**Московский авиационный институт**

(Национальный Исследовательский Университет).

Факультет №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 2

Выполнил  
Студент группы М3О-208Б  
Макаров Егор Алексеевич

Принял:

Барчев Николай Борисович

Титов Юрий Павлович

Москва, 2018 г.

**Содержание**

[Задание 3](#_Toc512251545)

[Псевдокод 4](#_Toc512251546)

[Сведения о программной реализации 6](#_Toc512251547)

[Инструкция пользователя 12](#_Toc512251548)

[Листинг программной разработки 18](#_Toc512251549)

[Результаты тестирования 27](#_Toc512251550)

# Задание

Часть 2

Подготовить программу, формирующую сведения о расположении городов по странам и континентам (столичные города выделить) на основе данных из двух внешних файлов, сформированных в результате работы программы, подготовленной в части 1.

В процессе проектирования предусмотреть необходимые по смыслу задания проверки корректности данных, а также адекватное задаче взаимодействие с пользователем.

**Во второй программе реализовать возможность вывода итоговой информации в двух режимах: избирательно на экран (по запросу пользователем конкретных данных с клавиатуры) и полностью – в отдельный текстовый файл. Оба режима вывода должны предоставлять СГРУППИРОВАННУЮ ПО СМЫСЛУ информацию ИЗ ОБОИХ ВХОДНЫХ ФАЙЛОВ в виде, УДОБНОМ ДЛЯ ВОСПРИЯТИЯ 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 **Паскаль.**

# Псевдокод

Существует (объявлено) некое хранилище общей информации о каждом городе, для которого нашлась общая информация из обеих файлов исходных данных. В начале работы оно пустое/ничего не содержит

Начало работы программы

вызов функции выбора файла (для файла городов);

вызов функции выбора файла (для файла стран);

если оба файла выбраны, то

вызов процедуры объединения информации из файлов городов и стран

// Для вывода информации на экран по запросу пользователя

ввод с клавиатуры поля-запроса для поиска;

вызов процедуры поиска по запросу и печати найденной информации на экран;

// Для вывода общей информации в отдельный файл

вызов функции выбора файла (для файла общей информации);

вызов процедуры печати общей информации в выбранный файл;

Конец работы программы.

Функции для программы:

функция выбора файла (для всех файлов логика выполнения одинаковая)

Начало

выбор файла для записи данных;

если выбран не существующий файл

то (1)

сообщить пользователю, что такого файла не существует;

(1)

если файл уже выбран в программе

то (2)

сообщить пользователю, что файл уже используется программой;

(2)

чтение информации из файла;

если файл пуст или содержит невозможную для чтения информацию

то (3)

сообщить пользователю, что данный файл не подходит по той или иной причине;

файл остаётся не выбранным;

конец;

(3)

Конец.

процедура объединения информации из файлов городов и стран

Начало

для каждой строки с информацией о городе из файла городов

делать (1)

для каждой строки с информацией о стране из файла стран

делать (2)

если названия стран из обеих строк совпадают

то (3)

сохранить строку как элемент хранилища с общей информацией

в формате «город;население;страна;континент»;

если название города совпадает со столицей страны, то

приписать к названию города «(столица)»;

(3)

(2)

(1)

Конец.

процедура поиска по запросу и печати найденной информации на экран

Начало

если поле для запроса не заполнено

то (1)

ничего не выводить;

конец.

(1)

поиск введённого пользователем запроса во всех элементах хранилища общей информации;

если введённое поле-запрос нашлось

то (3)

печать элемента хранилища;

(3)

если введённое поле-запрос не нашлось

то (4)

сообщить пользователю, что по данному запросу ничего не обнаружено;

(4)

Конец.

процедура печати общей информации в выбранный файл

Начало

если файл для общей информации не выбран

то (1)

сообщить пользователю, что для выполнения процедуры

необходимо выбрать файл для общей информации;

конец.

(1)

очистить файл для общей информации;

для каждого элемента хранилища общей информации

делать (2)

вывести в файл информацию в формате «город;население;страна;континент»;

(2)

Конец.

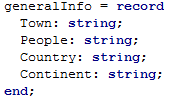
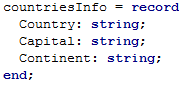
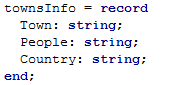
# Сведения о программной реализации

Приложение реализовано на языке программирования Delphi в среде разработки Borland Delphi 10 Microsoft Windows. Совместимо с операционными системами Windows 7, 8, 10. Требует   
3 МБ свободного места на жёстком диске и 500 КБ оперативной памяти.

Пользователю необходимо выбрать файлы с информацией о городах и странах, сформированных в процессе работы программы, описание которой представлено в части 1 этой работы. После чего формируется файл с общей информацией о городах и странах, среди которой пользователь может по запросу, введённому с клавиатуры, найти необходимую. Информация по запросу пользователя отображается на экране в виде таблицы.

**Сведения о программных единицах:**

Типы данных



Структуры, описывающие информацию о городах, о странах и общую информацию.

Глобальные переменные

townsFileName, countriesFileName, generalInfoFileName: string – имена выбранных пользователем файлов с информацией о городах, о странах и файла с общей информацией.

openFileDialog: TOpenDialog – диалоговое окно выбора файла.

towns: array of townsInfo – информация, считанная из файла городов.

countries: array of countriesInfo – ифнормация, считанная из файла стран.

generalInfoBase: array of generalInfo – общая информация, сформированная из информации, считанной из фйалов городов и стран. Выступает в роли хранилища общей информации, описанного в алгоритме (псевдокоде).

towns\_N, countries\_N, general\_N: integer – число записей о городах, странах и в хранилище общей информации.

Процедуры – обработчики событий

procedure TMainForm.FormCreate - обработчик события «загрузка формы», описывает и устанавливает начальные значения и параметры для диалогового окна, элементов на форме и глобальных переменных.

procedure TMainForm.menuAboutClick – обработчик события «Нажатие на кнопку «О программе» в меню «Справка»», отображает окно с информацией о программе.

procedure TMainForm.buttonTownsFileChooseClick – обработчик события «нажатие на кнопку выбора файла городов», вызывает диалоговое окно выбора файла городов и происходит чтение информации из выбранного файла.

procedure TMainForm.buttonCountriesFileChooseClick - обработчик события «нажатие на кнопку выбора файла стран», вызывает диалоговое окно выбора файла стран и происходит чтение информации из выбранного файла.

procedure TMainForm.buttonGeneralInfoFileChooseClick - обработчик события «нажатие на кнопку выбора файла для общей информации», вызывает диалоговое окно выбора файла для записи общей информации.

procedure TMainForm.buttonGeneralInfoWriteClick – обработчик события «нажатие на кнопку сохранения информации в файл», происходит запись всей информации из хранилища общей информации в выбранный для неё файл.

procedure TMainForm.textboxRequestFieldKeyPress – обработчик события «ввод поискового запроса в поле для запроса», происходит запрет на ввод определённых символов.

procedure TMainForm.buttonRequestInfoFindClick – обработчик события «нажатие на кнопку поиска по запросу», осуществляется поиск информации по запросу пользователя и печать её на экран.

