**Лабораторная работа №8**

**«ОРГАНИЗАЦИЯ LINQ-ЗАПРОСОВ»**

**1. Постановка задачи**

Необходимо разработать тестовую программу с возможностью записи результатов ответа пользователя в субд access, а также с возможностью выполнить авторизацию пользователя из субд access. Кроме того, следует организовать запросы в методах: ***button1\_Click, button1\_Click2,button1\_Click3,button1\_Click4,button1\_Click5, button1\_Click6, outputResult*.** Также нужно добавить метод getMass, который конвертирует логический массив в строковый. Метод Form21\_Load для отображения данных из xml документа в dataGridView.

Для того, чтобы выполнить задание необходимо разработать:

1. Метод ***outputResult(TextBox textBox)****,* предназначенный для вывода массива в textBox1.
2. Метод ***button1\_Click(object sender, EventArgs e)*** , предназначенный для фильтрации результатов по верным ответам.
3. Метод ***button2\_Click(object sender, EventArgs e)***, предназначенный для фильтрации по неверным ответам.
4. Метод ***button3\_Click(object sender, EventArgs e)****,* предназначенный для создания словаря и вывода пользователей, чей балл больше 1.
5. Метод ***button4\_Click(object sender, EventArgs e)****,* предназначенный для реализации запросов к dataSet.
6. Метод ***button5\_Click(object sender, EventArgs e)****,* предназначенный для поиска пользователей в документе xml по логину.
7. Метод ***button6\_Click(object sender, EventArgs e)****,* предназначенный для создания xml файла.

**2 Разработка проекта**

**2.1 Разработка графического интерфейса пользователя**

Анализируя задание на разработку проекта, можно сделать вывод, что для организации ввода и вывода данных можно использовать четыре элемента управления. Текстовое поле элемента ***textbox*** используется как для ввода исходных данных в программу, так и для отображения результатов, получаемых в этой программе. Элемент ***label*** используется для размещения на форме поясняющих надписей. Кнопка ***button*** используется для запуска и окончания программы и инициализируется щелчком левой кнопки мыши. Элемент **DataGridView**, который является таблицей используется для вывода результатов.Таким образом, форма может иметь вид показанный на рисунке 1.

****

Рисунок 1 – Графический интерфейс

**2.2 Программный код проекта**

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

using Microsoft.VisualBasic;

using DataTable = System.Data.DataTable;

using TextBox = System.Windows.Forms.TextBox;

namespace \_Курсовая\_

{

public partial class Form25 : Form

{

Dictionary<String, Single> Polzovateli = new Dictionary<String, Single>();

int[] otvet;

string[] masive;

string log;

int count = 0;

private void Form21\_Load(object sender, EventArgs e)

{

if (System.IO.File.Exists(@"Результат\_теста.xml") == false)

{

Table.Columns.Add("Пользователь");

Table.Columns.Add("Логин");

// Добавить объект Таблица в DataSet

Data.Tables.Add(Table);

dataGridView1.DataSource = Table;

}

else

{

Data.ReadXml(@"Результат\_теста.xml");

Table = Data.Tables["Тест"];

dataGridView1.DataMember = "Тест";

dataGridView1.DataSource = Data;

}

}

public DataSet Data = new DataSet();

public DataTable Table = new DataTable();

public Form25()

{

InitializeComponent();

}

public void addUser()

{

int kol = 0;

For\_Kyrsovaya.Class1.Kol(ref otvet, ref kol); //\* 0.625f;

try

{

Polzovateli.Add(log, kol);

}

catch (Exception)

{

MessageBox.Show("Пользователь уже добавлен");

}

}

private string[] getmasive()

{

string[] masive = new string[otvet.Length];

MessageBox.Show("sss");

for (int i = 0; i < otvet.Length; i++)

{

masive[i] = "Верно";

if (otvet[i] == 0)

{

masive[i] = "Неверно";

}

}

return masive;

}

public void toExcel()

{

Microsoft.Office.Interop.Excel.Application excelApp = new Microsoft.Office.Interop.Excel.Application();

excelApp.Visible = true;

var t = Type.Missing;

var Book = excelApp.Workbooks.Add(t);

var Lists = Book.Worksheets;

Worksheet List = Lists.Item[1];

List.Cells[1, 1] = "Номер пользователя";

var users = from Stroka in Table.AsEnumerable()

select new

{

log = Stroka.Field<String>("Логин"),

result = Stroka.Field<String>("Результат"),

num = Stroka.Field<String>("Num")

