Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт высоких технологий

наименование института

Вычислительная техника

наименование кафедры

Допускаю к защите

Руководитель:

М.Д. Каташевцев

 И.О. Фамилия

Обработка файлов на языке C#

Наименование темы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту по дисциплине

Программирование на языке высокого уровня

1.001.00.00 ПЗ

обозначение документа

 Выполнил студент группы ИС### 17-1 Иванов И.И

 шифр подпись Фамилия И.О.

 Нормоконтроль Каташевцев М.Д.

 подпись Фамилия И.О.

 Курсовой проект защищен с оценкой:

Иркутск 2018 г.

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ**

**НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

По курсу Программирование на языке высокого уровня

Студенту Иванову Ивану Ивановичу

(фамилия, инициалы)

Тема проекта Обработка файлов на языке C#

Исходные данные Вариант #,#,#,#

Рекомендуемая литература

1. Сосинская С.С. «Обработка файлов»: Методические указания по выполнению курсового проекта» Электронный каталог кафедры.

Графическая часть на - листах.

Дата выдачи задания 09 / 11 /2018 г.

Задание получил студент Иванов И.И

 подпись Фамилия И.О.

Дата представления проекта руководителю 21 / 12 /2018 г.

Руководитель курсового проектирования Каташевцев М.Д.

 подпись Фамилия И.О.

Содержание

[Введение 4](#_Toc531003483)

[1 Индивидуальное задание 5](#_Toc531003484)

[2 Иерархия форм и функций с описанием их назначения 6](#_Toc531003485)

[3 Алгоритмы основных событий 8](#_Toc531003486)

[4 Формы 13](#_Toc531003487)

[5 Коды основных событий 17](#_Toc531003488)

[6 Таблица тестов 22](#_Toc531003489)

[7 Результаты тестирования 24](#_Toc531003490)

[Заключение 37](#_Toc531003491)

[Список использованных источников 38](#_Toc531003492)

Введение

Требуется на основании индивидуального задания разработать Windows-приложение на языке С#, таблицу тестов и оттестировать программу для создания файла, добавления записи, удаления записи, изменения записи, просмотра файла и выполнения двух запросов. Удаление и изменение записи происходит по ее номеру. Способы выполнения действий – согласно индивидуальному заданию. При ошибочных ситуациях (отсутствие записи, отсутствие файла и т.д.) необходимо выдавать сообщения.

Visual C# является реализацией языка C# корпорацией Майкрософт. Visual Studio поддерживает Visual C# с полнофункциональным редактором кода, компилятором, шаблонами проектов, конструкторами, мастерами кода, мощным и простым в использовании отладчиком и многими другими средствами. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и другим полезным, правильным классам, что существенно ускоряет цикл разработки.

C#-это язык программирования, предназначенный для разработки самых разнообразных приложений. Язык C# прост, строго типизирован и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C.

Необходимо на основании индивидуального задания разработать Windows-приложение на языке С#, таблицу тестов и оттестировать программу для создания файла, добавления записи, удаления записи, изменения записи, просмотра файла и выполнения двух запросов. Удаление и изменение записи происходит по ее номеру. Способы выполнения действий – согласно индивидуальному заданию. При ошибочных ситуациях (отсутствие записи, отсутствие файла и т.д.) необходимо выдавать сообщения.

1 Индивидуальное задание

Вариант 1-1-4-1

 1 - Файл **"Студент"** (ФИО, шифр группы, пол, семейное положение, номер комнаты в общежитии).

1 - По номеру комнаты вывести список жильцов;

4 - Выдать список студентов с фамилией на заданную букву.