Вспомогательные процедуры

procedure Split(Delimiter: Char; Str: string; ListOfStrings: TStrings) – разделяет строку по разделителю. Входные параметры – символ-разделитель строки, сама строка, и контейнер для хранения получившихся после разделения подстрок.

procedure PrintInfoLine(k: integer; i: integer) - печать строки таблицы с общей информации на экран, где k – номер строки в таблице на экране, I – номер строки с информацией как элемента в хранилище общей информации.

procedure ChangeEnable(enable: Boolean) - активация/дизактивация элементов поиска и печати, где значение enable определяет действие: активация или дизактивация.

procedure ClearStringGrid(StringGrid: TStringGrid) - очищает таблицу с информацией, где StringGrid – очищаемая таблица.

procedure ShowSearch() – активация элементов поиска и печати.

procedure SetFileNameToLabel(fileName: string; fileLabel: TLabel) - добавление текста с файлом в текст label'а, где filename – имя файла, fileLabel – используемый label для записи текста.

procedure MergeFiles() - объединение (слияние) информации из двух файлов. Общая информация записывается в хранилище общей информации.

procedure EventsAfterChoose(secondFile: string) - действия после выбора обеих файлов исходных данных, где secondFile – файл, выбранный перед тем, который был выбран последним. Действия – активация элементов поиска и слияние файлов, если оба файла выбраны.

Диагностические сообщения

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. 1).

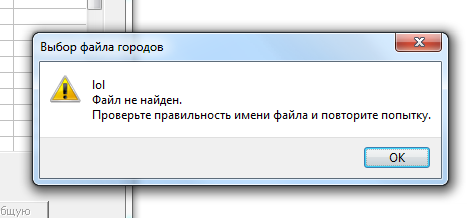


Рис. 1  
Сообщение о том, что выбранный файл не найден

Сообщение пользователю о том, что в выбранных файлах городов и стран нет общей информации (Рис. 2).

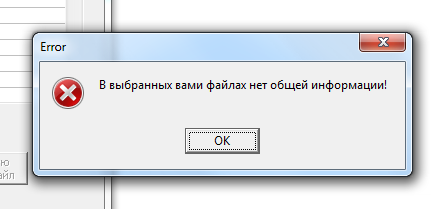


Рис. 2  
Сообщение о том, что в выбранных файлах нет общей информации

Сообщение пользователю о том, что выбранный в качестве файла городов/стран файл содержит информацию в некорректном формате, с примером формата информации в файле  
(Рис. 3, 4).

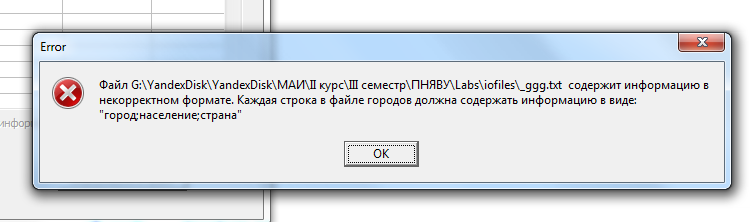


Рис. 3  
Сообщение о некорректности формата информации в файле городов

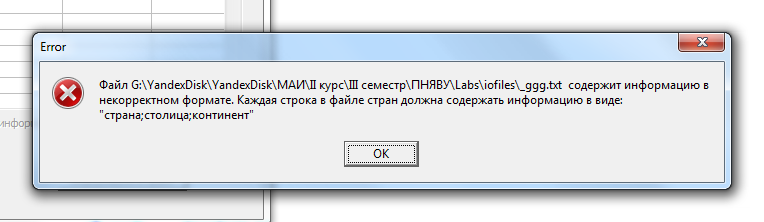
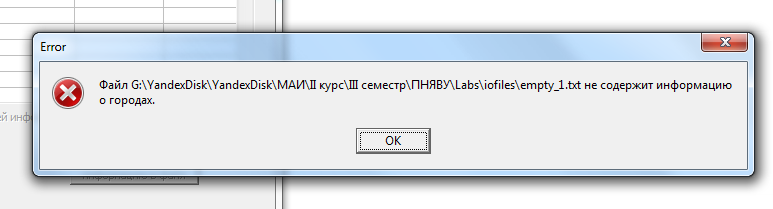


Рис. 5  
Сообщение о пустоте файла городов

Рис. 4  
Сообщение о некорректности формата информации в файле стран

Сообщение о том, что выбранный в качестве файла городов/стран файл не содержит информацию о городах/странах (Рис. 5, 6).



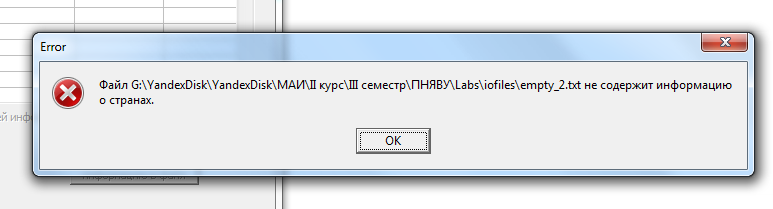


Рис. 6  
Сообщение о пустоте файла стран

Сообщение о том, что выбранный пользователем файл уже выбран и используется в качестве файла городов/стран/файла для общей информации (Рис. 7, 8, 9).

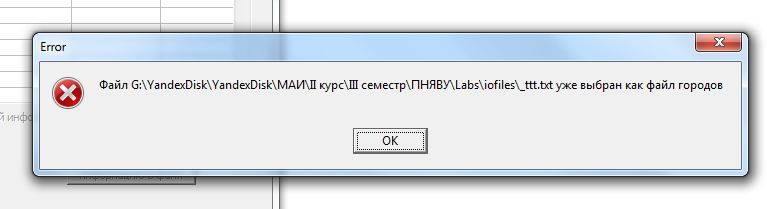
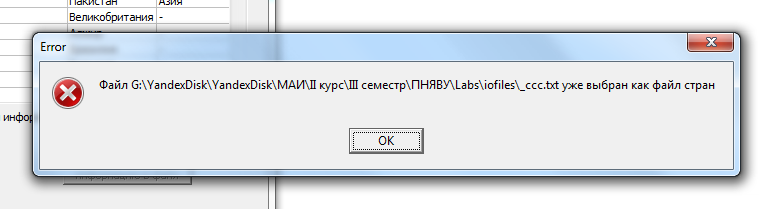


Рис. 8  
Сообщение о том, что файл уже выбран как файл стран

Рис. 7  
Сообщение о том, что файл уже выбран как файл городов



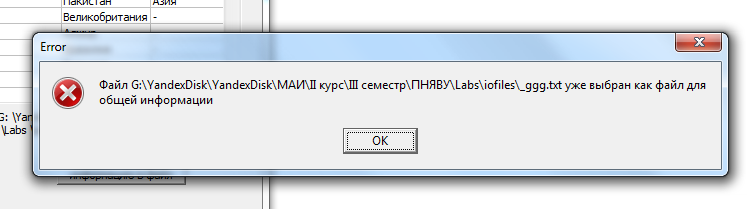


Рис. 9  
Сообщение о том, что файл уже выбран как файл для общей информации

Сообщение о том, что по запросу пользователя среди общей информации ничего не найдено (Рис. 10).

