**Московский авиационный институт**

(Национальный Исследовательский Университет).

Институт №3

«Системы управления, информатика и электроэнергетика».

Кафедра 304:

«Вычислительные машины, системы и сети».

Отчет по лабораторной работе №1

по учебной дисциплине

«Программирование на языках высокого уровня»

Часть 1

Выполнил  
Студент группы М3О-208Б  
Макаров Егор Алексеевич

Приняли

Барчев Николай Борисович

Титов Юрий Павлович

Москва, 2017 г.

**Содержание**

[Задание 3](#_Toc511151504)

[Описание метода решения поставленной задачи 4](#_Toc511151505)

[Описание основных алгоритмов решения задачи 4](#_Toc511151506)

[Псевдокод 5](#_Toc511151507)

[Сведения о программной реализации 6](#_Toc511151508)

[Инструкция пользователя 9](#_Toc511151509)

[Листинг программной разработки 15](#_Toc511151510)

[Результаты тестирования 21](#_Toc511151511)

# Задание

Часть 1

Подготовить программу, формирующую на основе информации, вводимой пользователем с клавиатуры, два внешних файла:

файл городов: состоит из записей, каждая из которых включает три поля – название города, количество жителей, название страны;  
файл стран: состоит из записей, каждая из которых включает три поля – название страны, название столицы, название континента.

В процессе проектирования предусмотреть необходимые по смыслу задания проверки корректности данных, а также адекватное задаче взаимодействие с пользователем.  
Учесть, что ограничения на размеры файлов отсутствуют, файлы в общем случае могут быть различной длины.  
**В ОБЯЗАТЕЛЬНОМ ПОРЯДКЕ ИСПОЛЬЗОВАТЬ языковые средства организации программных единиц. НЕ ИСПОЛЬЗОВАТЬ программные средства, структуры и форматы, поддерживающие работу с базами данных.**

Программу подготовить с использованием средств одной из реализаций языка программирования **С++**

# Описание метода решения поставленной задачи

Для выполнения задания подготовлено оконное приложение, предоставляющее его пользователю возможность записывать информацию о городах (название города, количество жителей, название страны) и странах (название страны, название столицы, название континента) в текстовые файлы типа .txt или .csv. Для записи информации пользователь может выбрать любой существующий файл любого из двух допустимых типов, а также выбрать режим работы: перезаписать выбранный файл или записывать новую информацию к уже имеющейся в файле. Информацию для записи в файл пользователь вводит с клавиатуры.

# Описание основных алгоритмов решения задачи

На этапе проектирования алгоритма выполнения поставленного задания были выделены следующие основные функциональные задачи:

1. Выбор файла для записи в него информации;
2. Ввод данных пользователем;
3. Проверка вводимых пользователем данных;
4. Запись введённых данных в выбранный файл;
5. Отображение сообщений пользователю

Решение функциональных задач:

1. Открывается диалоговое окно выбора файла (с ограничением на допустимый формат). После выбора файла пользователь выбирает режим работы с файлом – перезаписать его или добавить информацию к уже имеющейся;
2. Пользователь вводит с клавиатуры текстовые данные в поля ввода города/страны для записи их в соответствующий файл;
3. Поле количества жителей должно содержать в себе цифры и не должно содержать других символов, все остальные поля должны содержать в себе текст в виде букв и не должны содержать знаки препинания, цифры и другие символы, которые в основном не встречаются в названиях городов, стран и континентов. Регистр вводимых пользователем букв не является важным. Поля названия города для файла городов и названия страны для файла стран являются ключевыми и обязательными для ввода, остальные поля могут быть введены по желанию пользователя;
4. В файлы каждая новая запись будет добавляться на новой строке в формате: «поле\_1;поле\_2;поле\_3». Символ «;» (точка с запятой) служит разделителем между полями и является управляющим символом, не разрешённым для ввода (учесть его отсутствие при вводе информации в пункте 3). Пользователь добавляет запись в файл по нажатию кнопки.
5. В случае вопросов и ошибок, возникающих в процессе работы программы, например, пользователю нужно выбрать режим работы с файлом, или пользователь не ввёл обязательное для ввода поле записи, открывается диалоговое окно с соответствующим вопросом/ошибкой.

# Псевдокод

Начало работы программы

вызов функции выбора файла (для файла городов);

вызов функции выбора файла (для файла стран);

ввод с клавиатуры полей записей для соответствующих файлов;

если выбрана функция "добавить" для файла городов или для файла стран

то (1)

функция добавления записи, введённой пользователем, в соответствующий файл;

(1)

Конец работы программы.

функция выбора файла (для обоих файлов логика выполнения одинаковая)

Начало

выбор файла для записи данных;

если выбран существующий файл

то (2)

выбор режима использования файла (на дозапись/перезапись);

если выбрана перезапись

то (3)

очистить файл;

(3)

если выбрана дозапись

то (4)

файл будет использоваться на дозапись;

(4)

иначе (2)

сообщить пользователю, что такого файла не существует;

(2)

Конец.

