Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

Допускаю к защите

Руководитель:

В.А. Харахинов

И.О. Фамилия

Обработка файлов на языке C#

Наименование темы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовой работе по дисциплине

Программирование на языке высокого уровня

1.020.00.00 ПЗ

обозначение документа

Выполнил студент группы ИСТБ-19-1 Колосов В.Г.

шифр подпись Фамилия И.О.

Нормоконтроль Харахинов В.А.

подпись Фамилия И.О.

Курсовая работа защищена с оценкой:

Иркутск 2020г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

По курсу Программирование на языке высокого уровня

Студенту Колосову Вячеславу Германовичу

(фамилия, инициалы)

Тема работы Обработка файлов на языке C#

Исходные данные Вариант 12-1-3-2

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 02 /2020 г.

Задание получил студент Колосов В.Г.

подпись Фамилия И.О.

Дата представления работы руководителю 22 / 05 /2020 г.

Руководитель курсовой работы Харахинов В.А.

подпись Фамилия И.О.

Содержание

[Введение 4](#_Toc41559743)

[Индивидуальное задание 5](#_Toc41559744)

[1 Иерархия форм и функций с описанием их назначения 6](#_Toc41559745)

[2 События форм 7](#_Toc41559746)

[3 Алгоритмы основных событий 9](#_Toc41559747)

[4 Формы 14](#_Toc41559748)

[5 Коды основных событий 18](#_Toc41559749)

[6 Таблица тестов 27](#_Toc41559750)

[7 Результаты тестирования 29](#_Toc41559751)

[Заключение 42](#_Toc41559752)

[Список использованных источников 43](#_Toc41559753)

Введение

С# принадлежит семье языков программирования Microsoft и был разработан в 2000 году и стал частью первого релиза .NET framework. Язык С# сочетает в себе надежность С++ с дополнительными возможностями Java. Поэтому если вы хорошо знаете Java, можно легко переключиться на С# и наоборот.

По моему мнению C# это лучший язык программирования для разработки Windows-приложений. Также на C# можно разрабатывать мобильные приложения для IOS и Android благодаря фреймворку Xamarin. А также C# поддерживают многие игровые движки , такие как: Unity, WAVE, Xenko и т.п. Все эти факторы говорят о востребованности и современности данного языка программирования.

Требуется на основании индивидуального задания разработать иерархию форм и функций с описанием их назначения, 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.

Индивидуальное задание

Вариант 12-1-3-2

12- Файл “Сотрудники” (фамилия и инициалы сотрудника , номер телефона, адрес).

1. найти телефон сотрудника по его фамилии;
2. найти список сотрудников , чьи телефоны начинаются с заданной цифры, то есть принадлежат одной АТС;

2- Добавление в начало файла

1. Иерархия форм и функций с описанием их назначения

Иерархия показана на рис. 1.

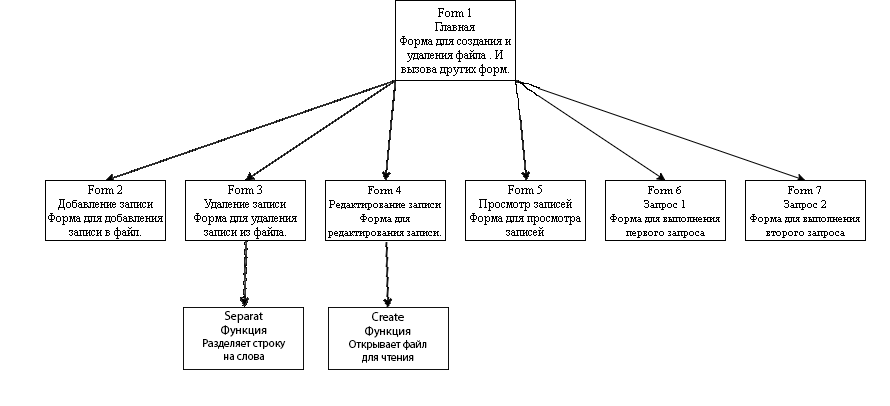


Рисунок 1 – Иерархия форм и функций

1. События форм

В табл. 1 перечислены события форм.

Таблица 1 – События форм

|  |  |
| --- | --- |
| Название события | Назначение |
| Form 1 | |
| создатьToolStripMenuItem\_Click | Создания нового файла |
| удалитьToolStripMenuItem1\_Click | Удаление файла |
| добавитьToolStripMenuItem\_Click | Вызов Form 2 для добавления записи к файлу |
| удалитьToolStripMenuItem\_Click | Вызов Form 3 для удаления записи из файла |
| редактироватьToolStripMenuItem\_Click | Вызов Form 4 для редактирования записей |
| 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 |
| Form 2 | |
| button1\_Click | Добавление записи в файл |
| button2\_Click | Очистить все поля ввода |
| Form 3 | |
| button1\_Click | Удаление записи из файла по ее номеру |
| button2\_Click | Очистить поле ввода |
| Form 4 | |
| Form4\_Load | При загрузке формы часть полей ввода и кнопок становятся не активна |
| button1\_Click | Редактирование записи в файле |
| button2\_Click | Поиск записи и заполнение полей ввода |
| button3\_Click | Очистить все поля ввода |
| Form 5 | |
| Form5\_Load | Загрузка записей из файла в массив структур |
| Form 6 | |
| button1\_Click | Вывод номера телефона по фамилии |
| button2\_Click | Очистка поля ввода |
| Form 7 | |
| button1\_Click | Вывод списка сотрудников по первой цифре номера телефона |
| button2\_Click | Очистка поля ввода |

3 Алгоритмы основных событий

**Form1**

**создатьToolStripMenuItem\_Click**

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

**удалитьToolStripMenuItem1\_Click**

1. Удаляем файл “stud.txt”
2. Сообщение “Файл удален!”

**Form2**

**button1\_Click**

ЕСЛИ файл “stud.txt” существует, ТО

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

ИНАЧЕ

* + 1. Присваиваем глобальным переменным значения введенные в maskedTextbox
    2. Записываем в начало файла новую строку , разделяя слова “|”
    3. Сообщение “Запись добавлена!”

ЕСЛИ ВСЕ

ИНАЧЕ

* 1. Сообщение “Файл не найден!”

ЕСЛИ ВСЕ

**Form3**

**Button1\_Click**

1. ЕСЛИ файл “stud.txt” существует, ТО
   1. ЕСЛИ maskedTextBox активно
      1. ЕСЛИ maskedTextBox пустое
         1. Сообщение “Введите номер записи”

ИНАЧЕ

* + - 1. Создаем новый массив dats длинной 20
      2. Создаем новый массив separator c разделительным знаком “|”
      3. Вызываем функция Create
      4. nd = вызываем функцию Separat
      5. ЕСЛИ nd<текста в текстовом поле
         1. Сообщение “Нет записи”

ИНАЧЕ

Открываем файл “stud.txt" для записи

ПОКА i<nd

ЕСЛИ i+1!= тексту в maskedTextBox

Перезаписываем оставшиеся строки в файл

ЕСЛИ ВСЕ

ПОКА ВСЕ

* + - * 1. Закрываем файл
        2. Очищаем maskedTextBox
        3. Сообщение “Запись удалена!”
  1. ИНАЧЕ ЕСЛИ comboBox активно
     1. ЕСЛИ comboBox пустое
        1. Сообщение “Введите номер записи”

ИНАЧЕ

* + - 1. Создаем новый массив dats длинной 20
      2. Создаем новый массив separator c разделительным знаком “|”
      3. Вызываем функция Create

ПОКА i<nd

* + - 1. Добавляем элементы в comboBox

ПОКА ВСЕ

* + - 1. ЕСЛИ nd<текста в текстовом поле
         1. Сообщение “Нет записи”

ИНАЧЕ

Открываем файл “stud.txt" для записи

ПОКА i<nd

ЕСЛИ i+1!= тексту в comboBox

Перезаписываем оставшиеся строки в файл

ЕСЛИ ВСЕ

ПОКА ВСЕ

* + - * 1. Закрываем файл

Очищаем comboBox

* + - * 1. Сообщение “Запись удалена!”

1. ИНАЧЕ
   1. Сообщение “Файл не создан”

ЕСЛИ ВСЕ

**Public int Separat**

1. ПОКА файл не дочитан до конца
   1. Разделяем строку на слова
   2. n++

ПОКА ВСЕ

1. Возвращаем значение n

**Form4**

**Button1\_Click**

1. ЕСЛИ файл “stud.txt” существует, ТО
   1. ЕСЛИ поле ввода номера записи принимает пустое значение, ТО
      1. Сообщение “Не все поля заполнены!”