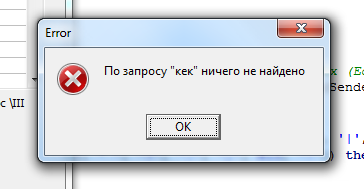


Рис. 10  
Сообщение о том, что по запросу пользователя ничего не найдено

# Инструкция пользователя

При запуске приложения открывается основное окно (Рис. 11). В начале работы пользователю доступны функции выбора файла. Пока файлы не выбраны, дальнейшая работа невозможна. Нажав на кнопки «Выбрать файл городов» и «Выбрать файл стран», пользователь выбирает соответствующие файлы в диалоговом окне выбора файла.

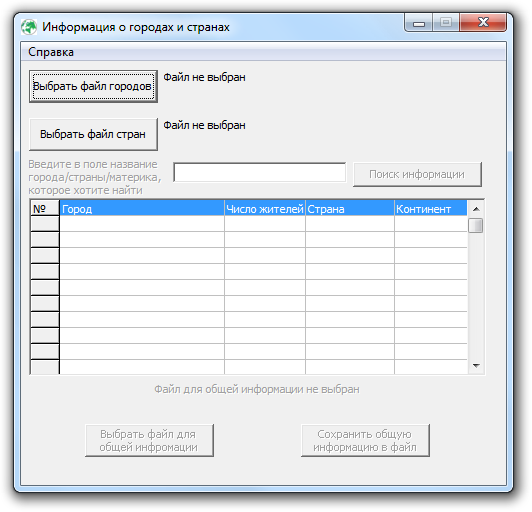


Рис. 11  
Основное окно приложения

Чтобы избежать ошибок в представлении информации в выбираемых пользователем файлах, рекомендуется использовать файлы, сформированные в процессе работы приложения «Города и страны» (часть 1 данной работы).

После выбора обоих файлов исходных данных, программа формирует на их основе общую информацию о городах и странах. Также пользователю становятся доступными функции работы с общей информацией: функция поиска общей информации по запросу и сохранения общей информации в файл (Рис. 12). Чтобы пользователь убедился, что программа выполняет данную задачу, на экране отображается общая информация в виде таблицы.

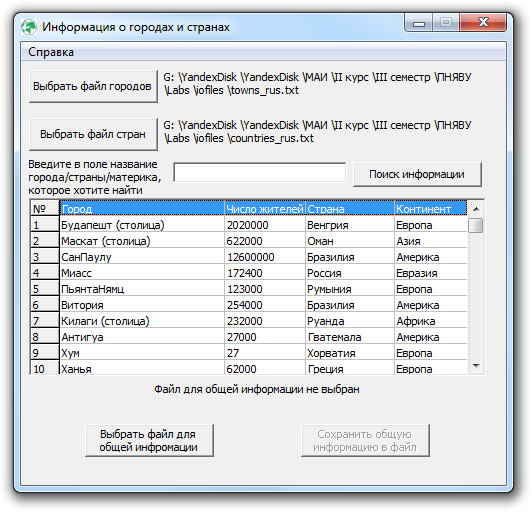


Рис. 12  
Основное окно приложения после выбора обоих файлов исходных данных

Для поиска информации по запросу, пользователю необходимо ввести в поле над таблицей свой запрос. Это может быть название города, страны, континента или любое другое слово или название, информацию о котором пользователь захочет найти. После заполнения поля для ввода запроса нужно нажать на кнопку «Поиск информации», после чего в таблице отобразится только та информация, которая удовлетворяет запросу пользователя, то есть содержит введённое название (Рис. 13).

Если по запросу пользователя ничего не будет найдено, на экране появится сообщение пользователю об этом, а таблица станет пустой.

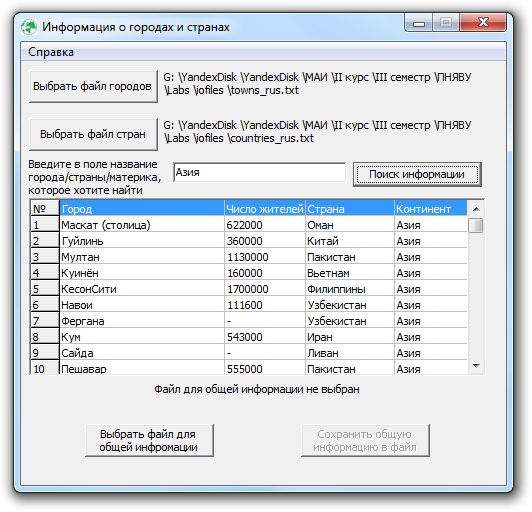


Рис. 13  
Основное окно приложения после поиска информации по запросу пользователя

Для того, чтобы сохранить общую информацию о городах и странах в файл, сначала необходимо выбрать файл для общей информации. После нажатия кнопки «Выбрать файл для общей информации» пользователь может выбрать файл в открывшемся диалоговом окне выбора файла. После выбора файла под таблицей будет написано, что для общей информации файл выбран (Рис. 14). Пользователю становится доступной кнопка «Сохранить общую информацию в файл». Нажав на неё, общая информация сохранится в выбранный пользователем файл, и под таблицей появится соответствующее сообщение об этом (Рис. 15).

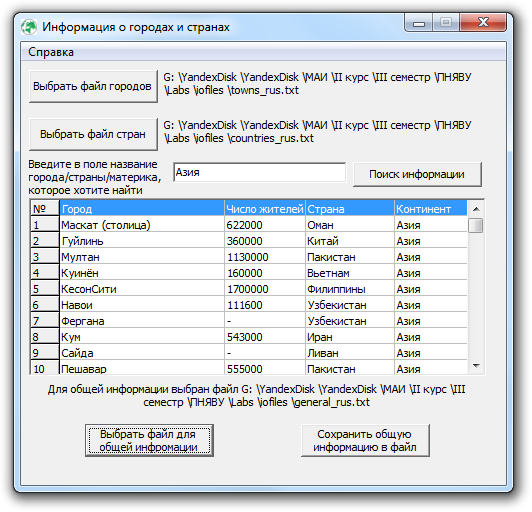


Рис. 14  
Файл для общей информации выбран

Операции поиска и сохранения информации пользователь может делать в произвольном порядке. В любой момент пользователь может изменить файл для общей информации и файлы городов и стран. После выбора новых файлов городов и/или стран работа программы вернётся в состояние, изображённое на рис. 12.