функция ввода данных, введённых пользователем, в файл

Начало

проверка заполненности основных полей;

проверка на наличие в введённом поле только букв/только цифр (зависит от поля);

проверка на наличие в введённом поле знаков препинания/управляющих символов;

если хотя бы одна из проверок обнаружила ошибку

то (1)

сообщение пользователю, какая ошибка им допущена;

иначе (1)

запись введённых пользователем данных в файл;

(1)

Конец.

# Сведения о программной реализации

Приложение реализовано на языке программирования С++ в среде разработки Microsoft Visual Studio 2013 с использованием Windows Forms. Совместимо с операционными системами Windows 7, 8, 10. Требует 50 МБ свободного места на жёстком диске и 20 МБ оперативной памяти.

Все поля записей пользователь вводит с клавиатуры. Обязательными для ввода являются название города для файла городов и название страны для файла стран, остальные данные пользователю вводить не обязательно, на местах не введённых полей ставится «-» (прочерк).

**Сведения о программных единицах:**

Глобальные переменные

String^ townsFileName, String^ countriesFileName – служат для хранения имён выбранных файлов городов и стран соответственно.

OpenFileDialog^ fileDialog – экземпляр класса диалогового окна открытия файла для работы с диалоговым окном.

StreamWriter^ swTowns, StreamWriter^ swCountries – потоки записи информации о городах и странах в соответствующие файлы.

Процедуры – обработчики событий

System::Void MainForm\_Load – обработчик события «загрузка формы», описывает и устанавливает начальные значения и параметры для диалогового окна и элементов на форме.

System::Void buttonTownsFileChoose\_Click – обработчик события «нажатие на кнопку выбора файла городов», вызывает диалоговое окно выбора файла городов. Также происходит выбор режима работы с файлом.

System::Void buttonTownsAdd\_Click – обработчик события «нажатие на кнопку «Добавить» для файла городов», добавляет новую запись из введённых пользователем полей в файл городов, используя поток записи информации о городах, содержит в себе проверку введённых полей.

System::Void buttonCountriesFileChoose\_Click - обработчик события «нажатие на кнопку выбора файла стран», вызывает диалоговое окно выбора файла городов. Также происходит выбор режима работы с файлом.

System::Void buttonCountriesAdd\_Click– обработчик события «нажатие на кнопку «Добавить» для файла стран», добавляет новую запись из введённых пользователем полей в файл стран, используя поток записи информации о городах, содержит в себе проверку введённых полей.

System::Void textBox\_OnlyLetters\_KeyPress – обработчик события «ввод информации в поля», разрешает пользователю вводить в поля названия города, названия станы, столицы и материка только буквы.

System::Void textBox\_OnlyDigits\_KeyPress - обработчик события «ввод информации в поля», разрешает пользователю вводить в поле количества жителей только цифры.

Вспомогательные процедуры

System::Void ChangeEnable\_Towns(bool enable) – процедура, делающая элементы программы, отвечающие за работу с файлом городов, доступными или недоступными для работы. Имеет в качестве входного параметра логическую переменную, которая отвечает на вопрос, какими нужно сделать элементы. Вызывается в процедурах выбора файла городов, завершения работы с файлом городов и при загрузке формы.

System::Void ChangeEnable\_Countries(bool enable) – процедура, делающая элементы программы, отвечающие за работу с файлом стран, доступными или недоступными для работы. Имеет в качестве входного параметра логическую переменную, которая отвечает на вопрос, какими нужно сделать элементы. Вызывается в процедурах выбора файла стран, завершения работы с файлом стран и при загрузке формы.

System::Void ClearInputs\_Towns() – процедура очистки полей для ввода информации о городах. Вызывается в процедуре добавления информации в файл городов.

System::Void ClearInputs\_Countries() – процедура очистки полей для ввода информации о странах. Вызывается в процедуре добавления информации в файл стран.

System::Void Timer\_Elapsed\_Method (System::Object^ sender, System:: EventArgs^ e) – процедура, убирающая сообщение рядом с кнопкой «Добавить» о добавлении информации в тот или иной файл по истечению таймера. Вызывается в процедурах добавления информации в файлы городов и стран.

System::Void aboutToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) – процедура, выдающая на экран информацию о программе в виде диалогового окна с сообщением.

Диагностические сообщения

1. Сообщение пользователю о том, что он не ввёл ключевое поле «Название города» для файла городов или «Название страны» для файла стран. Появляется, если пользователь не ввёл соответствующее записи ключевое поле (Рис. 3, 4).

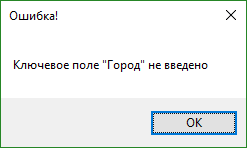
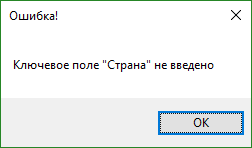
 

Рис. 2  
Сообщение о не введённом поле «Страна»

Рис. 1  
Сообщение о не введённом поле «Город»

1. Сообщение пользователю о том, что он выбрал для записи информации о городах (или странах) файл, уже использующийся для записи информации о странах (или городах). Возникает при попытке выбрать уже использующийся файл (Рис. 5, 6).