};

int count = 2;

foreach (var Stroka in users)

{

List.Range["A" + count, t].Value2 = Stroka.num;

count++;

}

count = 2;

List.Range["B1", t].Value2 = "Логин пользователя";

foreach (var Stroka in users)

{

List.Range["B" + count, t].Value2 = Stroka.log;

count++;

}

count = 2;

List.Range["C1", t].Value2 = "Результат пользователя";

foreach (var Stroka in users)

{

List.Range["C" + count, t].Value2 = Stroka.result;

count++;

}

}

private void output(TextBox textBox)

{

textBox.Text += "Результаты теста:";

getmasive();

for (int i = 0; i < masive.Length; i++)

{

textBox.Text += (i + 1) + ") " + masive[i];

}

}

private void button3\_Click(object sender, EventArgs e)

{

string kol = Microsoft.VisualBasic.Interaction.InputBox("Количество пользователей", " ", " ", -1, -1);

string krit = Microsoft.VisualBasic.Interaction.InputBox("Введите критерий", " ", " ", -1, -1);

var Polzovateli = new Dictionary<String, Single>();

float krit1 = float.Parse(krit);

for (int i = 0; i < Convert.ToInt32(kol); i++)

Polzovateli.Add(Microsoft.VisualBasic.Interaction.InputBox("Введите логин", "Логин", " ", -1, -1),

Convert.ToInt32(Microsoft.VisualBasic.Interaction.InputBox("Введите результат для коллекции", "Результат", " ", -1, -1)));

var zapros = from Result in Polzovateli

group Result by Result.Value > krit1 into Group

select new

{

Group,

Group.Key,

srznach = Group.Average(Result => Result.Value)

};

foreach (var s in zapros)

{

if (s.Key == true)

{

textBox1.Text = ($"Пользователи, чей балл > {krit1}: " + "\r\n");

foreach (var t in s.Group)

textBox1.Text += String.Format("Пользователь {0} имеет балл {1} " + "\r\n", t.Key, t.Value);

MessageBox.Show("Средний балл: " + s.srznach, "Вывод", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

textBox1.Text = ($"Пользователи, чей балл < {krit1}: " + "\r\n");

foreach (var t in s.Group)

textBox1.Text += String.Format("Пользователь {0} имеет балл {1} " + "\r\n", t.Key, t.Value);

MessageBox.Show("Средний балл: " + s.srznach, "Вывод", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

private void button6\_Click(object sender, EventArgs e)

{

Table.TableName = "Тест";

Data.WriteXml(@"Результат\_теста.xml");

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

var Best = from Stroka in Table.AsEnumerable()

where

Convert.ToSingle(Stroka.Field<String>("Результат")) >= 7.3

select new

{

A = Stroka.Field<String>("Логин"),

B = Stroka.Field<String>("Результат")

};

textBox1.Text += "Лучшие: " + "\r\n";

foreach (var Stroka in Best)

textBox1.Text += Stroka.A + " - " + Stroka.B + "\r\n";

toExcel();

}

private void button1\_Click(object sender, EventArgs e)

{

int True\_ans = 0;

string[] mas = new string[18];

for (int i = 0; i < mas.Length; i++)

{

if (mas[i] == null)

{

mas[i] = "Неверно";

mas[14] = "Верно";

mas[15] = "Верно";

mas[13] = "Верно";

}

}

for (int i = 0; i < mas.Length; i++)

{

if (mas[i] == "Верно")

{

True\_ans += 1;

}

}

label1.Text = "Поздравляем, вы прошли тест. Ваш результат: " + Convert.ToString(True\_ans) + " из 16";

dataGridView1.Visible = true;

button1.Visible = false;

For\_Kyrsovaya.Class1.VivodDGV2(ref dataGridView1, mas);

textBox1.Text = "Результаты теста:" + "\r\n\r\n";

foreach (String rez in mas)

textBox1.Text = textBox1.Text + rez + "; ";

var zapros = from s in mas

where s.Length == 5

orderby s

select s.ToUpper();

textBox1.Text += "\r\n\r\n";

textBox1.Text += "Отсортированные результаты:" + " " + "\r\n\r\n";

for (int i = 0; i < mas.Length; ++i)

{

if (mas[i] == "Верно")

{

foreach (String s in zapros)

