

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Пензенский государственный университет»

Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по курсу «Программирование»

на тему «Географическая база данных»

ОТ
20.05.26
Чураева

Выполнил:

студент группы 25BBB2

Чураева А. А.

Принял:

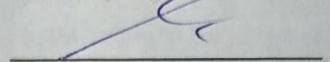
к.т.н., доцент Генералова А.А.

Пенза 2026

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники
Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

 М.А. Митрохин

«11» февраля 2026г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Чурской Алены Алексеевны Группа 25 ВВВ2

Тема проекта: **Разработка программы сложной структуры методом нисходящего программирования.**

Исходные данные (технические требования) на проектирование

*Программа "Географическая база данных"
Добавление, удаление, редактирование, поиск, сортировка записей;
Фильтрация записей по определенному критерию;
Создание новой базы данных; Сохранение текущей базы данных;
Загрузка сохраненных баз данных.*

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio.

Объем работы по курсу

1. Расчетная часть

Разработка программы.

2. Графическая часть

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания " 11 " февраля 2026г.

Дата защиты проекта " " г.

Руководитель Т. Генералов А. А.

Задание получил " 11 " февраля 2026г.

Студент Иванов И. И.

Содержание

Введение	5
Постановка задачи	6
Выбор решения	8
<i>Определение необходимых модулей программы</i>	<i>8</i>
<i>Определение структуры базы данных</i>	<i>9</i>
Описание разработки программы	10
Отладка и тестирование.....	12
Описание программы.....	13
Руководство пользователя	27
Заключение	36
Список литературы.....	37
Приложение А	38
Листинги программы	38
<i>Курсовая.cpp.....</i>	<i>38</i>
<i>Functions.h.....</i>	<i>43</i>
<i>Functions.cpp</i>	<i>45</i>

Введение

База данных - организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

База данных представляет собой упорядоченную информацию, объединенную в единое целое (например: список телефонных номеров, товаров, мест, работников и т.д.).

Актуальность и востребованность использования баз данных обусловлена необходимостью автоматизации учета и повышения эффективности оборота информацией в современных реалиях. В условиях роста объемов обрабатываемой информации использование бумажных носителей или таблиц Excel становится неэффективным: возникают временные задержки в поиске и появляется высокий риск ошибок и неточностей.

В наше время базы данных являются неотъемлемой частью любой организации, любого предприятия. Сайты используют базу данных чтобы хранить данные пользователя. Банки хранят в огромных базах данных данные о своих клиентах и их счетах.

Создавая базу данных, программист стремится упорядочить информацию по каким-либо критериям и быстро производить выборку с определенным сочетанием признаков. А сделать это можно, только если данные грамотно структурированы.

В процессе выполнения курсовой работы была разработана программа на языке C++ и Ассемблере, позволяющая выполнять работу (поиск, сортировка, фильтрация, добавление и удаление записей и т.д.) с географической базой данных.

Постановка задачи

Необходимо разработать программу «Географическая база данных». Информация в базе данных структурирована по определенным признакам: название страны, столица страны, континент на котором расположена страна, численность населения страны, площадь страны и является ли государство островным.

Программа должна предоставить возможности: выхода из программы, ввода новых данных о странах, удаления данных о стране, изменения данных о стране, сортировки стран по названию, фильтрации стран по континенту, поиска страны по названию, создания новой базы данных в файле, сохранения текущей базы данных в файл, загружать базы данных из файла и показывать всю базу данных.

Для реализации вышеперечисленных требований необходимо разработать интуитивно понятный интерфейс для пользователя, определить названия пунктов меню, определить тип интерфейса, продумать навигацию по базе данных, изучить функции для работы с файлами. Это необходимо для четкого понимания функционала программы.

Многомодульность программы. Необходимо разделить программу на логические модули. Это повысит локализацию ошибок при отладке и тестировании программы.

Необходимо использовать сложные типы данных – массивы, структуры, файлы. Необходимо определиться с элементами управления. Необходимо изучить приемы и механизмы псевдографического интерфейса. Это необходимо для понимания, как и почему на консоли появляются псевдографические символы, линии и т.д.

Устройство ввода информации – клавиатура. Необходимо научиться анализировать и различать события, возникающие от работы с клавиатурой. Необходимо однозначно идентифицировать и выполнять те или иные действия в зависимости от действий пользователя. Это необходимо для легкого использования программы.

Пользовательский интерфейс должен быть построен на основе меню. Необходимо изучить приемы программирования меню, выбрать подходящий вид меню. Это необходимо для создания интуитивно понятного интерфейса.

Заставка необходима для того, чтобы пользователь, запустивший программу, смог получить достаточную информацию о её создателе.

Сохранение в файле создано для удобства, с его помощью пользователь может передавать данные другим лицам.

Выбор решения

Определение необходимых модулей программы

Разработанная программа состоит из модулей, представленных на рисунке 1:



Рисунок 1 – Модули программы

Файлы: «курсовая.cpp», «Functions.h», «Functions.cpp» - необходимы для включения функций обработки данных. Данные файлы содержат функции выхода из программы, ввода новых данных о странах, удаления данных о стране, изменения данных о стране, сортировки стран по названию, фильтрации стран по континенту, поиска страны по названию, создания новой базы данных в файле, сохранения текущей базы данных в файл, загрузки базы данных из файла и показа всей базы данных.

Приложение представляет собой совокупность данных и модулей. Приложение использует данные, содержащиеся в файле «input.txt». Все вышеперечисленное удовлетворяет требованию модульности программы данной курсовой работы. Управление приложением осуществляется с помощью клавиатуры. Программа предназначена для работы с базой данных, база данных может быть взята из файла или введена самостоятельно. Открытие и сохранение выходного файла выполняется средствами данной программы. Программа дает возможность просматривать содержимое базы данных, искать данные, сортировать их, редактировать, удалять или добавлять и т.д. Данные из базы отображаются в программе в виде столбца данных: название страны, столица

страны, континент на котором расположена страна, численность населения страны, площадь страны и является ли государство островным.

Определение структуры базы данных

Для структурирования данных внутри файла выбраны следующие структуры данных:

```
struct Country {  
    string name;  
    string capital;  
    string continent;  
    long long population;  
    double area;  
    bool isIsland;  
};
```

Строки структуры:

- название страны,
- столица страны,
- континент, на котором расположена страна,
- численность населения страны,
- площадь страны,
- является ли государство островным

Все структуры имеют текстовый формат.

Выбранная структура проста и соответствует всем требованиям программы.

Описание разработки программы

Для написания данной программы использован язык программирования C++ и MASM.

В качестве среды программирования был выбран программный продукт Visual Studio 2022.

Сортировка данных в разработанной программе реализуется на языке Assembler.

Сохранение информации реализуется в файл, который пользователь сам создает.

Интерфейс информационной системы представлен в виде меню, т.к. он является наиболее удобным. Исполнение навигации программы по пунктам меню будет осуществляться с помощью клавиатуры, цифрами от 0 до 10.

Разработка программы началась с реализации набора функций, необходимых для работы с географической базой данных.

- Для ввода новых данных о странах и создания динамического массива используется функция `DataEntry()`.
- Для вывода всей базы данных на экран в заданном формате используется функция `Print()`.
- Для редактирования существующих записей по номеру используется функция `DataChange()`.
- Для удаления записей по номеру с перераспределением памяти используется функция `DeleteData()`.
- Для копирования массивов структур (по массивно) используется перегруженная функция `Copy()`, а для копирования отдельных элементов структуры — вторая перегрузка `Copy()`.
- Для сортировки записей по названию страны методом «пузырька» с использованием ассемблерной вставки для сравнения строк с названием используется функция `DataSorting()`.
- Для фильтрации записей по названию континента используется функция `FilterByContinent()`.

- Для поиска страны по её точному названию используется функция `SearchByCountryName()`.
- Для создания новой пустой базы данных используется функция `CreateNewDatabase()`.
- Для сохранения текущей базы данных в текстовый файл используется функция `SaveToFile()`.
- Для загрузки базы данных из текстового файла используется функция `LoadFromFile()`.

По порядку реализации модулей можно понять, что нами использовался нисходящий подход реализации программы. Этот метод основан на том, что сначала проектируются основные компоненты программы, а затем уже дорабатываются ее маленькие детали. Это позволяет упростить ее тестирование, так как каждый модуль тестируется сразу же после его реализации, что помогает нам в создании более структурированных программ.

Отладка и тестирование

В качестве среды разработки была выбрана программа Microsoft Visual Studio 2022. Программа обладает всеми средствами необходимыми при разработке и отладке программы. Для отладки использовались несколько возможностей Visual Studio: точка останова, трассировка, анализ содержимого переменных.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. В ходе тестирования было выявлено и исправлено немало ошибок, связанных с перерисовкой меню, с недопустимостью обработки некоторых пунктов меню и многое другое. Также были добавлены проверки, связанные с недопустимостью обработки некоторых пунктов меню при определенных условиях.

Описание программы

Приложение «Functions.exe» является основным модулем программы. При запуске программы происходит вывод меню, после чего программа готова к выполнению своих функций. Затем, выбрав соответствующую опцию, программа переходит к выполнению соответствующих функций. Выход из меню осуществляется после выбора одного из пунктов меню. Описание всех состояний выполнено в таблице 1. Графическое представление всех состояний показано на рисунке 2.

Таблица 1 — Описание всех состояний программы

Клавиши, вызывающее событие	Действие пользователя	Действие программы
0, Enter	Выбран пункт «Выход из программы»	Производится выход из программы
1, Enter	Выбран пункт «Ввод новых данных о странах»	Выполняется ввод новых данных в новую базу
2, Enter	Выбран пункт «Удаление данных о стране»	Производится удаление данных из базы данных
3, Enter	Выбран пункт «Изменение данных о стране»	Дается возможность изменить данные в базе
4, Enter	Выбран пункт «Сортировка стран по названию»	Дается возможность отсортировать данные по алфавиту
5, Enter	Выбран пункт «Фильтрация стран по континенту»	Дается возможность отфильтровать данные по континенту
6, Enter	Выбран пункт «Поиск страны по названию»	Дается возможность выполнить поиск страны по её названию
7, Enter	Выбран пункт «Создание новой базы данных в файле»	Производится создание базы данных в конкретном файле

Продолжение таблицы 1

8, Enter	Выбран пункт «Сохранение текущей базы данных в файл»	Дается возможность сохранить текущую базу данных в выбранный файл
9, Enter	Выбран пункт «Загрузка базы данных из файла»	Производится загрузка данных из определенного файла
10, Enter	Выбран пункт «Показать всю базу данных»	Дается возможность просмотреть всю базу данных

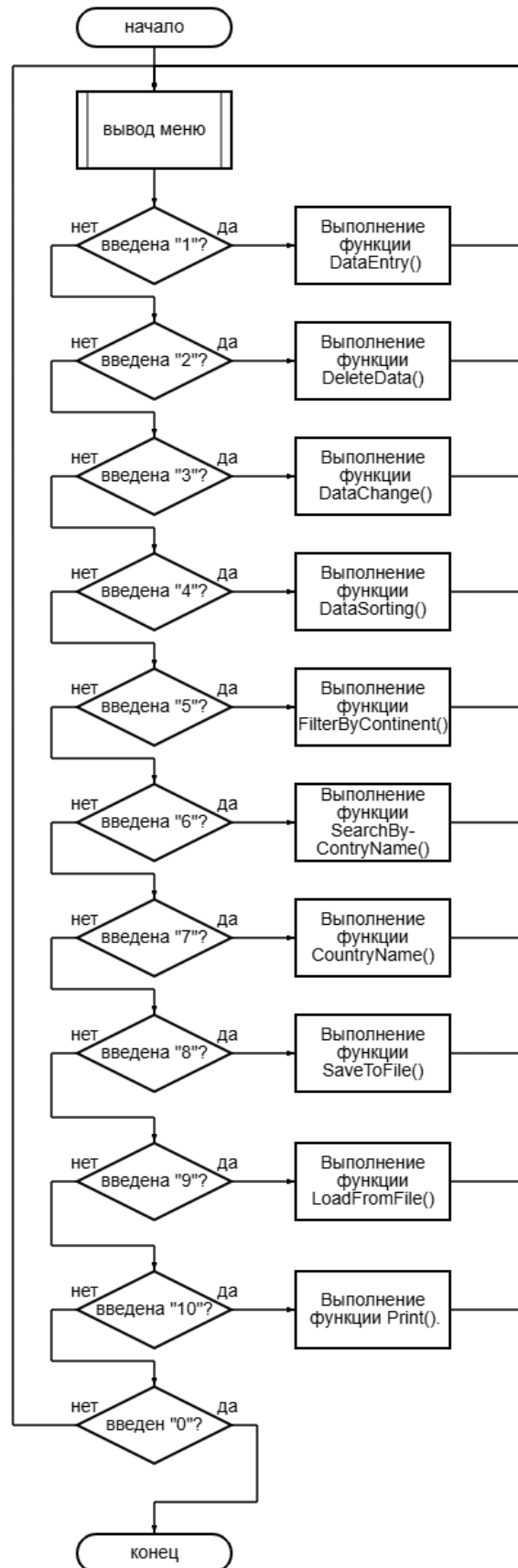


Рисунок 2 – Графическое представление к таблице 1

Для ввода новых данных о странах и создания динамического массива используется функция DataEntry() (рисунок 3).