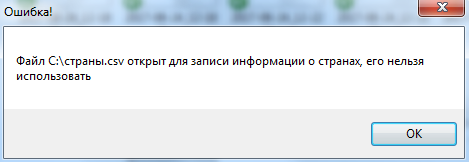


Рис. 3  
Сообщение об использовании файла для записи  
информации о странах

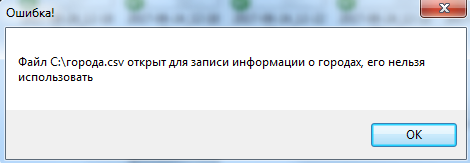


Рис. 4  
Сообщение об использовании файла для записи  
информации о городах

1. Сообщение пользователю о том, что файл с выбранным именем не найден. Возникает при выборе пользователем в диалоговом окне выбора файла городов/стран файла, которого не существует (Рис. 7).

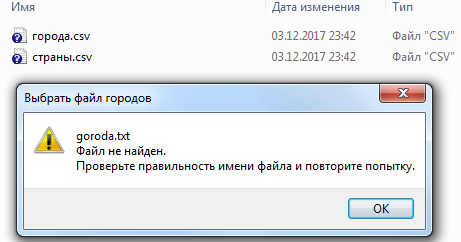


Рис. 5  
Сообщение о том, что выбранный файл не найден

# Инструкция пользователя

При запуске приложения открывается основное окно (Рис. 6). Оно состоит из двух частей, в каждой из которых происходит запись соответствующей информации (о городах или о странах) в соответствующие файлы. В начале работы возможность ввода полей и добавления записей недоступна, так как ещё не выбраны файлы. Для продолжения работы пользователю необходимо выбрать файлы, нажав на кнопку «Выбрать в файл» в части, соответствующей работе с файлом городов или стран.

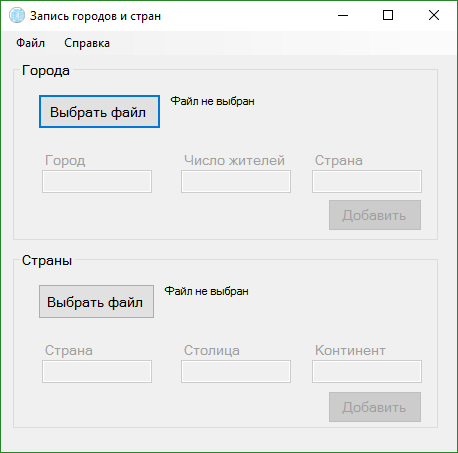


Рис. 6  
Основное окно приложения

По нажатию на кнопку «Выбрать файл» открывается диалоговое окно выбора файла, в котором пользователь может выбрать любой существующий файл формата .txt или .csv. (Рис. 7). После выбора файла и нажатия на кнопку «Открыть» пользователю предоставляется выбор режима работы с файлом: дозаписать информацию в файл, перезаписать весь файл или отменить выбор файла и вернуться к первоначальному окну без выбранного файла (Рис. 8). При выборе дозаписи (нажатии на кнопку «Да») последующая введённая информация будет добавляться к уже имеющейся в файле. При выборе перезаписи (нажатии на кнопку «Нет») файл очищается, и пользователь вводит информацию в пустой файл.

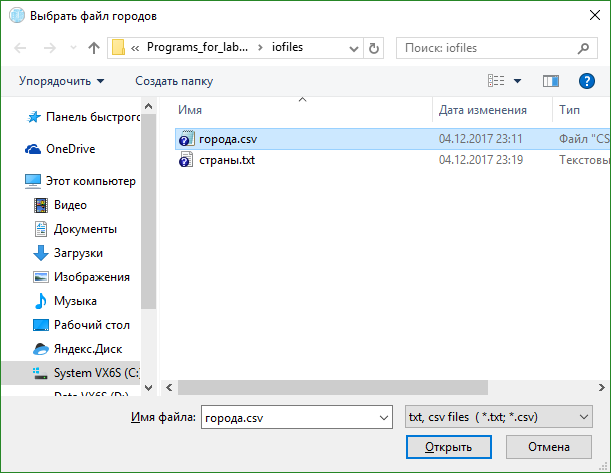


Рис. 7  
Диалоговое окно выбора файла (для файла городов)

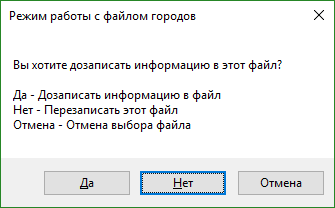


Рис. 8  
Окно выбора режима работы файла

После выбора файла и режима работы с ним поля ввода информации в файл становятся доступными для дальнейшей работы (Рис. 9), а рядом с кнопкой «Выбрать файл» написано имя выбранного файла и путь к нему.

Аналогичные действия могут быть произведены в части основного окна для работы с информацией о странах (Рис. 10). Для ввода разной информации нельзя использовать один и тот же файл.