ИНАЧЕ

* + 1. Создаем новый массив separator c разделительным знаком “|”
    2. Создаем новый массив dats длинной 20
    3. Открываем файл для чтения

ПОКА файл не дочитан до конца

* + - 1. Разделяем строку на слова
      2. Выводим слова из строки в текстовые поля

ПОКА ВСЕ

* + 1. Закрываем файл
    2. Открываем файл для записи
    3. Перезаписываем строки
    4. Сообщение “Запись изменена”

ИНАЧЕ

* 1. Сообщение “Файл не создан”

ЕСЛИ ВСЕ

**Button2\_Click**

1. ЕСЛИ файл “stud.txt” существует, ТО
   1. ЕСЛИ поле ввода номера записи принимает пустое значение, ТО
      1. Сообщение “ Не все поля заполнены!”

ИНАЧЕ

* + 1. Создаем новый массив separator c разделительным знаком “|”
    2. Создаем новый массив dats длинной 20
    3. Разделяем строку на слова
    4. Выводим слова из строки в текстовые поля

ИНАЧЕ

* 1. Сообщение “Файл не создан”

ЕСЛИ ВСЕ

**Form5**

**Load**

1. ЕСЛИ файл “stud.txt” существует, ТО
   1. Создаем новый массив separator c разделительным знаком “|”
   2. Открываем файл “stud.txt” для чтения
   3. nr=0

ПОКА файл не дочитан до конца

* + 1. Разделяем строку на слова
    2. Заполняем строку таблицы № nr словами
    3. nr++

ПОКА ВСЕ

* 1. Закрываем файл

ИНАЧЕ

* 1. Сообщение “Файл не создан!”

ЕСЛИ ВСЕ

**Form6**

**Button1\_Click**

1. ЕСЛИ файл “stud.txt” существует, ТО
   1. ЕСЛИ поле ввода maskedtextBox1 принимает пустое значение, ТО
      1. Сообщение “Не все поля заполнены!”

ИНАЧЕ

* + 1. Создаем новый массив separator c разделительным знаком “|”
    2. Создаем новый массив dats длинной 20
    3. Разделяем строку на слова
    4. ЕСЛИ удается найти введенную фамилию в файле
       1. Вывод номера телефона соответствующий фамилии

ИНАЧЕ

* + - 1. Сообщение “Сотрудник не найден!”

ЕСЛИ ВСЕ

ИНАЧЕ

1. Сообщение “Файл не создан!”

ЕСЛИ ВСЕ

**Form7  
Button1\_Click**

1. ЕСЛИ файл “stud.txt” существует, ТО
   1. ЕСЛИ поле ввода maskedtextBox1 принимает пустое значение, ТО
      1. Сообщение “Не все поля заполнены!”

ИНАЧЕ

* + 1. Создаем новый массив separator c разделительным знаком “|”
    2. Создаем новый массив dats длинной 20
    3. Разделяем строку на слова
    4. ЕСЛИ удается найти в файле номер телефона начинающийся на введенную цифру
       1. Вывод номера телефонов, начинающийся на введенную цифру

ИНАЧЕ

* + - 1. Сообщение “Сотрудников с таким номером нет!”

ЕСЛИ ВСЕ

ИНАЧЕ

1. Сообщение “Файл не создан!”

ЕСЛИ ВСЕ

4 Формы

В этом разделе показан внешний вид всех форм.

1. Главная форма (Рис. 2)

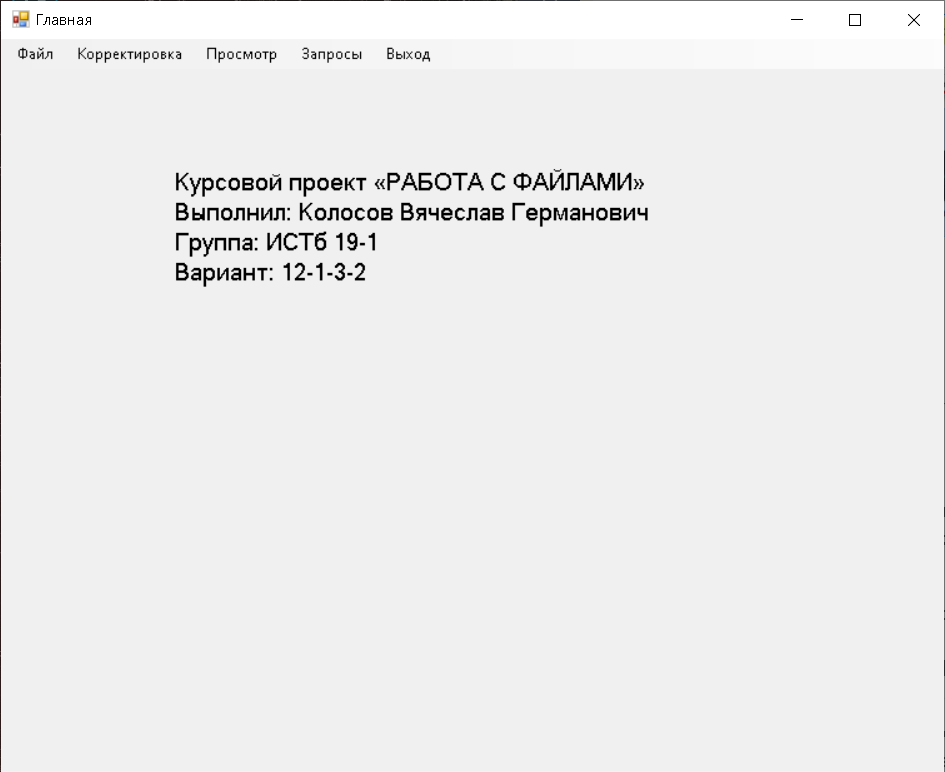


Рисунок 2 – Главная форма

2. Форма “Добавление записи” (Рис. 3)

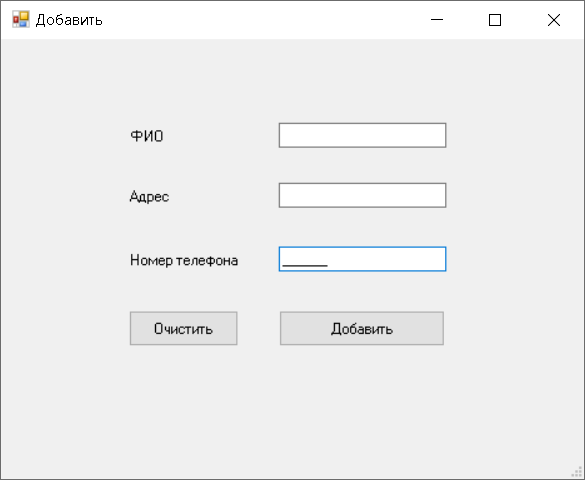


Рисунок 3 – Форма добавления записи

3. Форма “Редактирование записи” (Рис. 4)

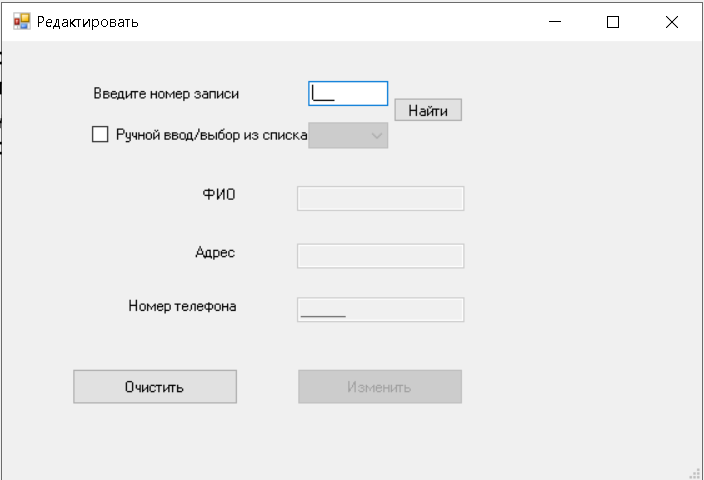


Рисунок 4– Форма редактирование записи

4. Форма “Удаление записи” (Рис. 5)

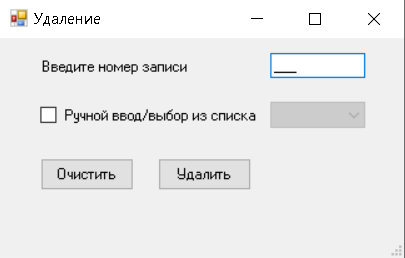


Рисунок 5 – Форма удаления записи

5. Форма “Просмотр записей” (Рис. 6)

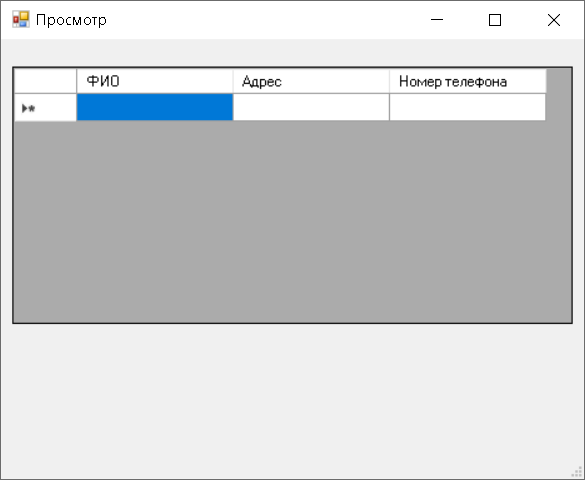


Рисунок 6 – Форма просмотра записи

6. Форма “Запрос 1” (Рис. 7)