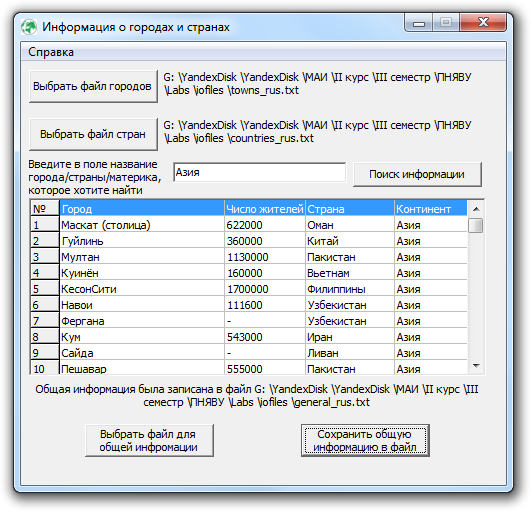


Рис. 15  
Общая информация сохранена в выбранный файл

После сохранение общей информации в выбранный пользователем файл, информация сохраняется в этом файле в виде, изображенном на рис. 16.

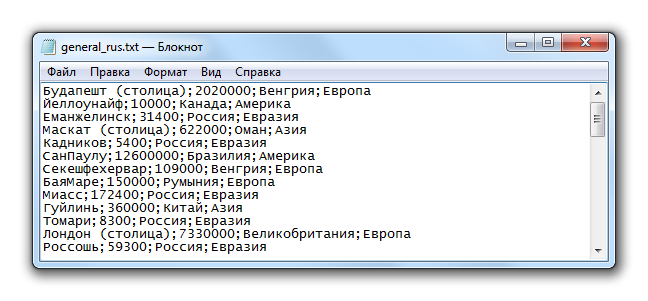


Рис. 16  
Общая информация в выбранном пользователем файле

В меню «Справка» можно узнать сведения о приложении (Рис. 17).

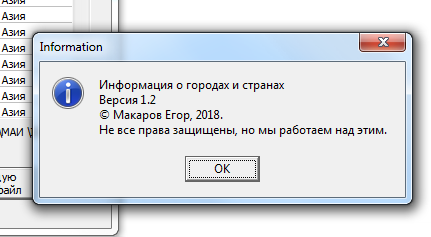


Рис. 17  
Меню «Справка» - «О программе»

# Листинг программной разработки

Файл UnitForm.pas

unit UnitForm;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Menus, Grids;

type

// Структура информации о городах

townsInfo = record

Town: string;

People: string;

Country: string;

end;

// Структура информации о странах

countriesInfo = record

Country: string;

Capital: string;

Continent: string;

end;

// Структура общей информации о городах и странах

generalInfo = record

Town: string;

People: string;

Country: string;

Continent: string;

end;

TMainForm = class(TForm)

buttonTownsFileChoose: TButton;

buttonCountriesFileChoose: TButton;

labelTownsFileChoose: TLabel;

labelCountriesFileChoose: TLabel;

labelRequestInfo: TLabel;

textboxRequestField: TEdit;

buttonGeneralInfoFileChoose: TButton;

labelGeneralInfoFileChoose: TLabel;

buttonGeneralInfoWrite: TButton;

buttonRequestInfoFind: TButton;

MainMenu: TMainMenu;

menuHelp: TMenuItem;

menuAbout: TMenuItem;

stringGridRequestInfo: TStringGrid;

procedure textboxRequestFieldKeyPress(Sender: TObject; var Key: Char);

procedure buttonGeneralInfoWriteClick(Sender: TObject);

procedure menuAboutClick(Sender: TObject);

procedure buttonRequestInfoFindClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure buttonGeneralInfoFileChooseClick(Sender: TObject);

procedure buttonTownsFileChooseClick(Sender: TObject);

procedure buttonCountriesFileChooseClick(Sender: TObject);

private // Вспомогательные процедуры

// Разделяет строки на подстроки по разделителю

procedure Split(Delimiter: Char; Str: string; ListOfStrings: TStrings);

// Печать строки таблицы с общей информации на экран

procedure PrintInfoLine(k: integer; i: integer);

// Активация/дизактивация элементов поиска и печати

procedure ChangeEnable(enable: Boolean);

// Очищает таблицу с информацией

procedure ClearStringGrid(StringGrid: TStringGrid);

// Активация элементов поиска и печати

procedure ShowSearch();

// Добавление текста с файлом в текст label'а

procedure SetFileNameToLabel(fileName: string; fileLabel: TLabel);

// Объединение (слияние) информации из двух файлов

procedure MergeFiles();

// Действия после выбора обеих файлов исходных данных

procedure EventsAfterChoose(secondFile: string);

public

{ Public declarations }

end;

var

MainForm: TMainForm;

townsFileName, countriesFileName, generalInfoFileName: string; // Имена файлов с данными

openFileDialog: TOpenDialog; // Диалоговое окно выбора файла

towns: array of townsInfo; // Информация из файла городов

countries: array of countriesInfo; // Информация из файла стран

towns\_N,countries\_N, general\_N: integer; // Число записей о городах, странах и общей

generalInfoBase: array of generalInfo; // Общая ниформация из файлов

implementation

{$R \*.dfm}

// Разделяет строки на подстроки по разделителю

procedure TMainForm.Split(Delimiter: Char; Str: string; ListOfStrings: TStrings);

begin

ListOfStrings.Clear;

ListOfStrings.Delimiter := Delimiter;

ListOfStrings.StrictDelimiter := True;

ListOfStrings.DelimitedText := Str;

end;

// Печать строки таблицы с общей информации на экран

procedure TMainForm.PrintInfoLine(k: integer; i: integer);

begin

// k - порядковый номер в таблице (для пользователя)

// i - порядковый номер в generalInfoBase

with MainForm.stringGridRequestInfo do

begin

Cells[0,k] := IntToStr(k);

Cells[1,k] := generalInfoBase[i].Town;

Cells[2,k] := generalInfoBase[i].People;

Cells[3,k] := generalInfoBase[i].Country;

Cells[4,k] := generalInfoBase[i].Continent;

end;

end;

// Активация/дизактивация элементов поиска и печати

procedure TMainForm.ChangeEnable(enable: Boolean);

begin

// Элементы поиска информации по запросу

MainForm.labelRequestInfo.Enabled := enable;

MainForm.textboxRequestField.Enabled := enable;

MainForm.buttonRequestInfoFind.Enabled := enable;

MainForm.stringGridRequestInfo.Enabled := enable;

// Элементы печати общей информации в файл

MainForm.buttonGeneralInfoFileChoose.Enabled := enable;

MainForm.labelGeneralInfoFileChoose.Enabled := enable;

MainForm.buttonGeneralInfoWrite.Enabled := enable;

end;

