

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΠΑΤΡΑΣ**

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Αριθμός 972

ΘΕΜΑ: Μελέτη και υλοποίηση προγράμματος υπολογισμού
κοινοχρήστων σε μία πολυκατοικία

ΕΙΣΗΓΗΤΕΣ:

ΠΕΤΡΟΠΟΥΛΟΣ ΔΗΜΗΤΡΗΣ
ΒΛΑΣΟΠΟΥΛΟΣ ΓΕΡΑΣΙΜΟΣ
ΓΙΑΝΟΥΤΣΟΣ ΗΛΙΑΣ

ΣΠΟΥΔΑΣΤΕΣ:

ΠΑΠΑΝΙΚΟΛΑΟΥ ΦΩΤΗΣ Α.Μ 4712
ΑΓΓΕΛΟΠΟΥΛΟΣ ΓΙΩΡΓΟΣ Α.Μ 4683
ΚΑΡΠΟΖΟΛΩΤΑΣ ΓΙΩΡΓΟΣ Α.Μ 4685

ΠΑΤΡΑ 2011

Πρόλογος

Στα πλαίσια της διπλωματικής αυτής εργασίας μελετήσαμε και υλοποιήσαμε ένα πρόγραμμα για την εύρεση των κοινοχρήστων μιας πολυκατοικίας . Όπως θα δούμε και στην συνέχεια αυτής της εργασίας θα μιλήσουμε για τα εργαλεία που χρησιμοποιήσαμε τα.

Τα εργαλεία με τα οποία υλοποιήσαμε την παρούσα διπλωματική εργασία είναι της Microsoft .NET(dot NET) με το οποίο φτιάξαμε τον γραφικό περιβάλλον GUI (graphic user interface) καθώς και την λειτουργικότητα του προγράμματος. Η γλώσσα προγραμματισμού είναι C# (διαβάζεται C sharp) και πρόκειται για μία high level γλώσσα προγραμματισμού. Πολλές μεγάλες εμπορικές εφαρμογές υλοποιούνται με την συγκεκριμένη γλώσσα που επιλέξαμε και εμείς να χρησιμοποιήσουμε.

Ως βάση δεδομένων χρησιμοποιήσαμε την Microsoft SQL Server με την οποία σχεδιάσαμε την βάση δεδομένων φτιάξαμε τους πίνακες που θέλαμε και την χρησιμοποιήσαμε για να αποθηκεύσουμε την βάση δεδομένων. Σε ένα από τα επόμενα κεφάλαια θα μιλήσουμε αναλυτικά για τον τρόπο που εργαστήκαμε για να δημιουργήσουμε την βάση δεδομένων .

Περιεχόμενα

| | |
|---|-----|
| Πρόλογος | 2 |
| Κεφάλαιο 1 ^ο :Η γλώσσα προγραμματισμού C# (C Sharp)..... | 5 |
| Εισαγωγή..... | 5 |
| Δημιουργία ενός νέου project | 6 |
| Προβολή κώδικα | 13 |
| Προσθήκη λειτουργικότητας στην εφαρμογή | 19 |
| Αποσφαλμάτωση (Debugging)..... | 23 |
| Αποθήκευση του project μας..... | 25 |
| Κεφάλαιο 2 ^ο : Γενικά περί MS SQL Server | 28 |
| Δημιουργία πινάκων..... | 29 |
| Κεφάλαιο 3 ^ο : Περιγραφή του προγράμματος..... | 41 |
| Υπολογισμός κοινοχρήστων..... | 41 |
| Περιήγηση στην εφαρμογή..... | 44 |
| Παράρτημα Α : Κώδικας | 55 |
| frmStart.cs..... | 55 |
| frmStoixeiaDiamerismatos.cs..... | 61 |
| frmStoixeiaPolikatikias.cs..... | 73 |
| frmReportPolikatikias.cs..... | 79 |
| frmReportDiamerismatos.cs..... | 93 |
| frmInsDapanes.cs | 113 |
| frmAddDiamerisma.cs..... | 128 |
| frmAddDapanesDiam.cs | 138 |
| polikatikia.cs..... | 148 |
| ParametropiisiDapanon.cs | 162 |
| InsDapanes.cs | 166 |
| diamerisma.cs..... | 177 |

| | |
|---------------------|-----|
| DapaniTherm.cs..... | 198 |
| Dapanes.cs..... | 211 |
| ActiveForm.cs..... | 215 |

Κεφάλαιο 1° :Η γλώσσα προγραμματισμού C# (C Sharp)

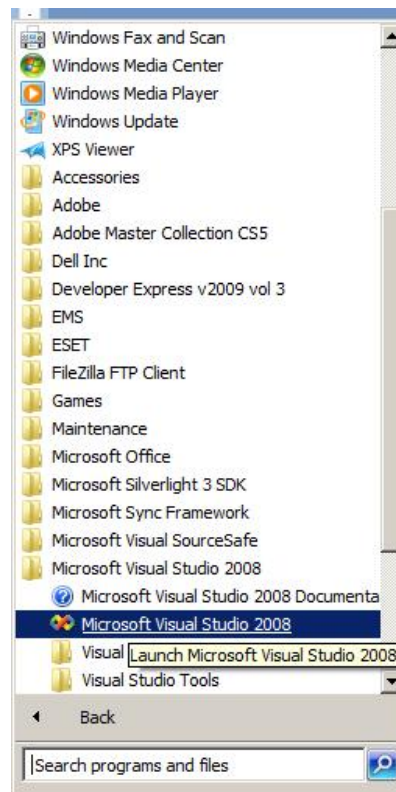
Εισαγωγή

Η C# (προφέρεται "σίι ζαρπ") είναι μια πολυδιάστατη γλώσσα προγραμματισμού που περιλαμβάνει δομημένο προγραμματισμό, συναρτησιακό προγραμματισμό, generic προγραμματισμό, αντικειμενοστραφής προγραμματισμό, και είναι προσανατολισμένο για επαναχρησιμοποιήσιμες βιβλιοθήκες . Αναπτύχθηκε από την Microsoft, στο πλαίσιο της πλατφόρμας του .NET και αργότερα εγκρίθηκε ως πρότυπο του Ecma (ECMA-334) και ISO (ISO/IEC 23270).

Η C# είναι μια απλή , σύγχρονη , γενικής χρήσης , αντικειμενοστρεφή γλώσσα προγραμματισμού . Η ομάδα ανάπτυξης τελεί υπό την ηγεσία του Anders Hejlsberg, ο οποίος είναι ο σχεδιαστής της Borland Turbo Pascal . Η σύνταξη της είναι αντικειμενοστρεφής με βάση την C++ . Η πιο πρόσφατη έκδοση της γλώσσας είναι η 3.0 που κυκλοφόρησε σε συνδυασμό με το .NET Framework 3.5 το 2007.

Δημιουργία ενός νέου project

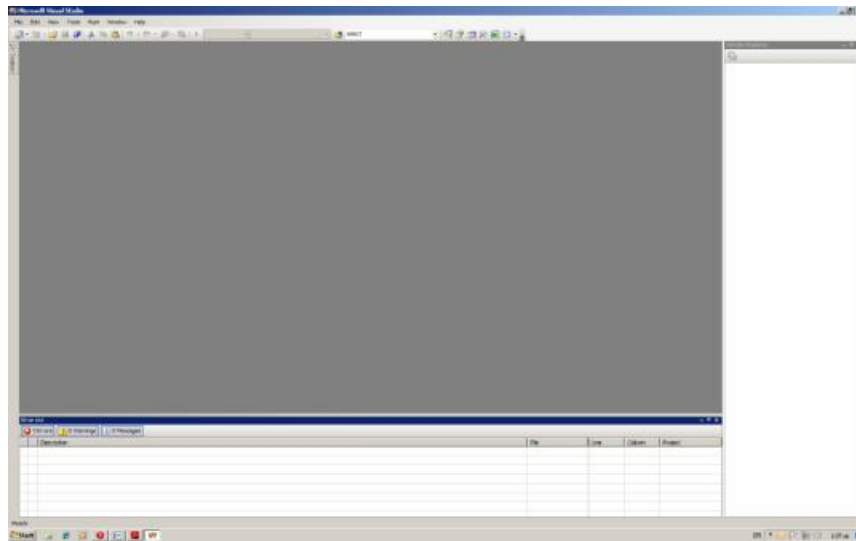
Μπορούμε να ανοίξουμε τη Microsoft Visual C# Express 2008 Express Edition , με το να ανοίξουμε το μενού «Εναρξη», και να βρούμε την ανάλογη συντόμευση



Εικόνα 1: Άνοιγμα του Microsoft Visual studio 2008

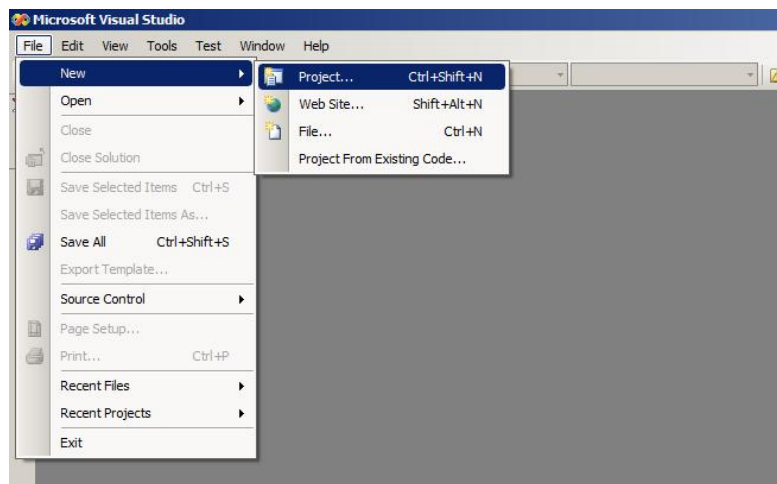
Στη συνέχεια ανοίγει το Microsoft Visual studio 2008 και βλέπουμε το περιβάλλον του

:



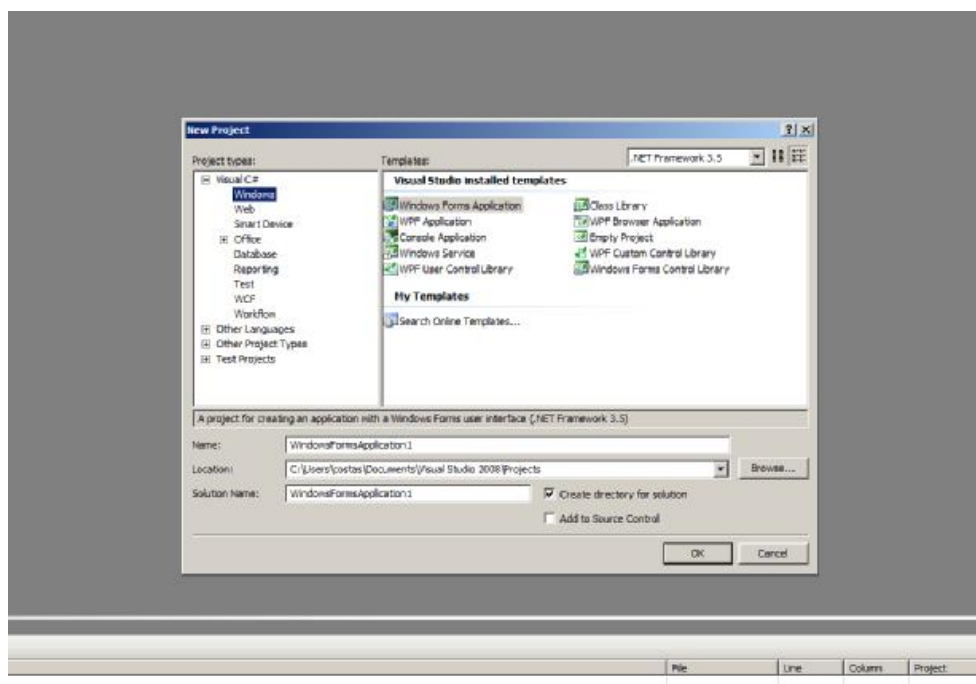
Εικόνα 2 : Αρχική οθόνη Microsoft Visual studio 2008

Για να δημιουργήσουμε ένα καινούργιο Project πατάμε στην πάνω μπάρα «File» «New» «Project»



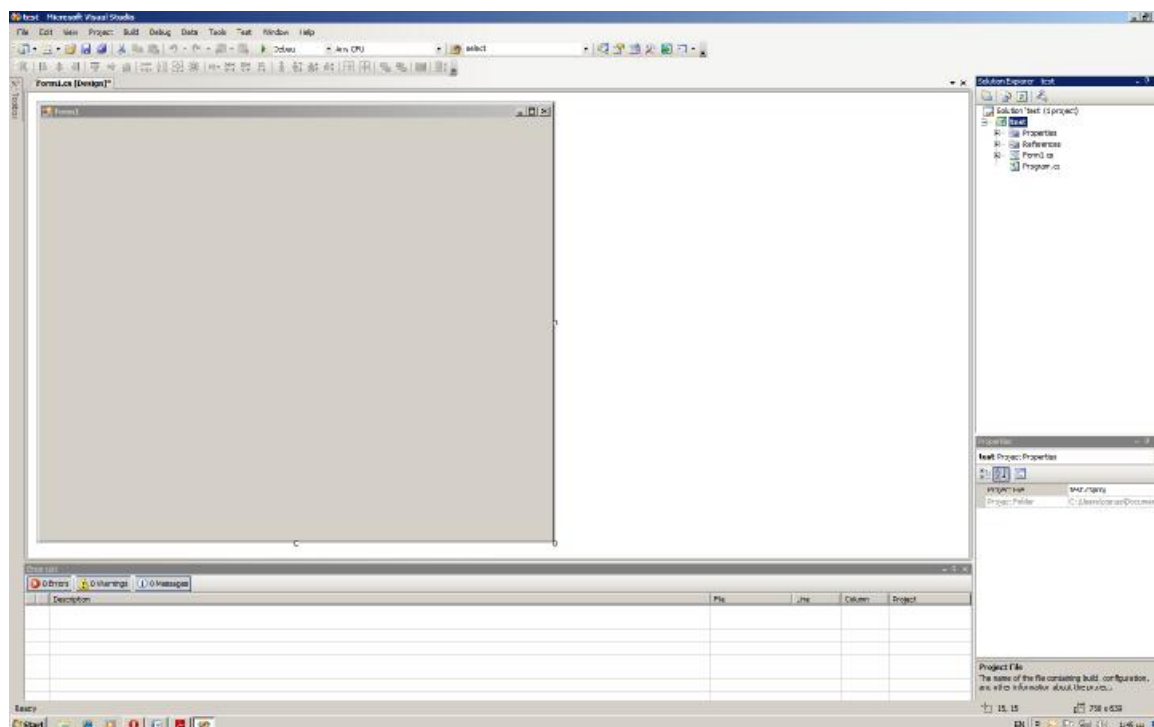
Εικόνα 3 : Πως δημιουργούμε νέο project

Και μας εμφανίζετε η παρακάτω οθόνη :



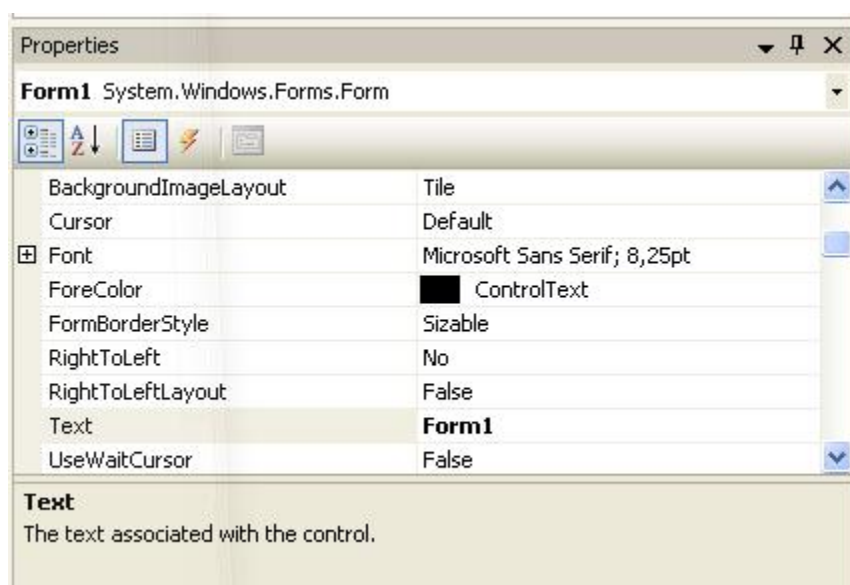
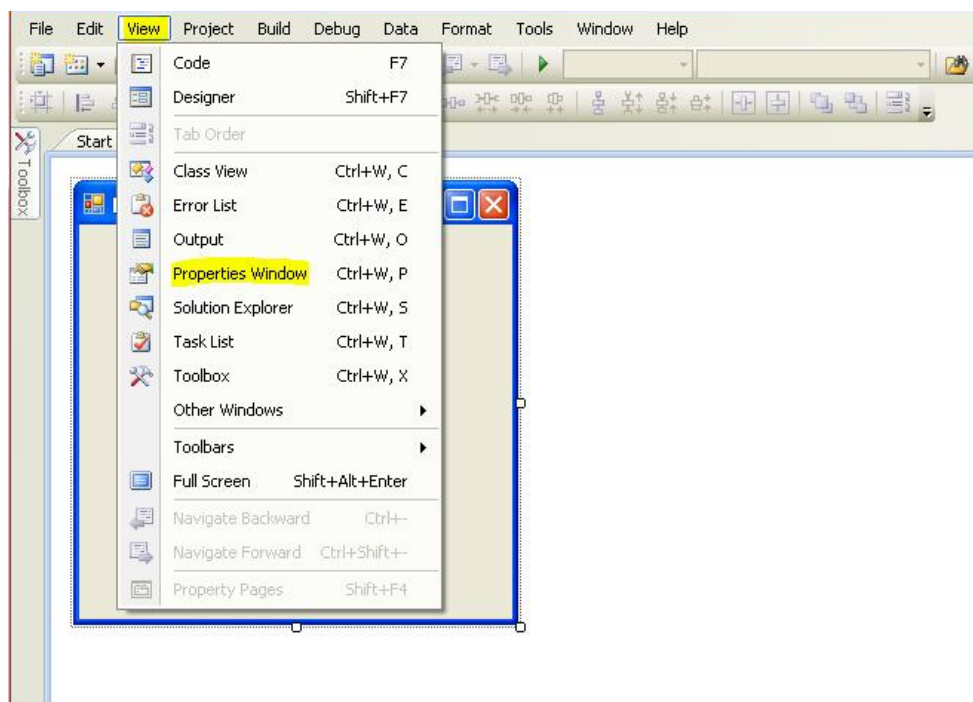
Εικόνα 4 : Επιλογή νέου project με Windows Form

Και μας εμφανίζετε η παρακάτω οθόνη με την οποία μπορείς να ξεκινήσεις να στήνεις το project σου με κουμπιά , πεδία κειμένου και οτιδήποτε άλλο χρειάζεσαι για να ολοκληρώσεις το υποκατασκευή



Έικονά 5^η : Η πρώτη μας εφαρμογή

Ας δοκιμάσουμε να αλλάξουμε κάποιες ιδιότητες του παραθύρου . Για να το κάνουμε αυτό , θα πρέπει να εμφανίσουμε ακόμα ένα μίνι παράθυρο, το παράθυρο **Properties** . Για να το κάνουμε αυτό, είτε πατάμε το πλίκτρο F4 είτε κάνουμε κλικ στο μενού **View** και μετά κλικ στο **Properties Window** .

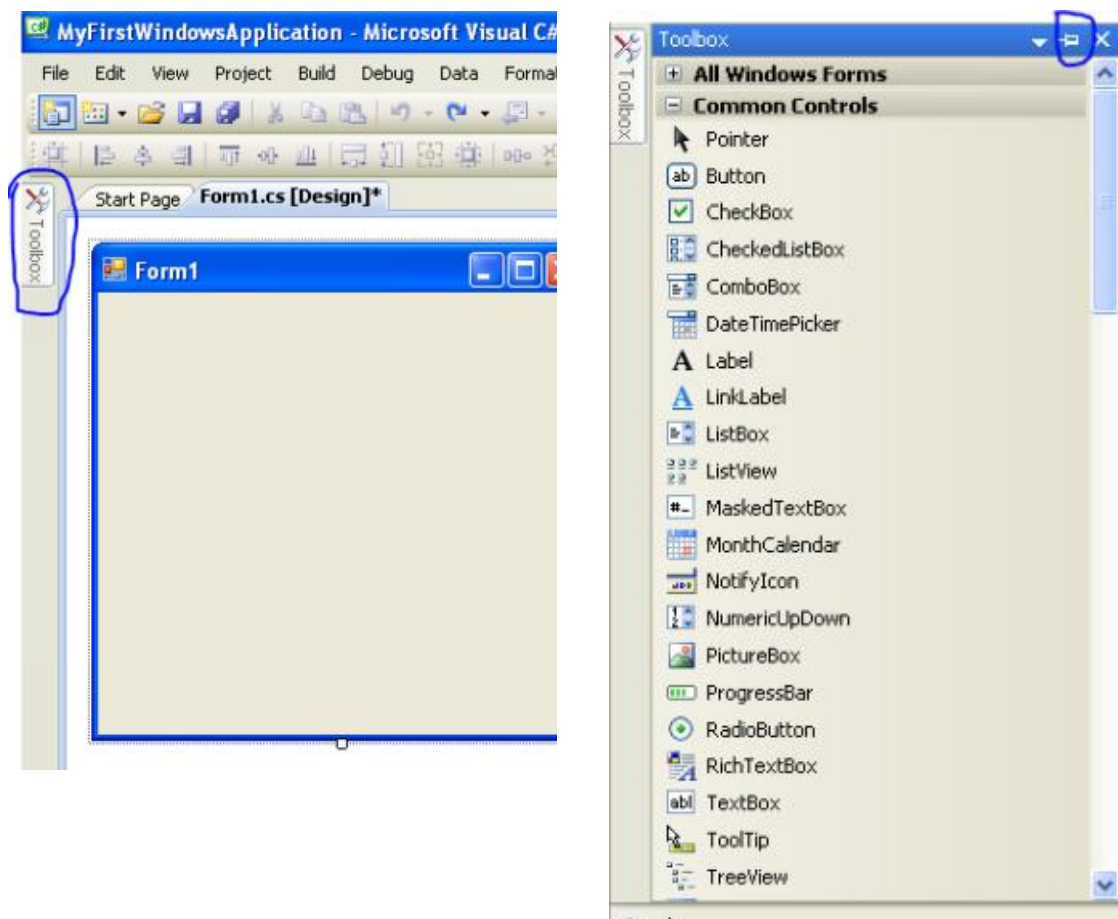


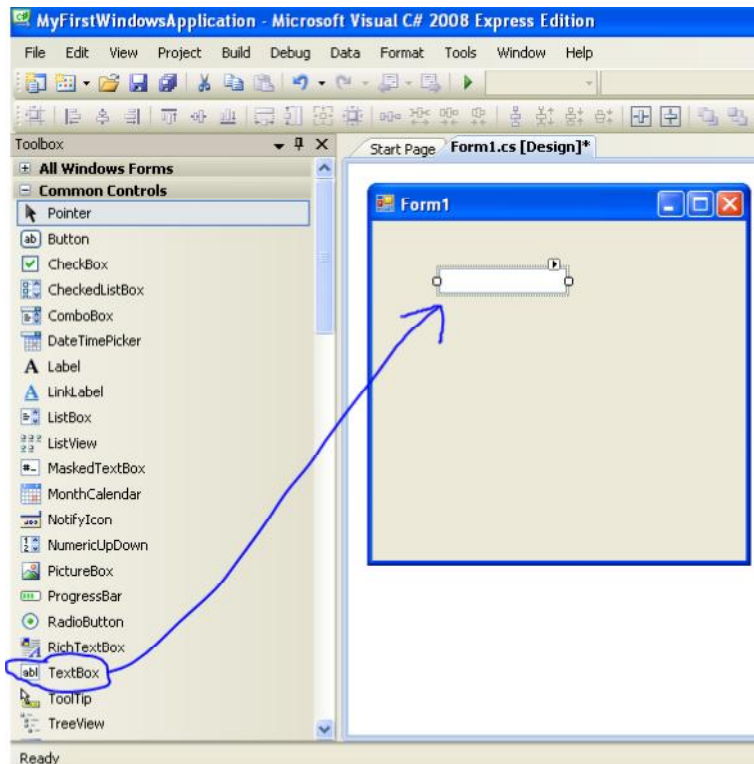
Εικόνα 6^η : Παράθυρο properties

Με αυτό το παράθυρο μπορούμε να επεξεργαστούμε τις ιδιότητες του παραθύρου της εφαρμογής δηλαδή να αλλάξουμε το όνομα της Φόρμας , το χρώμα της , το μέγεθος της και άλλα πολλά πράγματα .

Κάνοντας κλικ στην μπάρα «Toolbox» , στα αριστερά της οθόνης , θα μας εμφανιστούν τα διάφορα controls με τα οποία θα σχεδιάσουμε την φόρμα μας .

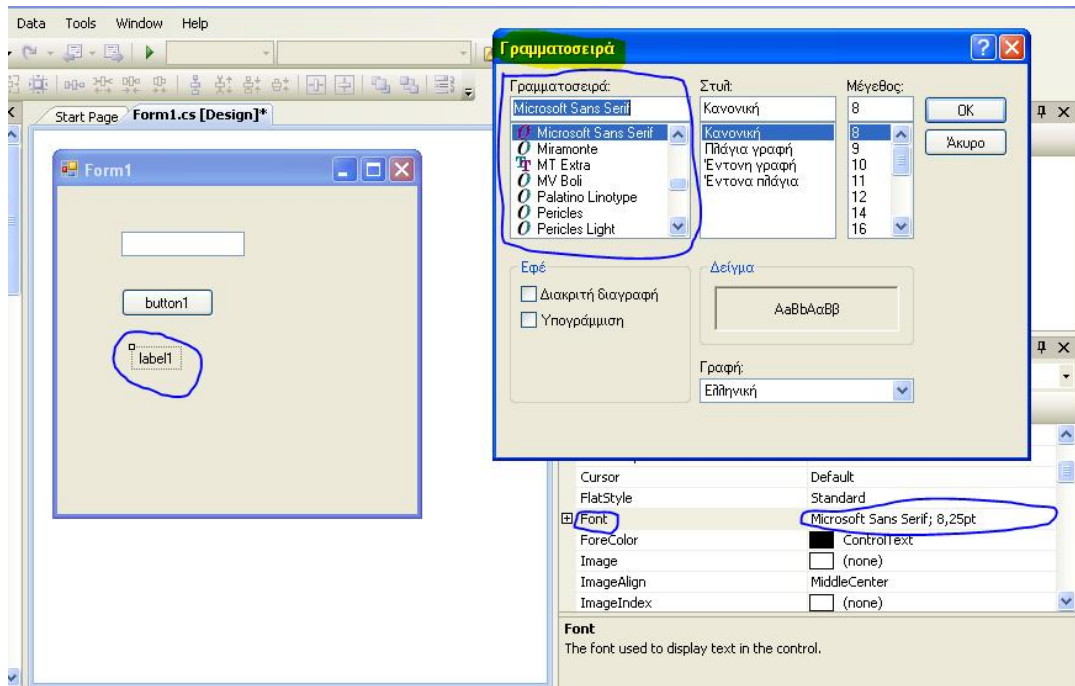
Μπορούμε λοιπόν να βάλουμε μέσα στην φόρμα μας πεδία κειμένου (textbox) , κουμπιά (buttons) και άλλα controls πατώντας το control το οποίο επιθυμούμε και το σέρνουμε στην φόρμα μας βάζοντας το στο σημείο που επιθυμούμε.





