Министерство образования и науки РФ

Государственное образовательное учреждение

Высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

Кафедра «Физика и прикладная математика» (ВлГУ)

Курсовая работа

По дисциплине «Прикладное программирование»

«Программный модуль учета клиентов туристического агентства.»

Выполнил:

Ст. ПКсп-114

Приняла: Ст. преподаватель  
 Шишкина М. В.

Владимир 2017 г.

**Оглавление**

[**Введение.** 3](#_Toc447867163)

[**Этапы проектирования программы.** 4](#_Toc447867164)

[**1.Проектирование функционала.** 4](#_Toc447867165)

[**2.Проектирование интерфейса.** 5](#_Toc447867166)

[**Функционал программы.** 6](#_Toc447867167)

[**1.Функция добавления клиента.** 6](#_Toc447867168)

[**2.Функция удаления клиента.** 6](#_Toc447867169)

[**3.Функция просмотра информации о клиенте.** 7](#_Toc447867170)

[**4.Функция поиска клиента в БД.** 8](#_Toc447867171)

[**5.Функция сбросов параметров поиска.** 8](#_Toc447867172)

[**6. Функция добавления рейса клиенту.** 9](#_Toc447867173)

[**Схема работы программы.** 10](#_Toc447867174)

[**Этап отладки и завершающих работ.** 10](#_Toc447867175)

[**Заключение.** 10](#_Toc447867176)

[**Приложение:** 12](#_Toc447867177)

[**Список использованных информационных источников:** 21](#_Toc447867178)

# **Введение.**

Компьютеры уже давно и прочно вошли в нашу жизнь. Они помогают решать задачи, на которые раньше приходилось тратить большие объемы времени и человеческих сил. Они развлекают и работают за нас. Именно с выполнением работы компьютером за человека связана моя курсовая работа.

Одна из самых утомляющих работ – ручное заполнение тысяч одинаковых форм, содержащих, к примеру, информацию о клиентах туристического агентства. Во всех крупных компаниях уже давно перешли с бумажного учета своих клиентов на электронный.

В данной работе я рассмотрю все этапы создания программы для учета клиентов - Программного модуля учета клиентов туристического агентства.

# **Этапы проектирования программы.**

## **1.Проектирование функционала.**

Итак, первым делом необходимо определиться с функционалом, которым будет обладать модуль. Во-первых, мы должны иметь возможность добавлять и удалять клиентов из базы данных. Во-вторых, мы должны иметь возможность редактировать данные о клиентах. В-третьих, должна быть возможность добавлять рейсы в историю клиентов и просматривать эту историю. И в-четвертых, должен быть реализован поиск клиентов по ряду признаков. Информация о клиенте должна содержать: Ф.И.О, номер паспорта и пол, а также информацию о всех рейсах совершенных клиентом.

Поиск должен осуществляться по любому из полей информации о клиенте.

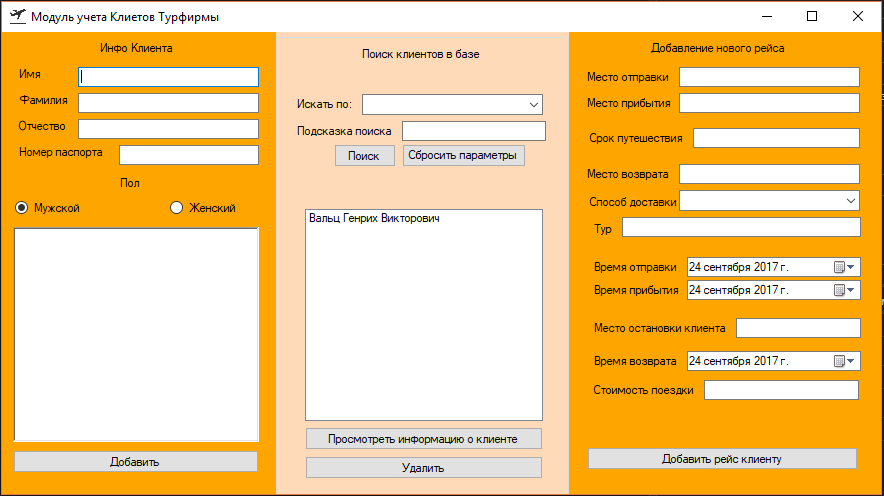
Секция добавления рейса должна иметь следующие поля: Дата и место отправки, прибытия и возвращения, срок путешествия (в днях), способ путешествия (тип используемого транспорта), тур, место остановки клиента и стоимость путешествия.