1 - Добавление в конец файла

2 Иерархия форм и функций с описанием их назначения



Схема 1 – Иерархия форм



Схема 2 – Иерархия функций

Таблица 1 –Название и назначение функций

|  |  |
| --- | --- |
| Название функции | Назначение |
| Form 1 |
| создатьToolStripMenuItem\_Click | Вызов Form 2 для создание нового файла |
| добавитьToolStripMenuItem\_Click | Вызов Form 3 для добавления записи к файлу |
| удалитьToolStripMenuItem\_Click | Вызов Form 4 для удаления записи из файла |
| просмотрToolStripMenuItem\_Click | Вызов Form 5 для просмотра записей |
| запрос1ToolStripMenuItem\_Click | Вызов Form 6 для выполнения 1 запроса |
| запрос2ToolStripMenuItem\_Click | Вызов Form 7 для выполнения 2 запроса |
| выходToolStripMenuItem\_Click | Закрыть форму |
| public static class Global | Создание глобальных переменных |
| public static int NameArr | Чтение файла в массиве строк |
| Form 2 |
| private void button1\_Click | Добавление записи в файл |
| private void button2\_Click | Закрыть форму |
| Form 3 |
| private void button1\_Click | Находит запись по введенному номеру |
| private void button2\_Click | Изменение данных записи после ее корректировки |
| private void button3\_Click | Закрыть форму |
| Form 4 |
| private void button1\_Click | Удалить запись из файла по номеру |
| private void button2\_Click | Закрыть форму |
| Form 5 |
| private void button1\_Click | Закрыть форму |
| private void Form5\_Load | Загрузка записей из файла в таблицу приложения |
| Form 6 |
| private void button1\_Click | Вывод списка жильцов по номеру комнаты |
| private void button2\_Click | Закрыть форму |
| Form 7 |
| private void button1\_Click | Вывод списка студентов с фамилией на заданную букву |
| private void button2\_Click | Закрыть форму |

3 Алгоритмы основных событий

 Функция NameArr

1. Открываем файл “stud.txt” для чтения
2. N=0
3. ПОКА не дошли до конца файла ВЫПОЛНИТЬ
	1. NovArr[N]=чтение строки из файла
	2. Записываем в массив Arr данные строки NovArr[N]
	3. student[N].name = Arr[0];
	4. student[N].num = Convert.ToInt32(Arr[2]);
	5. Если (Arr[3] == «Женский») student[N].sex = true;
	6. Если (Arr[3] == «Мужской») student[N].sex = false;
	7. student[N].grupp = Arr[1];
	8. Если (Arr[4] == «В активном поиске») student[N].sp = true;
	9. Если (Arr[4] == «В отношениях») student[N].sp = false;
	10. N=N+1

ПОКА ВСЕ

1. Закрываем файл “stud.txt”
2. Возврат N

 Выбор пункта меню создать ToolStripMenuItem\_Click на форме Form1

1. ЕСЛИ файл “stud.txt” существует, ТО
	1. Сообщение “Файл уже существует!”

ИНАЧЕ

* 1. Создаем новый файл “stud.txt”
	2. Закрываем файл “stud.txt”
	3. Сообщение “Файл создан!”

ЕСЛИ ВСЕ

 Событие нажатия кнопки Button1 на форме Form2(Добавление записи)

1. ЕСЛИ файл “stud.txt” существует, ТО
	1. Вызов функции NameArr для получения R, NovArr[] и student[]
	2. ЕСЛИ хотя бы одно поле ввода пустое, ТО
		1. Сообщение “Не все поля заполнены!”

ИНАЧЕ

* + 1. Создаем новый файл “stud.txt”, с добавлением новых записей в конец файла
		2. string text = "";
		3. string text1 = "";
		4. Если (radioButton1.Checked) text = radioButton1.Text;
		5. Если (radioButton2.Checked) text = radioButton2.Text;
		6. Если (radioButton3.Checked) text = radioButton3.Text;
		7. Если (radioButton4.Checked) text = radioButton4.Text;
		8. Записываем строку, содержащую значения глобальных переменных, в конец файла
		9. i=0
		10. ПОКА i<R ВЫПОЛНИТЬ
			1. Записываем в файл строку NovArr[i-1]
			2. i=i+1

ПОКАВСЕ

* + 1. Закрываем файл
		2. Сообщение “Запись добавлена!”

ЕСЛИ ВСЕ

ИНАЧЕ

* 1. Сообщение “Файл не найден!”

