

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος «Εκπαιδευτικό Λογισμικό»
Εγχειρίδιο Ανάλυσης και Σχεδιασμού της εφαρμογής

Απαλλακτική Εργασία	
Όνομα φοιτητή – Αρ. Μητρώου	Αντωνάκη Ελευθερία Π13005
	Μπερούκας Ευστάθιος Π13096
Ημερομηνία παράδοσης	27/9/2021



Εκφώνηση της άσκησης

Ζητείται να γίνει ένα αλληλεπιδραστικό λογισμικό εκπαίδευσης μαθητών για ένα από τα παρακάτω πεδία διδασκαλίας:

Εκπαιδευτικό παιχνίδι για την εκμάθηση της προπαίδειας των Μαθηματικών

Η εργασία αυτή θα περιλαμβάνει τρόπους παρουσίασης της διδακτικού υλικού με στόχο να γίνει το θέμα κατανοητό και να μπορεί να απομνημονευθεί από τους μαθητές και να

εμπεδωθεί η ύλη μέσω ασκήσεων. Ο κύριος σκοπός της εργασίας είναι ο καλός σχεδιασμός

και υλοποίηση του εκπαιδευτικού λογισμικού (διδασκαλία – αξιολόγηση του μαθητή) και όχι

η εισαγωγή μεγάλου μέρους κεφαλαίων.



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Εγχειρίδιο ανάλυσης και σχεδιασμού της εφαρμογής.....	4
1.1	Φόρμα 1 (Form1) – Σύνδεση:.....	4
1.2	Φόρμα 2 (Form2) – Εγγραφή:.....	5
1.3	Φόρμα 3 (Form3) – Welcome:.....	7
1.4	Φόρμα 4 (Form4) – Μαθήματα:	7
1.5	Φόρμα 5 (Form5) – Ασκήσεις:	8
1.6	Φόρμα 6 (Form6) – Εξέταση:	9
1.7	Φόρμα 7 (Form7) – Βαθμολογίες:	13
1.8	Φόρμα 8 (Form8) – Ενότητες:	14



1 Εγχειρίδιο ανάλυσης και σχεδιασμού της εφαρμογής

Υλοποιήσαμε μια εφαρμογή με όνομα Multiplication που διαθέτει μαθήματα προπαίδειας και ασκήσεις πολλαπλασιασμού. Για την υλοποίηση χρησιμοποιήσαμε το Visual Studio Community 2017 σε C#. Το πρόγραμμα χρησιμοποιεί φόρμες που δημιουργήθηκαν μέσω windows forms application.

1.1 Φόρμα 1 (Form1) – Σύνδεση:

Η Form1 είναι η πρώτη επαφή του χρήστη με το πρόγραμμα. Ο χρήστης καλείται να συνδεθεί στον λογαριασμό του γράφοντας το username και το password του στα αντίστοιχα πεδία και πατώντας το κουμπί «Σύνδεση». Αν τα στοιχεία που συμπλήρωσε ταιριάζουν με τα στοιχεία ενός υπάρχοντος λογαριασμού, τότε ανοίγει η Form3, που αποτελεί το κεντρικό μενού της εφαρμογής.

Η εφαρμογή ελέγχει αν τα πεδία που συμπληρώθηκαν αντιστοιχούν σε κάποιο entry της βάσης δεδομένων των χρηστών.

Αν δεν αντιστοιχούν τότε βγαίνει μήνυμα λάθους. Αν αντιστοιχούν πραγματοποιείται η σύνδεση και προχωρά στην Form3.

