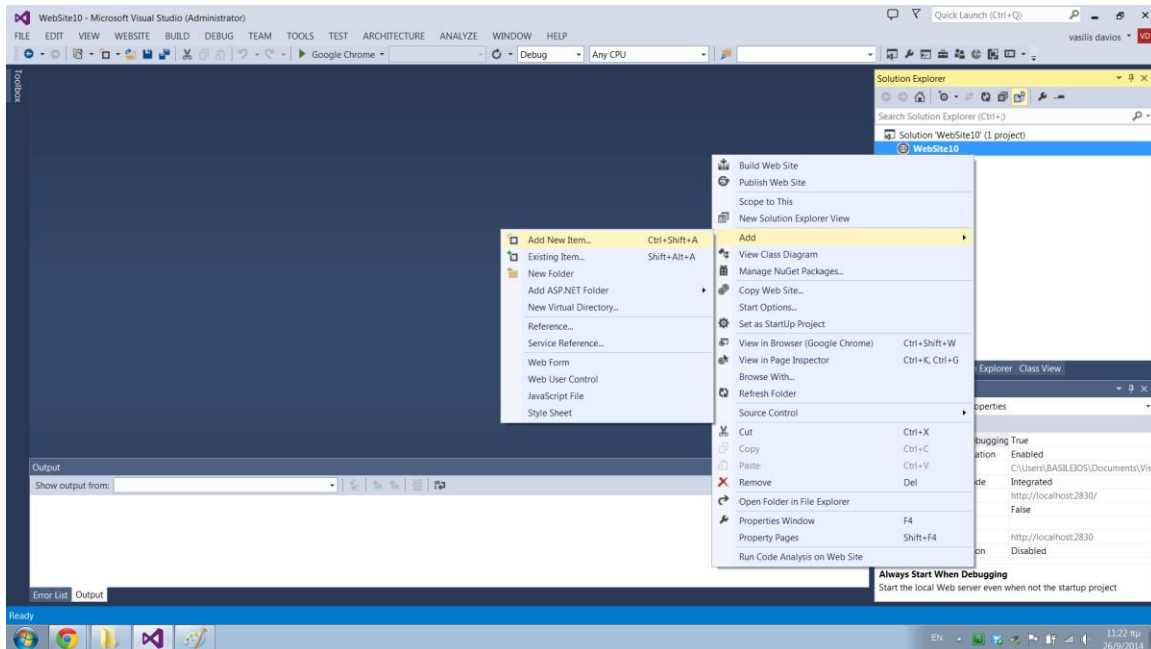
Δημιουργούμε ένα νέο κενό ιστότοπο επιλέγοντας File | New | website. Στη συνέχεια επιλέγουμε ASP.NET Empty Web Site και από το μενού Installed Templates τη γλώσσα προγραμματισμού που προτιμάμε και πατάμε OK. Στην περίπτωση μας αυτή είναι η C#.



εικ 17) Δημιουργία ενός κενού ιστότοπου από το Visual Studio

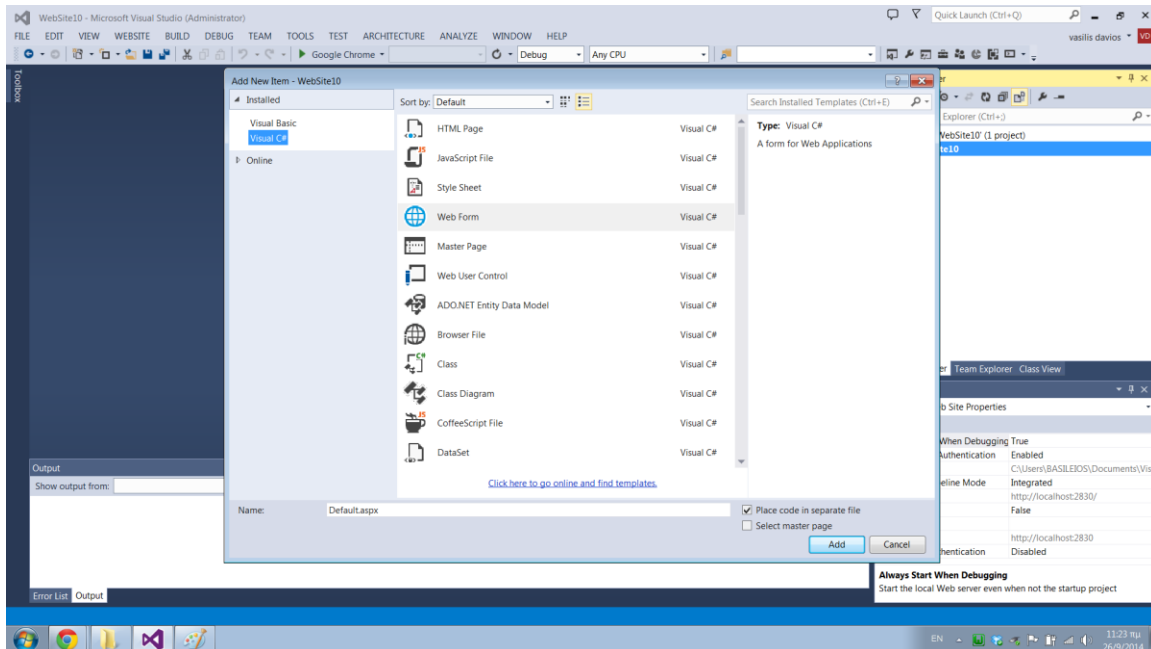
Σε αυτό το σημείο ο ιστότοπος που θέλουμε να δημιουργήσουμε είναι άδειος. Προσθέτουμε ένα Web Form κάνοντας δεξί κλικ στο website10 (όπως ονομάζεται το αρχείο μας) | add New item .





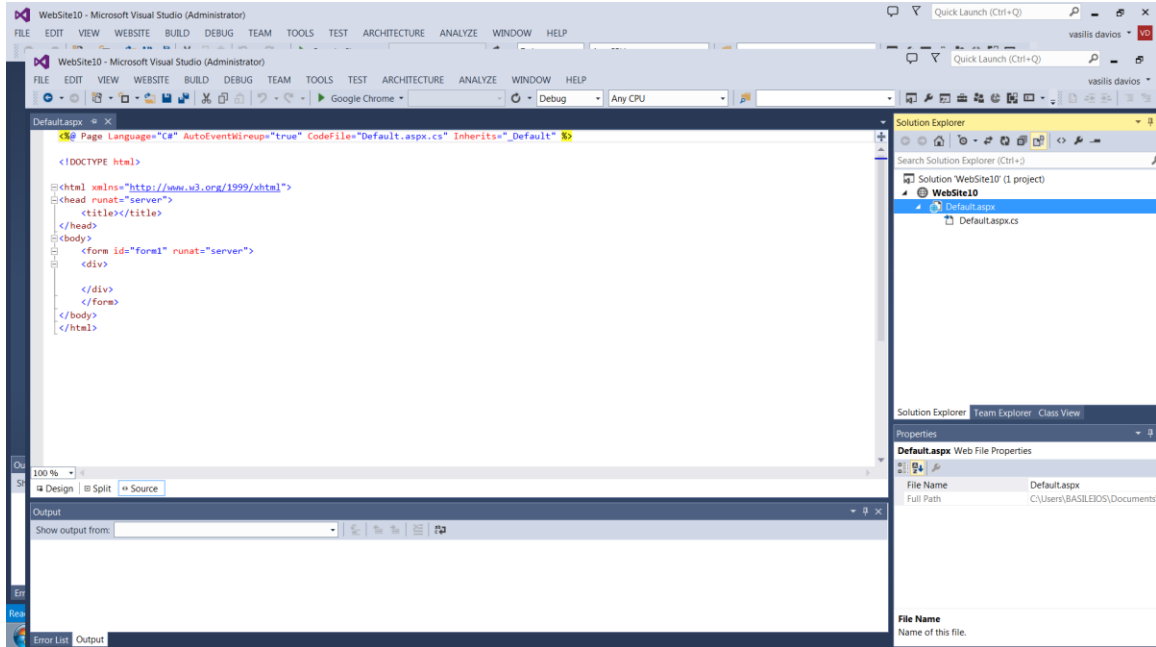
εικ 18) Προσθήκη ενός νέου αντικειμένου από το solution Explorer

Στο μενού που μας εμφανίζεται επιλέγουμε Add New Item. Από αυτό, επιλέγουμε Web Form και από το μενού Installed Templates | Visual C# και πατάμε το κουμπί Add.



εικ 19) Το παράθυρο με τις διάφορες επιλογές που μας εμφανίζεται κατά την προσθήκη ενός νέου αντικειμένου

Στην καινούργια σελίδα που εμφανίζεται με όνομα default.aspx προσθέτουμε τον html κώδικα μας.



