

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

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

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

Выполнил:

студент группы ФВВЗ

Рузов А.Е. 23

Принял:

Токарев А.Н.

Пенза, 2025

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

Дубову Александру Евгеньевичу

Группа

24 ВВВ2

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

Исходные данные (технические требования) на проектирование

*Всплывающие заголовки*

- добавление, удаление, редактирование, поиск и сортировка записей
- фильтрация записей по определенному критерию
- создание новой базы данных
- сохранение текущей базы данных
- работа с определенными баз. файлами

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 "13" 13 12 2025

Дата защиты проекта "29" 29 12 2025

Руководитель С.И.

Задание получил "13" 13 12 2025 г.

Студент Дубов Александр Евгеньевич

## Содержание

|                                     |    |
|-------------------------------------|----|
| Введение.....                       | 5  |
| Постановка задачи.....              | 6  |
| Выбор решения.....                  | 7  |
| Описание разработки программы.....  | 8  |
| Отладка и тестирование.....         | 10 |
| Описание программы.....             | 11 |
| Руководство пользователя.....       | 21 |
| Заключение.....                     | 33 |
| Список используемых источников..... | 34 |
| Приложение А.....                   | 35 |
| Приложение В.....                   | 43 |
| Приложение С.....                   | 44 |

## **Введение**

В современном мире информация стала важнейшим ресурсом, который требует систематизации для эффективной работы как отдельных людей, так и целых обществ. Благодаря развитию компьютерных технологий мы получили возможность значительно упростить обработку и хранение данных в различных сферах деятельности.

База данных – это организованная структура, содержащая структурированные данные, специально подготовленные для быстрой и удобной обработки с помощью компьютера. Рассматривая эволюцию баз данных в их современном понимании, мы можем проследить интересный путь развития.

Началом эры баз данных принято считать 1955 год, когда появилось первое программируемое оборудование для обработки записей с использованием перфокарт. В середине 1960-х годов произошел качественный скачок – появились сетевые базы данных, которые можно было обрабатывать через терминалы в режиме реального времени. Простые системы хранения данных постепенно эволюционировали в более сложные и эффективные модели.

Сегодня базы данных – это неотъемлемый элемент работы любой организации. Они активно используются на сайтах, в стриминговых сервисах, банковских системах для хранения и обработки огромных массивов информации о пользователях.

Разработанная программа представляет собой удобный инструмент для работы с базой данных расписания занятий. С её помощью любой пользователь может быстро получить полную информацию о предстоящих занятиях, включая время проведения, день недели, название предмета и данные преподавателя.

## **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 десяти пунктов:

1.     Добавить занятие. Пользователь вводит данные, необходимые для нового элемента базы данных в порядке, указанном в программе.
2.     Удалить занятие. Пользователь может удалить любую запись по ID.
3.     Редактирование занятий. Пользователь вводит ID занятия которое хочет отредактировать, после этого в программе он вводит новые данные(информацию о занятиях, преподавателе, дне и времени).
4.     Найти занятие. Пользователь может найти по номеру ID нужное ему занятие.
5.     Сортировать занятия. Пользователь может отсортировать по номеру ID в порядке возрастания.
6.     Фильтр по дням. Пользователь может найти все нужные занятия, которые проходят в один день, вписав нужную дату.
7.     Сохранить в файл. Пользователь может сохранить базу данных в отдельном файле, для этого ему нужно всего лишь дать название файлу.
8.     Загрузить из файла. Пользователь может загрузить уже имеющуюся базу данных в программу, для этого ему нужно указать название файла.
9.     Показать все занятия. Пользователь нажав эту кнопку может узнать всю информацию о всех занятиях.
10.    Выход. Завершение работы программы.

