МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

#### Факультет информационных технологий и компьютерной безопасности

(факультет)

### Кафедра Систем автоматизированного проектирования и информационных систем

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по дисциплине Программирование на платформе .NET Framework

Тема «Разработка пользовательских интерфейсов для программирования массивов

на платформе .NET Framework»

Выполнили студенты группы ИСТ-222 Е.Г. Гладнева

Подпись, дата Инициалы, фамилия

А.Н. Грипунова

Подпись, дата Инициалы, фамилия

Е.В. Журавлёва

Подпись, дата Инициалы, фамилия

## Руководитель Э.И. Воробьев

Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

2023

Лабораторная работа № 4

«Разработка пользовательских интерфейсов для программирования массивов

на платформе .NET Framework»

1 Цель работы

Целью лабораторной работы является изучение основ программирования массивов на платформе .NET Framework.

2 Задание на лабораторную работу

Был выбран вариант №9.

Выполнить задания в консольном приложении:

Задание 1. Сформировать одномерный (длиной n) и двумерный (длиной n на m) массивы по заданному правилу, где k – случайное число:



Рисунок 1 — Формула для задания 1

Задание 2. Заданы два массива А и В. В каждом из массивов найти наименьшее значение и умножить на него все элементы массивов. На печать вывести исходные и преобразованные массивы.

Задание 3. Дан массив А(n,n). Найти число элементов массива a(i,j) > t и просуммировать все эти элементы.

Задание 4. Сортировка элементов массива методом прямого включения.

Выполнить задание в приложении Windows Forms:

Написать программу для ввода/вывода данных в массив с различной длинной записей, то есть создать массив массивов, перечисляющий различные свойства (поля структуры) трех различных объектов (набор и количество полей должно быть разным) в соответствии с таблицей вариантов. Для варианта №9 это Вертолёт, Завод и Преподаватель.

3 Ход выполнения

1. В первую очередь создадим интерактивное меню (рисунок 2), с помощью которого можно будет перейти к любому заданию. Для этого добавим элемент ListBox с вариантами заданий.

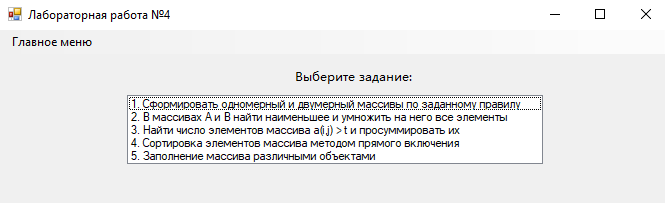


Рисунок 2 — Меню программы

2. Затем создаём внутри приложения Windows Forms консольное приложение ConsoleApp. В нём будут находиться первые 4 задания. Вызов конкретного задания происходит с помощью метода Process.Start, которому передаётся путь на исполняемый файл консольного приложения и аргумент запуска, который является номером задания.

3. Для выполнения первого задания запросим у пользователя размерности массивов. Затем с помощью цикла for заполним одномерный массив согласно формуле. Для заполнения двумерного массива нужно будет объявить соответствующую переменную, которая будет иметь тип List<List<double>>. Кроме того, теперь для итерации по элементам потребуется вложенный цикл for. Пример выполнения данного задания представлен на рисунке 3.

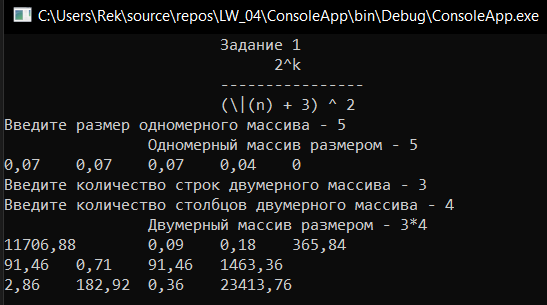


Рисунок 3 — Задание 1

4. При выполнении второго задания запрашиваем у пользователя размеры двух массивов, после чего заполняем их случайными значениями в пределах 40. После этого, используя цикл for, находим минимальное значение в первом массиве и затем, вновь используя for, умножаем на него все элементы. Аналогичные действия проделываем со вторым массивом. Пример выполнения данного задания представлен на рисунке 4.