Таким образом, было определенно, что программа будет выполнять 6 функций, реализованных в обработчиках событий при нажатии на соответствующие кнопки: добавление клиента, просмотр информации о клиенте, удаление клиента из БД, добавление рейса клиенту, поиск клиента в БД и сброса параметров поиска.

Полностью разобравшись с необходимым функционалом приступаем к проектированию интерфейса модуля.

## **2.Проектирование интерфейса.**

Проектирование интерфейса – не самая простая задача. Интерфейс должен быть интуитивен и понятен не только программисту, создавшему его, но и конечному пользователю, для которого собственно проект и разрабатывается. В ходе разработки интерфейс несколько раз менялся, пока не принял, на мой взгляд, достаточно приятный и понятный для стороннего пользователя вид:



*Рис. 1. Интерфейс программы*

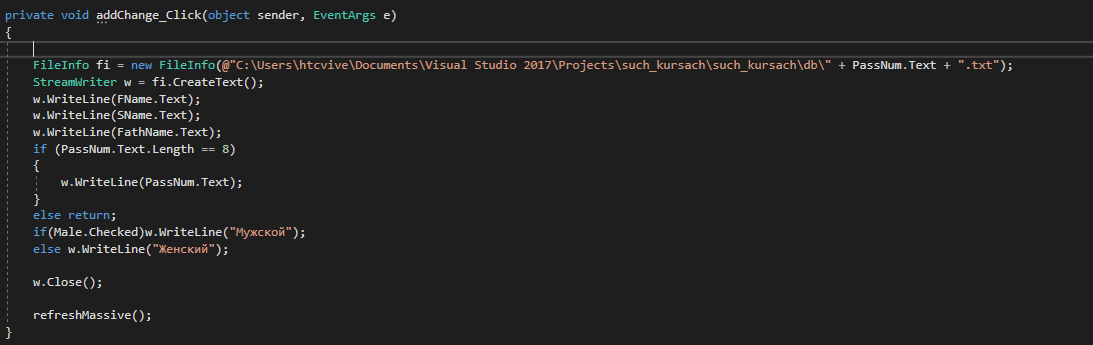
Первый блок – блок работы с клиентом: просмотра информации о нем, ее редактирование и просмотр истории его путешествий.

Второй блок – блок поиска информации о клиенте в базе данных модуля.

Третий блок – блок добавления рейса выбранному клиенту.

# **Функционал программы.**

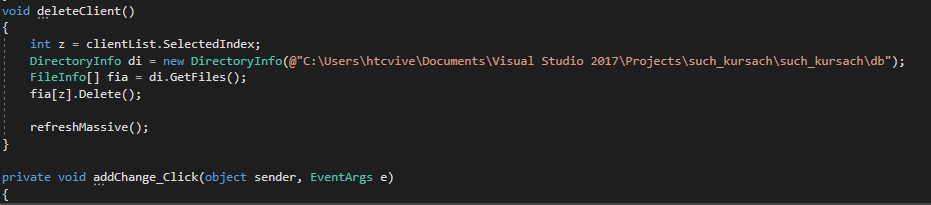
## **1.Функция добавления клиента.**



*Рис. 2. Код обработчика события «addChange\_Click»*

После нажатия кнопки управление передается обработчику события «addChange\_Click». Создается новый файл (если ранее не существовал клиент с таким номером паспорта) или открывается для записи уже существующий (если клиент существует). Программа берет данные веденые в полях ФИО, номера паспорта и проверяет выбранный пол, после чего добавляет в файл/изменяет уже существующую информацию в файле.