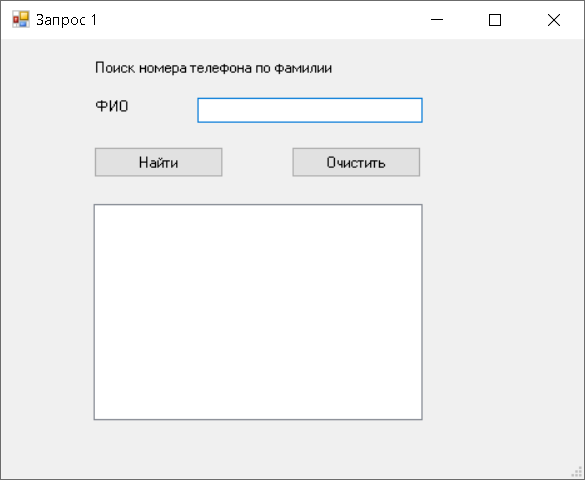


Рисунок 7 – Форма запроса 1

7. Форма “Запрос 2” (Рис. 8)

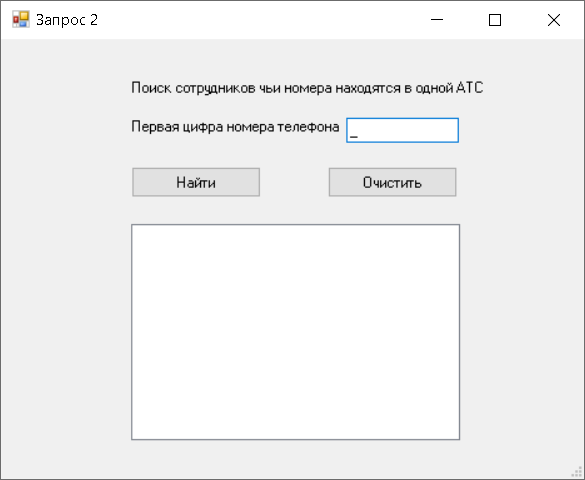


Рисунок 8 – Форма запроса 2

5 Коды основных событий

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!");

}

}

Выбор пункта меню удалить ToolStripMenuItem1\_Click на форме Form1

private void удалитьToolStripMenuItem1\_Click(object sender, EventArgs e)

{

File.Delete("stud.txt");

MessageBox.Show("Файл удален!");

}

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

{

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

{

if (maskedTextBox1.Text == "" || maskedTextBox2.Text == "" || maskedTextBox3.Text == "")

{

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

}

else

{

Global.name = maskedTextBox3.Text;

Global.phnomber = maskedTextBox1.Text;

Global.address = maskedTextBox2.Text;

string fileName = "stud.txt";

string firstLine = String.Format("{0}|{1}|{2}",

Global.name, Global.address, Global.phnomber);

File.WriteAllText(fileName, firstLine + Environment.NewLine + File.ReadAllText(fileName, Encoding.Default), Encoding.Default);

maskedTextBox3.Text = "";

maskedTextBox2.Text = "";

maskedTextBox1.Text = "";

MessageBox.Show("Запись добавлена!");

}

}

else

{

MessageBox.Show("Файл не создан");

}

}

Событие нажатие кнопки Button1 на форме Form3

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

{

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

{

if (maskedTextBox1.Enabled == true)

{

if (maskedTextBox1.Text == "")

{

MessageBox.Show("Введите номер строки");

}

else

{

dt[] dats = new dt[20];

char[] separator = new char[] { '|' };

Create();

Global.nd = Separat(dats, Global.n, separator);

if (Global.nd < Convert.ToInt32(maskedTextBox1.Text))

MessageBox.Show("Нет записи");

else

{

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

Global.sw = new StreamWriter(Global.aFile, Encoding.Default);

for (int i = 0; i < Global.nd; i++)

{

if ((i + 1) != Convert.ToInt32(maskedTextBox1.Text))

{

Global.name = dats[i].name;

Global.phnomber = dats[i].phnomber;

Global.address = dats[i].address;

Global.sw.WriteLine("{0}|{1}|{2}", Global.name, Global.address, Global.phnomber);

}

}

Global.sw.Close();

maskedTextBox1.Clear();

MessageBox.Show("Запись удалена!");

}

}

}

else if (comboBox1.Enabled == true)

{

if (comboBox1.Text == "")

{

MessageBox.Show("Выберите номер строки");

}

else

{

char[] separator = new char[] { '|' };

Create();

dt[] dats = new dt[20];

Global.nd = Separat(dats, Global.n, separator);

for (int i = 0; i < Global.nd; i++)

{

comboBox1.Items.Add(i + 1);

}

if (Global.nd < Convert.ToInt32(comboBox1.Text))

MessageBox.Show("Нет записи");

else

{

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

Global.sw = new StreamWriter(Global.aFile, Encoding.Default);

for (int i = 0; i < Global.nd; i++)

{

if ((i + 1) != Convert.ToInt32(comboBox1.Text))

{

Global.name = dats[i].name;

Global.phnomber = dats[i].phnomber;

Global.address = dats[i].address;

Global.sw.WriteLine("{0}|{1}|{2}", Global.name, Global.address, Global.phnomber);

}

}

Global.sw.Close();

checkBox1.Checked = false;

MessageBox.Show("Запись удалена!");

}

}

}

}

else

{

MessageBox.Show("Файл не создан");

}

}

Событие изменения состояния чекбокса на форме Form3

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

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

{

if (checkBox1.Checked == true)

{

maskedTextBox1.Enabled = false;

comboBox1.Enabled = true;

char[] separator = new char[] { '|' };

Create();

dt[] dats = new dt[20];

Global.nd = Separat(dats, Global.n, separator);

for (int i = 0; i < Global.nd; i++)

{

comboBox1.Items.Add(i + 1);

}

}

else

{

maskedTextBox1.Enabled = true;

comboBox1.Enabled = false;

comboBox1.Items.Clear();

}

}

else

{

MessageBox.Show("Файл не создан");

}

}

Метод Separat на форме Form3

public int Separat(dt[] dats, int n, char[] separator)

{

while (!Global.sr.EndOfStream)

{

string[] slov = Global.sr.ReadLine().Split(separator);

dats[n].name = slov[0];

dats[n].address = slov[1];

dats[n].phnomber = slov[2];

n++;

}

Global.sr.Close();

return n;

}

Метод Create на форме Form3

public void Create()

{

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

Global.sr = new StreamReader(Global.aFile, Encoding.Default);

}

Событие нажатия кнопки Button1 на форме Form4

private void button1\_Click\_1(object sender, EventArgs e)

{

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

{

if (maskedTextBox1.Text == "" || maskedTextBox2.Text == "" || maskedTextBox3.Text == "")

{

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

}

else

{

char[] separator = new char[] { '|' };

Create();

int nd = 0;

dt[] dats = new dt[20];

while (!Global.sr.EndOfStream)

{

string[] slov = Global.sr.ReadLine().Split(separator);

if (maskedTextBox4.Text == "")

{

if (nd == Convert.ToInt32(comboBox1.Text) - 1)

{

slov[0] = maskedTextBox3.Text;

slov[1] = maskedTextBox2.Text;

slov[2] = maskedTextBox1.Text;

}

}

else

{

if (nd == Convert.ToInt32(maskedTextBox4.Text) - 1)

{

slov[0] = maskedTextBox3.Text;

slov[1] = maskedTextBox2.Text;

slov[2] = maskedTextBox1.Text;

}

}

dats[nd].name = slov[0];

dats[nd].address = slov[1];

dats[nd].phnomber = slov[2];

nd++;

}

Global.sr.Close();

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

Global.sw = new StreamWriter(Global.aFile, Encoding.Default);

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

{

Global.sw.WriteLine($"{dats[i].name}|{dats[i].address}|{dats[i].phnomber}");

}

MessageBox.Show("Запись изменена!");

Global.sw.Close();

maskedTextBox4.Enabled = true;

maskedTextBox1.Enabled = false;

maskedTextBox2.Enabled = false;

maskedTextBox3.Enabled = false;

button1.Enabled = false;

maskedTextBox1.Clear();

maskedTextBox2.Clear();

maskedTextBox3.Clear();

maskedTextBox4.Clear();

checkBox1.Checked = false;

checkBox1.Enabled = true;

comboBox1.SelectedItem = null;

comboBox1.SelectedIndex = -1;

}

}

else

{

MessageBox.Show("Файл не создан");