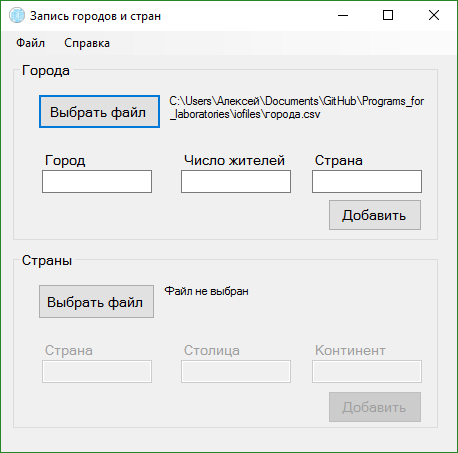


Рис. 9  
Основное окно после выбора файла городов

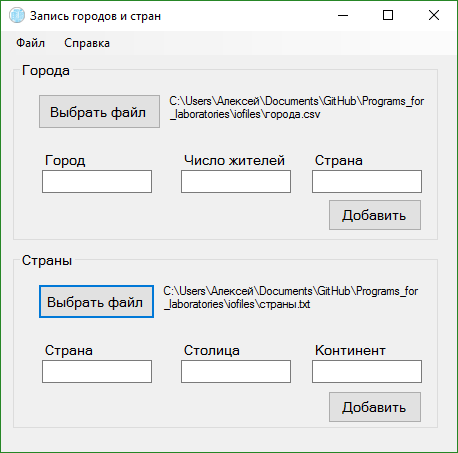


Рис. 10  
Основное окно после выбора файла городов

Далее пользователь может вводить информацию о городах и странах в соответствующие файлы, заполнив соответствующие поля (Рис. 11). Для добавления введённых полей в файл нужно нажать на кнопку «Добавить». По её нажатию введённая информация добавятся в соответствующий файл, а рядом с кнопкой «Добавить» появится сообщение о том, что информация была добавлена в файл, которое через некоторое время исчезнет, а поля для ввода автоматически очистятся, после чего можно продолжить ввод (Рис. 12). Такие поля, как «Город» для записи в фал городов и поле «Страна» для записи в файл стран должны быть введены пользователем обязательно, иначе добавить информацию в файл не получится.

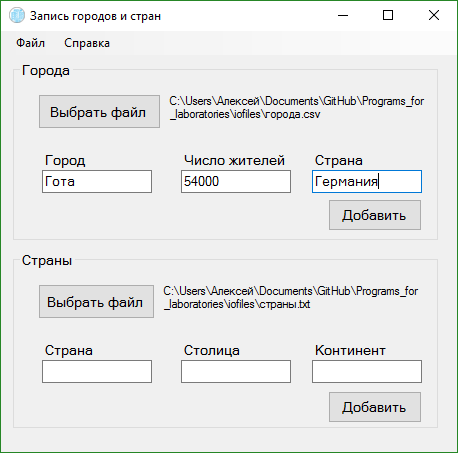


Рис. 11  
Основное окно с введёнными полями для записи в файл городов

Аналогичные действия могут быть произведены в части основного окна для работы с информацией о странах.

Для изменения файла, в который производится добавление вводимой информации, нужно повторно нажать на кнопку «Выбрать файл» и повторить все действия, относящиеся к выбору файла и описанные выше.

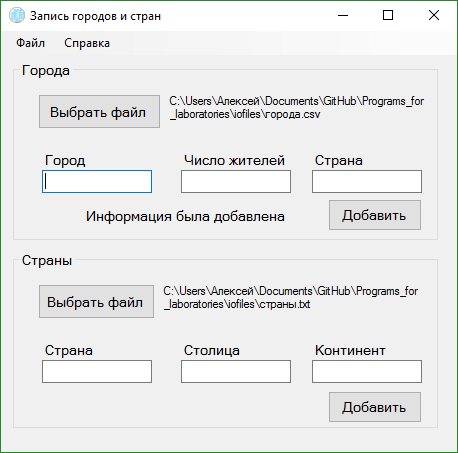


Рис. 12  
Основное окно после добавления информации в файл городов

В процессе работы программы формируются файлы городов и стран. Пример их содержания на рис. 13 и 14:

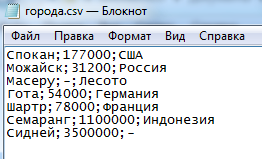
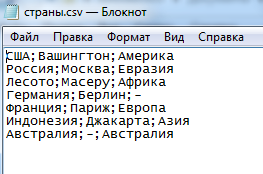
 

Рис. 14  
Содержание файла стран

Рис. 13  
Содержание файла городов

В меню «Справка» можно узнать сведения о данном приложении (Рис. 15).

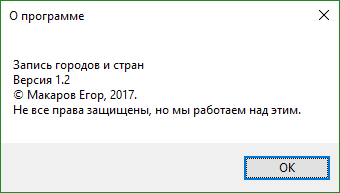


Рис. 15  
Меню «Справка» - «О программе»

Для завершения работы программы необходимо нажать на красный крестик в правом верхнем углу основного окна, либо выбрать меню «Файл» - «Выход» (Рис. 16).