Рисунок 3 – Блок схема функции DataEntry

Для вывода всей базы данных на экран в заданном формате используется функция Print() (рисунок 4).

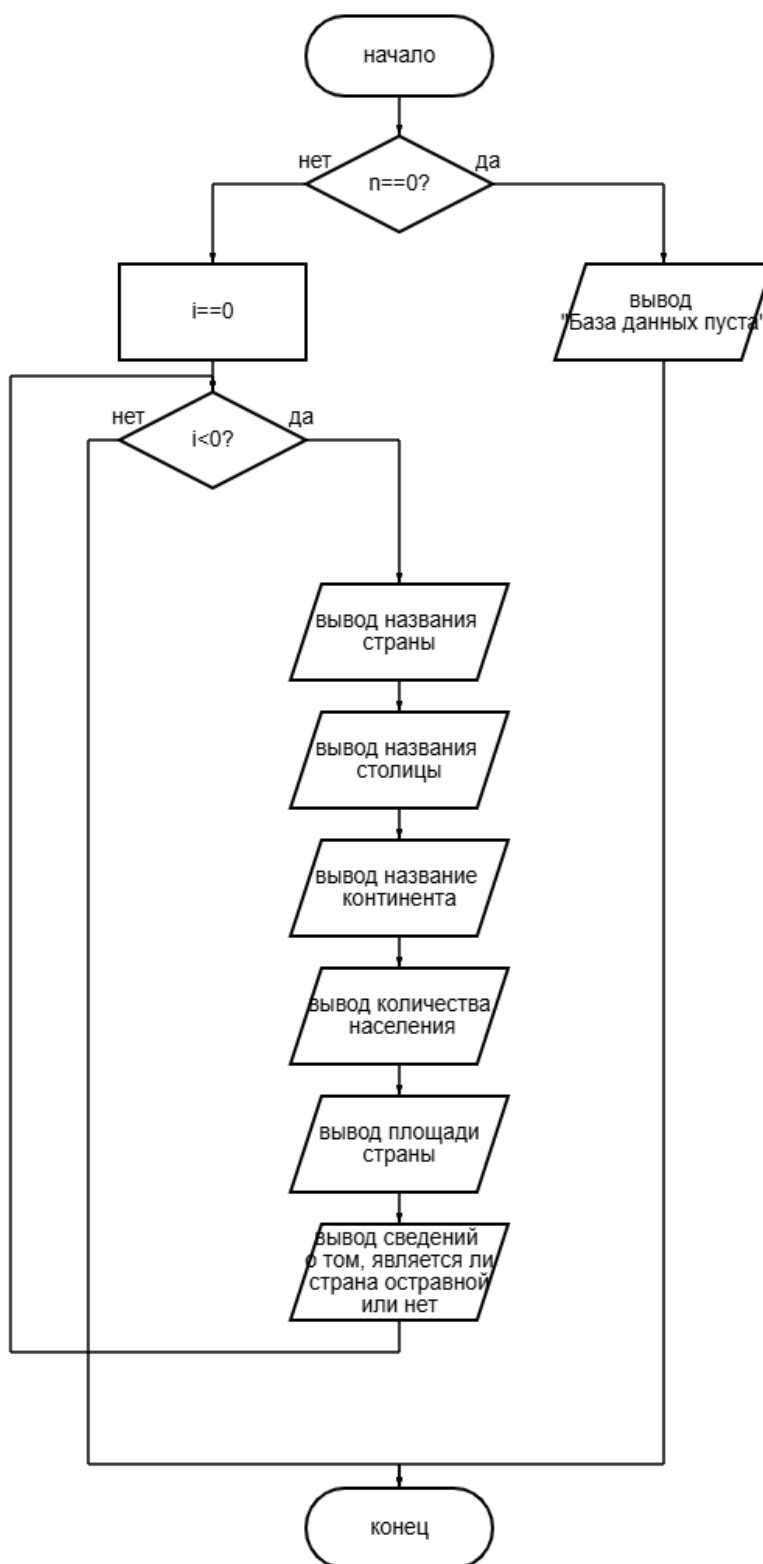


Рисунок 4— Блок схема функции Print

Для редактирования существующих записей по номеру используется функция DataChange() (рисунок 5).

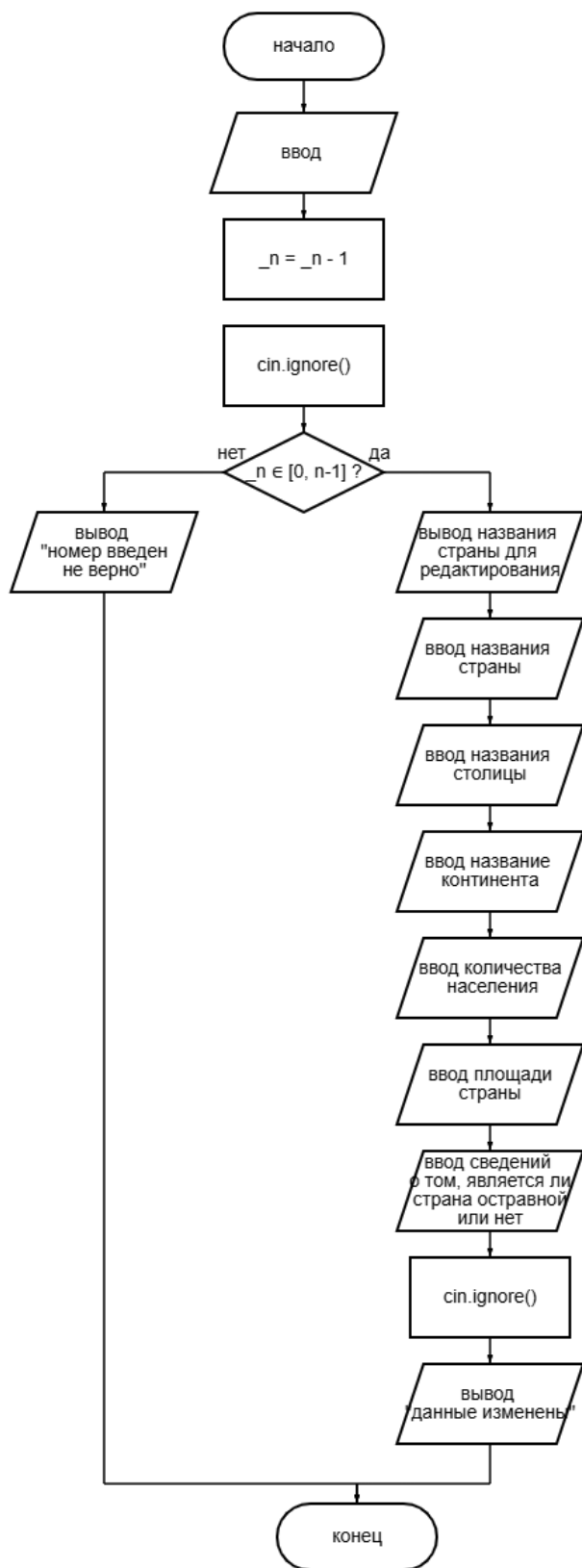


Рисунок 5 – Блок схема функции DataChange

Для удаления записей по номеру с перераспределением памяти используется функция DeleteData() (рисунок 6).

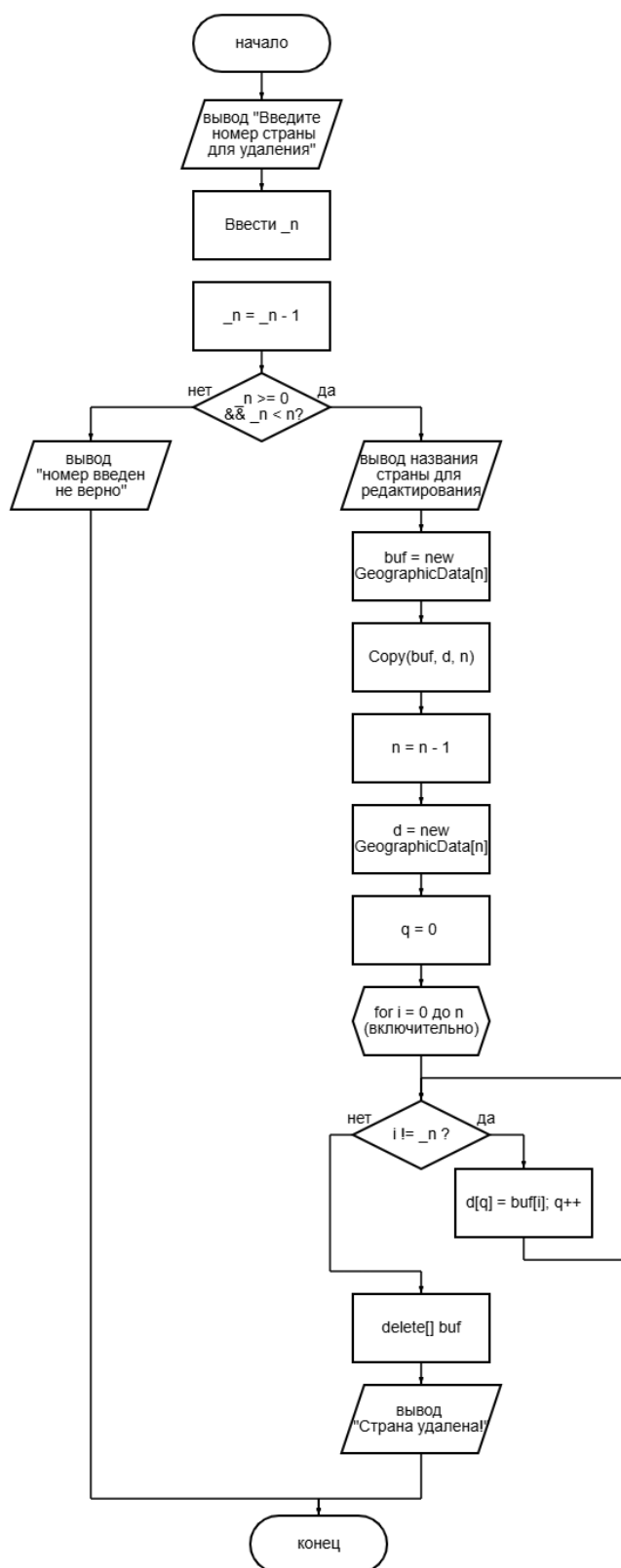


Рисунок 6 – Блок схема функции DeleteData

Для копирования массивов структур (по массивно) используется перегруженная функция `Copy()`, а для копирования отдельных элементов структуры — вторая перегрузка `Copy()` (рисунок 7).

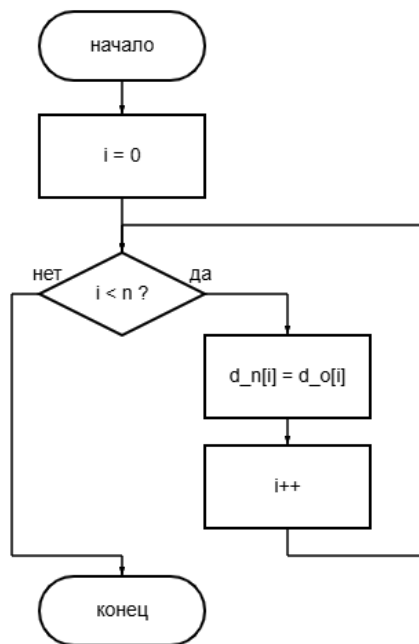


Рисунок 7 – Блок схема функции `Copy`

Для сортировки записей по названию страны методом «пузырька» с использованием ассемблерной вставки для сравнения строк с названием используется функция DataSorting() (рисунок 8).