ЕСЛИ ВСЕ

Событие нажатие кнопки Button1 на форме Form3(Поиск записи)

1. ЕСЛИ файл “stud.txt” существует, ТО
	1. ЕСЛИ поле ввода номера записи принимает пустое значение, ТО
		1. Сообщение “Введите номер записи”

ИНАЧЕ

* + 1. N = преобразованное в целое значение поля ввода для номера записи
		2. Вызов функции NameArr для получения R, RedArr[] и student[]
		3. ЕСЛИ N>0 и N-1<R, ТО
			1. Делаем доступными элементы управления
			2. Записываем строки из структуры student в поля ввода

ИНАЧЕ

* + - 1. Сообщение “Запись не найдена!”
			2. Делаем недоступными элементы управления

ЕСЛИ ВСЕ

 ЕСЛИ ВСЕ

ИНАЧЕ

* 1. Сообщение “Файл не найден!”

ЕСЛИВСЕ

 Событие нажатия кнопки Button3 на форме Form3(Изменение записи)

1. ЕСЛИ хотя бы одно поле ввода пустое, ТО
	1. Сообщение “Не все поля заполнены!”

ИНАЧЕ

* 1. Создаем новый файл “stud.txt”
	2. string text = "";
	3. string text1 = "";
	4. Если (radioButton1.Checked) text = radioButton1.Text;
	5. Если (radioButton2.Checked) text = radioButton2.Text;
	6. Если (radioButton3.Checked) text = radioButton3.Text;
	7. Если (radioButton4.Checked) text = radioButton4.Text;
	8. i=0
	9. ПОКА i<R ВЫПОЛНИТЬ
		1. ЕСЛИ i=N-1, ТО
			1. Записываем измененные значения глобальных переменных

ИНАЧЕ

* + - 1. Записываем без изменений в файл строку RedArr[i]

ЕСЛИ ВСЕ

* + 1. i=i+1

ПОКА ВСЕ

* 1. Закрываем файл
	2. Сообщение “Запись изменена”

ЕСЛИ ВСЕ

 Событие нажатия кнопки Button1 на форме Form4(Удаление записи)

1. ЕСЛИ файл “stud.txt” существует, ТО
	1. ЕСЛИ поле ввода номера записи принимает пустое значение, ТО
		1. Сообщение “Введите номер записи”

ИНАЧЕ

* + 1. N= преобразованное в целое значение поля ввода для номера записи
		2. Вызов функции NameArr для получения R, YdalArr[] и student[]
		3. ЕСЛИ N>0 и N-1<R, ТО
			1. Создаем новый файл “stud.txt”
			2. i=0
			3. ПОКА i <R ВЫПОЛНИТЬ
				1. ЕСЛИ i ≠ N-1, ТО

Записываем YdalArr[i] в следующую строку файла

ЕСЛИ ВСЕ

* + - * 1. i=i+1

ПОКА ВСЕ

* + - 1. Закрываем файл “stud.txt”
			2. Сообщение “Запись удалена!”

ИНАЧЕ

* + - 1. Сообщение “Запись не найдена!”

ЕСЛИ ВСЕ

ЕСЛИ ВСЕ

ИНАЧЕ

* 1. Сообщение “Файл не найден!”

ЕСЛИ ВСЕ

Событие Load на форме Form5

1. ЕСЛИ файл “stud.txt” существует, ТО
	1. Вызов функции NameArr для получения R, StrArray[] и student[]
	2. ЕСЛИ R=0, ТО
		1. Сообщение “Нет ни одной записи!”
		2. Закрыть форму

ИНАЧЕ

* + 1. i=0
		2. ПОКА i<R ВЫПОЛНИТЬ
			1. Добавляем строку в таблицу
			2. Заполняем строку таблицы соответствующими данными структуры student
			3. i=i+1

ПОКА ВСЕ

ЕСЛИ ВСЕ

ИНАЧЕ

* 1. Сообщение “Файл не найден!”
	2. Закрыть форму

ЕСЛИ ВСЕ

Событие нажатия кнопки Button1 на форме Form6(показать)

1. ЕСЛИ файл “stud.txt” существует, ТО
	1. ЕСЛИ поле ввода textBox1 принимает пустое значение, ТО
		1. Сообщение “Выберите № комнаты!”