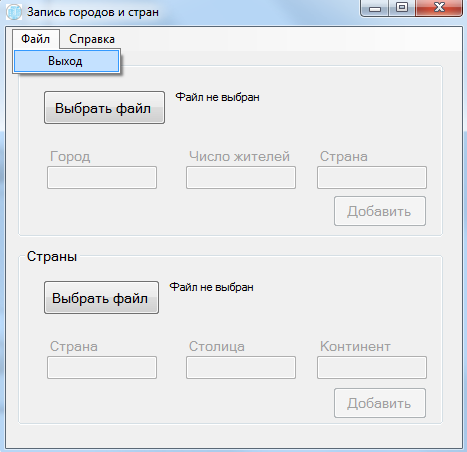


Рис. 16  
Меню «Файл» - «Выход»

# Листинг программной разработки

Файл MainForm.h

namespace LABA\_1\_1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::IO;

/// <summary>

/// Сводка для MainForm

/// </summary>

public ref class MainForm : public System::Windows::Forms::Form

{

public:

MainForm(void)

{

InitializeComponent();

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MainForm()

{

if (components)

{

delete components;

}

}

//Элементы на форме

private: System::Windows::Forms::GroupBox^ groupBoxTowns;

private: System::Windows::Forms::Button^ buttonTownsFileChoose;

private: System::Windows::Forms::Label^ labelTownsFileChoose;

private: System::Windows::Forms::Label^ labelTownsWriteTown;

private: System::Windows::Forms::Label^ labelTownsWritePeople;

private: System::Windows::Forms::Label^ labelTownsWriteCountry;

private: System::Windows::Forms::TextBox^ textBoxTownsWriteTown;

private: System::Windows::Forms::TextBox^ textBoxTownsWritePeople;

private: System::Windows::Forms::TextBox^ textBoxTownsWriteCountry;

private: System::Windows::Forms::Label^ labelTownsAdd;

private: System::Windows::Forms::Button^ buttonTownsAdd;

private: System::Windows::Forms::GroupBox^ groupBoxCountries;

private: System::Windows::Forms::Button^ buttonCountriesFileChoose;

private: System::Windows::Forms::Label^ labelCountriesFileChoose;

private: System::Windows::Forms::Label^ labelCountriesWriteCountry;

private: System::Windows::Forms::Label^ labelCountriesWriteCapital;

private: System::Windows::Forms::Label^ labelCountriesWriteContinent;

private: System::Windows::Forms::TextBox^ textBoxCountriesWriteCountry;

private: System::Windows::Forms::TextBox^ textBoxCountriesWriteCapital;

private: System::Windows::Forms::TextBox^ textBoxCountriesWriteContinent;

private: System::Windows::Forms::Label^ labelCountriesAdd;

private: System::Windows::Forms::Button^ buttonCountriesAdd;

private: System::Windows::Forms::Timer^ timer;

private: System::Windows::Forms::MenuStrip^ menuStrip;

private: System::Windows::Forms::ToolStripMenuItem^ справкаToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ aboutToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ файлToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ menuItemTownsStopInputToolStrip;

private: System::Windows::Forms::ToolStripMenuItem^ menuItemCountriesStopInputToolStrip;

private: System::ComponentModel::IContainer^ components;

/// <summary>

/// Обязательный метод для поддержки конструктора - не изменяйте

/// содержимое данного метода при помощи редактора кода.

/// </summary>

void InitializeComponent(void){ ... }

// Пути к файлам городов и стран для ввода информации

private: String^ townsFileName = String::Empty;

private: String^ countriesFileName = String::Empty;

// Диалоговое окно

private: OpenFileDialog^ fileDialog;

// Потоки для записи в файлы

private: StreamWriter^ swTowns;

private: StreamWriter^ swCountries;

/// <summary>

/// При загрузке формы

/// </summary>

private: System::Void MainForm\_Load(System::Object^ sender, System::EventArgs^ e) {

// Описание диалогового окно выбора файла

fileDialog = gcnew OpenFileDialog();

// Ограничения на тип выбираемого файла

fileDialog->Filter = L"txt, csv files | \*.txt; \*.csv";

// Установка пути к папке с файлами городов и стран

String^ dir = L"\\LABA\_1\_1\\LABA\_1\_1";

dir = Environment::CurrentDirectory->Remove

(Environment::CurrentDirectory->Length - dir->Length);

fileDialog->InitialDirectory = dir->Insert(dir->Length, L"\\iofiles");

// Активность элементов ввода = false, так как файлы не выбраны

ChangeEnable\_Towns(false);

ChangeEnable\_Countries(false);

labelTownsFileChoose->Text = "Файл не выбран";

labelCountriesFileChoose->Text = "Файл не выбран";

menuItemTownsStopInputToolStrip->Enabled = false;

menuItemCountriesStopInputToolStrip->Enabled = false;

// Пустые label'ы рядом с кнопками "Добавить"

labelTownsAdd->Text = String::Empty;

labelCountriesAdd->Text = String::Empty;

// Описание таймера для очистки label'ов рядом с кнопками "Добавить"

timer->Enabled = true;

timer->Interval = 3000;

timer->Tick += gcnew EventHandler(this, &MainForm::Timer\_Elapsed\_Method);