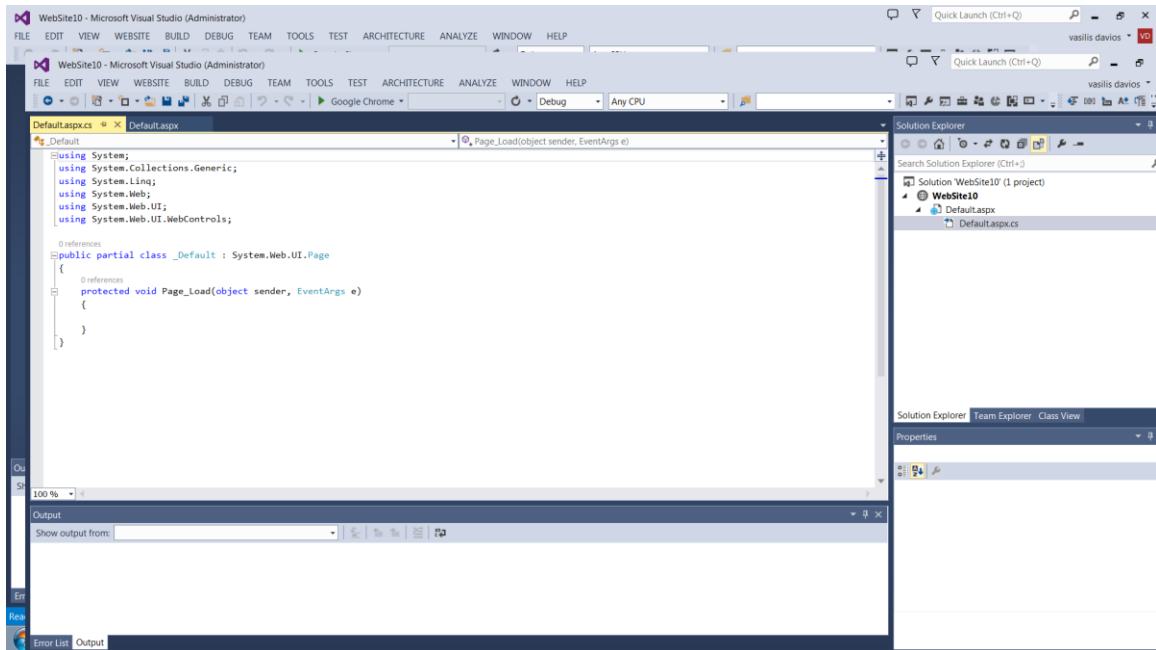
The screenshot shows the Microsoft Visual Studio interface with the file Default.aspx open. The code in the editor is as follows:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
</div>
</form>
</body>
</html>
```

The Solution Explorer on the right shows the project structure with Default.aspx selected. The Properties window below it shows the file name as Default.aspx and the full path as C:\Users\BASILEOS\Documents\.

εικ 20) Στην σελίδα default.aspx προσθέτουμε τον html κώδικα μας

Στην σελίδα Default.aspx.cs προσθέτουμε τον C# κώδικα μας.



The screenshot shows the Microsoft Visual Studio interface with the file Default.aspx.cs open. The code in the editor is as follows:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

The Solution Explorer on the right shows the project structure with Default.aspx.cs selected. The Properties window below it is empty.

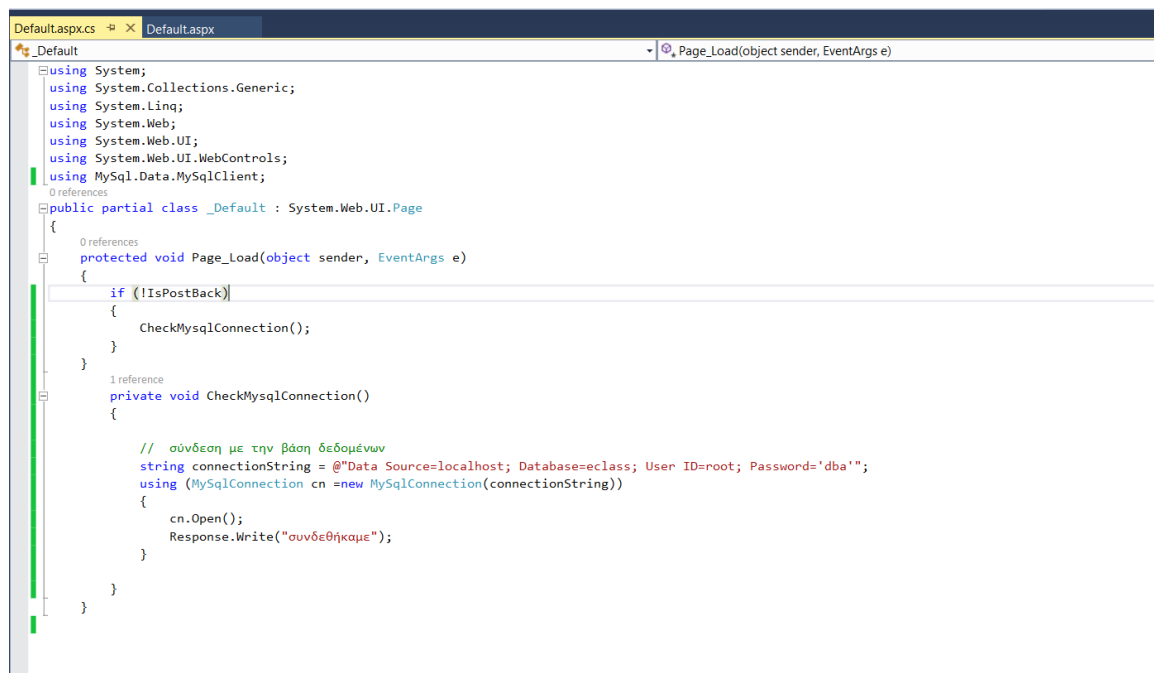
εικ 21) Η σελίδα Default.aspx.cs περιέχει των κώδικα C#.

Στη συνέχεια θα συνδέσουμε την ιστοσελίδα μας με μια βάση δεδομένων που έχουμε.

Για να συνδέσουμε την σελίδα μας με την βάση δεδομένων MySql που χρησιμοποιούμε θα χρειαστούμε να κάνουμε μια προσθήκη αναφοράς στο Project μας. Πατώντας δεξί κλικ στο Website10 που ονομάζεται το Project μας στο μενού που μας εμφανίζεται επιλέγουμε Add και στη συνέχεια πατάμε Reference. Την απαραίτητη αναφορά που χρειαζόμαστε για να συνδέσουμε το site μας με μια βάση δεδομένων MySql την κατεβάζουμε από την διεύθυνση : <http://dev.mysql.com/downloads/connector/net/1.0.html>.

Εφόσον προσθέσαμε την αναφορά μας στο Project είμαστε έτοιμοι να γράψουμε των κώδικα που θα μας χρειαστεί. Όπως έχουμε αναφέρει στα παραπάνω κεφάλαια η Html γλώσσα είναι η γλώσσα που χρησιμοποιούμε για να δημιουργήσουμε το γραφικό περιβάλλον του site μας. Η σύνδεση του site με την βάση δεδομένων θα γίνει γράφοντας τον κατάλληλο κώδικα C# στο αρχείο Default.aspx.cs.

Στην παρακάτω εικόνα βλέπουμε τον κώδικα που έκανε εφικτή την σύνδεση του site μας με την βάση δεδομένων.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MySql.Data.MySqlClient;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            CheckMysqlConnection();
        }
    }

    private void CheckMysqlConnection()
    {
        // σύνδεση με την βάση δεδομένων
        string connectionString = @"Data Source=localhost; Database=eclass; User ID=root; Password='dba'";
        using (MySqlConnection cn = new MySqlConnection(connectionString))
        {
            cn.Open();
            Response.Write("συνδεθήκαμε");
        }
    }
}
```

εικ 22) Ο κώδικας σύνδεσης του site με την βάση δεδομένων.

Το πρώτο βήμα που πρέπει να κάνουμε για να συνδέσουμε το site μας με την βάση δεδομένων, είναι να προσθέσουμε κάποιες βιβλιοθήκες που περιέχονται στην αναφορά που προσθέσαμε στο Project μας, ώστε το Visual Studio να αναγνωρίζει κάποιες από τις εντολές που θα γράψουμε στον κώδικα μας.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MySql.Data.MySqlClient;
```

0 references

εικ 23) Οι βιβλιοθήκες που χρησιμοποιούνται για το Project μας

Η βιβλιοθήκη που περιέχεται στην αναφορά μας και την προσθέσαμε στο Project μας είναι η `MySql.Data.MySqlClient`

Χωρίς αυτή την προσθήκη ο Text editor του Visual Studio δεν θα μπορούσε να κατανοήσει βασικές εντολές όπως το `connectionString`, που βλέπουμε στην παρακάτω εικόνα, και είναι το κομμάτι που δείχνει στο site που ακριβώς να συνδεθεί με ποια βάση και ποια είναι η ταυτότητα του χρήστη και ο κωδικός του ώστε να αποκτήσει πρόσβαση στη βάση.