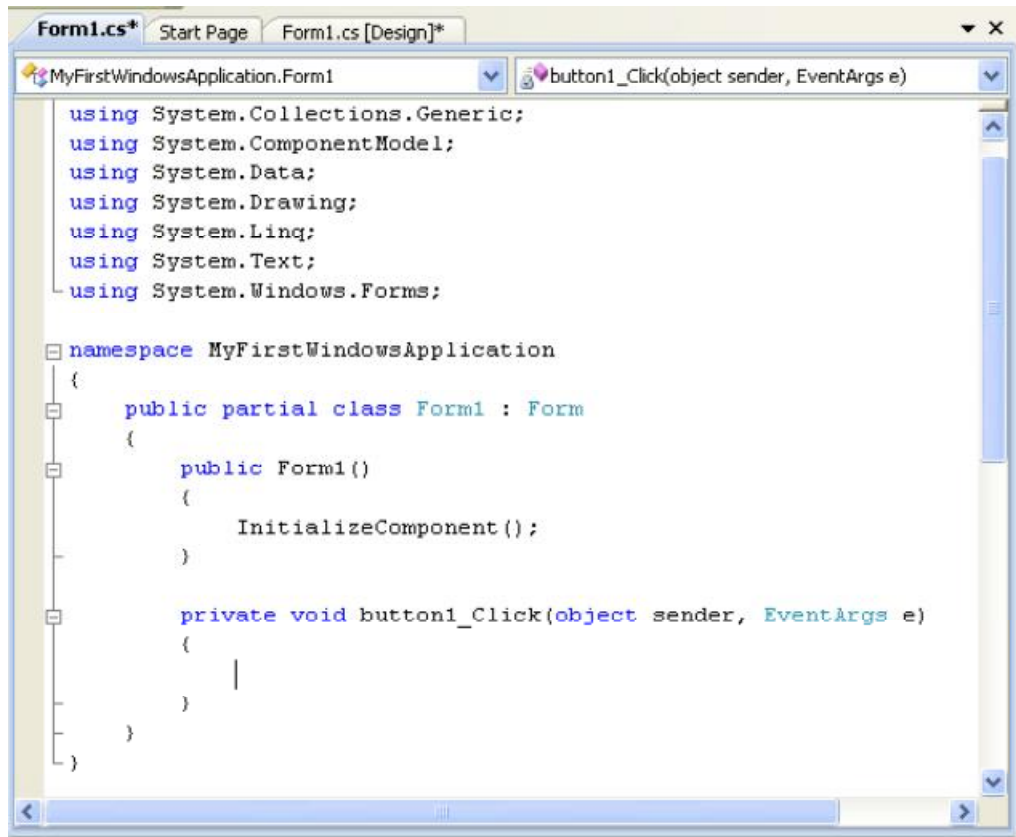
Εικόνα 7^η : Τοποθετώντας ένα control στο παρακάτω Form1.cs

Τα controls μπορούμε να τα επεξεργαστούμε μέσα από τις ιδιότητες που έχουν το κάθε control. Οι ιδιότητες που μπορούμε να αλλάξουμε είναι πάρα πολλά . Ενδεικτικά αναφέρω ότι μπορεί να αλλάξει το μέγεθος της γραμματοσειράς , την γραμματοσειρά , το ύψος , το μήκος , το όνομα του και πολλά άλλα.



Προβολή κώδικα

Για να γράψουμε κάποιο κώδικα για να δώσουν με σειρά από εντολές να εκτελεστούν κάνουμε διπλό κλικ στο κουμπί και μας μεταφέρει στο περιβάλλον όπου θα γράφουμε και μια προς μία της εντολές ώστε να φτιάξουμε το πρόγραμμά μας.



```
Form1.cs* Start Page Form1.cs [Design]*
MyFirstWindowsApplication.Form1 button1_Click(object sender, EventArgs e)
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

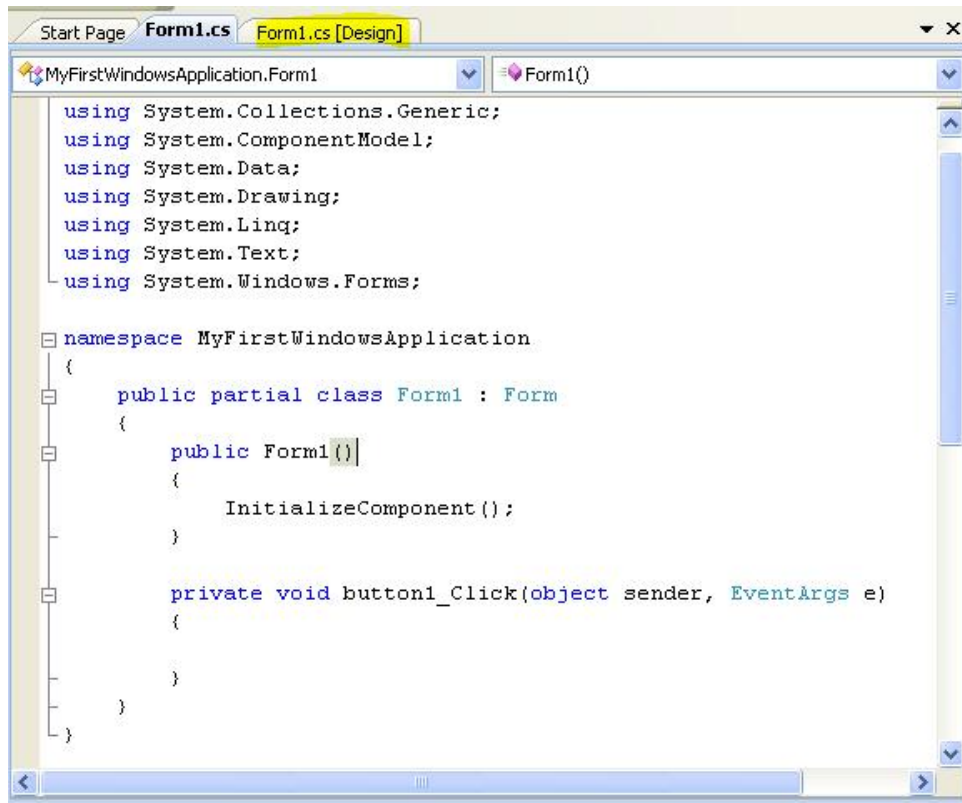
namespace MyFirstWindowsApplication
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            |
        }
    }
}
```

Εικόνα 8^η : Παράθυρο συγγραφής κώδικα

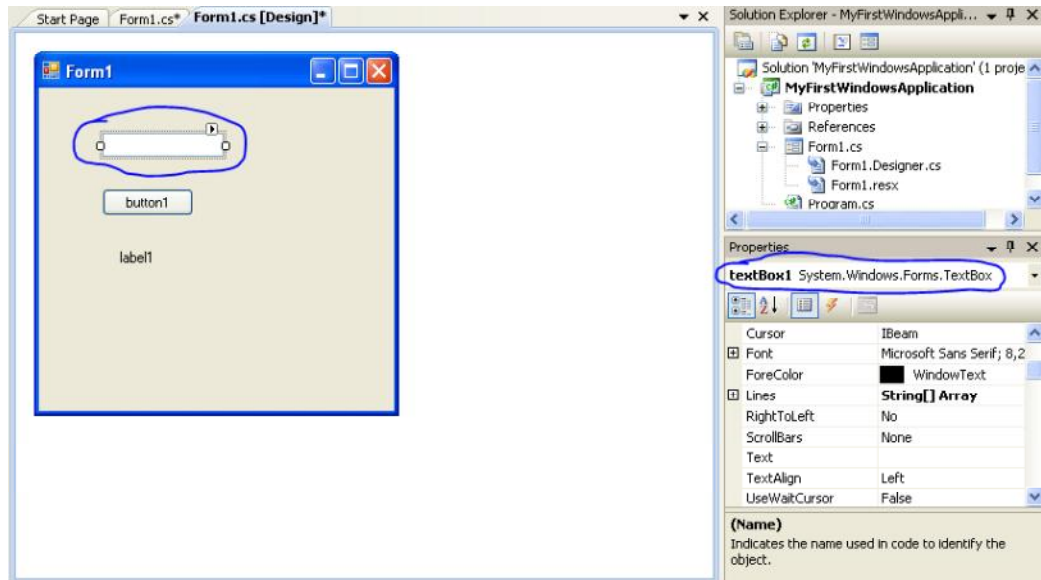
Παρατηρούμε ότι ο κέρσορας έχει τοποθετηθεί σε μία συνάρτηση , με το όνομα **Button1_Click**. Το διπλό κλικ που κάναμε στο Button δημιουργήθηκε την απαραίτητη υποδομή (αυτόματη συγγραφή κώδικα) ώστε η συνάρτηση Button1_Click να εκτελείται, όταν κάνουμε κλικ στο Button. Στη συνάρτηση αυτή, θα γράψουμε τον απαραίτητο κώδικα, ώστε με το κλικ του Button, το κείμενο που περιέχεται στο Textbox να αντιγράφεται στο Label.

Για να το πραγματοποιήσουμε αυτό, θα πρέπει να βρούμε τα **ονόματα (instance names)** του TextBox και του Label. Με τη χρήση του instance name, μπορούμε να αναφερθούμε στο control μέσα από τον κώδικά μας. Επίσης, αξίζει να αναφέρουμε ότι το TextBox και το Label είναι δύο **κλάσεις** του .NET Framework. Με το που τα τοποθετήσαμε πάνω στο παράθυρο Form1, δημιουργήσαμε δύο **instances**, ένα instance της κλάσης TextBox και ένα της κλάσης Label. Για να δούμε τα ονόματα που έχουν αυτά τα instances, κα πρέπει να γυρίσουμε πίσω στην οθόνη σχεδιασμού. Αυτό μπορούμε να το κάνουμε με κλικ στην καρτέλα Form1.cs [Design] πάνω από το παράθυρο συγγραφής κώδικα.



Εικόνα 18: Πώς αλλάζουμε από το παράθυρο συγγραφής κώδικα στο παράθυρο σχεδίασης

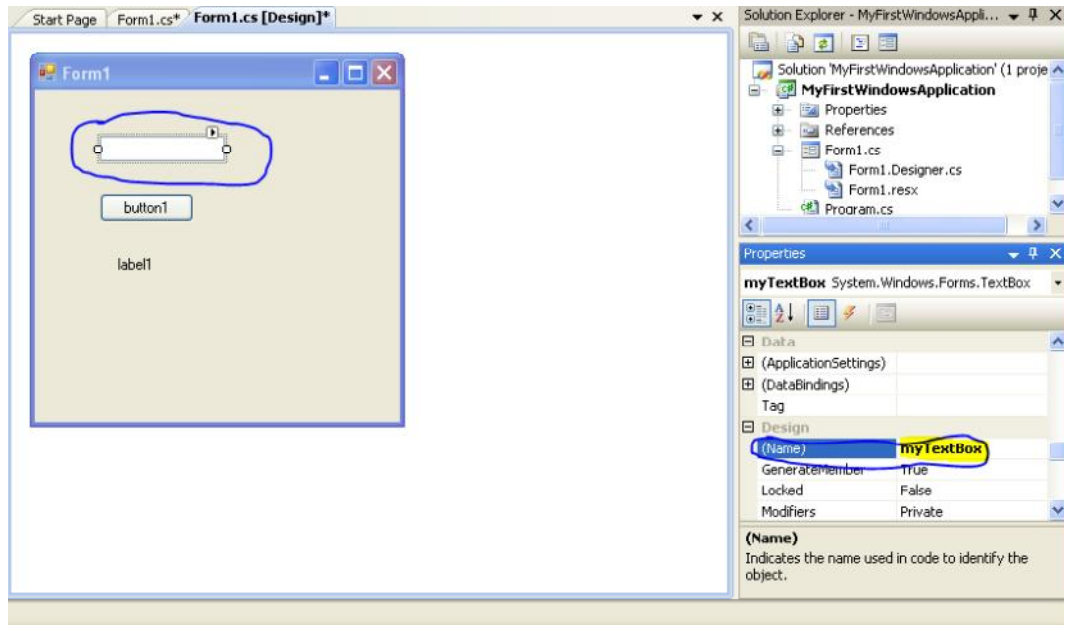
Αφού μεταφερθούμε στο παράθυρο σχεδίασης, για να βρούμε τα instance names του Textbox και του Label, αρκεί να κάνουμε ένα κλικ επάνω τους ώστε να τα επιλέξουμε. Κατόπιν, μπορούμε να δούμε πάνω πάνω στο μίνι παράθυρο Properties το instance name του control.



Εικόνα 9: Βρίσκονται το instance name του TextBox

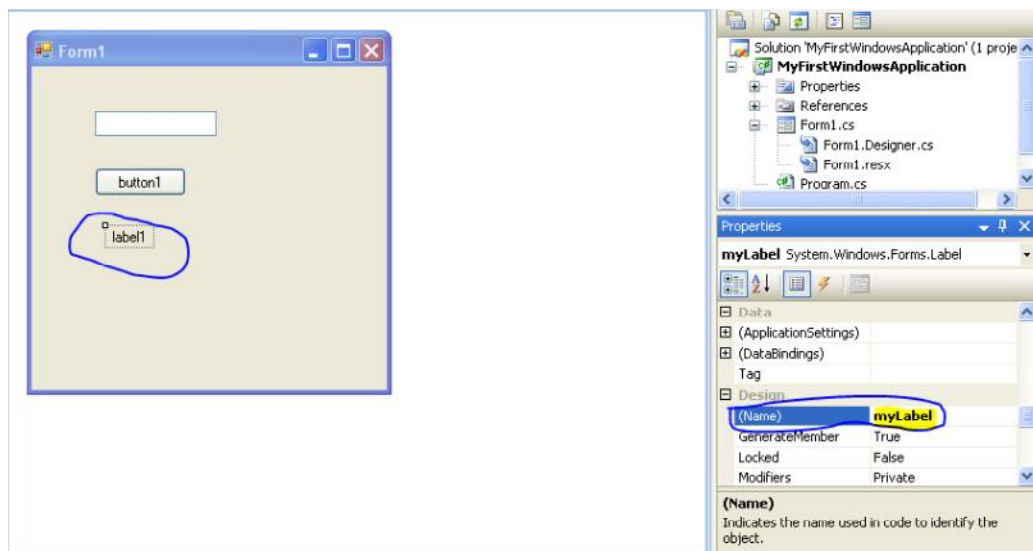
Βλέπουμε ότι το instance name του TextBox είναι **textBox1**. Επίσης, βλέπουμε και την κλάση της οποίας είναι instance το TextBox, και αυτή είναι η **System.Windows.Forms.TextBox**. Εδώ αξίζει να σημειώσουμε ότι το System.Windows.Forms είναι ένα **χώρος ονόματος (namespace)**. Το .NET είναι γεμάτο από τέτοια namespaces, τα οποία έχουν σαφή και περιγραφικά ονόματα με σκοπό να μας διευκολύνουν όταν αναζητούμε μια κλάση.

Αν θέλετε να αλλάξετε το textBox1 σε ένα πιο περιγραφικό όνομα, μπορείτε να μετακινήσετε την scrollbar στο μίνι παράθυρο properties μέχρι να βρείτε την ιδιότητα **(Name)**, και στην οποία μπορείτε να γράψετε το καινούριο instance name για το TextBox. Ας το μετονομάσουμε σε **myTextBox**.



Εικόνα 10: Αλλαγή του instance name του TextBox

Ας κάνουμε το ίδιο και για το Label, μετονομάζοντας το σε **myLabel**.



Εικόνα 11: Αλλαγή του instance name του Label

Εφόσον το κάνουμε και αυτό, μπορούμε να μεταφερθούμε στο παράθυρο συγγραφής κώδικα, κάνοντας κλικ στην καρτέλα Form1.cs, πάνω από το παράθυρο σχεδίασης. Με το που το κάνουμε αυτό, είμαστε έτοιμοι να γράψουμε τον απαραίτητο κώδικα, ώστε με το κλικ στο Button, το κείμενο που έχουμε γράψει στο TextBox να αντιγράφεται στο Label.

Προσθήκη λειτουργικότητας στην εφαρμογή

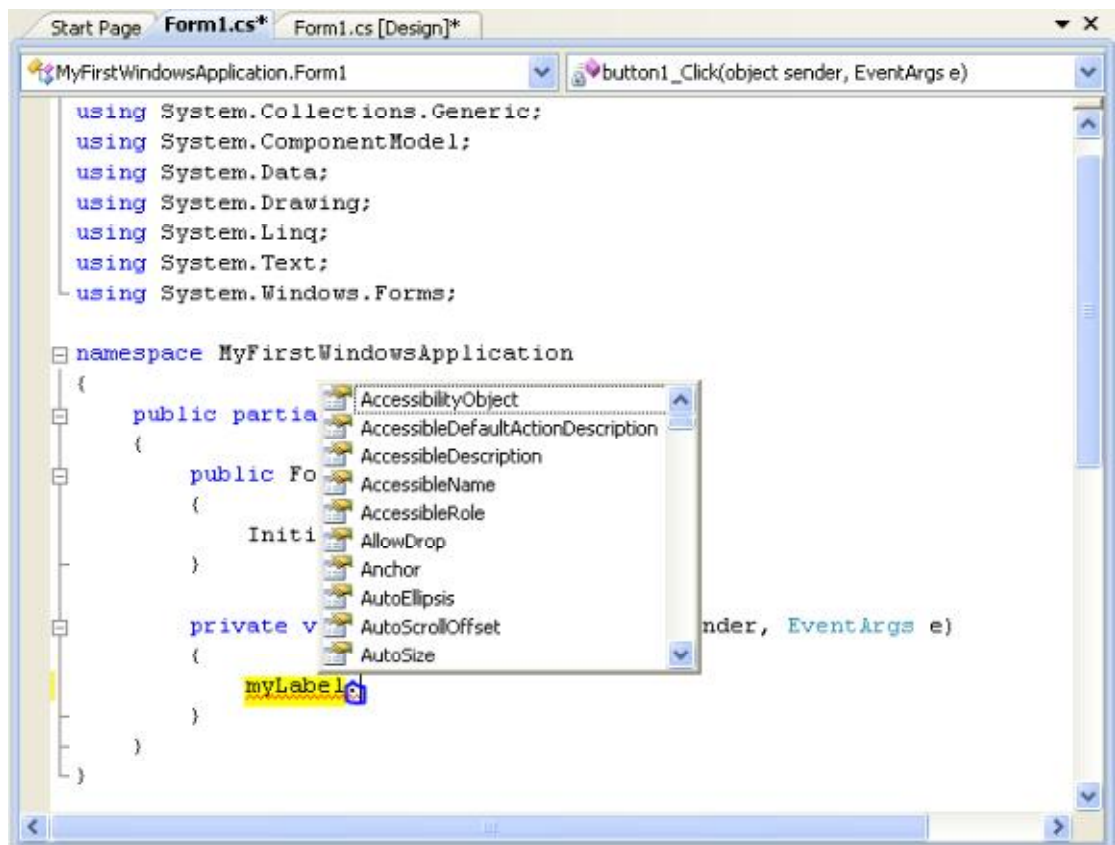
Είπαμε πριν ότι τόσο το TextBox όσο και το Label είναι κλάσεις. Η κάθε κλάση έχει ορισμένες κάποιες ιδιότητες, στις οποίες μπορούμε τόσο να δώσουμε τιμή, όσο και να την λάβουμε. Στην περίπτωση μας, θέλουμε να πάρουμε το κείμενο από το myTextBox, και να το βάλουμε στο κείμενο της myLabel. Πιο συγκεκριμένα, κα πρέπει να πάρουμε το κείμενο από την ιδιότητα **Text** του myTextBox, και να το τοποθετήσουμε στην ιδιότητα **Text** της myLabel. Επίσης, θέλουμε η παραπάνω ανάθεση τιμής να διεξαχθεί όταν κάνουμε κλικ στο Button . Για το σκοπό αυτό, θα γράψουμε τον απαραίτητο κώδικα μέσα στο σώμα της συνάρτησης button1_Click, που όπως αναφέραμε πιο πάνω, εκτελείται όταν κάνουμε κλικ στο Button.

Για να αποκτήσουμε πρόσβαση σε μια ιδιότητα ενός αντικειμένου, γράφουμε το instance name του, μετά τον χαρακτήρα '.' (τελεία),

χωρίς τα εισαγωγικά, και μετά πληκτρολογούμε το όνομα της ιδιότητας.

ΠΡΟΨΟΧΗ: Η C# είναι case sensitive γλώσσα, οπότε αντιμετωπίζει διαφορετικά τα πεζά από τα κεφαλαία γράμματα, δηλαδή το `myTextBox` είναι διαφορετικό από το `mytextbox`. Πάντα πρέπει να πληκτρολογείτε τα ονόματα με προσοχή.

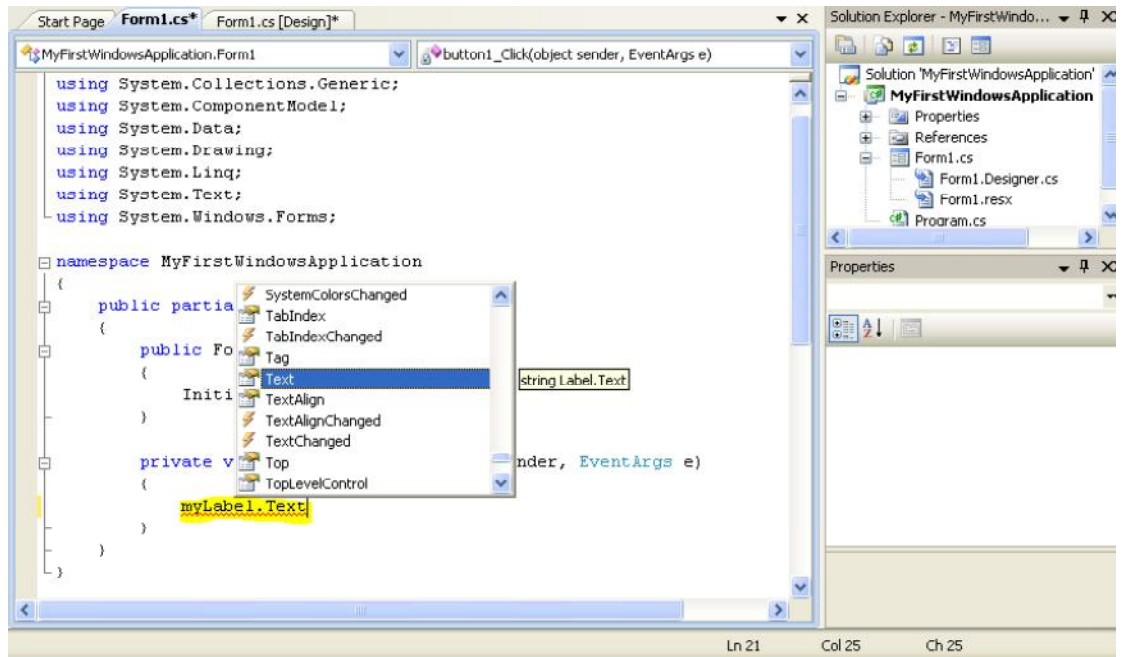
Όταν πληκτρολογήσουμε το όνομα του Label, και μετά πατήσουμε την τελεία, κα δούμε το παρακάτω



Εικόνα 12: Εμφάνιση του παραθύρου Intellisense

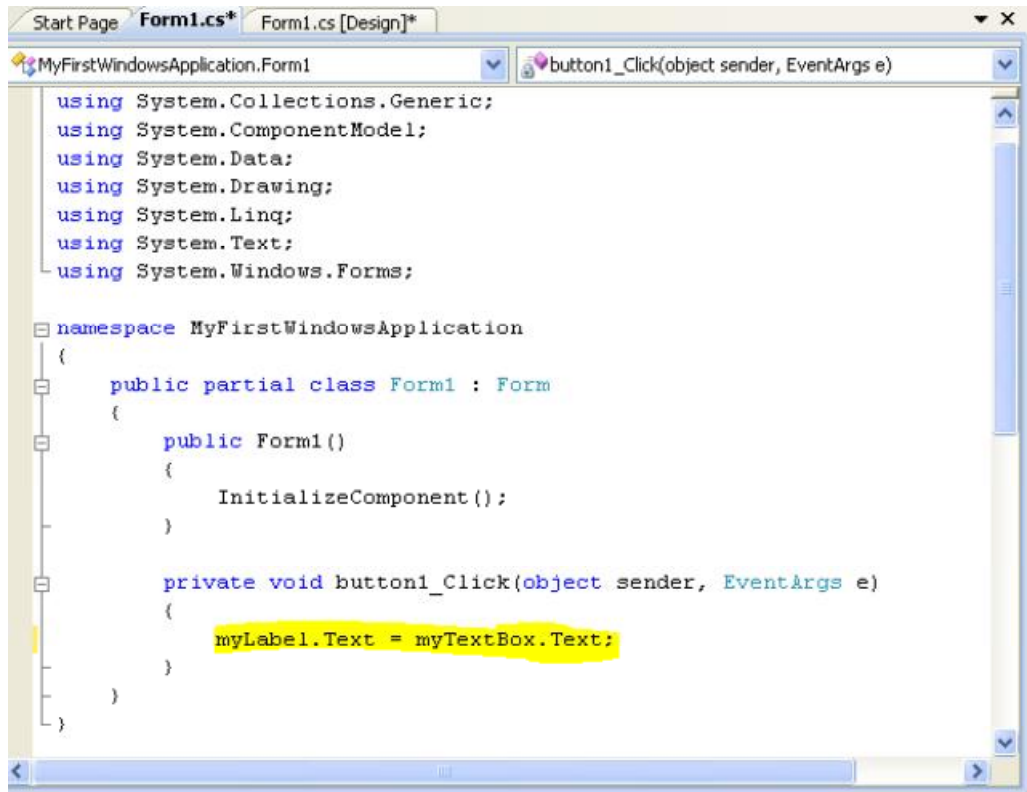
Το μίνι παράθυρο που εμφανίζεται όταν πληκτρολογήσουμε την τελεία, ονομάζεται **Intellisense** . Πρόκειται για μια δυνατότητα της Visual C#, η οποία μας επιτρέπει να δούμε όλα μαζί της συγκεκριμένης κλάσης (ιδιότητες, συναρτήσεις κ.α.). Είναι μια πολύ χρήσιμη δυνατότητα, με τη βοήθεια της οποίας μπορούμε πολύ εύκολα να επιλέξουμε την ιδιότητα η οποία μας ενδιαφέρει – ακόμα κι αν δεν θυμόμαστε το όνομά της – αφού όλα τα μέλη κάθε κλάσης του .NET Framework έχουν χαρακτηριστικά ονόματα.

Επιστρέφοντας στο παράδειγμά μας , αναφέραμε ότι χρειαζόμαστε την ιδιότητα `Text` της `myLabel`. Για να το κάνουμε αυτό, πληκτρολογούμε **Text**, αμέσως μετά την τελεία.



