Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа № 14
по дисциплине «Современные технологии программирования»

Выполнил:
студент группы ИП-712
Алексеев Степан
Владимирович
ФИО студента

Работу проверил:
ассистент кафедры Агалаков А.А.
ФИО преподавателя

Новосибирск 2020 г.

Оглавление

ЗАДАНИЕ 2
ТЕСТОВЫЕ НАБОРЫ ДАННЫХ 3
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ 4
ВЫВОД 6
ПРИЛОЖЕНИЕ 7
Листинг 1. Form1.cs 7
Листинг 2. Form2.cs 11
Листинг 3. Form1Tsets.cs 12

ЗАДАНИЕ

Реализовать приложение «Телефонная книга» работающее в ОС Windows.

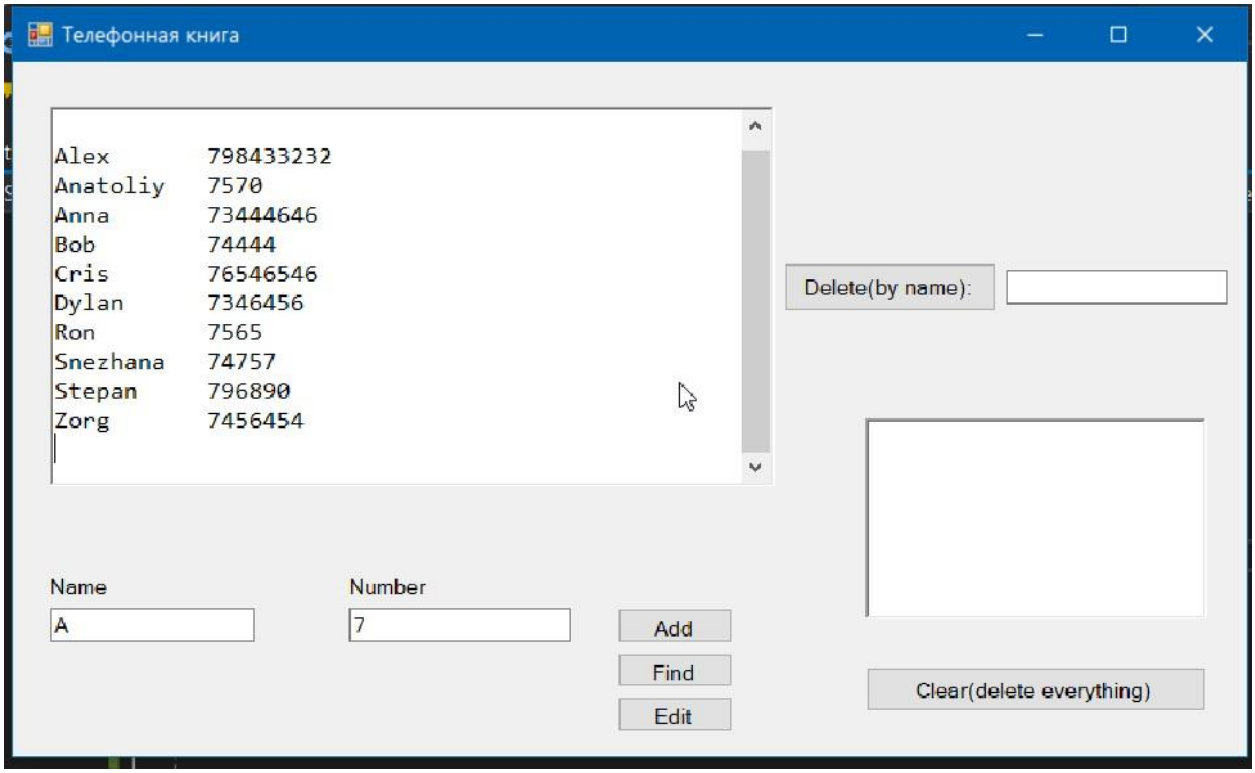
Приложение должно обеспечивать пользователю:

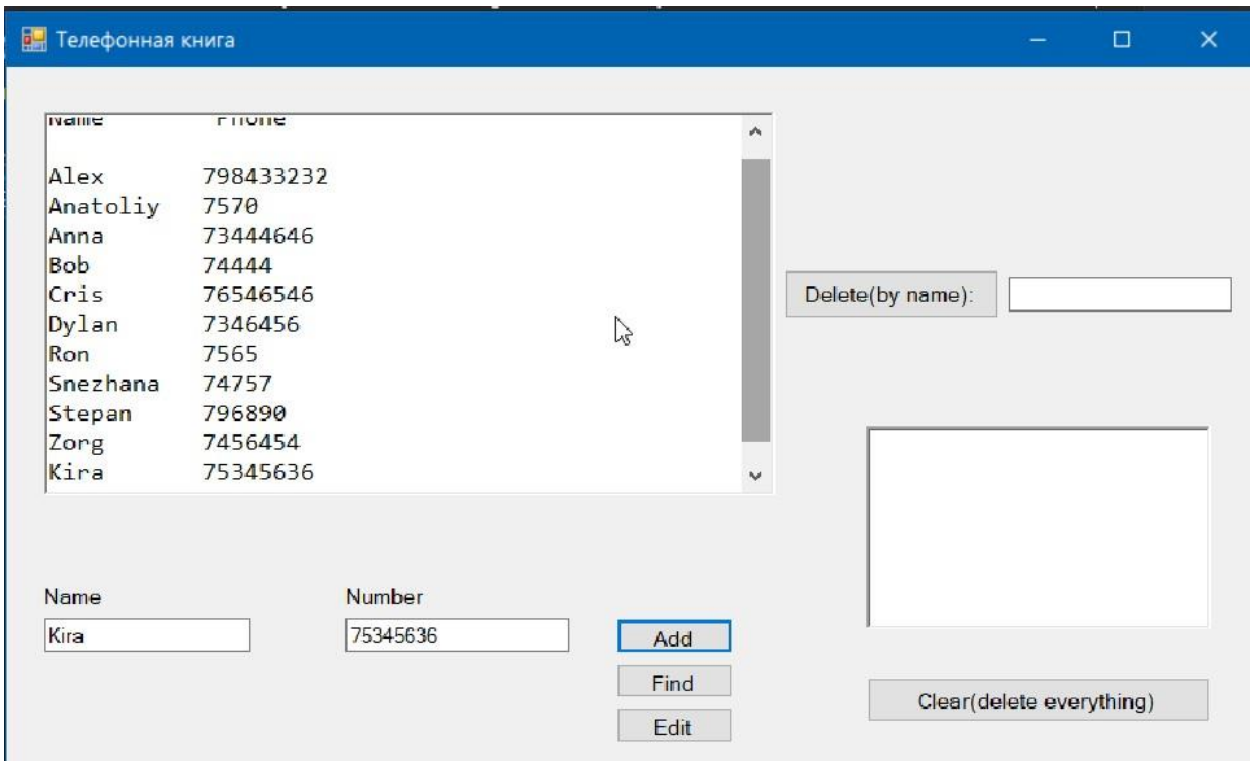
- ввод, редактирование и сохранение имён абонентов городской телефонной сети и номеров их телефонов
- записи должны храниться и отображаться в отсортированном по именам порядке;
- поиск по имени;
- удаление записи;
- очистку книги.