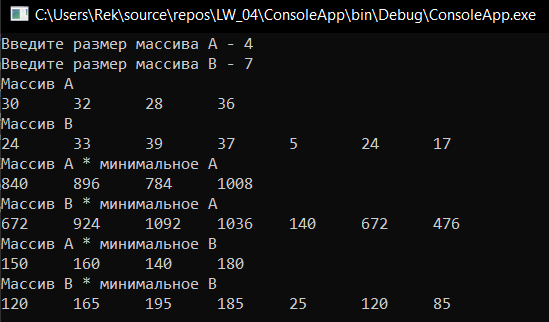


Рисунок 4 — Задание 2

5. В третьем задании запрашиваем размеры квадратной матрицы, после чего заполняем её случайными значениями. Также получаем от пользователя некоторое число. используя вложенный цикл, проходим по всем элементам матрицы и, если текущий элемент больше заданного числа, то прибавляем его к конечной сумме. Пример выполнения данного задания представлен на рисунке 5.

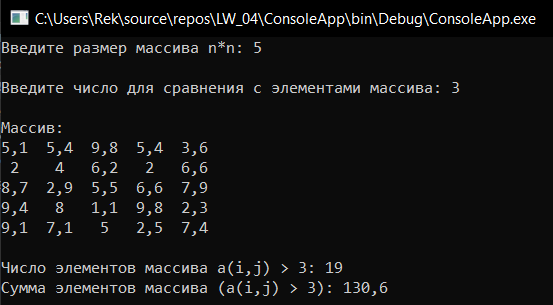


Рисунок 5 — Задание 3

6. В четвертом задании как обычно заполняем случайными значениями массив заданной пользователем размерности, после чего проводим сортировку методом прямого включения. Пример выполнения данного задания представлен на рисунке 6.

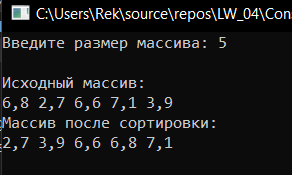


Рисунок 6 — Задание 4

7. Для последнего задания потребуются такие элементы, как TreeView для отображения существующих элементов и добавления новых, а также GroupBox, который будет содержать подробную информацию о каждом элементе. Также создаём 3 структуры: Вертолёт, Завод, Преподаватель, и 3 формы для их добавления соответственно. Вешний вид форм представлен на рисунках 7, 8, 9.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Рисунок 7 — Добавление вертолёта | Рисунок 8 — Добавление завода | Рисунок 9 — Добавление преподавателя |

При добавлении нового элемента через показанные выше меню, он добавляется как дочерний узел в соответствующий родительский узел TreeView. При нажатии на родительский узел открывается меню добавления нового элемента, а при нажатии на дочерний — подобная информация о нём. Пример такого отображения представлен на рисунках 10, 11, 12.

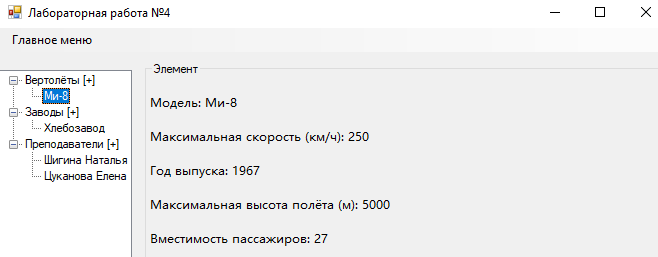


Рисунок 10 — Элемент «Вертолёт»