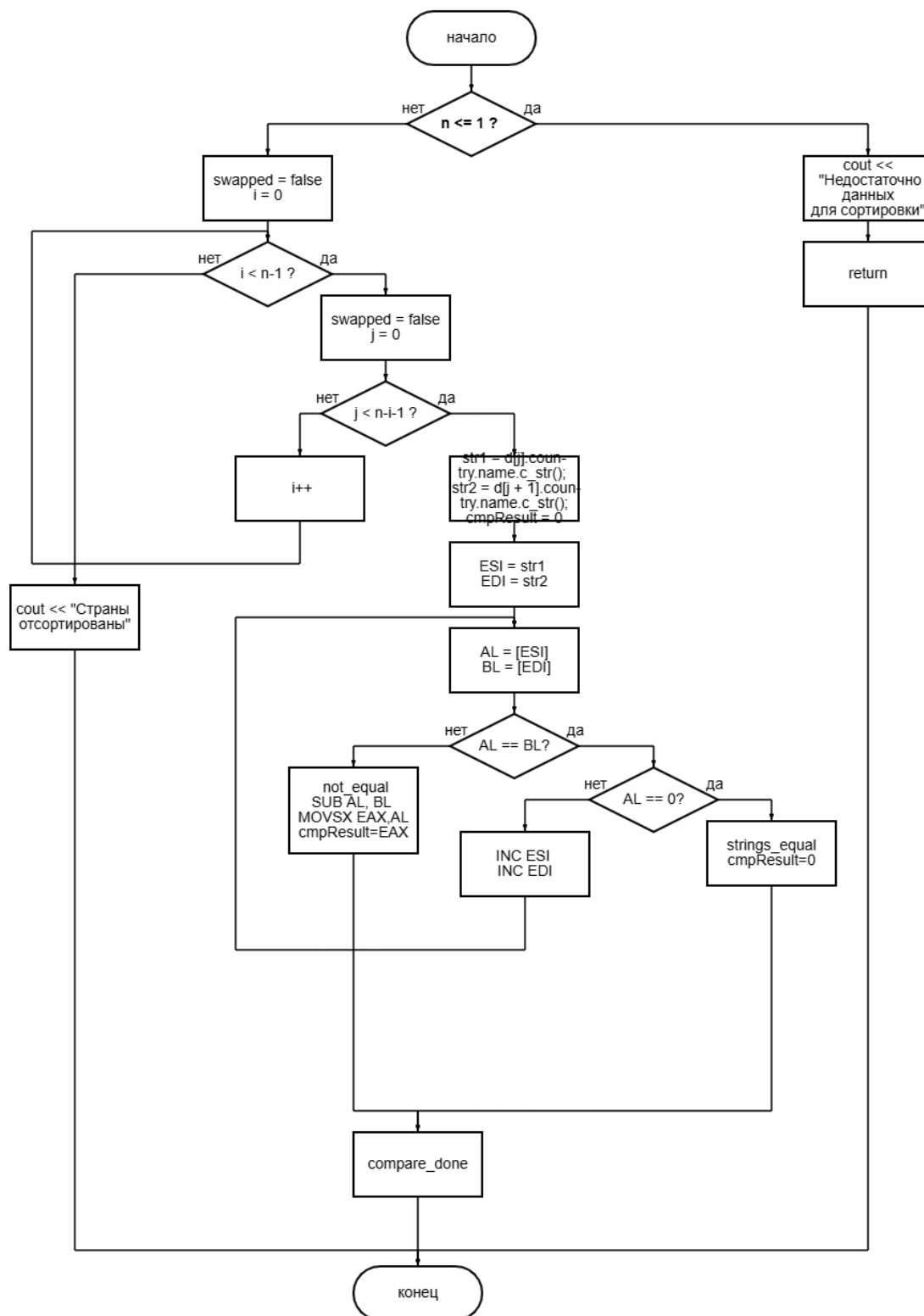


Рисунок 8 – Блок схема функции DataSorting

Для фильтрации записей по названию континента используется функция FilterByContinent() (рисунок 9).

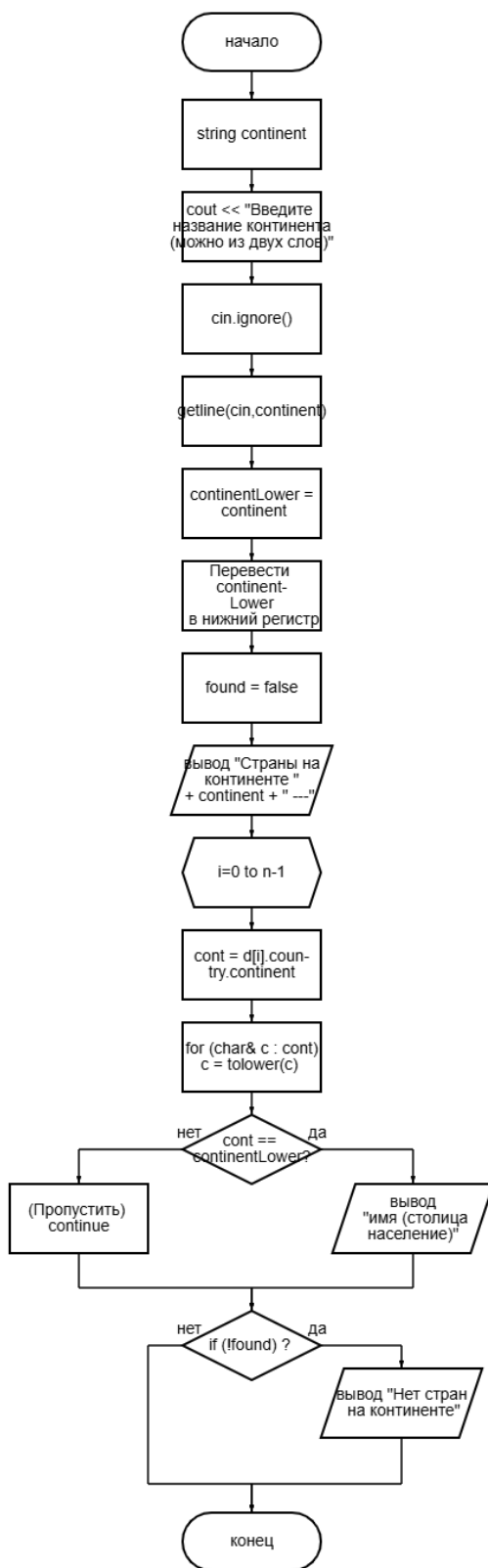


Рисунок 9 – Блок схема функции FilterByContinent

Для поиска страны по её точному названию используется функция SearchByCountryName() (рисунок 10).

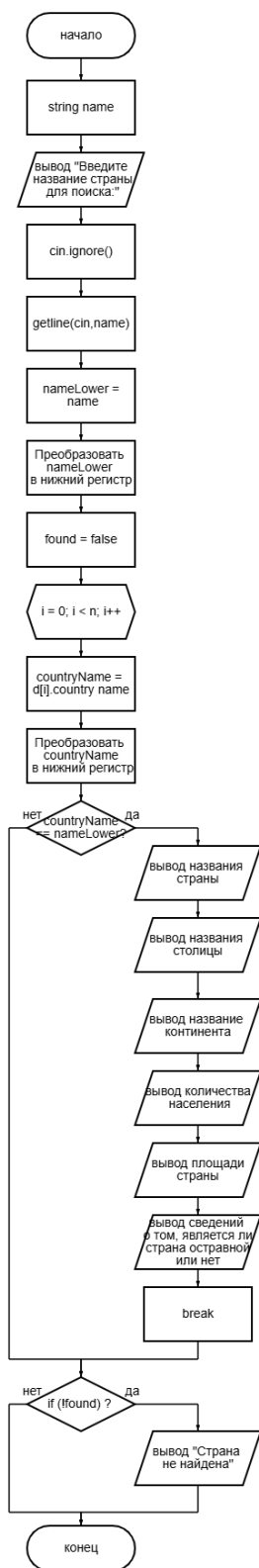


Рисунок 10 – Блок схема функции SearchByCountryName

Для создания новой пустой базы данных используется функция CreateNewDatabase() (рисунок 11).

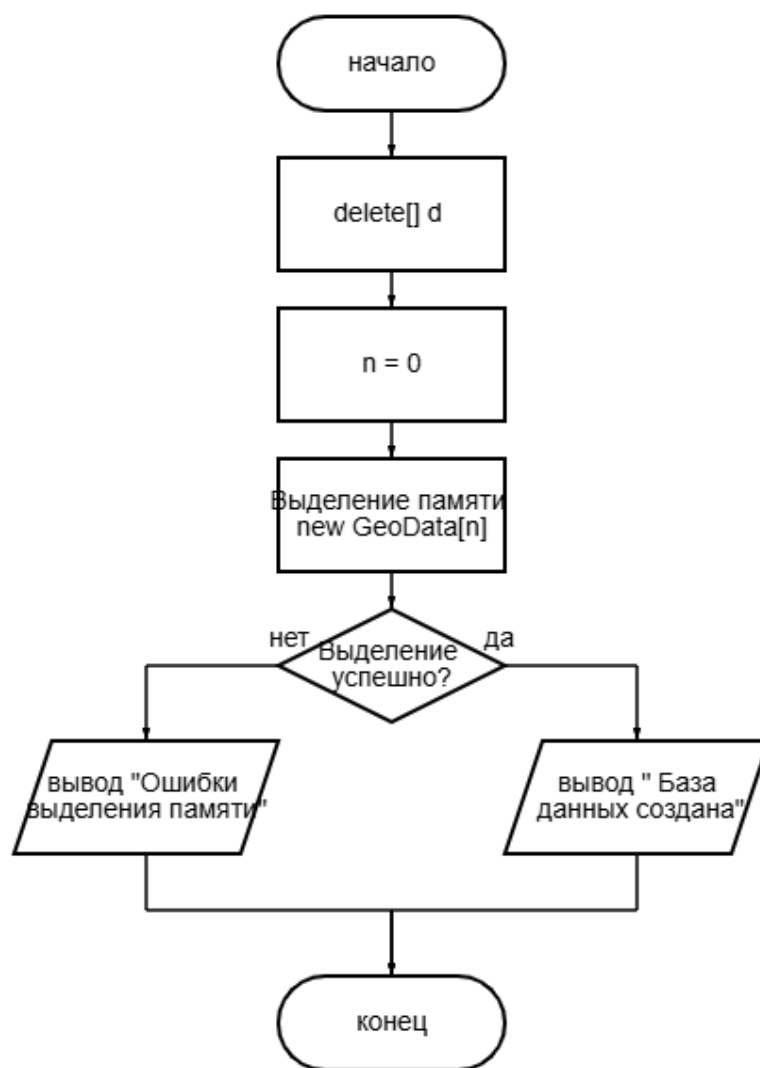


Рисунок 11 – Блок схема функции CreateNewDatabase

Для сохранения текущей базы данных в текстовый файл используется функция SaveToFile() (рисунок 12).

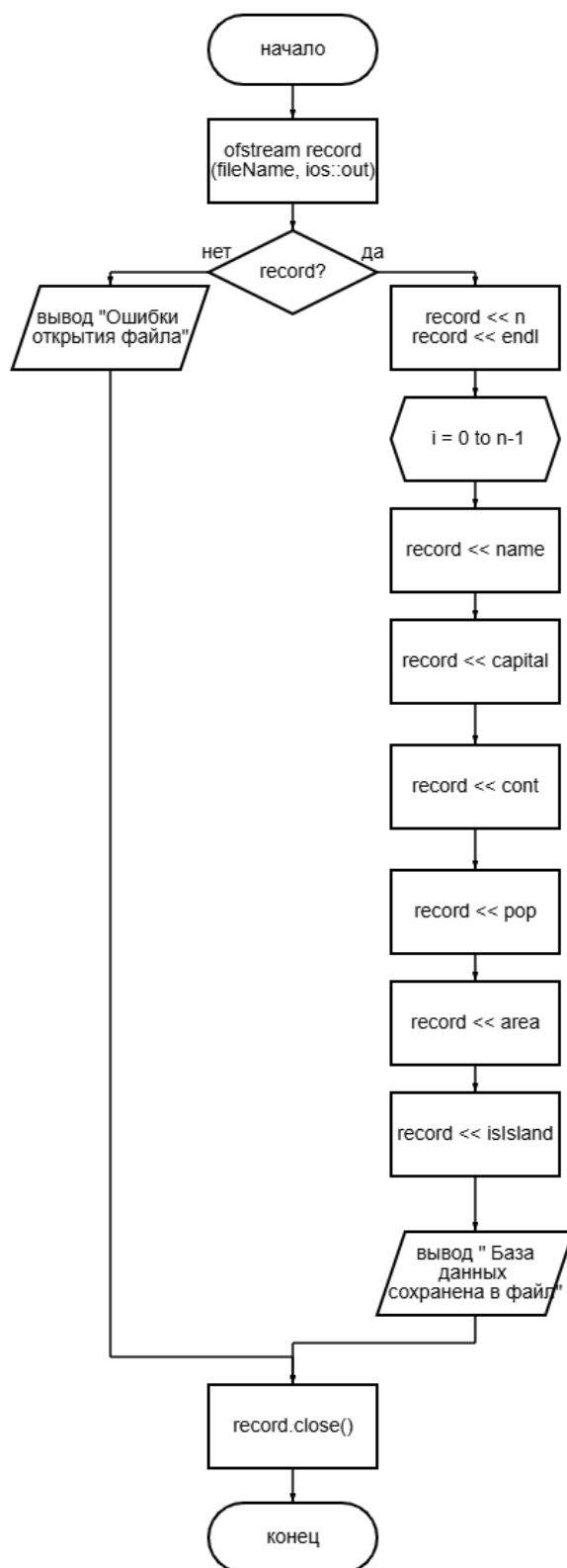


Рисунок 12 – Блок схема функции SaveToFile

Для загрузки базы данных из текстового файла используется функция LoadFromFile() (рисунок 13).