ИНАЧЕ

* + 1. Очищаем listBox1
		2. Вызов функции Zap3Arr для получения R, Zap3Arr [] и student[]
		3. i=0
		4. ПОКА i<R ВЫПОЛНИТЬ
			1. ЕСЛИ student[i].num[2] =Convert.ToInt32(maskedTextBox1 .Text), ТО
				1. Добавляем student[i].name в listBox1

ЕСЛИ ВСЕ

* + - 1. i=i+1

ПОКА ВСЕ

* + 1. ЕСЛИ listBox1 пустой, ТО
			1. Сообщение “Записей по запросу не найдено”

 ЕСЛИ ВСЕ

ЕСЛИ ВСЕ

ЕСЛИ ВСЕ

Событие нажатия кнопки Button1 на форме Form7(показать)

1. ЕСЛИ поле ввода textBox1 принимает пустое значение, ТО
	1. Сообщение “Введите первую букву фамилии”

ИНАЧЕ

* 1. Очищаем listBox1
	2. Вызов функции Zap1Arr для получения R, Zap1Arr [] и student[]
	3. i=0
	4. ПОКА i<R ВЫПОЛНИТЬ
		1. ЕСЛИ student[i].name[0] = Convert.ToChar(textBox1 .Text), ТО
			1. Добавляем student[i].name в listBox1

 ЕСЛИ ВСЕ

* + 1. i=i+1

 ПОКА ВСЕ

* 1. ЕСЛИ listBox1 пустой, ТО
		1. Сообщение “Записей по запросу не найдено”

ЕСЛИ ВСЕ

ЕСЛИ ВСЕ

4 Формы

В этом разделе показан внешний вид всех форм.

1. Главная форма (Рис.1)



Рисунок 1 – Главная форма

2. Форма “Добавление записи” (Рис.2)



Рисунок 2 – Форма добавления записи

3. Форма “Редактирование записи” (Рис.3)



Рисунок 3 – Форма редактирование записи

4. Форма “Удаление записи” (Рис.4)



Рисунок 4 – Форма удаления записи

5. Форма “Просмотр записей” (Рис.5)



Рисунок 5 – Форма просмотра записи

6. Форма “Запрос 1” (Рис.6)



Рисунок 6 – Форма запроса 1

7. Форма “Запрос 2” (Рис.7)



Рисунок 7 – Форма запроса 2

5 Коды основных событий

Функция NameArr

public static int NameArr(out string[] NovArr, out Student[] student)

 {

 NovArr = new string[100];

 student = new Student[100];

 aFile = new FileStream("stud.txt", FileMode.Open);

 SR = new StreamReader(Global.aFile);

 int N = 0;// проверка кол-во строк в файле

 string[] Arr;

 while (!SR.EndOfStream)

 {

 NovArr[N] = SR.ReadLine();

 Arr = NovArr[N].Split('|');

 student[N].name = Arr[0];

 student[N].num = Convert.ToInt32(Arr[2]);

 if (Arr[3] == "Женский") student[N].sex = true;

 if (Arr[3] == "Мужской") student[N].sex = false;

 student[N].grupp = Arr[1];

 if (Arr[4] == "В активном поиске") student[N].sp = true;

 if (Arr[4] == "В отношениях") student[N].sp = false;

 N++;

 }

 SR.Close();

 return N;

 }

Выбор пункта меню создать ToolStripMenuItem\_Click на форме Form1

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

 {

 if (File.Exists("stud.txt"))

 {

 MessageBox.Show("Файл уже существует!");

 }

 else

 {

 Global.aFile = new FileStream("stud.txt", FileMode.Create);

 Global.aFile.Close();

 MessageBox.Show("Файл создан!");

 }

 }

Событие нажатия кнопки Button1 на форме Form2
private void button1\_Click(object sender, EventArgs e)

 {

 if (File.Exists("stud.txt"))

 {

 R = Global.NameArr(out NovArr, out student);

 if (textBox1.Text == "" || textBox2.Text == "" || maskedTextBox1.Text == "" || (!radioButton1.Checked && !radioButton2.Checked))

 {

 MessageBox.Show("Не все поля заполнены!");