```
// σύνδεση με την βάση δεδομένων
string connectionString = @"Data Source=localhost; Database=eclass; User ID=root; Password='dba'";
using (MySqlConnection cn =new MySqlConnection(connectionString))
{
    cn.Open();
    Response.Write("συνδεθήκαμε");
}
```

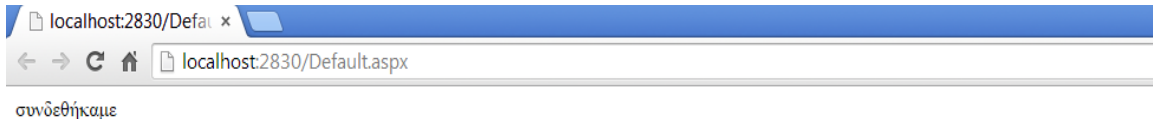
εικ 24) Ο κώδικας C# που κάνει εφικτή την σύνδεση του site με την βάση δεδομένων MySql

Κάτω από το `connectionString` βλέπουμε πως χρησιμοποιούμε μια κλάση ,η οποία περιέχει της απαραίτητες πληροφορίες από το `connectionString`, για να δημιουργήσουμε μια νέα σύνδεση με την βάση δεδομένων μας. Αναφερόμαστε στο κόμματα του κώδικα :

```
using (MySqlConnection cn =new MySqlConnection(connectionString))
```

Στην συνέχεια εφόσον η σύνδεση με την βάση δεδομένων έχει ολοκληρωθεί και είναι ανοιχτή εμφανίζεται στο browser μας ένα μήνυμα που μας ενημερώνει ότι συνδεθήκαμε επιτυχώς. Αυτό γίνεται μέσω του κώδικα:

```
{
    cn.Open();
    Response. Write ("συνδεθήκαμε");
}
```



εικ 25) Το αποτέλεσμα της σύνδεσης του site μας με την βάση δεδομένων

Σαν επόμενο βήμα θα προσπαθήσουμε να εμφανίσουμε στο site μας ένα πίνακα με πληροφορίες από την βάση δεδομένων μας.

Στις παρακάτω εικόνες ακολουθεί ο απαραίτητος κώδικας για την σύζευξη του ιστότοπου που δημιουργήσαμε με τις επιθυμητές πληροφορίες από την βάση δεδομένων που χρησιμοποιούμε.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MySql.Data.MySqlClient;
using System.Data;

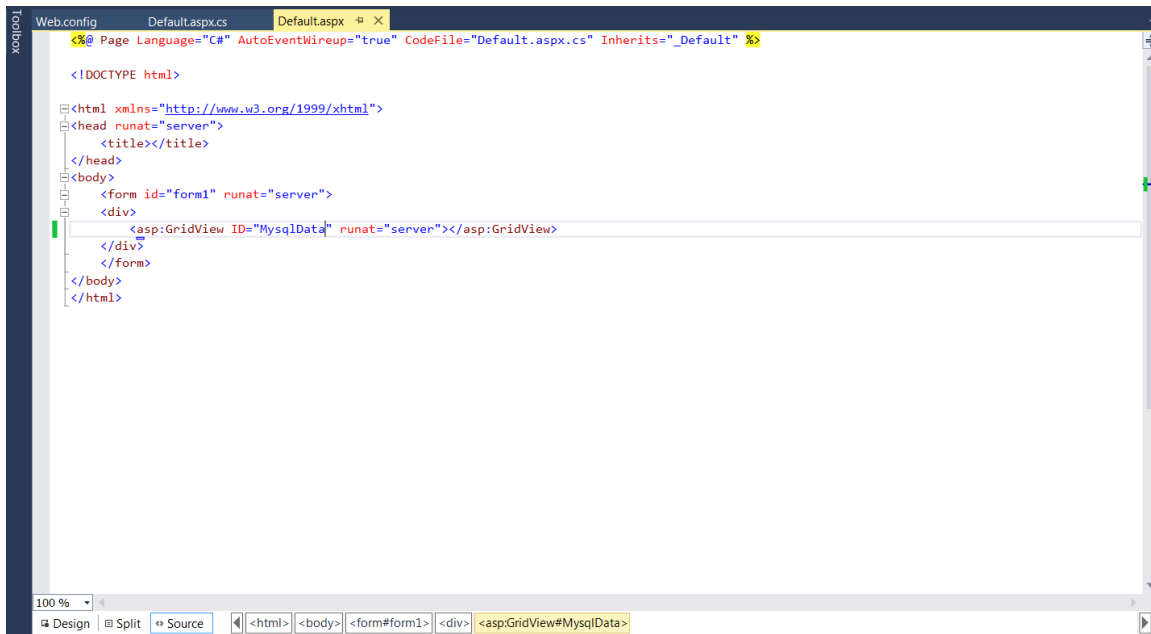
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            //CheckMySQLConnection();
            BindData();
        }
    }

    private void CheckMySQLConnection()
    {
        // σύνδεση με την βάση δεδομένων
        string connectionString = @"Data Source=localhost; Database=class; User ID=root; Password='dba'";
        using (MySqlConnection cn = new MySqlConnection(connectionString))
        {
            cn.Open();
            Response.Write("συνδεθήκαμε");
        }
    }

    private void BindData()
    {
        string connectionString = @"Data Source=localhost; Database=class; User ID=root; Password='dba'; convert zero datetime=True";
        using (MySqlConnection cn = new MySqlConnection(connectionString))
        {
            MySqlDataAdapter adp = new MySqlDataAdapter("select * from cours", cn);
            DataTable dt = new DataTable();

            adp.Fill(dt);
            if (dt.Rows.Count >= 0)
            {
                MySqlData.DataSource = dt;
                MySqlData.DataBind();
            }
        }
    }
}
```

εικ 25) Ο απαραίτητος C# κώδικας για το Project μας



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:GridView ID="MySQLData" runat="server"></asp:GridView>
</div>
</form>
</body>
</html>
```

εικ 26) Ο Html κώδικας

Το πρώτο βήμα που κάναμε για να ολοκληρώσουμε το Project μας ήταν να προσθέσουμε έναν πίνακα στο site μας. Για να μπορέσουμε να εμφανίσουμε τις πληροφορίες που επιθυμούμε από τη βάση δεδομένων σε μορφή πίνακα χρησιμοποιήσαμε το GridView. Το GridView μπορούμε να το βρούμε στο Toolbox του Visual Studio στο μενού Data. Το επόμενο βήμα που κάναμε ήταν να ονομάσουμε τον πίνακα μας ώστε να μπορέσουμε να κατευθύνουμε τις πληροφορίες που επιθυμούμε εκεί. Το όνομα που επιλέξαμε είναι MySQLData.

Περνώντας τώρα στο κομμάτι του C# κώδικα το πρώτο πράγμα που κάναμε ήταν να συνδέσουμε το site μας με τη βάση δεδομένων. Αυτό έγινε με τον εξής κώδικα:

```
string connectionString = @"Data Source=localhost; Database=eclass; User ID=root; Password='dba'; convert zero datetime=True";
using (MySQLConnection cn = new MySQLConnection(connectionString))
```

εικ 27) Ο κώδικας σύνδεσης του site μας με την βάση δεδομένων που χρησιμοποιούμε

Όπως και στο προηγούμενό Project που κάναμε πρέπει να δώσουμε τις απαραίτητες πληροφορίες στο site μας για να συνδεθεί με τη βάση δεδομένων. Το που βρίσκεται και πώς ονομάζεται η βάση δεδομένων, το όνομα και τον κωδικό του χρήστη με τα οποία θα γίνει εφικτή η σύνδεση με την βάση δεδομένων.

Σε αυτό το Project στόχος μας είναι να συνδεθούμε στην βάση να πάρουμε κάποιες πληροφορίες από ένα πίνακα της επιλογής μας που βρίσκεται στη βάση δεδομένων και να τα εμφανίσουμε στον ιστότοπο που δημιουργούμε.

Εφόσον συνδέσαμε τον ιστότοπο μας με την βάση δεδομένων περνάμε στο δεύτερο βήμα στην ανάκτηση δηλαδή των επιθυμητών πληροφοριών.

Αυτό έγινε με τον κώδικα που ακολουθεί στην παρακάτω εικόνα.

```
MySqlDataAdapter adp = new MySqlDataAdapter("select * from cours", cn);  
DataTable dt = new DataTable();
```

εικ 28) Ανάκτηση των επιθυμητών πληροφοριών και δημιουργία πίνακα

Ο `MySqlDataAdapter` είναι ένας προσαρμογές που χρησιμοποιείται σαν “γέφυρα” μεταξύ του `System. Data` και της `MySql` για την ανάκτηση και την αποθήκευση δεδομένων. Δίνουμε ένα όνομα στον `Data Adapter` και τον γεμίζουμε με τα δεδομένα που θέλουμε. Στην περίπτωση μας τα επιθυμητά δεδομένα βρίσκονται σε ένα πίνακα με την ονομασία `cours` στην βάση δεδομένων μας. Στη συνέχεια δημιουργούμε ένα καινούργιο πίνακα δεδομένων και τον ονομάζουμε (`dt`). Για να γεμίσουμε τον πίνακα με τα δεδομένα που έχουμε στον `Data Adapter` γράφουμε τον ακόλουθο κώδικα :

```
adp.Fill(dt);  
if (dt.Rows.Count >0)  
{  
    MySqlData.DataSource = dt;  
    MySqlData.DataBind();  
}
```

εικ 29) Δεδομένα του καινούργιου πίνακα

Δίνουμε την εντολή ο `Data Adapter` μας που τον έχουμε ονομάσει `adp` να γεμίσει τον καινούργιο μας πίνακα τον `dt`. Αυτό γίνεται με την εντολή : `adp.Fill(dt);`  
Το τελευταίο βήμα για την ολοκλήρωση του `Project` μας είναι να εμφανίσουμε τις πληροφορίες που έχουμε πάρει από την βάση δεδομένων μας στον πίνακα του `site` μας `GridView` τον οποίο έχουμε ονομάσει `MySqlData`. Η πηγή που θα μας δίνει αυτές τις πληροφορίες και θα γεμίζει αυτό τον πίνακα είναι ο `dt`. Αυτό γίνεται εφικτό με τις ακόλουθες δύο εντολές :

