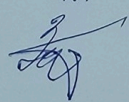


Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
по курсу «Программирование»
на тему «База данных»

20.06.26
ОТ


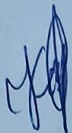
Выполнил:
студент группы 25BBB4
Герасимова Е.А

Принял:
Кандидат т.н. Токарев А.Н.

Пенза 2026

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники

Кафедра "Вычислительная техника"



"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

М.А. Митрохин

« ____ » ____ 20__ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Герасимовой Елене Андреевне Группа 25 ВВВ4

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

Исходные данные (технические требования) на проектирование

База данных "Столицы стран". Технические требования:

- | | |
|--|-----------------------------------|
| - Добавление записей | - Создание новой базы данных |
| - Удаление записей | - Сохранение текущей базы данных |
| - Редактирование записей | - Загрузка сохраненных баз данных |
| - Поиск и сортировка записей | |
| - Фильтрация записей по определенному критерию | |

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 “18” 02. 2026

Дата защиты проекта “ ”

Руководитель _____

Задание получил “25” 02 2026 г.

Студент Геращенко Геращенко Е.А.

Содержание

Введение.....	5
Постановка задачи.....	7
Выбор решения.....	8
Описание разработки программы.....	10
Отладка и тестирование.....	11
Описание программы.....	12
Руководство пользователя.....	24
Заключение	28
Список используемой литературы	29
Приложение А. Листинги программы	Ошибка! Закладка не определена.
Файл functions.h.....	Ошибка! Закладка не определена.
Файл source.cpp	Ошибка! Закладка не определена.
Файл filtnchange.cpp.....	Ошибка! Закладка не определена.
Файл database.cpp	Ошибка! Закладка не определена.
Файл record.cpp.....	Ошибка! Закладка не определена.
Файл sorting.cpp.....	Ошибка! Закладка не определена.
Файл capitals.cpp.....	Ошибка! Закладка не определена.
Приложение В. Результаты работы программы	Ошибка! Закладка не определена.
Приложение С. Результаты работы программы	Ошибка! Закладка не определена.

Введение

На сегодняшний день информация – ценный источник знаний, систематизации которой, позволяет оптимизировать работу как человека, так и общества в целом. Компьютерные технологии во многом упрощают нашу жизнь в самых разнообразных сферах нашей деятельности. И именно компьютерные технологии позволяют грамотно взаимодействовать с информацией, обрабатывать её и хранить на компактных носителях.

База данных представляет собой упорядоченную, структурированную совокупность данных, подобранных таким образом, что эти данные могут легко и быстро обрабатываться на ЭВМ.

История баз данных в узком смысле рассматривает базы данных в традиционном (современном) понимании. Эта история начинается с 1955 года, когда появилось программируемое оборудование обработки записей. Программное обеспечение этого времени поддерживало модель обработки записей на основе файлов. Для хранения данных использовались перфокарты.

Оперативные сетевые базы данных появились в середине 1960-х. Операции над оперативными базами данных обрабатывались в интерактивном режиме с помощью терминалов. Простые индексно-последовательные организации записей быстро развились к более мощной модели записей, ориентированной на наборы.

В наше время базы данных являются неотъемлемой частью любой организации, любого предприятия. Динамические сайты используют базу данных чтобы хранить данные пользователя, как и стриминговые сервисы с подписочной моделью. Банки хранят в огромных базах данных данные пользователей.

Данная программа является информационно-поисковой системой и позволяет работать с простейшей базой данных сотовых операторов, позволяющей любому человеку подобрать выгодный тарифный план под свои

нужды, отслеживать историю тарифных планов сотовых операторов и наблюдать историю их изменения.

Постановка задачи

Необходимо разработать программу для работы с базой данных сотовых операторов. Программа должна иметь интуитивно понятный интерфейс для пользователя. Необходимо определить названия пунктов меню, построить алгоритмы работы с массивами структур для фильтрации и сортировки записей по определенным критериям, изучить функции для работы с файлами, чтобы сохранять базу данных в файл и загружать из файла.

Многомодульность программы. Программа должна быть поделена на логические модули. Это упростит поиск ошибок при отладке и тестировании программы, а также позволит легко расширять функционал программы.

Использование сложных типов данных – массивов, структур, файлов. Это необходимо для более простой и интуитивной обработки данных в коде программы.

Режим работы видеосистемы – текстовый/графический. Для начала необходимо определиться с типом интерфейса и с элементами управления, затем необходимо изучить способы их реализации.

Устройство ввода-вывода – клавиатура и мышь. Необходимо различать и идентифицировать действия, произведенные с их помощью, это облегчит использование программы.

Интерфейс должен быть построен на основе меню. Это необходимо для создания интуитивно понятного интерфейса для пользователя.

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

Выбор решения

При запуске программы открывается приветственный экран. Далее появляется меню, состоящее из одиннадцати пунктов:

1. Показать все. На экране отобразится вся база данных.
2. Добавить запись. Пользователь вводит данные, необходимые для нового элемента базы данных в порядке, указанном в программе.
3. Удалить. Пользователь вводит номер записи, которую необходимо удалить. Программа находит данную запись и заменяет её ID на -1 для «отмены» отображения в функциях вывода. Поля при этом не заменяются.
4. Изменить. Пользователь вводит номер записи, которую необходимо изменить. Программа находит запись с данным номером и выводит ее на экран. Далее программа предлагает пользователю ввести новые данные для записи. Пользователь вводит новые данные, и программа сохраняет новую запись в базе данных.
5. Поиск. Пользователь вводит строку и программа осуществляет поиск среди пунктов страна и столица всех записей.
6. Сортировка. Программа выводит все записи базы данных в алфавитном порядке, относительно названия страны на экран.
7. Фильтр по континенту. Пользователь вводит континент и программа осуществляет вывод подходящих записей на экран.
8. Сохранить базу данных. Пользователь вводит название файла, в который необходимо записать данные. Если данный файл уже существует, то данные в нем перезаписываются, в противном случае создается новый файл, в который записывается база данных.
9. Загрузить базу данных. Пользователь вводит название файла, из которого необходимо считать данные. Если данный файл не существует или не соответствует формату, то программа выводит ошибку, иначе считывает базу данных из файла.