}

/// <summary>

/// Активация/дизактивация элементов ввода городов

/// <param name="enable"> True=активированы, Flase=Неактивны</param>

/// </summary>

private: System::Void ChangeEnable\_Towns(bool enable) {

textBoxTownsWriteTown->Enabled = enable;

textBoxTownsWritePeople->Enabled = enable;

textBoxTownsWriteCountry->Enabled = enable;

labelTownsWriteTown->Enabled = enable;

labelTownsWritePeople->Enabled = enable;

labelTownsWriteCountry->Enabled = enable;

labelTownsAdd->Enabled = enable;

buttonTownsAdd->Enabled = enable;

}

/// <summary>

/// Активация/дизактивация элементов ввода стран

/// <param name="enable">True=активированы Flase=Неактивны</param>

/// </summary>

private: System::Void ChangeEnable\_Countries(bool enable) {

textBoxCountriesWriteCountry->Enabled = enable;

textBoxCountriesWriteCapital->Enabled = enable;

textBoxCountriesWriteContinent->Enabled = enable;

labelCountriesWriteCountry->Enabled = enable;

labelCountriesWriteCapital->Enabled = enable;

labelCountriesWriteContinent->Enabled = enable;

labelCountriesAdd->Enabled = enable;

buttonCountriesAdd->Enabled = enable;

}

/// <summary>

/// Очистка полей ввода в GroupBox городов

/// </summary>

private: System::Void ClearInputs\_Towns() {

textBoxTownsWriteTown->Clear();

textBoxTownsWritePeople->Clear();

textBoxTownsWriteCountry->Clear();

}

/// <summary>

/// Очистка полей ввода в GroupBox стран

/// </summary>

private: System::Void ClearInputs\_Countries() {

textBoxCountriesWriteCountry->Clear();

textBoxCountriesWriteCapital->Clear();

textBoxCountriesWriteContinent->Clear();

}

/// <summary>

/// Событие по прошедшему timer->Tick прекращает таймер и обнуляет label'ы

/// </summary>

private: System::Void Timer\_Elapsed\_Method(System::Object^ sender, System::EventArgs^ e) {

timer->Stop();

labelTownsAdd->Text = String::Empty;

labelCountriesAdd->Text = String::Empty;

}

/// <summary>

/// Выбор файла городов

/// </summary>

private: System::Void buttonTownsFileChoose\_Click(System::Object^ sender, System::EventArgs^ e){

// Показывается диалоговое окно выбора файла

fileDialog->Title = L"Выбрать файл городов";

System::Windows::Forms::DialogResult res = this->fileDialog->ShowDialog();

// Файл не выбран

if (res == System::Windows::Forms::DialogResult::Cancel)

return;

townsFileName = fileDialog->FileName; // Запись пути к файлу городов

//Если файл уже выбран для записи стран

if (String::Equals(townsFileName, countriesFileName)) {

System::Windows::Forms::MessageBox::Show(L"Файл "+ countriesFileName + L" открыт для записи информации о странах, его нельзя использовать", L"Ошибка!");

return;

}

res = System::Windows::Forms::MessageBox::Show(L"Вы хотите дозаписать

информацию в этот файл?\n\nДа - Дозаписать информацию в файл\nНет –

Перезаписать этот файл\nОтмена - Отмена выбора файла", L"Режим работы с

файлом городов", System::Windows::Forms::MessageBoxButtons::YesNoCancel);

// Вопрос о до/пере-записи

bool append;

if (res == System::Windows::Forms::DialogResult::Yes)

append = true;

else if (res == System::Windows::Forms::DialogResult::No)

append = false;

else return;

// Если поток вывода уже используется - закрыть поток

if (swTowns != nullptr)

swTowns->Close();

//если append=true, то дозапись, иначе перезапись

swTowns = gcnew StreamWriter(townsFileName, append);

labelTownsFileChoose->Text = townsFileName;

ChangeEnable\_Towns(true); // Активация элементов ввода информации о городах

menuItemTownsStopInputToolStrip->Enabled = true;

}

/// <summary>

/// Добавление новой записи в файл городов

/// </summary>

private: System::Void buttonTownsAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

// Проверка наличия введённого ключевого поля

if (textBoxTownsWriteTown->Text == String::Empty) {

System::Windows::Forms::MessageBox::Show(L"Ключевое поле \"Город\" не

введено", L"Ошибка!");

textBoxTownsWriteTown->Focus();

return;

}

// Пропуск вместо невведённых полей

if (textBoxTownsWritePeople->Text == String::Empty)

textBoxTownsWritePeople->Text = "-";

if (textBoxTownsWriteCountry->Text == String::Empty)

textBoxTownsWriteCountry->Text = "-";

// Запись через поток StreamWriter, его очистка и очистка полей ввода

swTowns->WriteLine(textBoxTownsWriteTown->Text + ";" +

textBoxTownsWritePeople->Text + ";" + textBoxTownsWriteCountry->Text);

swTowns->Flush();

ClearInputs\_Towns();

textBoxTownsWriteTown->Focus();

labelTownsAdd->Text = L"Информация была добавлена";

timer->Start(); // Начало таймера для label'а

}

/// <summary>

/// Выбор файла стран

/// </summary>

private: System::Void buttonCountriesFileChoose\_Click(System::Object^ sender,System::EventArgs^ e){

// аналог buttonTownsFileChoose\_Click

// Показывается диалоговое окно выбора файла

fileDialog->Title = L"Выбрать файл стран";