## Описание разработки программы

Разработанная программа представляет собой систему управления базой данных учебных занятий и состоит из следующих модулей:

**main.cpp** – основной файл программы, содержащий главное меню и логику взаимодействия с пользователем.

**func.h** – заголовочный файл, определяющий структуру данных и объявления функций для работы с базой данных.

Модули программы:

**main.cpp** – основной модуль программы. Включает:

1. Функцию `main`, которая обеспечивает взаимодействие с пользователем через консольное меню.
2. Функции для отображения данных (`displayLessons`) и меню (`displayMenu`).
3. Обработку пользовательского ввода и вызов соответствующих функций для работы с базой данных.

**func.h** – заголовочный файл, содержащий:

1. Определение структуры `Lesson`, которая хранит информацию о занятиях (ID, предмет, преподаватель, день, время).
2. Объявления функций для работы с базой данных, таких как добавление, удаление, редактирование, поиск, сортировка, сохранение и загрузка данных.

Функции для работы с данными (реализованы в `main.cpp`):

1. `addLesson` – добавление нового занятия в базу данных с проверкой корректности ID.
2. `deleteLesson` – удаление занятия по ID.
3. `editLesson` – редактирование существующего занятия.
4. `searchLesson` – поиск занятия по ID.
5. `sortLessons` – сортировка занятий по ID.
6. `saveToFile` – сохранение данных в файл.

7. `loadFromFile` – загрузка данных из файла.
8. `filterByDay` – фильтрация занятий по дню недели.
9. `displayMenu` – вывод меню.
10. `displayLessons` – отображение списка занятий в консоли.

Принцип разработки:

Программа разрабатывалась с использованием нисходящего подхода. Сначала были реализованы основные компоненты (структура данных и базовые функции для работы с ними), а затем добавлены дополнительные возможности, такие как сортировка, фильтрация и работа с файлами. Такой подход позволил упростить тестирование и обеспечить модульность кода.

Особенности:

Использование стандартной библиотеки C++ для работы с данными (`vector`, `algorithm`, `fstream` и др.).

1. Поддержка русского языка в консоли (`setlocale`).
2. Простое и понятное меню для пользователя.

Программа предназначена для управления учебными занятиями и может быть легко расширена для добавления новых функций, таких как дополнительные фильтры или сортировки.

## **Отладка и тестирование**

Для создания программного продукта была использована современная интегрированная среда разработки Microsoft Visual Studio 2022, которая предоставляет полный набор инструментов для эффективной разработки и отладки приложений.

В процессе отладки программного обеспечения активно применялись различные встроенные инструменты Visual Studio, включая установку точек останова для приостановки выполнения программы, трассировку исполнения кода и детальный анализ значений переменных в режиме реального времени.

Процесс тестирования осуществлялся на протяжении всего цикла разработки – как в ходе создания программы, так и после завершения основной работы. В результате тестирования был выявлен целый ряд технических проблем, связанных с:

1. Обработкой файлов.
2. Управлением памятью.
3. Работой со структурными данными и их полями.

Все обнаруженные проблемы были тщательно проанализированы и устранены для обеспечения стабильной работы программного продукта.

## Описание программы

Программа main.cpp 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. Описание состояний программы выполнено в таблице 1.

**Таблица 1 – действия в меню.**

| Клавиши, вызывающее событие | Действие пользователя                 | Действие программы                                                                         |
|-----------------------------|---------------------------------------|--------------------------------------------------------------------------------------------|
| 1                           | Выбран пункт: “Добавить занятие”      | Запускается диалог добавления занятий и удаления записи из БД.(см. рис. 3)                 |
| 2                           | Выбран пункт: “Удалить занятие”       | Запускается диалог удаления занятий и удаление занятий из БД.(см. рис. 4)                  |
| 3                           | Выбран пункт: “Редактировать занятие” | Запускается диалог изменения занятия и изменение занятий из БД.(см. рис. 5)                |
| 4                           | Выбран пункт: “Найти занятие”         | Программа считывает ID и выводит подходящие записи. (см. рис. 6)                           |
| 5                           | Выбран пункт: “Сортировать занятие”   | Записи сортируются по величине ID от меньшего к большему и выводятся на экран (см. рис. 7) |
| 6                           | Выбран пункт: “Фильтр по дням”        | Программа считывает дату и выводит подходящие записи.(см. рис. 10)                         |
| 7                           | Выбран пункт: “Сохранить в файл”      | Запускается диалог и запись существующей БД в файл. (см. рис. 8)                           |
| 8                           | Выбран пункт: “Загрузить из файла”    | Запускается диалог и загрузка БД из файла. (см. рис. 9)                                    |
| 9                           | Выбран пункт: “Показать все занятия”  | На экран выводятся все записи находящиеся в базе данных                                    |