10. Очистить базу. Программа полностью очищает ту базу данных, которая открыта на данный момент.

11. Выход. Выход из программы.

Навигация в меню осуществляется с помощью клавиатуры, вводом чисел от 0 до 11.

Описание разработки программы

Разработанная программа состоит из нескольких модулей, которые реализовывались в следующем порядке:

1. capitals.cpp – основной файл программы. Меню программы.
2. functions.h – файл с объявлением функций и структур
3. source.cpp – файл с функциями clean_stdin и get_str, которые отвечают за безопасное считывание вводимых пользователем данных.
4. filtnchange.cpp – файл с функциями edit_or_delete и search_or_filter, которые отвечают за удаления или редактирования записи и поиск по строке.
5. database.cpp – файл с функциями load_db, save_db, print_all и check_capacity, которые отвечают за загрузку базы данных из файла, сохранение базы данных, вывод базы данных в консоли и проверку памяти.
6. record.cpp – файл с функциями free_record, set_record_field, input_record и add_record, которые отвечают за добавление нового элемента.
7. sorting.cpp – файл с ассемблерной вставкой и функцией sort_records, которые отвечают за сортировку базы данных по алфавиту.

По порядку реализации модулей можно понять, что использовался нисходящий подход реализации программы. Этот метод основан на том, что сначала проектируются основные компоненты программы, а затем уже дорабатываются ее мелкие детали. Это позволяет упростить ее тестирование, так как каждый модуль тестируется сразу же после его реализации, что помогает 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 во время разработки и также после завершения разработки. В ходе нее было выявлено огромное количество проблем, связанных с работой с файлами, работой с памятью, работой с полями структуры.

Описание программы

Приложение Capitals.exe является основным модулем программы. При запуске программы происходит вывод заставки-приветствия, после чего пользователю необходимо выбрать пункт меню для дальнейшего использования базы данных. Выход из меню осуществляется после выбора пользователем нужного пункта.

Функция `clean_stdin()` «съедает» всё, что осталось в буфере ввода до символа новой строки. Это для того чтобы следующий `fgets/get_str` не «прочитал» пустую строку. Блок-схема функции `clean_stdin()` представлена на Рисунке 1.

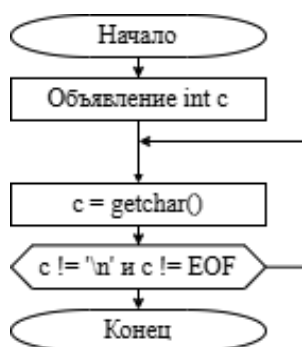


Рисунок 1 - Блок-схема функции `clean_stdin`

Функция `get_str(charbuffer)` безопасно читает строку с клавиатуры и удаляет символы `\r` и `\n` в конце. Используется везде, где нужно ввести текст. Блок-схема функции `get_str(charbuffer)` представлена на Рисунке 2.

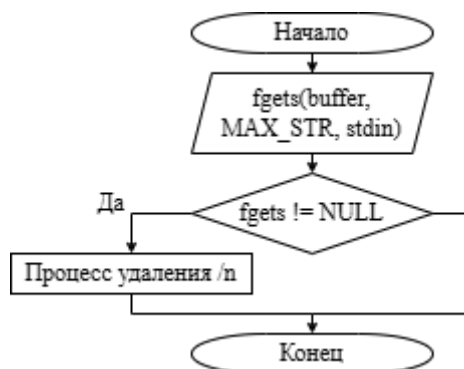


Рисунок 2 - Блок-схема функции `get_str`

Функция `edit_or_delete(Databasedb,intis_delete)` используется для удаления или редактирования записи. В начале функция проверяет, есть ли записи в базе, если пусто, то выйти. Если записи есть, то программа спрашивает

номер записи, при некорректном вводе сообщает об ошибке. Далее если выполняется удаление, то освобождается память записи, все записи после нее сдвигаются влево и счетчик уменьшается. Если это редактирование, то показываются текущие данные, запрашиваются новые, а после запись обновляется. Блок-схема функции `edit_or_delete(Database db, int is_delete)` представлена на Рисунке 3.