```
MySqlData.DataSource = dt;  
MySqlData.DataBind ();
```

Παρακάτω ακολουθεί το αποτέλεσμα του `Project` που μας εμφανίζεται στο πρόγραμμα πλοήγησης:

course_id	code	language	Course	intitule	description	course_keywords	course_addon	course_license	visible	titulaires	tske_code	departmentUrlName	departmentUrl	lastVisit	lastEdt	expirationDate	first_create	type	disc_quota	video_quota	group_qt
1	TMA100	greek	Εργαστήριο Εφαρμοσμένου Ηλεκτρομαγνητισμού	Κουρούσης, Σπυρίδης, Τζαμπίκος, Εμμανήλ			0	1			TMA100			1/1/0001	1/1/0001	19/7/2014 3:28:46 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
2	TMA101	greek	Εφαρμοσμένος Ηλεκτρομαγνητισμός	Πετρίδης, Κωσταντίνος			0	1			TMA101			1/1/0001	1/1/0001	19/7/2014 4:14:34 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
3	TMA102	greek	Φυσικά Συστήματα Επισκοπιών	Ευώγγεος, Κώστας			0	1			TMA102			1/1/0001	1/1/0001	19/7/2014 4:22:17 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
4	TMA103	greek	CAD και Κατοικολογία	Χατζόπουλος, Ιωάννης			0	1			TMA103			1/1/0001	1/1/0001	16/9/2014 8:26:39 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
5	TMA104	greek	ΜΑΘΗΜΑΤΙΚΑ ΜΕ ΧΡΗΣΗ ΗΥ 1	Πλάκης, Δημήτρης			0	2			TMA104			1/1/0001	1/1/0001	16/9/2014 8:30:04 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
6	TMA105	greek	Εφαρμογές των Μικροελεγκτών στις Τριεπισκοπιές	Μπαρμπούσανης, Ιωάννης			0	1			TMA105			1/1/0001	1/1/0001	16/9/2014 8:42:12 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
7	TMA106	greek	Φυσικά Συστήματα Επισκοπιών (Epr)	Κώστας, Ευώγγεος			0	1			TMA106			1/1/0001	1/1/0001	16/9/2014 8:45:04 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
8	TMA107	greek	Ανάλυση & Σύνθεση Κυκλωμάτων με ΗΥ	Δρ. Αντώνης, Ι. Κωνσταντίνος, Επισκοπος, Κοθηγητής			0	1			TMA107			1/1/0001	1/1/0001	16/9/2014 8:52:11 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
9	TMA108	greek	Αρχιτεκτονική ΗΥ (Εργαστήριο)	Πετρίδης, Νικόλαος, Ρηγάκης, Ηρακλής			0	1			TMA108			1/1/0001	1/1/0001	16/9/2014 8:53:34 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
10	TMA109	greek	Αρχιτεκτονική ΗΥ [TA5002]	Πετρίδης, Νικόλαος			0	1			TMA109			1/1/0001	1/1/0001	16/9/2014 8:54:43 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
11	TMA110	greek	Δορυφορικές (και Κινητές) Επισκοπιές (E) [TA7102]	Βαριάμπαση, Ιωάννης, Κατατοπούλης, Θεόδωρος			0	1			TMA110			1/1/0001	1/1/0001	16/9/2014 8:56:04 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	
12	TMA111	greek	Δορυφορικές (και Κινητές) Επισκοπιές (Θ)	Βαριάμπαση, Ιωάννης			0	2			TMA111			1/1/0001	1/1/0001	16/9/2014 8:57:30 μμ	pre	2.09715E+08	1.04855E+08	1.04855E	

εικ 30) Το αποτέλεσμα του Project μας στο πρόγραμμα πλοήγησης

Τελειώνοντας με την θεωρία και τα παραδείγματα για τις γλώσσες και τα εργαλεία που χρησιμοποιήσαμε για την τροποποίηση του eclass στο επόμενο κεφάλαιο θα δούμε τον κώδικα που κρύβεται πίσω από το eclass και την εικόνα που θα δούμε αν το ανοίξουμε από το κινητό μας.

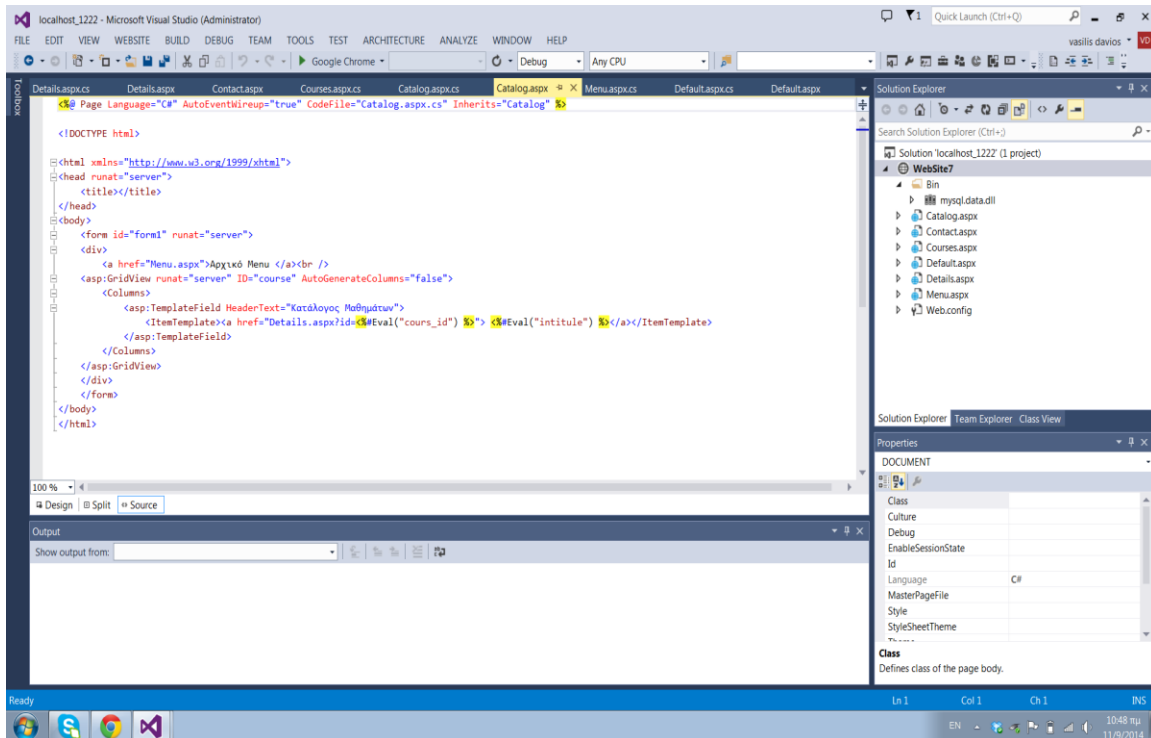


## Κεφάλαιο 4

Περιγραφή κώδικα του E-class

## Τροποποιώντας το e-class για κινητά τηλέφωνα

Η δημιουργία του eclass αποτελείται από δύο κομμάτια. Το πρώτο κομμάτι είναι η πρόσοψη του site, που για να την δημιουργήσουμε χρησιμοποιήσαμε την γλώσσα HTML. Για το καλλωπισμό του site (χρώματα επιλογή γραμματοσειράς κλπ.) χρησιμοποιούμε την CSS.

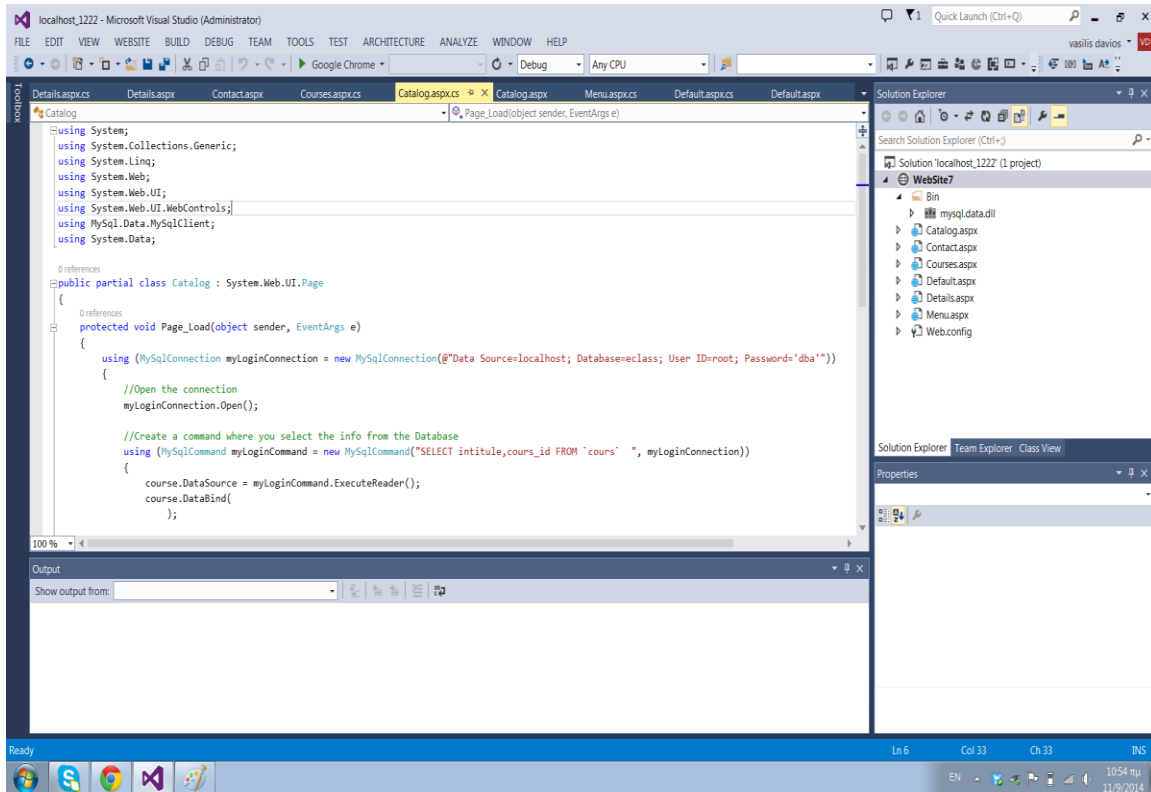


The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Contains ASP.NET code for a page. The code includes a form with a menu and a grid view of course details. Key elements include:
  - Form: `<form id="form1" runat="server">`
  - Menu: `<a href="Menu.aspx">Αρχικό Menu </a><br />`
  - GridView: `<asp:GridView runat="server" ID="course" AutoGenerateColumns="false">`
  - ItemTemplate: `<ItemTemplate><a href="Details.aspx?id={Eval("course_id")}>{Eval("intitle")}</a></ItemTemplate>`
- Solution Explorer:** Shows the project structure for 'WebSite7', including files like `Catalog.aspx`, `Contact.aspx`, `Courses.aspx`, `Default.aspx`, `Details.aspx`, `Menu.aspx`, and `Web.config`.
- Properties Window:** Shows the 'DOCUMENT' properties for the selected file, such as `Class`, `Culture`, `Debug`, `EnableSessionState`, `Id`, `Language` (set to 'C#'), `MasterPageFile`, `Style`, and `StyleSheetTheme`.

εικ 31) Ο κώδικας html που δημιουργήσαμε στο Visual Studio για την διαμόρφωση της πρόσοψης του ιστότοπου e-class

Το δεύτερο κομμάτι αφορά την σύνδεση του site με την βάση δεδομένων, που χρησιμοποιούμε για να αποθηκεύουμε τους χρήστες και περεταίρω στοιχεία για αυτούς. Για να γίνει εφικτή αυτή η σύνδεση χρησιμοποιήσαμε την γλώσσα προγραμματισμού C#.



εικ 32) Η σύνδεση της κάθε σελίδας με την βάση δεδομένων γίνεται με την χρήση της C# στο Visual Studio όπως φαίνεται στην παραπάνω εικόνα

# Παρουσίαση και επεξήγηση του κώδικα του eclass

Σ αυτό το κεφάλαιο θα δούμε μια μια τις σελίδες του eclass και τον κώδικα που τρέχει πίσω από τις σελίδες.

## 1) Master Page

Για την διαμόρφωση του eclass χρησιμοποιήσαμε μια έτοιμη βιβλιοθήκη CSS η οποία λέγεται bootstrap. Την τοποθετήσαμε σε μια ξεχωριστή σελίδα την οποία ονομάσαμε Master Page. Στην συνέχεια συνδέσαμε κάθε σελίδα του eclass που δημιουργήσαμε μ' αυτήν την Master Page που περιέχει κάποιες βασικές επιλογές που κάναμε και θέλαμε να υπάρχουν σε όλες τις σελίδες μας, όπως την γραμματοσειρά που χρησιμοποιήσαμε, τον τρόπο που θα εμφανίζονται οι λίστες μας κ.α. Επίσης μέσα στη βιβλιοθήκη του bootstrap υπάρχει και ο απαραίτητος κώδικας JavaScript για την λειτουργία διάφορων κουμπιών που βάλαμε στο eclass.