|    |                       |                                      |
|----|-----------------------|--------------------------------------|
| 10 | Выбран пункт: “Выход” | Выполняется выход в<br>главное меню. |
|----|-----------------------|--------------------------------------|

На рисунке 1 изображена схема программы.

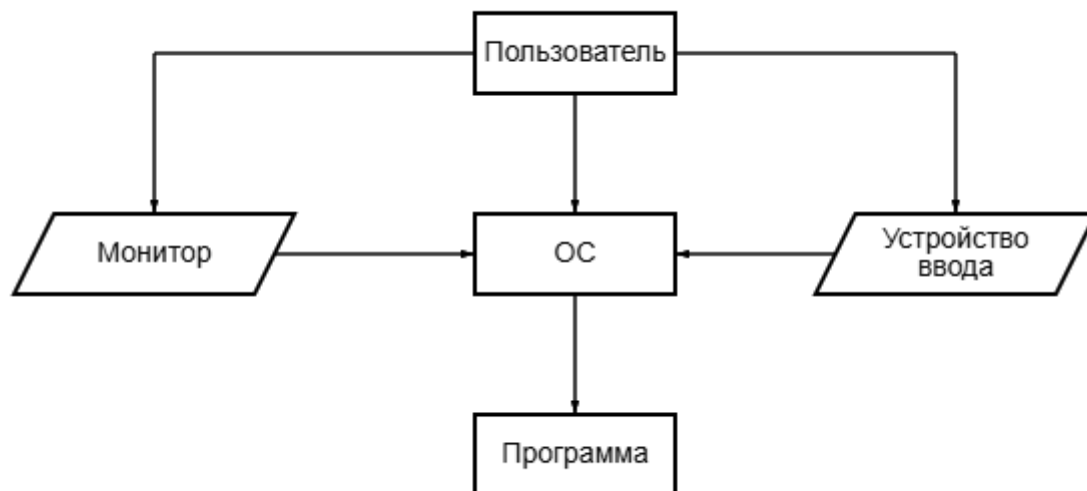


Рисунок 1.

На рисунке 2 изображена схема работы программы.