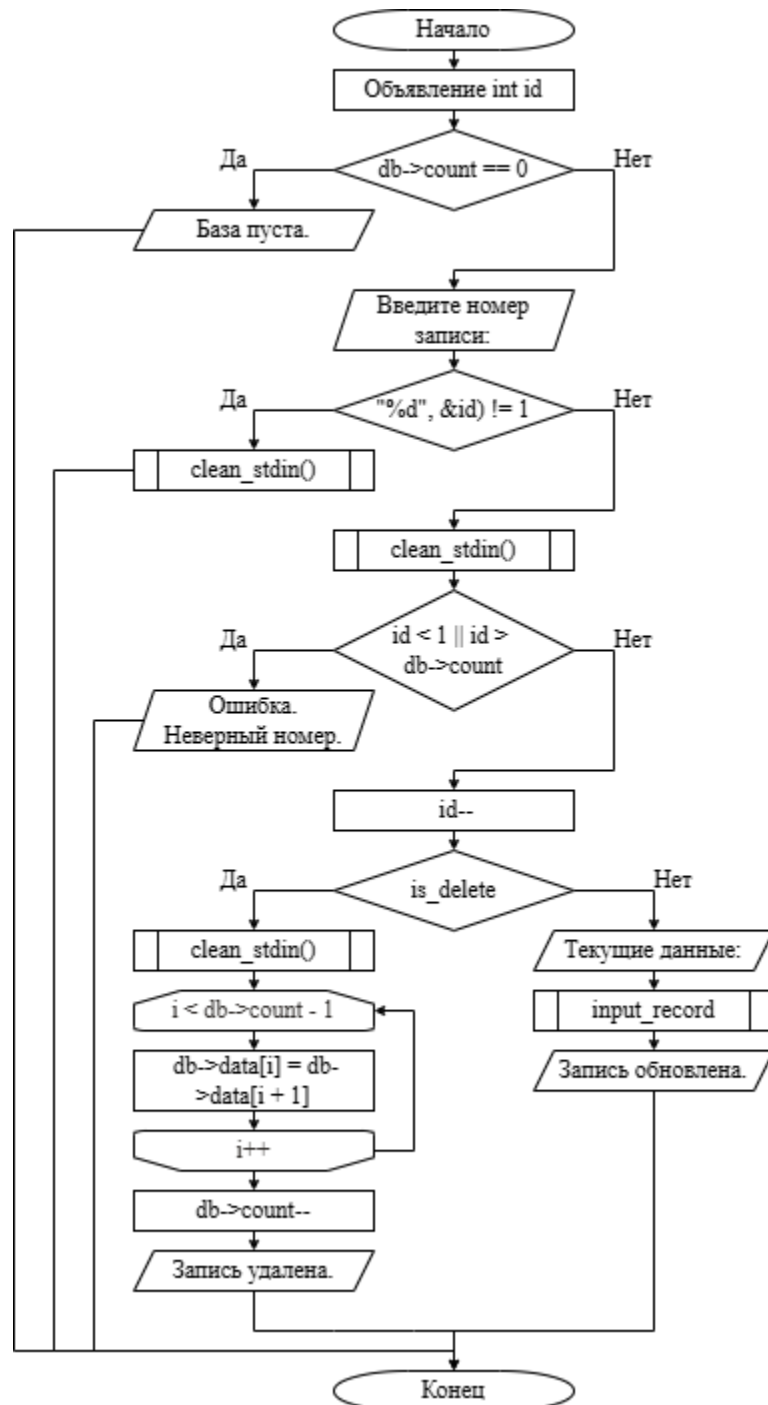


Рисунок 3 - Блок-схема функции `edit_or_delete`

Функция `search_or_filter(Database* db, int is_filter)` спрашивает у пользователя строку для поиска, с помощью цикла проходится по всей базе данных проверяя каждую запись на совпадение. Если совпадения есть то выводит на экран, в противном случае сообщает о том что ничего не найдено. Функция проверяет только страну и столицу при этом не затрагивая континент. Блок-схема функции `search_or_filter(Database* db, int is_filter)` представлена на Рисунке 4.

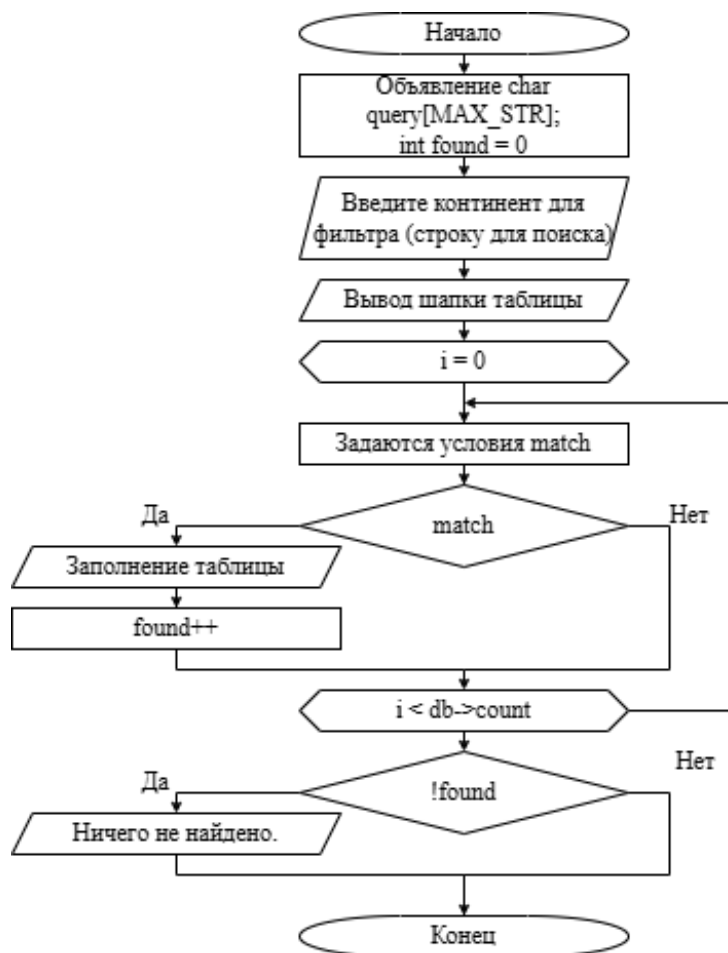


Рисунок 4 - Блок-схема функции `search_or_filter`

Функция `load_db(Database db)` загружает базу данных из файла `capitals.txt`. Если файла нет, то сообщает об ошибке и выходит. В начале открывает файл для чтения и читает файл построчно, с учетом разбивания строки на части по символу `|`. При первом вызове ей передается строка, при последующих — `NULL`, чтобы она продолжала работать с той же строкой. В цикле каждая разобранная строка преобразуется в `Record` и добавляется в базу данных, предварительно проверив ёмкость через `check_capacity`. Блок-схема функции `load_db(Database db)` представлена на Рисунке 5.