}

}

Событие нажатия кнопки Button2 на форме Form4

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

{

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

{

if (maskedTextBox4.Enabled==true)

{

int nd = 0;

int n;

string lin = "";

dt[] dats = new dt[20];

char[] separator = new char[] { '|' };

Create();

while (!Global.sr.EndOfStream)

{

lin = Global.sr.ReadLine();

string[] slov = lin.Split(separator);

dats[nd].address = slov[1];

dats[nd].phnomber = slov[2];

dats[nd].name = slov[0];

nd++;

if (!int.TryParse(maskedTextBox4.Text, out n) )

{

MessageBox.Show("Ошибка");

}

else

{

if (nd == Convert.ToInt32(maskedTextBox4.Text))

{

maskedTextBox3.Text = slov[0];

maskedTextBox2.Text = slov[1];

maskedTextBox1.Text = slov[2];

}

}

}

Global.sr.Close();

if (int.TryParse(maskedTextBox4.Text, out n))

{

if (nd < Convert.ToInt32(maskedTextBox4.Text) || maskedTextBox4.Text == "0")

{

MessageBox.Show("Нет записи");

}

else

{

maskedTextBox1.Enabled = true;

maskedTextBox2.Enabled = true;

maskedTextBox3.Enabled = true;

button1.Enabled = true;

maskedTextBox4.Enabled = false;

checkBox1.Enabled = false;

}

}

else

{

MessageBox.Show("Ошибка");

}

}

else if(comboBox1.Enabled == true)

{

int nd = 0;

int n;

string lin = "";

dt[] dats = new dt[20];

char[] separator = new char[] { '|' };

Create();

while (!Global.sr.EndOfStream)

{

lin = Global.sr.ReadLine();

string[] slov = lin.Split(separator);

dats[nd].address = slov[1];

dats[nd].phnomber = slov[2];

dats[nd].name = slov[0];

nd++;

if (!int.TryParse(comboBox1.Text, out n))

{

MessageBox.Show("Ошибка");

}

else

{

if (nd == Convert.ToInt32(comboBox1.Text))

{

maskedTextBox3.Text = slov[0];

maskedTextBox2.Text = slov[1];

maskedTextBox1.Text = slov[2];

}

}

}

Global.sr.Close();

if (int.TryParse(comboBox1.Text, out n))

{

if (nd < Convert.ToInt32(comboBox1.Text))

MessageBox.Show("Нет записи");

else

{

maskedTextBox1.Enabled = true;

maskedTextBox2.Enabled = true;

maskedTextBox3.Enabled = true;

button1.Enabled = true;

comboBox1.Enabled = false;

checkBox1.Enabled = false;

}

}

else

{

MessageBox.Show("Ошибка");

}

}

}

else

{

MessageBox.Show("Файл не создан!");

}

}

Событие Load на форме Form5

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

{

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

{

string lin;

char[] separator = new char[] { '|' };

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

Global.sr = new StreamReader(Global.aFile, Encoding.Default);

int nr = 0;

while (!Global.sr.EndOfStream)

{

lin = Global.sr.ReadLine();

string[] slov = lin.Split(separator);

Global.name = slov[0];

Global.address = slov[1];

Global.phnomber = slov[2];

this.dataGridView1.Rows.Add();

dataGridView1[0, nr].Value = Global.name;

dataGridView1[1, nr].Value = Global.address;

dataGridView1[2, nr].Value = Convert.ToString(Global.phnomber);

nr++;

}

Global.sr.Close();

}

else

{

MessageBox.Show("Файл не создан");

}

}

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("Не все поля заполнены!");

}

else

{

string lin;

char[] separator = new char[] { ',' };

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

Global.sr = new StreamReader(Global.aFile, Encoding.Default);

int n = 1;

while (!Global.sr.EndOfStream)

{

lin = Global.sr.ReadLine();

string[] slov = lin.Split(separator);

Global.name = slov[0];

Global.address = slov[1];

Global.phnomber = slov[2];

if(slov[0].IndexOf(maskedTextBox1.Text) == 0)

{

this.listBox1.Items.Insert(0, slov[2]+$" (ФИО-{slov[0]})");

n++;

}

}

Global.sr.Close();

if (n == 1)

{

MessageBox.Show("Cотрудник не найден!");

}

}

}

else

{

MessageBox.Show("Файл не создан!");

}

}

Событие нажатия кнопки Button1 на форме Form7

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

{

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

{

if (maskedTextBox1.Text == "")

{

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

}

else

{

string lin;

char[] separator = new char[] { ',' };

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

Global.sr = new StreamReader(Global.aFile, Encoding.Default);

int n = 1;

while (!Global.sr.EndOfStream)

{

lin = Global.sr.ReadLine();

string[] slov = lin.Split(separator);

Global.name = slov[0];

Global.address = slov[1];

Global.phnomber = slov[2];

if (slov[2].IndexOf(maskedTextBox1.Text) == 0)

{

this.listBox1.Items.Insert(0, slov[0]+$" (номер-{slov[2]})");

n++;

}

}

Global.sr.Close();

if (n==1)

{

MessageBox.Show("Cотрудников с таким номером телефона нет!");

}

}

}

else

{

MessageBox.Show("Файл не создан!");

}

}

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.9) |
| 2 | Повторное создание файла |  | Сообщение “Файл уже существует” (Рис.10) |
| 3 | Добавление записи (успех) | Колосов Вячеслав Германович  Ломоносова 74/1  684512 | Сообщение “Запись добавлена” (Рис.11) |
| 4 | Ошибка при добавлении записи (заполнены не все поля) | Колосов Вячеслав Германович  684512 | Сообщение “Не все поля заполнены” (Рис.12) |
| 5 | Ошибка при добавлении записи (файл не cоздан) |  | Сообщение “Файл не создан” (Рис.13) |
| 6 | Удаление записи (успех) | 2 | Сообщение “Запись удалена” (Рис.14) |
| 7 | Ошибка при удалении записи (не введен номер записи) |  | Сообщение “Не все поля заполнены” (Рис.15) |
| 8 | Ошибка при удалении записи (запись не найдена) | 99 | Сообщение “Нет записи” (Рис.16) |
| 9 | Ошибка при удалении записи (файл не создан) | 3 | Сообщение “Файл не создан” (Рис.17) |
| 10 | Поиск записи для редактирования (успех) | 3 | Колосов Вячеслав Германович  Ломоносова 74/1  684512 |
| 11 | Поиск записи для редактирования (не введен номер записи) |  | Сообщение “Ошибка” (Рис.18) |
| 12 | Поиск записи для редактирования (запись не найдена) | 99 | Сообщение “Нет записи” (Рис.19) |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| **N теста** | **Назначение теста** | **Входные данные для теста** | **Выходные данные для теста** |
| 13 | Поиск записи для редактирования (файл не создан) | 1 | Сообщение “Файл не создан!” (Рис.20) |
| 14 | Редактирование записи (успех) | 3 | Колосов Вячеслав Германович  Ломоносова 74/1  666666  Сообщение “Запись изменена” (Рис.21) |
| 15 | Редактирование записи(не все поля заполнены) | 3 | Колосов Вячеслав Германович  Ломоносова 74/1  Сообщение “ Не все поля заполнены!” (Рис.22) |
| 16 | Просмотр записей (успех) |  | Рис.23 |
| 17 | Просмотр записей (нет записей) |  | Рис.24 |
| 18 | Просмотр записей (нет файла) |  | Сообщение “Файл не создан” (Рис.25) |
| 19 | Запрос 1 (успех) | Колосова | 689045 |
| 20 | Запрос 1 (нет записей) | Иванов | Сообщение “Сотрудник не найден!” (Рис.27) |
| 21 | Запрос 1(пустые поля ввода) |  | Сообщение “Не все поля заполнены” (Рис.28) |
| 22 | Запрос 1 (нет файла) | Иванов | Сообщение “Файл не создан” (Рис.29) |
| 23 | Запрос 2 (успех) | 6 | Колосов ВячеславГерманович  Колосова Елена Валерьевна |
| 24 | Запрос 2 (пустое поле ввода) |  | Сообщение “Не все поля заполнены” (Рис.31) |
| 25 | Запрос 2(Нет файла) |  | Сообщение “Файл не создан”(Рис.32) |
| 26 | Запрос 2 (Нет записей) | 1 | Сообщение “Сотрудники с такими телефонами не найдены ”(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 9.