Να σημειωθεί πως η βάση δεδομένων είναι τοπική και χρειάζεται αλλαγή του path στον παρακάτω κώδικα.

```
private void button1_Click(object sender, EventArgs e)
{
    if (isValid())
    {
        using (SqlConnection conn = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Stathis\Source\Repos\elef-
ant\Multiplication\Multiplication\Database1.mdf;Integrated Security=True"))
        {
            string query = "SELECT * FROM LoginTable WHERE USERNAME = '" +
txtusername.Text.Trim() + "' AND PASSWORD = '" + txtpassword.Text.Trim() + "'";
            SqlDataAdapter sda = new SqlDataAdapter(query, conn);
            DataTable dta = new DataTable();
            sda.Fill(dta);

            if (dta.Rows.Count == 1)
            {
                query = "SELECT id FROM LoginTable WHERE USERNAME = '" +
txtusername.Text.Trim() + "'";
                sda = new SqlDataAdapter(query, conn);
                sda.Fill(dta);

                Program.activeID =
Convert.ToInt32(dta.Rows[0]["id"].ToString());
                Program.activeUser = txtusername.Text.Trim();
            }
        }
    }
}
```



```
        this.Hide();  
        MainPage f3 = new MainPage();  
        f3.Show();  
    }  
    else  
    {  
        MessageBox.Show("Δεν βρέθηκε χρήστης με αυτά τα στοιχεία.  
Παρακαλώ προσπαθήστε ξανά ή εγγραφείτε ως νέος χρήστης.", "Error!");  
    }  
}  
}
```

Αν ο χρήστης δεν έχει ήδη λογαριασμό τότε μπορεί να πατήσει στο κουμπί «Νέος Λογαριασμός». Τότε ανοίγει η Form2 για να κάνει εγγραφή.

Η Form1, όπως και κάθε φόρμα στο πρόγραμμα, διαθέτει ένα κουμπί «Βοήθεια», που εξηγεί τις επιλογές της φόρμας στον χρήστη.

1.2 Φόρμα 2 (Form2) – Εγγραφή:

Η Form2 είναι η φόρμα εγγραφής. Ο χρήστης καλείται να συμπληρώσει τα στοιχεία του νέου του λογαριασμού στα πεδία email, username, password και επιβεβαίωση password και πατώντας το κουμπί «Δημιουργία Λογαριασμού».

Η φόρμα ελέγχει τα στοιχεία που έχουν εισαχθεί και αν δεν υπάρχει ήδη λογαριασμός με αυτό το username ή το email και αν τα δύο password που εισήχθησαν είναι ίδια, τότε δημιουργείται ο νέος χρήστης.

Όπως και πριν, η εφαρμογή ελέγχει την βάση δεδομένων για να δει αν τα συμπληρωμένα πεδία πληρούν τις προϋποθέσεις. Αν όχι, τότε βγαίνει αντίστοιχο μήνυμα και ο χρήστης δεν δημιουργείται.