 }

 else

 {

 Global.aFile = new FileStream("stud.txt", FileMode.Append);

 Global.SW = new StreamWriter(Global.aFile);

 string text = "";

 string text1 = "";

 if (radioButton1.Checked) text = radioButton1.Text;

 if (radioButton2.Checked) text = radioButton2.Text;

 if (radioButton3.Checked) text1 = radioButton3.Text;

 if (radioButton4.Checked) text1 = radioButton4.Text;

 Global.SW.WriteLine("{0}|{1}|{2}|{3}|{4}", textBox1.Text, textBox2.Text, maskedTextBox1.Text, text, text1);

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

 {

 Global.SW.WriteLine(NovArr[i - 1]);// запись массива строк в файл

 }

 Global.SW.Close();

 MessageBox.Show("Запись добавлена!");

 }

 }

 else

 {

 MessageBox.Show("Файл не найден!");

 }

 }

Событие нажатие кнопки Button1 на форме Form3

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

 {

 if (File.Exists("stud.txt"))

 {

 if (maskedTextBox4.Text == "")

 {

 MessageBox.Show("Введите номер записи!");

 }

 else

 {

 N = Convert.ToInt32(maskedTextBox4.Text);

 R = Global.NameArr(out RedArr, out student);

 if (N > 0 && (N - 1) < R)

 {

 textBox1.Enabled = true;

 textBox2.Enabled = true;

 radioButton1.Enabled = true;

 radioButton2.Enabled = true;

 radioButton3.Enabled = true;

 radioButton4.Enabled = true;

 maskedTextBox1.Enabled = true;

 button3.Enabled = true;

 textBox1.Text = student[N - 1].name;

 textBox2.Text = Convert.ToString(student[N - 1].grupp);

 maskedTextBox1.Text = Convert.ToString(student[N - 1].num);

 if (student[N - 1].sex) radioButton1.Checked = true;

 else radioButton2.Checked = true;

 if (student[N - 1].sp) radioButton3.Checked = true;

 else radioButton3.Checked = true;

 }

 else

 {

 MessageBox.Show("Запись не найдена!");

 textBox1.Enabled = false;

 textBox2.Enabled = false;

 radioButton1.Enabled = false;

 radioButton2.Enabled = false;

 radioButton3.Enabled = false;

 radioButton4.Enabled = false;

 maskedTextBox1.Enabled = false;

 button3.Enabled = false;

 }

 }

 }

 else

 {

 MessageBox.Show("Файл не найден!");

 }

 }

Событие нажатия кнопки Button3 на форме Form3

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

 {

 if (textBox1.Text == "" || maskedTextBox1.Text == "" || maskedTextBox4.Text == "")

 {

 MessageBox.Show("Не все поля заполнены!");

 }

 else

 {

 Global.aFile = new FileStream("stud.txt", FileMode.Create);

 Global.SW = new StreamWriter(Global.aFile);

 string text = "";

 string text1 = "";

 if (radioButton1.Checked) text = radioButton1.Text;

 if (radioButton2.Checked) text = radioButton2.Text;

 if (radioButton1.Checked) text1 = radioButton3.Text;

 if (radioButton2.Checked) text1 = radioButton4.Text;

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

 {

 if (i == N - 1)

 {

 Global.SW.WriteLine("{0}|{1}|{2}|{3}|{4}", textBox1.Text, textBox2.Text, maskedTextBox1.Text, text, text1);// запись измененной строки

 }

 else

 {

 Global.SW.WriteLine(RedArr[i]);// запись без изменений

 }

 }

 Global.SW.Close();

 MessageBox.Show("Запись изменена!");

 }

 }

Событие нажатия кнопки Button1 на форме Form4

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

 {

 if (File.Exists("stud.txt"))

 {

 if (maskedTextBox1.Text == "")

 {

 MessageBox.Show("Введите номер записи!");

 }

 else

 {

 string[] YdalArr;

 Global.Student[] student;

 int R = Global.NameArr(out YdalArr, out student);

 int N = Convert.ToInt32(maskedTextBox1.Text);

 if (N > 0 && N - 1 < R)

 {

 Global.aFile = new FileStream("stud.txt", FileMode.Create);