// Очищает таблицу с информацией

procedure TMainForm.ClearStringGrid(StringGrid: TStringGrid);

var

i: integer;

begin

with StringGrid do

begin

for i := 0 to (ColCount - 1) do

Cols[i].Clear;

// Шапка таблицы

Cells[0,0] := '№';

Cells[1,0] := 'Город';

Cells[2,0] := 'Число жителей';

Cells[3,0] := 'Страна';

Cells[4,0] := 'Континент';

end;

end;

// Активация элементов поиска и печати

procedure TMainForm.ShowSearch();

begin

ChangeEnable(true);

MainForm.buttonGeneralInfoWrite.Enabled := false;

end;

// Добавление текста с файлом в текст label'а

// с пробелом для вместимости в форму

procedure TMainForm.SetFileNameToLabel(fileName: string; fileLabel: TLabel);

var i: integer; // Переменная цикла

OutPutList: TStringList; // Промежуточный список подстрок

begin

OutPutList := TStringList.Create;

fileLabel.Caption := '';

Split('\', fileName, OutPutList);

for i := 0 to OutPutList.Count - 1 do

begin

fileLabel.Caption := fileLabel.Caption + OutPutList[i];

if (i <> (OutPutList.Count - 1)) then

fileLabel.Caption := fileLabel.Caption + ' \';

end;

// Добавление пробелов необходимо, чтобы

// разделить путь к файлу на слова.

// После этого перенос пути файла в label'е, если

// он в него не влезает, возможен

end;

// Объединение (слияние) информации из двух файлов

procedure TMainForm.MergeFiles();

var

i,j,k: integer; // Переменные циклов

flag: boolean; // =false, если нет ни одного города из страны/нет страны для города

begin

k := 0; // Число записей

// Слияние информации

for i := 0 to towns\_N - 1 do // по городам

begin

flag := false;

for j := 0 to countries\_N - 1 do // по странам

begin

if ((towns[i].Country = countries[j].Country)) then

begin

flag := true;

SetLength(generalInfoBase,k+1);

generalInfoBase[k].Town := towns[i].Town;

generalInfoBase[k].People := towns[i].People;

generalInfoBase[k].Country := countries[j].Country;

generalInfoBase[k].Continent := countries[j].Continent;

if (Pos(towns[i].Town, countries[j].Capital) = 1) then

generalInfoBase[k].Town := towns[i].Town + ' (столица)';

// Печать общей информации на экран

PrintInfoLine(k+1,k);

inc(k);

end;

end; // j

// Нет страны для города i

if not(flag) then

begin

SetLength(generalInfoBase,k+1);

generalInfoBase[k].Town := towns[i].Town;

generalInfoBase[k].People := towns[i].People;

generalInfoBase[k].Country := towns[i].Country;

generalInfoBase[k].Continent := '-';

// Печать информации на экран

PrintInfoLine(k+1,k);

inc(k);

end;

end; // i

// Нет ни одного города из страны j

for j := 0 to countries\_N - 1 do

begin

flag := false;

for i := 0 to towns\_N - 1 do

begin

if ((towns[i].Country = countries[j].Country)) then

begin

// Встречен город i из страны j

flag := true;

break;

end;

end; // i

// Запись страны j без городов

if not(flag) then

begin

SetLength(generalInfoBase,k+1);

generalInfoBase[k].Town := '-';

generalInfoBase[k].People := '-';

generalInfoBase[k].Country := countries[j].Country;

generalInfoBase[k].Continent := countries[j].Continent;

// Печать информации на экран

PrintInfoLine(k+1,k);

inc(k);

end;

end; // j

// В случае отсутствия общей информцаии

if (k = 0) then

begin

MessageDlg('В выбранных вами файлах нет общей информации!', mtError, [mbOk], 0);

// Программа в состоянии после выбора файла городов

ChangeEnable(false);

countriesFileName := '';

MainForm.labelCountriesFileChoose.Caption := 'Файл не выбран';

Exit;

end;

general\_N := k;

MainForm.stringGridRequestInfo.RowCount := general\_N;

end;

// Действия после выбора файлов исходных данных

procedure TMainForm.EventsAfterChoose(secondFile: string);

begin

// начальный вид таблицы и label'а под ним

MainForm.labelGeneralInfoFileChoose.Caption := 'Файл для общей информации не выбран';

MainForm.textboxRequestField.Text := '';

ClearStringGrid(MainForm.stringGridRequestInfo);

if (secondFile <> '') then

begin

ShowSearch(); // Отобразить поиск

MergeFiles(); // Объединить информацию из файлов

end;

end;

// Конец вспомогательных процедур

// Начало обработчиков событий

// Выбор файла городов и чтение информации о странах

procedure TMainForm.buttonTownsFileChooseClick(Sender: TObject);

var

f:TextFile; // Переменная для файла

k: integer; // Счётчик городов

OutPutList: TStringList; // Промежуточный список подстрок

s: string; // Промежуточная строка с содержимым строки файла

begin

// Описание работы диалогового окна

openFileDialog.Title := 'Выбор файла городов';

if openFileDialog.Execute then

// Файл уже используется в программе

if (openFileDialog.FileName = countriesFileName) then

MessageDlg('Файл ' + openFileDialog.FileName + ' уже выбран как файл стран', mtError, [mbOk], 0)

else if (openFileDialog.FileName = generalInfoFileName) then

MessageDlg('Файл ' + openFileDialog.FileName

+ ' уже выбран как файл для общей информации', mtError, [mbOk], 0)

else

// Файл ещё не используется

begin

townsFileName := openFileDialog.FileName;

// Чтение информации о городах

OutPutList := TStringList.Create;

AssignFile(f, townsFileName);

Reset(f);

// Файл пуст

if (Eof(f)) then

begin

MessageDlg('Файл ' + townsFileName+ ' не содержит информацию о городах.', mtError, [mbOk], 0);

townsFileName := '';

Exit;

end;

k := 0; // Счётчик городов

while (not Eof(f)) do

begin

Readln(f,s);

s := Utf8ToAnsi(s);

Split(';', s, OutPutList);

// Проверка формата информации в файле

if (OutPutList.Count <> 3) then

begin

MessageDlg('Файл ' + townsFileName + ' содержит информацию в некорректном формате.'