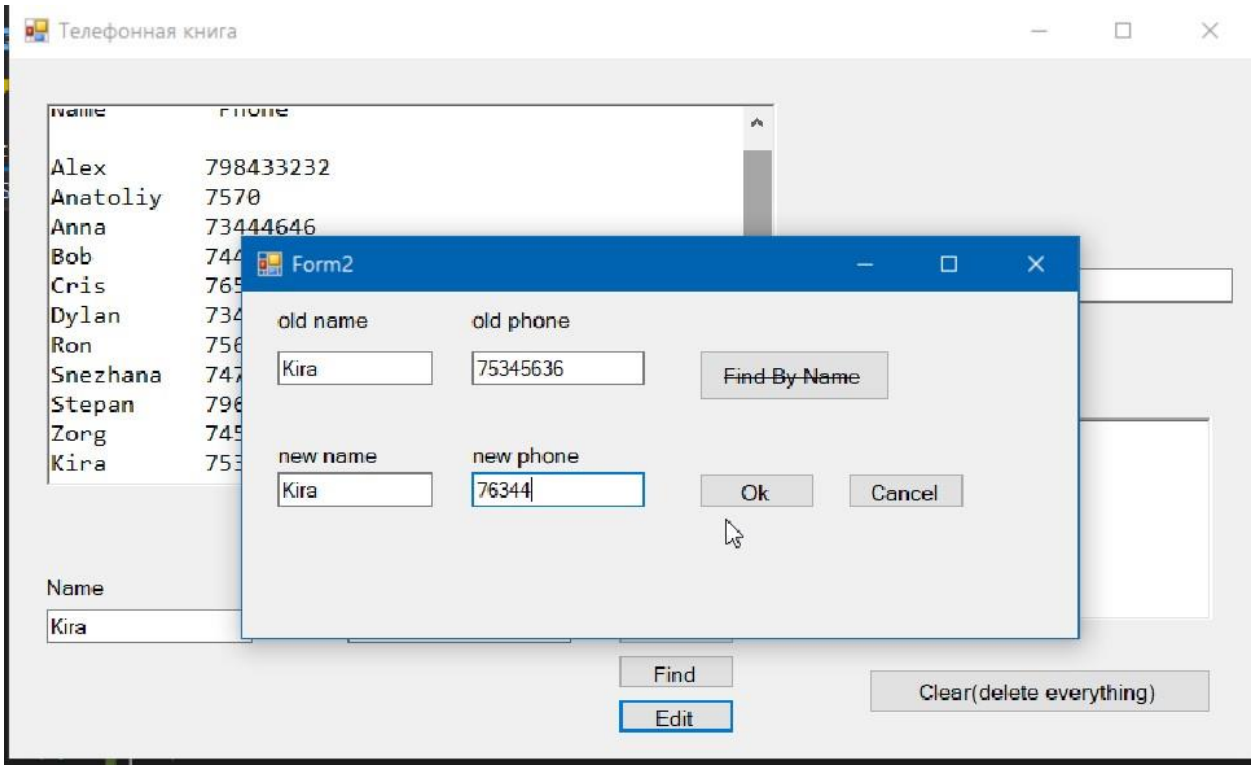
ТЕСТОВЫЕ НАБОРЫ ДАННЫХ

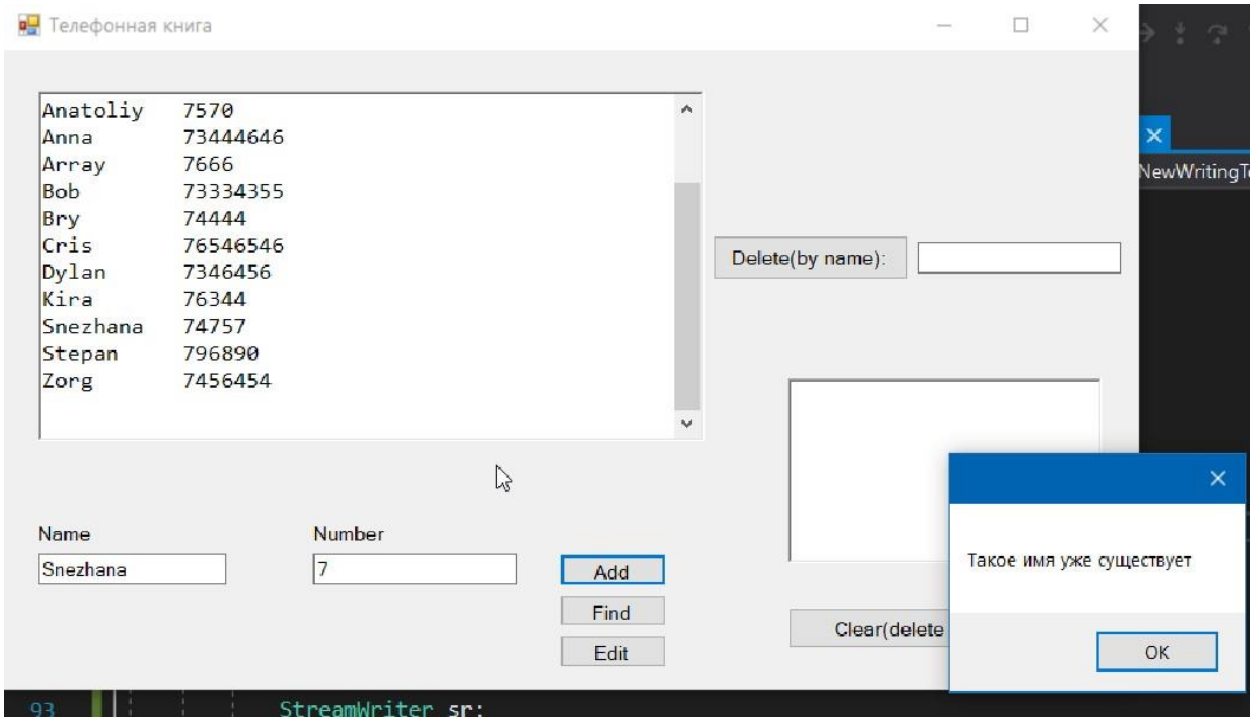
```
Form1 f1 = new Form1(path);
        f1.ReadFromAFileAndWriteTo_dict(path);
        Assert.IsTrue(f1.dict.ContainsKey("Bob"));
```