```
<meta charset="utf-8"/>
<meta http-equiv="X-UA-Compatible" content="IE=edge"/>
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<link href="bootstrap.min.css" rel="stylesheet"/>
<script src="jquery.min.js"></script>

<script src="bootstrap.min.js"></script>
```

### 3) Το κεφάλι του html κώδικα της Master Page

Στο κεφάλι του κώδικα της Master Page, όπως βλέπουμε στην παραπάνω εικόνα, βάζουμε τις απαραίτητες βιβλιοθήκες του bootstrap δηλώνοντας πως είναι css αρχείο stylesheet και βάζουμε και τις απαραίτητες βιβλιοθήκες JavaScript, της jQuery και της bootstrap.

```
<body style="padding: 15px">
  <form id="form1" runat="server">
    <br /><br />

    <nav class="navbar navbar-default" role="navigation">
      <div class="container-fluid">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#menu">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
        </div>

        <div class="collapse navbar-collapse" id="menu">
          <ul class="nav navbar-nav">
            <li><a href="Courses.aspx">Τα μαθήματα μου </a></li>
            <li><a href="Catalog.aspx">Κατάλογος μαθημάτων </a></li>
            <li><a href="Contact.aspx">Επικοινωνία </a></li>
            <li><asp:LinkButton ID="LinkButton1" runat="server" OnClick="LinkButton1_Click">Logout</asp:LinkButton></li>
          </ul>
        </div><!-- /.navbar-collapse -->
      </div>
    </nav>

    <asp:ContentPlaceHolder runat="server" ID="main"></asp:ContentPlaceHolder>
  </form>
</body>
```

34) Το σώμα του html κώδικα της Master Page

Όπως βλέπουμε στην εικόνα 34 στο σώμα του html κώδικα της Master Page προσθέτουμε κάποιες βασικές επιλογές που θέλουμε να ισχύουν σε όλες τις σελίδες που θα δημιουργήσουμε για το eclass. Πρώτον βάζουμε ένα σύνδεσμο με το logo του ΤΕΙ( `img src="..."`).

Στην συνέχεια ορίζουμε πώς θα εμφανίζεται η λίστα με τις επιλογές πατώντας το κουμπί του μενού.

```
<nav class="navbar navbar-default" role="navigation">
<div class="container-fluid">
<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#menu">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
</div>
</div>
```

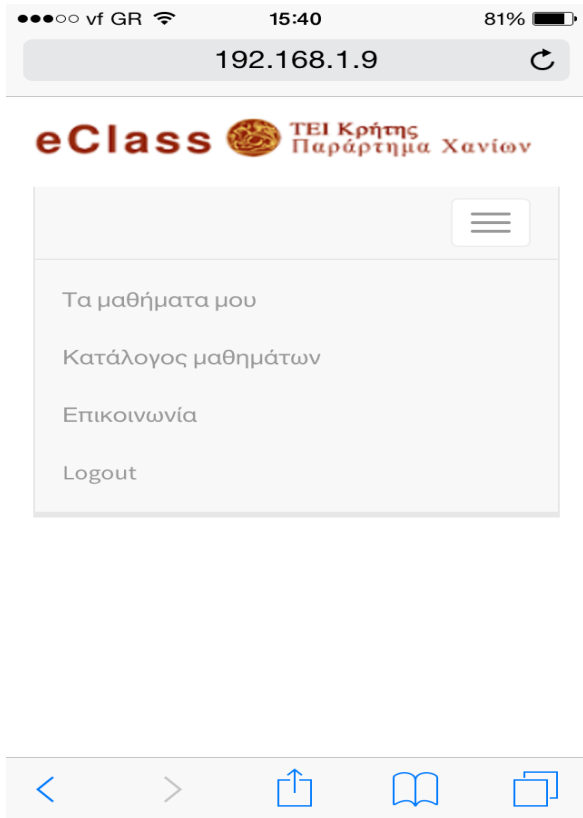
35) Ο κώδικας που ορίζει τον τρόπο που θα εμφανίζεται η λίστα όταν πατήσουμε το κουμπί του μενού

Τέλος επιλέξαμε σε κάθε σελίδα μας να εμφανίζεται ένα βασικό μενού με κάποιες επιλογές. Ο κώδικας που γράψαμε για να γίνεται αυτό είναι ο εξής :

```
<div class="collapse navbar-collapse" id="menu">
<ul class="nav navbar-nav">
<li><a href="Courses.aspx">Τα μαθήματα μου </a></li>
<li><a href="Catalog.aspx">Κατάλογος μαθημάτων </a></li>
<li><a href="Contact.aspx">Επικοινωνία </a></li>
<li><asp:LinkButton ID="LinkButton1" runat="server" OnClick="LinkButton1_Click">Logout</asp:LinkButton></li>
</ul>
</div><!-- /.navbar-collapse -->
</div>
</nav>
```

36) Οι τρεις βασικές επιλογές που έχει την δυνατότητα να χρησιμοποιήσει ο χρήστης από οποιαδήποτε σελίδα.

## 2) Το βασικό μενού



37) Το menu του eclass

Στην παραπάνω εικόνα βλέπουμε το menu του eclass όπως εμφανίζεται στην οθόνη ενός κινητού τηλεφώνου.

### A) Html κώδικας

Σαν html κώδικα το μόνο που χρειάστηκε να κάνουμε είναι να συνδέσουμε την σελίδα του μενού με την Master Page. Ο κώδικας για να γίνει αυτή η σύνδεση είναι ο εξής :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Menu.aspx.cs"
Inherits="Menu" MasterPageFile="~/MasterPage.master" %>
<asp:Content runat="server" ContentPlaceHolderID="main"></asp:Content>
```

Ο παραπάνω κώδικας μας συνδέει με την Master Page που περιέχει αυτά που επιθυμούσαμε να εμφανίζονται στην αρχική μας σελίδα.