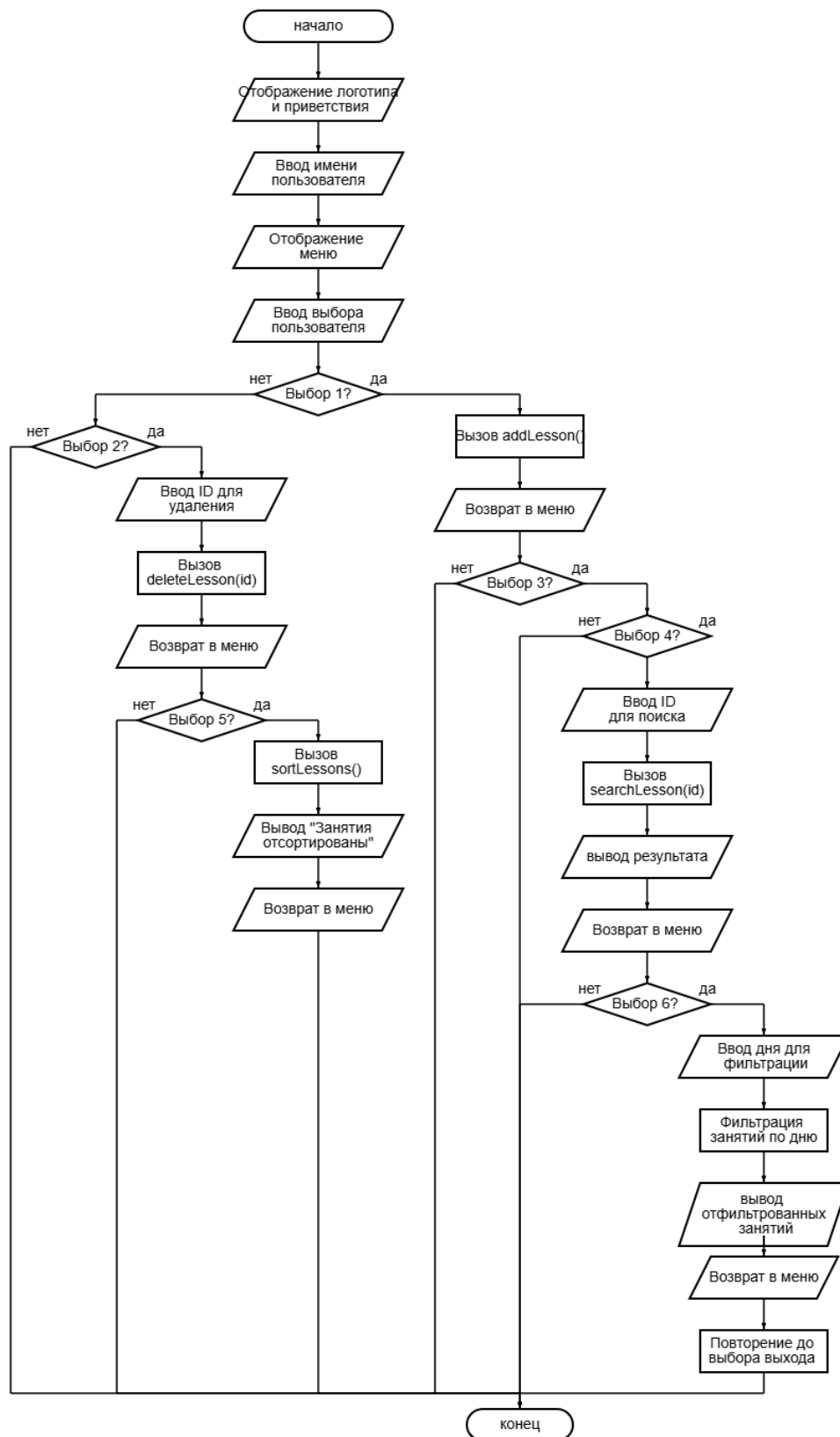


Рисунок 2 функции -- main.

На рисунке 3 изображена работа функции, которая добавляет занятия в БД.