Παράλληλα με τον νέο χρήστη, δημιουργείται entry στον πίνακα ErrorTable της βάσης δεδομένων, στον οποίο και θα καταχωρηθούν αργότερα τα αποτελέσματα των ασκήσεων που θα κάνει ο χρήστης. Προς το παρόν, απλά τον συμπληρώνουμε με 0.

```
private void RegisterButton_Click(object sender, EventArgs e)  
{  
    using (SqlConnection conn = new SqlConnection(@"Data  
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Stathis\Source\Repos\elef-  
ant\Multiplication\Multiplication\Database1.mdf;Integrated Security=True"))  
    {  
        if (IsValid())  
        {  
            if (txtconfpassword.Text.Trim() == txtpassword.Text.Trim())  
            {  
                string query = "SELECT * FROM LoginTable WHERE USERNAME = '"  
+ txtusername.Text.Trim() + "' OR EMAIL = '" + txtemail.Text.Trim() + "'";  
                SqlDataAdapter sda = new SqlDataAdapter(query, conn);  
                DataTable dta = new DataTable();  
                sda.Fill(dta);  
  
                if (dta.Rows.Count == 1)  
                {
```



```
        MessageBox.Show("Ο χρήστης υπάρχει ήδη. Παρακαλώ κάντε  
είσοδο στην εφαρμογή ή δημιουργείστε νέο λογαριασμό", "Error!");  
    }  
    else  
    {  
        conn.Open();  
        query = "INSERT INTO LoginTABLE  
VALUES(@username,@password,@email)";  
        cmd = new SqlCommand(query, conn);  
        cmd.Parameters.AddWithValue("username",  
txtusername.Text);  
        cmd.Parameters.AddWithValue("password",  
txtpassword.Text);  
        cmd.Parameters.AddWithValue("email", txtemail.Text);  
        cmd.ExecuteNonQuery();  
        MessageBox.Show("Your Account is created . Please login  
now.", "Done!", MessageBoxButtons.OK, MessageBoxIcon.Information);  
  
        query = "INSERT INTO ErrorTable " +  
"(ec1,ec2,ec3,ec4,ec5,ec6,ec7,ec8,ec9,ec10,corrects,total) " +  
"VALUES(@ec1,@ec2,@ec3,@ec4,@ec5,@ec6,@ec7,@ec8,@ec9,@ec10,@corrects,@total)";  
        cmd = new SqlCommand(query, conn);  
        cmd.Parameters.AddWithValue("ec1", 0);  
        cmd.Parameters.AddWithValue("ec2", 0);  
        cmd.Parameters.AddWithValue("ec3", 0);  
        cmd.Parameters.AddWithValue("ec4", 0);  
        cmd.Parameters.AddWithValue("ec5", 0);  
        cmd.Parameters.AddWithValue("ec6", 0);  
        cmd.Parameters.AddWithValue("ec7", 0);  
        cmd.Parameters.AddWithValue("ec8", 0);  
        cmd.Parameters.AddWithValue("ec9", 0);  
        cmd.Parameters.AddWithValue("ec10", 0);  
        cmd.Parameters.AddWithValue("corrects", 0);  
        cmd.Parameters.AddWithValue("total", 0);  
        cmd.ExecuteNonQuery();  
    }  
}  
else  
{  
    MessageBox.Show("Ο κωδικός πρόσβασης δεν επιβεβαιώθηκε.  
Παρακαλώ προσπαθήστε ξανά", "Error!");  
}  
}  
  
private bool isValid()  
{  
    if (txtpassword.Text.TrimStart() == string.Empty ||  
txtusername.Text.TrimStart() == string.Empty || txtemail.Text.TrimStart() ==  
string.Empty || txtconfpassword.Text.TrimStart() == string.Empty)  
    {  
        MessageBox.Show("Παρακαλούμε συμπληρώστε όλα τα ζητούμενα στοιχεία",  
"Error!");  
        return false;  
    }  
}
```



```
    }  
    return true;  
}
```

Υπάρχει και σε αυτήν την φόρμα κουμπί βοήθειας, καθώς και το κουμπί «Έχω ήδη λογαριασμό» που επιστρέφει τον χρήστη στην φόρμα σύνδεσης (Form1).

1.3 Φόρμα 3 (Form3) – Welcome:

Η Form3 αποτελεί το κεντρικό μενού του προγράμματος και για αυτόν τον λόγο δεν έχει κάποια λειτουργία πέρα από την επιλογή της επόμενης φόρμας μέσω των κουμπιών «Μαθήματα», «Ασκήσεις», «Βαθμολογίες», «Αποσύνδεση» και «Έξοδος».

Πατώντας στο «Μαθήματα» ο χρήστης πηγαίνει στην Form4, που περιέχει τα μαθήματα ανά ενότητα. Πατώντας στο κουμπί «Ασκήσεις» τότε πηγαίνει στην Form5 που περιέχει τις επιλογές ασκήσεων. Πατώντας στο «Βαθμολογίες» πηγαίνει στην Form7 που του δείχνει τις βαθμολογίες του. Πατώντας στο «Αποσύνδεση» τότε αποσυνδέεται ο λογαριασμός του χρήστη και επιστρέφει στην Form1. Τέλος, πατώντας στο «Έξοδος» τερματίζεται το πρόγραμμα.

Υπάρχει επίσης ένα κουμπί βοήθειας.

1.4 Φόρμα 4 (Form4) – Μαθήματα:

Η Form4 περιέχει επιλογές για τις ενότητες μαθημάτων που επιθυμεί να μελετήσει ο χρήστης. Τα κουμπιά «Προπαίδεια του 1-10» οδηγούν στην Form8, που ανάλογα με την επιλογή, δείχνει και την αντίστοιχη ενότητα μαθήματος.