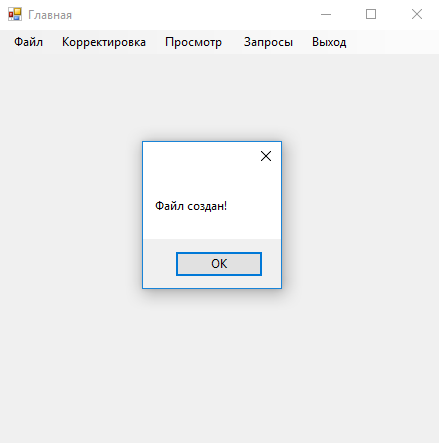


Рисунок 9 – Результат выполнения теста 1

Результат тестирования создания файла, когда файл уже существует (тест 2) показан на рисунке 10

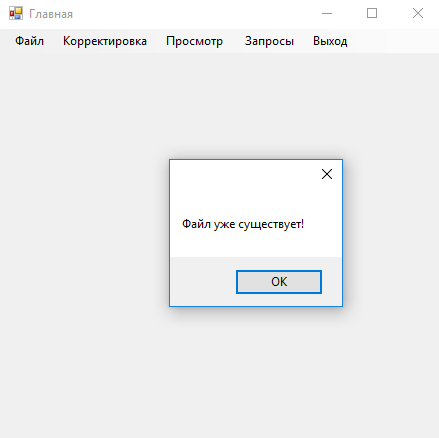


Рисунок 10– Результат выполнения теста 2

Результат тестирования добавления записи(тест 3) показан на рисунке 11

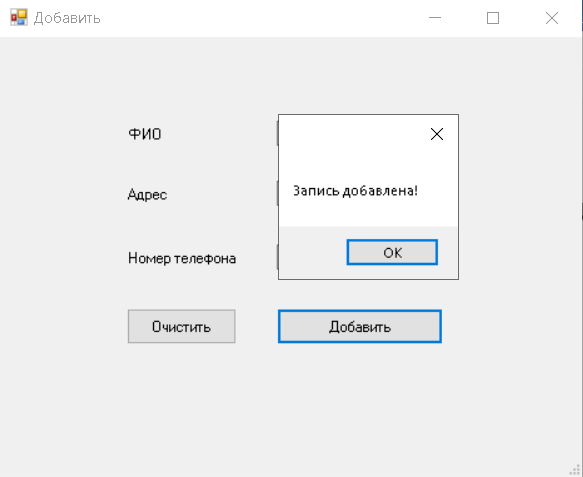


Рисунок 11 – Результат выполнения теста 3

Результат тестирования добавления записи, когда не все поля заполнены(тест 4) показан на рисунке 12

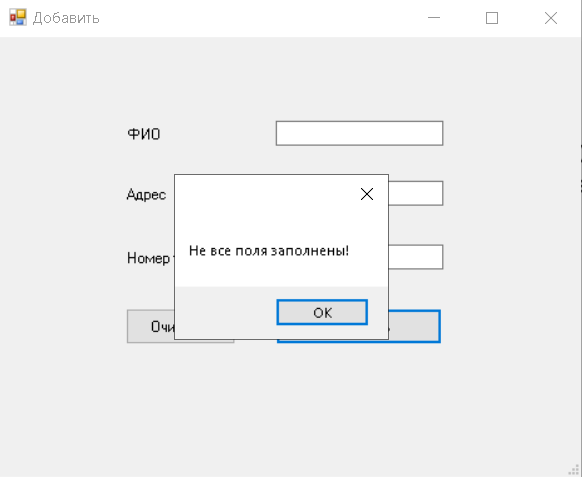


Рисунок 12 – Результат выполнения теста 4

Результат тестирования добавления записи, когда файл не создан(тест 5) показан на рисунке 13

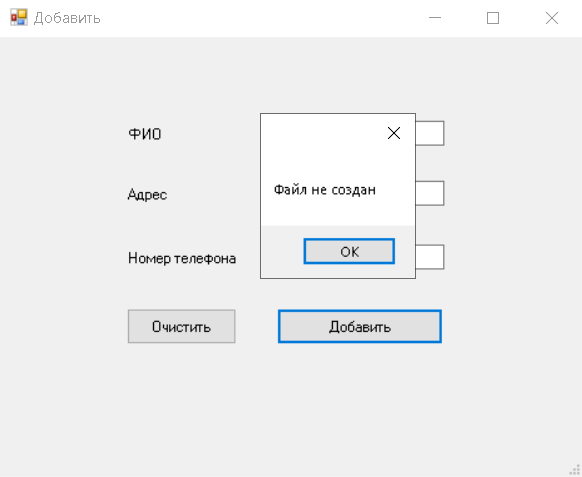


Рисунок 13 – Результат выполнения теста 5

Результат тестирования удаления записи (тест 6) показан на рисунке 14

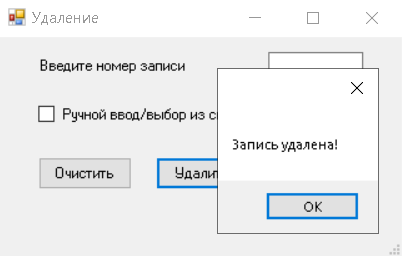


Рисунок 14– Результат выполнения теста 6

Результат тестирования удаления записи, когда не введен номер записи (тест 7) показан на рисунке 15

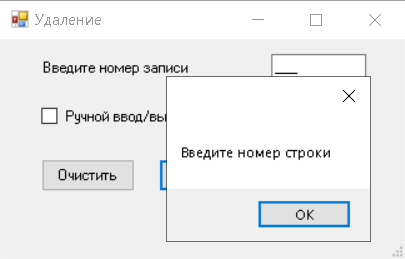


Рисунок 15 – Результат выполнения теста 7

Результат тестирования удаления записи, когда запись с таким номером не существует (тест 8) показан на рисунке 16

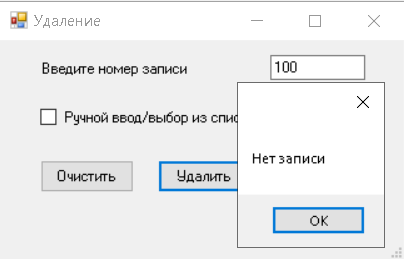


Рисунок 16 – Результат выполнения теста 8

Результат тестирования удаления записи, когда файл не создан(тест 9) показан на рисунке 17

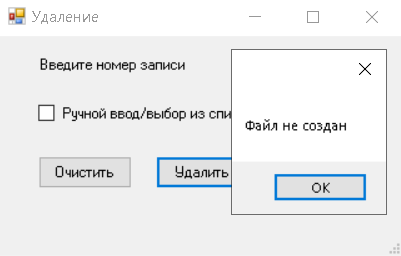


Рисунок 17 – Результат выполнения теста 9

Результат тестирования изменения записи, нажатие на кнопку “Найти” (тест 10) показан на рисунке 18

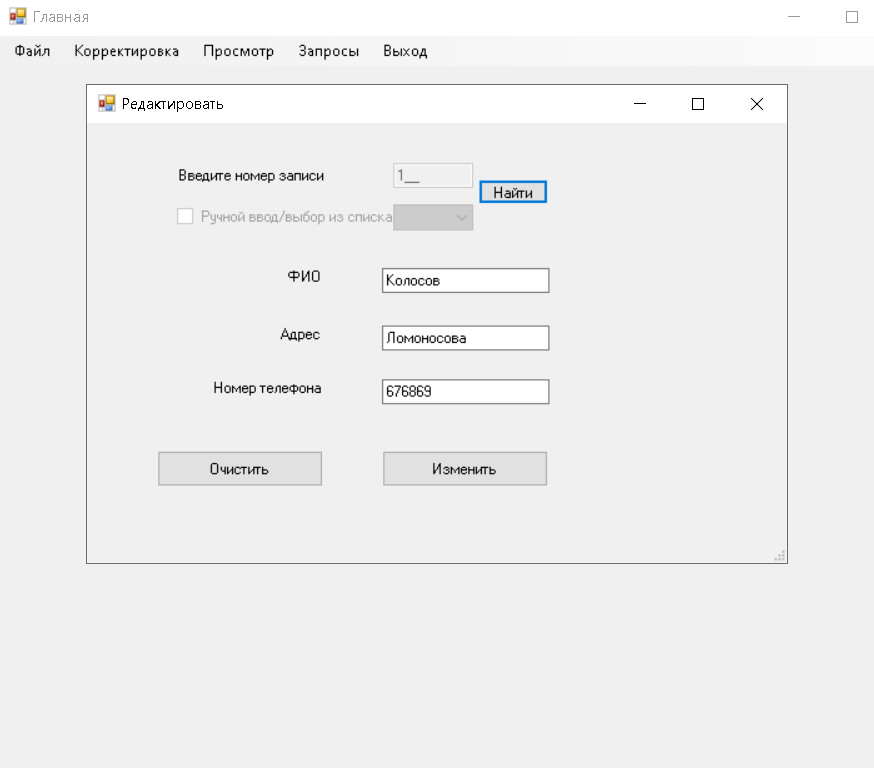


Рисунок 18 – Результат выполнения теста 10

Результат тестирования изменения записи, нажатие на кнопку “Найти” когда не введен номер записи (тест 11) показан на рисунке 19