Εικόνα 13: Πρόσβαση στην ιδιότητα Text της myLabel

Φυσικά, αντί να πληκτρολογήσουμε Text, μπορούμε να την επιλέξουμε από το μίνι παράθυρο Intellisense. Στη συνέχεια, επειδή θέλουμε η Text ιδιότητα να **λάβει** τιμή, θα πληκτρολογήσουμε τον χαρακτήρα '=' (ίσον), χωρίς τα εισαγωγικά . Στη συνέχεια, η τιμή που θέλουμε να μεταφερθεί στο myLabel.Text είναι η τιμή (κείμενο) που έχει πληκτρολογήσει ο χρήσης, η οποία και είναι αποθηκευμένη στην ιδιότητα Text του myTextBox . Συνεπώς, για να πάρουμε αυτή την τιμή, θα πληκτρολογήσουμε μετά το '=' το εξής : **myTextBox.Text**. Τέλος , κάθε πρόταση στην C# **πρέπει** να τελειώνει με τον χαρακτήρα ';', χωρίς τα εισαγωγικά, οπότε κα προσθέσουμε και ένα ελληνικό ερωτηματικό. Άρα, ο κώδικας που θα έχουμε γράψει κα είναι ο παρακάτω



```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

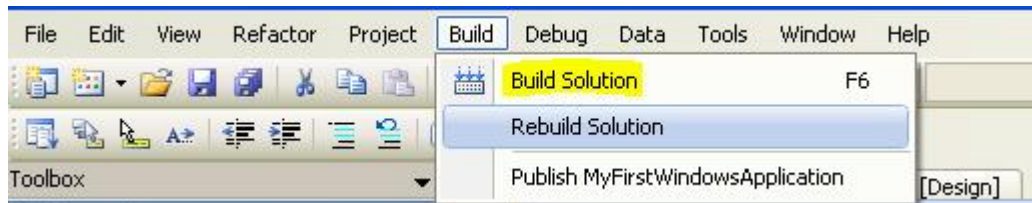
namespace MyFirstWindowsApplication
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            myLabel.Text = myTextBox.Text;
        }
    }
}
```

Εικόνα 14: Ο κώδικας για μεταφορά του Text του myTextBox στο Text του myLabel

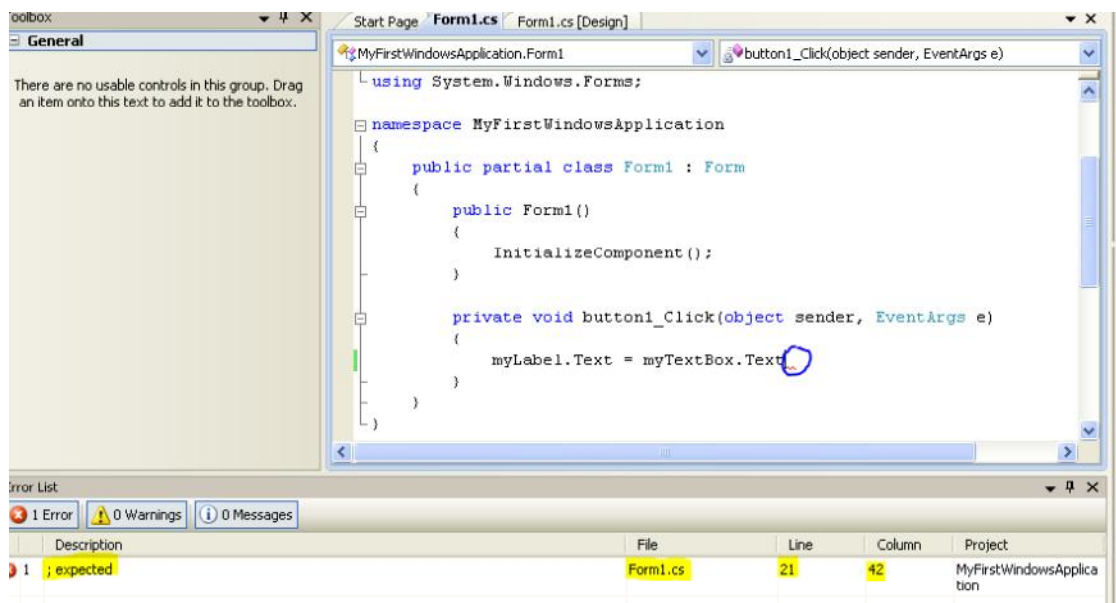
Αποσφαλμάτωση (Debugging)

Για να δείτε αν έχετε κάνει κάποιο λάθος, μπορείτε να κάνετε ένα build το solution σας . Αυτό μπορεί να γίνει είτε με το να πατήσετε το πλήκτρο F6 ή να κάνετε κλικ στο **Build -> Build Solution**.



Εικόνα 15: Μενού Build Solution

Για παράδειγμα, αν έχουμε ξεχάσει το ελληνικό ερωτηματικό και επιλέξουμε να κάνουμε build το solution μας, τότε θα εμφανιστεί το παρακάτω μίνι παράθυρο, με αναφορές στο αρχείο που βρέθηκε το σφάλμα, τη γραμμή στην οποία εμφανίστηκε, καθώς και ένα σύντομο μήνυμα λάθους. Παράλληλα, στο παράθυρο εμφάνισης κώδικα, υπάρχει περίπτωση να εμφανιστεί μια κόκκινη γραμμή, στο σημείο όπου υπάρχει σφάλμα στον κώδικα μας.



Εικόνα 16: Αναφορά λαθών στο solution μας

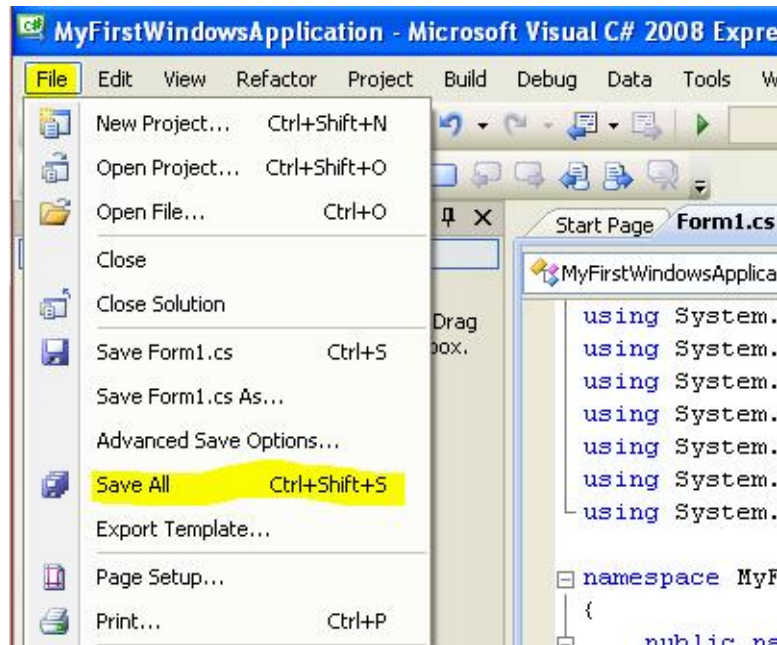
Αφού λοιπόν διορθώσουμε όσα λάθη ανακαλύψουμε στον κώδικά μας, είμαστε έτοιμοι να εκκινήσουμε την εφαρμογή μας. Πατάμε το πλήκτρο F5 ή κάνουμε κλικ στο μενού Debug □Start Debugging και, αν δεν υπάρχουν σφάλματα, η εφαρμογή εμφανίζεται στην οθόνη. Ας δοκιμάσουμε αν δουλεύει ο κώδικας που γράψαμε . Πληκτρολογήστε το κείμενο που επιθυμείτε στο TextBox, και κάντε κλικ στο Button. Αν όλα πήγαν καλά, τότε το κείμενο που έχουμε γράψει στο TextBox κα εμφανιστεί στο Label.



Εικόνα 17: Δοκιμή της εφαρμογής μας

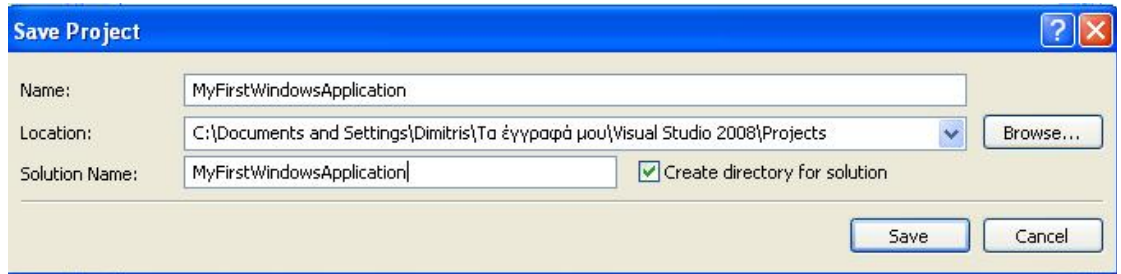
Αποθήκευση του project μας

Αφότου κλείσουμε την εφαρμογή μας, για να αποθηκεύσουμε το project μας, κα πρέπει να κάνουμε κλικ στο μενού File □Save All.



Εικόνα 18: Μενού File->Save All

Αφού το κάνουμε αυτό, θα μας παρουσιαστεί ένα παράθυρο, στο οποίο μπορούμε να δώσουμε ένα όνομα για το project μας, την τοποθεσία στην οποία θέλουμε να αποθηκευτεί καθώς και ένα όνομα για το Solution μας . Επιπροσθέτως , μπορείτε να επιλέξετε αν θέλετε να δημιουργηθεί ξεχωριστό directory για το solution. Με το που δώσετε τιμές στα παραπάνω (ή τα αφήσετε στις προτεινόμενες) , μπορείτε να κάνετε κλικ στο κουμπί **Save**, για να αποθηκεύσετε το project μας .



Εικόνα 19: Αποθήκευση του solution μας

Κεφάλαιο 2^ο : Γενικά περί MS SQL Server

Το Microsoft SQL Server 2008 είναι η πιο εξελιγμένη έκδοση μιας σειράς εργαλείων που στο σύνολο τους αποτελούν αναμφίβολα ένα από τα δημοφιλέστερα συστήματα διαχείρισης βάσεων δεδομένων. Ο SQL Server δεν είναι απλώς μια σχεσιακή βάση δεδομένων αλλά μια ολόκληρη οικογένεια από ξεχωριστά, αλλά συνεργαζόμενα εσωτερικά μεταξύ τους εργαλεία. Τα κυριότερα από αυτά τα εργαλεία είναι η Μηχανή Σχεσιακής Βάσης Δεδομένων (Relational Database Engine), οι Υπηρεσίες Μετασχηματισμού Δεδομένων (Data Transformation Services), οι Υπηρεσίες Ανάλυσης Δεδομένων (Analysis Services), Υπηρεσίες Μεταδεδομένων (Meta Data Services), η δημιουργία αντιτύπων (Replication), και οι ερωταποκρίσεις σε ελεύθερη γλώσσα (English Query).

Το SQL Server ακολουθεί την αρχιτεκτονική του μοντέλου πελάτης/εξυπηρετητής. Ο SQL Server εξυπηρετητής (server) αποτελείται από την βάση δεδομένων – το σύνολο των αποθηκευμένων δεδομένων, συμπεριλαμβανομένων και του ημερολογίου και των αρχείων ελέγχου, τις διάφορες διαδικασίες - συμπεριλαμβανομένων των διαδικασιών του SQL Server και των διαδικασιών των χρηστών συνολικά. Σε ένα σύστημα μπορεί να εγκαταστήσουμε παραπάνω από έναν εξυπηρετητή και η διάκριση μεταξύ τους γίνεται δίνοντας σε κάθε έναν ένα μοναδικό όνομα του στιγμιότυπου του (instance name). Ο πελάτης μπορεί να συνδεθεί σε οποιοδήποτε στιγμιότυπο εξυπηρετητή είναι προσβάσιμο μέσω δικτύου.

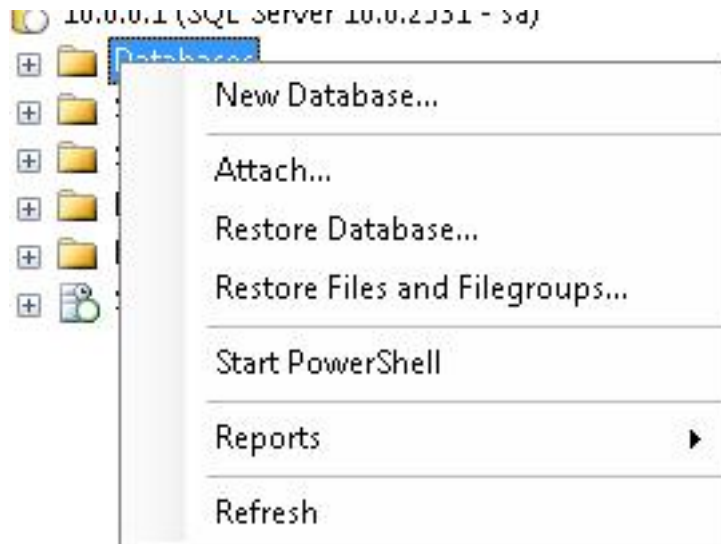
Το Microsoft SQL Server 2008 δεν περιλαμβάνει μόνο μια καλή μηχανή βάσης δεδομένων αλλά και αρκετές εφαρμογές που έχουν ένα εξαιρετικά πλούσιο περιβάλλον χρήσης. Ανάμεσα τους , το Enterprise Manager, που μας επιτρέπει να πραγματοποιήσουμε λειτουργίες διαχείρισης στις βάσεις δεδομένων και το Query Analyzer, ένα χρήσιμο εργαλείο για τους προγραμματιστές.

Κύρια πλεονεκτήματα του Microsoft SQL Server είναι η ταχύτητα, η αντοχή σε μεγάλους όγκους δεδομένων, η εξαιρετική αξιοπιστία, η ασφάλεια δεδομένων και η ευελιξία.

Όλα τα παραπάνω χαρακτηριστικά κάνουν το SQL Server ένα πολύ χρήσιμο εργαλείο για την ανάπτυξη της οντολογίας της εφαρμογής μας . Επιπλέον η ευρεία αποδοχή του εργαλείου αυτού, παρέχει τη δυνατότητα στους χρήστες του μοντέλου να μπουν εύκολα στα εσωτερικά στοιχεία του και να βοηθήσουν στην προσαρμογή του στις εκάστοτε επιχειρηματικές συνθήκες αλλά και στις όποιες μελλοντικές επεκτάσεις του.

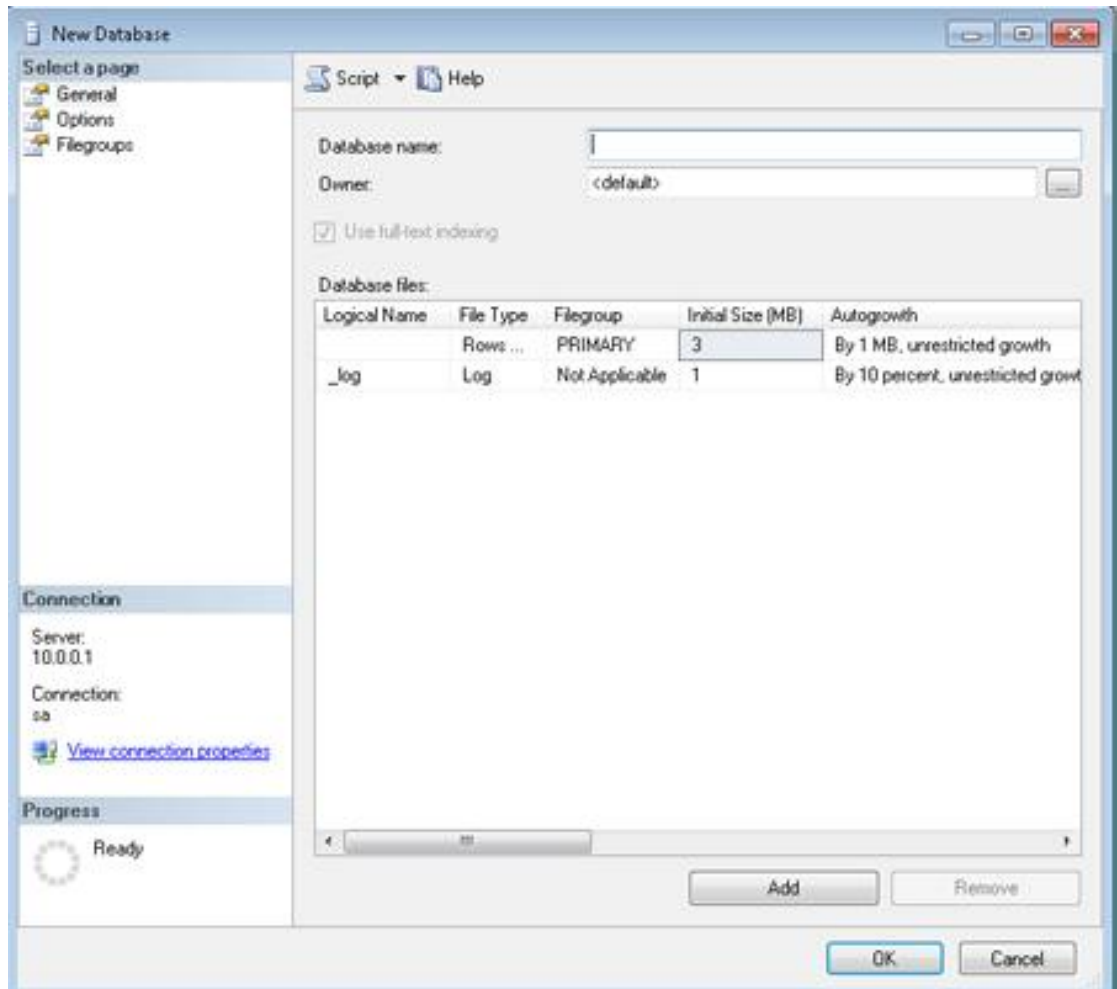
Δημιουργία πινάκων

Σε αυτό το κεφάλαιο θα δούμε πως μπορούμε με το Microsoft SQL Server εύκολα και γρήγορα , μέσα από το γραφικό του περιβάλλον , να δημιουργήσουμε μία βάση δεδομένων και έπειτα να δημιουργήσουμε πίνακες οι οποίοι θα εξυπηρετούν την δομή της βάσης δεδομένων που έχουμε σχεδιάσει για το συγκεκριμένο πρόγραμμα.



Εικόνα 21: Δημιουργία βάσης δεδομένων

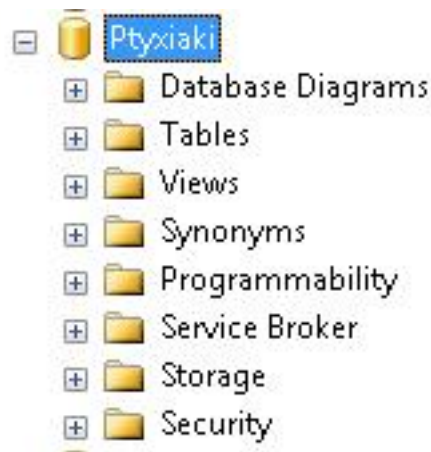
Στο παρακάτω σχήμα βλέπουμε την γραφική απεικόνιση της ονομασίας της βάσης δεδομένων



Εικόνα 22: Όνομα βάσης δεδομένων

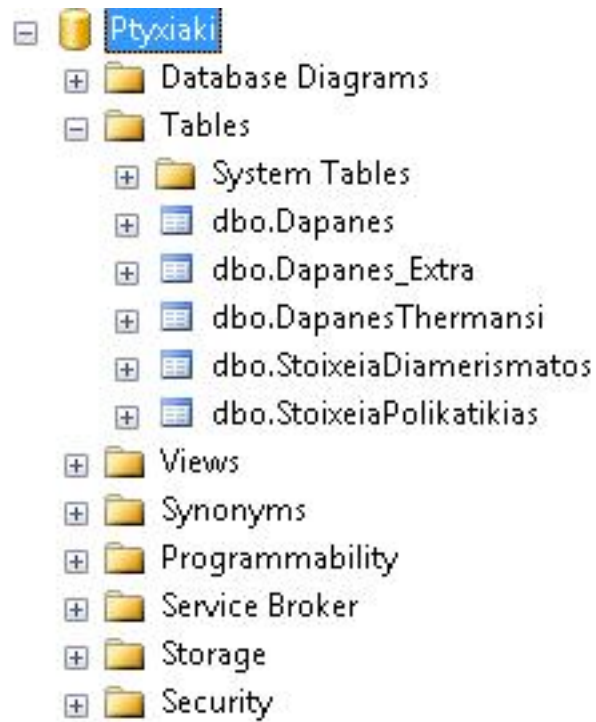
Πατώντας το όνομα λοιπόν της βάσης δεδομένων στο πεδίο Database name και έπειτα πατώντας το κουμπί « OK ». Με αυτό τον τρόπο ολοκληρώνεται η δημιουργία της βάσης δεδομένων . Εμεις για

το συγκεκριμένη υλοποίηση ονομάσαμε την βάση δεδομένων « Ptychiaki » .

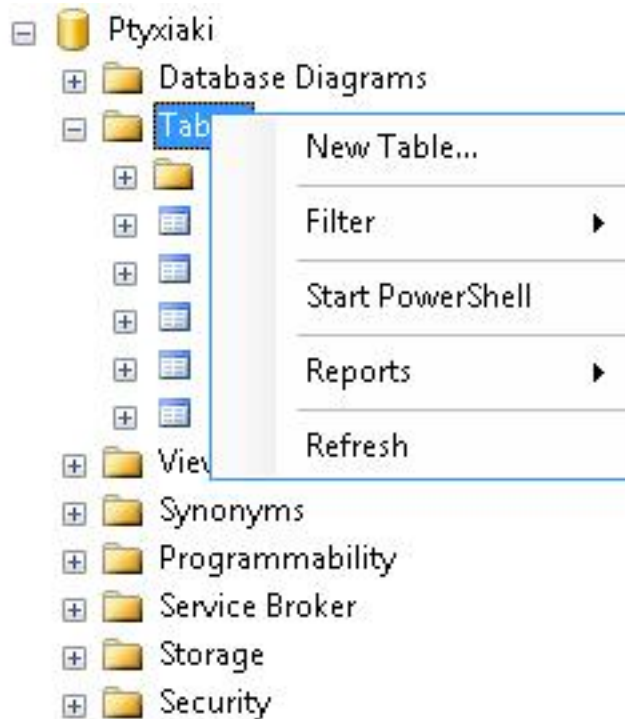


Εικόνα 22 : Δημιουργία της βάσης δεδομένων που εργαστήκαμε

Όπως παρατηρούμε κάτω από την βάση δεδομένων που δημιουργήσαμε « Ptychiaki » υπάρχει ένας φάκελος ο οποίος περιέχει όλους τους πίνακες που θα χρειαστούμε για την υλοποίηση του συγκεκριμένου προγράμματος . Για να δημιουργήσουμε έναν καινούργιο πίνακα κάνουμε δεξί κλικ στον φάκελο Tables και πατάμε New Table .



Εικόνα 23 : Οι πίνακες που χρησιμοποιήσαμε



Εικόνα 24 : Τρόπος για δημιουργία καινούργιο πίνακα

Ο καινούργιος πίνακας δημιουργείται και μας δίνετε η δυνατότητα μέσα απο το γραφικό περιβάλλον του MS SQL SERVER να γράψουμε την δομή του πίνακα κάθε φορά που θέλουμε. Εδώ ορίζουμε πιο θα είναι τα πρωτεύον κλειδί αν θα επιτρέπονται τα πεδία να είναι κένο και άλλες ιδιότητες οι οποίες μας βοηθάνε να φτιάξουμε τον πίνακα έτσι όπως θέλουμε .

Παρακάτω παρατηρούμε την δομή των πινάκων που χρησιμοποιήσαμε στο συγκεκριμένο πρόγραμμα . Οι πίνακες αυτοί είναι :

A) Δαπάνες


- Β) Επιπλέον Δαπάνες
- Γ) Δαπάνες θέρμανσης
- Δ) Στοιχεία Διαμερίσματος
- Ε) Στοιχεία πολυκατικής



αντίστοιχα.





The image shows a screenshot of a database management system window titled "ΠΙΣΤΩΤΗΡ/ΦΥΣΙΚΟΙ - dba.Table_1". The window displays a table structure with the following columns:

| Column Name | Data Type | Allow Nulls |
|-------------|-----------|--------------------------|
| | | <input type="checkbox"/> |

| TECHSRV.Ptyxiaki - dbo.Dapanes | | | |
|---|--------------------|---------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
|  | d | int | <input type="checkbox"/> |
| | Dapanes | varchar(50) | <input checked="" type="checkbox"/> |
| | YpologismosDapanis | bit | <input checked="" type="checkbox"/> |
| | Poso | decimal(8, 2) | <input checked="" type="checkbox"/> |
| | Etos | varchar(10) | <input checked="" type="checkbox"/> |
| | Month | varchar(20) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

| TECHSRV.Ptyxiaki - dbo.Dapanes_Extra | | | |
|---|---|-------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
|  |  | int | <input type="checkbox"/> |
| | Dapanes_Extra | varchar(50) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

| TECHSRV.Ptyxiak...DapanesThermansi | | | |
|---|---|---------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
|  |  | int | <input type="checkbox"/> |
| | HSRV.Ptyxiaki - dbo.DapanesThermansi | | <input checked="" type="checkbox"/> |
| | Ecos_menm | int | <input checked="" type="checkbox"/> |
| | Month_Therm | varchar(20) | <input checked="" type="checkbox"/> |
| | CodeDiamerismatos | varchar(10) | <input checked="" type="checkbox"/> |
| | Metrisi | decimal(8, 2) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

| TECHSRV.Ptyxiaki...eiaDiamerismatos | | | |
|-------------------------------------|-----------------|---------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| ▶ | Orofos | nchar(10) | <input checked="" type="checkbox"/> |
| | Diamerisma | int | <input checked="" type="checkbox"/> |
| | TMDiamerismatos | decimal(8, 2) | <input checked="" type="checkbox"/> |
| | Notes | varchar(50) | <input checked="" type="checkbox"/> |
| | NameIdioktitis | varchar(50) | <input checked="" type="checkbox"/> |
| | PhoneIdioktitis | varchar(50) | <input checked="" type="checkbox"/> |
| | NameEnikiasti | varchar(50) | <input checked="" type="checkbox"/> |
| | PhoneEnikiasti | varchar(50) | <input checked="" type="checkbox"/> |
| | ei | decimal(6, 2) | <input checked="" type="checkbox"/> |
| | fi | decimal(6, 2) | <input checked="" type="checkbox"/> |
| | Ansanser | bit | <input checked="" type="checkbox"/> |
| | Garden | bit | <input checked="" type="checkbox"/> |
| | CloseDiamerisma | bit | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

| TECHSRV.Ptyxiaki ...oixeiaPolikatikias | | | |
|--|----------------------|---------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| ▶ | Address | varchar(50) | <input type="checkbox"/> |
| | Number | nchar(10) | <input checked="" type="checkbox"/> |
| | City | varchar(50) | <input checked="" type="checkbox"/> |
| | TK | int | <input checked="" type="checkbox"/> |
| | Name_admin | varchar(50) | <input checked="" type="checkbox"/> |
| | Phone_admin | varchar(50) | <input checked="" type="checkbox"/> |
| | Orofoi | int | <input checked="" type="checkbox"/> |
| | Number_diamerismatos | int | <input checked="" type="checkbox"/> |
| | Ansanser | bit | <input checked="" type="checkbox"/> |
| | Garden | bit | <input checked="" type="checkbox"/> |
| | Thermansi | bit | <input checked="" type="checkbox"/> |
| | TM_polikatikias | decimal(8, 2) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Εικόνα 25 : Δημιουργία πίνακα και των πεδίων που εμπεριέχονται

Κεφάλαιο 3^ο : Περιγραφή του προγράμματος

Υπολογισμός κοινοχρήστων

Την λειτουργία ενός προγράμματος το οποίο να μπορεί να υπολογίζει τα κοινόχρηστα μίας πολυκατοικίας θα σας παρουσιάσουμε παρακάτω . Η υλοποίηση του συγκεκριμένου προγράμματος έγινε σύμφωνα με την ισχύουσα νομοθεσία του 2009 κατα την οποία ο υπολογισμός των κοινοχρηστων ανα διαμέρισμα γίνεται σύμφωνα με τα χιλιοστά που καταλαμβάνει το κάθε διαμέρισμα . Έτσι λοιπόν αφού καταχωριθούν όλα τα έξοδα της πολυκατοικίας , όπως κοινόχρηστο ρεύμα , κοινόχρηστο λογαριασμό της ΕΥΔΑΠ , έξοδα ανελκυστήρα , διαμόρφωση ακαλύπτου και άλλα, κάνουμε αναγωγή στα χιλιοστά που ανήκει το κάθε διαμέρισμα και έτσι βγάζουμε τον ποσό που θα πρέπει να πληρώσει ο κάθε ιδιοκτήτης διαμερίσματος .

Για τον υπολογισμό του ποσοστού συμμετοχής κάθε διαμερίσματος στο πετρέλαιο εφαρμόσαμε την παρακάτω εξίσωση δαπανών :

$$\Pi(\%) = [fi * \epsilon i + (1 - \Sigma(fi \times \epsilon i)) \times (\epsilon i \times \omega i) / (\epsilon i \times \omega i)] \times 100$$

Όπου :

$\Pi(\%)$: είναι το ποσοστό συμμετοχής του διαμερίσματος στις δαπάνες θέρμανσης

Τα fi και ϵi είναι σταθεροί συντελεστές οι οποίοι βρίσκονται στον πίνακα κατανομής δαπανών που συνέταξε ο μηχανικός για την οικοδομή.

Το ω_i είναι η ένδειξη της ωρομέτρητη για το διαμέρισμα i .

Το f_i ονομάζεται συντελεστής παραμένουσας επιβάρυνσης ιδιοκτησίας και υπολογίζεται απο την σχέση

$$f_i = \omega - (x+y+z)$$

όπου ω για κτίρια με θερμομόνωση σύμφωνα με τον κανονισμό

0.55 για ρετιρέ και διαμερίσματα πάνω από πιλοτές

0.60 για τα υπόγεια και ημιυπόγεια

0.65 για ενδιάμεσα διαμερίσματα

Ενώ για ω χωρίς θερμομόνωση

0.45 για ρετιρέ και διαμερίσματα πάνω από πιλοτές

0.50 για τα υπόγεια και ημιυπόγεια

0.55 για ενδιάμεσα διαμερίσματα

Για τον συντελεστή x

1) $x=0$

Αν οι κεντρικές σωληνώσεις για περισσότερα από τα 2/3 των σωμάτων της ιδιοκτησίας διέρχονται μέσα από την ιδιοκτησία .