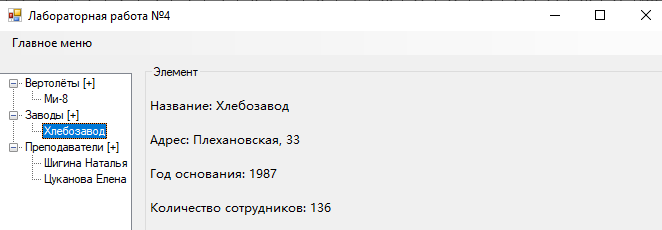


Рисунок 11 — Элемент «Хлебозавод»

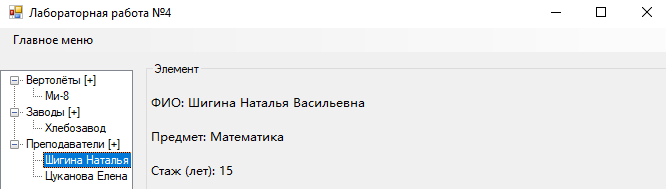


Рисунок 12 — Элемент «Преподаватель»

Вывод

В результате выполнения лабораторной работы мы изучили основы разработки пользовательского интерфейса для программирования массивов на платформе .NET Framework.

Приложение А

Листинг программного кода языка C#

Консольное приложение

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp

{

internal class Program

{

static void Main(string[] args)

{

if (args.Length == 0)

{

Console.WriteLine("Запуск программы возможен только из формы");

return;

}

switch (args[0])

{

case "1":

{

//Среди заданных чисел найти количество четных чисел.

Console.WriteLine("\t\t\tЗадание 1");

Console.WriteLine("\t\t\t 2^k");

Console.WriteLine("\t\t\t----------------");

Console.WriteLine("\t\t\t(\\|(n) + 3) ^ 2");

int n1;

Console.Write("Введите размер одномерного массива - ");

n1 = Int32.Parse(Console.ReadLine());

exercise1\_one\_dimensional\_array(n1);

int n, m;

Console.Write("Введите количество строк двумерного массива - ");

n = Int32.Parse(Console.ReadLine());

Console.Write("Введите количество столбцов двумерного массива - ");

m = Int32.Parse(Console.ReadLine());

exercise1\_two\_dimensional\_array(n, m);

Console.ReadKey();

break;

}

case "2":

{

int A;

Console.Write("Введите размер массива A - ");

A = Int32.Parse(Console.ReadLine());

int B;

Console.Write("Введите размер массива B - ");

B = Int32.Parse(Console.ReadLine());

while (A <= 0 || B <= 0)

{

Console.WriteLine("ОШИБКА! Вы ввели что-то не так, попробуйте снова");

Console.Write("Введите размер массива A - ");

A = Int32.Parse(Console.ReadLine());

Console.Write("Введите размер массива B - ");

B = Int32.Parse(Console.ReadLine());

}

exercise2(A, B);

Console.ReadKey();

break;

}

case "3":

{

uint n;

double t, sum = 0, k = 0;

Console.Write("Введите размер массива n\*n: ");

n = GetInt();

Console.Write("\nВведите число для сравнения с элементами массива: ");

t = GetDouble();

Console.WriteLine("\nМассив:");

List<List<double>> A = new List<List<double>>();

List<double> row = new List<double>();

Random rand = new Random();

for (int i = 0; i < n; i++)

{

row = new List<double>();

for (int j = 0; j < n; j++) row.Add(i + j);

A.Add(row);

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

A[i][j] = rand.NextDouble() + rand.Next(10);

A[i][j] = Math.Round(A[i][j], 1);

if (A[i][j] > t)

{

sum += A[i][j];

k++;

}

if (A[i][j] == (int)A[i][j])

Console.Write(" " + A[i][j].ToString() + " ");

else Console.Write(A[i][j].ToString() + " ");

}

Console.WriteLine();

}

Console.WriteLine("\nЧисло элементов массива a(i,j) > {0}: {1} ", t, k);

Console.WriteLine("Сумма элементов массива (a(i,j) > {0}): {1}\n ", t, Math.Round(sum, 2));

Console.ReadKey();

break;

}

case "4":

{

uint m;

Console.Write("Введите размер массива: ");

m = GetInt();

double[] arr = new double[m];