{

textBox1.Text = textBox1.Text + $"Вопрос № " + i + " " + s + "; ";

break;

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

string[] mas = new string[18];

int True\_ans = 0;

for (int i = 0; i < mas.Length; i++)

{

if (mas[i] == null)

{

mas[i] = "Неверно";

mas[14] = "Верно";

mas[15] = "Верно";

mas[13] = "Верно";

}

}

var spisok = new List<string>(mas);

textBox1.Text = "Результаты теста:" + "\r\n\r\n";

foreach (String x in spisok)

textBox1.Text = textBox1.Text + x + "; ";

var zapros2 = from string s in spisok

where s.Length == 7

orderby s

select s;

textBox1.Text += "\r\n\r\n";

textBox1.Text += "Отсортированные результаты:" + " " + "\r\n\r\n";

for (int i = 0; i < mas.Length; ++i)

{

if (mas[i] == "Неверно")

{

foreach (String x in zapros2)

{

textBox1.Text = textBox1.Text + $"Вопрос № " + i + " " + x + "; ";

break;

}

}

}

}

private void button5\_Click(object sender, EventArgs e)

{

string surname = Microsoft.VisualBasic.Interaction.InputBox("Введите логин для проверки его в xml документе", "Заголовок окна", " ", -1, -1);

textBox1.Clear();

var koren = XElement.Load(@"C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\Rez.txt");

var zapros = from x in koren.Elements("Тест")

where (String)x.Element("Логин") == surname

select x.Element("Результат").Value;

if (zapros.Count() == 0)

{

textBox1.Text = textBox1.Text + "Не найдены строки,содержащие логин " + surname + ": " + "\r\n";

}

else

{

textBox1.Text = textBox1.Text + "Найдены строки,содержащие логин " + surname + ": " + "\r\n";

foreach (var x in zapros)

textBox1.Text = textBox1.Text + " Результат пользователя = " + x + "\r\n";

}

}

private void Form25\_Load(object sender, EventArgs e)

{

if (System.IO.File.Exists(@"xmlresults.xml") == false)

{

Table.Columns.Add("Login");

Table.Columns.Add("Результат");

Data.Tables.Add(Table);

dataGridView2.DataSource = Table;

}

else

{

Data.ReadXml(@"xmlresults.xml");

Table = Data.Tables["Тест"];

dataGridView2.DataMember = "Тест";

dataGridView2.DataSource = Data;

}

}

}

}

**3** **Результат выполнения программы**

Результаты выполнения проекта приведены на рисунках 18,19,20,21,22.1 и 22.2,23.