 Global.SW = new StreamWriter(Global.aFile);

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

 {

 if (i != N - 1)

 Global.SW.WriteLine(YdalArr[i]); // запись строк , кроме удаленной

 }

 Global.SW.Close();

 MessageBox.Show("Запись удалена!");

 }

 else

 {

 MessageBox.Show("Запись не найдена!");

 }

 }

 }

 else

 {

 MessageBox.Show("Файл не найден!");

 }

 }

Событие Load на форме Form5

private void Form5\_Load(object sender, EventArgs e)

 {

 if (File.Exists("stud.txt"))

 {

 string[] StrArray;

 Global.Student[] student;

 int R = Global.NameArr(out StrArray, out student);

 if (R == 0)

 {

 MessageBox.Show("Нет ни одной записи!");

 Close();

 }

 else

 {

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

 {

 dataGridView1.Rows.Add();

 dataGridView1[0, i].Value = student[i].name;

 dataGridView1[1, i].Value = student[i].grupp;

 dataGridView1[2, i].Value = student[i].num;

 if (student[i].sex) dataGridView1[3, i].Value = "Женский";

 else dataGridView1[3, i].Value = "Мужской";

 if (student[i].sp) dataGridView1[4, i].Value = "В активном поиске";

 else dataGridView1[4, i].Value = "В отношениях";

 }

 }

 }

 else

 {

 MessageBox.Show("Файл не найден!");

 Close();

 }

 }

Событие нажатия кнопки Button1 на форме Form6

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

 {

 if (File.Exists("stud.txt"))

 {

 if (maskedTextBox1.Text == "")

 {

 MessageBox.Show("Выберите № комнаты!");

 }

 else

 {

 listBox1.Items.Clear();

 string[] Zap3Arr;

 Global.Student[] student;

 int R = Global.NameArr(out Zap3Arr, out student);

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

 {

 if (student[i].num[2] == Convert.ToInt32(maskedTextBox1.Text))

 {

 listBox1.Items.Add(student[i].name);// запись дисциплин на основе выбранной специальности

 }

 }

 if (listBox1.Items.Count == 0) MessageBox.Show("Записей по запросу не найдено!");

 }

 }

 else

 {

 MessageBox.Show("Файл не найден!");

 }

 }

Событие нажатия кнопки Button1 на форме Form7

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

 {

 if (textBox1.Text == "")

 {

 MessageBox.Show("Введите первую букву фамилии ");

 }

 else

 {

 listBox1.Items.Clear();

 string[] Zap1Arr;

 Global.Student[] student;

 int R = Global.NameArr(out Zap1Arr, out student);

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

 {

 if (student[i].name[0] == Convert.ToChar(textBox1.Text))

 {

 listBox1.Items.Add(student[i].name);

 }

 }

 if (listBox1.Items.Count == 0) MessageBox.Show("Записей по запросу не найдено");

 }

 }

6 Таблица тестов

В этом разделе описаны примеры тестов для проверки работоспособности программы.