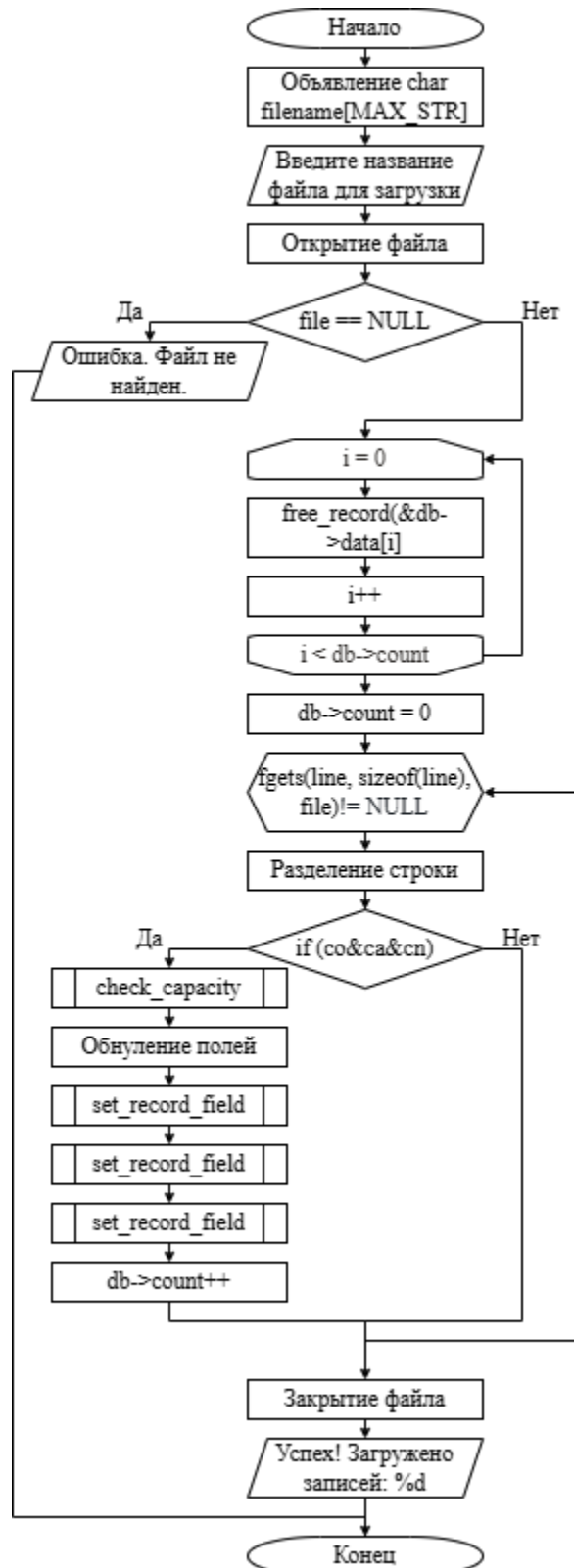


Рисунок 5 - Блок-схема функции load_db

Функция `save_db(Database* db)` открывает файл в режиме записи. Если файл по какой-то причине создать нельзя, то выдаст ошибку. Если всё хорошо, то проходится по всем записям в базе, склеивает через разделитель `|` и записывает в файл. В конце файл закрывается. Блок-схема функции `save_db(Database* db)` представлена на Рисунке 6.

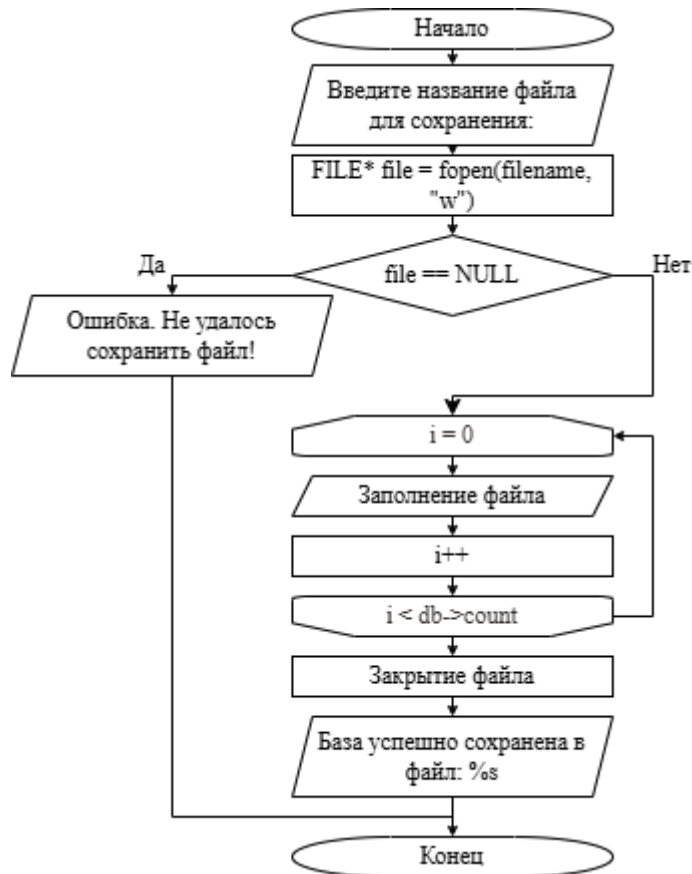


Рисунок 6 - Блок-схема функции save_db

Функция print_all(Database* db) показывает пользователю весь список стран, в виде ровной таблицы. Блок-схема функции print_all(Database* db) представлена на Рисунке 7.