### B) C# κώδικας

Στον C# κώδικα θέλαμε απλά να επιβεβαιώνουμε πως ο χρήστης είναι συνδεδεμένος ώστε να του εμφανιστεί η αρχική σελίδα αλλιώς θα του ζητούσαμε να συνδεθεί με τον λογαριασμό του. Αυτό γίνεται με τον ακόλουθο κώδικα

```

public partial class Menu : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Request.IsAuthenticated)
        {
            Response.Redirect("Default.aspx");
        }
    }
}

```

38) Ο C# κώδικας της αρχικής σελίδας

### 3) Τα μαθήματα που παρακολουθεί ο χρήστης

●○○○ vf GR 15:40 81%  
 192.168.1.9

**eClass** ΤΕΙ Κρήτης  
 Παράρτημα Χανίων

Τα μαθήματα που παρακολουθώ

<a href="#">Εργαστήριο Εφαρμοσμένου Ηλεκτρομαγνητισμού</a>	<a href="#">Διαγραφή</a>
<a href="#">CAD και Κατασκευή</a>	<a href="#">Διαγραφή</a>
<a href="#">Αρχιτεκτονική Η/Υ [ΤΛ5002]</a>	<a href="#">Διαγραφή</a>
<a href="#">Δορυφορικές (και Κινητές) Επικοινωνίες (Ε) [ΤΛ7102]</a>	<a href="#">Διαγραφή</a>
<a href="#">Δορυφορικές (και Κινητές) Επικοινωνίες (Θ) [ΤΛ7002]</a>	<a href="#">Διαγραφή</a>
<a href="#">Επεξεργασία Φωνής και Εικόνας (Εργαστήριο)</a>	<a href="#">Διαγραφή</a>
<a href="#">Εργαστήριο Ηλεκτρονικών</a>	<a href="#">Διαγραφή</a>

Στην παραπάνω εικόνα φαίνεται η λίστα με τα μαθήματα που παρακολουθεί ο χρήστης και του δίνεται η δυνατότητα να διαγραφεί από όποιο μάθημα επιθυμεί.

## A) Html κώδικας

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Courses.aspx.cs" Inherits="Courses" MasterPageFile="~/MasterPage.master" %>
<asp:Content runat="server" ContentPlaceHolderID="main">
  <div>
    <asp:GridView runat="server" ID="course" AutoGenerateColumns="false" CssClass="table table-striped">
      <Columns>
        <asp:TemplateField HeaderText="Τα μαθήματα που παρακολουθώ">
          <ItemTemplate><a href="Details.aspx?id=<%#Eval("course_id") %>"><%#Eval("intitule") %></a></ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField><ItemTemplate><a href="courses.aspx?Deleteid=<%#Eval("course_id") %>">Διαγραφή</a></ItemTemplate>
        </asp:TemplateField>
      </Columns>
    </asp:GridView>
  </div></asp:Content>
```

39) Html κώδικας της σελίδας με τα μαθήματα του χρήστη

Όπως βλέπουμε στην παραπάνω εικόνα αφού συνδέσαμε την σελίδα μας με την Master Page, στην συνέχεια βάλαμε ένα τίτλο στην λίστα με τα μαθήματα του χρήστη, και δημιουργήσαμε δύο links. Το πρώτο μας εμφανίζει τα μαθήματα του χρήστη και το δεύτερο διαγράφει τον χρήστη από όποιο μάθημα αυτός επιλέξει.

## B) C# κώδικας

```
public partial class Courses : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Request.IsAuthenticated)
        {
            Response.Redirect("Default.aspx");
        }
        using (MySQLConnection mySqlConnection = new MySQLConnection(@"Data Source=localhost; Database=class; User ID=root; Password=dba"))
        {
            //delete user
            mySqlConnection.Open();
            using (MySQLCommand mySqlCommand = new MySQLCommand("delete from course_user where course_id=" + Request["deleteid"] + " and user_id=" + User.Identity.Name, mySqlConnection))
            {
                if (!string.IsNullOrEmpty(Request["deleteid"]))
                {
                    mySqlCommand.ExecuteNonQuery();
                }
            }

            //Create a command where you select the info from the Database
            using (MySQLCommand mySqlCommand = new MySQLCommand("SELECT intitule,course.course_id FROM `course` join course_user on course.course_id = course_user.course_id where user_id = " + User.Identity.Name, mySqlConnection))
            {
                course.DataSource = mySqlCommand.ExecuteReader();
                course.DataBind();
            }

            //close connection
            mySqlConnection.Close();
        }
    }
}
```

40) Ο C# κώδικας της σελίδας μας



Στο πρώτο κομμάτι του κώδικα επιβεβαιώνουμε πως ο χρήστης μας είναι συνδεδεμένος και δίνουμε τις απαραίτητες πληροφορίες στην εφαρμογή μας ώστε να πάει και να συνδεθεί στην βάση δεδομένων μας.

```
if (!Request.IsAuthenticated)
{
    Response.Redirect ("Default.aspx");
}
Using (SqlConnection myLoginConnection = new SqlConnection (@"Data
Source=localhost; Database=eclass; User ID=root; Password='dba'"))
```

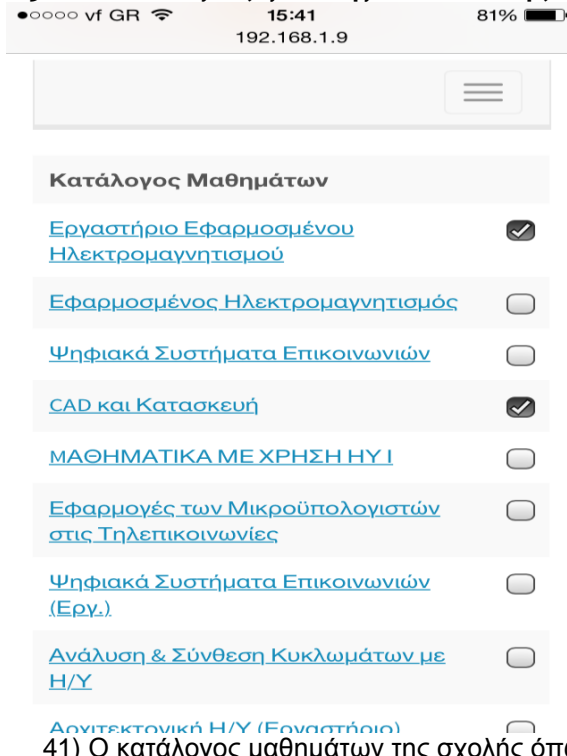
Στην συνέχεια παίρνουμε τις συγκεκριμένες πληροφορίες που επιθυμούμε από την βάση δεδομένων και τις εμφανίζουμε στην οθόνη του κινητού τηλεφώνου. Αυτό γίνεται με τον ακόλουθο κώδικα.

```
Using (MySQL Command myLoginCommand = new MySqlCommand ("SELECT intitule,
cours.cours_id FROM `cours` join cours_user on cours.cours_id =
cours_user.cours_id where user_id = «+ User.Identity.Name, myLoginConnection))
{
    Course.DataSource = myLoginCommand.ExecuteReader ();
    Course.DataBind (
        );
}
}
```

Το τελευταίο κομμάτι που μας έμεινε είναι ο κώδικας που δίνει την δυνατότητα στον χρήστη να διαγραφεί από όποιο μάθημα επιθυμεί. Δίνουμε αυτή την δυνατότητα στο χρήστη με τον εξής κώδικα

```
MyLoginConnection.Open ();
using (MySqlCommand myLoginCommand = new MySqlCommand("delete from
cours_user where courtside=" + Request["deleteid"] + " and user_id= " +
User.Identity.Name, myLoginConnection))
{
    if (!string.IsNullOrEmpty(Request["deleteid"]))
    {
        myLoginCommand.ExecuteNonQuery();
    }
}
```

## 4) Κατάλογος μαθημάτων της σχολής



41) Ο κατάλογος μαθημάτων της σχολής όπως εμφανίζεται στην οθόνη του χρήστη.

Για να δημιουργήσουμε την σελίδα με τον κατάλογο μαθημάτων της σχολής και να δώσουμε την δυνατότητα στον χρήστη να εγγραφεί ή να διαγραφεί σε όποιο μάθημα επιθυμεί γράψαμε τον εξής κώδικα

### A) Html κώδικας

Στο πρώτο μέρος του κώδικα συνδέουμε την σελίδα μας με την Master Page.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Catalog.aspx.cs" Inherits="Catalog" MasterPageFile="~/MasterPage.master" %>
```

Στην συνέχεια βάζουμε ένα link το οποίο θα μας εμφανίζει τα μαθήματα της σχολής και δίνουμε μια ονομασία στην λίστα αυτή.

```
<asp:Content runat="server" ContentPlaceHolderID="main">
  <div>
    <asp:GridView runat="server" ID="course" AutoGenerateColumns="false"
      CssClass="table table-striped">
      <Columns>
        <asp:TemplateField HeaderText="Κατάλογος Μαθημάτων">
          <ItemTemplate><a href="Details.aspx?id=<#Eval("cours_id") %>"
            <#Eval("intitule") %></a></ItemTemplate>
        </asp:TemplateField>
      </Columns>
    </asp:GridView>
  </div>
</asp:Content>
```

Τέλος δημιουργούμε ένα κουτί δίπλα από κάθε μάθημα και του δίνουμε την ιδιότητα όταν ο χρήστη το πατάει να τον γράφει στο αντίστοιχο μάθημα που

βρίσκεται το κουτί. Αν είναι ήδη γραμμένος στο μάθημα τότε μέσα στο κουτί θα εμφανίζεται ένα τικ που θα του το υποδεικνύει.

```
<asp:TemplateField>
<ItemTemplate>
    <input type="checkbox" <%=IsRegistered((int) Eval("cours_id"))?"checked":"" %>
    onclick="register(this,<%=Eval("cours_id")%>)" />
</ItemTemplate>
</asp:TemplateField>
</Columns>
</asp:GridView>
</div>
<script>
    function register(checkbox,id) {
        var registered = $(checkbox).is(":checked");
        $.ajax({url:"register.aspx?id="+id+"&action=" + registered});
    }
</script>

</asp:Content>
```

Για να λειτουργήσει αυτό το κουτί χρειάστηκε ο παραπάνω κώδικας javascript που βλέπουμε στην function.