2) $x=0.06$

Αν οι κεντρικές σωληνώσεις λιγότερο από το 1/3 των σωμάτων διέρχονται μέσα από την ιδιοκτησία, ή αν οι κεντρικές σωληνώσεις θέρμανσης διέρχονται από κοινόχρηστους χώρους (πλειοψηφία των οικοδομών με σύγχρονη κατασκευή)

$$3) x = 0.03$$

Για τις υπόλοιπες περιπτώσεις

Για τον συντελεστή y

Το y εξαρτάται από το εμβαδόν του διαμερίσματος και είναι :

$$y = 0$$

Για εμβαδόν ίσο ή μεγαλύτερο από 110 m^2

$$y = 0.04$$

Για εμβαδόν από 75 m^2 μέχρι 109.99 m^2

$$y = 0.08$$

Για εμβαδόν από 40 m^2 μέχρι 74.99 m^2

$$y = 0.12$$

Για εμβαδόν μικρότερο από 40 m^2

Για τον συντελεστή z

Το z εξαρτάται από τον παράγοντα $\sigma_f = F_{\varepsilon\xi i} / F_{\text{παρ}i}$ όπου :

$F_{\varepsilon\xi i}$: εξωτερική παράπλευρη επιφάνεια της ιδιοκτησίας ΧΩΡΙΣ την οροφή και το δάπεδο.

$F_{\text{παρ}i}$: συνολική παράπλευρη επιφάνεια της ιδιοκτησίας ΧΩΡΙΣ την οροφή και το δάπεδο.

Αν $\sigma_f \leq 0.20$ τότε $z=0.00$

$0.20 < \sigma_f < 0.35$ τότε $z=0.05$

$0.35 < \sigma_f < 0.50$ τότε $z=0.10$

$0.50 < \sigma_f < 0.65$ τότε $z=0.15$

$\sigma_f > 0.65$ τότε $z=0.20$

Περιήγηση στην εφαρμογή

Η παρακάτω οθόνη μας δίνει την δυνατότητα να καταχωρίσουμε τα στοιχεία της πολυκατοικίας και κάποια χαρακτηριστικά τα οποία είναι χρήσιμα για των υπολογισμό των κοινοχρήστων .

Η δυνατότητα που έχει αυτή η οθόνη μπορεί να κάνει εισαγωγή στοιχείων στην βάση καθώς επίσης και αναβάθμιση των στοιχείων σε περίπτωση που γίνουν αλλαγές.

Πρόγραμμα διαχείρισης κοινοχρήστων - [Καθορισμός Στοιχείων Πολυκατοικίας]

Καθορισμό Στοιχείων Δαπάνες Αναφορές Βοήθεια

Καθορισμός Στοιχείων Πολυκατοικίας

Διεύθυνση :

Αριθμός :

Πόλη :

ΤΚ :

Όνομα διαχειριστή :

Τηλέφωνο Διαχειριστή :

Στοιχεία πολυκατοικίας

Όροφοι

Αριθμός Διαμερισμάτων

Ασανσέρ

Κήπος

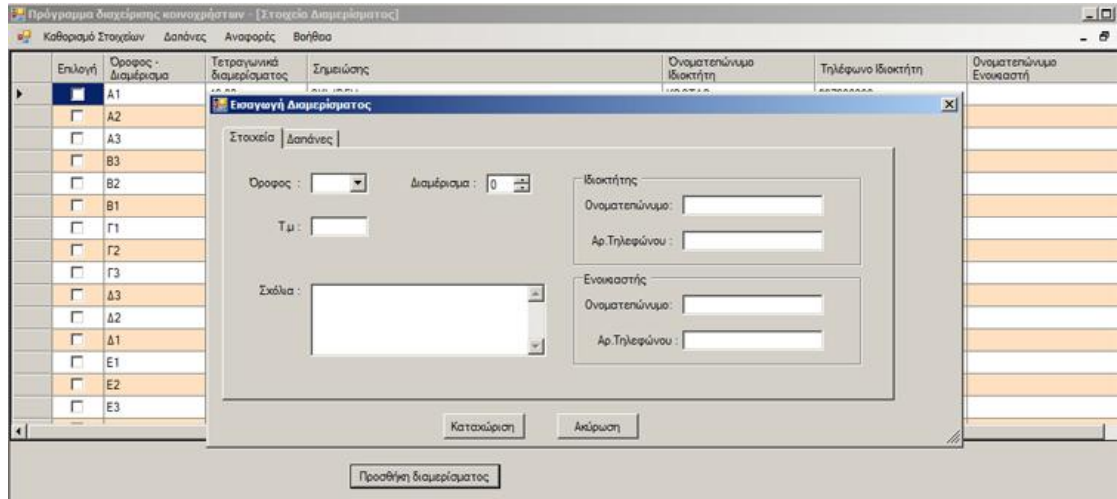
Αυτόνομη θέρμανση

Κεντρική

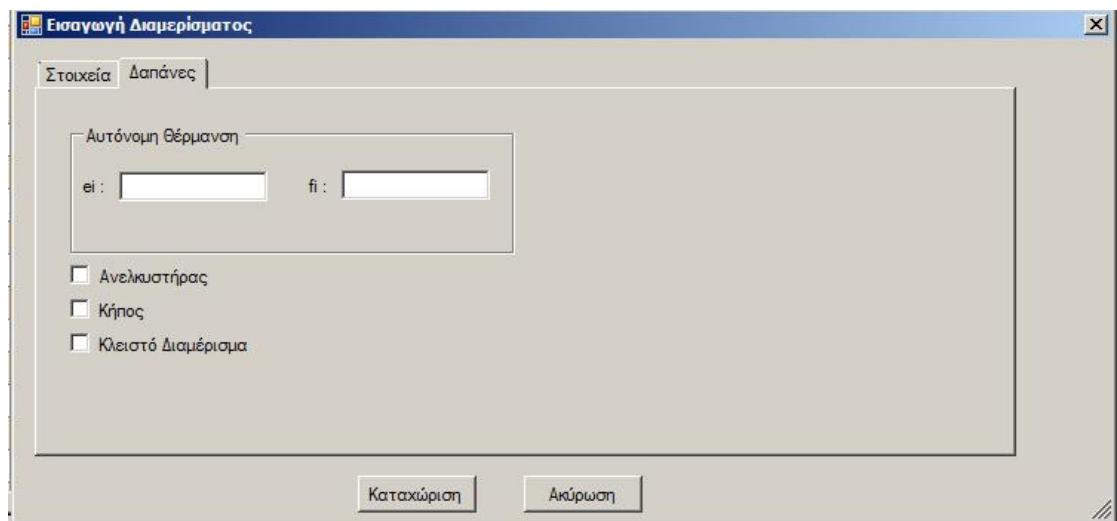
Σύνολο τ.μ. πολυκατοικίας

Εικόνα 26 : Φόρμα εισαγωγής δεδομένων της πολυκατοικίας

Στα παρακάτω γραφήματα βλέπουμε συνολικά τα διαμερίσματα της πολυκατοικίας καθώς επίσης και την καταχώριση όπως και την τροποποίηση των δεδομένων που αφορά στα έξοδα του κάθε διαμερίσματος .



Εικόνα 27 : Εισαγωγή δεδομένων για κάθε διαμέρισμα



Εικόνα 27 : Εισαγωγή δαπανών για κάθε διαμέρισμα

Παρακάτω βλέπουμε μία φόρμα διαχείρισης των δαπανών που έχει η πολυκατοικία όπως (ΔΕΥ , ΕΥΔΑΠ , Ανεγκυστήρα κ.α.)

Πρόγραμμα διαχείρισης κοινοχρήστων - [Δαπάνες Πολυκατοικίας]

Καθορισμό Στοιχείων Δαπάνες Αναφορές Βοήθεια

Επιπλέον δαπάνες :

Έτος : Μήνας :

Δαπάνες :

Υπολογισμός Δαπάνης

ανά χιλιοστά ανά διαμέρισμα Ποσό :

| | Έτος | Μήνα | Δαπάνη | Υπολογισμός Δαπάνης ανα χιλιοστά | Ποσό |
|---|------|------------|--------------|-------------------------------------|---------|
| ▶ | 2009 | Ιούλιος | Δ.Ε.Η. | <input checked="" type="checkbox"/> | 400,00 |
| | 2009 | Ιανουάριος | ΑΝΕΛΚΥΣΤΗΡΑΣ | <input checked="" type="checkbox"/> | 145,00 |
| | 2009 | Ιούλιος | ΕΥΔΑΠ | <input checked="" type="checkbox"/> | 50,00 |
| | 2009 | Ιούλιος | ΠΕΤΡΕΛΑΙΟ | <input checked="" type="checkbox"/> | 2550,00 |
| | 2009 | Ιούλιος | ΚΗΠΟΣ | <input checked="" type="checkbox"/> | 50,00 |
| | 2009 | Ιούλιος | 999999 | <input checked="" type="checkbox"/> | 50,00 |
| | 2009 | Ιούνιος | Δ.Ε.Η. | <input checked="" type="checkbox"/> | 250,00 |
| | 2009 | Ιούνιος | ΑΝΕΛΚΥΣΤΗΡΑΣ | <input checked="" type="checkbox"/> | 50,00 |
| | 2009 | Ιούνιος | ΕΥΔΑΠ | <input checked="" type="checkbox"/> | 150,00 |
| | 2009 | Ιούνιος | ΠΕΤΡΕΛΑΙΟ | <input checked="" type="checkbox"/> | 50,00 |

Εικόνα 28 : Εισαγωγή δαπανών για την πολυκατοικία

Φόρμα συμπλήρωσης των στοιχείων για τον υπολογισμό του ποσοστού συμμετοχής στα έξοδα πετρελαίου .

Πρόγραμμα διαχείρισης κοινοχρήστων - [Δαπάνες διαμερίσματος]

Καθορισμό Στοιχείων Δαπάνες Αναφορές Βοήθεια

Θέρμανση

Έτος : Μήνας :

Κωδικός Διαμερίσματος :

Τωρινή Μέτρηση :

Καταχώριση

| | Επιλογή | Έτος | Μήνας | Διαμέρις | Ποσό |
|---|--------------------------|------|---------|----------|-------|
| ▶ | <input type="checkbox"/> | 2009 | Ιούνιος | ΙΣ1 | 40,00 |
| | <input type="checkbox"/> | 2009 | Ιούλιος | ΙΣ2 | 56,00 |
| | <input type="checkbox"/> | 2009 | Ιούλιος | A1 | 40,00 |
| | <input type="checkbox"/> | 2009 | Ιούνιος | A2 | 42,00 |
| | <input type="checkbox"/> | 2009 | Ιούλιος | A3 | 50,00 |
| | <input type="checkbox"/> | 2009 | Ιούλιος | B1 | 50,00 |
| | <input type="checkbox"/> | 2009 | Ιούλιος | B2 | 50,00 |

Εικόνα 29 : Φόρμα εισαγωγής δαπανών

Παρακάτω απεικονίζουμε πως μπορούμε να ελέγξουμε τα έξοδα ανά διαμέρισμα , έτος και μήνα καθώς επίσης και μαζική ενημέρωση αναφοράς ανά διαμέρισμα . Η εκτύπωση των αναφορών γίνεται σε MS Word ώστε να μπορούν να τυπώνονται και να αποστέλλονται στα διαμερίσματα σαν αναφορά των μηνιαίων εξόδων .

Πρόγραμμα διαχείρισης κοινοχρήστων - [Αναφορές Διαμερίματος]

Καθορισμό Στοιχείων Δαπάνες Αναφορές Βοήθεια

Έτος : [Dropdown]
Μήνας : [Dropdown]
Διαμέρισμα : [Dropdown]

Αναφορά για όλα τα διαμερίσματα

Δημιουργία Αναφοράς

Γενικές Δαπάνες :

Στοιχεία Διαμερισμάτων :

Ωρομετρητής πετρελαίου :

Εικόνα 30 : Φόρμα έκδοση αναφορών για το κάθε διαμέρισμα

Πρόγραμμα διαχείρισης κοινοχρήστων - [Αναφορές πολυκατοικίας]

Καθορισμό Στοιχείων Δαπάνες Αναφορές Βοήθεια

Έτος : Μήνας :

Δημιουργία Αναφοράς

Δαπάνες

Στοιχεία Διαμερισμάτων :

Ωρομετρητής πετρελαίου :

Εικόνα 30 : Φόρμα έκδοση αναφορών για την πολυκατοικία

Παράρτημα Α : Κώδικας

frmStart.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using AppLib;

namespace ptixiaki
{
    public partial class frmStart : Form
    {
        private ActiveForm oActiveForm = new ActiveForm();

        public frmStart()
        {
            InitializeComponent();
        }

        private void CascadeToolStripMenuItem_Click(object sender,
EventArgs e)
        {
```

```

        LayoutMdi(MdiLayout.Cascade);
    }

    private void TileVerticalToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        LayoutMdi(MdiLayout.TileVertical);
    }

    private void TileHorizontalToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        LayoutMdi(MdiLayout.TileHorizontal);
    }

    private void ArrangelconsToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        LayoutMdi(MdiLayout.Arrangelcons);
    }

    private void CloseAllToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        foreach (Form childForm in MdiChildren)
        {
            childForm.Close();
        }
    }

```



```

private void menuStoixeiaPolikatikias_Click(object sender, EventArgs
e)
{
    if (oActiveForm.ChkIfActive("frmStoixeiaPolikatikias"))
    {
        Application.OpenForms["frmStoixeiaPolikatikias"].BringToFront();
    }
    else
    {
        frmStoixeiaPolikatikias frmfrmStoixeiaPolikatikias = new
frmStoixeiaPolikatikias();
        frmfrmStoixeiaPolikatikias.MdiParent = this;
        frmfrmStoixeiaPolikatikias.WindowState =
FormWindowState.Maximized;
        frmfrmStoixeiaPolikatikias.Show();
        this.Refresh();
    }
}

```

```

private void menuStoixeiaDiamerismatos_Click(object sender,
EventArgs e)
{
    if (oActiveForm.ChkIfActive("frmStoixeiaDiamerismatos"))
    {
        Application.OpenForms["frmStoixeiaDiamerismatos"].BringToFront();
    }
    else
    {
        frmStoixeiaDiamerismatos frmfrmStoixeiaDiamerismatos = new
frmStoixeiaDiamerismatos();
    }
}

```

```

        frmfrmStoixeiaDiamerismatos.MdiParent = this;
        frmfrmStoixeiaDiamerismatos.WindowState =
FormWindowState.Maximized;
        frmfrmStoixeiaDiamerismatos.Show();
        this.Refresh();
    }

}

private void menuInsDapanes_Click(object sender, EventArgs e)
{
    if (oActiveForm.ChkIfActive("frmInsDapanes"))
    {
        Application.OpenForms["frmInsDapanes"].BringToFront();
    }
    else
    {
        frmInsDapanes frmfrmInsDapanes = new frmInsDapanes();
        frmfrmInsDapanes.MdiParent = this;
        frmfrmInsDapanes.WindowState = FormWindowState.Maximized;
        frmfrmInsDapanes.Show();
        this.Refresh();
    }
}

private void btnAddDapanesDiamerismatos_Click(object sender,
EventArgs e)
{
    if (oActiveForm.ChkIfActive("frmAddDapanesDiam"))

```

```

    {
        Application.OpenForms["frmAddDapanesDiam"].BringToFront();
    }
    else
    {
        frmAddDapanesDiam frmfrmAddDapanesDiam = new
frmAddDapanesDiam();
        frmfrmAddDapanesDiam.MdiParent = this;
        frmfrmAddDapanesDiam.WindowState =
FormWindowState.Maximized;
        frmfrmAddDapanesDiam.Show();
        this.Refresh();
    }
}

```

```

private void ToolStripMenuItemPolikatikias_Click(object sender,
EventArgs e)
{
    if (oActiveForm.ChkIfActive("frmReportPolikatikias"))
    {
        Application.OpenForms["frmReportPolikatikias"].BringToFront();
    }
    else
    {
        frmReportPolikatikias frmfrmReportPolikatikias = new
frmReportPolikatikias();
        frmfrmReportPolikatikias.MdiParent = this;
        frmfrmReportPolikatikias.WindowState =
FormWindowState.Maximized;
        frmfrmReportPolikatikias.Show();
    }
}

```

```

        this.Refresh();
    }
}

private void ToolStripMenuItemDiamerismatos_Click(object sender,
EventArgs e)
{
    if (oActiveForm.ChkIfActive("frmReportDiamerismatos"))
    {

Application.OpenForms["frmReportDiamerismatos"].BringToFront();
    }
    else
    {
        frmReportDiamerismatos frmfrmReportDiamerismatos = new
frmReportDiamerismatos();
        frmfrmReportDiamerismatos.MdiParent = this;
        frmfrmReportDiamerismatos.WindowState =
FormWindowState.Maximized;
        frmfrmReportDiamerismatos.Show();
        this.Refresh();
    }
}
}
}

```

frmStoixeiaDiamerismatos.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ptixiaki
{
    public partial class frmStoixeiaDiamerismatos : Form
    {
        AppLib.diamerisma DiamerismaCls = new
        AppLib.diamerisma(AppLib.DB_Connection.SQLConnection());

        public frmStoixeiaDiamerismatos()
        {
            InitializeComponent();
        }

        private void btnDiamerismaSubmit_Click(object sender, EventArgs e)
        {
            frmAddDiamerisma frm = new frmAddDiamerisma();
            frm.ShowDialog();

            #region FilldataGridViewDiamerisma
```

```

        this.dataGridViewDiamerisma.DataSource =
DiamerismaCls.DatagridFillDiamerismaList();

        if (this.dataGridViewDiamerisma.ColumnCount != 0)
        {
            this.dataGridViewDiamerisma.RowHeadersVisible = true;
            this.dataGridViewDiamerisma.AutoGenerateColumns = false;
            this.dataGridViewDiamerisma.AllowUserToAddRows = false;

            if (this.dataGridViewDiamerisma.Columns.Contains("ChkSelect"))
            {
                this.dataGridViewDiamerisma.Columns.Remove("ChkSelect");
            }

            DataGridViewCheckBoxColumn chkcolumnEdition = new
DataGridViewCheckBoxColumn();
            {
                chkcolumnEdition.HeaderText = "Επιλογή";
                chkcolumnEdition.Name = "ChkSelect";
                chkcolumnEdition.AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
                chkcolumnEdition.FlatStyle = FlatStyle.Standard;
                chkcolumnEdition.CellTemplate = new
DataGridViewCheckBoxCell();
            }

            this.dataGridViewDiamerisma.Columns.Insert(0,
chkcolumnEdition);
            this.dataGridViewDiamerisma.Columns[0].ReadOnly = false;

```

```
        this.dataGridViewDiamerisma.Columns[1].HeaderText = "Όροφος  
- Διαμέρισμα";
```

```
        this.dataGridViewDiamerisma.Columns[1].ReadOnly = true;
```

```
        this.dataGridViewDiamerisma.Columns[2].Visible = false;//Το  
διαμέρισμα το έχω βάλει από πάνω
```

```
        this.dataGridViewDiamerisma.Columns[3].HeaderText =  
"Τετραγωνικά διαμερίσματα";
```

```
        this.dataGridViewDiamerisma.Columns[3].ReadOnly = true;
```

```
        this.dataGridViewDiamerisma.Columns[3].Width = 100;
```

```
        this.dataGridViewDiamerisma.Columns[4].HeaderText =  
"Σημειώσεις";
```

```
        this.dataGridViewDiamerisma.Columns[4].ReadOnly = true;
```

```
        this.dataGridViewDiamerisma.Columns[4].Width = 350;
```

```
        this.dataGridViewDiamerisma.Columns[5].HeaderText =  
"Όνοματεπώνυμο Ιδιοκτήτη";
```

```
        this.dataGridViewDiamerisma.Columns[5].ReadOnly = true;
```

```
        this.dataGridViewDiamerisma.Columns[5].Width = 150;
```

```
        this.dataGridViewDiamerisma.Columns[6].HeaderText =  
"Τηλέφωνο Ιδιοκτήτη";
```

```
        this.dataGridViewDiamerisma.Columns[6].ReadOnly = true;
```

```
        this.dataGridViewDiamerisma.Columns[6].Width = 150;
```

```
        this.dataGridViewDiamerisma.Columns[7].HeaderText =  
"Όνοματεπώνυμο Ενοικιαστή";
```

```
        this.dataGridViewDiamerisma.Columns[7].ReadOnly = true;
```

```
        this.dataGridViewDiamerisma.Columns[7].Width = 150;
```

```

        this.dataGridViewDiamerisma.Columns[8].HeaderText =
"Τηλέφωνο Ενοικιαστή";
        this.dataGridViewDiamerisma.Columns[8].ReadOnly = true;
        this.dataGridViewDiamerisma.Columns[8].Width = 150;

        this.dataGridViewDiamerisma.Columns[9].HeaderText = "ει";
        this.dataGridViewDiamerisma.Columns[9].ReadOnly = true;
        this.dataGridViewDiamerisma.Columns[9].Width = 30;

        this.dataGridViewDiamerisma.Columns[10].HeaderText = "fi";
        this.dataGridViewDiamerisma.Columns[10].ReadOnly = true;
        this.dataGridViewDiamerisma.Columns[10].Width = 30;

        this.dataGridViewDiamerisma.Columns[11].HeaderText =
"Ανελκυστήρας";
        this.dataGridViewDiamerisma.Columns[11].ReadOnly = true;
        this.dataGridViewDiamerisma.Columns[11].Width = 100;

        this.dataGridViewDiamerisma.Columns[12].HeaderText =
"Κήπος";
        this.dataGridViewDiamerisma.Columns[12].ReadOnly = true;
        this.dataGridViewDiamerisma.Columns[12].Width = 100;

        this.dataGridViewDiamerisma.Columns[13].HeaderText = "Κλειστό
διαμέρισμα";
        this.dataGridViewDiamerisma.Columns[13].ReadOnly = true;
        this.dataGridViewDiamerisma.Columns[13].Width = 100;
    }
#endregion

```



```

        this.Refresh();
    }

    private void frmStoixeiaDiamerismatos_Load(object sender, EventArgs
e)
    {
        #region FilldataGridViewDiamerisma

        this.dataGridViewDiamerisma.DataSource =
DiamerismaCls.DatagridFillDiamerismaList();

        if (this.dataGridViewDiamerisma.ColumnCount != 0)
        {
            this.dataGridViewDiamerisma.RowHeadersVisible = true;
            this.dataGridViewDiamerisma.AutoGenerateColumns = false;
            this.dataGridViewDiamerisma.AllowUserToAddRows = false;

            if (this.dataGridViewDiamerisma.Columns.Contains("ChkSelect"))
            {
                this.dataGridViewDiamerisma.Columns.Remove("ChkSelect");
            }

            DataGridViewCheckBoxColumn chkcolumnEdition = new
DataGridViewCheckBoxColumn();
            {
                chkcolumnEdition.HeaderText = "Επιλογή";
                chkcolumnEdition.Name = "ChkSelect";
                chkcolumnEdition.AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
                chkcolumnEdition.FlatStyle = FlatStyle.Standard;
            }
        }
    }

```

```

        chkcolumnEdition.CellTemplate = new
DataGridViewCheckBoxCell();
    }

    this.dataGridViewDiamerisma.Columns.Insert(0,
chkcolumnEdition);
    this.dataGridViewDiamerisma.Columns[0].ReadOnly = false;

    this.dataGridViewDiamerisma.Columns[1].HeaderText = "Όροφος
- Διαμέρισμα";
    this.dataGridViewDiamerisma.Columns[1].ReadOnly = true;

    this.dataGridViewDiamerisma.Columns[2].Visible = false;//Το
διαμέρισμα το έχω βάλει από πάνω

    this.dataGridViewDiamerisma.Columns[3].HeaderText =
"Τετραγωνικά διαμερίσματα";
    this.dataGridViewDiamerisma.Columns[3].ReadOnly = true;
    this.dataGridViewDiamerisma.Columns[3].Width = 100;

    this.dataGridViewDiamerisma.Columns[4].HeaderText =
"Σημειώσεις";
    this.dataGridViewDiamerisma.Columns[4].ReadOnly = true;
    this.dataGridViewDiamerisma.Columns[4].Width = 350;

    this.dataGridViewDiamerisma.Columns[5].HeaderText =
"Όνοματεπώνυμο Ιδιοκτήτη";
    this.dataGridViewDiamerisma.Columns[5].ReadOnly = true;
    this.dataGridViewDiamerisma.Columns[5].Width = 150;

```

```
        this.dataGridViewDiamerisma.Columns[6].HeaderText =  
"Τηλέφωνο Ιδιοκτήτη";  
        this.dataGridViewDiamerisma.Columns[6].ReadOnly = true;  
        this.dataGridViewDiamerisma.Columns[6].Width = 150;  
  
        this.dataGridViewDiamerisma.Columns[7].HeaderText =  
"Όνοματεπώνυμο Ενοικιαστή";  
        this.dataGridViewDiamerisma.Columns[7].ReadOnly = true;  
        this.dataGridViewDiamerisma.Columns[7].Width = 150;  
  
        this.dataGridViewDiamerisma.Columns[8].HeaderText =  
"Τηλέφωνο Ενοικιαστή";  
        this.dataGridViewDiamerisma.Columns[8].ReadOnly = true;  
        this.dataGridViewDiamerisma.Columns[8].Width = 150;  
  
        this.dataGridViewDiamerisma.Columns[9].HeaderText = "ei";  
        this.dataGridViewDiamerisma.Columns[9].ReadOnly = true;  
        this.dataGridViewDiamerisma.Columns[9].Width = 30;  
  
        this.dataGridViewDiamerisma.Columns[10].HeaderText = "fi";  
        this.dataGridViewDiamerisma.Columns[10].ReadOnly = true;  
        this.dataGridViewDiamerisma.Columns[10].Width = 30;  
  
        this.dataGridViewDiamerisma.Columns[11].HeaderText =  
"Ανελκυστήρας";  
        this.dataGridViewDiamerisma.Columns[11].ReadOnly = true;  
        this.dataGridViewDiamerisma.Columns[11].Width = 100;  
  
        this.dataGridViewDiamerisma.Columns[12].HeaderText =  
"Κήπος";  
        this.dataGridViewDiamerisma.Columns[12].ReadOnly = true;
```

```

        this.dataGridViewDiamerisma.Columns[12].Width = 100;

        this.dataGridViewDiamerisma.Columns[13].HeaderText = "Κλειστό
διαμέρισμα";
        this.dataGridViewDiamerisma.Columns[13].ReadOnly = true;
        this.dataGridViewDiamerisma.Columns[13].Width = 100;
    }
    #endregion
}

private void
dataGridViewDiamerisma_RowHeaderMouseDoubleClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    DiamerismaCls.Orofos =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[1].Value.ToString();
    DiamerismaCls.Diamerisma =
Convert.ToInt16(this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[2].V
alue);
    DiamerismaCls.Tetragonika =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[3].Value.ToString();

    DiamerismaCls.note =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[4].Value.ToString();
    DiamerismaCls.Idioktitis =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[5].Value.ToString();
    DiamerismaCls.phonIdioktitis =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[6].Value.ToString();
    DiamerismaCls.Enokiastis =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[7].Value.ToString();
}

```

```

        DiamerismaCls.phoneEnoikiastis =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[8].Value.ToString();
        DiamerismaCls.ei =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[9].Value.ToString();
        DiamerismaCls.fi =
this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[10].Value.ToString();

        DiamerismaCls.Ansanser =
Convert.ToBoolean(this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[
11].Value);
        DiamerismaCls.Garden =
Convert.ToBoolean(this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[
12].Value);
        DiamerismaCls.Kleistodiamerisma =
Convert.ToBoolean(this.dataGridViewDiamerisma.Rows[e.RowIndex].Cells[
13].Value);

        frmAddDiamerisma frm = new frmAddDiamerisma(DiamerismaCls);
        frm.ShowDialog();

        #region FilldataGridViewDiamerisma

        this.dataGridViewDiamerisma.DataSource =
DiamerismaCls.DatagridFillDiamerismaList();

        if (this.dataGridViewDiamerisma.ColumnCount != 0)
        {
            this.dataGridViewDiamerisma.RowHeadersVisible = true;
            this.dataGridViewDiamerisma.AutoGenerateColumns = false;
            this.dataGridViewDiamerisma.AllowUserToAddRows = false;