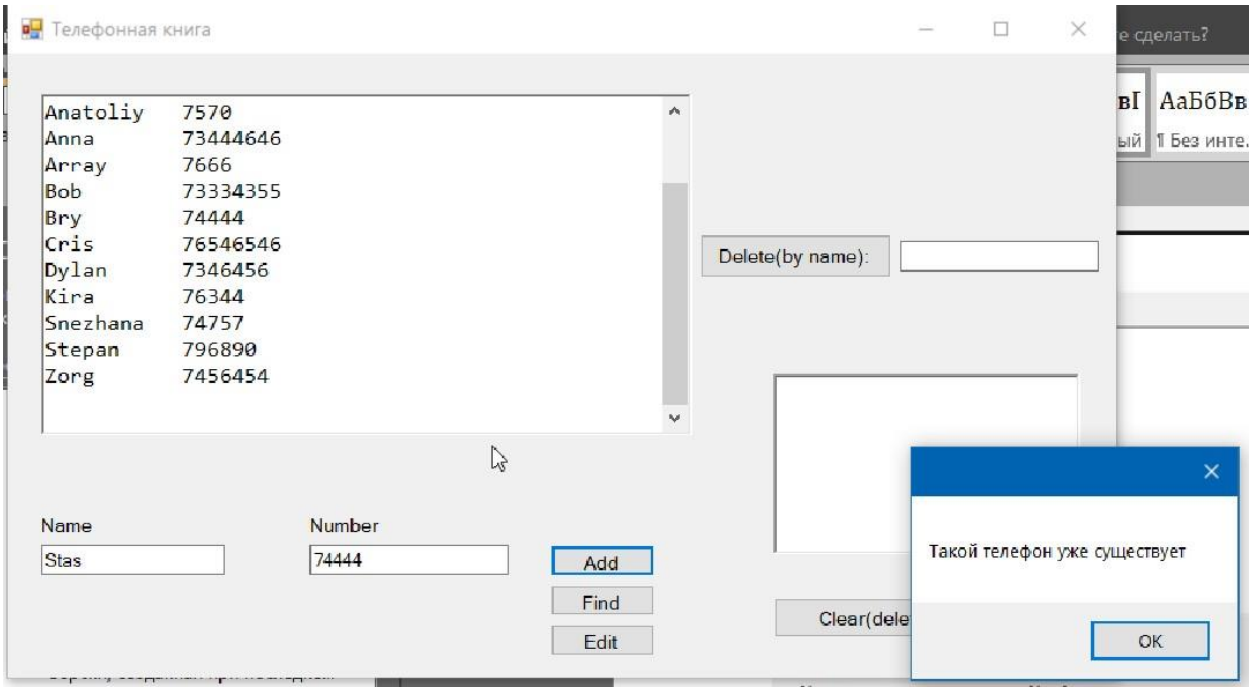
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ






Телефонная книга
Alex 798433232
Anatoliy 7570
Anna 73444646
Bob 744
Cris 765
Dylan 734
Ron 756
Snezhana 747
Stepan 796
Zorg 745
Kira 753
Form2
old name
old phone
Kira
75345636
Find By Name
new name
new phone
Kira
76344
Ok
Cancel
Name
Kira
Find
Edit
Clear(delete everything)


Телефонная книга
Anatoliy 7570
Anna 73444646
Array 7666
Bob 73334355
Bry 74444
Cris 76546546
Dylan 7346456
Kira 76344
Snezhana 74757
Stepan 796890
Zorg 7456454
Delete(by name):
Name
Snezhana
Number
7
Add
Find
Edit
Clear(delete
Такое имя уже существует
OK
StreamWriter sr;



ВЫВОД

Научился создавать оконные приложения , вызывать новые окна из текущего, связывать объекты разных классов в оконном приложении. Реализовал телефонную книгу полностью обеспечивающую работу требуемого функционала. Протестировал все функции. Пришёл к выводу о необходимости создания более специализированных методов, т.к. если прописать в одном методе слишком много функционала и он начнёт зависеть от других методов, то в случае провала теста придётся проверять все те методы, от которых зависит текущий.

ПРИЛОЖЕНИЕ

Листинг 1. Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace STP_14_PhoneBook
{
    public partial class Form1 : Form
    {
        public Dictionary<string, long> dict;
        public string[] stringsToSplit = { "n/", "t/", " ", "  "
};// Добавить проверку на уникальность номера
        public string path = "";
        public Form1(string path)
        {
            this.path = path;
            InitializeComponent();
            ReadFromAFileAndWriteTo_dict(path);//прочитал в dict
            Sort_dictAndWriteToFileFrom_dict(path);//сортировку
запускаю только при запуске
            printFRom_dictToRichTextBox();//распечатал в ртб.
Остальное по кнопкам
            textBox1.Text = "A";
            textBox2.Text = "7";
        }

        public async void ReadFromAFileAndWriteTo_dict(string
path)         //инициализирует начальное состояние
        {
            using (StreamReader sr = new StreamReader(path,
System.Text.Encoding.Default))
                try
                {
                    dict = new Dictionary<string, long>();
                    string line;
                    while ((line = sr.ReadLine()) != null)
//read from a stream(a file)
                    {
                        if (line != "\n" && line != "\t" && line
!= "\0" && line != "")
                        {
```

```
                            string a =
line.Split(stringsToSplit, 2,
StringSplitOptions.RemoveEmptyEntries)[0];
                            string b =
line.Split(stringsToSplit, 2,
StringSplitOptions.RemoveEmptyEntries)[1];
                            long numB = long.Parse(b);
                            dict.Add(a, numB);
//write to dict
                        }
                    }
                    sr.Close();
                }
                catch (Exception e)
                {

                }
        }
        public async void Sort_dictAndWriteToFileFrom_dict(
string path)
        {
            var myList = from entry in dict orderby entry.Key
ascending select entry;
            using (StreamWriter sr = new StreamWriter(path))
            {
                foreach (var item in myList)
                {
                    sr.WriteLine(item.Key + " " + item.Value +
"\n");
                }
                sr.Close();
                //dict = (Dictionary)myList;
            }
        }
        public void printFRom_dictToRichTextBox()
        {
            richTextBox1.AppendText(String.Format("{0, -10} {1,
-10}\n\n", "Name ", " Phone "));
            foreach (var item in dict)
            {
                //richTextBox1.AppendText(String.Format("Name: "
+ item.Key + " phone: " + item.Value + "\n"));
                richTextBox1.AppendText(String.Format("{0, -10}
{1, -10}\n", item.Key, item.Value));
            }

        }
        public void
forButton6AddANewWritingTo_dictAndToRichtexboxAndToFile()
        {
            string a = textBox1.Text;
```

```
            string b = textBox2.Text;
            long longB = long.Parse(b);
            if (dict.ContainsKey(a))
            {
                MessageBox.Show("Такое имя уже существует");
                return;
            }
            else if (dict.ContainsValue(longB))
            {
                MessageBox.Show("Такой телефон уже существует");
                return;
            }