System::Windows::Forms::DialogResult res = this->fileDialog->ShowDialog();

if (res == System::Windows::Forms::DialogResult::Cancel)

return;

countriesFileName = fileDialog->FileName;

// Если файл уже выбран для записи городов

if (String::Equals(townsFileName, countriesFileName)) {

System::Windows::Forms::MessageBox::Show(L"Файл " + countriesFileName + L" открыт для записи информации о городах, его нельзя использовать", L"Ошибка!");

return;

}

res = System::Windows::Forms::MessageBox::Show(L"Вы хотите дозаписать

информацию в этот файл?\n\nДа - Дозаписать информацию в файл\nНет –

Перезаписать этот файл\nОтмена - Отмена выбора файла", L"Режим работы с

файлом городов", System::Windows::Forms::MessageBoxButtons::YesNoCancel);

// Вопрос о до/пере-записи

bool append;

if (res == System::Windows::Forms::DialogResult::Yes)

append = true;

else if (res == System::Windows::Forms::DialogResult::No)

append = false;

else return;

// Если поток вывода уже используется - закрыть поток

if (swCountries != nullptr)

swCountries->Close();

swCountries = gcnew StreamWriter(countriesFileName, append);

labelCountriesFileChoose->Text = countriesFileName;

ChangeEnable\_Countries(true);

menuItemCountriesStopInputToolStrip->Enabled = true;

}

/// <summary>

/// Добавление новой записи в файл стран

/// </summary>

private: System::Void buttonCountriesAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

// аналог buttonTownsAdd\_Click

// Проверка наличия введённого ключевого поля

if (textBoxCountriesWriteCountry->Text == String::Empty) {

System::Windows::Forms::MessageBox::Show(L"Ключевое поле \"Страна\" не введено", L"Ошибка!");

textBoxCountriesWriteCountry->Focus();

return;

}

// Пропуск вместо невведённых полей

if (textBoxCountriesWriteCapital->Text == String::Empty)

textBoxCountriesWriteCapital->Text = "-";

if (textBoxCountriesWriteContinent->Text == String::Empty)

textBoxCountriesWriteContinent->Text = "-";

swCountries->WriteLine(textBoxCountriesWriteCountry->Text + ";"

+ textBoxCountriesWriteCapital->Text + ";" + textBoxCountriesWriteContinent

->Text);

swCountries->Flush();

ClearInputs\_Countries();

textBoxCountriesWriteCountry->Focus();

labelCountriesAdd->Text = L"Информация была добавлена";

timer->Start(); // Начало таймера для label'а

}

/// <summary>

/// Ограничение на ввод в textBox'ы - только буквы

/// </summary>

private: System::Void textBox\_OnlyLetters\_KeyPress(System::Object^ sender,

System::Windows::Forms::KeyPressEventArgs^ e) {

if (!Char::IsLetter(e->KeyChar) && !(e->KeyChar == (char)8))

// (char)8 = Backspace

e->Handled = true;

}

/// <summary>

/// Ограничение на ввод в textBox'ы - только цифры

/// </summary>

private: System::Void textBox\_OnlyDigits\_KeyPress(System::Object^ sender,

System::Windows::Forms::KeyPressEventArgs^ e) {

if (!Char::IsDigit(e->KeyChar) && !(e->KeyChar == (char)8))

e->Handled = true;

}

/// <summary>

/// О программе

/// </summary>

private: System::Void aboutToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

MessageBox::Show(L"Запись городов и стран\nВерсия 1.2\n© Макаров Егор, 2017.\nНе все права защищены, но мы работаем над этим.", L"О программе");

}

Файл MainForm.cpp

#include "MainForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

void Main(array<String^>^ args) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

LABA\_1\_1::MainForm form;

Application::Run(%form);

}

# Результаты тестирования

Тест №1

Цель теста: проверить работу программы в случае выбора несуществующего файла для записи информации о городах/странах.

Входные данные: в диалоговом окне выбора файла городов/стран введено имя несуществующего файла (Рис. 17)

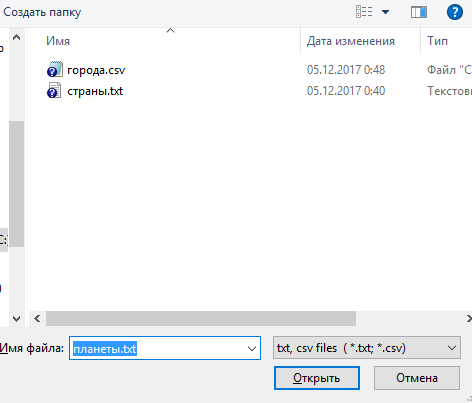


Рис. 17  
В диалоговом окне введено имя несуществующего файла

Полученный результат: соответствующее сообщение об ошибке (Рис 18).

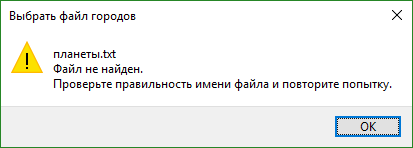


Рис. 18  
Сообщение о том, что выбранный файл не найден

Вывод: ошибок в работе программы не обнаружено.