```

```

if (this.dataGridViewDiamerisma.Columns.Contains("ChkSelect"))
{
    this.dataGridViewDiamerisma.Columns.Remove("ChkSelect");
}

DataGridViewCheckBoxColumn chkcolumnEdition = new
DataGridViewCheckBoxColumn();
{
    chkcolumnEdition.HeaderText = "Επιλογή";
    chkcolumnEdition.Name = "ChkSelect";
    chkcolumnEdition.AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
    chkcolumnEdition.FlatStyle = FlatStyle.Standard;
    chkcolumnEdition.CellTemplate = new
DataGridViewCheckBoxCell();
}

this.dataGridViewDiamerisma.Columns.Insert(0,
chkcolumnEdition);
this.dataGridViewDiamerisma.Columns[0].ReadOnly = false;

this.dataGridViewDiamerisma.Columns[1].HeaderText = "Όροφος
- Διαμέρισμα";
this.dataGridViewDiamerisma.Columns[1].ReadOnly = true;

this.dataGridViewDiamerisma.Columns[2].Visible = false;//Το
διαμέρισμα το έχω βάλει από πάνω

this.dataGridViewDiamerisma.Columns[3].HeaderText =
"Τετραγωνικά διαμερίσματος";
this.dataGridViewDiamerisma.Columns[3].ReadOnly = true;

```

```
this.dataGridViewDiamerisma.Columns[3].Width = 100;

this.dataGridViewDiamerisma.Columns[4].HeaderText =
"Σημειώσης";
this.dataGridViewDiamerisma.Columns[4].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[4].Width = 350;

this.dataGridViewDiamerisma.Columns[5].HeaderText =
"Όνοματεπώνυμο Ιδιοκτήτη";
this.dataGridViewDiamerisma.Columns[5].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[5].Width = 150;

this.dataGridViewDiamerisma.Columns[6].HeaderText =
"Τηλέφωνο Ιδιοκτήτη";
this.dataGridViewDiamerisma.Columns[6].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[6].Width = 150;

this.dataGridViewDiamerisma.Columns[7].HeaderText =
"Όνοματεπώνυμο Ενοικιαστή";
this.dataGridViewDiamerisma.Columns[7].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[7].Width = 150;

this.dataGridViewDiamerisma.Columns[8].HeaderText =
"Τηλέφωνο Ενοικιαστή";
this.dataGridViewDiamerisma.Columns[8].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[8].Width = 150;

this.dataGridViewDiamerisma.Columns[9].HeaderText = "ει";
this.dataGridViewDiamerisma.Columns[9].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[9].Width = 30;
```

```

this.dataGridViewDiamerisma.Columns[10].HeaderText = "fi";
this.dataGridViewDiamerisma.Columns[10].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[10].Width = 30;

this.dataGridViewDiamerisma.Columns[11].HeaderText =
"Ανελκυστήρας";
this.dataGridViewDiamerisma.Columns[11].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[11].Width = 100;

this.dataGridViewDiamerisma.Columns[12].HeaderText =
"Κήπος";
this.dataGridViewDiamerisma.Columns[12].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[12].Width = 100;

this.dataGridViewDiamerisma.Columns[13].HeaderText = "Κλειστό
διαμέρισμα";
this.dataGridViewDiamerisma.Columns[13].ReadOnly = true;
this.dataGridViewDiamerisma.Columns[13].Width = 100;
}
#endregion

this.Refresh();

}

}
}

```


frmStoixeiaPolikatikias.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using AppLib;
using System.Data.SqlClient;

namespace ptixiaki
{
    public partial class frmStoixeiaPolikatikias : Form
    {
        AppLib.polikatikia polikatikiaCls = new
        polikatikia(AppLib.DB_Connection.SQLConnection());
        bool flagUpdate = false;

        //polikatikia pol = new polikatikia();
        //bool insData = false;

        public frmStoixeiaPolikatikias()
        {
            InitializeComponent();
        }

        private void frmStoixeiaPolikatikias_Load(object sender, EventArgs e)
```

```

{
    polikatikaCls.Check_Stoixeia();
    if (polikatikaCls.Address != null)
    {
        this.txtAddress.Text = polikatikaCls.Address;

        this.txtNumber.Text = polikatikaCls.Number;
        this.txtCity.Text = polikatikaCls.City;
        this.txtTK.Text = polikatikaCls.TK.ToString();
        this.txtDiaxiristis.Text = polikatikaCls.Diaxiristis;
        this.txtDiaxiristisPhone.Text = polikatikaCls.PhoneDiaxiristis;
        this.numorofoi.Value = Convert.ToInt16(polikatikaCls.Orofoi);
        this.numaritmosdiamerismaton.Value =
Convert.ToInt16(polikatikaCls.ArithmosDiamerismaton);
        this.chkasanser.Checked = polikatikaCls.Asanser;
        this.chkkipos.Checked = polikatikaCls.Kipos;
        if (polikatikaCls.autonomithermansi)
        {
            this.rdbautonomithermansi.Checked = true;
        }
        if (!polikatikaCls.autonomithermansi)
        {
            this.rdbkentriki.Checked = true;
        }
        this.txtsinolotm.Text = polikatikaCls.tm.Replace(",",".");

        flagUpdate = true;
        this.btnSubmit.Text = "Διόρθωση";
    }
}

```

```

private void btnSubmit_Click(object sender, EventArgs e)
{
    bool emptyValue = false;

    if (this.txtAddress.Text == "")
    {
        errorProvider.SetError(this.txtAddress, "Δεν έχετε συμπληρώσει
την διεύθυνση");
        emptyValue = true;
    }

    if (this.txtNumber.Text == "")
    {
        errorProvider.SetError(this.txtNumber, "Δεν έχετε συμπληρώσει
την αριθμό");
        emptyValue = true;
    }

    if (this.txtCity.Text == "")
    {
        errorProvider.SetError(this.txtCity, "Δεν έχετε συμπληρώσει την
πόλη");
        emptyValue = true;
    }

    if (this.txtTK.Text == "")
    {
        errorProvider.SetError(this.txtTK, "Δεν έχετε συμπληρώσει την
πόλη");
        emptyValue = true;
    }
}

```

```

    }

    if (this.txtDiaxiristis.Text == "")
    {
        errorProvider.SetError(this.txtDiaxiristis, "Δεν έχετε συμπληρώσει
τον διαχειριστή");
        emptyValue = true;
    }

    if (this.txtDiaxiristisPhone.Text == "")
    {
        errorProvider.SetError(this.txtDiaxiristisPhone, "Δεν έχετε
συμπληρώσει το τηλέφωνο του διαχειριστή");
        emptyValue = true;
    }

    if (this.numorofoi.Value == 0)
    {
        errorProvider.SetError(this.numorofoi, "Δεν έχετε συμπληρώσει
τους ορόφους");
        emptyValue = true;
    }

    if (this.numaritmosdiamerismaton.Value == 0)
    {
        errorProvider.SetError(this.numaritmosdiamerismaton, "Δεν έχετε
συμπληρώσει τα διαμερίσματα");
        emptyValue = true;
    }

    if (!emptyValue)

```

```

{
    if (flagUpdate)
    {
        polikatikiaCls.Address = this.txtAddress.Text.Trim().ToUpper();
        polikatikiaCls.Number = this.txtNumber.Text;
        polikatikiaCls.City = this.txtCity.Text.Trim().ToUpper();
        polikatikiaCls.TK = Convert.ToInt16(this.txtTK.Text);
        polikatikiaCls.Diaxiristis =
this.txtDiaxiristis.Text.Trim().ToUpper();
        polikatikiaCls.PhoneDiaxiristis = this.txtDiaxiristisPhone.Text;
        polikatikiaCls.Orofoi = Convert.ToInt16(this.numorofoi.Value);
        polikatikiaCls.ArithmosDiamerismaton =
Convert.ToInt16(this.numaritmosdiamerismaton.Value);
        polikatikiaCls.Asanser = this.chkasanser.Checked;
        polikatikiaCls.Kipos = this.chkkipos.Checked;
        if (this.rdbautonomithermansi.Checked)
        {
            polikatikiaCls.autonomithermansi = true;
        }
        if (this.rdbkentriki.Checked)
        {
            polikatikiaCls.autonomithermansi = false;
        }
        polikatikiaCls.tm = this.txtsinolotm.Text;

        if (polikatikiaCls.Update())
        {
            MessageBox.Show("Εισαγωγή των στοιχείων της
πολυκατοικίας έγινε με επιτυχία", "Μήνυμα", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
    }
}

```

```

else
{
    MessageBox.Show("Πρόβλημα στη καταχώριση δεδομένων",
"Μήνυμα σφάλματος", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else
{
    polikatikiaCls.Address = this.txtAddress.Text.Trim().ToUpper();
    polikatikiaCls.Number = this.txtNumber.Text;
    polikatikiaCls.City = this.txtCity.Text.Trim().ToUpper();
    polikatikiaCls.TK = Convert.ToInt16(this.txtTK.Text);
    polikatikiaCls.Diaxiristis =
this.txtDiaxiristis.Text.Trim().ToUpper();
    polikatikiaCls.PhoneDiaxiristis = this.txtDiaxiristisPhone.Text;
    polikatikiaCls.Orofoi = Convert.ToInt16(this.numorofoi.Value);
    polikatikiaCls.ArithmosDiamerismaton =
Convert.ToInt16(this.numaritmosdiamerismaton.Value);
    polikatikiaCls.Asanser = this.chkasanser.Checked;
    polikatikiaCls.Kipos = this.chkkipos.Checked;
    if (this.rdbautonomithermanshi.Checked)
    {
        polikatikiaCls.autonomithermanshi = true;
    }
    if (this.rdbkentriki.Checked)
    {
        polikatikiaCls.autonomithermanshi = false;
    }
    polikatikiaCls.tm = this.txtsinolotm.Text;

    if (polikatikiaCls.Insert())

```

```

        {
            MessageBox.Show("Εισαγωγή των στοιχείων της
πολυκατοικίας έγινε με επιτυχία", "Μήνυμα", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("Πρόβλημα στη καταχώριση δεδομένων",
"Μήνυμα σφάλματος", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
}
}
}

```

frmReportPolikatikias.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

```

```

using Word = Microsoft.Office.Interop.Word;
using System.IO;

namespace ptixiaki
{
    public partial class frmReportPolikatikias : Form
    {
        AppLib.InsDapanes DapaniCls = new
AppLib.InsDapanes(AppLib.DB_Connection.SQLConnection());
        AppLib.diamerisma DiamerismaCls = new
AppLib.diamerisma(AppLib.DB_Connection.SQLConnection());
        AppLib.DapaniTherm DapaniThermCls = new
AppLib.DapaniTherm(AppLib.DB_Connection.SQLConnection());
        AppLib.polikatikia PolikatikiaCls = new
AppLib.polikatikia(AppLib.DB_Connection.SQLConnection());

        Word.Application oWord;

        public frmReportPolikatikias()
        {
            InitializeComponent();
        }

        private void frmReportPolikatikias_Load(object sender, EventArgs e)
        {
        }

        private void btnPrintReport_Click(object sender, EventArgs e)
        {
            string TemplateName = "ReportPolikatikias.doc";

```



```

        if ((this.cmbEtos.SelectedIndex == -1) &&
(this.cmbMinas.SelectedIndex == -1))
        {
            MessageBox.Show("Πρέπει να επιλέξετε υποχρεωτικά έτος και
μήνα", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            #region FilldataGridViewDiamesisma

            this.dataGridViewReposrPolikatikias.DataSource =
DapaniCls.DatagridFillDapanesListParam(this.cmbEtos.SelectedItem.ToStri
ng(), this.cmbMinas.SelectedItem.ToString());

            if (this.dataGridViewReposrPolikatikias.ColumnCount != 0)
            {
                this.dataGridViewReposrPolikatikias.RowHeadersVisible = true;
                this.dataGridViewReposrPolikatikias.AutoGenerateColumns =
false;

                this.dataGridViewReposrPolikatikias.AllowUserToAddRows =
false;

                this.dataGridViewReposrPolikatikias.Columns[0].HeaderText =
"id";

                this.dataGridViewReposrPolikatikias.Columns[0].Visible = false;

                this.dataGridViewReposrPolikatikias.Columns[1].HeaderText =
"Έτος";

                this.dataGridViewReposrPolikatikias.Columns[1].ReadOnly =
true;

```

```
this.dataGridViewReposrPolikatikias.Columns[1].Width = 60;

this.dataGridViewReposrPolikatikias.Columns[2].HeaderText =
"Μήνα";
this.dataGridViewReposrPolikatikias.Columns[2].ReadOnly =
true;
this.dataGridViewReposrPolikatikias.Columns[2].Width = 60;

this.dataGridViewReposrPolikatikias.Columns[3].HeaderText =
"Dapani";
this.dataGridViewReposrPolikatikias.Columns[3].Visible = false;

this.dataGridViewReposrPolikatikias.Columns[4].HeaderText =
"Δαπάνη";
this.dataGridViewReposrPolikatikias.Columns[4].ReadOnly =
true;
this.dataGridViewReposrPolikatikias.Columns[4].Width = 80;

this.dataGridViewReposrPolikatikias.Columns[5].HeaderText =
"Υπολογισμός Δαπάνης";
this.dataGridViewReposrPolikatikias.Columns[5].ReadOnly =
true;
this.dataGridViewReposrPolikatikias.Columns[5].Width = 90;

this.dataGridViewReposrPolikatikias.Columns[6].HeaderText =
"Ποσό";
this.dataGridViewReposrPolikatikias.Columns[6].ReadOnly =
true;
this.dataGridViewReposrPolikatikias.Columns[6].Width = 60;
```

```

}
#endregion

#region DatagridFillDapanesThermList

this.dataGridViewPetrelaio.DataSource =
DapaniThermCls.DatagridFillDapanesThermList();

if (this.dataGridViewPetrelaio.ColumnCount != 0)
{
    this.dataGridViewPetrelaio.RowHeadersVisible = true;
    this.dataGridViewPetrelaio.AutoGenerateColumns = false;
    this.dataGridViewPetrelaio.AllowUserToAddRows = false;

    this.dataGridViewPetrelaio.Columns[0].Visible = false;//To
διαμέρισμα το έχω βάλει από πάνω

    this.dataGridViewPetrelaio.Columns[1].HeaderText = "Έτος";
    this.dataGridViewPetrelaio.Columns[1].ReadOnly = true;
    this.dataGridViewPetrelaio.Columns[1].Width = 50;

    this.dataGridViewPetrelaio.Columns[2].HeaderText = "Μήνας";
    this.dataGridViewPetrelaio.Columns[2].ReadOnly = true;
    this.dataGridViewPetrelaio.Columns[2].Width = 150;

    this.dataGridViewPetrelaio.Columns[3].HeaderText =
"Διαμέρισμα index";
    this.dataGridViewPetrelaio.Columns[3].Visible = false;

```

```

        this.dataGridViewPetrelaio.Columns[4].HeaderText =
"Διαμέρισμα";
        this.dataGridViewPetrelaio.Columns[4].ReadOnly = true;
        this.dataGridViewPetrelaio.Columns[4].Width = 50;

        this.dataGridViewPetrelaio.Columns[5].HeaderText = "Ποσό";
        this.dataGridViewPetrelaio.Columns[5].ReadOnly = true;
        this.dataGridViewPetrelaio.Columns[5].Width = 50;

    }
    #endregion

    #region FillDataGridStoixeiaDiamerismatos

        this.dataGridViewStoixeiaDiamerismatos.DataSource =
DiamerismaCls.DatagridFillDiamerismaList();

        if (this.dataGridViewStoixeiaDiamerismatos.ColumnCount != 0)
        {
            this.dataGridViewStoixeiaDiamerismatos.RowHeadersVisible =
true;

            this.dataGridViewStoixeiaDiamerismatos.AutoGenerateColumns = false;
            this.dataGridViewStoixeiaDiamerismatos.AllowUserToAddRows
= false;

            this.dataGridViewStoixeiaDiamerismatos.Columns[0].HeaderText = "Όροφος
- Διαμέρισμα";
            this.dataGridViewStoixeiaDiamerismatos.Columns[0].ReadOnly
= true;

```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[1].Visible =  
false;//Το διαμέρισμα το έχω βάλει από πάνω
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[2].HeaderText =  
"Τετραγωνικά διαμερίσματος";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[2].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[2].Width =  
100;
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[3].HeaderText =  
"Σημειώσης";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[3].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[3].Width =  
350;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[3].Visible =  
false;
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[4].HeaderText =  
"Όνοματεπώνυμο Ιδιοκτήτη";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[4].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[4].Width =  
150;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[5].HeaderText =  
"Τηλέφωνο Ιδιοκτήτη";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[5].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[5].Width =  
150;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[6].HeaderText =  
"Όνοματεπώνυμο Ενοικιαστή";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[6].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[6].Width =  
150;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[6].Visible =  
false;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[7].HeaderText =  
"Τηλέφωνο Ενοικιαστή";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[7].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[7].Width =  
150;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[7].Visible =  
false;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[8].HeaderText = "ει";
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[8].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[8].Width =  
30;
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[9].HeaderText = "fi";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[9].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[9].Width =  
30;
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[10].HeaderText =  
"Ανελκυστήρας";
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[10].ReadOnly = true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[10].Width =  
100;
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[11].HeaderText =  
"Κήπος";
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[11].ReadOnly = true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[11].Width =  
100;
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[12].HeaderText =  
"Κλειστό διαμέρισμα";
```

```

this.dataGridViewStoixeiaDiamerismatos.Columns[12].ReadOnly = true;
    this.dataGridViewStoixeiaDiamerismatos.Columns[12].Width =
100;
    }

#endregion

//if (!File.Exists(Application.StartupPath + "\\" + TemplateName))
//{
//    MessageBox.Show("Δεν υπάρχει η φόρμα της αίτησης !",
"Προσοχή", MessageBoxButtons.OK, MessageBoxIcon.Warning);
//    return;
//}

//object oMissing = System.Reflection.Missing.Value;
//object oTemplate = Application.StartupPath + "\\" +
TemplateName;

//oWord = new Word.Application();

//oWord.Visible = true;

//object ReadOnlyRecommended = true;

//oWord.Documents.Open(ref oTemplate, ref oMissing, ref
ReadOnlyRecommended, ref oMissing, ref oMissing, ref oMissing, ref
oMissing, ref oMissing, ref oMissing, ref oMissing, ref oMissing,
ref oMissing, ref oMissing, ref oMissing, ref oMissing);

```



```

        //oWord.Selection.MoveDown(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveDown(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveDown(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveLeft(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveLeft(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveLeft(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveLeft(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveLeft(ref oMissing, ref oMissing, ref
oMissing);
        //oWord.Selection.MoveLeft(ref oMissing, ref oMissing, ref
oMissing);
        //for (int i = 0; i <
this.dataGridViewReposrPolikatikias.Rows.Count; i++)
        //{
        //
oWord.Selection.TypeText(this.dataGridViewReposrPolikatikias.Rows[i].Cell
s[4].Value.ToString());

        // for (int j = 0; j <
this.dataGridViewStoixeiaDiamerismatos.Rows.Count; j++)
        // {
        //     oWord.Selection.Move(ref oMissing, ref oMissing);
        //     //for (int k = 0; k <
this.dataGridViewStoixeiaDiamerismatos.Rows.Count; k++)

```

```

//    //{
//    // if
(this.dataGridViewReposrPolikatikias.Rows[k].Cells[5].Value.ToString() ==
"ΙΣ1")
//    // {
//    //     oWord.Selection.TypeText("");
//    // }
//    // else if
(this.dataGridViewReposrPolikatikias.Rows[k].Cells[5].Value.ToString() ==
"ΙΣ2")
//    // {

//    // }
//    //}
// }
//}

object number = 1;

if (!File.Exists(Application.StartupPath + "\\ " + TemplateName))
{
    MessageBox.Show("Δεν υπάρχει η φόρμα της αίτησης !",
"Προσοχή", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}

object oMissing = System.Reflection.Missing.Value;
object oTemplate = Application.StartupPath + "\\ " +
TemplateName;

oWord = new Word.Application();

```

```
oWord.Visible = true;
```

```
object ReadOnlyRecommended = true;
```

```
oWord.Documents.Open(ref oTemplate, ref oMissing, ref  
ReadOnlyRecommended, ref oMissing, ref oMissing, ref oMissing, ref  
oMissing, ref oMissing, ref oMissing, ref oMissing, ref oMissing, ref oMissing,  
ref oMissing, ref oMissing, ref oMissing, ref oMissing);
```

```
decimal sinolo = 0;
```

```
oWord.Selection.MoveDown(ref oMissing, ref oMissing, ref  
oMissing);
```

```
for (int i = 0; i < this.dataGridViewReposrPolikatikias.Rows.Count;  
i++)
```

```
{
```

```
oWord.Selection.TypeText(this.dataGridViewReposrPolikatikias.Rows[i].Cell  
s[4].Value.ToString());
```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
```

```
oWord.Selection.TypeText(this.dataGridViewReposrPolikatikias.Rows[i].Cell  
s[1].Value.ToString());
```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
```

```
oWord.Selection.TypeText(this.dataGridViewReposrPolikatikias.Rows[i].Cell  
s[2].Value.ToString());
```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
```

```

oWord.Selection.TypeText(this.dataGridViewReposrPolikatikias.Rows[i].Cells[6].Value.ToString() + " €");
    oWord.Selection.Move(ref oMissing, ref oMissing);
    sinolo +=
Convert.ToDecimal(this.dataGridViewReposrPolikatikias.Rows[i].Cells[6].Value);
    oWord.Selection.InsertRows(ref number);
}
oWord.Selection.TypeText("");
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.TypeText("Σύνολο :");
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.TypeText(sinolo.ToString() + " €");

    object FileName = Application.StartupPath + "\\Save" +
TemplateName + ".doc";
    oWord.ActiveDocument.SaveAs(ref FileName, ref oMissing, ref
oMissing, ref oMissing, ref oMissing, ref oMissing, ref
ReadOnlyRecommended, ref oMissing, ref oMissing, ref oMissing, ref
oMissing, ref oMissing, ref oMissing, ref oMissing, ref oMissing, ref
oMissing);
    oWord.Visible = true;

}
}
}
}

```

frmReportDiamerismatos.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Word = Microsoft.Office.Interop.Word;
using System.IO;

namespace ptixiaki
{
    public partial class frmReportDiamerismatos : Form
    {
        AppLib.InsDapanes DapaniCls = new
AppLib.InsDapanes(AppLib.DB_Connection.SQLConnection());
        AppLib.diamerisma DiamerismaCls = new
AppLib.diamerisma(AppLib.DB_Connection.SQLConnection());
        AppLib.DapaniTherm DapaniThermCls = new
AppLib.DapaniTherm(AppLib.DB_Connection.SQLConnection());
        AppLib.polikatikia PolikatikiaCls = new
AppLib.polikatikia(AppLib.DB_Connection.SQLConnection());
    }
}
```

```

Word.Application oWord;

public frmReportDiamerismatos()
{
    InitializeComponent();
}

private void btnPrintReport_Click(object sender, EventArgs e)
{
    string TemplateName = "ReportDiamerismatos.doc";
    int i_diamer = 0;
    decimal pososto;
    PolikatikiaCls.SelectTM();
    decimal sunoloxilioswn = 0;
    int sunoloDiamerismatwn = 17;
    decimal sunolo = 0;

    sunoloxilioswn = Convert.ToDecimal(PolikatikiaCls.tm);

    if (((this.cmbEtos.SelectedIndex == -1) ||
(this.cmbMinas.SelectedIndex == -1) ||
(this.cmbDiamerismata.SelectedIndex == -1)) &&
(!this.chkAllDiamerismata.Checked))
    {
        MessageBox.Show("Πρέπει να επιλέξετε έτος μήνα και
διαμέρισμα!", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if ((this.chkAllDiamerismata.Checked) &&
((this.cmbEtos.SelectedIndex == -1) || (this.cmbMinas.SelectedIndex == -1)))

```

```

    {
        MessageBox.Show("Πρέπει να επιλέξετε έτος και μήνα!",
"Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        #region DatagridFillDapanesListParam

        this.dataGridViewGenikesDapanes.DataSource =
DapaniCls.DatagridFillDapanesListParam(this.cmbEtos.SelectedItem.ToStri
ng(), this.cmbMinas.SelectedItem.ToString());

        if (this.dataGridViewGenikesDapanes.ColumnCount != 0)
        {
            this.dataGridViewGenikesDapanes.RowHeadersVisible = true;
            this.dataGridViewGenikesDapanes.AutoGenerateColumns =
false;

            this.dataGridViewGenikesDapanes.AllowUserToAddRows =
false;

            this.dataGridViewGenikesDapanes.Columns[0].HeaderText =
"id";

            this.dataGridViewGenikesDapanes.Columns[0].Visible = false;

            this.dataGridViewGenikesDapanes.Columns[1].HeaderText =
"Έτος";

            this.dataGridViewGenikesDapanes.Columns[1].ReadOnly =
true;

            this.dataGridViewGenikesDapanes.Columns[1].Width = 60;

```

```

        this.dataGridViewGenikesDapanes.Columns[2].HeaderText =
"Μήνα";
        this.dataGridViewGenikesDapanes.Columns[2].ReadOnly =
true;
        this.dataGridViewGenikesDapanes.Columns[2].Width = 60;

        this.dataGridViewGenikesDapanes.Columns[3].HeaderText =
"Dapani";
        this.dataGridViewGenikesDapanes.Columns[3].Visible = false;

        this.dataGridViewGenikesDapanes.Columns[4].HeaderText =
"Δαπάνη";
        this.dataGridViewGenikesDapanes.Columns[4].ReadOnly =
true;
        this.dataGridViewGenikesDapanes.Columns[4].Width = 80;

        this.dataGridViewGenikesDapanes.Columns[5].HeaderText =
"Υπολογισμός Δαπάνης ανα χιλιοστά";
        this.dataGridViewGenikesDapanes.Columns[5].ReadOnly =
true;
        this.dataGridViewGenikesDapanes.Columns[5].Width = 200;

        this.dataGridViewGenikesDapanes.Columns[6].HeaderText =
"Ποσό";
        this.dataGridViewGenikesDapanes.Columns[6].ReadOnly =
true;
        this.dataGridViewGenikesDapanes.Columns[6].Width = 60;

    }
#endregion

```



```

#region DatagridFillDapanesThermList

this.dataGridViewPetrelaio.DataSource =
DapaniThermCls.DatagridFillDapanesThermList();

if (this.dataGridViewPetrelaio.ColumnCount != 0)
{
    this.dataGridViewPetrelaio.RowHeadersVisible = true;
    this.dataGridViewPetrelaio.AutoGenerateColumns = false;
    this.dataGridViewPetrelaio.AllowUserToAddRows = false;

    this.dataGridViewPetrelaio.Columns[0].Visible = false;//To
διαμέρισμα το έχω βάλει από πάνω

    this.dataGridViewPetrelaio.Columns[1].HeaderText = "Ετος";
    this.dataGridViewPetrelaio.Columns[1].ReadOnly = true;
    this.dataGridViewPetrelaio.Columns[1].Width = 50;

    this.dataGridViewPetrelaio.Columns[2].HeaderText = "Μήνας";
    this.dataGridViewPetrelaio.Columns[2].ReadOnly = true;
    this.dataGridViewPetrelaio.Columns[2].Width = 150;

    this.dataGridViewPetrelaio.Columns[3].HeaderText =
"Διαμέρισμα index";
    this.dataGridViewPetrelaio.Columns[3].Visible = false;

    this.dataGridViewPetrelaio.Columns[4].HeaderText =
"Διαμέρισμα";
    this.dataGridViewPetrelaio.Columns[4].ReadOnly = true;

```

```

this.dataGridViewPetrelaio.Columns[4].Width = 50;

this.dataGridViewPetrelaio.Columns[5].HeaderText = "Ποσό";
this.dataGridViewPetrelaio.Columns[5].ReadOnly = true;
this.dataGridViewPetrelaio.Columns[5].Width = 50;