Ανάλογα με την επιλογή του χρήστη τότε αλλάζει η μεταβλητή Program.lessonChoice με την οποία καθορίζεται από την Form8 ποιο μάθημα επιλέχθηκε. Στον παρακάτω κώδικα έχει πατηθεί το κουμπί «Προπαίδεια του 5».

```
private void button5_Click(object sender, EventArgs e)  
{  
    Program.lessonChoice = 5;  
    this.Hide();  
    pinakes f8 = new pinakes();  
    f8.Show();  
}
```

Η Form4 έχει επίσης 3 κουμπιά για επιλογή ασκήσεων. Αν πατηθεί το «Ασκήσεις με επιλογή ενοτήτων», τότε ανοίγει η Form5, που επιτρέπει την επιλογή συγκεκριμένων ενοτήτων για εξάσκηση. Αν πατηθεί το «Επαναληπτικές Ασκήσεις», τότε ανοίγει η Form6, έχοντας ασκήσεις από όλες τις ενότητες.

Με την μεταβλητή Program.examChoice καθορίζει η Form6 τι είδους ασκήσεις πρέπει να δημιουργήσει.



Στο παράδειγμα του παρακάτω κώδικα, επιλέγονται επαναληπτικές ασκήσεις.

```
private void button13_Click(object sender, EventArgs e)
{
    Program.examChoice = 1;
    this.Hide();
    test f6 = new test();
    f6.Show();
}
```

Αν πατηθεί το «Ασκήσεις με βάση τα λάθη», τότε ανοίγει η Form6, έχοντας ασκήσεις που δημιουργούνται με βάση τις αδυναμίες του χρήστη, δηλαδή σε ποιες προπαίδειες τείνει να κάνει λάθη. Τα δεδομένα με βάση τα οποία το καθορίζει είναι στην βάση δεδομένων, θα αναλυθεί σε άλλη φόρμα.

1.5 Φόρμα 5 (Form5) – Ασκήσεις:

Η Form5 περιέχει ένα CheckedListBox με τις 10 ενότητες μαθημάτων. Επιλέγοντας ποιες επιθυμεί ο χρήστης και πατώντας «Εκκίνηση», ανοίγει η Form6 με ασκήσεις για τις επιλεγμένες ενότητες μαθημάτων.

Αυτό γίνεται με την χρήση του array Program.epilegmene, στον οποίο δίνουμε την τιμή 1 σε κάθε κελί που αντιστοιχεί σε επιλογή του CheckedListBox. Η Form6 καλείται με την τιμή 0 στην μεταβλητή Program.examChoice, που αντιστοιχεί σε ασκήσεις με επιλεγμένες ενότητες. Αν δεν επιλεγθεί καμία ενότητα βγαίνει αντίστοιχο μήνυμα.

```
private void button1_Click(object sender, EventArgs e)
{
    if (checkedListBox1.CheckedItems.Count == 0)
    {
        MessageBox.Show("Παρακαλώ επιλέξτε τουλάχιστον μία ενότητα.");
        return;
    }
    else
    {
        Array.Clear(Program.epilegmene, 0, Program.epilegmene.Length);
        foreach (int s in checkedListBox1.CheckedIndices)
        {
            Program.epilegmene[s] = 1;
        }
        Program.examChoice = 0; //askiseis me epilogi
        this.Hide();
        test f6 = new test();
        f6.Show();
    }
}
```

Υπάρχουν επίσης τα κουμπιά «Ασκήσεις με βάση τα λάθη» και «Επαναληπτικές ασκήσεις», που κάνουν ότι και τα αντίστοιχα κουμπιά στην Form4. Το κουμπί «Πίσω» επιστρέφει τον χρήστη στην Form4 με τα μαθήματα. Υπάρχει και κουμπί βοήθειας.