B) Ο C# κώδικας

Το πρώτο κομμάτι του κώδικα αφορά την επιλογή των μαθημάτων της σχολής και την εμφάνισή τους σε μια λίστα.

```
using (MySqlCommand myLoginCommand = new MySqlCommand("SELECT
intitule,cours_id FROM `cours` ", myLoginConnection))
{
    var reader = myLoginCommand.ExecuteReader();
    course.DataSource = reader;
    course.DataBind();
    reader.Close();
}
```

Στην συνέχεια βρίσκουμε σε ποια μαθήματα είναι εγγεγραμμένος ο χρήστης που έχει συνδεθεί στο eclass.

```
using (MySqlCommand myLoginCommand = new MySqlCommand("SELECT * FROM
`cours_user` where user_id= " +User.Identity.Name, myLoginConnection))
{
    var reader = myLoginCommand.ExecuteReader();
    while (reader.Read())
    {
        courses.Add((int)reader["cours_id"]);
    }
    reader.Close();
}
```

Τέλος δίνουμε την δυνατότητα στην εφαρμογή να διαγράψει ή να εγγράψει τον χρήστη στα μαθήματα, ανάλογα με τον αν είναι ήδη εγγεγραμμένος στο μάθημα ή όχι.

```
myLoginConnection.Open();
```

```

using (SqlCommand myLoginCommand = new SqlCommand("delete from
cours_user where user_id=" + Request["deleteid"] + " and user_id= "
+User.Identity.Name, myLoginConnection))
{
    if (!string.IsNullOrEmpty(Request["deleteid"]))
    {
        myLoginCommand.ExecuteNonQuery();
    }
}

```

## 5)Μενού μαθήματος

Από τον κατάλογο μαθημάτων της σχολής αλλά και από την λίστα με τα μαθήματα που παρακολουθεί ο χρήστης μας δίνεται η δυνατότητα να επισκεφθούμε το μάθημα της επιλογής μας. Μέσα στο κάθε μάθημα μας δίνονται κάποιες δυνατότητες.

●○○○○ vf GR 15:41 81%

192.168.1.9

**eClass** ΤΕΙ Κρήτης  
Παράρτημα Χανίων

Ανακοινώσεις

Ομάδες Χρηστών

Εργαστηριακό μάθημα 7ου εξαμήνου,  
υποχρεωτικό στην κατεύθυνση  
Τηλεπικοινωνιών. Αντιστοιχεί σε 7 ΔΜ.

**Λέξεις Κλειδιά:** Δορυφόροι, Κινητά τηλέφωνα

42) Μενού μαθήματος

## A) Html κώδικας

Στον Html κώδικα όπως βλέπουμε έχουμε δημιουργήσει δύο links. Το ένα είναι οι ανακοινώσεις του μαθήματος και το άλλο είναι οι ομάδες χρηστών. Οι ομάδες χρηστών εμφανίζονται μόνο εάν ο καθηγητής του μαθήματος έχει δημιουργήσει ομάδες.

```
<asp:Content runat="server" ContentPlaceHolderID="main">
  <div>
    <a href="announcements.aspx?id=<%=Request["id"] %>" class="btn btn-warning
">Ανακοινώσεις</a><br />
    <%if (HasGroups){ %><br />
    <a href="groups.aspx?id=<%=Request["id"] %>" class="btn btn-primary
">Ομάδες Χρηστών</a><br />
    <%} %>
  <p>
```

Επίσης αν ο καθηγητής έχει γράψει κάποιες πληροφορίες για το μάθημα έχουμε επιλέξει να εμφανίζονται.

```
<asp:Literal runat="server" ID="description"></asp:Literal>
</p>

</div></asp:Content>
```

## B) Ο C# κώδικας

Στο πρώτο κομμάτι του κώδικα ζητάμε από την εφαρμογή να συνδεθεί με την βάση δεδομένων και να φέρει τις ανακοινώσεις του μαθήματος.

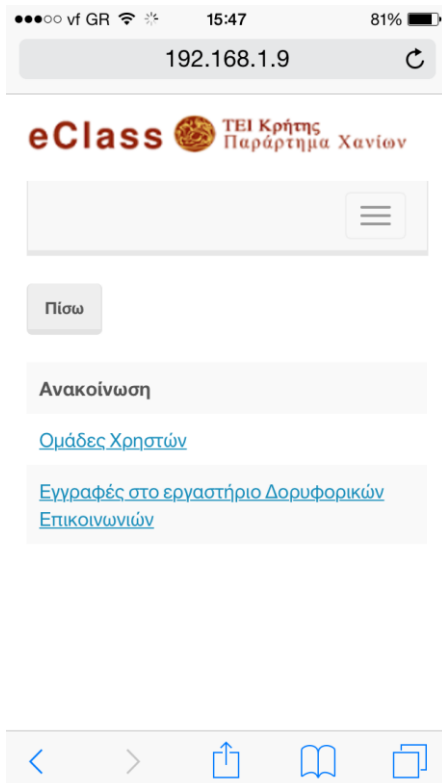
```
using (MySqlCommand myLoginCommand = new MySqlCommand("SELECT
Unit_resources.comments from `unit_resources` join course_units on
unit_resources.unit_id=course_units.id where course_id= "+Request["id"],
myLoginConnection))
{
  SqlDataReader details = myLoginCommand.ExecuteReader();
  details.Read();
  description.Text = details.GetString(0);
  details.Close();
}
```

Στη συνέχεια κάνουμε έναν έλεγχο αν έχουν δημιουργηθεί ομάδες χρηστών και αν όντως υπάρχουν τις συνδέουμε με τον link μας.

```
using (MySqlCommand myLoginCommand = new MySqlCommand("SELECT * from `group` where
course_id= " + Request["id"], myLoginConnection))
{
  SqlDataReader details = myLoginCommand.ExecuteReader();
  if (details.Read())
  {
    HasGroups = true;
  }
}
```

## 6) Ανακοινώσεις

Επιλέγοντας το link με τις ανακοινώσεις στο μενού του μαθήματος μας εμφανίζεται μία λίστα με τις ανακοινώσεις που έχει κάνει ο καθηγητής.



### 43) Λίστα ανακοινώσεων

#### A) Html κώδικας

Στο πρώτο κομμάτι του κώδικα δημιουργούμε ένα κουμπί που πατώντας το μας επιστρέφει στο μενού του μαθήματος.

```
<a href="details.aspx?id=<%=Request["id"] %>" class="btn btn-default">Πίσω  
</a><br /><br />
```

Στη συνέχεια δημιουργούμε μια λίστα με τίτλο «ανακοίνωση» και από κάτω εμφανίζονται οι ανακοινώσεις του μαθήματος σαν links.

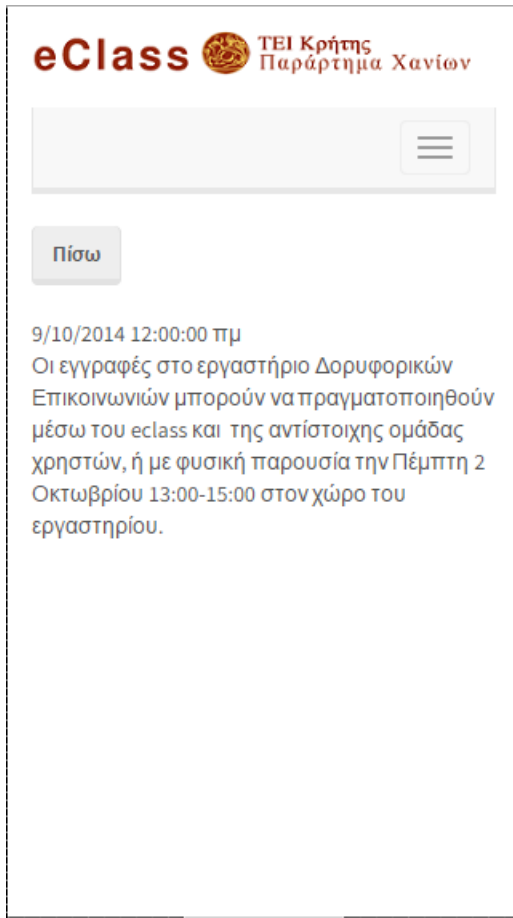
```
<div>
    <asp:GridView runat="server" ID="course" AutoGenerateColumns="false"
    CssClass="table table-striped">
        <Columns>
            <asp:TemplateField HeaderText="Ανακοίνωση"><ItemTemplate><a
            href="Announcement.aspx?id=<%=Eval("id") %>&courseid=<%=Request["id"] %>">
            <%=Eval("title") %></a></ItemTemplate></asp:TemplateField>
        </Columns>
    </asp:GridView>
</div>
</asp:Content>
```

## B) Ο C# κώδικας

Στο C# κώδικα το μόνο που χρειάστηκε να κάνουμε είναι να συνδέσουμε την εφαρμογή μας με την βάση δεδομένων και να πάρει από την βάση μας τις ανακοινώσεις.

```
using (SqlCommand myLoginCommand = new SqlCommand("SELECT * from annonces
where cours_id= "+Request["id"], myLoginConnection))
{
    var reader = myLoginCommand.ExecuteReader();
    course.DataSource = reader;
    course.DataBind();
    reader.Close();
}
```

Αν ο χρήστης επιλέξει κάποιο link από τις ανακοινώσεις τότε θα του εμφανιστεί στο κινητό του η παρακάτω εικόνα.