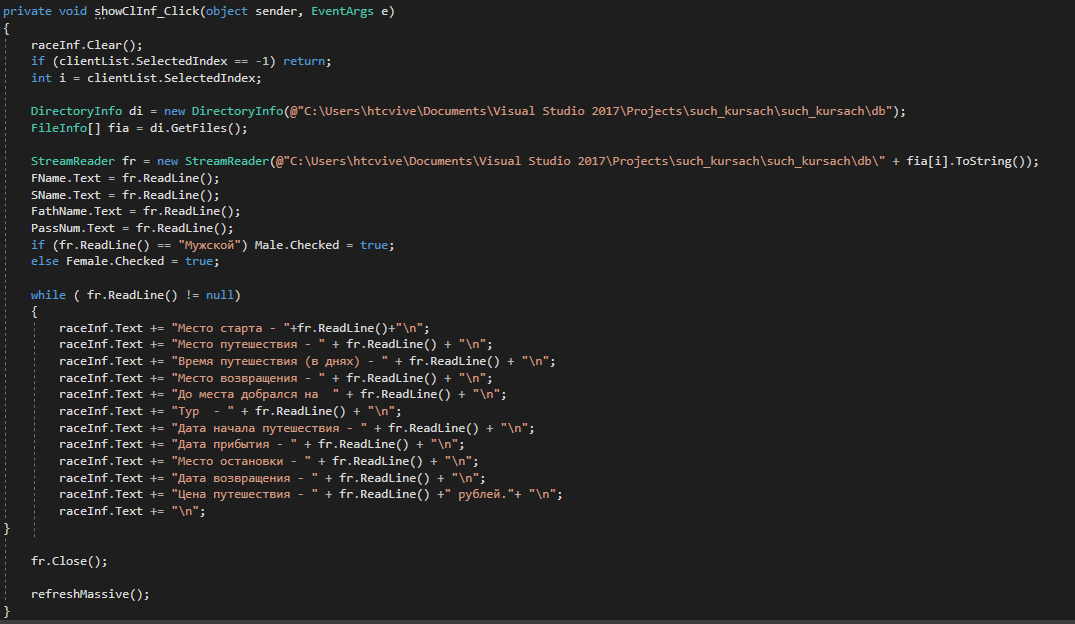
## **2.Функция удаления клиента.**



*Рис. 3. Код обработчика события «deleteClient»*

После нажатия кнопки управление передается обработчику события «deleteClient», который удаляет текстовый файл, выбранного в списке клиента и обновляет список клиентов.

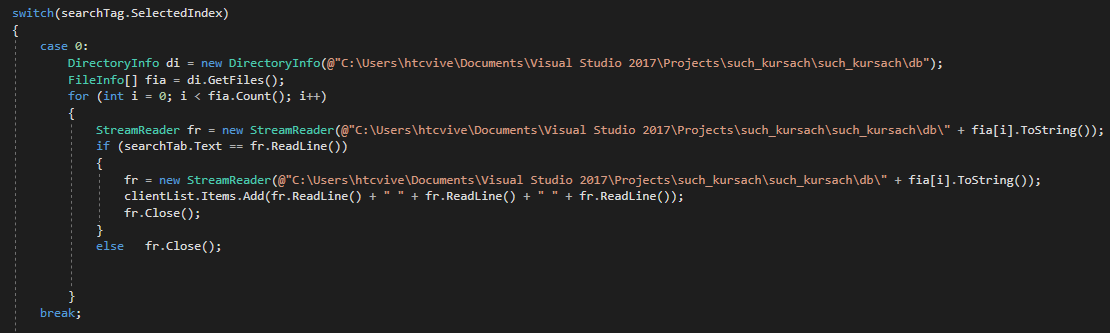
## **3.Функция просмотра информации о клиенте.**



*Рис. 4. Код обработчика события «showClInf\_Click»*

После нажатия кнопки управление передается обработчику события «showClInf\_Click», после чего происходит подгрузка текстового файла выбранного клиента: Поля заполняются соответствующей им информацией, и textbox снизу – полной информацией о всех совершенных клиентом рейсах.