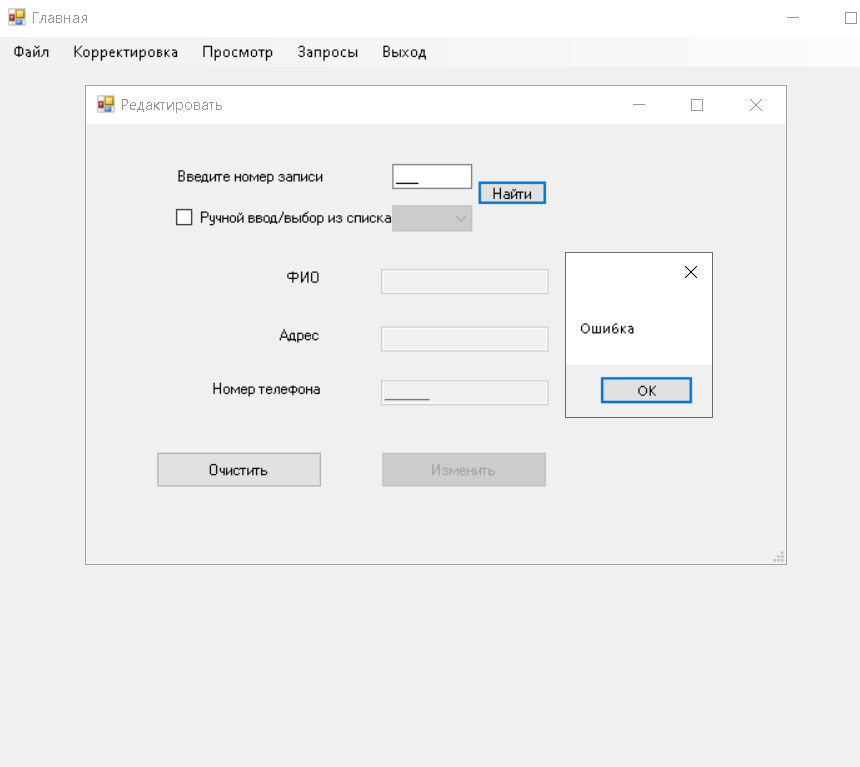


Рисунок 19 – Результат выполнения теста 11

Результат тестирования изменения записи, нажатие на кнопку “Найти” когда запись с таким номером не существует (тест 12) показан на рисунке 20

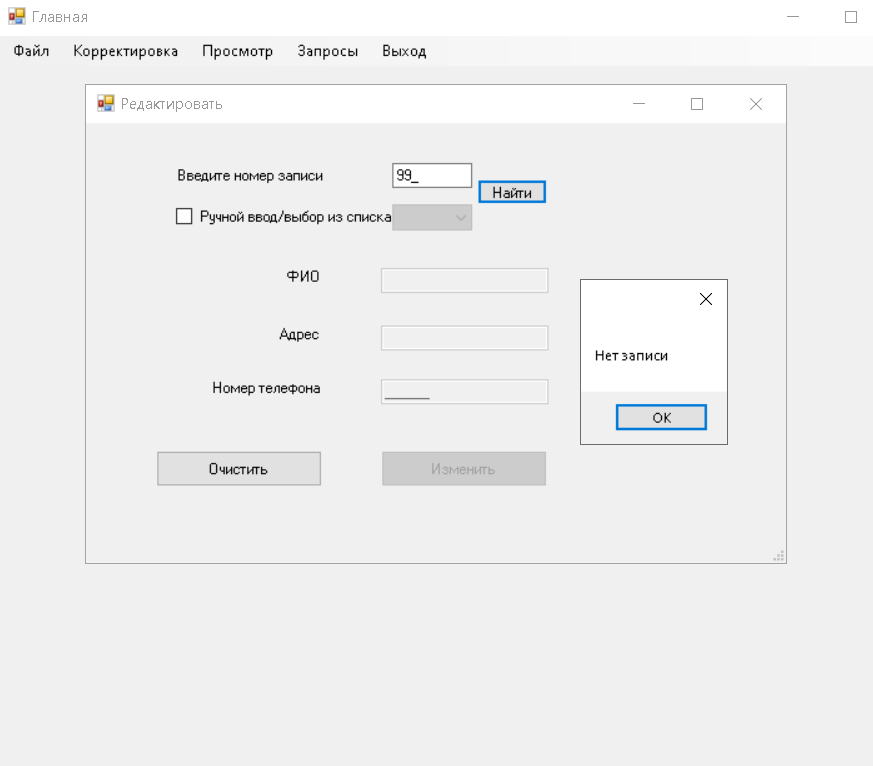


Рисунок 20 – Результат выполнения теста 12

Результат тестирования изменения записи, нажатие на кнопку “Найти” когда файл не создан (тест 13) показан на рисунке 21

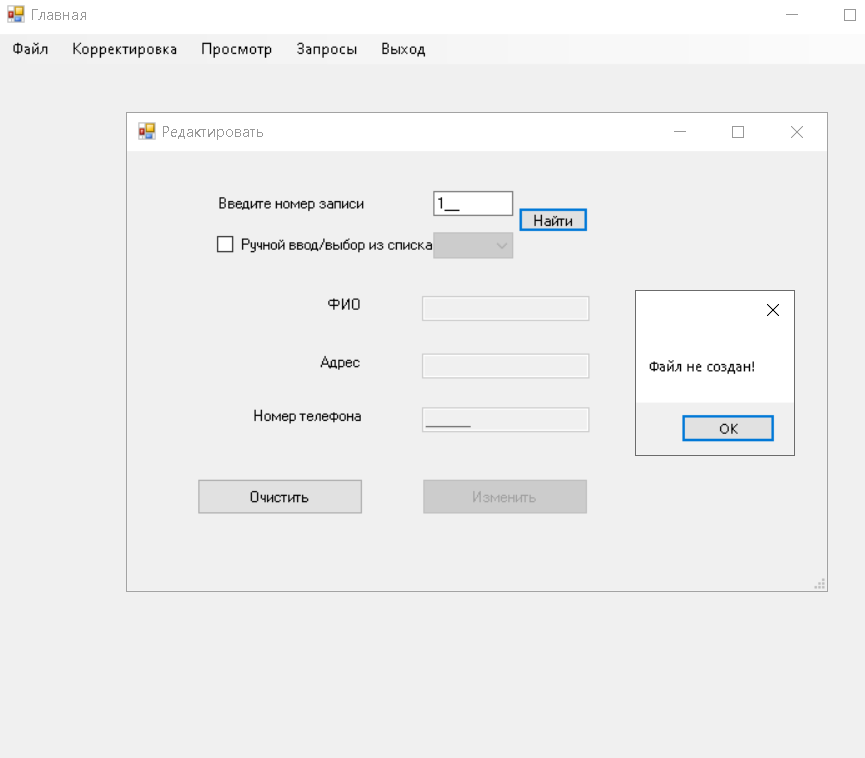


Рисунок 21 – Результат выполнения теста 13

Результат тестирования изменения записи, нажатие на кнопку “Изменить”(тест 14) показан на рисунке 22

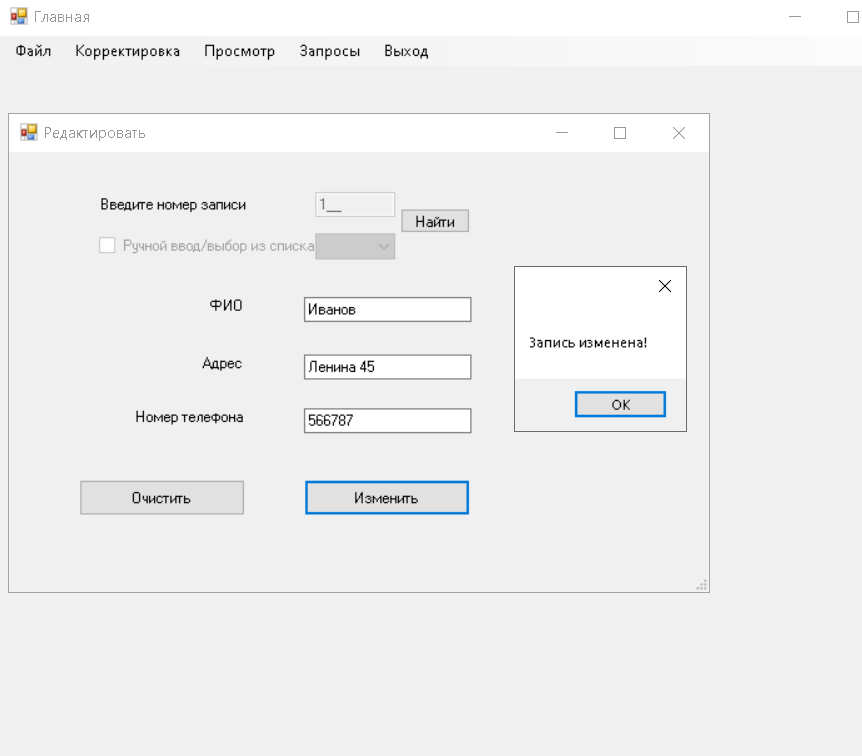


Рисунок 22 – Результат выполнения теста 14

Результат тестирования изменения записи, нажатие на кнопку “Изменить”, когда не все поля заполнены (тест 15) показан на рисунке 23