44) Ανακοίνωση μαθήματος

#### A) Html κώδικας

Όπως φαίνεται και στην εικόνα 44 μας δίνεται η δυνατότητα να επιστρέψουμε στο μενού του μαθήματος αυτό έγινε εφικτό προσθέτοντας τον ακόλουθο κώδικα.

```
<asp:Content runat="server" ContentPlaceHolderID="main">  
    <a href="announcements.aspx?id=<%=Request["courseid"] %>" class="btn btn-  
default">Πίσω </a><br /><br />
```

Στην συνέχεια δημιουργούμε ένα ID το οποίο θα πάρει σαν κείμενο την ανακοίνωση και θα την εμφανίσει στην οθόνη του κινητού τηλεφώνου

```
    <div>  
        <asp:Label runat="server" ID="date"></asp:Label><br />  
        <asp:Literal ID="description" runat="server"></asp:Literal>  
    </div>  
</asp:Content>
```



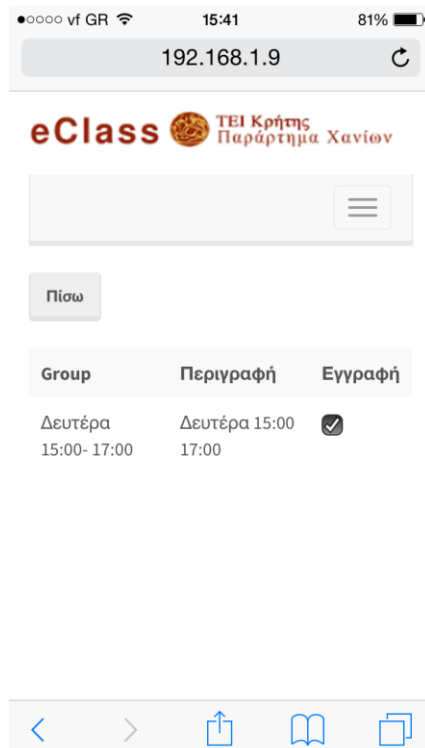
## B) Ο C# κώδικας

Το μόνο που χρειάζεται να κάνουμε για αυτήν την σελίδα είναι να συνδέσουμε την εφαρμογή μας με την βάση δεδομένων να εντοπίσουμε το κείμενο της ανακοίνωσης και να το επιστρέψουμε στο ID ώστε να εμφανιστεί στην οθόνη του κινητού τηλεφώνου. Αυτό γίνεται εφικτό με τον παρακάτω κώδικα.

```
using (MySQLCommand myLoginCommand = new MySQLCommand("SELECT * from annonces
where id= " + Request["id"], myLoginConnection))
{
    IDataReader details = myLoginCommand.ExecuteReader();
    details.Read();
    date.Text = details["temps"].ToString();
    description.Text = details["contenu"].ToString();
    details.Close();
}
```

## 7) Ομάδες χρηστών

Στην σελίδα με τις ομάδες χρηστών, όπως φαίνεται και στην παρακάτω εικόνα, μας δίνονται κάποιες πληροφορίες όπως η ημέρα και η ώρα της ομάδας που είναι εγγεγραμμένος ο χρήστης και επίσης μας δίνεται η δυνατότητα εγγραφής/διαγραφής από την ομάδα.



### 45) Ομάδες χρηστών

## A) Html κώδικας

Πρώτα δημιουργούμε την δυνατότητα στον χρήστη να επιστρέψει στο μενού του μαθήματος.

```
<asp:Content runat="server" ContentPlaceHolderID="main">
    <a href="details.aspx?id=<%=Request["id"] %>" class="btn btn-
default">Πίσω </a><br /><br />
```

Στην συνέχεια δημιουργούμε ένα πίνακα στον οποίο μπορούμε να δούμε την το γκρούπ μας την περιγραφή την οποία έχει δώσει ο καθηγητής του μαθήματος και τέλος την δυνατότητα εγγραφής/διαγραφής στην ομάδα.

```
<asp:GridView runat="server" ID="group" AutoGenerateColumns="false"
CssClass="table table-striped">
    <Columns>
        <asp:TemplateField HeaderText="Group">
            <ItemTemplate> <%=Eval("name") %></ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField
HeaderText="Περιγραφή">
            <ItemTemplate> <%=Eval("description") %></ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Εγγραφή">
            <ItemTemplate>
                <input type="checkbox" <%=IsRegistered((int)
Eval("id"))? "checked": "" %> onclick="register(this, <%=Eval("id") %>)" />
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```

Για να λειτουργήσει αυτό το κουτί και να δώσουμε την δυνατότητα στον χρήστη της εγγραφής και της διαγραφής χρειάστηκε ο παρακάτω JavaScript κώδικας

```
<script>
    function register(checkbox,id) {
        var registered = $(checkbox).is(":checked");
        $.ajax({url:"registergroup.aspx?id="+id+"&action=" + registered});
    }
</script></asp:Content>
```

## B) Ο C# κώδικας

Πρώτα συνδέεται η εφαρμογή και ελέγχει σε ποιες ομάδες χρηστών είναι εγγεγραμμένος ο χρήστης.

```
using (MySQLCommand myLoginCommand = new MySQLCommand("SELECT * FROM
`group_members` where user_id= " + User.Identity.Name, myLoginConnection))
{
    var reader = myLoginCommand.ExecuteReader();
    while (reader.Read())
    {
        mygroups.Add((int)reader["group_id"]);
    }
    reader.Close();
}
```

Στη συνέχεια εμφανίζουμε τις ομάδες χρηστών του μαθήματος.

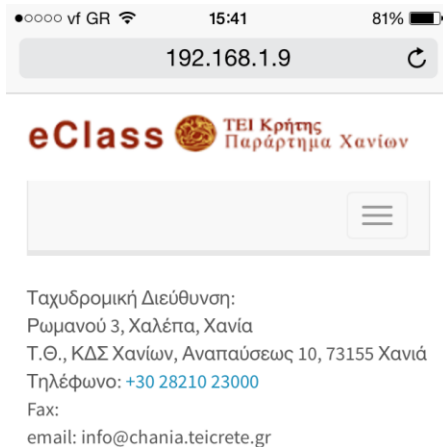
```
using (MySQLCommand myLoginCommand = new MySQLCommand("SELECT * from `group` where
course_id= " + Request["id"], myLoginConnection))
{
    var reader = myLoginCommand.ExecuteReader();
    group.DataSource = reader;
    group.DataBind();
    reader.Close();
}
```

Αν η εφαρμογή μας εντοπίσει πως ο χρήστης είναι γραμμένος σε κάποια από τις ομάδες χρηστών του μαθήματος, τότε θα βάλει ένα tik στο κουτάκι που βρίσκεται στην ομάδα του χρήστη, ώστε να τον ενημερώσει πως εκεί είναι εγγεγραμμένος. Αν πάλι δεν είναι γραμμένος σε καμία ομάδα αυτού του μαθήματος τότε όλα τα κουτάκια θα εμφανιστούν κενά.

```
protected bool IsRegistered(int groupId)
{
    return mygroups.Contains(groupId);
}
```

## 8) Επικοινωνία

Η τελευταία σελίδα του eclass έχει κάποια βασικά στοιχεία επικοινωνίας με την σχολή που ενδεχόμενα να χρειαστεί ο χρήστης.



46) Στοιχεία επικοινωνίας με το ίδρυμα

Για την δημιουργία αυτής της σελίδας χρειάστηκε μόνο ένα αρχείο Html.

Δημιουργήσαμε απλά ένα πλαίσιο το οποίο θα περιέχει τις χρήσιμες πληροφορίες

```
<asp:Content runat="server" ContentPlaceHolderID="main">
  <div>
    Ταχυδρομική Διεύθυνση: <br />
    Ρωμανού 3, Χαλέπα, Χανία <br />
    Τ.Θ., ΚΔΣ Χανίων, Αναπαύσεως 10, 73155 Χανιά <br />
    Τηλέφωνο: +30 28210 23000 <br />
    Fax: <br />
    email: info@chania.teicrete.gr <br />
  </div></asp:Content>
```

## Αναφορές

### AL

- The Art of Computer Programming by Donald Knuth
- Compilers : Principles, Techniques and Tools by Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman
- Begging ASP.NET 4.5 : in C# and VB by Imar Spaanjars
- HTML, CSS, and JavaScript Mobile Development For Dummies by William Harrel
- C# 2010 All-in-One For Dummies by Bill Sempf, Charles Sphar, Stephen R. Davis
- Beginning ASP.NET MVC 4 by Jose Guay Paz
- Codecademy (online lessons)
- W3 coding (online lessons)
- Pluralsight (online lessons)