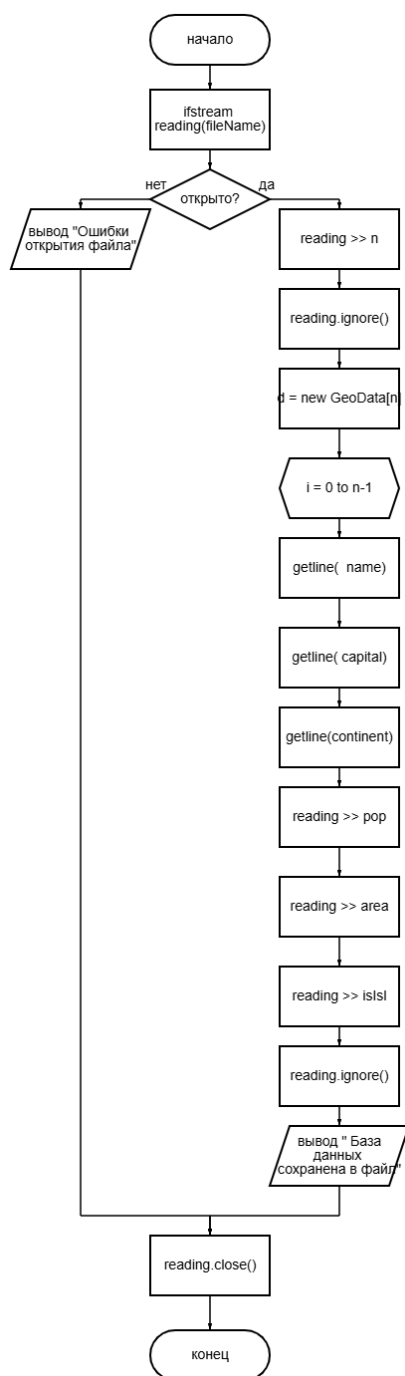


Рисунок 13 – Блок схема функции LoadFromFile

Все функции обеспечивают слаженную работу все программы. Что полностью соответствует требованиям пользователя.

Руководство пользователя

Программа «Functions.exe» предназначена для хранения информации о странах. Программа имеет интуитивно понятный интерфейс и поддерживает такие операции как выход из программы, ввод новых данных о странах, удаление данных о стране, изменения данных о стране, сортировку стран по названию, фильтрацию стран по континенту, поиск страны по названию, создание новой базы данных в файле, сохранение текущей базы данных в файл, загрузку базы данных из файла и показ всей базы данных (рисунок 14).

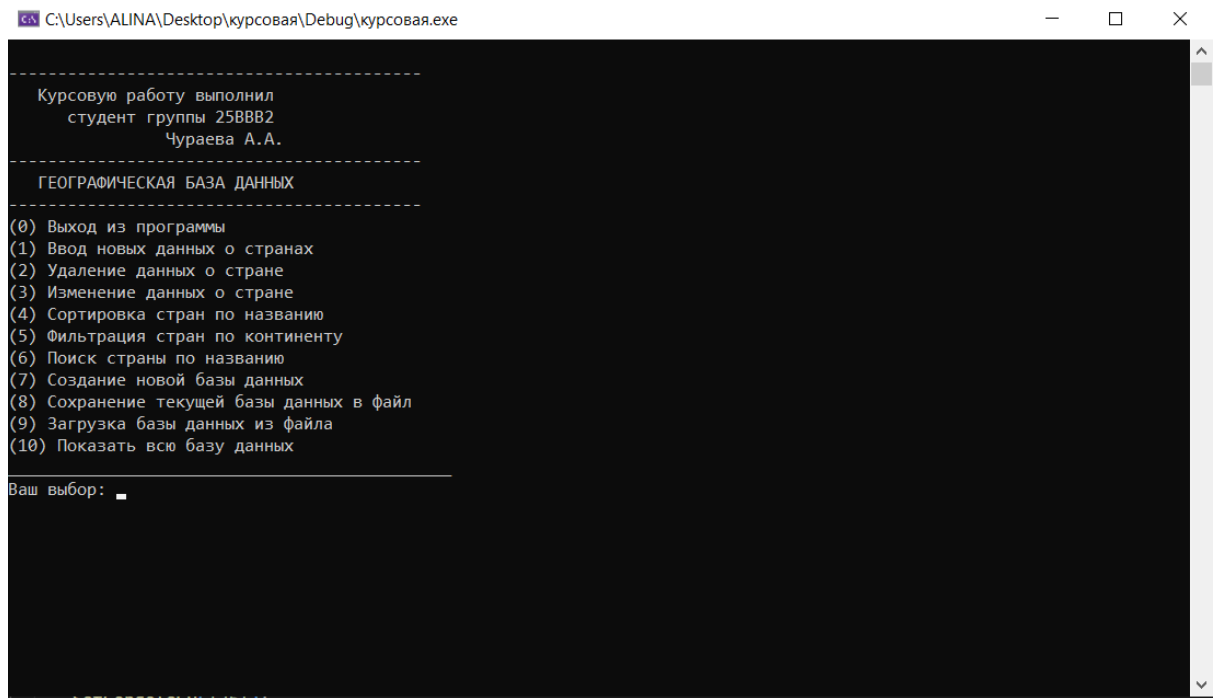


Рисунок 14 – Вывод меню

Для выхода из программы следует выбрать соответствующий пункт (рисунок 15).

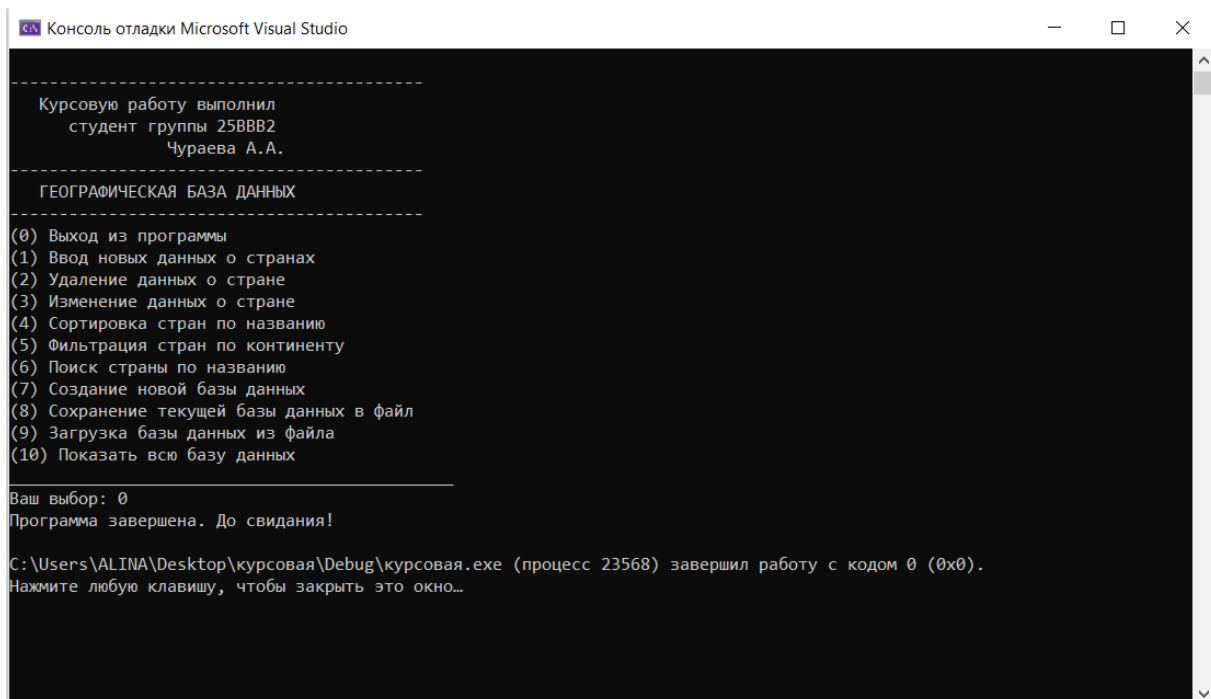


Рисунок 15 – Выход из программы

Для ввода новых данных о странах в базу данных необходимо выбрать пункт «Ввод новых данных о странах» (рисунок 16).

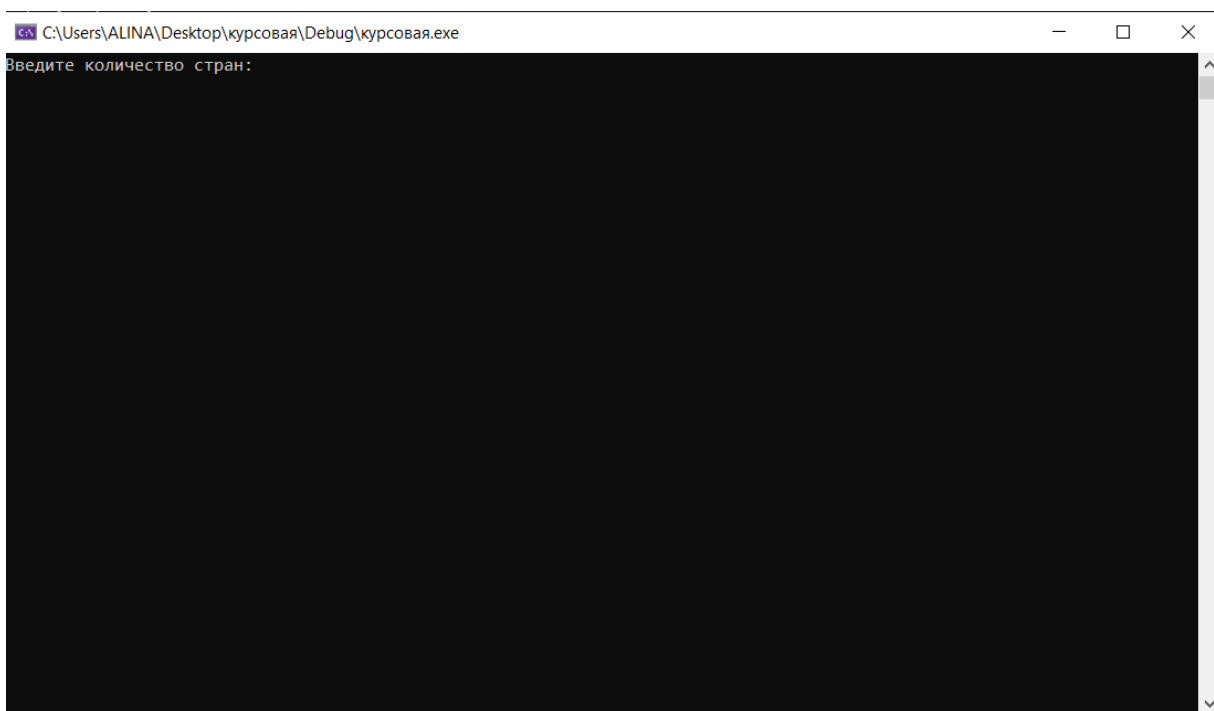


Рисунок 16 – Окно для ввода количества новых стран

В окне предлагается ввести количество стран для ввода. После ввода количества стран будет открыто окно ввода для пользователя.

Пользователь после ввода данных может войти в меню и выбрать любой пункт для дальнейших действий (рисунок 17).