## **4.Функция поиска клиента в БД.**

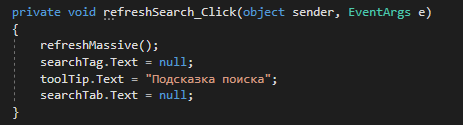


*Рис. 5. Код обработчика события «search\_Click»*

(В качестве примера приведен один из 5 блоков поиска по определенном признаку. Между собой они отличаются лишь индексом строки поиска.)

После нажатия кнопки управление передается обработчику события «search\_Click». Программа считывает индекс из атрибутов comboBox’a (searchTag) и запускает поисковый запрос с соответствующей индексу строкой в каждом тестовом файле каждого клиента. После, в списке клиентов останутся лишь удовлетворившие запросу клиенты.

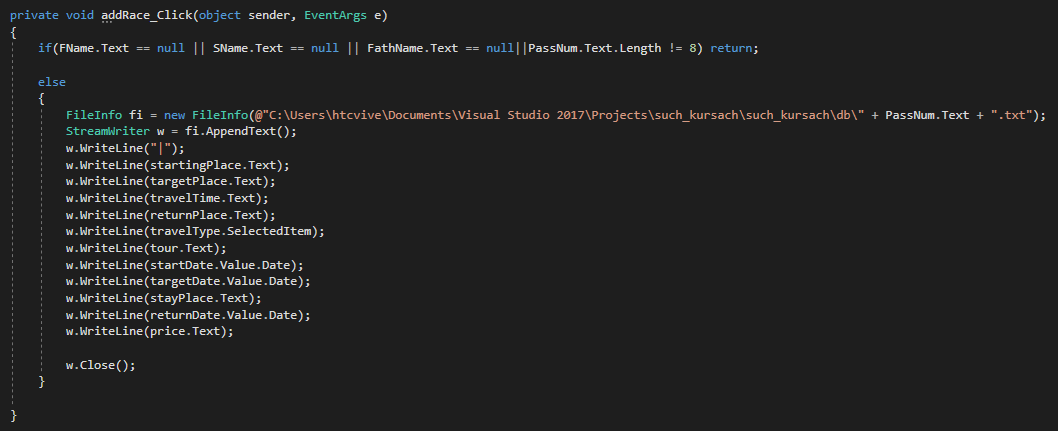
## **5.Функция сбросов параметров поиска.**



*Рис. 6. Код обработчика события «refreshSearch\_Click»*

После нажатия кнопки управление передается обработчику события «refreshSearch\_Click». Список клиентов снова будет состоять из всех доступных файлов, поле поискового запроса будет обнулено, а подсказка и элемент по которому ведется поиск вернутся к своим стандартным значениям.

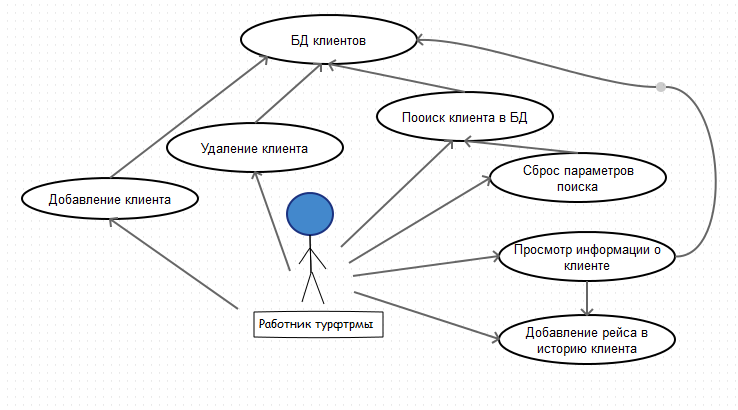
## **6. Функция добавления рейса клиенту.**



*Рис. 7. Код обработчика события «addRace\_Click»*

После нажатия кнопки управление передается обработчику события «addRace\_Click». Данные о рейсе будут считаны из соответствующих текстовых полей и добавлены выбранному клиенту в историю.