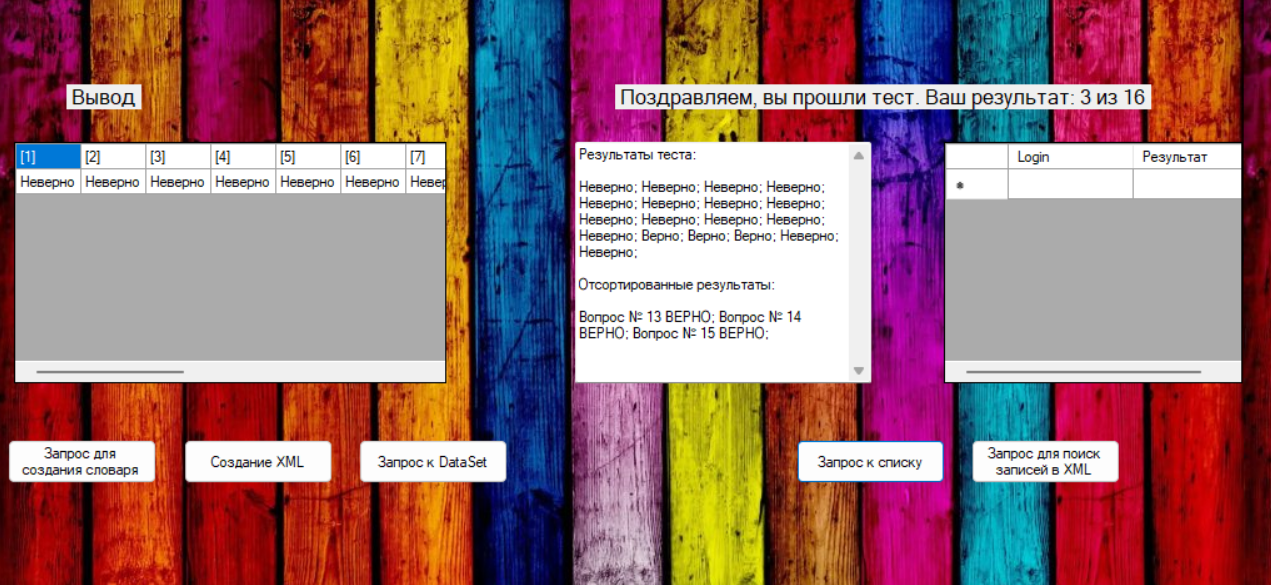


Рисунок 2 – ling-запрос к массиву

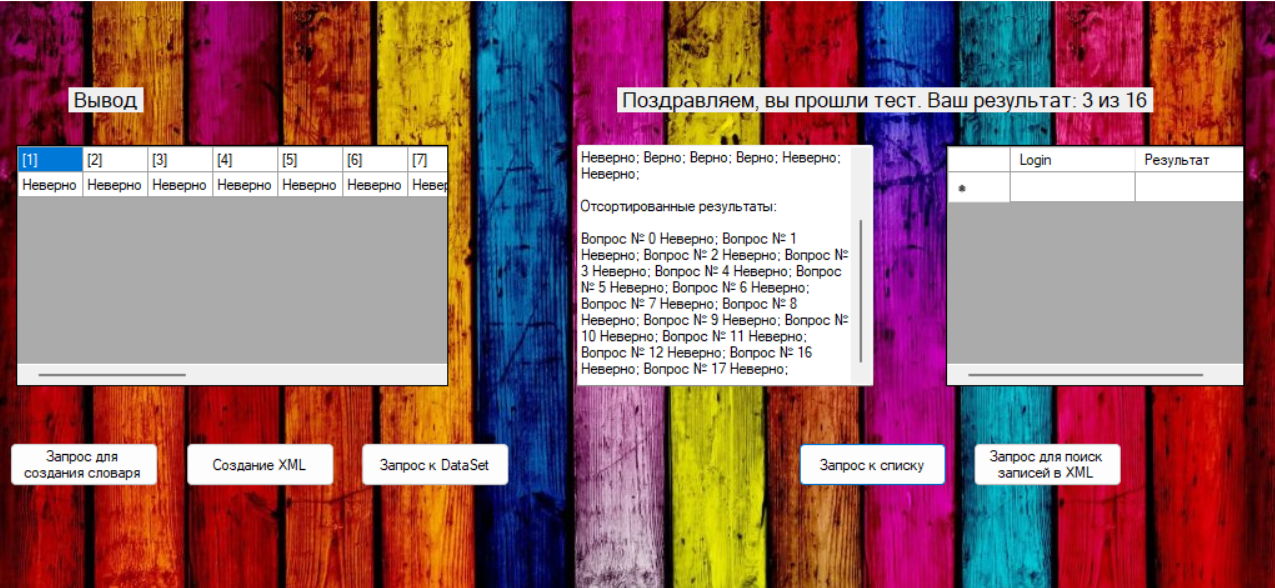


Рисунок 3 – Запрос к списку

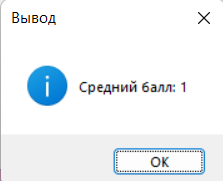


Рисунок 4 – Запрос для поиска пользователей с баллом больше 3

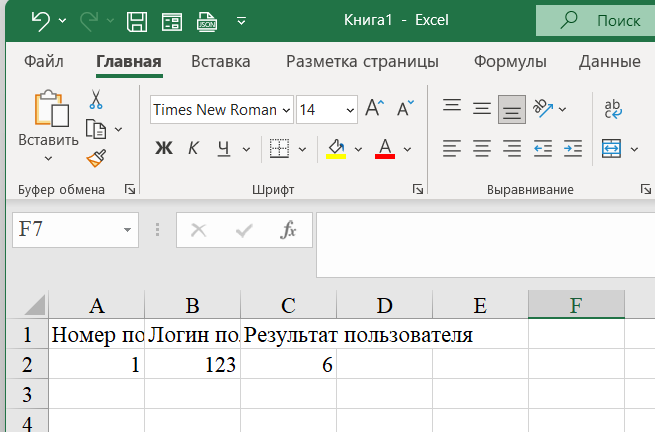


Рисунок 5 - linq-запрос для поиска пользователей с лучшими результатами