Random random = new Random();

Console.WriteLine("\nИсходный массив: ");

for (int i = 0; i < m; i++)

{

arr[i] = random.NextDouble() + random.Next(10);

arr[i] = Math.Round(arr[i], 1);

Console.Write(arr[i] + " ");

}

Sort(arr, m);

Console.WriteLine("\nМассив после сортировки: ");

for (int i = 0; i < m; i++)

Console.Write(arr[i] + " ");

Console.ReadKey();

break;

}

}

}

static void exercise1\_one\_dimensional\_array(int n) //одномерный массив

{

Random random = new Random();

int k;

double[] arr = new double[n];

Console.WriteLine("\t\tОдномерный массив размером - {0}", n);

for (int i = 0; i < n; i++)

{

k = random.Next(10);

if (i == n - 1)

{

Console.WriteLine(arr[i]);

}

else

{

arr[i] = Math.Round((Math.Pow(2, k)) / (Math.Pow((Math.Sqrt(n) + 3), 2)), 2);

Console.Write(arr[i] + "\t");

}

}

}

static void exercise1\_two\_dimensional\_array(int n, int m) // двумерный массив

{

int k;

List<List<double>> Mas = new List<List<double>>(); //динамический двумерный массив

//строка массива

double zamena;

Random random = new Random();

Console.WriteLine("\t\tДвумерный массив размером - {0}\*{1}", n, m);

for (int i = 0; i < n; i++)

{

List<double> row = new List<double>(); //создается новая строка, в которую мы запихиваем значения zamena

for (int j = 0; j < m; j++)

{

k = random.Next(20);

zamena = Math.Round((Math.Pow(2, k)) / (Math.Pow((Math.Sqrt(n) + 3), 2)), 2);

row.Add(zamena);

Console.Write(row[j].ToString() + "\t");

}

Mas.Add(row); //заполняем наш массив по строчкам

Console.WriteLine();

}

}

static void exercise2(int A, int B)

{

Random random = new Random();

int[] array\_a = new int[A];

int[] array\_b = new int[B];

int[] array\_a\_a = new int[A];

int[] array\_a\_b = new int[B];

int[] array\_b\_a = new int[A];

int[] array\_b\_b = new int[B];

int min\_A = 1000;

int min\_B = 1000;

Console.WriteLine("Массив А");

for (int i = 0; i < A; i++) //заполняем массив А рандомн числами в пределах 40

{

array\_a[i] = random.Next(40);

Console.Write(array\_a[i] + "\t");

}

Console.WriteLine();

Console.WriteLine("Массив B");

for (int i = 0; i < B; i++)//заполняем массив B рандомн числами в пределах 40

{

array\_b[i] = random.Next(40);

Console.Write(array\_b[i] + "\t");

}

Console.WriteLine();

for (int i = 0; i < A; i++) //находим минимальное в массиве А

{

if (array\_a[i] < min\_A)

{

min\_A = array\_a[i];

}

}

for (int i = 0; i < B; i++) //находим минимальное в массиве B

{

if (array\_b[i] < min\_B)

{

min\_B = array\_b[i];

}

}

Console.WriteLine("Массив А \* минимальное А");

for (int i = 0; i < A; i++) //перемножаем минимальное массива А на все элементы массива А

{

array\_a\_a[i] = array\_a[i] \* min\_A;

Console.Write(array\_a\_a[i] + "\t");

}

Console.WriteLine();

Console.WriteLine("Массив В \* минимальное А");

for (int i = 0; i < B; i++) //перемножаем минимальное массива А на все элементы массива В

{

array\_a\_b[i] = array\_b[i] \* min\_A;

Console.Write(array\_a\_b[i] + "\t");

}

Console.WriteLine();

Console.WriteLine("Массив А \* минимальное В");

for (int i = 0; i < A; i++) //перемножаем минимальное массива В на все элементы массива А

{

array\_b\_a[i] = array\_a[i] \* min\_B;

Console.Write(array\_b\_a[i] + "\t");