+ ' Каждая строка в файле городов должна содержать информацию в виде: '

+ '"город;население;страна"', mtError, [mbOk], 0);

townsFileName := '';

CloseFile(f);

Exit;

end;

// Сохранение информации

SetLength(Towns,k+1);

Towns[k].Town := OutPutList[0];

Towns[k].People := OutPutList[1];

Towns[k].Country := OutPutList[2];

inc(k);

end;

CloseFile(f);

towns\_N := k;

SetFileNameToLabel(townsFileName, MainForm.labelTownsFileChoose);

EventsAfterChoose(countriesFileName);

end;

end;

// Выбор файла стран и чтение информации о странах

procedure TMainForm.buttonCountriesFileChooseClick(Sender: TObject);

var

f:TextFile; // Переменная для файла

k: integer; // Счётчик стран

OutPutList: TStringList; // Промежуточный список подстрок

s: string; // Промежуточная строка

begin

// Описание работы диалогового окна

openFileDialog.Title := 'Выбор файла стран';

if openFileDialog.Execute then

// Файл уже используется в программе

if (openFileDialog.FileName = townsFileName) then

MessageDlg('Файл ' + openFileDialog.FileName + ' уже выбран как файл городов', mtError, [mbOk], 0)

else if (openFileDialog.FileName = generalInfoFileName) then

MessageDlg('Файл ' + openFileDialog.FileName + ' уже выбран как файл для общей информации', mtError, [mbOk], 0)

else

// Файл ещё нигде не используется

begin

countriesFileName := openFileDialog.FileName;

// Чтение информации о странах

OutPutList := TStringList.Create;

AssignFile(f, countriesFileName);

Reset(f);

// Файл пуст

if (Eof(f)) then

begin

MessageDlg('Файл ' + countriesFileName + ' не содержит информацию о странах.', mtError, [mbOk], 0);

countriesFileName := '';

Exit;

end;

k := 0; // Счётчик стран

while (not Eof(f)) do

begin

Readln(f,s);

s := Utf8ToAnsi(s);

Split(';', s, OutPutList);

// Проверка формата информации в файле

if (OutPutList.Count <> 3) then

begin

MessageDlg('Файл ' + countriesFileName + ' содержит информацию в некорректном формате.'

+ ' Каждая строка в файле стран должна содержать информацию в виде: '

+ '"страна;столица;континент"', mtError, [mbOk], 0);

countriesFileName := '';

CloseFile(f);

Exit;

end;

// Сохранение информации

SetLength(Countries,k+1);

Countries[k].Country := OutPutList[0];

Countries[k].Capital := OutPutList[1];

Countries[k].Continent := OutPutList[2];

inc(k);

end;

CloseFile(f);

countries\_N := k;

SetFileNameToLabel(countriesFileName, MainForm.labelCountriesFileChoose);

EventsAfterChoose(townsFileName);

end;

end;

// Выбор файла для общей информации

procedure TMainForm.buttonGeneralInfoFileChooseClick(Sender: TObject);

begin

// Описание работы диалогового окна

openFileDialog.Title := 'Выбор файла для общей информации';

if openFileDialog.Execute then

if (openFileDialog.FileName = townsFileName) then

MessageDlg('Файл ' + openFileDialog.FileName + ' уже выбран как файл городов', mtError, [mbOk], 0)

else if (openFileDialog.FileName = countriesFileName) then

MessageDlg('Файл ' + openFileDialog.FileName + ' уже выбран как файл стран', mtError, [mbOk], 0)

else

begin // Файл ещё нигде не используется

generalInfoFileName := openFileDialog.FileName;

SetFileNameToLabel('Для общей информации выбран файл ' + generalInfoFileName, MainForm.labelGeneralInfoFileChoose);

MainForm.buttonGeneralInfoWrite.Enabled := true;

end;

end;

// Запись общей информации в выбранный для неё файл

procedure TMainForm.buttonGeneralInfoWriteClick(Sender: TObject);

var

f:TextFile; // Переменная для файлов исходных данных

i: integer; // Переменная цикла

begin

AssignFile(f, generalInfoFileName);

Rewrite(f);

for i := 0 to general\_N - 1 do

with generalInfoBase[i] do

Writeln(f,Town + ';' + People + ';' + Country + ';' + Continent);

CloseFile(f);

SetFileNameToLabel('Общая информация была записана в файл ' + generalInfoFileName,MainForm.labelGeneralInfoFileChoose);

end;

// Поиск информации по запросу и печать её на экране

procedure TMainForm.buttonRequestInfoFindClick(Sender: TObject);

var

i: integer; // Переменная цикла

k: integer; // Счётчик напечатанных строк с информацией

s: string; // Текст из TextBox (Edit)

begin

s := LowerCase(textboxRequestField.Text);

if (s <> '') then

begin

ClearStringGrid(MainForm.stringGridRequestInfo);

k := 0;

// Печать всего на экран

if ((s = 'всё') or (s = 'все')) then

begin

for i := 0 to general\_N - 1 do

begin

PrintInfoLine(i+1,i);

end;

end //всё

else begin // запрос

for i := 0 to general\_N - 1 do

begin

with generalInfoBase[i] do

if ((Pos(s, LowerCase(Town)) + Pos(s, LowerCase(People))

+ Pos(s, LowerCase(Country)) + Pos(s, LowerCase(Continent)))>0)then

begin

PrintInfoLine(k+1,i);

inc(k);

end;

end;//i

// Ничего по запросу не найдено

if (k = 0) then

begin

MessageDlg('По запросу "' + s + '" ничего не найдено', mtError, [mbOk], 0);

Exit;

end;

end;//запрос

end;//''

end;

// Запрет на ввод некорректных символов в TextBox (Edit)

procedure TMainForm.textboxRequestFieldKeyPress(Sender: TObject; var Key: Char);

begin

// Список запретных для ввода символов

if (Key in [';', '<', '>', '/', '?', '/', '\', '|',

'=', ' ', ':', '.', ',', #13, '!']) then

Key := #0;

end;

// Форма создана

procedure TMainForm.FormCreate(Sender: TObject);

var

myDir: string; // Путь к папке с файлами (городов, стран, с общей информацией)

w: integer; // Параметр толщины одной буквы в строке (в пикселях)

begin

// Дизактивация элементов на форме в начале работы

ChangeEnable(false);

townsFileName := '';

countriesFileName := '';

generalInfoFileName := '';

// Установка пути к папке с файлами

//myDir := GetCurrentDir;

myDir := '..\iofiles';

// Описание диалогового окна выбора файлов

openFileDialog := TOpenDialog.Create(self);

openFileDialog.InitialDir := myDir;

openFileDialog.Options := [ofReadOnly, ofFileMustExist];

openFileDialog.Filter := 'txt, csv files|\*.txt; \*.csv';

// Параметры таблицы с информацией

ClearStringGrid(stringGridRequestInfo);

with stringGridRequestInfo do

begin

ColCount := 5;

DefaultRowHeight := 15;

w := 4;

// Толщина столбца = w \* число символов в ячейке столбца

ColWidths[0] := w\*7; // Номер строки

ColWidths[2] := w\*20; // Число жителей

ColWidths[3] := w\*22; // Страна

ColWidths[4] := w\*18; // Континент

// Город = толщина всей таблицы - толщина остальных столбцов

ColWidths[1] := Width - 25 - ColWidths[0] - ColWidths[2] - ColWidths[3] - ColWidths[4];

end;

end;

// Окно с информацией о программе

procedure TMainForm.menuAboutClick(Sender: TObject);

begin

MessageDlg('Информация о городах и странах' + #13#10 + 'Версия 1.2' + #13#10

+ '© Макаров Егор, 2018.' + #13#10 + 'Не все права защищены, '

+ 'но мы работаем над этим.',mtInformation,[mbOk],0);

end;

end.