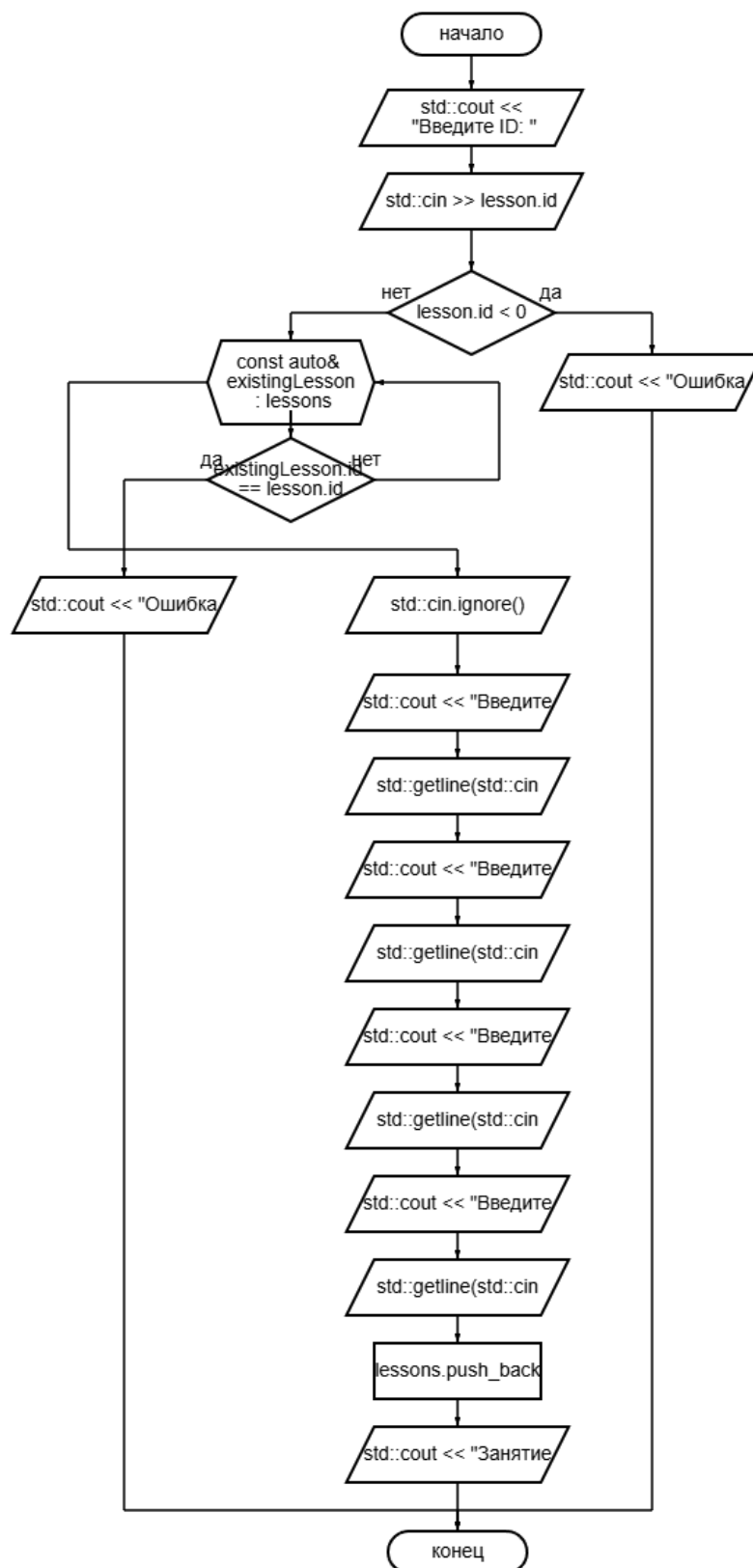


Рисунок 3 функции — add\_lesson.

На рисунке 4 изображена функция, которая удаляет записи из БД.