}

Console.WriteLine();

Console.WriteLine("Массив В \* минимальное В");

for (int i = 0; i < B; i++) //перемножаем минимальное массива В на все элементы массива В

{

array\_b\_b[i] = array\_b[i] \* min\_B;

Console.Write(array\_b\_b[i] + "\t");

}

Console.WriteLine();

}

private static uint GetInt()

{

int cur\_y = Console.CursorTop;

int cur\_x = Console.CursorLeft;

uint res;

while (!uint.TryParse(Console.ReadLine(), out res))

{

Console.SetCursorPosition(cur\_x, cur\_y);

Console.WriteLine("Не удалось получить значение, попробуйте ещё раз. Нажмите любую клавишу...");

Console.ReadKey();

Console.SetCursorPosition(cur\_x, cur\_y);

Console.Write(new string(' ', Console.WindowWidth));

Console.SetCursorPosition(cur\_x, cur\_y);

}

return res;

}

private static double GetDouble()

{

int cur\_y = Console.CursorTop;

int cur\_x = Console.CursorLeft;

double res;

while (!double.TryParse(Console.ReadLine(), out res))

{

Console.SetCursorPosition(cur\_x, cur\_y);

Console.WriteLine("Не удалось получить значение, попробуйте ещё раз. Нажмите любую клавишу...");

Console.ReadKey();

Console.SetCursorPosition(cur\_x, cur\_y);

Console.Write(new string(' ', Console.WindowWidth));

Console.SetCursorPosition(cur\_x, cur\_y);

}

return res;

}

public static void Sort(double[] num, uint size)

{

for (int i = 1; i < size; i++)

{

double value = num[i];

int index = i;

while ((index > 0) && (num[index - 1] > value))

{

num[index] = num[index - 1];

index--;

}

num[index] = value;

}

}

}

}

Приложение Б

Листинг программного кода языка C#

Добавление элементов

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;

namespace LW\_04

{

public partial class Ex\_5 : UserControl

{

public Ex\_5()

{

InitializeComponent();

}

public static List<Helicopter> helicopters = new List<Helicopter>();

public static List<Factory> factories = new List<Factory>();

public static List<Teacher> teachers = new List<Teacher>();

private void Select(object sender, EventArgs e)

{

string name = treeView1.SelectedNode.Name;

switch (name)

{

case "Helicopters":

{

AddHelicopter f = new AddHelicopter();

if (f.ShowDialog() == DialogResult.OK)

{

Helicopter hel = f.h;

treeView1.Nodes[name].Nodes.Add(hel.model, hel.model);

helicopters.Add(hel);

}

break;

}

case "Factories":

{

AddFactory f = new AddFactory();

if (f.ShowDialog() == DialogResult.OK)

{

Factory fac = f.fc;

treeView1.Nodes[name].Nodes.Add(fac.name, fac.name);

factories.Add(fac);

}

break;

}

case "Teachers":

{

AddTeacher f = new AddTeacher();

if (f.ShowDialog() == DialogResult.OK)

{

Teacher th = f.th;

treeView1.Nodes[name].Nodes.Add(th.name, th.name);

teachers.Add(th);

}

break;

}

default:

{

foreach(Helicopter h in helicopters)

{

if (name == h.model)

{

groupBox1.Controls.Clear();

Label txt = new Label();

txt.Anchor = AnchorStyles.None;

txt.AutoSize = true;

txt.Dock = DockStyle.Fill;

txt.Font = new Font("Microsoft YaHei", 9F, FontStyle.Regular, GraphicsUnit.Point, (byte)(204));

txt.Name = "label10";

txt.Size = new Size(89, 20);

txt.Text = string.Format("\nМодель: {0}\n\n" +

"Максимальная скорость (км/ч): {1:g6}\n\n" +

"Год выпуска: {2}\n\n" +

"Максимальная высота полёта (м): {3:g6}\n\n" +

"Вместимость пассажиров: {4}",

h.model, h.max\_speed, h.year, h.max\_height, h.capacity);

groupBox1.Controls.Add(txt);

return;