            StreamWriter sr;
            using (sr = new StreamWriter(path, true)) ;//true -
to approve appending
            try
            {
                dict.Add(a, longB);
                richTextBox1.AppendText(String.Format("{0, -10}
{1, -10}\n", a, b));
                sr = new StreamWriter(path, true);
                sr.WriteLine("\n" + a + " " + b);
                MessageBox.Show("no exceptions during saving");
            }
            catch (Exception ee)
            {
                richTextBox2.AppendText(ee.ToString());

                MessageBox.Show("Problem saving occured: \n" +
ee.ToString());
            }
            sr.Close();
            clearRTB();
            Sort_dictAndWriteToFileFrom_dict(path);
            printFRom_dictToRichTextBox();
        }
        private void
button6AddANewWritingTo_dictAndToRichtexboxAndToFile(object
sender, EventArgs e)
        {

forButton6AddANewWritingTo_dictAndToRichtexboxAndToFile();
        }
        public void clearRTB()
        {
            richTextBox1.Clear();
        }
        public void forButton1ClearFileAnd_dictAndRtb()
        {
            dict = null;
```

```
            richTextBox1.Clear();
            StreamWriter sr;// = new StreamWriter(path); ;
            try
            {
                sr = new StreamWriter(path);
                sr.WriteLine("");
                MessageBox.Show("no exceptions during saving");
                sr.Close();
            }
            catch (Exception ee)
            {
                richTextBox2.AppendText(ee.ToString());
                MessageBox.Show("Problem saving occured: \n" +
ee.ToString());
            }

        }
        private void button1ClearFileAnd_dictAndRtb(object
sender, EventArgs e)
        {
            forButton1ClearFileAnd_dictAndRtb();
        }


        public void startingTestingInitializationOfdict()
        {
            dict = new Dictionary<string, long>();
            dict.Add("Stepan", 79133895118);
            dict.Add("Alena", 79234512938);
        }

        public void forButton3Delete()
        {
            string nameToDelete = textBox3.Text;
            dict.Remove(nameToDelete);
            richTextBox1.Clear();
            Sort_dictAndWriteToFileFrom_dict(path);
            printFRom_dictToRichTextBox();
        }
        private void button3Delete(object sender, EventArgs e)
        {
            forButton3Delete();
        }

        public void forButton7Find()
        {
            long phone;
            string nameToFind = textBox1.Text;
            dict.TryGetValue(nameToFind, out phone);
            textBox2.Text = phone.ToString();
        }
```

```
        private void button7Find(object sender, EventArgs e)
        {
            forButton7Find();
        }
        public void forButton4Edit()
        {
            string nameToEdit = textBox1.Text;
            string phoneToedit = textBox2.Text;
            long phoneLong = long.Parse(phoneToedit);