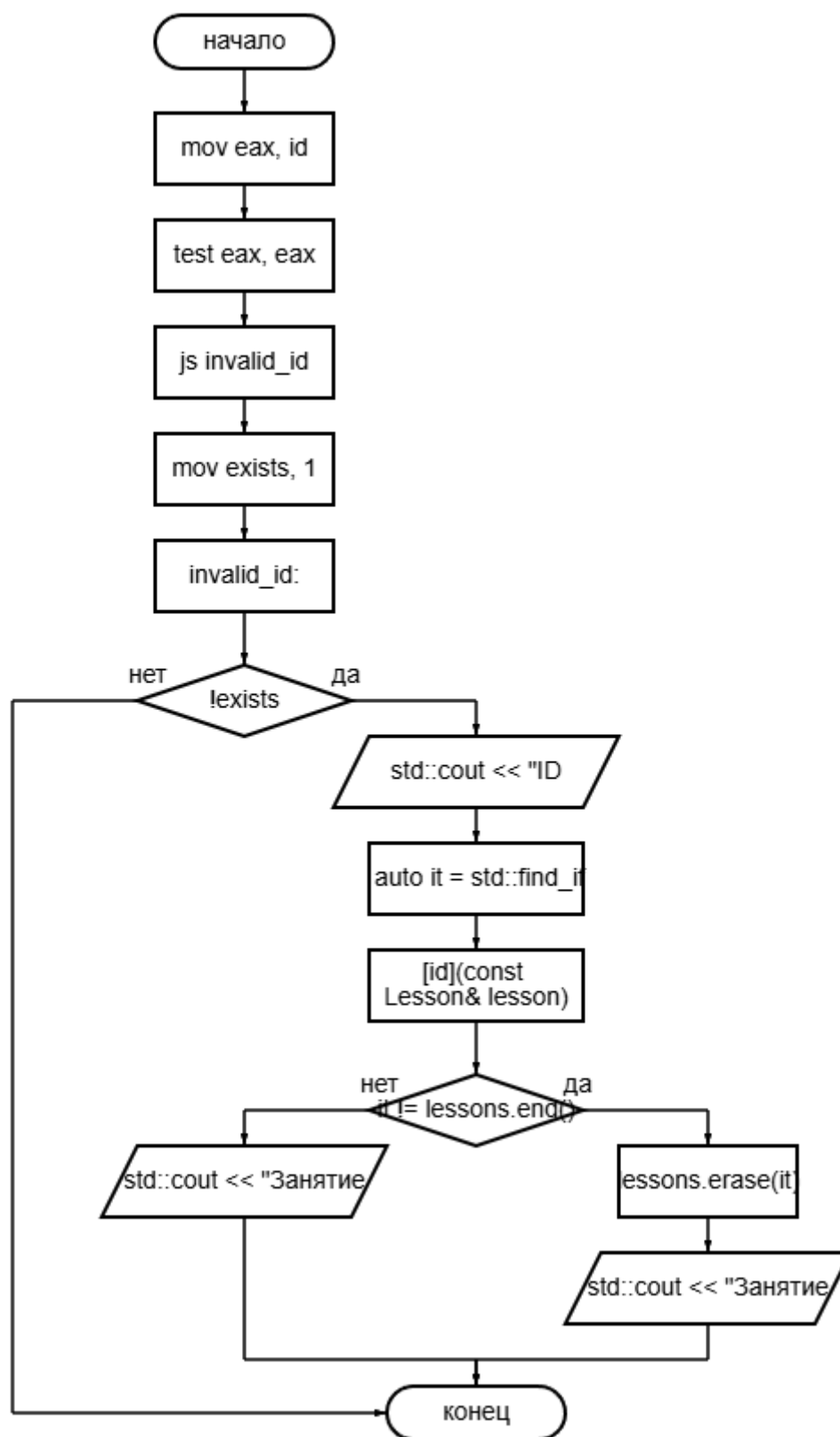


Рисунок 4 функции — deleteLesson.

На рисунке 5 изображена функция, которая редактирует существующие записи в БД.