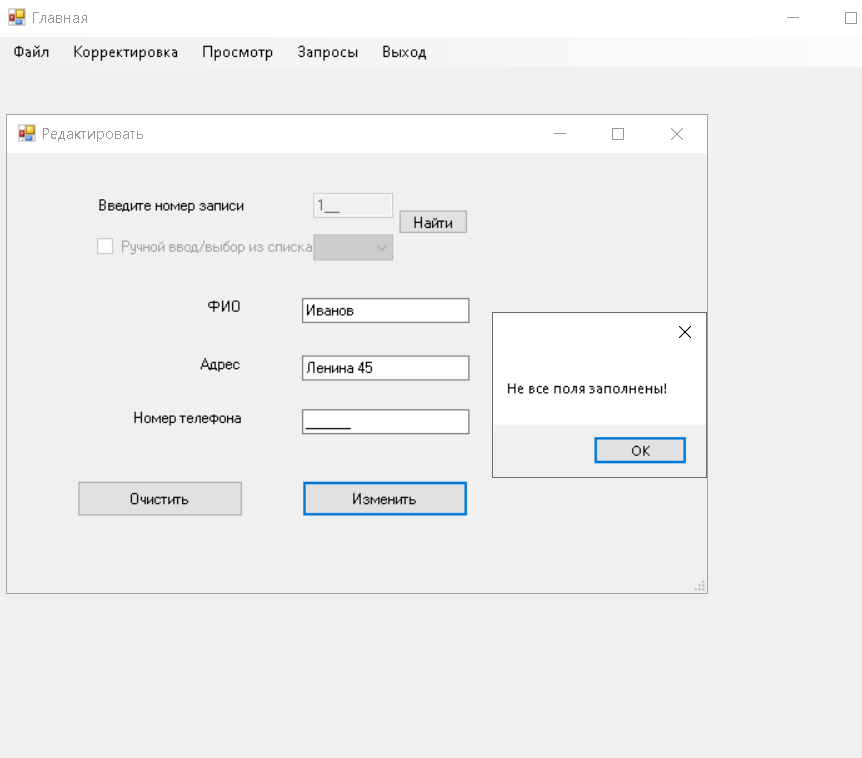


Рисунок 23 – Результат выполнения теста 15

Результат тестирования формы просмотра записей, (тест 16) показан на рисунке 24

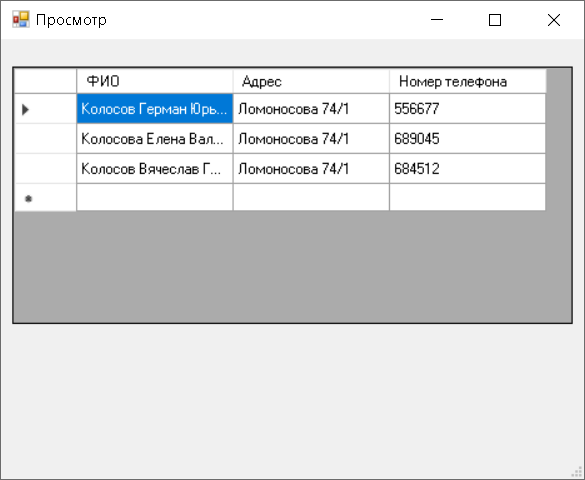


Рисунок 24 – Результат выполнения теста 16

Результат тестирования формы просмотра записей, когда в файле нет записей(тест 17) показан на рисунке 25

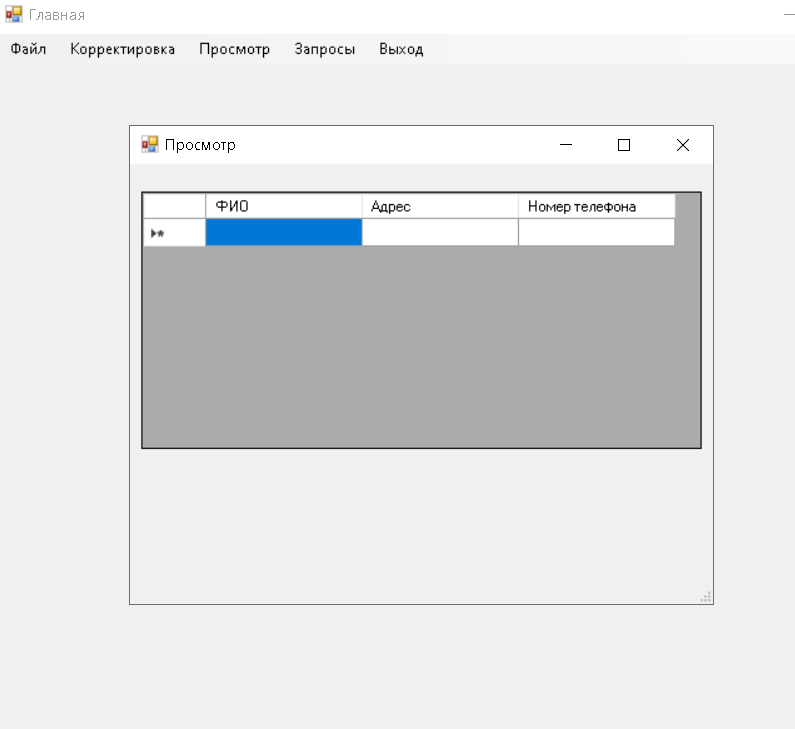


Рисунок 25 – Результат выполнения теста 17

Результат тестирования формы просмотра записей, когда файл не создан (тест 18) показан на рисунке 26

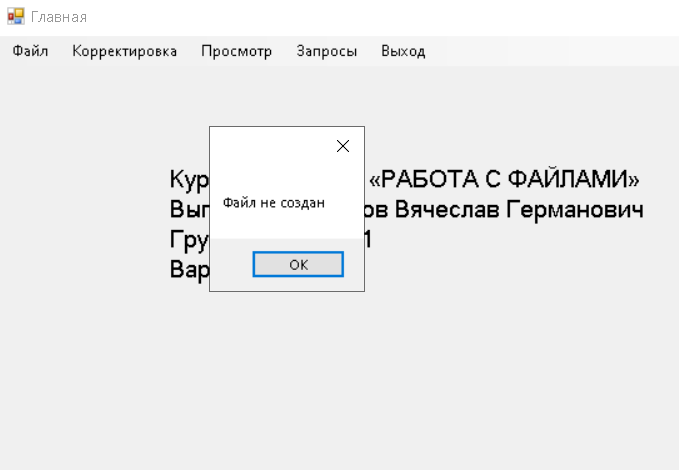


Рисунок 26 – Результат выполнения теста 18

Результат тестирования запроса 1, (тест 19) показан на рисунке 27

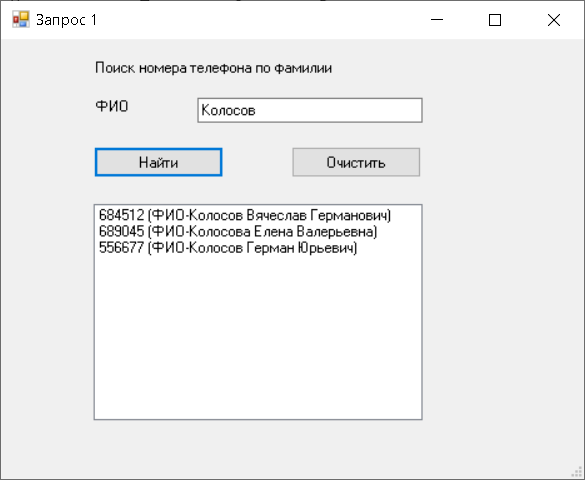


Рисунок 27 – Результат выполнения теста 19

Результат тестирования запроса 1, когда не найдено записей по запросу (тест 20) показан на рисунке 28

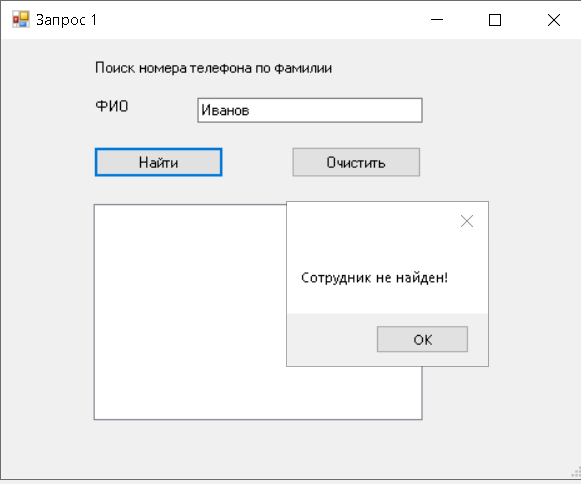


Рисунок 28 – Результат выполнения теста 20

Результат тестирования запроса 1, когда не введен ФИО сотрудника (тест 21) показан на рисунке 29