Таблица 2 –Тесты для проверки работоспособности программы

|  |  |  |  |
| --- | --- | --- | --- |
| **N теста** | **Назначение теста** | **Входные данные для теста** | **Выходные данные для теста** |
| 1 | Создание файла |  | Сообщение “Файл создан” (Рис.8) |
| 2 | Повторное создание файла |  | Сообщение “Файл уже существует” (Рис.9) |
| 3 | Добавление записи (успех) | Петров Петр ПетровичРТб-1623МужскойВ активном поиске | Сообщение “Запись добавлена” (Рис.10) |
| 4 | Ошибка при добавлении записи (заполнены не все поля) | Иванова Инна Ивановна27ЖенскийВ отношениях | Сообщение “Не все поля заполнены” (Рис.11) |
| 5 | Ошибка при добавлении записи (файл не найден) |  | Сообщение “Файл не найден” (Рис.12) |
| 6 | Удаление записи (успех) | 2 | Сообщение “Запись удалена” (Рис.13) |
| 7 | Ошибка при удалении записи (не введен номер записи) |  | Сообщение “Введите номер записи” (Рис.14) |
| 8 | Ошибка при удалении записи (запись не найдена) | 99 | Сообщение “Запись не найдена” (Рис.15) |
| 9 | Ошибка при удалении записи (файл не найден) | 3 | Сообщение “Файл не найден” (Рис.16) |
| 10 | Поиск записи для редактирования (успех) | 2 | Иванов Иван Ивановичтрб-1227МужскойВ отношениях |
| 11 | Поиск записи для редактирования (не введен номер записи) |  | Сообщение “Введите номер записи” (Рис.18) |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | Поиск записи для редактирования (запись не найдена) | 99 | Сообщение “Запись не найдена” (Рис.19) |
| 13 | Поиск записи для редактирования (файл не найден) | 1 | Сообщение “Записей по запросу не найдено” (Рис.20) |
| 14 | Редактирование записи (успех) | 2 | Иванов Иван ИвановичПРм-1227МужскойВ активном поискеСообщение “Запись изменена” (Рис.21) |
| 15 | Редактирование записи(не все поля заполнены) | 3 | Иванова Инна ИвановнаТРб-16ЖенскийВ активном поискеСообщение “ Не все поля заполнены” (Рис.22) |
| 16 | Просмотр записей (успех) |  | Рис.23 |
| 17 | Просмотр записей (нет записей) |  | Сообщение “Нет ни одной записи” (Рис.24) |
| 18 | Просмотр записей (нет файла) |  | Сообщение “Файл не найден” (Рис.25) |
| 19 | Запрос 1 (успех) | 27 |  Иванов Иван ИвановичИванова Инна Ивановна |
| 20 | Запрос 1 (нет записей) | 2 | Сообщение “Записей по запросу не найдено” (Рис.27) |
| 21 | Запрос 1(пустые поля ввода) |  | Сообщение “Введите № комнаты” (Рис.28) |
| 22 | Запрос 1 (нет файла) | 8 | Сообщение “Файл не найден” (Рис.29) |
| 23 | Запрос 2 (успех) | И |  Иванов Иван ИвановичИванова Инна Ивановна |
| 24 | Запрос 2 (пустое поле ввода) |  | Сообщение “Введите первую букву фамилии” (Рис.31) |
| 25 | Запрос 2(Нет файла) |  | Сообщение “Файл не найден”(Рис.32) |
| 26 | Запрос 2 (Нет записей) | Г | Сообщение “Записей по запросу не найдено ”(Рис.33) |

7 Результаты тестирования

В этом разделе показан внешний вид и выполненное событие во всех формах при выполнении тестов.

Результат тестирования создания файла (тест 1) показан на рисунке 8.



Рисунок 8 - Результат выполнения теста 1

Результат тестирования создания файла, когда файл уже существует(тест 2) показан на рисунке 9



Рисунок 9- Результат выполнения теста 2

Результат тестирования добавления записи(тест 3) показан на рисунке 10



Рисунок 10 - Результат выполнения теста 3

Результат тестирования добавления записи, когда не все поля заполнены(тест 4) показан на рисунке 11



Рисунок 11 - Результат выполнения теста 4

Результат тестирования добавления записи, когда файл не создан(тест 5) показан на рисунке 12



Рисунок 12 - Результат выполнения теста 5

Результат тестирования удаления записи (тест 6) показан на рисунке 13



Рисунок 13- Результат выполнения теста 6

Результат тестирования удаления записи, когда не введен номер записи (тест 7) показан на рисунке 14



Рисунок 14 - Результат выполнения теста 7

Результат тестирования удаления записи, когда запись с таким номером не существует(тест 8) показан на рисунке 15



Рисунок 15 - Результат выполнения теста 8

Результат тестирования удаления записи, когда файл не создан(тест 9) показан на рисунке 16



Рисунок 16 - Результат выполнения теста 9

Результат тестирования изменения записи, нажатие на кнопку “Найти” (тест 10) показан на рисунке 17



Рисунок 17 - Результат выполнения теста 10

Результат тестирования изменения записи, нажатие на кнопку “Найти” когда не введен номер записи (тест 11) показан на рисунке 18