1.6 Φόρμα 6 (Form6) – Εξέταση:

Η Form6 περιέχει τις ασκήσεις στις οποίες καλείται να απαντήσει ο χρήστης συμπληρώνοντας τα αντίστοιχα textbox και πατώντας «Υποβολή».

Κατά το άνοιγμα της φόρμας (form_load) κατασκευάζονται οι ασκήσεις ανάλογα με τις προηγούμενες επιλογές του χρήστη στην Form4 ή Form5.

Αρχικά ορίζονται ο πίνακας askiseis, η λίστα enotites και μια μεταβλητή counter που θα χρησιμοποιηθούν στην δημιουργία των ασκήσεων.

```
int[] askiseis = new int[20];  
List<int> enotites = new List<int>();  
int counter = 0; ;  
Random rnd = new Random();
```

Στην συνέχεια, αν ο χρήστης έχει επιλέξει επαναληπτικές ασκήσεις τότε η λίστα enotites παίρνει τιμές 1 έως 10 και το counter τιμή 10.

```
if (Program.examChoice == 1)  
{  
    enotites.Clear();  
    enotites.Add(1);  
    enotites.Add(2);  
    enotites.Add(3);  
    enotites.Add(4);  
    enotites.Add(5);  
    enotites.Add(6);  
    enotites.Add(7);  
    enotites.Add(8);  
    enotites.Add(9);  
    enotites.Add(10);  
    label11.Text = "Έχετε επιλέξει τις επαναληπτικές ασκήσεις όλων των  
ενοτήτων.";   
    counter = 10;  
}
```

Αν ο χρήστης έχει επιλέξει ενότητες μέσω του CheckedListbox της Form5, τότε στην λίστα enotites, δίνουμε τις τιμές που έχει ο πίνακας Program.Epilegmena, που αντιπροσωπεύουν τις επιλεγμένες ενότητες και αυξάνει το counter για κάθε μία.

```
else if (Program.examChoice == 0)  
{  
    enotites.Clear();  
    for (int i = 0; i < 10; i++)  
    {  
        if (Program.epilegmena[i] == 1)  
        {  
            enotites.Add(i + 1);  
            counter++;  
        }  
    }  
}
```

Αλλάζει το μήνυμα στην κορυφή της φόρμας ανάλογα με τις επιλεγμένες ενότητες.

```
string epwp = "";  
if (counter == 1)  
{  
    epwp = enotites.ElementAt(0).ToString() + ".";  
}
```



```
else if (counter == 2)
{
    for (int i = 0; i < counter; i++)
    {
        if (i != counter - 1)
        {
            epwp = epwp + " " + enotites.ElementAt(i).ToString() + "
";
        }
        else
        {
            epwp = epwp + " και " + enotites.ElementAt(i).ToString()
+ ".";
        }
    }
}
else
{
    for (int i = 0; i < counter; i++)
    {
        if (i != counter - 1)
        {
            epwp = epwp + " " + enotites.ElementAt(i).ToString() + ",
";
        }
        else
        {
            epwp = epwp + " και " + enotites.ElementAt(i).ToString()
+ ".";
        }
    }
}

if (counter == 1)
{
    label1.Text = "Έχετε επιλέξει την ενότητα" + epwp;
}
else
{
    label1.Text = "Έχετε επιλέξει τις ενότητες" + epwp;
}
}
```

Αν ο χρήστης έχει επιλέξει ασκήσεις με βάση τα λάθη του, τότε διαβάζουμε τις καταχωρήσεις στην βάση δεδομένων με τα λάθη του χρήστη.

Οι καταχωρήσεις αποτελούνται από έναν δείκτη για κάθε ενότητα μαθήματος, που δείχνει πόσες φορές έχει κάνει λάθος σε πολλαπλασιασμό αυτής της ενότητας και δύο δείκτες που είναι οι συνολικές απαντήσεις που έχει δώσει ο χρήστης και το σύνολο των σωστών απαντήσεων.