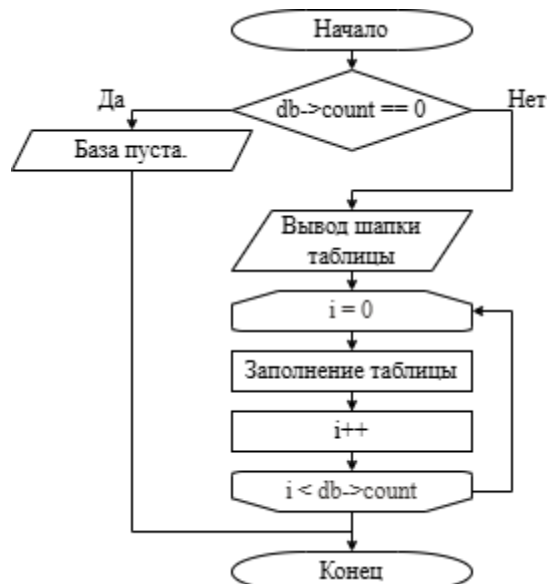


Рисунок 7 - Блок-схема функции print_all

Функция check_capacity(Database* db) Проверяет, хватит ли места. Если нет — увеличивает ёмкость в 2 раза (задаёт 5 при первом вызове).

Динамический рост массива. Без этого нельзя добавлять записи бесконечно. Блок-схема функции `check_capacity(Database* db)` представлена на Рисунке 8.

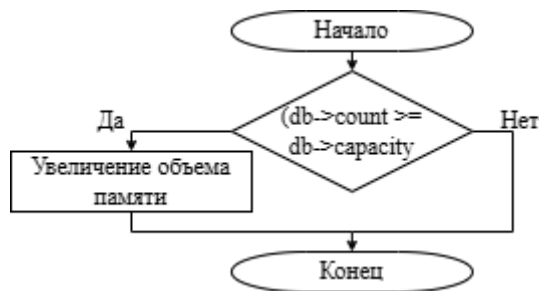


Рисунок 8 - Блок-схема функции `check_capacity`

Функция `input_record(Record* r)` запрашивает у пользователя данные (название страны, столицу и континент) и сохраняет их в переданную структуру записи. Блок-схема функции `input_record(Record* r)` представлена на Рисунке 9.



Рисунок 9 - Блок-схема функции `input_record`

Функция `free_record(Record* r)`. выполняет безопасное освобождение динамической памяти, которая ранее была выделена под строковые поля конкретной записи (страна, столица и континент). После очистки памяти функция обнуляет все указатели, чтобы предотвратить случайное обращение программы к уже удаленным данным и избежать критических сбоев. Блок-схема функции `free_record(Record* r)` представлена на Рисунке 10.

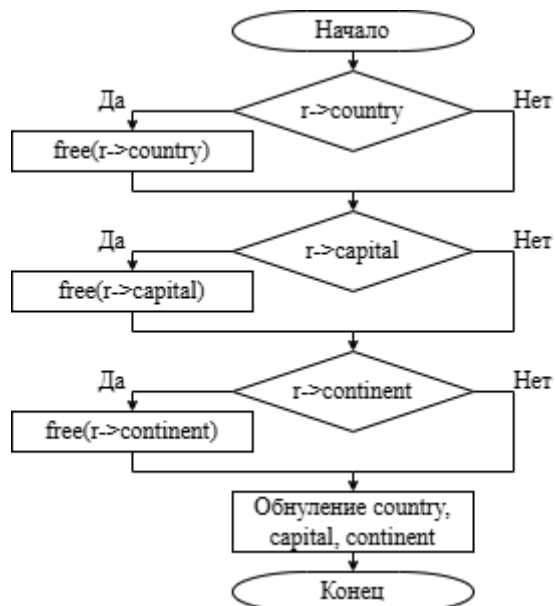


Рисунок 10 - Блок-схема функции free_record

Функция `set_record_field(char** field, constchar* value)` выполняет безопасное обновление динамического текстового поля. Она удаляет старое значение, выделяет ровно столько новой памяти, сколько требует новый текст, и копирует его туда. Блок-схема функции `set_record_field(char** field, const char* value)` представлена на Рисунке 11.

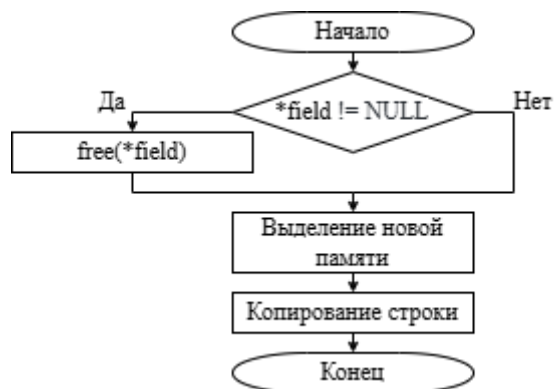


Рисунок11 - Блок-схема функции set_record_field

Функция `add_record(Database* db)` создает и добавляет новую запись в базу данных. Блок-схема функции `add_record(Database* db)` представлена на Рисунке 12.