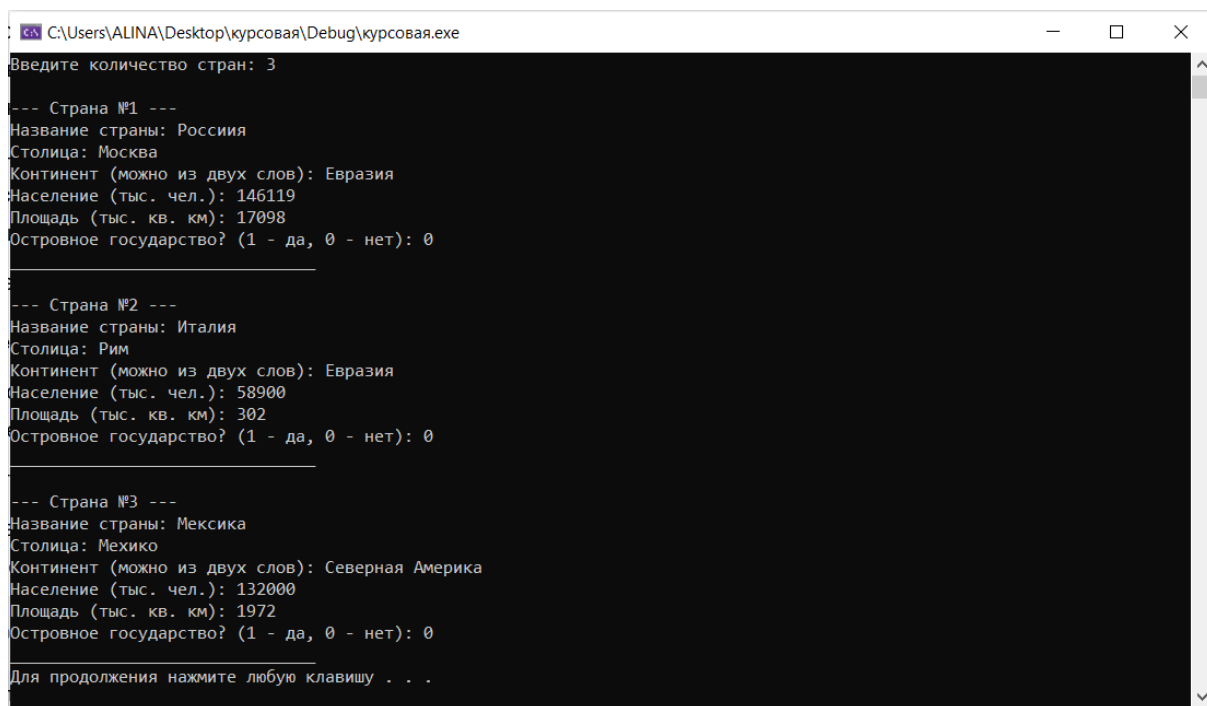


Рисунок 17 – Вывод данных о странах

Для удаления данных о стране пользователю нужно выбрать пункт под цифрой 2. В данном окне будет предложен выбор того, какую именно страну пользователь хочет удалить (рисунок 18).

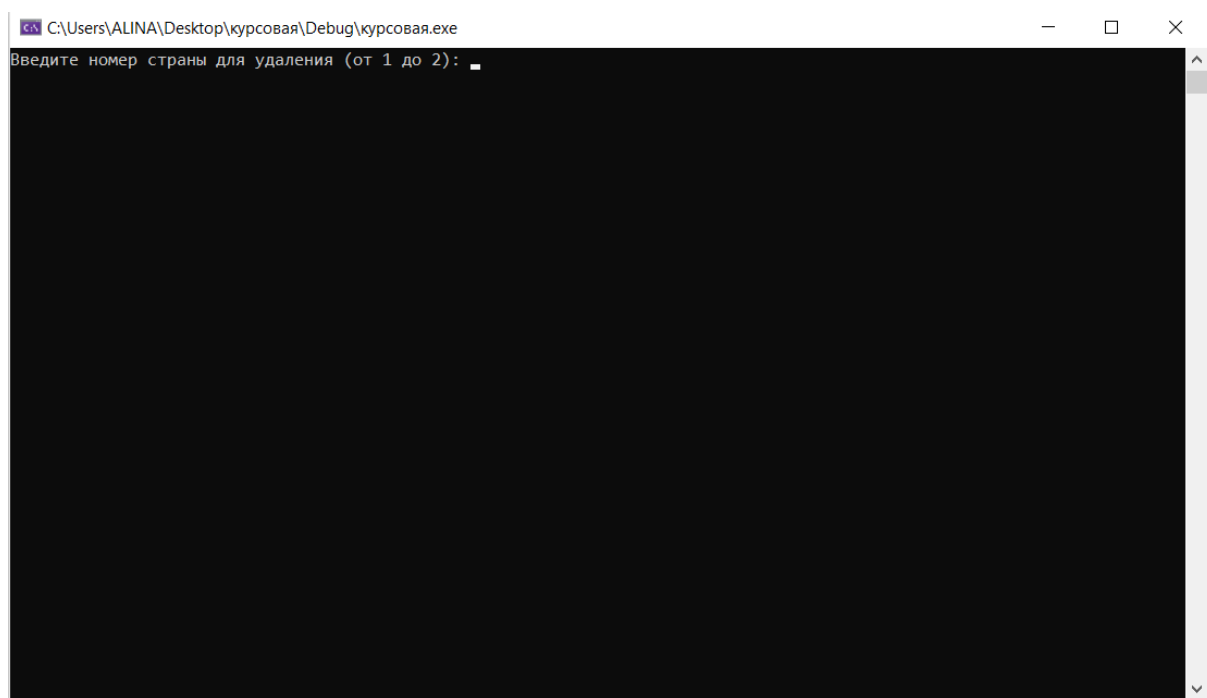


Рисунок 18 – Окно удаления данных о странах

После соответствующего выбора программа сообщит о удалении выбранной страны и предоставит возможность выхода в меню (рисунок 19).

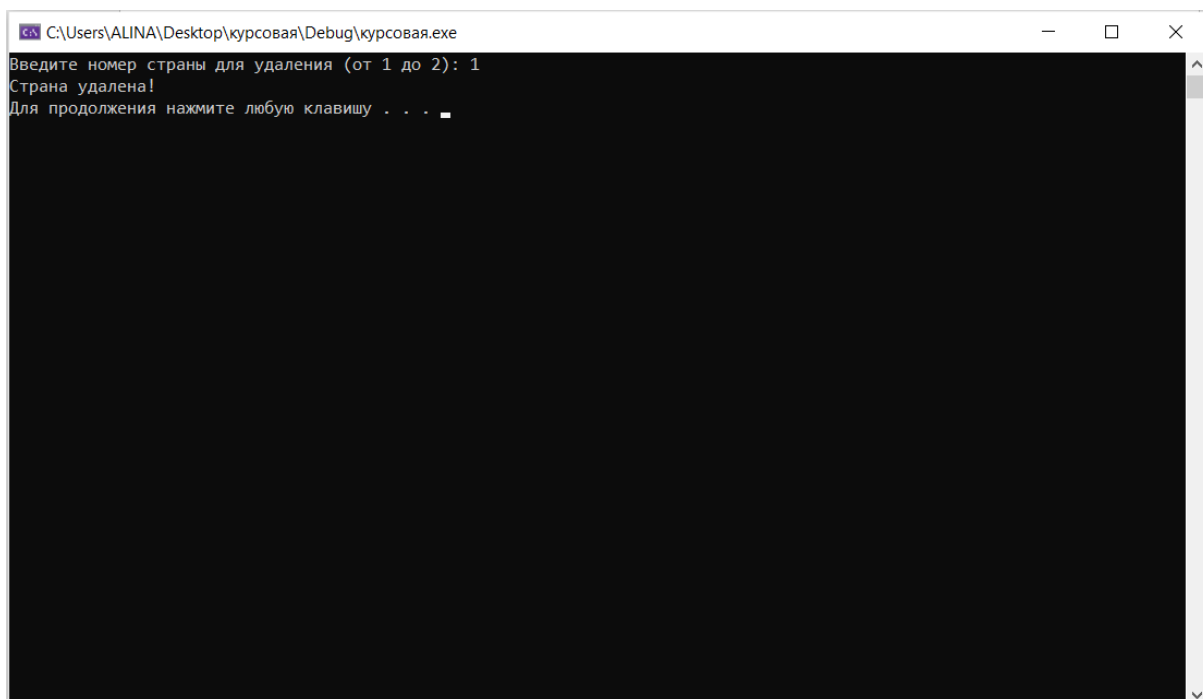


Рисунок 19 – Окно после удаления данных о странах

Для изменения данных о стране необходимо выбрать соответствующий пункт. В окне будет представлен выбор страны для изменения. После чего пользователь может осуществить нужные ему изменения (рисунок 20).

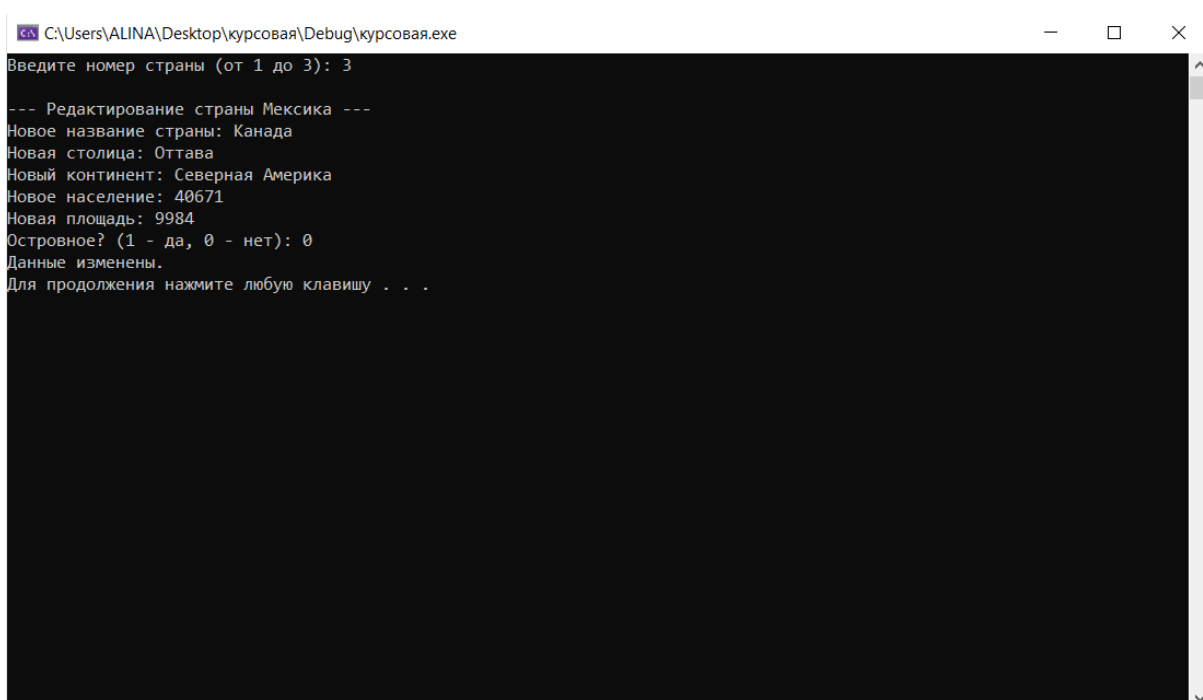


Рисунок 20 – Вывод окна с изменением данных выбранной страны

Для сортировки стран по названию пользователю нужно выбрать пункт «Сортировка стран по названию». После чего выйдет сообщение сообщающее об этом (рисунок 21).