            Form2 f2 = new Form2(ref dict, textBox1.Text,
textBox2.Text, this, path);
            f2.Show();
        }
        private void button4Edit(object sender, EventArgs e)
        {
            forButton4Edit();
        }
    }
}
```

Листинг 2. Form2.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace STP_14_PhoneBook
{
    public partial class Form2 : Form
    {
        Dictionary<string, long> dict;
        string name;
        string phone;
        Form1 f1;
        string path;
        public Form2(ref Dictionary<string, long> dict, string
name, string phone, Form1 f1, string path)
        {
            this.dict = dict;
            this.name = name;
            this.phone = phone;
            this.f1 = f1;
            this.path = path;
```

```
            InitializeComponent();
            textBox1.Text = name;
            textBox2.Text = phone;
            textBox3.Text = name;
            textBox4.Text = phone;
        }

        private void button4_Ok(object sender, EventArgs e)
        {
            long newPhone = long.Parse(textBox4.Text);
            string newName = textBox3.Text;
            if (dict.ContainsKey(newName) )
            {
                MessageBox.Show("Такое имя уже существует");
                return;
            }
            else if (dict.ContainsValue(newPhone))
            {
                MessageBox.Show("Такой телефон уже существует");
                return;
            }
            dict.Remove(name);
            dict.Add(newName, newPhone);//пока только записал в
dict
            f1.Sort_dictAndWriteToFileFrom_dict(path);
            f1.clearRTB();
            f1.printFRom_dictToRichTextBox();
            this.Close();
        }

        private void button2_Find(object sender, EventArgs e)
        {

        }

        private void ButtonCancel(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Листинг 3. Form1Tsets.cs

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using STP_14_PhoneBook;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.CompilerServices;
```

```
using System.Text;
using System.Threading.Tasks;

namespace STP_14_PhoneBook.Tests
{
    [TestClass()]
    public class Form1Tests
    {
        public Dictionary<string, long> dict;
        public string[] stringsToSplit = { "n/", "t/", " ", "  "
};
        string path =
"C:/Users/stepa/repos2/STP_14_PhoneBook/STP_14_PhoneBook/bookTes
t.txt";
        public string path6 =
"C:/Users/stepa/repos2/STP_14_PhoneBook/STP_14_PhoneBook/bookBut
ton6.txt";
        public string path7 =
"C:/Users/stepa/repos2/STP_14_PhoneBook/STP_14_PhoneBook/book7.t
xt";
        public string path7_ =
"C:/Users/stepa/repos2/STP_14_PhoneBook/STP_14_PhoneBook/book7_.
txt";

        [TestMethod()]
        public void ReadFromAFileAndWriteTo_dictTest()
        {
            Form1 f1 = new Form1(path);
            f1.ReadFromAFileAndWriteTo_dict(path);
            Assert.IsTrue(f1.dict.ContainsKey("Bob"));
        }

        [TestMethod()]
        public void Sort_dictAndWriteToFileFrom_dictTest()
        {
            string pathIn =
"C:/Users/stepa/repos2/STP_14_PhoneBook/STP_14_PhoneBook/bookTes
tToSortIn.txt";
            string pathOut =
"C:/Users/stepa/repos2/STP_14_PhoneBook/STP_14_PhoneBook/bookTes
tToSortOut.txt";
            Form1 f1 = new Form1(path);
            f1.ReadFromAFileAndWriteTo_dict(pathIn);
            f1.Sort_dictAndWriteToFileFrom_dict(pathOut);
            string line = "";
            using (StreamReader sr = new StreamReader(pathOut,
System.Text.Encoding.Default))
                try
                {//читаю первую строку и удостоверяюсь, что
наименьшее имя оказалось первым
```

```
                    line = sr.ReadLine().Split(stringsToSplit,
2, StringSplitOptions.RemoveEmptyEntries)[0];
                    sr.Close();
                }
                catch (Exception e) { }
            File.Delete(pathOut);
            Assert.AreEqual(line, "Alex");
        }

        [TestMethod()]
        public void
ForButton6AddANewWritingTo_dictAndToRichtexboxAndToFileTest()
        {
            Form1 f1 = new Form1(path6);
            f1.textBox1.Text = "Carla";
            f1.textBox2.Text = "79563434";

f1.forButton6AddANewWritingTo_dictAndToRichtexboxAndToFile();
            Assert.IsTrue(f1.dict.ContainsKey("Carla"));
            f1.textBox3.Text = "Carla";
            f1.forButton3Delete();
        }

        [TestMethod()]
        public void forButton1ClearFileAnd_dictAndRtbTest()
        {
            Form1 f1 = new Form1(path7);
            f1.forButton1ClearFileAnd_dictAndRtb();
            Assert.IsNull(f1.dict);
            f1.Close();
            File.Copy(path7_, path7, true);//true разрешает
перезаписать существующий файл
        }

        [TestMethod()]
        public void forButton3DeleteTest()
        {
            Form1 f1 = new Form1(path7);
            f1.textBox3.Text = "Bob";
            f1.forButton3Delete();
            Assert.IsFalse(f1.dict.ContainsKey("Bob"));
            f1.dict.Add("Bob", 1847834);
        }

        [TestMethod()]
        public void forButton7FindTest()
        {
            Form1 f1 = new Form1(path7);
            f1.textBox1.Text = "Cris";
            f1.forButton7Find();
            Assert.AreEqual(f1.textBox2.Text, "76546546");
```

```
            }
        }
    }
```