}
#endregion

#region FillDataGridStoixeiaDiamerismatos

this.dataGridViewStoixeiaDiamerismatos.DataSource =
DiamerismaCls.DatagridFillDiamerismaList();

if (this.dataGridViewStoixeiaDiamerismatos.ColumnCount != 0)
{
    this.dataGridViewStoixeiaDiamerismatos.RowHeadersVisible =
true;

this.dataGridViewStoixeiaDiamerismatos.AutoGenerateColumns = false;
    this.dataGridViewStoixeiaDiamerismatos.AllowUserToAddRows
= false;

this.dataGridViewStoixeiaDiamerismatos.Columns[0].HeaderText = "Όροφος
- Διαμέρισμα";
    this.dataGridViewStoixeiaDiamerismatos.Columns[0].ReadOnly
= true;

    this.dataGridViewStoixeiaDiamerismatos.Columns[1].Visible =
false;//Το διαμέρισμα το έχω βάλει από πάνω

```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[2].HeaderText =  
"Τετραγωνικά διαμερίσματα";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[2].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[2].Width =  
100;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[3].HeaderText =  
"Σημείωσης";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[3].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[3].Width =  
350;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[3].Visible =  
false;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[4].HeaderText =  
"Όνοματεπώνυμο Ιδιοκτήτη";  
        this.dataGridViewStoixeiaDiamerismatos.Columns[4].ReadOnly  
= true;  
        this.dataGridViewStoixeiaDiamerismatos.Columns[4].Width =  
150;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[5].HeaderText =  
"Τηλέφωνο Ιδιοκτήτη";
```

```
        this.dataGridViewStoixeiaDiamerismatos.Columns[5].ReadOnly
= true;
        this.dataGridViewStoixeiaDiamerismatos.Columns[5].Width =
150;

        this.dataGridViewStoixeiaDiamerismatos.Columns[6].HeaderText =
"Όνοματεπώνυμο Ενοικιαστή";
        this.dataGridViewStoixeiaDiamerismatos.Columns[6].ReadOnly
= true;
        this.dataGridViewStoixeiaDiamerismatos.Columns[6].Width =
150;
        this.dataGridViewStoixeiaDiamerismatos.Columns[6].Visible =
false;

        this.dataGridViewStoixeiaDiamerismatos.Columns[7].HeaderText =
"Τηλέφωνο Ενοικιαστή";
        this.dataGridViewStoixeiaDiamerismatos.Columns[7].ReadOnly
= true;
        this.dataGridViewStoixeiaDiamerismatos.Columns[7].Width =
150;
        this.dataGridViewStoixeiaDiamerismatos.Columns[7].Visible =
false;

        this.dataGridViewStoixeiaDiamerismatos.Columns[8].HeaderText = "ει";
        this.dataGridViewStoixeiaDiamerismatos.Columns[8].ReadOnly
= true;
        this.dataGridViewStoixeiaDiamerismatos.Columns[8].Width =
30;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[9].HeaderText = "fi";
    this.dataGridViewStoixeiaDiamerismatos.Columns[9].ReadOnly
= true;
    this.dataGridViewStoixeiaDiamerismatos.Columns[9].Width =
30;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[10].HeaderText =
"Ανελκυστήρας";
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[10].ReadOnly = true;
    this.dataGridViewStoixeiaDiamerismatos.Columns[10].Width =
100;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[11].HeaderText =
"Κήπος";
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[11].ReadOnly = true;
    this.dataGridViewStoixeiaDiamerismatos.Columns[11].Width =
100;
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[12].HeaderText =
"Κλειστό διαμέρισμα";
```

```
this.dataGridViewStoixeiaDiamerismatos.Columns[12].ReadOnly = true;
    this.dataGridViewStoixeiaDiamerismatos.Columns[12].Width =
100;
```

```

    }

#endregion

decimal eifi = 0;
decimal wiei = 0;

if (this.dataGridViewGenikesDapanes.Rows.Count != 0)
{
    for (int j = 0; j <
this.dataGridViewStoixeiaDiamerismatos.Rows.Count; j++)
    {
        if (!this.chkAllDiamerismata.Checked)
        {
            if
(this.dataGridViewStoixeiaDiamerismatos.Rows[j].Cells[0].Value.ToString()
== this.cmbDiamerismata.SelectedItem.ToString())
            {
                i_diamer = j;
            }
        }

        eifi +=
Convert.ToDecimal(this.dataGridViewStoixeiaDiamerismatos.Rows[j].Cells[8
].Value) *
Convert.ToDecimal(this.dataGridViewStoixeiaDiamerismatos.Rows[j].Cells[8
].Value);

        for (int k = 0; k < this.dataGridViewPetrelaio.Rows.Count;
k++)

```

```

        {
            if
            (this.dataGridViewStoixeiaDiamerismatos.Rows[j].Cells[0].Value.ToString()
            == this.dataGridViewPetrelaio.Rows[k].Cells[4].Value.ToString())
            {
                wiei +=
                decimal.Multiply(Convert.ToDecimal(this.dataGridViewStoixeiaDiamerismato
                s.Rows[j].Cells[8].Value),
                Convert.ToDecimal(this.dataGridViewPetrelaio.Rows[k].Cells[5].Value));
                break;
            }
        }
    }

    object number = 1;

    if (!File.Exists(Application.StartupPath + "\\\" + TemplateName))
    {
        MessageBox.Show("Δεν υπάρχει η φόρμα της αίτησης !",
        "Προσοχή", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    object oMissing = System.Reflection.Missing.Value;
    object oTemplate = Application.StartupPath + "\\\" +
    TemplateName;

    oWord = new Word.Application();
    oWord.Visible = true;
    object ReadOnlyRecommended = true;

```

```
oWord.Documents.Open(ref oTemplate, ref oMissing, ref
ReadOnlyRecommended, ref oMissing, ref oMissing, ref
oMissing, ref oMissing, ref oMissing, ref oMissing, ref oMissing,
ref oMissing, ref oMissing, ref oMissing, ref oMissing);
```

```
if (!this.chkAllDiamerismata.Checked)
```

```
{
```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
```

```
oWord.Selection.TypeText(this.cmbDiamerismata.SelectedItem.ToString());
```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
```



```

oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);

oWord.Selection.TypeText(this.cmbEtos.SelectedItem.ToString() + "/" +
this.cmbMinas.SelectedItem.ToString());
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.MoveDown(ref oMissing, ref oMissing, ref
oMissing);
    for (int q = 0; q <
this.dataGridViewGenikesDapanes.Rows.Count; q++)
    {

oWord.Selection.TypeText(this.dataGridViewGenikesDapanes.Rows[q].Cells[4].Value.ToString());
oWord.Selection.Move(ref oMissing, ref oMissing);
    if
(this.dataGridViewGenikesDapanes.Rows[q].Cells[4].Value.ToString().Trim()
== "ΠΕΤΡΕΛΑΙΟ")
    {
        pososto =
YpologismosPetrailaiou(Convert.ToDecimal(this.dataGridViewStoixeiaDiame
rismatos.Rows[i_diamer].Cells[8].Value),
Convert.ToDecimal(this.dataGridViewStoixeiaDiamerismatos.Rows[i_diamer
].Cells[8].Value),
Convert.ToDecimal(this.dataGridViewPetrelaio.Rows[1].Cells[5].Value), eifi,
wiei) *
Convert.ToDecimal(this.dataGridViewGenikesDapanes.Rows[q].Cells[6].Value);

```

```

oWord.Selection.TypeText(pososto.ToString().Remove(6));
    }
    else
    {
        decimal dap =
Convert.ToDecimal(this.dataGridViewGenikesDapanes.Rows[q].Cells[6].Value) *
Convert.ToDecimal(this.dataGridViewStoixeiaDiamerismatos.Rows[i_diamer
].Cells[2].Value) / sunoloxilioswn;
        oWord.Selection.TypeText(dap.ToString().Remove(5));
    }
    oWord.Selection.Move(ref oMissing, ref oMissing);
    oWord.Selection.InsertRows(ref number);
}
}
else//----->Αναφορα για όλα τα
διαμερίσματα
{
    for (int i = 0; i <
this.dataGridViewStoixeiaDiamerismatos.Rows.Count; i++)
    {
        int i_petrel = 0;
        if (i == 0) //----->To format του word που έχω φτιάξει
        {
            oWord.Selection.Move(ref oMissing, ref oMissing);
            oWord.Selection.Move(ref oMissing, ref oMissing);
            oWord.Selection.Move(ref oMissing, ref oMissing);
            oWord.Selection.Move(ref oMissing, ref oMissing);
            oWord.Selection.Move(ref oMissing, ref oMissing);
            oWord.Selection.Move(ref oMissing, ref oMissing);
        }
    }
}

```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
```

```
oWord.Selection.TypeText(this.dataGridViewStoixeiaDiamerismatos.Rows[i]
.Cells[0].Value.ToString());
```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
```

```
oWord.Selection.TypeText(this.cmbEtos.SelectedItem.ToString() + "/" +
this.cmbMinas.SelectedItem.ToString());
```

```
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.Move(ref oMissing, ref oMissing);
oWord.Selection.MoveDown(ref oMissing, ref oMissing,
```

```
ref oMissing);
```

```
}
```

```

else// -----> To format το φτιάχνω εγω
{
    //oWord.Selection.Move(ref oMissing, ref oMissing);
    oWord.Selection.Rows.Select();
    oWord.Selection.Copy();
    oWord.Selection.Move(ref oMissing, ref oMissing);
    //oWord.Selection.SelectRow();
    oWord.Selection.InsertBreak(ref oMissing);
    oWord.Selection.Move(ref oMissing, ref oMissing);
    oWord.Selection.Paste();
    oWord.Selection.MoveUp(ref oMissing, ref oMissing, ref
oMissing);

    oWord.Selection.TypeText("ΔΙΑΜΕΡΙΣΜΑ");
    oWord.Selection.Move(ref oMissing, ref oMissing);

oWord.Selection.TypeText(this.dataGridViewStoixeiaDiamerismatos.Rows[i]
.Cells[0].Value.ToString());
    oWord.Selection.Move(ref oMissing, ref oMissing);
    oWord.Selection.InsertRows(ref number);
    oWord.Selection.TypeText("ΕΤΟΣ/ΜΗΝΑΣ");
    oWord.Selection.Move(ref oMissing, ref oMissing);

oWord.Selection.TypeText(this.cmbEtos.SelectedItem.ToString() + "/" +
this.cmbMinas.SelectedItem.ToString());
    oWord.Selection.Move(ref oMissing, ref oMissing);
    oWord.Selection.InsertRows(ref number);

}

//-----
//          Τυπώνω τις δαπάνες

```

```

//-----

        for (int q = 0; q <
this.dataGridViewGenikesDapanes.Rows.Count; q++)
        {

oWord.Selection.TypeText(this.dataGridViewGenikesDapanes.Rows[q].Cell
s[4].Value.ToString());
                oWord.Selection.Move(ref oMissing, ref oMissing);
                if
(this.dataGridViewGenikesDapanes.Rows[q].Cells[4].Value.ToString().Trim()
== "ΠΙΤΡΕΛΑΙΟ")
                {
                        for (int w = 0; w <
this.dataGridViewPetrelaiou.Rows.Count; w++)
                        {

                                if
(this.dataGridViewStoixeiaDiamerismatos.Rows[i].Cells[0].Value.ToString()
== this.dataGridViewPetrelaiou.Rows[w].Cells[4].Value.ToString())
                                {
                                        i_petrel = w;
                                        break;
                                }
                        }

                pososto =
YpologismosPetrailaiou(Convert.ToDecimal(this.dataGridViewStoixeiaDiamer
ismatos.Rows[i].Cells[8].Value),
Convert.ToDecimal(this.dataGridViewStoixeiaDiamerismatos.Rows[i].Cells[8
].Value),

```



```

    }

    decimal dap =
Convert.ToDecimal(this.dataGridViewGenikesDapanes.Rows[q].Cells[6].Value) *
Convert.ToDecimal(this.dataGridViewStoixeiaDiamerismatos.Rows[i_diamer
].Cells[2].Value) / sunoloxilioswn;

oWord.Selection.TypeText(dap.ToString().Remove(5));
    }

    oWord.Selection.Move(ref oMissing, ref oMissing);
    oWord.Selection.InsertRows(ref number);
}

}
}

    object FileName = Application.StartupPath + "\\Save" +
TemplateName;
    oWord.ActiveDocument.SaveAs(ref FileName, ref oMissing, ref
oMissing, ref oMissing, ref oMissing, ref oMissing, ref
ReadOnlyRecommended, ref oMissing, ref oMissing, ref oMissing, ref
oMissing, ref oMissing, ref oMissing, ref oMissing, ref
oMissing);
    oWord.Visible = true;

}
else
{

```

```

        MessageBox.Show("Δεν υπάρχουν δαπάνες για αυτό το μήνα!",
        "Πληροφορία", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
}

```

```

decimal ΥπολογισμοςPetrailaiou(decimal ei, decimal fi, decimal
wi, decimal SEiFi, decimal Seiwi)

```

```

{
    decimal x = ei * fi;
    decimal pososto;

    if (Seiwi != 0)
    {
        pososto = x + (1 - (SEiFi)) * ((ei * wi) / Seiwi);
        //pososto = x + (1 - (SEiFi)) * ((ei * wi) / Seiwi);
    }
    else
    {
        pososto = x;
    }

    return pososto;
}

```

```

private void chkAllDiamerismata_CheckedChanged(object sender,
EventArgs e)

```

```

{
    if (this.chkAllDiamerismata.Checked)

```



```

    {
        this.cmbEtos.Enabled = true;
        this.cmbMinas.Enabled = true;
        this.cmbDiamerismata.Enabled = false;
    }
    else
    {
        this.cmbEtos.Enabled = true;
        this.cmbMinas.Enabled = true;
        this.cmbDiamerismata.Enabled = true;
    }
}
}
}
}

```

frmInsDapanes.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ptixiaki

```

```

{
    public partial class frmInsDapanes : Form
    {
        AppLib.InsDapanes DapaniCls = new
AppLib.InsDapanes(AppLib.DB_Connection.SQLConnection());
        AppLib.Dapanes DapanesCls = new
AppLib.Dapanes(AppLib.DB_Connection.SQLConnection());

        bool emptyValue1 = false;

        bool emptyValue = false;
        bool flagEdit = false;
        int id1 = 0;

        public frmInsDapanes()
        {
            InitializeComponent();
        }

        private void frmInsDapanes_Load(object sender, EventArgs e)
        {
            #region Fill Combobox -> GroupBox Submit

            System.Collections.ArrayList CategoryList =
DapanesCls.CreateDapanesList();
            this.cmbDapanes.DataSource = CategoryList;
            if (this.cmbDapanes.Items.Count != 0)
            {
                this.cmbDapanes.DisplayMember = "Dapanes";
                this.cmbDapanes.ValueMember = "id";
                this.cmbDapanes.SelectedIndex = -1;
            }
        }
    }
}

```

```
}
```

```
#endregion
```

```
#region FilldataGridViewDiamerisma
```

```
this.dataGridViewDapanes.DataSource =  
DapaniCls.DatagridFillDapanesList();
```

```
if (this.dataGridViewDapanes.ColumnCount != 0)
```

```
{
```

```
    this.dataGridViewDapanes.RowHeadersVisible = true;  
    this.dataGridViewDapanes.AutoGenerateColumns = false;  
    this.dataGridViewDapanes.AllowUserToAddRows = false;
```

```
    this.dataGridViewDapanes.Columns[0].HeaderText = "id";  
    this.dataGridViewDapanes.Columns[0].Visible = false;
```

```
    this.dataGridViewDapanes.Columns[1].HeaderText = "Έτος";  
    this.dataGridViewDapanes.Columns[1].ReadOnly = true;  
    this.dataGridViewDapanes.Columns[1].Width = 60;
```

```
    this.dataGridViewDapanes.Columns[2].HeaderText = "Μήνα";  
    this.dataGridViewDapanes.Columns[2].ReadOnly = true;  
    this.dataGridViewDapanes.Columns[2].Width = 80;
```

```
    this.dataGridViewDapanes.Columns[3].HeaderText = "Dapani";  
    this.dataGridViewDapanes.Columns[3].Visible = false;
```

```
    this.dataGridViewDapanes.Columns[4].HeaderText = "Δαπάνη";
```

```

this.dataGridViewDapanes.Columns[4].ReadOnly = true;
this.dataGridViewDapanes.Columns[4].Width = 100;

this.dataGridViewDapanes.Columns[5].HeaderText =
"Υπολογισμός Δαπάνης ανα χιλιοστά";
this.dataGridViewDapanes.Columns[5].ReadOnly = true;
this.dataGridViewDapanes.Columns[5].Width = 200;

this.dataGridViewDapanes.Columns[6].HeaderText = "Ποσό";
this.dataGridViewDapanes.Columns[6].ReadOnly = true;
this.dataGridViewDapanes.Columns[6].Width = 60;

}
#endregion
}

private void btnSubmit_Click(object sender, EventArgs e)
{
    if (this.cmbetos.SelectedIndex == -1)
    {
        errorProvider.SetError(this.cmbetos, "Δεν έχετε επιλέξει έτος");
        emptyValue = true;
    }

    if (this.cmbminas.SelectedIndex == -1)
    {
        errorProvider.SetError(this.cmbminas, "Δεν έχετε επιλέξει μήνα");
        emptyValue = true;
    }

    if (this.cmbDapanes.SelectedIndex == -1)

```

```

    {
        errorProvider.SetError(this.cmbDapanes, "Δεν έχετε επιλέξει
δαπάνες");
        emptyValue = true;
    }

    if ((!this.rdbxiliosta.Checked) && (!this.rdbDiamerisma.Checked))
    {
        errorProvider.SetError(this.rdbDiamerisma, "Δεν έχετε επιλέξει
δαπάνη");
        emptyValue = true;
    }

    if (this.txtPoso.Text == "")
    {
        errorProvider.SetError(this.txtPoso, "Δεν έχετε επιλέξει ποσό");
        emptyValue = true;
    }

    if (!emptyValue)
    {
        if (flagEdit)
        {
            DapaniCls.Etos = this.cmbetos.SelectedItem.ToString();
            DapaniCls.Minas = this.cmbminas.SelectedItem.ToString();
            DapaniCls.Dapani =
Convert.ToInt16(this.cmbDapanes.SelectedValue);
            if (this.rdbxiliosta.Checked)
            {
                DapaniCls.YpoiDapani = true;
            }
        }
    }

```

```

else
{
    DapaniCls.YpoiDapani = false;
}
DapaniCls.poso = this.txtPoso.Text;

if (DapaniCls.UpdateDapani(id1))
{
    MessageBox.Show("Η καταχώριση των στοιχείων έγινε με
επιτυχία", "Μήνυμα", MessageBoxButtons.OK,
MessageBoxIcon.Information);

    this.cmbDapanes.SelectedIndex = -1;
    this.cmbetos.SelectedIndex = -1;
    this.cmbminas.SelectedIndex = -1;
    this.txtPoso.Text = "";

    #region FilldataGridViewDiamerisma

    this.dataGridViewDapanes.DataSource =
DapaniCls.DatagridFillDapanesList();

    if (this.dataGridViewDapanes.ColumnCount != 0)
    {
        this.dataGridViewDapanes.RowHeadersVisible = true;
        this.dataGridViewDapanes.AutoGenerateColumns = false;
        this.dataGridViewDapanes.AllowUserToAddRows = false;

        this.dataGridViewDapanes.Columns[0].HeaderText = "id";
        this.dataGridViewDapanes.Columns[0].Visible = false;
    }
}

```

```

        this.dataGridViewDapanes.Columns[1].HeaderText =
"Ετος";

        this.dataGridViewDapanes.Columns[1].ReadOnly = true;
        this.dataGridViewDapanes.Columns[1].Width = 60;

        this.dataGridViewDapanes.Columns[2].HeaderText =
"Μήνα";

        this.dataGridViewDapanes.Columns[2].ReadOnly = true;
        this.dataGridViewDapanes.Columns[2].Width = 80;

        this.dataGridViewDapanes.Columns[3].HeaderText =
"Dapani";

        this.dataGridViewDapanes.Columns[3].Visible = false;

        this.dataGridViewDapanes.Columns[4].HeaderText =
"Δαπάνη";

        this.dataGridViewDapanes.Columns[4].ReadOnly = true;
        this.dataGridViewDapanes.Columns[4].Width = 100;

        this.dataGridViewDapanes.Columns[5].HeaderText =
"Υπολογισμός Δαπάνης ανα χιλιοστά";

        this.dataGridViewDapanes.Columns[5].ReadOnly = true;
        this.dataGridViewDapanes.Columns[5].Width = 90;

        this.dataGridViewDapanes.Columns[6].HeaderText =
"Ποσό";

        this.dataGridViewDapanes.Columns[6].ReadOnly = true;
        this.dataGridViewDapanes.Columns[6].Width = 60;
    }
    #endregion
}

```

```

else
{
    MessageBox.Show("Πρόβλημα στην καταχώριση των
στοιχείων", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else
{
    DapaniCls.Etos = this.cmbetos.SelectedItem.ToString();
    DapaniCls.Minas = this.cmbminas.SelectedItem.ToString();
    DapaniCls.Dapani =
Convert.ToInt16(this.cmbDapanes.SelectedValue);
    if (this.rdbxiliosta.Checked)
    {
        DapaniCls.YpoiDapani = true;
    }
    else
    {
        DapaniCls.YpoiDapani = false;
    }
    DapaniCls.poso = this.txtPoso.Text;

    if (DapaniCls.InsertDapani())
    {
        MessageBox.Show("Η καταχώριση των στοιχείων έγινε με
επιτυχία", "Μήνυμα", MessageBoxButtons.OK,
MessageBoxIcon.Information);

        #region FilldataGridViewDiamerisma

```



```

        this.dataGridViewDapanes.DataSource =
DapaniCls.DatagridFillDapanesList();

        if (this.dataGridViewDapanes.ColumnCount != 0)
        {
            this.dataGridViewDapanes.RowHeadersVisible = true;
            this.dataGridViewDapanes.AutoGenerateColumns = false;
            this.dataGridViewDapanes.AllowUserToAddRows = false;

            this.dataGridViewDapanes.Columns[0].HeaderText = "id";
            this.dataGridViewDapanes.Columns[0].Visible = false;

            this.dataGridViewDapanes.Columns[1].HeaderText =
"Ετος";

            this.dataGridViewDapanes.Columns[1].ReadOnly = true;
            this.dataGridViewDapanes.Columns[1].Width = 60;

            this.dataGridViewDapanes.Columns[2].HeaderText =
"Μήνα";

            this.dataGridViewDapanes.Columns[2].ReadOnly = true;
            this.dataGridViewDapanes.Columns[2].Width = 80;

            this.dataGridViewDapanes.Columns[3].HeaderText =
"Dapani";

            this.dataGridViewDapanes.Columns[3].Visible = false;

            this.dataGridViewDapanes.Columns[4].HeaderText =
"Δαπάνη";

            this.dataGridViewDapanes.Columns[4].ReadOnly = true;
            this.dataGridViewDapanes.Columns[4].Width = 100;

```

```

        this.dataGridViewDapanes.Columns[5].HeaderText =
"Υπολογισμός Δαπάνης ανα χιλιοστά";
        this.dataGridViewDapanes.Columns[5].ReadOnly = true;
        this.dataGridViewDapanes.Columns[5].Width = 90;

        this.dataGridViewDapanes.Columns[6].HeaderText =
"Ποσό";
        this.dataGridViewDapanes.Columns[6].ReadOnly = true;
        this.dataGridViewDapanes.Columns[6].Width = 60;
    }
    #endregion
}

else
{
    MessageBox.Show("Πρόβλημα στην καταχώριση των
στοιχείων", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
}

}

private void
dataGridViewDapanes_RowHeaderMouseDoubleClick(object sender,
DataGridViewCellEventArgs e)
{
    int etos = 0;

```

```
switch
(this.dataGridViewDapanes.Rows[e.RowIndex].Cells[2].Value.ToString())
{
    case "2009":
        etos = 0;
        break;
    case "2010":
        etos = 1;
        break;
    case "2011":
        etos = 2;
        break;
    case "2012":
        etos = 3;
        break;
    case "2013":
        etos = 4;
        break;
    case "2014":
        etos = 5;
        break;
    case "2015":
        etos = 6;
        break;
    case "2016":
        etos = 7;
        break;
    case "2017":
        etos = 8;
        break;
    case "2018":
```

```
        etos = 9;
        break;
    case "2019":
        etos = 10;
        break;
    case "2020":
        etos = 11;
        break;
    case "2021":
        etos = 12;
        break;
    default:
        break;
}
```

```
this.cmbetos.SelectedIndex = etos;
```

```
int minas = 0;
```

```
switch
```

```
(this.dataGridViewDapanes.Rows[e.RowIndex].Cells[3].Value.ToString())
```

```
{
    case "Ιανουάριος":
        etos = 0;
        break;
    case "Φεβρουάριος":
        etos = 1;
        break;
    case "Μάρτιος":
        etos = 2;
        break;
}
```

```
case "Απρίλιος":  
    etos = 3;  
    break;  
case "Μάιος":  
    etos = 4;  
    break;  
case "Ιουνιος":  
    etos = 5;  
    break;  
case "Ιουλιος":  
    etos = 6;  
    break;  
case "Αυγουστος":  
    etos = 7;  
    break;  
case "Σεπτέμβριος":  
    etos = 8;  
    break;  
case "Οκτώβριος":  
    etos = 9;  
    break;  
case "Νοέμβριος":  
    etos = 10;  
    break;  
case "Δεκέμβριος":  
    etos = 11;  
    break;  
default:  
    break;  
}
```

```

        this.cmbminas.SelectedIndex = minas;

        this.cmbDapanes.SelectedIndex =
Convert.ToInt32(this.dataGridViewDapanes.Rows[e.RowIndex].Cells[4].Value) - 1;

        if
(Convert.ToBoolean(this.dataGridViewDapanes.Rows[e.RowIndex].Cells[6].Value))
        {
            this.rdbxiliosta.Checked = true;
        }
        else
        {
            this.rdbDiamerisma.Checked = true;
        }
        this.txtPoso.Text =
this.dataGridViewDapanes.Rows[e.RowIndex].Cells[7].Value.ToString();
        id1 =
Convert.ToInt16(this.dataGridViewDapanes.Rows[e.RowIndex].Cells[1].Value);

        flagEdit = true;
        this.btnSubmit.Text = "Διόρθωση";
    }

private void btnParDapanesSubmit_Click(object sender, EventArgs e)
{
    if (this.txtDapani.Text == "")
    {

```

```

        errorProvider.SetError(this.txtDapani, "Δεν έχετε συμπληρώσει την
διεύθυνση");
        emptyValue1 = true;
    }

    if (!emptyValue1)
    {
        DapanesCls.dapanes = this.txtDapani.Text.Trim().ToUpper();

        if (DapanesCls.InsertExtraDapani())
        {
            MessageBox.Show("Η καταχώριση έγινε με επιτυχία!",
"Μήνυμα", MessageBoxButtons.OK, MessageBoxIcon.Information);
            #region Fill Combobox -> GroupBox Submit

            System.Collections.ArrayList CategoryList =
DapanesCls.CreateDapanesList();
            this.cmbDapanes.DataSource = CategoryList;
            if (this.cmbDapanes.Items.Count != 0)
            {
                this.cmbDapanes.DisplayMember = "Dapanes";
                this.cmbDapanes.ValueMember = "id";
                this.cmbDapanes.SelectedIndex = -1;
            }

            #endregion
            this.txtDapani.Text = "";
        }
        else
        {

```

```
        MessageBox.Show("Πρόβλημα στη καταχώριση δεδομένων!",  
"Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);  
    }  
  
    }  
    }  
    }  
}
```

frmAddDiamerisma.cs

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
using AppLib;  
using System.Data.SqlClient;  
  
namespace ptixiaki  
{  
    public partial class frmAddDiamerisma : Form  
    {  
        AppLib.diamerisma Diamerismatacls = new  
diamerisma(AppLib.DB_Connection.SQLConnection());
```



```

bool flag1 = false;

public frmAddDiamerisma()
{
    InitializeComponent();
}

public frmAddDiamerisma(AppLib.diamerisma Diamerismacls)
{
    InitializeComponent();

    Diamerismatacls = Diamerismacls;
    flag1 = true;

    //this.txtarithmosenoikiasti.Text = "dfsgsdfgsdfg";
}

private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnKataxorisi_Click(object sender, EventArgs e)
{
    bool emptyValue = false;

    if (this.cmbKodikos.SelectedIndex == -1)
    {
        errorProvider.SetError(this.cmbKodikos, "Δεν έχετε επιλέξει
όροφο");
        emptyValue = true;
    }
}