Ο κώδικας παρακάτω είναι μόνο για τον πρώτο δείκτη, της πρώτης ενότητας. Στην συνέχεια επαναλαμβάνεται και για τους υπόλοιπους.

```
else if (Program.examChoice == 2)
{
```



```
label1.Text = "Έχετε επιλέξει ασκήσεις με βάση τα λάθη σας.";
enotites.Clear();
int crs;
int tls;
int[] usererrors = new int[10];
using (SqlConnection conn = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Stathis\Source\Repos\elef-
ant\Multiplication\Multiplication\Database1.mdf;Integrated Security=True"))
{
    //int crs;
    //int tls;
    //int[] usererrors = new int[10];

    conn.Open();

    //retrieve user data
    //corrects
    string query = "SELECT ec1" +
        " FROM ErrorTable WHERE ID = '" + Program.activeID + "'";
    SqlCommand cmd = new SqlCommand(query, conn);
    SqlDataAdapter sda = new SqlDataAdapter(query, conn);
    DataTable dta = new DataTable();
    sda.Fill(dta);
    usererrors[0] = Convert.ToInt32(dta.Rows[0]["ec1"].ToString());
}
```

Στο τέλος έχουμε τους δείκτες των ενοτήτων στον πίνακα usererrors. Για όποια ενότητα βρούμε πως ο χρήστης έχει πάνω από 5 λάθη, τότε θεωρούμε πως χρειάζεται επανάληψη και προσθέτουμε την ενότητα στην λίστα enotites.

```
for (int i = 0; i < 10; i++)
{
    if (usererrors[i] > 5)
    {
        counter++;
        enotites.Add(i + 1);
    }
}
```

Στην συνέχεια γεμίζουμε τον πίνακα ασκήσεις με ότι έχει η enotites μέσα, όποια από τις 3 προηγούμενες περιπτώσεις και να είχαμε. Παίρνουμε τυχαία στοιχεία της enotites με την χρήση τυχαίων αριθμών ανάμεσα στο 0 και το counter.

```
for (int i = 0; i < 20; i++)
{
    int pop = rnd.Next(0, counter);
    askiseis[i] = enotites.ElementAt(pop);
}
```

Τέλος, φτιάχνουμε τις ασκήσεις βάζοντας στα label της φόρμας έναν τυχαίο αριθμό και έναν από τους αριθμούς του πίνακα askiseis. Επαναλαμβάνεται 20 φορές για όλα τα label.

```
label181.Text = askiseis[19].ToString();
pr1 = rnd.Next(1, 11);
label179.Text = pr1.ToString();
```

Πατώντας το πλήκτρο «Νέες Ασκήσεις» γίνεται η ίδια διαδικασία, αλλάζοντας τις τιμές στα label.



Πατώντας το πλήκτρο «Υποβολή» γίνεται έλεγχος των απαντήσεων. Ο παρακάτω κώδικας εκτελείται για κάθε πολλαπλασιασμό. Αν η απάντηση του χρήστη είναι λάθος τότε το πεδίο απάντησης γίνεται κόκκινο, αυξάνεται ο μετρητής errorcnt και σε έναν πίνακα errorAr[] αυξάνουμε κατά 1 το κελί που αντιστοιχεί στους πολλαπλασιαστές της λανθασμένης πράξης.

```
int temp1 = Int32.Parse(label2.Text);
int temp2 = Int32.Parse(label4.Text);
int temp3 = Int32.Parse(textBox1.Text);
if (temp3 != (temp2 * temp1))
{
    errorcnt++;
    textBox1.BackColor = Color.Red;
    for (int i = 0; i < 10; i++)
    {
        if (temp1 == (i + 1))
        {
            errorAr[i] = errorAr[i] + 1;
        }
        else if (temp2 == (i + 1))
        {
            errorAr[i] = errorAr[i] + 1;
        }
    }
}
else
{
    textBox1.BackColor = Color.Green;
}
```

Αφού ελέγξει όλους τους πολλαπλασιασμούς, εμφανίζει ένα μήνυμα με τον αριθμό των λαθών που έκανε ο χρήστης.