Файл LABA\_1\_2.dpr

program LABA\_1\_2;

uses

Forms,

UnitForm in 'UnitForm.pas' {MainForm};

{$R \*.res}

begin

Application.Initialize;

Application.CreateForm(TMainForm, MainForm);

Application.Run;

end.

# Результаты тестирования

Тест №1

Цель теста: проверить работу программы в случае работы пользователя согласно инструкции пользователя. Все исходные и выходные данные имеют допустимый формат и не содержат ошибок, диагностические сообщения об ошибках не возникают в процессе работы программы.

Входные данные: в качестве файла с информацией о городах пользователь выбирает файл с информацией на рис. 18, в качестве файла с информацией о странах – файл с информацией на рис. 19.

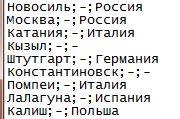
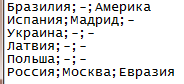
 

Рис. 19  
Содержимое файла стран

Рис. 18  
Содержимое файла городов

Можно заметить, что в файле стран нет информации о странах Германия и Италия, а информация о городах из этих стран имеется в файле городов. В файле городов имеются города (Кызыл, Константиновск), у которых неизвестна страна. Также в файле стран есть информация о странах (Бразилия, Латвия), о городах из которых нет информации в файле городов.

Ожидаемый результат: Будет сформирован файл с общей информацией. Общей информацией для обоих файлов является страна, поэтому для городов, у которых известна страна, и информация о ней есть в файле стран, общая информация будет сформирована на основе обоих файлов. Города, являющиеся столицами стран, у которых информация о столице известна, будут помечены как столичные (к их названиям будет приписка «Столица»). Для городов, у которых страна неизвестна, в общей информации поля «Страна» и «Континент» останутся не заполненными, остальная информация будет взята только из файла городов. Для городов, у которых о стране нет информации в файле стран, общая информация будет взята только из файла городов, а поле «Континент» останется пустым. Для стран, информации о городах из которых нет в файле городов, будет сформирована общая информация на основе информации только из файла стран, поля «Город» и «Население» останутся не заполненными.

Полученный результат: файлы, описанные в предыдущих пунктах, успешно выбраны пользователем, его работа не вызвала ошибок в работе программы, сообщений об ошибках также не было в процессе работы. В окне приложения в форме таблицы отображена общая информация после слияния информации из выбранных файлов. Пользователь выбирает файл для записи общей информации и нажимает кнопку «Сохранить общую информацию». В окне приложения написано, что информация успешно сохранена в выбранный пользователем файл. На рис. 20 изображено содержимое файла с общей информацией.

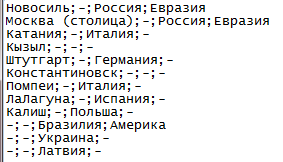


Рис. 20  
Содержимое файла городов

Вывод: полученный результат совпал с ожидаемым. Ошибок в работе приложения не обнаружено.

Тест №2

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

Входные данные: в диалоговом окне выбора файла городов/стран/файла для общей информации введено имя несуществующего файла (Рис. 21)

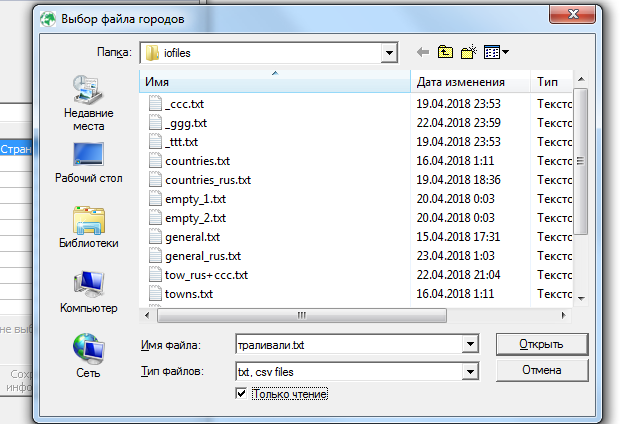


Рис. 21  
В диалоговом окне введено имя несуществующего файла

Полученный результат: соответствующее сообщение об ошибке (Рис 22).

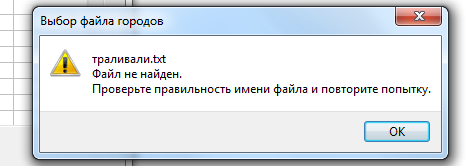


Рис. 22  
Сообщение о том, что выбранный файл не найден

Вывод: ошибок в работе программы не обнаружено.

Тест №3

Цель теста: проверить работу программы в случае выбора файла с информацией о городах/странах/файла для общей информации, уже выбранного как файл для общей информации/файл с информацией о городах/странах.

Входные данные: в диалоговом окне выбора файла выбирается файл, уже выбранный как файл с информацией о городах/странах/файл для общей информации.

Полученный результат: соответствующие сообщения об ошибках (Рис. 23, 24, 25). Файл городов/стран/для общей информации остаётся не выбран.

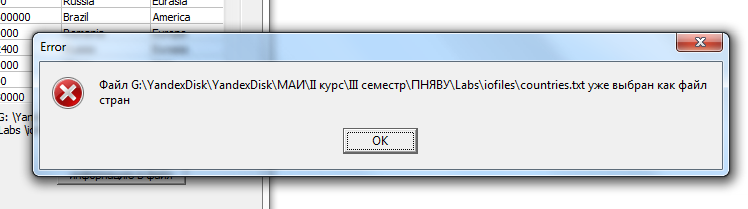


Рис. 23  
Сообщение о том, что файл уже выбран как файл стран

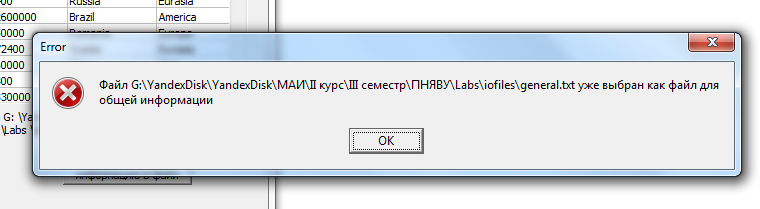


Рис. 24  
Сообщение о том, что файл уже выбран как файл для общей информации

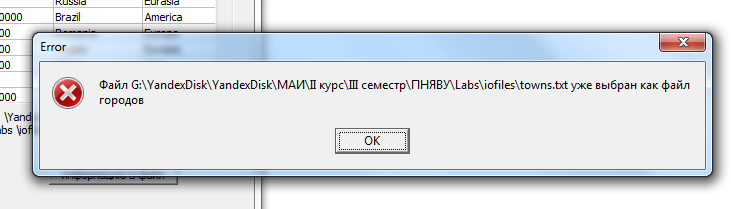


Рис. 25  
Сообщение о том, что файл уже выбран как файл городов

Вывод: ошибок в работе программы не обнаружено.