Рисунок 12 - Блок-схема функции add_record

Функция sort_records(Database* db) выстраивает все страны строго по алфавиту. Здесь используется ассемблерная вставка asm_strcmp(const char* s1, const char* s2), которая сортирует записи методом Bubble Sort. Блок-схема ассемблерной вставки asm_strcmp(const char* s1, const char* s2) представлена на Рисунке 13. Блок-схема функции sort_records(Database* db) представлена на Рисунке 14.

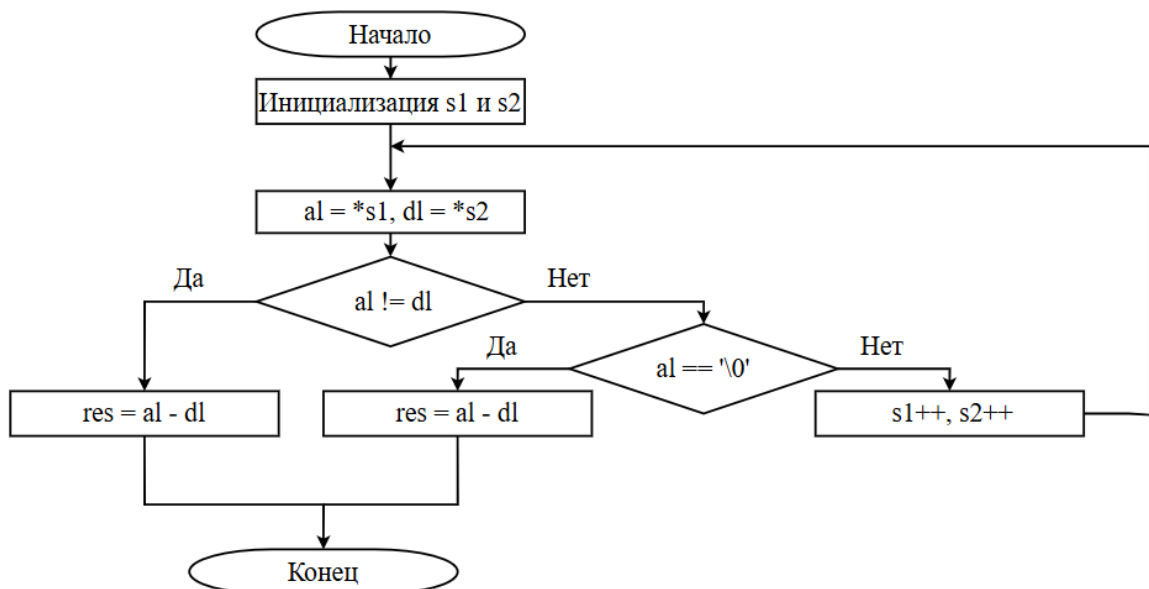


Рисунок 13 - Блок-схема функции asm_strcmp

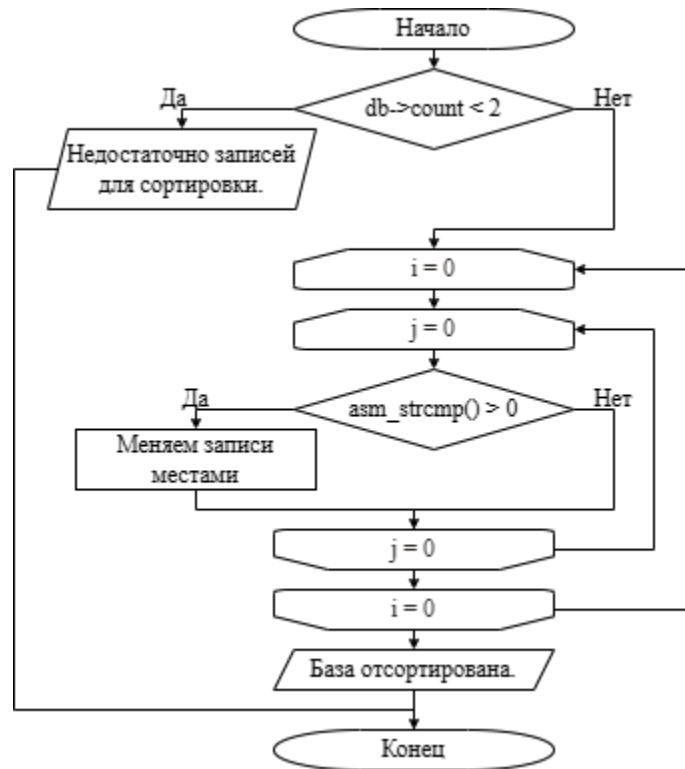


Рисунок 14 - Блок-схема функции sort_records

Основная функция программы main() объединяет все описанные выше функции. При запуске выводится меню, где каждый пункт отвечает за определенное действие и содержит в себе необходимые функции. Также при выборе пользователем пункта меню функция очищает консоль. Блок-схема функции main() представлена на Рисунках 15, 16, 17.