Στην συνέχεια, παίρνουμε τις τιμές των δεικτών με τα λάθη ανά ενότητα από την βάση δεδομένων, όπως είδαμε και προηγουμένως στην δημιουργία ασκήσεων με βάση τα λάθη. Οι τιμές των δεικτών είναι στο array usererrors. Στους δείκτες αυτούς προσθέτουμε τα λάθη που έγιναν καινούριες ασκήσεις.

```
query = " UPDATE ErrorTABLE SET EC1 = @ec1, EC2 = @ec2, EC3 = @ec3, EC4 = @ec4, EC5 = @ec5, EC6 = @ec6, EC7 = @ec7, EC8 = @ec8, EC9 = @ec9, EC10 = @ec10, CORRECTS = @Corrects, TOTAL = @Total WHERE ID = '" + Program.activeID + "'";

cmd = new SqlCommand(query, conn);
cmd.Parameters.Add("ec1", usererrors[0] + errorAr[0]);
cmd.Parameters.Add("ec2", usererrors[1] + errorAr[1]);
cmd.Parameters.Add("ec3", usererrors[2] + errorAr[2]);
cmd.Parameters.Add("ec4", usererrors[3] + errorAr[3]);
cmd.Parameters.Add("ec5", usererrors[4] + errorAr[4]);
cmd.Parameters.Add("ec6", usererrors[5] + errorAr[5]);
cmd.Parameters.Add("ec7", usererrors[6] + errorAr[6]);
cmd.Parameters.Add("ec8", usererrors[7] + errorAr[7]);
cmd.Parameters.Add("ec9", usererrors[8] + errorAr[8]);
cmd.Parameters.Add("ec10", usererrors[9] + errorAr[9]);
cmd.Parameters.AddWithValue("Corrects", crs + correctcnt);
cmd.Parameters.AddWithValue("Total", t1s + 20);
cmd.ExecuteNonQuery();
```



Έχουμε και μια συνθήκη που ελέγχει αν οι ασκήσεις που ολοκληρώθηκαν ήταν ασκήσεις με βάση τα λάθη και αν ο χρήστης έκανε κάτω από 4 λάθη. Αν ναι, τότε θεωρεί πως ο χρήστης έχει βελτιωθεί και του μειώνει τους δείκτες λάθους στην βάση δεδομένων.

```
if ((Program.examChoice == 2) && (errorcnt < 4))
{
    query = " UPDATE ErrorTABLE SET EC1 = @ec1, EC2 = @ec2, EC3 = @ec3, EC4 = @ec4, EC5 = @ec5, EC6 = @ec6, EC7 = @ec7, EC8 = @ec8, EC9 = @ec9, EC10 = @ec10 WHERE ID = '" + Program.activeID + "'";
    cmd = new SqlCommand(query, conn);
    cmd.Parameters.Add("ec1", 1);
    cmd.Parameters.Add("ec2", 1);
    cmd.Parameters.Add("ec3", 1);
    cmd.Parameters.Add("ec4", 1);
    cmd.Parameters.Add("ec5", 1);
    cmd.Parameters.Add("ec6", 1);
    cmd.Parameters.Add("ec7", 1);
    cmd.Parameters.Add("ec8", 1);
    cmd.Parameters.Add("ec9", 1);
    cmd.Parameters.Add("ec10", 1);
    cmd.ExecuteNonQuery();
}
```

Η Form6 έχει επίσης ένα κουμπί «Πίσω» και ένα κουμπί «Βοήθεια».

1.7 Φόρμα 7 (Form7) – Βαθμολογίες:

Η Form7 αποτελείται από 2 πίνακες που δείχνουν στον χρήστη την πρόοδό του σε κάθε ενότητα και το συνολικό ποσοστό επιτυχίας του στις ασκήσεις.

Έχει επίσης κουμπί «Πίσω» και «Βοήθεια».