Тест №4

Цель теста: проверить работу программы в случае выбора файла, который не содержит информацию о городах/странах (выбранный файл пуст).

Входные данные: в качестве файла с информацией о городах/странах выбирается пустой файл.

Полученный результат: соответствующие сообщения об ошибках (Рис. 26, 27). Файл городов/стран остаётся не выбран.

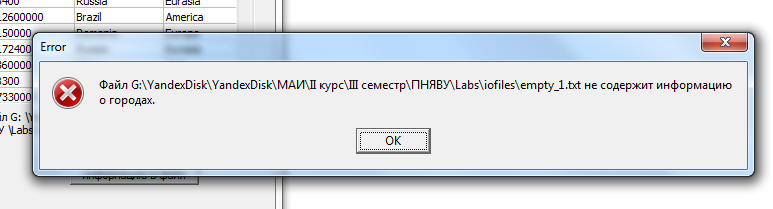
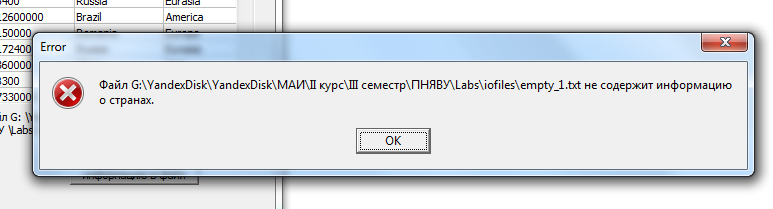


Рис. 27  
Сообщение о том, что файл пуст (для выбора файла с информацией о странах)

Рис. 26  
Сообщение о том, что файл пуст (для выбора файла с информацией о городах)



Вывод: ошибок в работе программы не обнаружено.

Тест №5

Цель теста: проверить работу программы в случае выбора файла информацией о городах/странах, который содержит информацию в некорректном виде.

Входные данные: выбирается файл исходных данных, в котором каждая строка с информацией не представлена в виде «город;население;страна» (для городов) или «страна;столица;континент» (для стран).

Полученный результат: соответствующие сообщения об ошибках с описанием корректного представления информации о городах/странах (Рис. 28, 29).

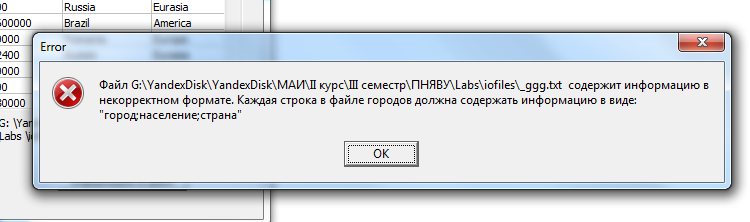


Рис. 28  
Сообщение о некорректном формате данных в файле городов с описанием корректного формата

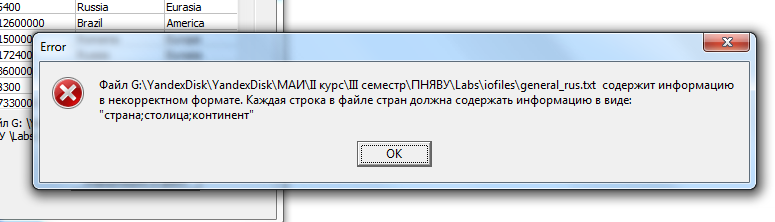


Рис. 29  
Сообщение о некорректном формате данных в файле стран с описанием корректного формата

Вывод: ошибок в работе программы не обнаружено.

Тест №6

Цель теста: проверить работу программы в случае выбора файлов с информацией о городах и странах, в которых нет общей информации.

Входные данные: в качестве файлов с информацией о городах выбран файл, в котором описанные города находятся в странах, информации о которых нет в выбранном файле с информацией о странах.

Полученный результат: соответствующее сообщение пользователю (Рис. 30). Приложение возвращается в состояние, в котором файл городов выбран, а файл стран – нет (Рис. 31).

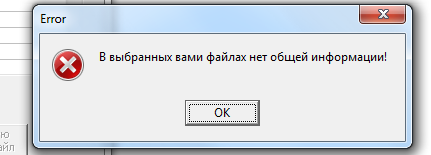


Рис. 30  
Сообщение об отсутствии общей информации в выбранных пользователем файлах.

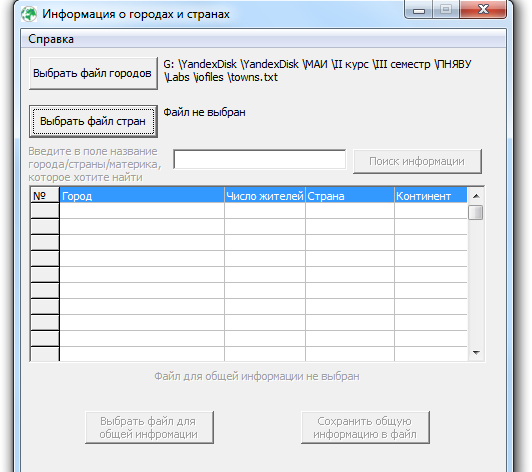


Рис. 31  
Состояние приложения после выбора файлов без общей информации.

Вывод: ошибок в работе программы не обнаружено.

Тест №7

Цель теста: проверить работу программы в случае отсутствия информации, удовлетворяющей поисковому запросу пользователя.

Входные данные: пользователь вводит в поле для запроса название, которое отсутствует среди общей информации о городах и странах (Рис. 32).

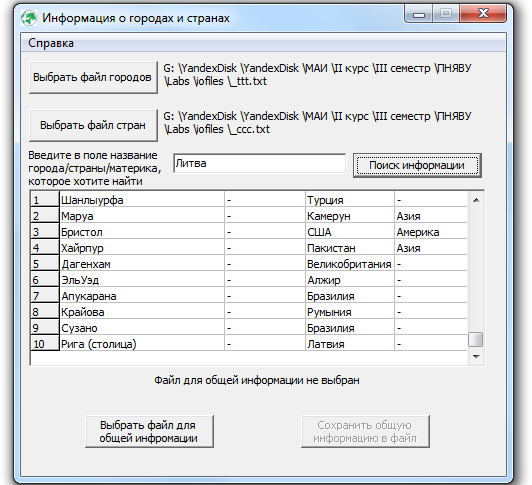
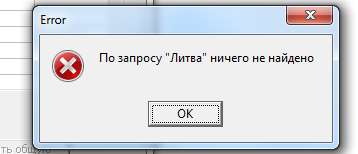


Рис. 33  
Информация по запросу пользователя не найдена.

Рис. 32  
В поле запроса введено название, отсутствующее среди общей информации.

Полученный результат: соответствующее сообщение об ошибке (Рис. 33), таблица с информацией становится пустой.



Вывод: ошибок в работе программы не обнаружено.