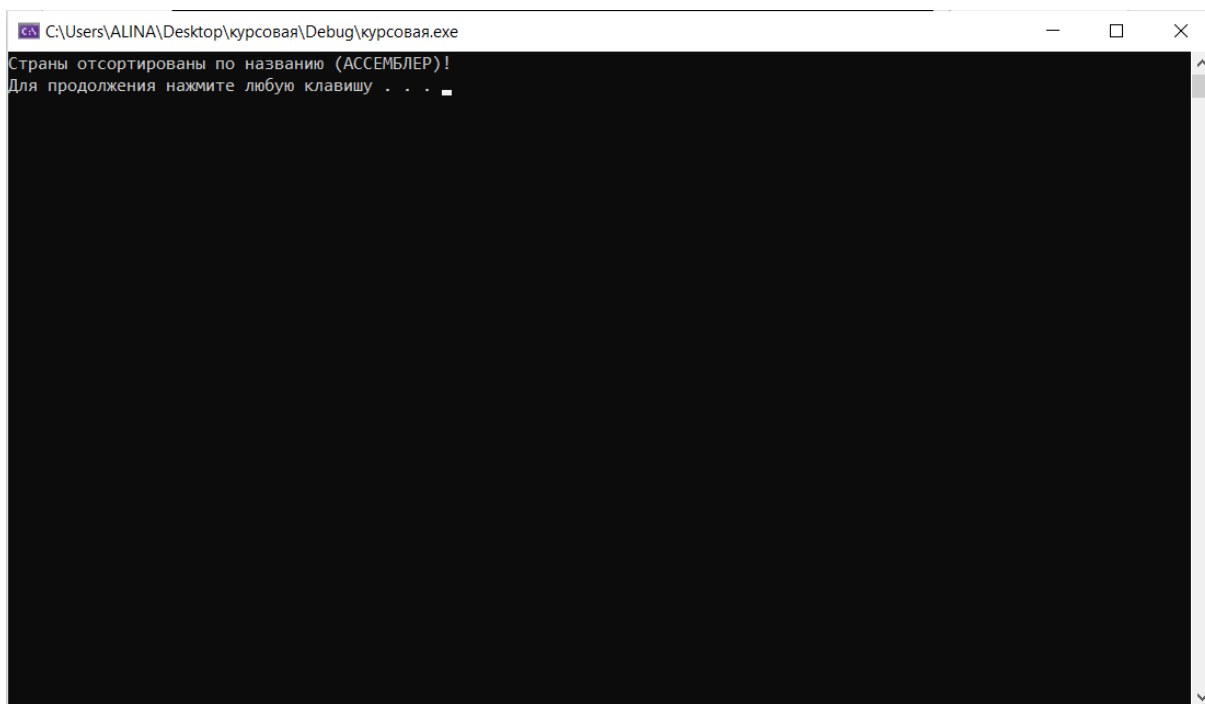


Рисунок 21 – Вывод окна сортировки

Результат сортировки можно посмотреть, выбрав пункт «Показать всю базу данных» (рисунок 22).

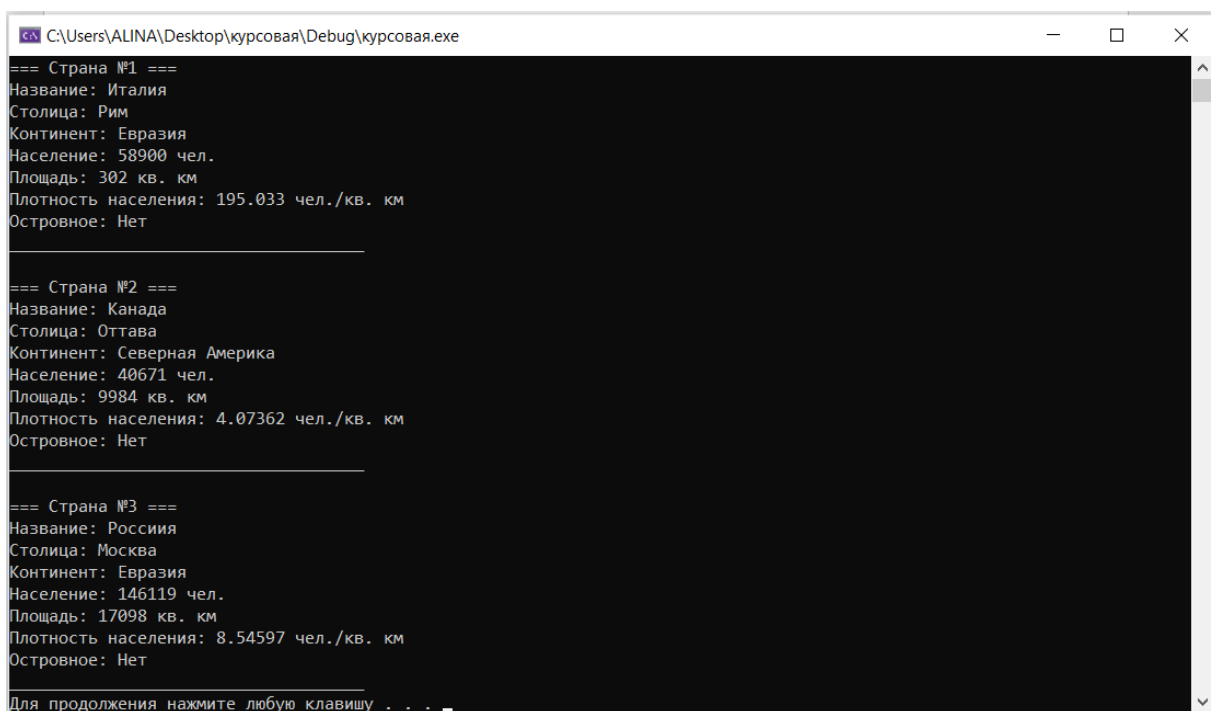


Рисунок 22 – Вывод отсортированной базы данных

Так же в программе доступна фильтрация стран по континенту под цифрой 5. Результат фильтрации показан ниже (рисунок 23).

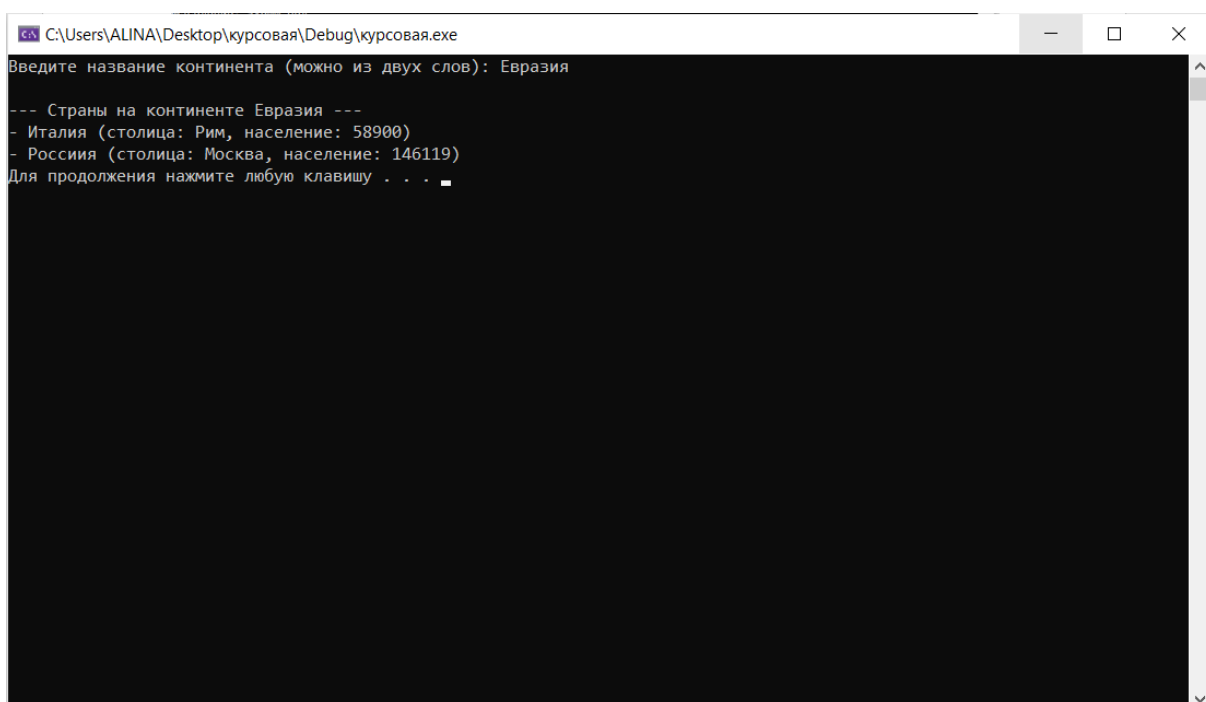


Рисунок 23 – Вывод результата фильтрации

Программа осуществляет поиск стран по названию (рисунок 24).

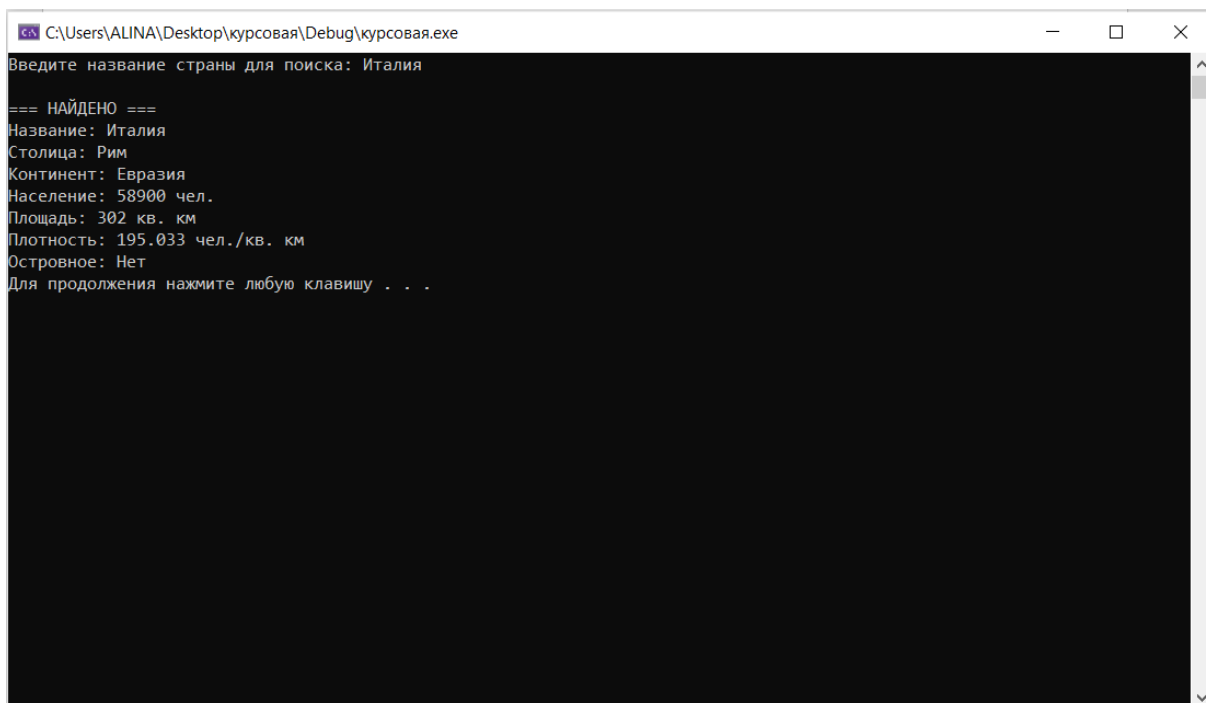


Рисунок 24 – Вывод окна с поиском страны по названию

Программа реализует создание новой базы данные, при выборе соответствующей команды (рисунок 25).

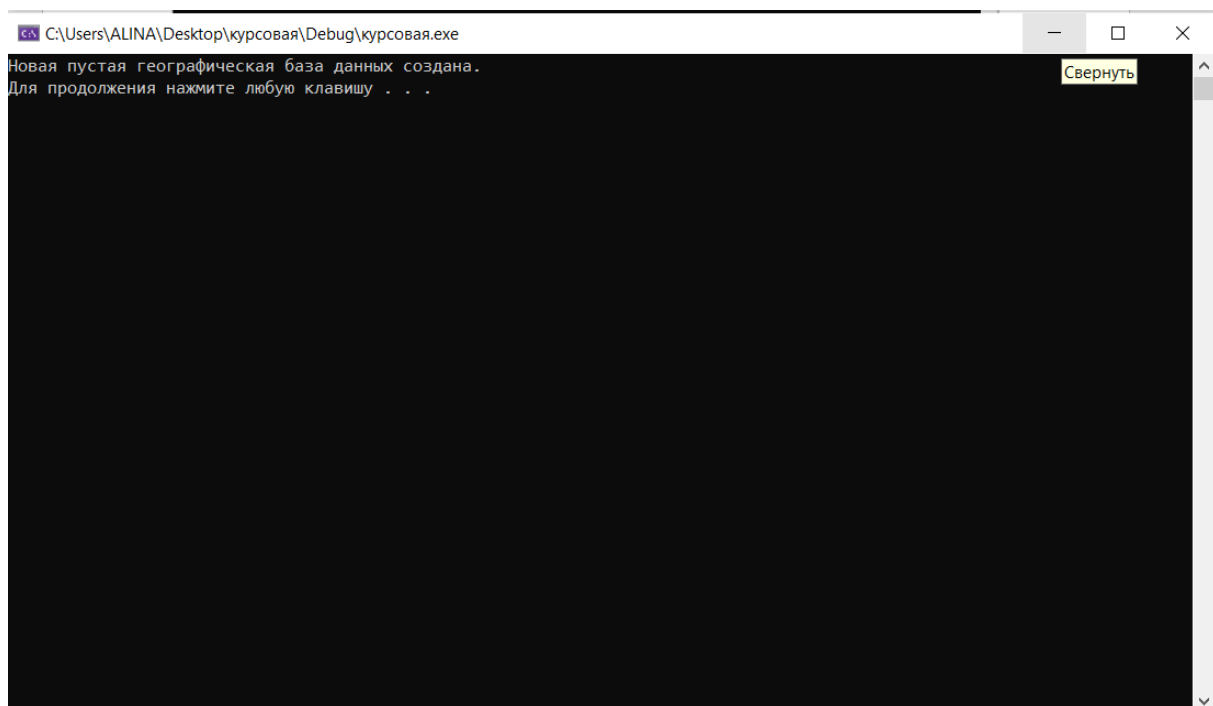


Рисунок 25 – Вывод окна создания новой базы данных

При выборе пункта «Сохранение текущей базы данных в файл» текущая база данных о странах сохраняется в файл с тем названием, которое укажет пользователь (рисунок 26).