```

```

    }

    if (this.numericdiamerisma.Value == 0)
    {
        errorProvider.SetError(this.numericdiamerisma, "Δεν έχετε επιλέξει
όροφο διαμερίσματος");
        emptyValue = true;
    }

    if (this.txttm.Text == "")
    {
        errorProvider.SetError(this.txttm, "Δεν έχετε συμπλήρωση τα
τετραγωνικά ");
        emptyValue = true;
    }

    if ((this.txtonomateponumoidioktiti.Text == "") &&
(this.txtonomateponimoenoikiasti.Text == ""))
    {
        if (this.txtonomateponumoidioktiti.Text == "")
        {
            errorProvider.SetError(this.txtonomateponumoidioktiti, "Δεν
έχετε συμπλήρωση το όνομα του ιδιοκτήτη");
        }
        if (this.txtonomateponimoenoikiasti.Text == "")
        {
            errorProvider.SetError(this.txtonomateponimoenoikiasti, "Δεν
έχετε συμπλήρωση το όνομα του ενοικιαστή");
        }
        emptyValue = true;
    }
}

```

```

        if ((this.txtarithmosenoikiasti.Text == "") &&
            (this.txtarithmosidioktiti.Text == ""))
        {
            if (this.txtarithmosenoikiasti.Text == "")
            {
                errorProvider.SetError(this.txtarithmosenoikiasti, "Δεν έχετε
συμπλήρωση το τηλέφωνο του ιδιοκτήτη");
            }
            if (this.txtarithmosidioktiti.Text == "")
            {
                errorProvider.SetError(this.txtarithmosidioktiti, "Δεν έχετε
συμπλήρωση το τηλέφωνο του ενοικιαστή");
            }
            emptyValue = true;
        }

        if (this.txtei.Text == "")
        {
            errorProvider.SetError(this.txtei, "Δεν έχετε συμπλήρωση το ει ");
            emptyValue = true;
        }

        if (this.txtfi.Text == "")
        {
            errorProvider.SetError(this.txtfi, "Δεν έχετε συμπλήρωση το fi ");
            emptyValue = true;
        }

        if (!emptyValue)
        {

```

```

        if (!flag1)
        {
            if
(Diamerismatacls.Select_Check(Convert.ToString(this.cmbKodikos.Selected
Item), Convert.ToInt16(this.numericdiamerisma.Value)))
            {
                Diamerismatacls.Orofos =
Convert.ToString(this.cmbKodikos.SelectedItem);
                Diamerismatacls.Diamerisma =
Convert.ToInt16(this.numericdiamerisma.Value);
                Diamerismatacls.Tetragonika = this.txttm.Text;
                Diamerismatacls.note = this.txtsxolia.Text.Trim().ToUpper();
                Diamerismatacls.Idioktitis =
this.txtomateponumoidioktiti.Text.Trim().ToUpper();
                Diamerismatacls.phoneldioktitis =
this.txtarithmosidioktiti.Text;
                Diamerismatacls.Enoikiastis =
this.txtomateponimoenoikiasti.Text.Trim().ToUpper();
                Diamerismatacls.phoneNoikiastis =
this.txtarithmosenoikiasti.Text;
                Diamerismatacls.ei = this.txtei.Text;
                Diamerismatacls.fi = this.txtfi.Text;

                Diamerismatacls.Ansanser = this.chkanelkistiras.Checked;
                Diamerismatacls.Garden = this.chkkipos.Checked;
                Diamerismatacls.Kleistodiamerisma =
this.chkClose.Checked;

            if (Diamerismatacls.Insert())
            {

```

```

        MessageBox.Show("Εισαγωγή των στοιχείων του
διαμερίσματος έγινε με επιτυχία", "Μήνυμα", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        this.Close();
    }
    else
    {
        MessageBox.Show("Πρόβλημα στη καταχώριση
δεδομένων", "Μήνυμα σφάλματος", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}
else
{
    DialogResult ans = MessageBox.Show("Υπάρχει ήδη αυτό το
διαμέρισμα!Θέλετε να διορθώσης τα δεδομένα?", "Μήνυμα",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (DialogResult.Yes == ans)
    {
        Diamerismatacls.Orofos =
Convert.ToString(this.cmbKodikos.SelectedIndex);
        Diamerismatacls.Diamerisma =
Convert.ToInt16(this.numericdiamerisma.Value);
        Diamerismatacls.Tetragonika = this.txttm.Text;
        Diamerismatacls.note =
this.txtsxolia.Text.Trim().ToUpper();
        Diamerismatacls.Idioktitis =
this.txtonomateponumoidioktiti.Text.Trim().ToUpper();
        Diamerismatacls.phoneldioktitis =
this.txtarithmosidioktiti.Text;
    }
}
}

```

```

        Diamerismatacls.Enoikiastis =
this.txtonomateponimoenoikiasti.Text.Trim().ToUpper();
        Diamerismatacls.phoneEnoikiastis =
this.txtarithmosenoikiasti.Text;
        Diamerismatacls.ei = this.txtei.Text;
        Diamerismatacls.fi = this.txtfi.Text;

        Diamerismatacls.Ansanser = this.chkanelkistiras.Checked;
        Diamerismatacls.Garden = this.chkkipos.Checked;
        Diamerismatacls.Kleistodiamerisma =
this.chkClose.Checked;

        if
(Diamerismatacls.Update(Convert.ToString(this.cmbKodikos.SelectedItem),
Convert.ToInt16(this.numericdiamerisma.Value))
        {
            MessageBox.Show("Διόρθωση των στοιχείων του
διαμερίσματος έγινε με επιτυχία", "Μήνυμα", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            this.Close();
        }
        else
        {
            MessageBox.Show("Πρόβλημα στη καταχώριση
δεδομένων", "Μήνυμα σφάλματος", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
    else
    {
        this.cmbKodikos.SelectedIndex = -1;
    }
}

```



```

        Diamerismatacls.phoneEnoikiastis =
this.txtarithmosenoikiasti.Text;
        Diamerismatacls.ei = this.txtei.Text;
        Diamerismatacls.fi = this.txtfi.Text;

        Diamerismatacls.Ansanser = this.chkanelkistiras.Checked;
        Diamerismatacls.Garden = this.chkkipos.Checked;
        Diamerismatacls.Kleistodiamerisma = this.chkClose.Checked;

        if
(Diamerismatacls.Update(Convert.ToString(this.cmbKodikos.SelectedItem),
Convert.ToInt16(this.numericdiamerisma.Value)))
        {
            MessageBox.Show("Διόρθωση των στοιχείων του
διαμερίσματος έγινε με επιτυχία", "Μήνυμα", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            this.Close();
        }
        else
        {
            MessageBox.Show("Πρόβλημα στη καταχώριση δεδομένων",
"Μήνυμα σφάλματος", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void frmAddDiamerisma_Load(object sender, EventArgs e)
{
    if (flag1)

```



```

{
    string orof = Convert.ToString(Diamerismatacls.Diamerisma);
    string orof1 = Diamerismatacls.Orofos.Replace( orof , "");
    Int16 orof2 = -1;
    switch (orof1)
    {
        case "ΙΣ":
            orof2 = 0;
            break;
        case "Α":
            orof2 = 1;
            break;
        case "Β":
            orof2 = 2;
            break;
        case "Γ":
            orof2 = 3;
            break;
        case "Δ":
            orof2 = 4;
            break;
        case "Ε":
            orof2 = 5;
            break;
        default:
            break;
    }

    this.cmbKodikos.SelectedIndex = orof2;
    this.numericdiamerisma.Value = Diamerismatacls.Diamerisma;
    this.txttm.Text = Diamerismatacls.Tetragonika.Replace(",",".");
}

```



```

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ptixiaki
{
    public partial class frmAddDapanesDiam : Form
    {
        AppLib.DapaniTherm DapaniThermCls = new
AppLib.DapaniTherm(AppLib.DB_Connection.SQLConnection());
        bool flagInsert = false;
        bool flagUpdate = false;
        int update_id = 0;

        public frmAddDapanesDiam()
        {
            InitializeComponent();
        }

        private void frmAddDapanesDiam_Load(object sender, EventArgs e)
        {
            #region FilldataGridViewDiamerisma

            this.dataGridView1.DataSource =
DapaniThermCls.DatagridFillDapanesThermList();

            if (this.dataGridView1.ColumnCount != 0)

```

```

{
    this.dataGridView1.RowHeadersVisible = true;
    this.dataGridView1.AutoGenerateColumns = false;
    this.dataGridView1.AllowUserToAddRows = false;

    if (this.dataGridView1.Columns.Contains("ChkSelect"))
    {
        this.dataGridView1.Columns.Remove("ChkSelect");
    }

    DataGridViewCheckBoxColumn chkcolumnEdition = new
DataGridViewCheckBoxColumn();
    {
        chkcolumnEdition.HeaderText = "Επιλογή";
        chkcolumnEdition.Name = "ChkSelect";
        chkcolumnEdition.AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
        chkcolumnEdition.FlatStyle = FlatStyle.Standard;
        chkcolumnEdition.CellTemplate = new
DataGridViewCheckBoxCell();
    }

    this.dataGridView1.Columns.Insert(0, chkcolumnEdition);
    this.dataGridView1.Columns[0].ReadOnly = false;

    this.dataGridView1.Columns[1].Visible = false;//Το διαμέρισμα το
έχω βάλει από πάνω

    this.dataGridView1.Columns[2].HeaderText = "Έτος";
    this.dataGridView1.Columns[2].ReadOnly = true;
    this.dataGridView1.Columns[2].Width = 50;

```

```

this.dataGridView1.Columns[3].HeaderText = "Μήνας";
this.dataGridView1.Columns[3].ReadOnly = true;
this.dataGridView1.Columns[3].Width = 150;

this.dataGridView1.Columns[4].HeaderText = "Διαμέρισμα index";
this.dataGridView1.Columns[4].Visible = false;

this.dataGridView1.Columns[5].HeaderText = "Διαμέρισμα";
this.dataGridView1.Columns[5].ReadOnly = true;
this.dataGridView1.Columns[5].Width = 50;

this.dataGridView1.Columns[6].HeaderText = "Ποσό";
this.dataGridView1.Columns[6].ReadOnly = true;
this.dataGridView1.Columns[6].Width = 50;

}
#endregion

}

private void btnSubmit_Click(object sender, EventArgs e)
{
    flagInsert = false;
    if (this.cmbEtos.SelectedIndex == -1)
    {
        errorProvider.SetError(this.cmbEtos, "Δεν έχετε διαλέξει έτος!");
        flagInsert = true;
    }
    if (this.cmbMinas.SelectedIndex == -1)

```

```

{
    errorProvider.SetError(this.cmbMinas, "Δεν έχετε διαλέξει μήνα!");
    flagInsert = true;
}
if (this.cmbkodikosdiamerismatos.SelectedIndex == -1)
{
    errorProvider.SetError(this.cmbkodikosdiamerismatos, "Δεν έχετε
διαλέξει διαμέρισμα!");
    flagInsert = true;
}
if (this.txttorini.Text == "")
{
    errorProvider.SetError(this.txttorini, "Δεν έχετε εισάγει μέτρηση
πετρελαίου!");
    flagInsert = true;
}

if (!flagInsert)
{
    DapaniThermCls.Etos = this.cmbEtos.SelectedItem.ToString();
    DapaniThermCls.Minas = this.cmbMinas.SelectedItem.ToString();
    DapaniThermCls.Diamerisma =
this.cmbkodikosdiamerismatos.SelectedIndex;
    DapaniThermCls.Metrisi = this.txttorini.Text;

    if (!flagUpdate)
    {
        if (DapaniThermCls.InsertDapanTherm())
        {

```

```
        MessageBox.Show("Η διόρθωση των δεδομένων έγινε με  
επιτυχία!", "Πληροφορία", MessageBoxButtons.OK,  
        MessageBoxIcon.Information);
```

```
        #region FilldataGridViewDiamesisma
```

```
        this.dataGridView1.DataSource =  
        DapaniThermCls.DatagridFillDapanesThermList();
```

```
        if (this.dataGridView1.ColumnCount != 0)
```

```
        {
```

```
            this.dataGridView1.RowHeadersVisible = true;
```

```
            this.dataGridView1.AutoGenerateColumns = false;
```

```
            this.dataGridView1.AllowUserToAddRows = false;
```

```
            if (this.dataGridView1.Columns.Contains("ChkSelect"))
```

```
            {
```

```
                this.dataGridView1.Columns.Remove("ChkSelect");
```

```
            }
```

```
            DataGridViewCheckBoxColumn chkcolumnEdition = new  
            DataGridViewCheckBoxColumn();
```

```
            {
```

```
                chkcolumnEdition.HeaderText = "Επιλογή";
```

```
                chkcolumnEdition.Name = "ChkSelect";
```

```
                chkcolumnEdition.AutoSizeMode =
```

```
                DataGridViewAutoSizeColumnMode.DisplayedCells;
```

```
                chkcolumnEdition.FlatStyle = FlatStyle.Standard;
```

```
                chkcolumnEdition.CellTemplate = new
```

```
                DataGridViewCheckBoxCell();
```

```
            }
```

```

this.dataGridView1.Columns.Insert(0, chkcolumnEdition);
this.dataGridView1.Columns[0].ReadOnly = false;

this.dataGridView1.Columns[1].Visible = false;//To
διαμέρισμα το έχω βάλει από πάνω

this.dataGridView1.Columns[2].HeaderText = "Ετος";
this.dataGridView1.Columns[2].ReadOnly = true;
this.dataGridView1.Columns[2].Width = 100;

this.dataGridView1.Columns[3].HeaderText = "Μήνας";
this.dataGridView1.Columns[3].ReadOnly = true;
this.dataGridView1.Columns[3].Width = 350;

this.dataGridView1.Columns[4].HeaderText = "Διαμέρισμα
index";

this.dataGridView1.Columns[4].Visible = false;

this.dataGridView1.Columns[5].HeaderText =
"Διαμέρισμα";
this.dataGridView1.Columns[5].ReadOnly = true;
this.dataGridView1.Columns[5].Width = 150;

this.dataGridView1.Columns[6].HeaderText = "Ποσό";
this.dataGridView1.Columns[6].ReadOnly = true;
this.dataGridView1.Columns[6].Width = 150;

}
#endregion

```



```

    }
    else
    {
        MessageBox.Show("Πρόβλημα στη καταχώρηση των
δεδομένων!", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
else
{
    if (DapaniThermCls.UpdateDapanTherm(update_id))
    {
        MessageBox.Show("Η καταχώρηση των δεδομένων έγινε με
επιτυχία!", "Πληροφορία", MessageBoxButtons.OK,
MessageBoxIcon.Information);

        #region FilldataGridViewDiamerisma

        this.dataGridView1.DataSource =
DapaniThermCls.DatagridFillDapanesThermList();

        if (this.dataGridView1.ColumnCount != 0)
        {
            this.dataGridView1.RowHeadersVisible = true;
            this.dataGridView1.AutoGenerateColumns = false;
            this.dataGridView1.AllowUserToAddRows = false;

            if (this.dataGridView1.Columns.Contains("ChkSelect"))
            {
                this.dataGridView1.Columns.Remove("ChkSelect");
            }
        }
    }
}

```

```

        DataGridViewCheckBoxColumn chkcolumnEdition = new
DataGridViewCheckBoxColumn();
    {
        chkcolumnEdition.HeaderText = "Επιλογή";
        chkcolumnEdition.Name = "ChkSelect";
        chkcolumnEdition.AutoSizeMode =
DataGridViewAutoSizeColumnMode.DisplayedCells;
        chkcolumnEdition.FlatStyle = FlatStyle.Standard;
        chkcolumnEdition.CellTemplate = new
DataGridViewCheckBoxCell();
    }

    this.dataGridView1.Columns.Insert(0, chkcolumnEdition);
    this.dataGridView1.Columns[0].ReadOnly = false;

    this.dataGridView1.Columns[1].Visible = false;//To
διαμέρισμα το έχω βάλει από πάνω

    this.dataGridView1.Columns[2].HeaderText = "Έτος";
    this.dataGridView1.Columns[2].ReadOnly = true;
    this.dataGridView1.Columns[2].Width = 100;

    this.dataGridView1.Columns[3].HeaderText = "Μήνας";
    this.dataGridView1.Columns[3].ReadOnly = true;
    this.dataGridView1.Columns[3].Width = 350;

    this.dataGridView1.Columns[4].HeaderText = "Διαμέρισμα
index";

    this.dataGridView1.Columns[4].Visible = false;

```



```

        this.cmbkodikosdiamerismatos.SelectedItem =
this.dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
        this.txttorini.Text =
this.dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();
        this.update_id =
Convert.ToInt32(this.dataGridView1.Rows[e.RowIndex].Cells[1].Value);

        this.btnSubmit.Text = "Διόρθωση";
        flagUpdate = true;
    }

}
}

```

Καθώς επίσης και της κλάσης που χρησιμοποίησα

polikatikia.cs

```

using System;
using System.IO;
using System.Text;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;

```

```

namespace AppLib
{
    public class polikatikia
    {
        #region Δηλώσεις
        private string ConnStr;

        private string m_Address;
        private string m_Number;
        private string m_City;
        private Int16 m_TK;
        private string m_Diaxiristis;
        private string m_PhoneDiaxiristis;
        private Int16 m_Orofoi;
        private Int16 m_ArithmosDiamerismaton;
        private bool m_Asanser;
        private bool m_Kipos;
        private bool m_autonomithermansi;
        private string m_tm;

        #endregion

        #region Constructor

        public polikatikia()
        {
        }

        public polikatikia(string ConnectionString)
        {
            this.ConnStr = ConnectionString;
        }
    }
}

```

```
}
```

```
#endregion
```

```
#region Μέθοδοι - Properties
```

```
public Int16 TK
```

```
{
```

```
    get
```

```
    {
```

```
        return m_TK;
```

```
    }
```

```
    set
```

```
    {
```

```
        m_TK = value;
```

```
    }
```

```
}
```

```
public string Address
```

```
{
```

```
    get
```

```
    {
```

```
        return m_Address;
```

```
    }
```

```
    set
```

```
    {
```

```
        m_Address = value;
```

```
    }
```

```
}
```

```
public string Number
```

```
{
```

```
    get
    {
        return m_Number;
    }
    set
    {
        m_Number = value;
    }
}

public string City
{
    get
    {
        return m_City;
    }
    set
    {
        m_City = value;
    }
}

public string Diaxiristis
{
    get
    {
        return m_Diaxiristis;
    }
    set
    {
        m_Diaxiristis = value;
    }
}
```

```
    }  
}  
  
public string PhoneDiaxiristis  
{  
    get  
    {  
        return m_PhoneDiaxiristis;  
    }  
    set  
    {  
        m_PhoneDiaxiristis = value;  
    }  
}  
  
public Int16 ArithmosDiamerismaton  
{  
    get  
    {  
        return m_ArithmosDiamerismaton;  
    }  
    set  
    {  
        m_ArithmosDiamerismaton = value;  
    }  
}  
  
public Int16 Orofoi  
{  
    get  
    {
```



```
        return m_Orofoi;
    }
    set
    {
        m_Orofoi = value;
    }
}

public bool Asanser
{
    get
    {
        return m_Asanser;
    }
    set
    {
        m_Asanser = value;
    }
}

public bool Kipos
{
    get
    {
        return m_Kipos;
    }
    set
    {
        m_Kipos = value;
    }
}
```

```
public bool autonomithermansi
{
    get
    {
        return m_autonomithermansi;
    }
    set
    {
        m_autonomithermansi = value;
    }
}
```

```
public string tm
{
    get
    {
        return m_tm;
    }
    set
    {
        m_tm = value;
    }
}
```

```
#endregion
```

```
#region public bool Insert()
```

```
public bool Insert()
{
    try
```

```

    {
        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();

        System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();
        cmdInsert.Connection = oConn;

        int _Asanser = 0;
        int _Kipos = 0;
        int _autonomithermansi = 0;

        if(Asanser)
        {
            _Asanser = 1;
        }
        else
        {
            _Asanser = 0;
        }
        if(Kipos)
        {
            _Kipos = 1;
        }
        else
        {
            _Kipos = 0;
        }
        if(autonomithermansi)

```

```

    {
        _autonomithermansi = 1;
    }
    else
    {
        _autonomithermansi = 0;
    }

    cmdInsert.CommandText = "INSERT INTO StoixeiaPolikatikias " +
        "(Address, Number, City, TK ,
Name_admin, Phone_admin, Orofoi, Number_diamerismatos, Ansanser,
Garden, Thermansi, TM_polikatikias) " +
        "VALUES " +
        "(" + this.Address + "," + this.Number + ","
+ this.City + "," + this.TK + "," + this.Diaxiristis + "," + this.PhoneDiaxiristis
+ "," + this.Orofoi + "," + this.ArithmosDiamerismaton + "," + _Asanser +
"," + _Kipos + "," + _autonomithermansi + "," + this.tm + ")";
    cmdInsert.ExecuteNonQuery();

    oConn.Close();

    return true;
}
catch
{
    return false;
}
}

```

```

#endregion

#region public AppLib.polikatikia Check_Stoixeia()

public AppLib.polikatikia Check_Stoixeia()
{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;
    oConn.Open();

    System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
    cmdCheck.Connection = oConn;
    cmdCheck.CommandText = "Select * from StoixeiaPolikatikias";

    //cmdCheck.ExecuteNonQuery();

    System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();
    while (CheckReader.Read())
    {
        this.Address = CheckReader[0].ToString();
        this.Number = CheckReader[1].ToString();
        this.City = CheckReader[2].ToString();
        this.TK = Convert.ToInt16(CheckReader[3]);
        this.Diaxiristis = CheckReader[4].ToString();
        this.PhoneDiaxiristis = CheckReader[5].ToString();
        this.Orofoi = Convert.ToInt16(CheckReader[6]);
        this.ArithmosDiamerismaton = Convert.ToInt16(CheckReader[7]);
    }
}
}

```

```
if (Convert.ToInt16(CheckReader[8]) == 1)
{
    this.Asanser = true;
}
else
{
    this.Asanser = false;
}
if (Convert.ToInt16(CheckReader[9]) == 1)
{
    this.Kipos = true;
}
else
{
    this.Kipos = false;
}
if (Convert.ToInt16(CheckReader[10]) == 1)
{
    this.autonomithermansi = true;
}
else
{
    this.autonomithermansi = false;
}
this.tm = CheckReader[11].ToString();
}
oConn.Close();

return this;

//oConn.Close();
```

```

    }

#endregion

#region public bool Update()

public bool Update()
{
    try
    {
        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();

        System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();
        cmdInsert.Connection = oConn;

        int _Asanser = 0;
        int _Kipos = 0;
        int _autonomithermansi = 0;

        if(Asanser)
        {
            _Asanser = 1;
        }
        else
        {

```

```

        _Asanser = 0;
    }
    if(Kipos)
    {
        _Kipos = 1;
    }
    else
    {
        _Kipos = 0;
    }
    if(autonomithermansi)
    {
        _autonomithermansi = 1;
    }
    else
    {
        _autonomithermansi = 0;
    }

    cmdInsert.CommandText = "Update StoixeiaPolikatikias " +
        "SET Address =" + this.Address +
        ",Number =" + this.Number +
        ",City =" + this.City +
        ",TK =" + this.TK +
        ",Name_admin =" + this.Diaxiristis +
        ",Phone_admin =" + this.PhoneDiaxiristis +
        ",Orofoi =" + this.Orofoi +
        ",Number_diamerismatos =" +
this.ArithmosDiamerismaton +
        ",Ansanser =" + _Asanser +
        ",Garden =" + _Kipos +

```



```

        ",Thermansi =" + _autonomithermansi +
        ",TM_polikatikias =" + this.tm + """;

cmdInsert.ExecuteNonQuery();

oConn.Close();

    return true;
}
catch
{
    return false;
}

}
#endregion

#region public AppLib.polikatikia SelectTM()

public AppLib.polikatikia SelectTM()
{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = DB_Connection.SQLConnection();
    oConn.Open();
    System.Data.SqlClient.SqlCommand cmdSelect = new
System.Data.SqlClient.SqlCommand();
    cmdSelect.Connection = oConn;
    cmdSelect.CommandText = "select TM_polikatikias from
StoixeiaPolikatikias";

```

```

        System.Data.SqlClient.SqlDataReader Reader =
cmdSelect.ExecuteReader();

        while (Reader.Read())
        {
            this.tm = Reader[0].ToString();
        }

        oConn.Close();
        return this;
    }
    #endregion
}
}

```

ParametropiisiDapanon.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace AppLib
{
    public class ParametropiisiDapanon
    {

```

```
#region Δηλώσης μεταβλητών
```

```
private Int16 m_ID;  
private Int16 m_PolikatikiaID;  
private bool m_Garden;  
private bool m_Thermansi;  
private Int16 m_orofoi;  
private bool m_ansaner;  
private Int16 m_Diamerismata;
```

```
#endregion
```

```
#region Constructor
```

```
public ParametropiisiDapanon()  
{  
  
}
```

```
#endregion
```

```
#region Μέθοδοι - Properties
```

```
public Int16 ID  
{  
    get  
    {  
        return m_ID;  
    }  
    set  
    {
```

```

        m_ID = value;
    }
}

public Int16 PolikatikialID
{
    get
    {
        return m_PolikatikialID;
    }
    set
    {
        m_PolikatikialID = value;
    }
}

public bool Garden
{
    get
    {
        return m_Garden;
    }
    set
    {
        m_Garden = value;
    }
}

public bool Thermansi
{
    get

```

```
{  
    return m_Thermansi;  
}  
set  
{  
    m_Thermansi = value;  
}  
}
```

```
public Int16 orofoi  
{  
    get  
    {  
        return m_orofoi;  
    }  
    set  
    {  
        m_orofoi = value;  
    }  
}
```

```
public bool ansanser  
{  
    get  
    {  
        return m_ansaner;  
    }  
    set  
    {  
        m_ansaner = value;  
    }  
}
```

```

    }

    public Int16 Diamerismata
    {
        get
        {
            return m_Diamerismata;
        }
        set
        {
            m_Diamerismata = value;
        }
    }

    #endregion

    #region Insert()

    #endregion
}
}

```

InsDapanes.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

```

```

namespace AppLib
{
    public class InsDapanes
    {
        #region Δηλώσεις

        private string ConnStr;

        private Int16 m_id;
        private string m_Etos;
        private string m_Minas;
        private Int16 m_Dapani;
        private string m_Dapani1;
        private bool m_YpolDapanis;
        private string m_Posos;

        #endregion

        #region Constructor

        public InsDapanes()
        {

        }

        public InsDapanes(string ConnectionString)
        {
            this.ConnStr = ConnectionString;
        }
    }
}

```

```
#endregion
```

```
#region Μέθοδοι - Properties
```

```
public Int16 id
```

```
{  
    get  
    {  
        return m_id;  
    }  
    set  
    {  
        m_id = value;  
    }  
}
```

```
public string Etos
```

```
{  
    get  
    {  
        return m_Etos;  
    }  
    set  
    {  
        m_Etos = value;  
    }  
}
```

```
public string Minas
```

```
{  
    get
```



```
{  
    return m_Minas;  
}  
set  
{  
    m_Minas = value;  
}  
}
```

```
public Int16 Dapani  
{  
    get  
    {  
        return m_Dapani;  
    }  
    set  
    {  
        m_Dapani = value;  
    }  
}
```

```
public string Dapani1  
{  
    get  
    {  
        return m_Dapani1;  
    }  
    set  
    {  
        m_Dapani1 = value;  
    }  
}
```

```
}  
public bool YpoiDapani  
{  
    get  
    {  
        return m_YpoiDapanis;  
    }  
    set  
    {  
        m_YpoiDapanis = value;  
    }  
}  
  
public string poso  
{  
    get  
    {  
        return m_Poso;  
    }  
    set  
    {  
        m_Poso = value;  
    }  
}  
  
#endregion  
  
#region public bool InsertDapani()  
  
public bool InsertDapani()  
{
```

```

try
{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;
    oConn.Open();

    System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();
    cmdInsert.Connection = oConn;
    int YpolDapani1;
    if (this.YpolDapani)
    {
        YpolDapani1 = 1;
    }
    else
    {
        YpolDapani1 = 0;
    }
    cmdInsert.CommandText = "INSERT INTO Dapanes " +

"(Dapanes,YpologismosDapanis,Poso,Etos,Month) " +
        "VALUES " +
        "(" + this.Dapani + "," + YpolDapani1 + ","
+ this.poso + "," + this.Etos + "," + this.Minas + ")";
    cmdInsert.ExecuteNonQuery();

    oConn.Close();

    return true;
}