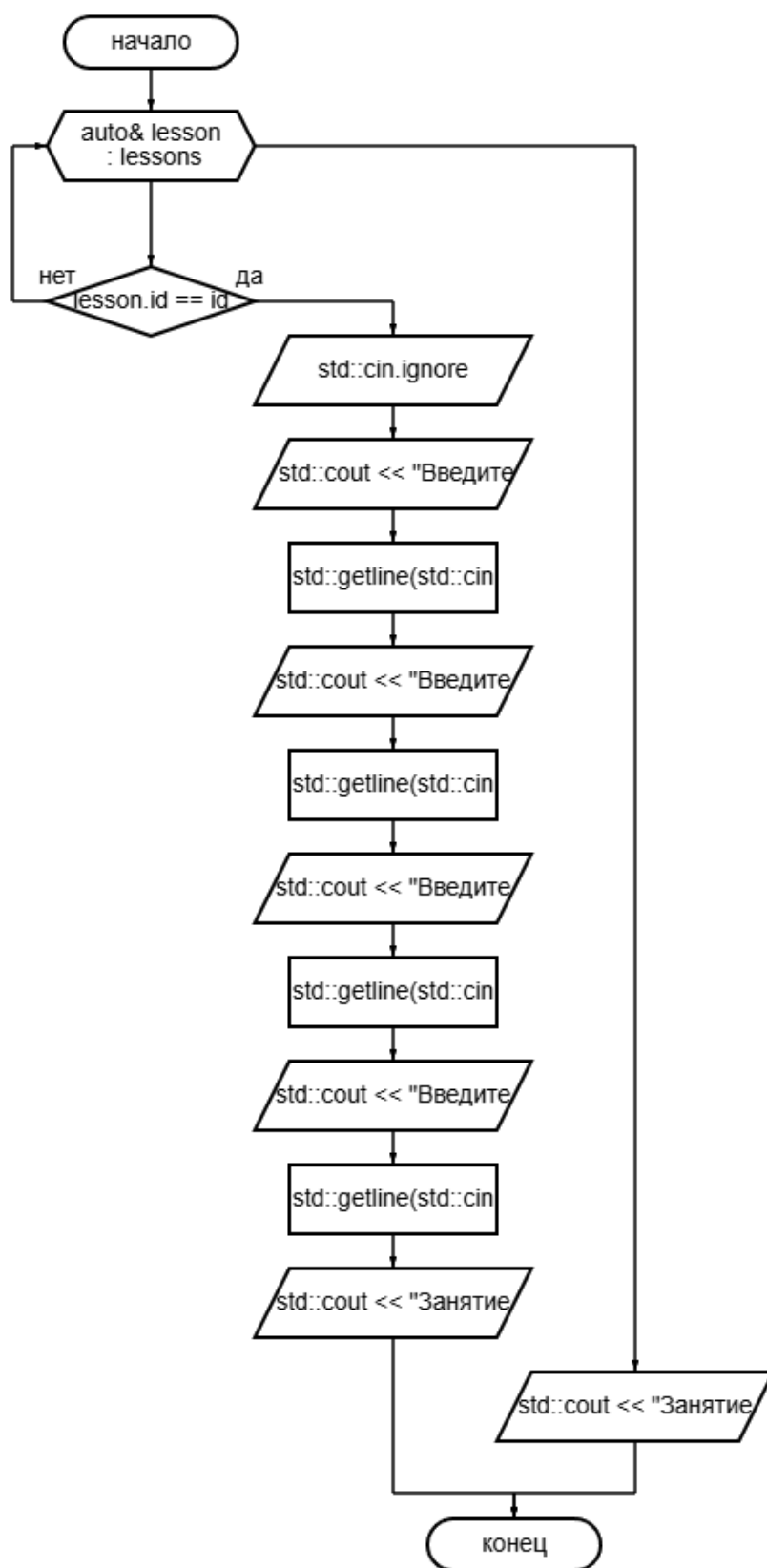


Рисунок 5 функции — editLesson.

На рисунке 6 изображена функция, которая ищет по ID занятия в БД.

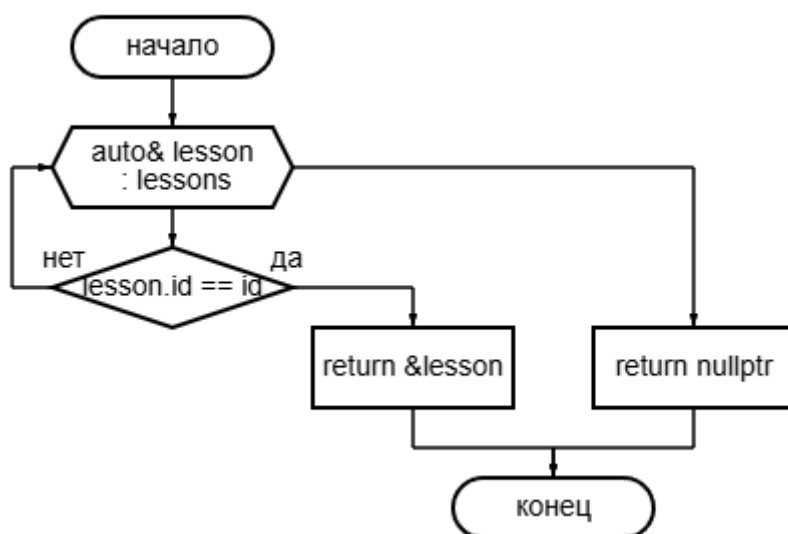


Рисунок 6 функции — `searchLesson`.