Рисунок 18 - Результат выполнения теста 11

Результат тестирования изменения записи, нажатие на кнопку “Найти” когда запись с таким номером не существует (тест 12) показан на рисунке 19



Рисунок 19 - Результат выполнения теста 12

Результат тестирования изменения записи, нажатие на кнопку “Найти” когда файл не создан (тест 13) показан на рисунке 20



Рисунок 20 - Результат выполнения теста 13

Результат тестирования изменения записи, нажатие на кнопку “Изменить”(тест 14) показан на рисунке 21



Рисунок 21 - Результат выполнения теста 14

Результат тестирования изменения записи, нажатие на кнопку “Изменить”, когда не все поля заполнены (тест 15) показан на рисунке 22



Рисунок 22 - Результат выполнения теста 15

Результат тестирования формы просмотра записей, (тест 16) показан на рисунке 23



Рисунок 23 - Результат выполнения теста 16

Результат тестирования формы просмотра записей, когда в файле нет записей(тест 17) показан на рисунке 24



Рисунок 24 - Результат выполнения теста 17

Результат тестирования формы просмотра записей, когда файл не создан (тест 18) показан на рисунке 25



Рисунок 25 - Результат выполнения теста 18

Результат тестирования запроса 1, (тест 19) показан на рисунке 26



Рисунок 26 - Результат выполнения теста 19

Результат тестирования запроса 1, когда не найдено записей по запросу (тест 20) показан на рисунке 27



Рисунок 27 - Результат выполнения теста 20

Результат тестирования запроса 1, когда не введен № комнаты (тест 21) показан на рисунке 28



Рисунок 28 - Результат выполнения теста 21

Результат тестирования запроса 1, когда файл не создан (тест 22) показан на рисунке 29



Рисунок 29 - Результат выполнения теста 22

Результат тестирования запроса 2 (тест 23) показан на рисунке 30



Рисунок 30 - Результат выполнения теста 23

Результат тестирования запроса 2, когда не введена 1 буква фамилии (тест 24) показан на рисунке 31



Рисунок 31 - Результат выполнения теста 24

Результат тестирования запроса 2, когда файл не создан (тест 25) показан на рисунке 32



Рисунок 32 - Результат выполнения теста 25

Результат тестирования запроса 2, когда не существует фамилий на эту букву(тест 26) показан на рисунке 33



Рисунок 33 - Результат выполнения теста 26

Все тесты программы сошлись с предполагаемой таблицей тестов, из этого следует, что программа работает без ошибок.

Заключение

C# — простой, современный, объектно-ориентированный язык программирования. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.

**В результате курсового проекта было разработано Windows-приложение, которое работает с базой данных студентов.** Способы выполнения действий – согласно индивидуальному заданию. **В приложении можно создать файл** или использовать уже существующий файл**: “stud.txt”. Добавлять новые данные в БД, изменять БД и также удалять записи из нее по их номеру.** В приложение реализован просмотр файла **и выполнение двух запросов: поиск** жильцов по номеру комнаты и **поиск** студентов с фамилией на заданную букву**.** При ошибочных ситуациях (отсутствие записи, отсутствие файла и т.д.) выдаются сообщения.

Приложение было протестировано на самые разнообразные ошибки, при разработке все они были исправлены.

Таким образом, цель работы достигнута, поставленные задачи выполнены. Полученные знания и навыки в создании оконного приложения на языке C# пригодятся мне в будущем для совершенствования в профессиональной сфере.

Список использованных источников

1. Сосинская С.С. «Обработка файлов»: Методические указания по выполнению курсового проекта» Электронный каталог кафедры.
2. Гуннерсон Э. Введение в C#. СПб: Питер, 2001.
3. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013. - 320 c.
4. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013. - 496 c.
5. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 486 c.
6. Биллинг В.А. Основы программирования на C#. - Т.: Интернет-университет информационных технологий, Бином, 2012. - 488 с.
7. Евдокимов П.В. C# на примерах. - М.: Наука и техника,2016. - 304 с.
8. Нейгел К., Ивьен Б., Глинн Д., Уотсон К., Скиннер М. C# 4.0 и платформа .NET 4, 2011.