```

```

        catch
        {
            return false;
        }
    }

#endregion

#region public bool UpdateDapani()

public bool UpdateDapani(int dap_id)
{
    try
    {
        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();

        System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();
        cmdInsert.Connection = oConn;
        int YpolDapani1;
        if (this.YpolDapani)
        {
            YpolDapani1 = 1;
        }
        else
        {
            YpolDapani1 = 0;
        }
    }
}

```

```

    }
    cmdInsert.CommandText = "Update Dapanes " +
        " SET Dapanes = " + this.Dapani +
        ",YpologismosDapanis="+ YpolDapani1 +
        ",Poso=" + this.poso.Replace(",",".") +
        ",Etos=" + this.Etos +
        ",Month=" + this.Minas + "" +
        " WHERE id=" + dap_id;

    cmdInsert.ExecuteNonQuery();

    oConn.Close();

    return true;
}
catch
{
    return false;
}
}

#endregion

#region public System.Collections.ArrayList DatagridFillDapanesList

public System.Collections.ArrayList DatagridFillDapanesList()
{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;

```

```

oConn.Open();

System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
cmdCheck.Connection = oConn;
cmdCheck.CommandText = "SELECT
Dapanes.id,Dapanes_Extra.Dapanes_Extra, Dapanes.Dapanes,
Dapanes.YpologismosDapanis, Dapanes.Poso, Dapanes.Etos,
Dapanes.Month FROM Dapanes INNER JOIN Dapanes_Extra ON
Dapanes.Dapanes = Dapanes_Extra.id";

System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();

ArrayList DapaniList = new ArrayList();

while (CheckReader.Read())
{
AppLib.InsDapanes cmdDapanes = new InsDapanes();

cmdDapanes.id = Convert.ToInt16(CheckReader[0]);

cmdDapanes.Dapani1 = CheckReader[1].ToString().Trim();
cmdDapanes.Dapani = Convert.ToInt16(CheckReader[2]);
if (Convert.ToInt16(CheckReader[3]) == 1)
{
cmdDapanes.YpolDapani = true;
}
else
{

```

```

        cmdDapanes.YpoIDapani = false;
    }
    cmdDapanes.poso = CheckReader[4].ToString();
    cmdDapanes.Etos = CheckReader[5].ToString();
    cmdDapanes.Minas = CheckReader[6].ToString();

    DapaniList.Add(cmdDapanes);
}

oConn.Close();

return DapaniList;

}

#endregion

#region public System.Collections.ArrayList
DatagridFillDapanesListParam

public System.Collections.ArrayList
DatagridFillDapanesListParam(string Year, string Month)
{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;
    oConn.Open();

    System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
    cmdCheck.Connection = oConn;

```

```
cmdCheck.CommandText = "SELECT
Dapanes.id,Dapanes_Extra.Dapanes_Extra, Dapanes.Dapanes,
Dapanes.YpologismosDapanis, Dapanes.Poso, Dapanes.Etos,
Dapanes.Month FROM Dapanes INNER JOIN Dapanes_Extra ON
Dapanes.Dapanes = Dapanes_Extra.id WHERE Dapanes.Etos =" + Year +
" AND Dapanes.Month=" + Month + """;
```

```
System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();
```

```
ArrayList DapaniList = new ArrayList();
```

```
while (CheckReader.Read())
```

```
{
```

```
AppLib.InsDapanes cmdDapanes = new InsDapanes();
```

```
cmdDapanes.id = Convert.ToInt16(CheckReader[0]);
```

```
cmdDapanes.Dapani1 = CheckReader[1].ToString().Trim();
```

```
cmdDapanes.Dapani = Convert.ToInt16(CheckReader[2]);
```

```
if (Convert.ToInt16(CheckReader[3]) == 1)
```

```
{
```

```
cmdDapanes.YpolDapani = true;
```

```
}
```

```
else
```

```
{
```

```
cmdDapanes.YpolDapani = false;
```

```
}
```

```
cmdDapanes.poso = CheckReader[4].ToString();
```

```
cmdDapanes.Etos = CheckReader[5].ToString();
```



```
        cmdDapanes.Minas = CheckReader[6].ToString();

        DapaniList.Add(cmdDapanes);
    }

    oConn.Close();

    return DapaniList;

}

#endregion

}
}
```

diamerisma.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace AppLib
{
    public class diamerisma
    {
        #region Δηλώσεις
```

```
private string ConnStr;

private string m_Orofos;
private Int16 m_Diamerisma;
private string m_Tetragonika;
private string m_note;
private string m_Idioktitis;
private string m_phoneldioktitis;
private string m_Enoikiastis;
private string m_phoneEnoikiastis;
private string m_ei;
private string m_fi;
private bool m_Ansanser;
private bool m_Garden;
private bool m_Kleistodiamerisma;

#endregion

#region Constructor

public diamerisma()
{
}

public diamerisma(string ConnectionString)
{
    this.ConnStr = ConnectionString;
}
}
```

```
#endregion
```

```
#region Μέθοδοι - Properties
```

```
public string Orofos  
{  
    get  
    {  
        return m_Orofos;  
    }  
    set  
    {  
        m_Orofos = value;  
    }  
}
```

```
public Int16 Diamerisma  
{  
    get  
    {  
        return m_Diamerisma;  
    }  
    set  
    {  
        m_Diamerisma = value;  
    }  
}
```

```
public string Tetragonika  
{  
    get
```

```
{  
    return m_Tetragonika;  
}  
set  
{  
    m_Tetragonika = value;  
}  
}
```

```
public string note  
{  
    get  
    {  
        return m_note;  
    }  
    set  
    {  
        m_note = value;  
    }  
}
```

```
public string Idioktitis  
{  
    get  
    {  
        return m_Idioktitis;  
    }  
    set  
    {  
        m_Idioktitis = value;  
    }  
}
```

```
}
```

```
public string phoneldioktitis  
{  
    get  
    {  
        return m_phoneldioktitis;  
    }  
    set  
    {  
        m_phoneldioktitis = value;  
    }  
}
```

```
public string Enoikiastis  
{  
    get  
    {  
        return m_Enoikiastis;  
    }  
    set  
    {  
        m_Enoikiastis = value;  
    }  
}
```

```
public string phoneNoikiastis  
{  
    get  
    {  
        return m_phoneNoikiastis;
```

```
    }  
    set  
    {  
        m_phoneEnoikiastis = value;  
    }  
}
```

```
public string ei  
{  
    get  
    {  
        return m_ei;  
    }  
    set  
    {  
        m_ei = value;  
    }  
}
```

```
public string fi  
{  
    get  
    {  
        return m_fi;  
    }  
    set  
    {  
        m_fi = value;  
    }  
}
```

```
}
```

```
public bool Ansanser
```

```
{
```

```
    get
```

```
    {
```

```
        return m_Ansanser;
```

```
    }
```

```
    set
```

```
    {
```

```
        m_Ansanser = value;
```

```
    }
```

```
}
```

```
public bool Garden
```

```
{
```

```
    get
```

```
    {
```

```
        return m_Garden;
```

```
    }
```

```
    set
```

```
    {
```

```
        m_Garden = value;
```

```
    }
```

```
}
```

```
public bool Kleistodiamerisma
```

```
{
```

```
    get
```

```
    {
```

```
        return m_Kleistodiamerisma;
```

```

    }
    set
    {
        m_Kleistodiamerisma = value;
    }
}

#endregion

#region public int Insert()

public bool Insert()
{
    try
    {
        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();

        System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();
        cmdInsert.Connection = oConn;

        int asanser = 0;
        if (this.Ansanser)
        {
            asanser = 1;
        }
        else
        {

```



```

        ansenser = 0;
    }

    int kipos = 0;
    if (this.Garden)
    {
        kipos = 1;
    }
    else
    {
        kipos = 0;
    }

    int closediamerisma = 0;
    if (this.Kleistodiamerisma)
    {
        closediamerisma = 1;
    }
    else
    {
        closediamerisma = 0;
    }

    cmdInsert.CommandText = "INSERT INTO StoixeiaDiamerismatos
" +
        "(Orofos, Diamerisma, TMDiamerismatos,
Notes, Nameldioktitis, Phoneldioktitis, NameEnikiasti, PhoneEnikiasti, ei, fi,
Ansanser, Garden, CloseDiamerisma) " +
        "VALUES " +
        "(" + this.Orofos + "," + this.Diamerisma +
", " + this.Tetragonika + "," + this.note + "," + this.Idioktitis + "," +

```

```

this.phoneldioktitis + "," + this.Enoikiastis + "," + this.phoneEnoikiastis +
"," + this.ei + "," + this.fi + "," + asanser + "," + kipos + "," +
closediamerisma + "));
        cmdInsert.ExecuteNonQuery();

        oConn.Close();

        return true;
    }
    catch
    {
        return false;
    }
}

#endregion

#region public bool Update()

public bool Update(string _orofos, Int16 _diamerisma)
{
    try
    {
        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();
    }
}

```

```
System.Data.SqlClient.SqlCommand cmdInsert = new  
System.Data.SqlClient.SqlCommand();  
cmdInsert.Connection = oConn;
```

```
int asanser = 0;  
if (this.Ansanser)  
{  
    asanser = 1;  
}  
else  
{  
    asanser = 0;  
}
```

```
int kijos = 0;  
if (this.Garden)  
{  
    kijos = 1;  
}  
else  
{  
    kijos = 0;  
}
```

```
int closediamerisma = 0;  
if (this.Kleistodiamerisma)  
{  
    closediamerisma = 1;  
}  
else  
{
```

```

        closediamerisma = 0;
    }

    cmdInsert.CommandText = "Update StoixeiaDiamerismatos " +
        "SET TMDiamerismatos =" +
this.Tetragonika +
        ",Notes =" + this.note +
        ",Nameldioktitis =" + this.Idioktitis +
        ",Phoneldioktitis =" + this.phoneldioktitis +
        ",NameEnikiasti =" + this.Enoikiastis +
        ",PhoneEnikiasti =" + this.phoneEnoikiastis
+
        ",ei =" + this.ei +
        ",fi =" + this.fi +
        ",Ansanser =" + ansanser +
        ",Garden =" + kipos +
        ",CloseDiamerisma =" + closediamerisma +
        " WHERE Orofos =" + _orofos + " AND
Diamerisma =" + _diamerisma +"";

    cmdInsert.ExecuteNonQuery();

    oConn.Close();

    return true;
}
catch
{
    return false;
}
}

```

```

#endregion

#region SelectCheck()

public bool Select_Check(string _orofos, Int16 _diamerisma)
{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;
    oConn.Open();

    System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
    cmdCheck.Connection = oConn;
    cmdCheck.CommandText = "Select * from StoixeiaDiamerismatos
WHERE Orofos =" + _orofos + " AND Diamerisma =" + _diamerisma;

    //cmdCheck.ExecuteNonQuery();
    bool flagCheckDiamerisma = false;

    System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();
    while (CheckReader.Read())
    {
        //this.ID = Convert.ToInt16(CheckReader[0]);
        this.Orofos = CheckReader[0].ToString();
        this.Diamerisma = Convert.ToInt16(CheckReader[1]);
        this.Tetragonika = CheckReader[2].ToString();
        this.note = CheckReader[3].ToString();
        this.Idioktitis = CheckReader[4].ToString();
    }
}

```

```

this.phoneldioktitis = CheckReader[5].ToString();
this.Enokiastis = CheckReader[6].ToString();
this.phoneEnokiastis = CheckReader[7].ToString();
this.ei = CheckReader[8].ToString();
this.fi = CheckReader[9].ToString();
if (Convert.ToInt16(CheckReader[10]) == 1)
{
    this.Ansanser = true;
}
else
{
    this.Ansanser = false;
}
if (Convert.ToInt16(CheckReader[11]) == 1)
{
    this.Garden = true;
}
else
{
    this.Garden = false;
}
if (Convert.ToInt16(CheckReader[12]) == 1)
{
    this.Kleistodiamerisma = true;
}
else
{
    this.Kleistodiamerisma = false;
}
flagCheckDiamerisma = true;
}

```

```

oConn.Close();

if (flagCheckDiamerisma)
{
    return false;
}
else
{
    return true;
}
}
#endregion

#region public System.Collections.ArrayList DatagridFillDiamerismaList

public System.Collections.ArrayList DatagridFillDiamerismaList()
{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;
    oConn.Open();

    System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
    cmdCheck.Connection = oConn;
    cmdCheck.CommandText = "Select * from StoixeiaDiamerismatos
order by StoixeiaDiamerismatos.Orofos";

    System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();

```

```

ArrayList DiamerismaList = new ArrayList();

while (CheckReader.Read())
{
    AppLib.diamerisma cmdDiamerisma = new diamerisma();

    cmdDiamerisma.Orofos = CheckReader[0].ToString().Trim() +
CheckReader[1].ToString();
    cmdDiamerisma.Diamerisma =
Convert.ToInt16(CheckReader[1].ToString());
    cmdDiamerisma.Tetragonika = CheckReader[2].ToString();
    cmdDiamerisma.note = CheckReader[3].ToString();
    cmdDiamerisma.Idioktitis = CheckReader[4].ToString();
    cmdDiamerisma.phonIdioktitis = CheckReader[5].ToString();
    cmdDiamerisma.Enoikiastis = CheckReader[6].ToString();
    cmdDiamerisma.phoneEnoikiastis = CheckReader[7].ToString();
    cmdDiamerisma.ei = CheckReader[8].ToString();
    cmdDiamerisma.fi = CheckReader[9].ToString();
    if (Convert.ToInt16(CheckReader[10]) == 1)
    {
        cmdDiamerisma.Ansanser = true;
    }
    else
    {
        cmdDiamerisma.Ansanser = false;
    }
    if (Convert.ToInt16(CheckReader[11]) == 1)
    {
        cmdDiamerisma.Garden = true;
    }
}

```



```

else
{
    cmdDiamerisma.Garden = false;
}
if (Convert.ToInt16(CheckReader[12]) == 1)
{
    cmdDiamerisma.Kleistodiamerisma = true;
}
else
{
    cmdDiamerisma.Kleistodiamerisma = false;
}

DiamerismaList.Add(cmdDiamerisma);
}

oConn.Close();

return DiamerismaList;

}

#endregion

#region public System.Collections.ArrayList
DatagridFillDiamerismaList(string diamerisma1)

public System.Collections.ArrayList DatagridFillDiamerismaList(string
diameterisma1)
{
    string orofos = "";

```

```
int diamer=0;

switch (diamerisma1)
{
    case "IΣ1" :
        orofos = "IΣ";
        diamer = 1;
        break;
    case "IΣ2":
        orofos = "IΣ";
        diamer = 2;
        break;
    case "A1":
        orofos = "A";
        diamer = 1;
        break;
    case "A2":
        orofos = "A";
        diamer = 2;
        break;
    case "A3":
        orofos = "A";
        diamer = 3;
        break;
    case "B1":
        orofos = "B";
        diamer = 1;
        break;
    case "B2":
        orofos = "B";
        diamer = 2;
```

```
    break;
case "B3":
    orofos = "B";
    diamer = 3;
    break;
case "Γ1":
    orofos = "Γ";
    diamer = 1;
    break;
case "Γ2":
    orofos = "Γ";
    diamer = 2;
    break;
case "Γ3":
    orofos = "Γ";
    diamer = 3;
    break;
case "Δ1":
    orofos = "Δ";
    diamer = 1;
    break;
case "Δ2":
    orofos = "Δ";
    diamer = 2;
    break;
case "Δ3":
    orofos = "Δ";
    diamer = 3;
    break;
case "E1":
    orofos = "E";
```

```

        diamer = 1;
        break;
    case "E2":
        orofos = "E";
        diamer = 2;
        break;
    case "E3":
        orofos = "E";
        diamer = 3;
        break;
}

```

```

        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();

```

```

        System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
        cmdCheck.Connection = oConn;
        cmdCheck.CommandText = "Select * from StoixeiaDiamerismatos
WHERE Orofos = " + orofos + " AND Diamerisma = " + diamer;

```

```

        System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();

```

```

        ArrayList DiamerismaList = new ArrayList();

```

```

        while (CheckReader.Read())
        {
            AppLib.diamerisma cmdDiamerisma = new diamerisma();

```

```

        cmdDiamerisma.Orofos = CheckReader[0].ToString().Trim() +
CheckReader[1].ToString();
        cmdDiamerisma.Diamerisma =
Convert.ToInt16(CheckReader[1].ToString());
        cmdDiamerisma.Tetragonika = CheckReader[2].ToString();
        cmdDiamerisma.note = CheckReader[3].ToString();
        cmdDiamerisma.Idioktitis = CheckReader[4].ToString();
        cmdDiamerisma.phonIdioktitis = CheckReader[5].ToString();
        cmdDiamerisma.Enoikiastis = CheckReader[6].ToString();
        cmdDiamerisma.phoneEnoikiastis = CheckReader[7].ToString();
        cmdDiamerisma.ei = CheckReader[8].ToString();
        cmdDiamerisma.fi = CheckReader[9].ToString();
        if (Convert.ToInt16(CheckReader[10]) == 1)
        {
            cmdDiamerisma.Ansanser = true;
        }
        else
        {
            cmdDiamerisma.Ansanser = false;
        }
        if (Convert.ToInt16(CheckReader[11]) == 1)
        {
            cmdDiamerisma.Garden = true;
        }
        else
        {
            cmdDiamerisma.Garden = false;
        }
        if (Convert.ToInt16(CheckReader[12]) == 1)
        {

```

```
        cmdDiamerisma.Kleistodiamerisma = true;
    }
    else
    {
        cmdDiamerisma.Kleistodiamerisma = false;
    }

    DiamerismaList.Add(cmdDiamerisma);
}

oConn.Close();

return DiamerismaList;
}

#endregion

}
}
```

DapaniTherm.cs

```
using System;
using System.Collections.Generic;
```

```

using System.Linq;
using System.Text;
using System.Collections;

namespace AppLib
{
    public class DapaniTherm
    {
        #region Δηλώσεις

        string ConnStr;
        Int32 m_id;
        string m_Etos;
        string m_Minas;
        Int32 m_Diamerisma;
        string m_diamerisma1;
        string m_Metrisi;

        string Diamer;
        #endregion

        #region Constructor

        public DapaniTherm()
        {
        }

        public DapaniTherm(string ConnectionString)
        {
            this.ConnStr = ConnectionString;
        }
    }
}

```

```
}
```

```
#endregion
```

```
#region Properties
```

```
public Int32 id
```

```
{
```

```
    set
```

```
    {
```

```
        m_id = value;
```

```
    }
```

```
    get
```

```
    {
```

```
        return m_id;
```

```
    }
```

```
}
```

```
public string Etos
```

```
{
```

```
    set
```

```
    {
```

```
        m_Etos = value;
```

```
    }
```

```
    get
```

```
    {
```

```
        return m_Etos;
```

```
    }
```

```
}
```

```
public string Minas
```



```
{
    set
    {
        m_Minas = value;
    }
    get
    {
        return m_Minas;
    }
}

public Int32 Diamerisma
{
    set
    {
        m_Diamerisma = value;
    }
    get
    {
        return m_Diamerisma;
    }
}

public string Diamerisma1
{
    set
    {
        m_diamerisma1 = value;
    }
    get
    {
```

```

        return m_diamerisma1;
    }
}

public string Metrasi
{
    set
    {
        m_Metrasi = value;
    }
    get
    {
        return m_Metrasi;
    }
}

#endregion

#region InsertDapanTherm()

public bool InsertDapanTherm()
{
    try
    {
        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();

        System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();

```

```

cmdInsert.Connection = oConn;

cmdInsert.CommandText = "INSERT INTO DapanesThermansi
(Etos_Therm,Month_Therm,CodeDiamerismatos,Metrisi) VALUES (" +
this.Etos + "," + this.Minas + "," + this.Diamerisma + "," + this.Metrisi + ")";
cmdInsert.ExecuteNonQuery();

oConn.Close();

return true;
}
catch
{
return false;
}
}

#endregion

#region UpdateDapanTherm()

public bool UpdateDapanTherm(int updateid)
{
try
{
System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
oConn.ConnectionString = this.ConnStr;
oConn.Open();

```

```

        System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();
        cmdInsert.Connection = oConn;

        cmdInsert.CommandText = "Update DapanesThermansi SET
Etos_Therm = " + this.Etos +
                                ", Month_Therm = " +
this.Minas +
                                ", CodeDiamerismatos = " +
this.Diamerisma +
                                ", Metrasi = " +
this.Metrasi.Replace(", ", ".") + " WHERE id = " + updateid + """;
        cmdInsert.ExecuteNonQuery();

        oConn.Close();

        return true;
    }
    catch
    {
        return false;
    }
}

#endregion

#region public System.Collections.ArrayList
DatagridFillDapanesThermList

public System.Collections.ArrayList DatagridFillDapanesThermList()

```

```

{

    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;
    oConn.Open();

    System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
    cmdCheck.Connection = oConn;
    cmdCheck.CommandText = "Select * from DapanesThermansi";

    System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();

    ArrayList DiamerismaList = new ArrayList();

    while (CheckReader.Read())
    {
        AppLib.DapaniTherm cmdDapaniTherm = new DapaniTherm();

        cmdDapaniTherm.id = Convert.ToInt32(CheckReader[0]);
        cmdDapaniTherm.Etos = CheckReader[1].ToString();
        cmdDapaniTherm.Minas = CheckReader[2].ToString();
        switch (Convert.ToInt32(CheckReader[3].ToString()))
        {
            case 0:
                Diamer = "IΣ1";
                break;
            case 1:

```

```
    Diamer = "IΣ2";  
    break;  
case 2:  
    Diamer = "A1";  
    break;  
case 3:  
    Diamer = "A2";  
    break;  
case 4:  
    Diamer = "A3";  
    break;  
case 5:  
    Diamer = "B1";  
    break;  
case 6:  
    Diamer = "B2";  
    break;  
case 7:  
    Diamer = "B3";  
    break;  
case 8:  
    Diamer = "Γ1";  
    break;  
case 9:  
    Diamer = "Γ2";  
    break;  
case 10:  
    Diamer = "Γ3";  
    break;  
case 11:  
    Diamer = "Δ1";
```

```

        break;
    case 12:
        Diamer = "Δ2";
        break;
    case 13:
        Diamer = "Δ3";
        break;
    case 14:
        Diamer = "E1";
        break;
    case 15:
        Diamer = "E2";
        break;
    case 16:
        Diamer = "E3";
        break;
    default:

        break;
}
cmdDapaniTherm.Diamerisma1 = Diamer;
cmdDapaniTherm.Metrisi = CheckReader[4].ToString();

DiamerismaList.Add(cmdDapaniTherm);
}

oConn.Close();

return DiamerismaList;

```

```

    }

    #endregion

    #region public System.Collections.ArrayList
    DatagridFillDapanesThermList

    public System.Collections.ArrayList
    DatagridFillDapanesThermList(Int32 diamer)
    {
        System.Data.SqlClient.SqlConnection oConn = new
    System.Data.SqlClient.SqlConnection();
        oConn.ConnectionString = this.ConnStr;
        oConn.Open();

        System.Data.SqlClient.SqlCommand cmdCheck = new
    System.Data.SqlClient.SqlCommand();
        cmdCheck.Connection = oConn;
        cmdCheck.CommandText = "Select * from DapanesThermansi
    WHERE CodeDiamerismatos = " + diamer + """;

        System.Data.SqlClient.SqlDataReader CheckReader =
    cmdCheck.ExecuteReader();

        ArrayList DiamerismaList = new ArrayList();

        while (CheckReader.Read())
        {
            AppLib.DapaniTherm cmdDapaniTherm = new DapaniTherm();

            cmdDapaniTherm.id = Convert.ToInt32(CheckReader[0]);

```



```
cmdDapaniTherm.Etos = CheckReader[1].ToString();
cmdDapaniTherm.Minas = CheckReader[2].ToString();
switch (Convert.ToInt32(CheckReader[3].ToString()))
{
    case 1:
        Diamer = "IΣ1";
        break;
    case 2:
        Diamer = "IΣ2";
        break;
    case 3:
        Diamer = "A1";
        break;
    case 4:
        Diamer = "A2";
        break;
    case 5:
        Diamer = "A3";
        break;
    case 6:
        Diamer = "B1";
        break;
    case 7:
        Diamer = "B2";
        break;
    case 8:
        Diamer = "B3";
        break;
    case 9:
        Diamer = "Γ1";
        break;
}
```

```

case 10:
    Diamer = "Γ2";
    break;
case 11:
    Diamer = "Γ3";
    break;
case 12:
    Diamer = "Δ1";
    break;
case 13:
    Diamer = "Δ2";
    break;
case 14:
    Diamer = "Δ3";
    break;
case 15:
    Diamer = "E1";
    break;
case 16:
    Diamer = "E2";
    break;
case 17:
    Diamer = "E3";
    break;
default:

    break;
}
cmdDapaniTherm.Diamerisma1 = Diamer;
cmdDapaniTherm.Metrisi = CheckReader[4].ToString();

```

```
        DiamerismaList.Add(cmdDapaniTherm);
    }

    oConn.Close();

    return DiamerismaList;
}

#endregion

}
}
```

Dapanes.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace AppLib
{
    public class Dapanes
    {
        #region Δηλώσεις
```

```
private string ConnStr;
private int m_id;
private string m_dapanes;

#endregion

#region Constructor

public Dapanes()
{

}

public Dapanes(string ConnectionString)
{
    this.ConnStr = ConnectionString;
}

#endregion

#region Μέθοδοι - Propeties

public int id
{
    get
    {
        return m_id;
    }
    set
    {
```

```

        m_id = value;
    }
}

public string dapanes
{
    get
    {
        return m_dapanes;
    }
    set
    {
        m_dapanes = value;
    }
}

```

```
#endregion
```

```
#region public System.Collections.ArrayList CreateCategoryList()
```

```
public System.Collections.ArrayList CreateDapanesList()
```

```

{
    System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();
    oConn.ConnectionString = this.ConnStr;
    oConn.Open();

    System.Data.SqlClient.SqlCommand cmdCheck = new
System.Data.SqlClient.SqlCommand();
    cmdCheck.Connection = oConn;
    cmdCheck.CommandText = "Select * from Dapanes_Extra";
}

```

```

        System.Data.SqlClient.SqlDataReader CheckReader =
cmdCheck.ExecuteReader();

        ArrayList DapanesList = new ArrayList();

        while (CheckReader.Read())
        {
            AppLib.Dapanes cmdDapanes = new Dapanes();

            cmdDapanes.id = Convert.ToInt16(CheckReader[0]);
            cmdDapanes.dapanes = CheckReader[1].ToString().Trim();

            DapanesList.Add(cmdDapanes);
        }

        oConn.Close();

        return DapanesList;
    }

#endregion

#region InsertExrtaDapani

public bool InsertExtraDapani()
{
    try
    {
        System.Data.SqlClient.SqlConnection oConn = new
System.Data.SqlClient.SqlConnection();

```

```

oConn.ConnectionString = this.ConnStr;
oConn.Open();

System.Data.SqlClient.SqlCommand cmdInsert = new
System.Data.SqlClient.SqlCommand();
cmdInsert.Connection = oConn;

cmdInsert.CommandText = "INSERT INTO Dapanes_Extra
(Dapanes_Extra) VALUES (" + this.dapanes + ")";
cmdInsert.ExecuteNonQuery();

oConn.Close();

return true;
}
catch
{
return false;
}

}

#endregion

}
}

```

ActiveForm.cs

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace AppLib
{
    public class ActiveForm
    {
        public bool ChkIfActive(string FormName)
        {
            foreach (Form childForm in
Application.OpenForms["frmStart"].MdiChildren)
            {
                if (FormName == childForm.Name.ToString())
                {
                    return true;
                }
            }
            return false;
        }
    }
}
```


Βιβλιογραφία

- [1] MICHAELIS , εκδόσης Γκιούρδας «ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ C#»
- [2] Herbert,Schildt , McGraw-Hill Education - Europe «C#»
- [3] Susan,Harkins, Martin W.P.,Reid , Apress «SQL»
- [4] Neil,Pike , Apress «SQL Server»
- [5] <http://msdn.microsoft.com/el-gr/>
- [6] www.csharp-station.com/Tutorial.aspx
- [7] <http://www.w3schools.com/sql/default.asp>
- [8] <http://www.studentguru.gr>
- [9] <http://www.dotnetzone.gr>
- [10] <http://www.sql-tutorial.net/>