# **Схема работы программы.**



# **Этап отладки и завершающих работ.**

На этом этапе проводились финальные тесты программы, отладка уже имеющегося кода и шлифовка интерфейса. В ходе отладки были устранены несколько неприятных ошибок, вызывающих сбои в работе программы, интерфейс был приведен к своему конечному виду, работа программы была протестирована на нескольких пользователях, с разным уровнем владения компьютером.

# **Заключение.**

В ходе выполнения этой работы был разработан программный модуль, позволяющий ввести учет клиентов туристической фирмы.

Модуль позволяет добавлять и удалять пользователей, редактировать информацию о них, искать их в БД по выбранному параметру, а также добавлять рейсы и просматривать историю путешествий пользователя. Он является конечным продуктом и может быть внедрен в работу какого-либо туристического агентства.

Также были закреплены мои навыки работы с файлами, строками и формами в С#.

# **Приложение:**

Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace such\_kursach

{

public partial class Form1 : Form{

public Form1()

{

InitializeComponent();

refreshMassive();

}

void refreshMassive()

{

clientList.Items.Clear();

DirectoryInfo di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

FileInfo[] fia = di.GetFiles();

for (int i = 0; i < fia.Count(); i++)

{

StreamReader fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

clientList.Items.Add(fr.ReadLine() + " " + fr.ReadLine() + " " + fr.ReadLine());

fr.Close();

}

}

void deleteClient()

{

int z = clientList.SelectedIndex;

DirectoryInfo di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

FileInfo[] fia = di.GetFiles();

fia[z].Delete();

refreshMassive();

}

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

{

FileInfo fi = new FileInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + PassNum.Text + ".txt");

StreamWriter w = fi.CreateText();

w.WriteLine(FName.Text);

w.WriteLine(SName.Text);

w.WriteLine(FathName.Text);

if (PassNum.Text.Length == 8)

{

w.WriteLine(PassNum.Text);

}

else return;

if(Male.Checked)w.WriteLine("Мужской");

else w.WriteLine("Женский");

w.Close();

refreshMassive();

}

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

{

raceInf.Clear();

if (clientList.SelectedIndex == -1) return;

int i = clientList.SelectedIndex;

DirectoryInfo di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

FileInfo[] fia = di.GetFiles();

StreamReader fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\"+fia[i].ToString());

FName.Text = fr.ReadLine();

SName.Text = fr.ReadLine();

FathName.Text = fr.ReadLine();

PassNum.Text = fr.ReadLine();

if (fr.ReadLine() == "Мужской") Male.Checked = true;

else Female.Checked = true;

while ( fr.ReadLine() != null)

{

raceInf.Text += "Место старта - "+fr.ReadLine()+"\n";

raceInf.Text += "Место путешествия - " + fr.ReadLine() + "\n";

raceInf.Text += "Время путешествия (в днях) - " + fr.ReadLine() + "\n";

raceInf.Text += "Место возвращения - " + fr.ReadLine() + "\n";

raceInf.Text += "До места добрался на " + fr.ReadLine() + "\n";

raceInf.Text += "Тур - " + fr.ReadLine() + "\n";

raceInf.Text += "Дата начала путешествия - " + fr.ReadLine() + "\n";

raceInf.Text += "Дата прибытия - " + fr.ReadLine() + "\n";

raceInf.Text += "Место остановки - " + fr.ReadLine() + "\n";

raceInf.Text += "Дата возвращения - " + fr.ReadLine() + "\n";

raceInf.Text += "Цена путешествия - " + fr.ReadLine() +" рублей."+ "\n";

raceInf.Text += "\n";

}

fr.Close();

refreshMassive();

}

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

{

clientList.Items.Clear();

switch(searchTag.SelectedIndex)

{

case 0:

DirectoryInfo di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

FileInfo[] fia = di.GetFiles();

for (int i = 0; i < fia.Count(); i++)

{

StreamReader fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

if (searchTab.Text == fr.ReadLine())

{

fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

clientList.Items.Add(fr.ReadLine() + " " + fr.ReadLine() + " " + fr.ReadLine());

fr.Close();

}

else fr.Close();