Тест №2

Цель теста: проверить работу программы в случае выбора для записи информации о странах файла, выбранного для записи информации о городах и наоборот.

Входные данные: в диалоговом окне выбора файла стран выбирается файл, уже выбранный для записи информации о городах, и наоборот: в диалоговом окне выбора файла городов выбирается фай, уже выбранный для записи информации о странах.

Полученный результат: соответствующие сообщения об ошибках (Рис. 19, 20).

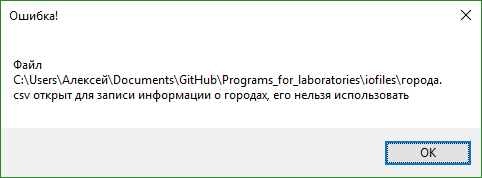


Рис. 19  
Сообщение о том, что выбранный файл уже открыт для записи информации о городах

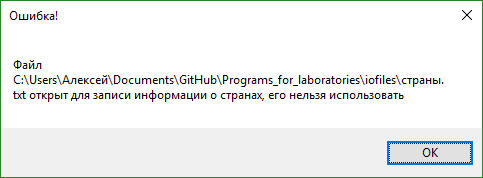


Рис. 20  
Сообщение о том, что выбранный файл уже открыт для записи информации о городах

Вывод: ошибок в работе программы не обнаружено.

Тест №3

Цель теста: проверить работу программы при попытке добавить информацию в файл при не введённом поле, обязательном для ввода (для информации о городах и для информации о странах).

Входные данные: обязательные для ввода поля не введены, остальные поля могут быть как введены, так и остаться пустыми (Рис. 21).

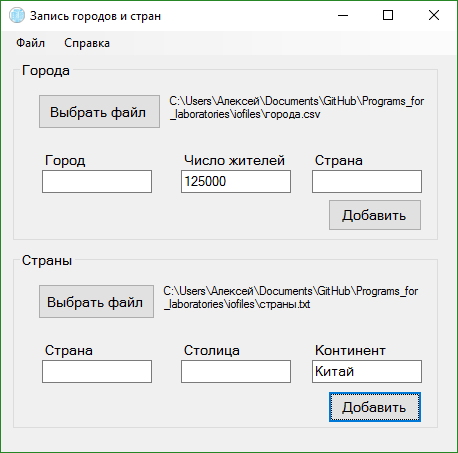


Рис. 21  
Обязательные для ввода поля не введены

Полученный результат: сообщения об ошибках (Рис. 21, 22).

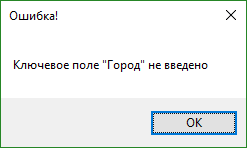
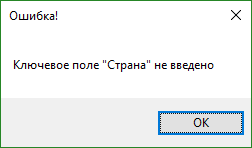
 

Рис. 23  
Сообщение о не введённом поле «Страна» для файла стран

Рис. 22  
Сообщение о не введённом поле «Город» для файла городов

Вывод: ошибок в работе программы не обнаружено.

Тест №4

Цель теста: проверить работу программы при попытке добавления в файл городов/стран записей, состоящих только из обязательных для ввода полей.

Входные данные: для записи информации в файл городов заполнено только поле «Город», остальные поля не заполнены. Для записи информации в файл стран заполнено только поле «Страна», остальные поля не заполнены (Рис. 24).

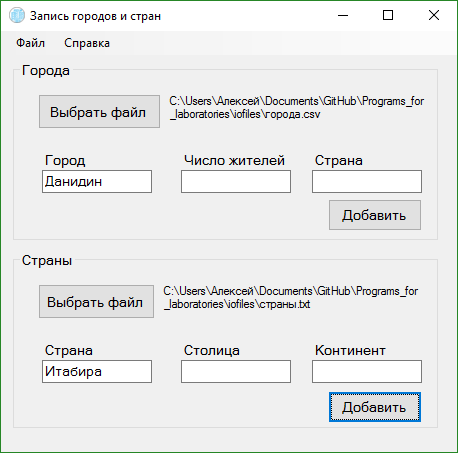


Рис. 24  
Заполнены только поле «Город» для файла городов и поле «Страна» для файла стран

Полученный результат: по нажатию кнопок «Добавить» для файлов городов и стран информация в каждый из файлов будет добавлена (Рис. 25).

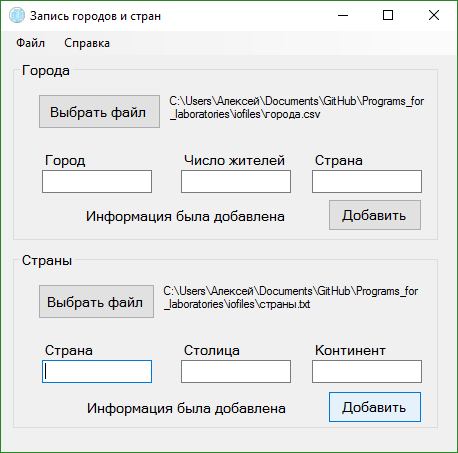


Рис. 25  
Информация была добавлена в оба файла

Вывод: ошибок в работе программы не обнаружено.