Рисунок 15 - Блок-схема функции `main`

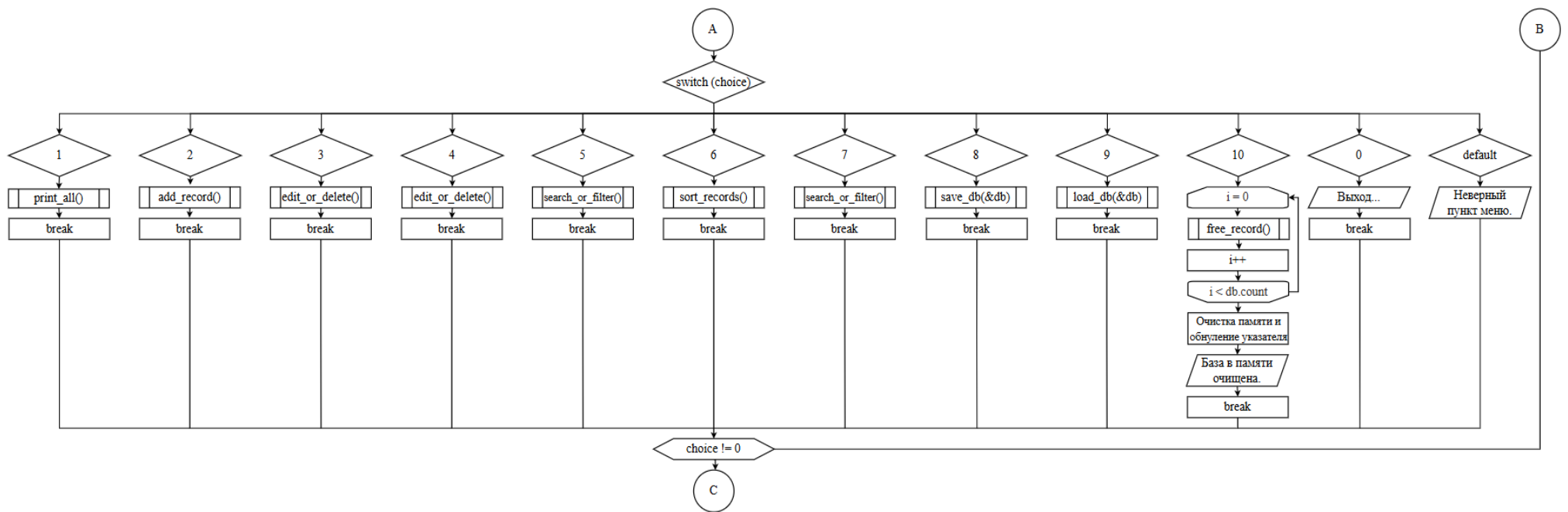


Рисунок 16 - Блок-схема функции main

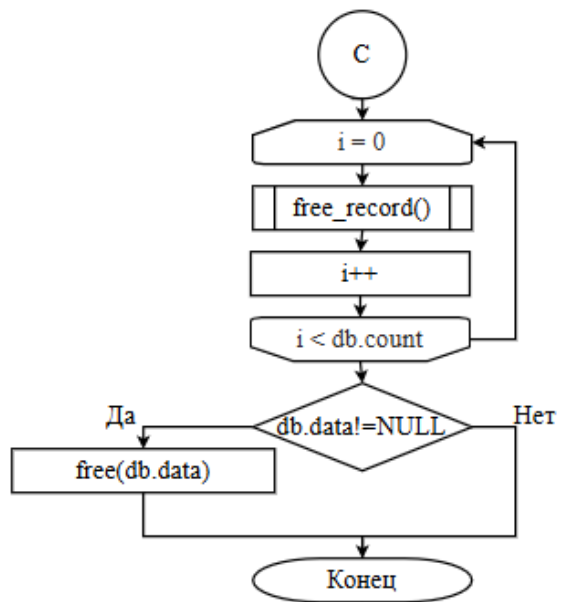


Рисунок 17 - Блок-схема функции main

Руководство пользователя

Программа capitals.cpp предназначена для хранения информации о странах их столицах и континентах на котором они располагаются. Программа имеет интуитивно понятный интерфейс и поддерживает такие операции как добавление новой записи в базу данных, удаление записи из базы данных, изменение записи в базе данных, поиск записи в базе данных, сохранение существующей базы данных в файл, импорт базы данных из файла, сортированный вывод записей базы данных и фильтрация записей в базе данных.

Для открытия и вывода на экран существующего каталога (Рисунок 18) необходимо выбрать пункт меню «Показать все». Результат представлен на Рисунке 19.

```
=== МЕНЮ ===
1. Показать все
2. Добавить
3. Удалить
4. Изменить
5. Поиск
6. Сортировка
7. Фильтр по континенту
8. Сохранить
9. Загрузить базу
10. Очистить текущую базу
0. Выход
Выбор:
```

Рисунок 18 – Главное меню

№	СТРАНА	СТОЛИЦА	КОНТИНЕНТ
1	россия	москва	евразия
2	австралия	канберра	австралия
3	сша	вашингтон	северная америка
4	канада	торонто	северная америка

Рисунок 19 – Вывод таблицы

Для добавления записи в каталог необходимо выбрать пункт меню «Добавить». Далее необходимо заполнить форму структуры из трех пунктов: ввести название страны, нажать клавишу Enter, ввести столицу страны, нажать клавишу Enter, ввести континент, нажать клавишу Enter. В случае если не

возникли ошибки на экране появится надпись «Запись добавлена!». (Рисунок 20)

```
--- Добавление записи ---  
Страна: россия  
Столица: москва  
Континент: евразия  
Запись добавлена!
```

Рисунок 20 – Добавление записи

Для удаления записи из каталога необходимо выбрать пункт меню «Удалить». Далее необходимо ввести номер удаляемой записи и нажать клавишу Enter. В случае если не возникли ошибки на экране появится надпись «Запись удалена!» (Рисунок 21).

```
Введите номер записи (1-4): 4  
Запись удалена.
```

Рисунок 21 – Результат удаления записи

Для изменения записи необходимо выбрать пункт «Изменить» в меню. Далее необходимо ввести номер изменяемой записи и ввести новые данные для экземпляра из трех пунктов: ввести название страны, нажать клавишу Enter, ввести столицу страны, нажать клавишу Enter, ввести континент, нажать клавишу Enter. В случае если не возникли ошибки на экране появится надпись «Запись обновлена» (Рисунок 22).

```
Введите номер записи (1-2): 2  
Текущие данные: италия, рим, евразия  
Страна: австралия  
Столица: канберра  
Континент: австралия  
Запись обновлена.
```

Рисунок 22 – Обновление записи

Для того чтобы найти определенную запись или группу записей необходимо выбрать пункт меню «Поиск». Далее необходимо ввести строку которая присутствует в названии или столице страны, которая ищется, нажать клавишу Enter (Рисунок 23).