}

break;

case 1:

di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

fia = di.GetFiles();

for (int i = 0; i < fia.Count(); i++)

{

StreamReader fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

fr.ReadLine();

{

fr.Close();

fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

clientList.Items.Add(fr.ReadLine() + " " + fr.ReadLine() + " " + fr.ReadLine());

}

fr.Close();

}

break;

case 2:

di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

fia = di.GetFiles();

for (int i = 0; i < fia.Count(); i++)

{

StreamReader fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

fr.ReadLine();

fr.ReadLine();

{

fr.Close();

fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

clientList.Items.Add(fr.ReadLine() + " " + fr.ReadLine() + " " + fr.ReadLine());

}

fr.Close();

}

break;

case 3:

di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

fia = di.GetFiles();

for (int i = 0; i < fia.Count(); i++)

{

StreamReader fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

fr.ReadLine();

fr.ReadLine();

fr.ReadLine();

{

fr.Close();

fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

clientList.Items.Add(fr.ReadLine() + " " + fr.ReadLine() + " " + fr.ReadLine());

}

fr.Close();

}

break;

case 4:

di = new DirectoryInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db");

fia = di.GetFiles();

for (int i = 0; i < fia.Count(); i++)

{

StreamReader fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

fr.ReadLine();

fr.ReadLine();

fr.ReadLine();

fr.ReadLine();

{

fr.Close();

fr = new StreamReader(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + fia[i].ToString());

clientList.Items.Add(fr.ReadLine() + " " + fr.ReadLine() + " " + fr.ReadLine());

}

fr.Close();

}

break;

}

}

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

{

refreshMassive();

searchTag.Text = null;

toolTip.Text = "Подсказка поиска";

searchTab.Text = null;

}

private void searchTag\_SelectedIndexChanged(object sender, EventArgs e)

{

switch (searchTag.SelectedIndex)

{

case 0: toolTip.Text = "Введите имя."; break;

case 1: toolTip.Text = "Введите фамилию."; break;

case 2: toolTip.Text = "Введите отчество."; break;

case 3: toolTip.Text = "Номер паспорта."; break;

case 4: toolTip.Text = "Введите пол (мужской/женский)"; break;

}

}

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

{

deleteClient();

}

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

{

if(FName.Text == null || SName.Text == null || FathName.Text == null||PassNum.Text.Length != 8) return;

else

{

FileInfo fi = new FileInfo(@"C:\Users\Михаил\Documents\Visual Studio 2013\Projects\such\_kursach\such\_kursach\db\" + PassNum.Text + ".txt");

StreamWriter w = fi.AppendText();

w.WriteLine("|");

w.WriteLine(startingPlace.Text);

w.WriteLine(targetPlace.Text);

w.WriteLine(travelTime.Text);

w.WriteLine(returnPlace.Text);

w.WriteLine(travelType.SelectedItem);

w.WriteLine(tour.Text);

w.WriteLine(startDate.Value.Date);

w.WriteLine(targetDate.Value.Date);

w.WriteLine(stayPlace.Text);

w.WriteLine(returnDate.Value.Date);

w.WriteLine(price.Text);

w.Close();

}

}

}

}

# **Список использованных информационных источников:**

Дж. Рихтер. CLR via C#. Программирование на платформе Microsoft. NET Framework 4.5 на языке C#. 4-е изд./Дж. Рихтер – СПб.: Питер – 2013.-896с.

Язык программирования C#. Классика Computer Science. 4-е изд./

А. Хейлсберг, М.Торгерсен, С.Вилтамут, П. Голд – СПб.: Питер – 2013.-784с.

Э.Стиллмен. Изучаем C#. 3-е изд./ Э.Стиллмен, Дж.Грин – СПб.: Питер – 2014.-816с.

Хабрахабр, справочная информация по языку C# -

http://[www.habrahabr.ru/hub/csharp/](http://www.habrahabr.ru/hub/csharp/)

Джозеф Албахари. C# 5.0. Справочник. Полное описание языка/ Джозеф Албахари, Бен Албахари - Москва.: Вильямс – 2014.-1008с.