}

}

foreach (Factory fc in factories)

{

if (name == fc.name)

{

groupBox1.Controls.Clear();

Label txt = new Label();

txt.Anchor = AnchorStyles.None;

txt.AutoSize = true;

txt.Dock = DockStyle.Fill;

txt.Font = new Font("Microsoft YaHei", 9F, FontStyle.Regular, GraphicsUnit.Point, (byte)(204));

txt.Name = "label10";

txt.Size = new Size(89, 20);

txt.Text = string.Format("\nНазвание: {0}\n\n" +

"Адрес: {1}\n\n" +

"Год основания: {2}\n\n" +

"Количество сотрудников: {3}",

fc.name, fc.address, fc.foundation\_year, fc.employees);

groupBox1.Controls.Add(txt);

return;

}

}

foreach (Teacher th in teachers)

{

if (name == th.name)

{

groupBox1.Controls.Clear();

Label txt = new Label();

txt.Anchor = AnchorStyles.None;

txt.AutoSize = true;

txt.Dock = DockStyle.Fill;

txt.Font = new Font("Microsoft YaHei", 9F, FontStyle.Regular, GraphicsUnit.Point, (byte)(204));

txt.Name = "label10";

txt.Size = new Size(89, 20);

txt.Text = string.Format("\nФИО: {0}\n\n" +

"Предмет: {1}\n\n" +

"Стаж (лет): {2}",

th.name, th.subject, th.experience);

groupBox1.Controls.Add(txt);

return;

}

}

break;

}

}

}

}

}

Приложение В

Листинг программного кода языка C#

Добавление вертолёта

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LW\_04

{

public partial class AddHelicopter : Form

{

public AddHelicopter()

{

InitializeComponent();

CenterToScreen();

}

public Helicopter h = new Helicopter();

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

{

h.model = textBox1.Text;

try

{

h.max\_speed = Double.Parse(textBox2.Text);

h.year = Int32.Parse(textBox3.Text);

h.max\_height = Double.Parse(textBox4.Text);

h.capacity = Int32.Parse(textBox5.Text);

}

catch (FormatException)

{

MessageBox.Show(this,

"Проверьте корректность заполнения полей",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1);

return;

}

if (h.model.Length == 0)

{

MessageBox.Show(this,

"Все поля должны быть заполнены",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1);

return;

}

DialogResult = DialogResult.OK;

Close();

}

}

}

Приложение Г

Листинг программного кода языка C#

Добавление завода

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 static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace LW\_04

{

public partial class AddFactory : Form

{

public AddFactory()

{

InitializeComponent();

CenterToScreen();

}

public Factory fc = new Factory();

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

{

fc.name = textBox1.Text;

fc.address = textBox2.Text;

try

{

fc.foundation\_year = Int32.Parse(textBox3.Text);

fc.employees = Int32.Parse(textBox4.Text);

}

catch (FormatException)

{

MessageBox.Show(this,

"Проверьте корректность заполнения полей",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1);

return;

}

if (fc.name.Length == 0 || fc.address.Length == 0)

{

MessageBox.Show(this,

"Все поля должны быть заполнены",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1);

return;

}

DialogResult = DialogResult.OK;

Close();

}

}

}

Приложение Е

Листинг программного кода языка C#

Добавление преподавателя

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 static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace LW\_04

{

public partial class AddTeacher : Form

{

public AddTeacher()

{

InitializeComponent();

CenterToScreen();

}

public Teacher th = new Teacher();

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

{

th.name = textBox1.Text;

th.subject = textBox2.Text;

try

{

th.experience = Int32.Parse(textBox3.Text);

}

catch (FormatException)

{

MessageBox.Show(this,

"Проверьте корректность заполнения полей",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1);

return;

}

if (th.name.Length == 0 || th.subject.Length == 0)

{

MessageBox.Show(this,

"Все поля должны быть заполнены",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1);

return;

}

DialogResult = DialogResult.OK;

Close();

}

}

}