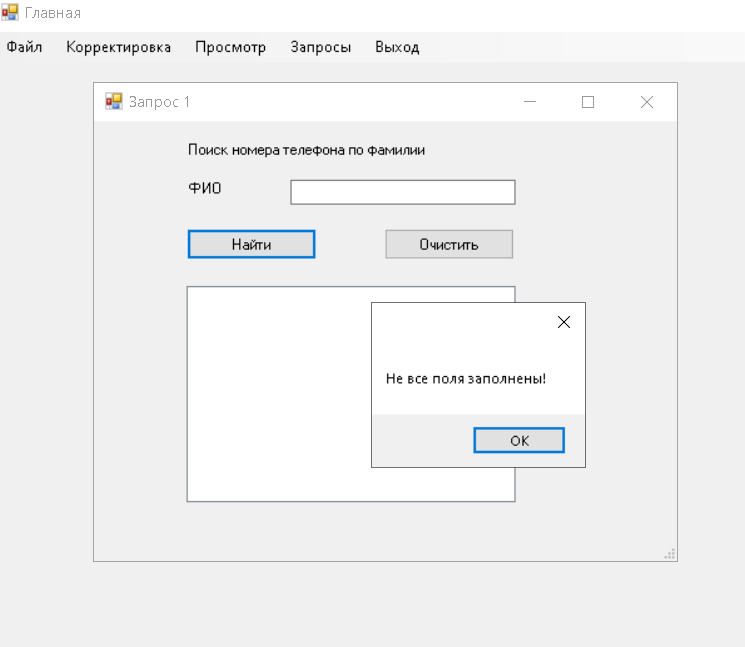


Рисунок 29 – Результат выполнения теста 21

Результат тестирования запроса 1, когда файл не создан (тест 22) показан на рисунке 30

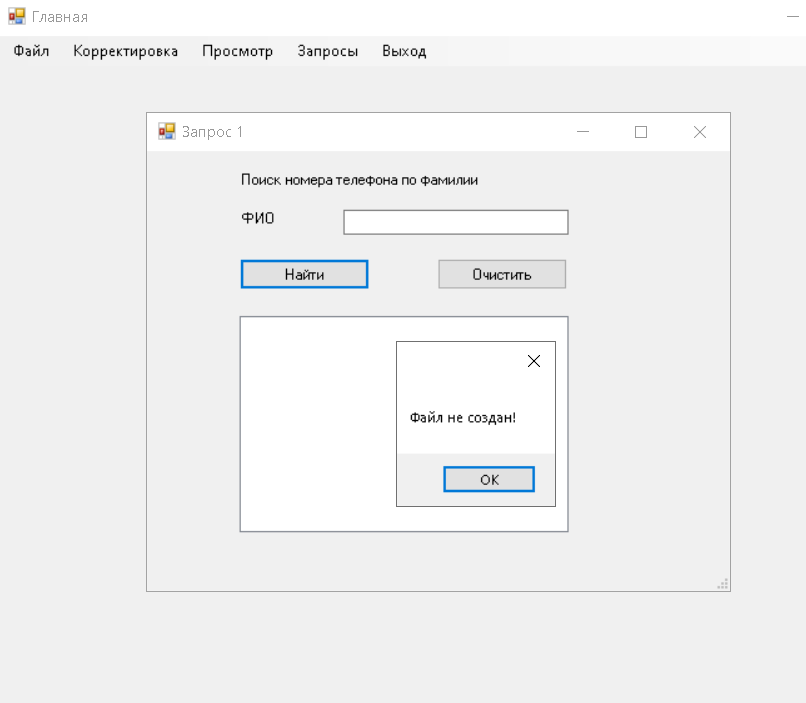


Рисунок 30 – Результат выполнения теста 22

Результат тестирования запроса 2 (тест 23) показан на рисунке 31

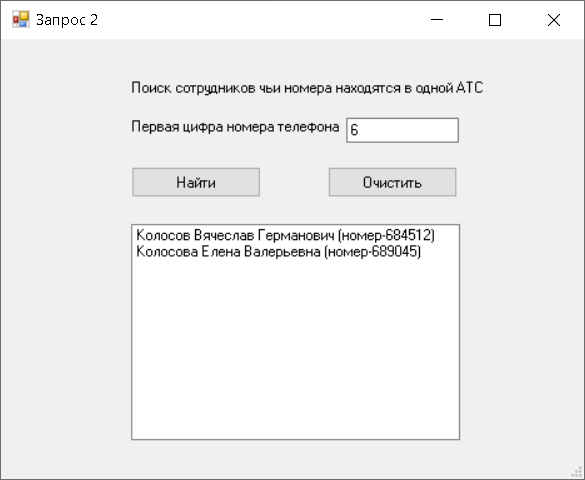


Рисунок 31 – Результат выполнения теста 23

Результат тестирования запроса 2, когда не введена 1 цифра номера (тест 24) показан на рисунке 32

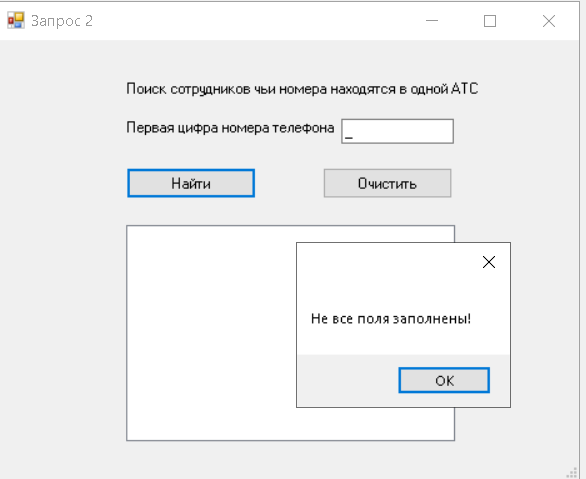


Рисунок 32 – Результат выполнения теста 24

Результат тестирования запроса 2, когда файл не создан (тест 25) показан на рисунке 33

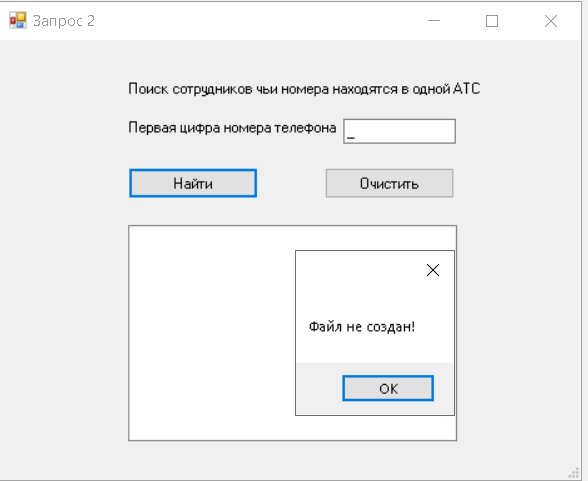


Рисунок 33 – Результат выполнения теста 25

Результат тестирования запроса 2, когда нет сотрудников с таким номером телефона (тест 26) показан на рисунке 34

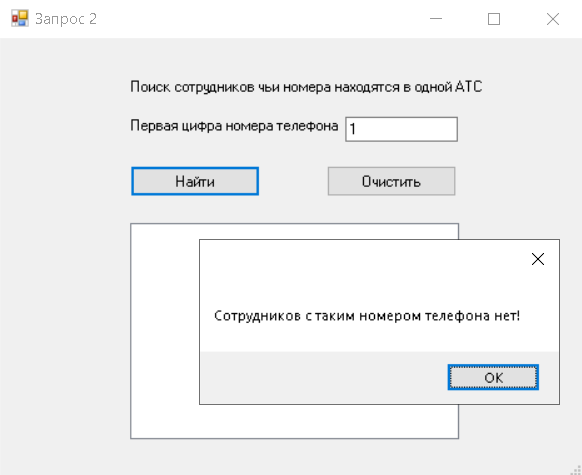


Рисунок 34 – Результат выполнения теста 26

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

Заключение

В результате выполнения курсового проекта, мною было разработано Windows приложение, которое можно использовать даже в реальных условиях. Например для создания не большой базы данных с информацией о сотрудниках фирмы: ФИО , адрес , номер телефона. В программе можно редактировать и удалять записи из файла, также выполнять два запроса. Приложение было тщательно оттестировано, сбоев в работе не обнаружено.

В ходе выполнения, я научился разрабатывать Windows приложение с большим количеством окон, уяснил как нужно связывать эти окна между собой. Научился работать с глобальными переменными и структурами. Усвоил как можно работать с файлами в программе. Также научился тестировать приложение на наличие ошибок.

Приобретенные мною знания пригодятся мне в моем будущей профессиональной деятельности.

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.