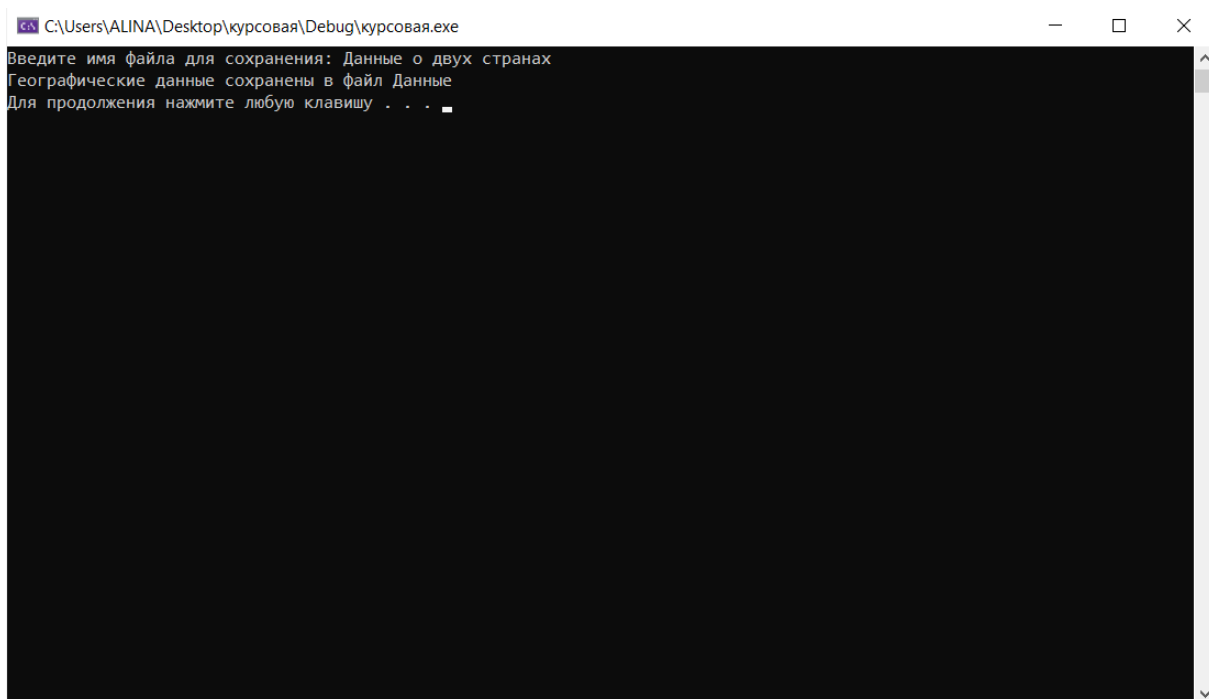


Рисунок 26 – Вывод окна с сохранением базы данных в файл

Реализация этого пункта программы показана ниже (рисунок 27).

Данные о двух странах	16.05.2026 17:10	Файл	1 КБ
курсовая.cpp	16.05.2026 16:58	Исходный файл C...	4 КБ
курсовая.sln	21.04.2026 23:54	Visual Studio Solut...	2 КБ
курсовая.vcxproj	21.04.2026 23:58	VC++ Project	7 КБ
курсовая.vcxproj.filters	21.04.2026 23:58	VC++ Project Filte...	2 КБ
курсовая.vcxproj.user	21.04.2026 23:54	Файл "USER"	1 КБ

Рисунок 27 – Реализация сохранения базы данных в файл

При выборе реализации «Загрузка базы данных из файла» пользователю предлагается ввести название файла из которого будут взяты данные (рисунок 28).

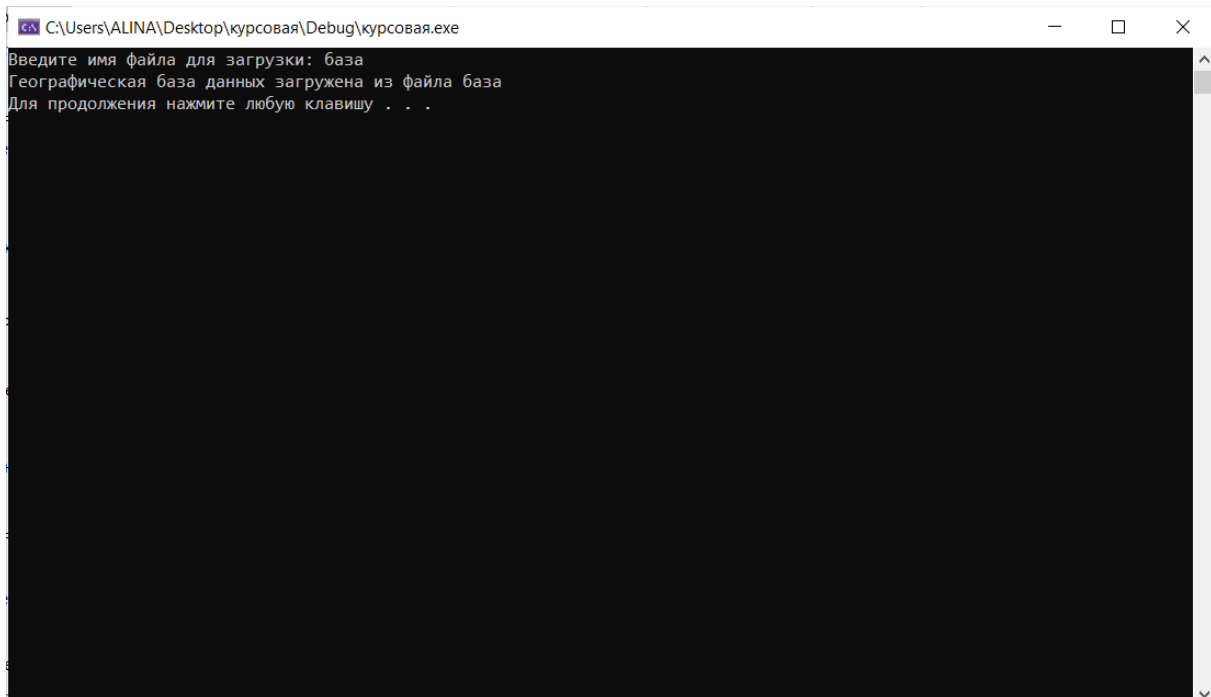


Рисунок 28 – Вывод окна загрузки базы данных из файла

Данные можно посмотреть, выбрав пункт «Показать всю базу данных» (рисунок 29).

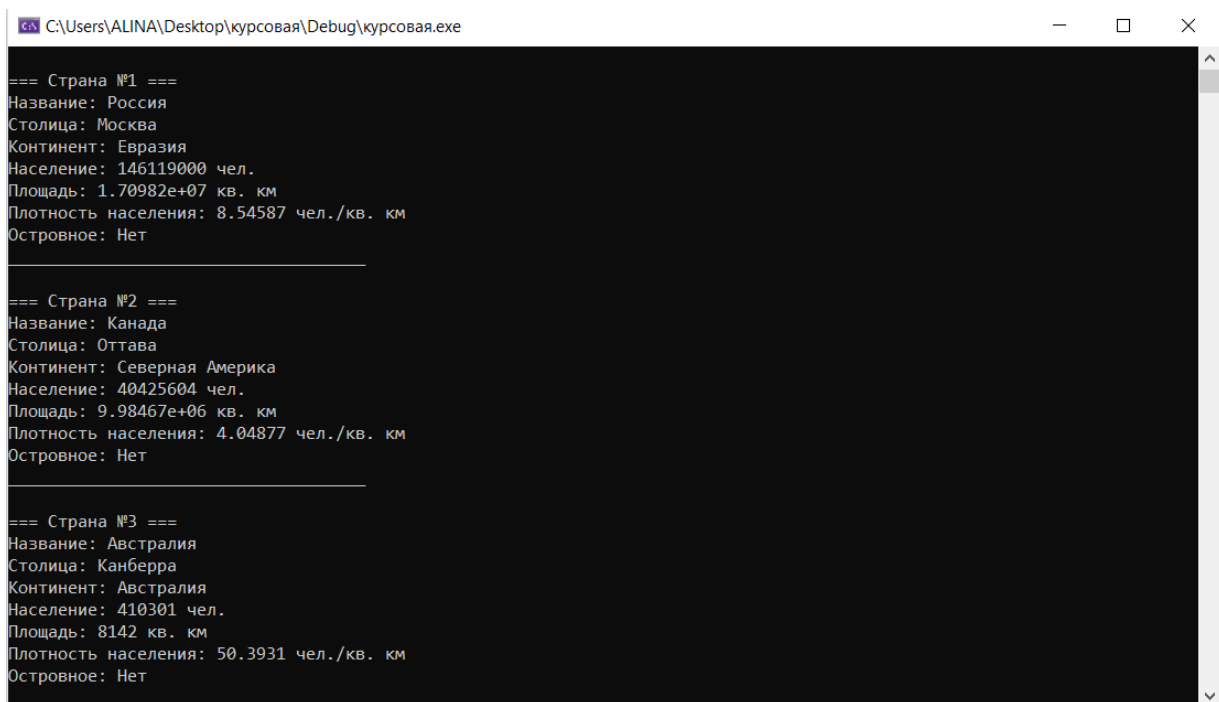


Рисунок 29 – Вывод всей базы данных

Руководство подробно показывает все возможности программы. Пользователь с его помощью может быстро разобраться с функционалом программы.

Заключение

При выполнении данной курсовой работы получены навыки разработки многомодульных программ. Были освоены приемы создания псевдографического интерфейса, изучены функции работы с консолью, способы обработки событий от клавиатуры. Усвоены механизмы реализации меню. Изучены функции работы с файлами. Также были получены основные навыки отладки и тестирования программ и программирования в среде Visual Studio 2022 на языках C++ и Assembler.

В рамках курсовой работы написана программа для работы с географической базой данных. Программа предоставляет небольшой, но достаточный для использования список возможностей. Не требует большого количества системных ресурсов и обеспечивает быструю обработку базы данных.

В дальнейшем программу можно улучшить, путем добавления графического интерфейса, что позволит улучшить навигацию по меню программы, путем добавления возможности использования мыши.

Список литературы

1. Керниган Б., Ритчи Д. Язык программирования Си. — Москва: Финансы и статистика, 1985. — 279 с.
2. Прата, С. Язык программирования C++. Лекции и упражнения / С. Прата. — М. : Издательство «ДиаСофтЮП», 2005. — 1104 с..
3. Шилдт Г. Полный справочник по C / пер. с англ. [и др.]; под ред. В. П. Леонтьева. — М. : Вильямс, 2002. — 712 с. — ISBN 5-8459-0339-6.
4. Зубков, С. В. Assembler для DOS, Windows и Linux / С. В. Зубков. — Москва : ДМК Пресс, 2011.
5. Аблязов Р. З. Программирование на ассемблере на платформе x86–64 / Р. З. Аблязов. — Москва : ДМК Пресс, 2011. — 304 с. — ISBN 978-5-94074-676-8.
6. Культин, Никита Борисович. Основы программирования Microsoft Visual C++ 2010 [Текст] / Никита Культин. — Санкт-Петербург : БХВ-Петербург, 2010. — 376 с. : ил. — (Самоучитель). — ISBN 978-5-9775-0520-8.