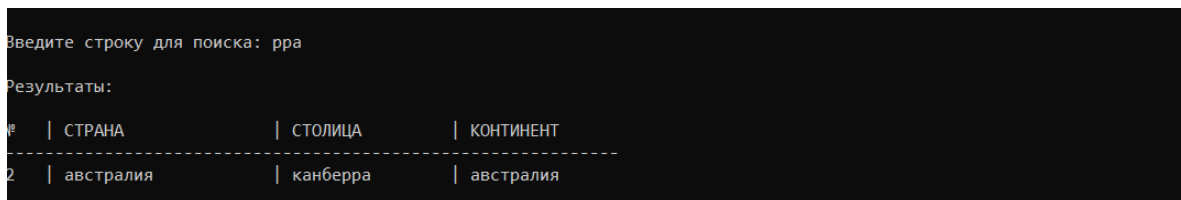


Рисунок 23 – Результат поиска

Для того чтобы записи были выведены в отсортированном виде необходимо в главном меню выбрать пункт «Сортировка». Текущая база данных будет отсортирована в алфавитном порядке (Рисунок 24).

```
База отсортирована.
```

Рисунок 24 – Результат сортировки

Для того чтобы записи были выведены в отфильтрованном виде необходимо в главном меню выбрать пункт «Фильтр по континенту». Далее необходимо ввести континент, страны которого нужно вывести на экран, нажать клавишу Enter (Рисунок 25).

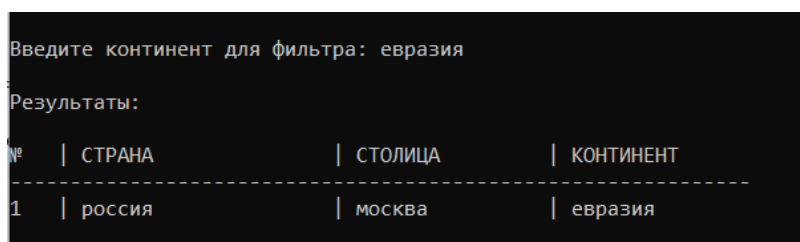


Рисунок 25 – Результат фильтрации

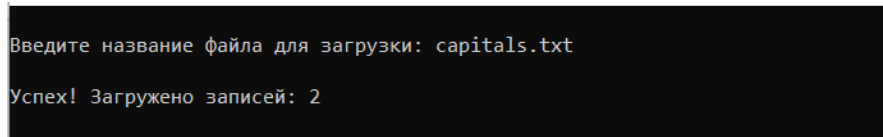
Для того чтобы сохранить текущую базу данных в файл необходимо выбрать пункт меню «Сохранить базу данных». Далее ввести имя файла, в который будет записана база данных. Если файл уже существует, то программа удалит информацию из файла и запишет в него каталог. В случае если не возникли ошибки на экране появится надпись «База данных успешно сохранена в файл:» (Рисунок 26).

```
Введите название файла для сохранения: capitals.txt
База успешно сохранена в файл: capitals.txt
```

Рисунок 26 – Результат сохранения

Для того чтобы импортировать базу данных из файла необходимо выбрать пункт меню «Загрузить базу данных». Далее ввести имя файла, из

которого произойдет чтение данных, нажать клавишу Enter. Текущая база данных будет заменена на новую, из файла. В случае ошибки программа выдаст сообщение «Ошибка. Файл не найден». В случае если не возникли ошибки на экране появится надпись «Успех! Загружено записей:» (Рисунок 27).



```
Введите название файла для загрузки: capitals.txt
Успех! Загружено записей: 2
```

Рисунок 27 – Результат загрузки

Для того чтобы очистить текущую базу данных из файла необходимо выбрать пункт меню «Очистить текущую базу данных». Текущая база данных будет очищена и на экране появится надпись «База в памяти очищена» (Рисунок 28).



```
База в памяти очищена.
```

Рисунок 28 – Результат очистки

Для выхода из программы необходимо выбрать пункт меню «Выход» и нажать клавишу Enter.

Заключение

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ. Была освоена работа с файлами, изучены функции работы с консолью и оперативной памятью. Также были получены основные навыки отладки и тестирования программ и программирования в среде VisualStudio 2022 на языках Си/C++ и Assembler.

В рамках курсовой работы была написана программа для работы с базой данных сотовых операторов. Она имеет все необходимые функции для работы с базами данных.

Список используемой литературы

1. ГОСТ 19.701-90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения : межгосударственный стандарт : дата введения 1992-01-01 / Изд. официальное. – Москва : Стандартиформ, 2010. – 23 с
2. Аблязов Р. З. Программирование на ассемблере на платформе x86-64. 2016 г. / Р. З. Аблязов; Москва: ДМ К Пресс, 2011. – 304 с.: ил.
3. Керниган Б. Ритчи Д. Язык программирования С. 1985 г. – 290 с.
4. Прата С. Язык программирования C++. Лекции и упражнения / 6-е изд. : Москва : И.Д. Вильямс", 2012. – 1248 с.
5. Тюгашев А. А. Языки программирования. Стандарт третьего поколения : Учебное пособие / А. А. Тюгашев; СПб.: Питер, 2021. – 336 с.: ил. – (Учебник для вузов).