Κατά την δημιουργία της φόρμας, διαβάζονται τα δεδομένα του πίνακα ErrorTable.

```
int crs;
int t1s;
int[] usererrors = new int[10];
using (SqlConnection conn = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Stathis\Source\Repos\elefant\Multiplication\Multiplication\Database1.mdf;Integrated Security=True"))
{
    conn.Open();

    //retrieve user data
    string query = "SELECT ec1" +
        " FROM ErrorTable WHERE ID = '" + Program.activeID + "'";
    SqlCommand cmd = new SqlCommand(query, conn);
    SqlDataAdapter sda = new SqlDataAdapter(query, conn);
    DataTable dta = new DataTable();
    sda.Fill(dta);
    usererrors[0] = Convert.ToInt32(dta.Rows[0]["ec1"].ToString());
```

Όπως και πριν, παραθέτουμε ένα μικρό κομμάτι του κώδικα για λόγους συντομίας.

Στην συνέχεια, δίνουμε τις παραπάνω τιμές στα label για να τα δει ο χρήστης.

```
label13.Text = usererrors[0].ToString();
label15.Text = usererrors[1].ToString();
label17.Text = usererrors[2].ToString();
label19.Text = usererrors[3].ToString();
```



```
label18.Text = usererrors[4].ToString();  
label16.Text = usererrors[5].ToString();  
label14.Text = usererrors[6].ToString();  
label12.Text = usererrors[7].ToString();  
label20.Text = usererrors[8].ToString();  
label21.Text = usererrors[9].ToString();
```

Και του δείχνουμε και το ποσοστό επιτυχίας του. Στον παραπάνω και παρακάτω κώδικα crs είναι οι συνολικές σωστές απαντήσεις του και t1s οι συνολικές του απαντήσεις.

```
double temp1 = Convert.ToDouble(crs);  
double temp2 = Convert.ToDouble(t1s);  
double temp = (temp1 / temp2) * 100;  
  
if (temp2 == 0)  
{  
    label23.Text = "--";  
}  
else  
{  
    label23.Text = temp.ToString("0.0") + "%";  
}
```

1.8 Φόρμα 8 (Form8) – Ενότητες:

Στην Form8, έχουμε θεωρία για τις ενότητες του μαθήματος. Ανάλογα με την επιλογή του χρήστη στην Form4, έχουμε και τον αντίστοιχο πίνακα προπαίδειας μαζί με συμβουλές. Κατά την δημιουργία της φόρμας, χρησιμοποιούμε την μεταβλητή Program.lessonChoice που περιέχει την επιλογή του χρήστη. Με βάση αυτήν την μεταβλητή σχεδιάζουμε τους πίνακες.

```
int i = Program.lessonChoice;  
label00.Text = i.ToString();  
label10.Text = i.ToString();  
label20.Text = i.ToString();  
label30.Text = i.ToString();  
label40.Text = i.ToString();  
label50.Text = i.ToString();  
label60.Text = i.ToString();  
label70.Text = i.ToString();  
label80.Text = i.ToString();  
label90.Text = i.ToString();  
  
int a = (Int32.Parse(label00.Text) * Int32.Parse(label01.Text) ) ;  
label02.Text = a.ToString();  
a = (Int32.Parse(label10.Text) * Int32.Parse(label11.Text));  
label12.Text = a.ToString();  
a = (Int32.Parse(label20.Text) * Int32.Parse(label21.Text));  
label22.Text = a.ToString();  
a = (Int32.Parse(label30.Text) * Int32.Parse(label31.Text));  
label32.Text = a.ToString();  
a = (Int32.Parse(label40.Text) * Int32.Parse(label41.Text));  
label42.Text = a.ToString();  
a = (Int32.Parse(label50.Text) * Int32.Parse(label51.Text));  
label52.Text = a.ToString();
```



```
a = (Int32.Parse(label60.Text) * Int32.Parse(label61.Text));  
label62.Text = a.ToString();  
a = (Int32.Parse(label70.Text) * Int32.Parse(label71.Text));  
label72.Text = a.ToString();  
a = (Int32.Parse(label80.Text) * Int32.Parse(label81.Text));  
label82.Text = a.ToString();  
a = (Int32.Parse(label90.Text) * Int32.Parse(label91.Text));  
label92.Text = a.ToString();
```