Приложение А

Листинги программы

Курсовая.cpp

```
#include "Functions.h"

int _stateMenu;

void Menu() {

    cout << "\n-----" << endl;

    cout << "    Курсовую работу выполнил" << endl;

    cout << "        студент группы 25BBB2" << endl;

    cout << "                Чураева А.А." << endl;

    cout << "-----" << endl;

    cout << "    ГЕОГРАФИЧЕСКАЯ БАЗА ДАННЫХ" << endl;

    cout << "-----" << endl;

    cout << "(0) Выход из программы" << endl;

    cout << "(1) Ввод новых данных о странах" << endl;

    cout << "(2) Удаление данных о стране" << endl;

    cout << "(3) Изменение данных о стране" << endl;

    cout << "(4) Сортировка стран по названию" << endl;

    cout << "(5) Фильтрация стран по континенту" << endl;

    cout << "(6) Поиск страны по названию" << endl;

    cout << "(7) Создание новой базы данных в файле" << endl;

    cout << "(8) Сохранение текущей базы данных в файл" << endl;

    cout << "(9) Загрузка базы данных из файла" << endl;

    cout << "(10) Показать всю базу данных" << endl;
```

```

        cout << " _____" <<
endl;

        cout << "Ваш выбор: ";

        cin >> _stateMenu;
    }

int main() {

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);


    int amountOfData = 0;

    string fileName;

    GeographicData* d = new GeographicData[amountOfData];


    Menu();

    while (_stateMenu != 0) {

        system("cls");


        switch (_stateMenu) {

        case 1:

            DataEntry(d, amountOfData);

            break;


        case 2:

            if (amountOfData > 0) DeleteData(d, amountOfData);

            else cout << "Нет данных для удаления." << endl;

            break;

```

```

case 3:

    if (amountOfData > 0) DataChange(d, amountOfData);

    else cout << "Нет данных для изменения." << endl;

    break;


case 4:

    if (amountOfData > 0) DataSorting(d, amountOfData);

    else cout << "Нет данных для сортировки." << endl;

    break;


case 5:

    if      (amountOfData      >      0)      FilterByContinent(d,
amountOfData);

    else cout << "Нет данных для фильтрации." << endl;

    break;


case 6:

    if      (amountOfData      >      0)      SearchByCountryName(d,
amountOfData);

    else cout << "Нет данных для поиска." << endl;

    break;


case 7:

    CreateNewDatabase(d, amountOfData);

    break;

```

```

case 8:

    if (amountOfData > 0) {

        cout << "Введите имя файла для сохранения: ";

        cin >> fileName;

        SaveToFile(d, amountOfData, fileName);

    }

    else cout << "Нет данных для сохранения." << endl;

    break;


case 9:

    cout << "Введите имя файла для загрузки: ";

    cin >> fileName;

    LoadFromFile(d, amountOfData, fileName);

    break;


case 10:

    Print(d, amountOfData);

    break;


default:

    cout << "Неверная команда. Попробуйте снова." << endl;

    break;

}


system("pause");

```

```
        system("cls");

        Menu();

    }

    delete[] d;

    cout << "Программа завершена. До свидания!" << endl;

    return 0;

}
```

Functions.h

```
#pragma once

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

#include <cctype> // для tolower

using namespace std;

struct Country {

    string name;           // название страны

    string capital;        // столица

    string continent;      // континент

    long long population;  // население

    double area;           // площадь (кв. км)

    bool isIsland;         // островное государство или нет

};

struct GeographicData {

    Country country;

};

void DataEntry(GeographicData* (&d), int& n);

void Print(GeographicData* d, int n);
```

```
void DataChange(GeographicData* (&d), int n);  
void DeleteData(GeographicData* (&d), int& n);  
void Copy(GeographicData* (&d_n), GeographicData* (&d_o), int n);  
void Copy(GeographicData& d_n, GeographicData& d_o);  
void DataSorting(GeographicData* d, int n);  
void FilterByContinent(GeographicData* d, int n);  
void SearchByCountryName(GeographicData* d, int n);  
void CreateNewDatabase(GeographicData* (&d), int& n);  
void SaveToFile(GeographicData* d, int n, string fileName);  
void LoadFromFile(GeographicData* (&d), int& n, string fileName);
```

Functions.cpp

```
#include "Functions.h"

void DataEntry(GeographicData* (&d), int& n) {

    cout << "Введите количество стран: ";

    cin >> n;

    cin.ignore();

    d = new GeographicData[n];

    for (int i = 0; i < n; i++) {

        cout << "\n--- Страна №" << i + 1 << " ---" << endl;

        cout << "Название страны: ";

        getline(cin, d[i].country.name);

        cout << "Столица: ";

        getline(cin, d[i].country.capital);

        cout << "Континент (можно из двух слов): ";

        getline(cin, d[i].country.continent);

        cout << "Население (чел.): ";

        cin >> d[i].country.population;

        cout << "Площадь (кв. км): ";
```

```

    cin >> d[i].country.area;

    cout << "Островное государство? (1 - да, 0 - нет): ";
    cin >> d[i].country.isIsland;

    cin.ignore();

    cout << "_____ " << endl;
}
}

void Print(GeographicData* d, int n) {
    if (n == 0) {
        cout << "База данных пуста." << endl;
        return;
    }

    for (int i = 0; i < n; i++) {
        cout << "\n=== Страна №" << i + 1 << " ===" << endl;
        cout << "Название: " << d[i].country.name << endl;
        cout << "Столица: " << d[i].country.capital << endl;
        cout << "Континент: " << d[i].country.continent << endl;
        cout << "Население: " << d[i].country.population << " чел."
<< endl;

        cout << "Площадь: " << d[i].country.area << " кв. км" <<
endl;
    }
}

```

```

        cout << "Плотность населения: " <<
(double)d[i].country.population / d[i].country.area << " чел./кв.
км" << endl;

        cout << "Островное: " << (d[i].country.isIsland ? "Да" :
"Нет") << endl;

        cout << "_____ " << endl;
    }
}

```

```

void DataChange(GeographicData* (&d), int n) {

    int _n;

    cout << "Введите номер страны (от 1 до " << n << "): ";

    cin >> _n;

    _n--;

    cin.ignore();

    if (_n >= 0 && _n < n) {

        cout << "\n--- Редактирование страны " << d[_n].country.name
<< " ---" << endl;

        cout << "Новое название страны: ";

        getline(cin, d[_n].country.name);

        cout << "Новая столица: ";

        getline(cin, d[_n].country.capital);

        cout << "Новый континент: ";
    }
}

```

```

        getline(cin, d[_n].country.continent);

        cout << "Новое население: ";

        cin >> d[_n].country.population;

        cout << "Новая площадь: ";

        cin >> d[_n].country.area;

        cout << "Островное? (1 - да, 0 - нет): ";

        cin >> d[_n].country.isIsland;

        cin.ignore();

        cout << "Данные изменены." << endl;
    }

    else

        cout << "Номер введен не верно." << endl;
}

void DeleteData(GeographicData* (&d), int& n) {
    int _n;

    cout << "Введите номер страны для удаления (от 1 до " << n <<
    "): ";

    cin >> _n;

    _n--;
}

```

```

    if (_n >= 0 && _n < n) {

        GeographicData* buf = new GeographicData[n];

        Copy(buf, d, n);

        --n;

        d = new GeographicData[n];

        int q = 0;

        for (int i = 0; i <= n; i++) {

            if (i != _n) {

                d[q] = buf[i];

                ++q;

            }

        }

        delete[] buf;

        cout << "Страна удалена!" << endl;

    }

    else

        cout << "Номер введен не верно!" << endl;

}

void Copy(GeographicData* (&d_n), GeographicData* (&d_o), int n) {

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

        d_n[i] = d_o[i];

}

```

```

void Copy(GeographicData& d_n, GeographicData& d_o) {
    d_n.country.name = d_o.country.name;
    d_n.country.capital = d_o.country.capital;
    d_n.country.continent = d_o.country.continent;
    d_n.country.population = d_o.country.population;
    d_n.country.area = d_o.country.area;
    d_n.country.isIsland = d_o.country.isIsland;
}

void DataSorting(GeographicData* d, int n) {
    if (n <= 1) {
        cout << "Недостаточно данных для сортировки." << endl;
        return;
    }

    bool swapped;

    for (int i = 0; i < n - 1; i++) {
        swapped = false;

        for (int j = 0; j < n - i - 1; j++) {

            const char* str1 = d[j].country.name.c_str();
            const char* str2 = d[j + 1].country.name.c_str();

            int cmpResult = 0;

```

```

__asm {

    mov esi, str1;

    mov edi, str2;


compare_loop :

    mov al, [esi];

        mov bl, [edi];


        cmp al, bl;

        jne not_equal;

        test al, al;

        je strings_equal;


        inc esi;

        inc edi;

        jmp compare_loop;


not_equal :


        sub al, bl;

        movsx eax, al;

        mov cmpResult, eax;

        jmp compare_done


strings_equal :

```

```

        mov cmpResult, 0; строки равны

        compare_done :

    }

    if (cmpResult > 0) {

        GeographicData temp;

        Copy(temp, d[j]);

        Copy(d[j], d[j + 1]);

        Copy(d[j + 1], temp);

        swapped = true;

    }

}

if (!swapped) break;

}

cout << "Страны отсортированы по названию (АССЕМБЛЕР)!" << endl;

}

void FilterByContinent(GeographicData* d, int n) {

    string continent;

    cout << "Введите название континента (можно из двух слов): ";

    cin.ignore();

    getline(cin, continent);

    string continentLower = continent;

    for (char& c : continentLower) c = tolower(c);

```

```

    bool found = false;

    cout << "\n--- Страны на континенте " << continent << " ---" <<
endl;

    for (int i = 0; i < n; i++) {

        string cont = d[i].country.continent;

        for (char& c : cont) c = tolower(c);

        if (cont == continentLower) {

            cout << "- " << d[i].country.name << " (столица: " <<
d[i].country.capital
                << ", население: " << d[i].country.population << ")"
<< endl;

            found = true;

        }

    }

    if (!found)

        cout << "Нет стран на континенте " << continent << endl;

}

void SearchByCountryName(GeographicData* d, int n) {

    string name;

    cout << "Введите название страны для поиска: ";

    cin.ignore();

    getline(cin, name);

    string nameLower = name;

```

```

    for (char& c : nameLower) c = tolower(c);

    bool found = false;

    for (int i = 0; i < n; i++) {

        string countryName = d[i].country.name;

        for (char& c : countryName) c = tolower(c);

        if (countryName == nameLower) {

            cout << "\n=== НАЙДЕНО ===" << endl;

            cout << "Название: " << d[i].country.name << endl;

            cout << "Столица: " << d[i].country.capital << endl;

            cout << "Континент: " << d[i].country.continent << endl;

            cout << "Население: " << d[i].country.population << "
чел." << endl;

            cout << "Площадь: " << d[i].country.area << " кв. км"
<< endl;

            cout << "Плотность: " << (double)d[i].country.population
/ d[i].country.area << " чел./кв. км" << endl;

            cout << "Островное: " << (d[i].country.isIsland ? "Да"
: "Нет") << endl;

            found = true;

            break;

        }

    }

    if (!found)

        cout << "Страна \"" << name << "\" не найдена." << endl;

}

```

```

void CreateNewDatabase(GeographicData* (&d), int& n) {

    delete[] d;

    n = 0;

    d = new GeographicData[n];

    cout << "Новая пустая географическая база данных создана." <<
endl;

}

void SaveToFile(GeographicData* d, int n, string fileName) {

    ofstream record(fileName, ios::out);

    if (record) {

        record << n << endl;

        for (int i = 0; i < n; i++) {

            record << d[i].country.name << endl;

            record << d[i].country.capital << endl;

            record << d[i].country.continent << endl;

            record << d[i].country.population << endl;

            record << d[i].country.area << endl;

            record << d[i].country.isIsland << endl;

        }

        cout << "Географические данные сохранены в файл " << fileName
<< endl;

    }

    else

        cout << "Ошибка открытия файла!" << endl;
}

```

```

        record.close();
    }

void LoadFromFile(GeographicData* (&d), int& n, string fileName) {
    ifstream reading(fileName);

    if (reading) {
        reading >> n;
        reading.ignore();

        d = new GeographicData[n];
        for (int i = 0; i < n; i++) {
            getline(reading, d[i].country.name);
            getline(reading, d[i].country.capital);
            getline(reading, d[i].country.continent);
            reading >> d[i].country.population;
            reading >> d[i].country.area;
            reading >> d[i].country.isIsland;
            reading.ignore();
        }

        cout << "Географическая база данных загружена из файла " <<
        fileName << endl;
    }

    else
        cout << "Ошибка открытия файла" << endl;

    reading.close();
}

```