На рисунке 7 изображена функция, которая сортирует записи в БД.

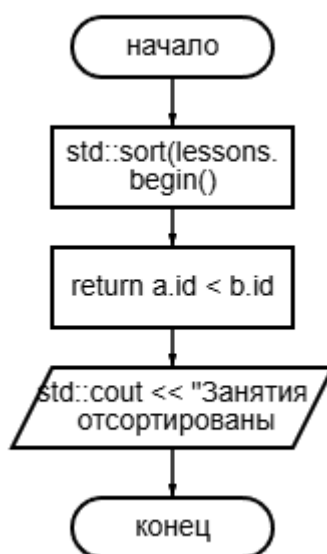


Рисунок 7 функции — `sortLessons`.

На рисунке 8 изображена функция, которая сохраняет записи в БД.

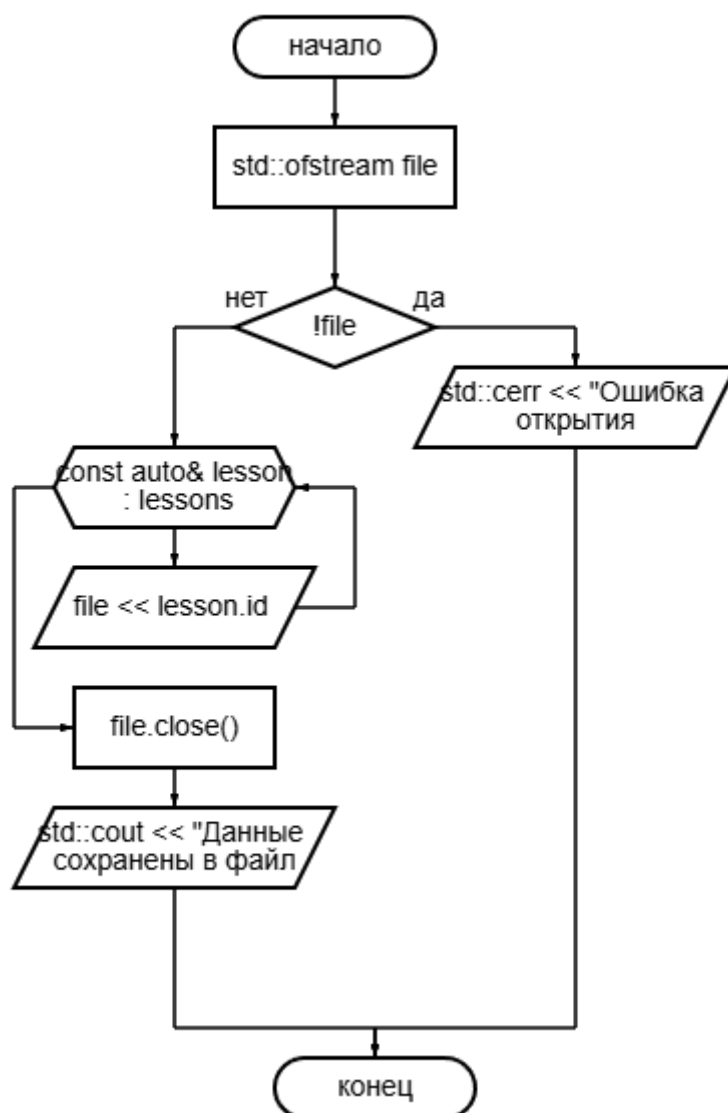


Рисунок 8 функции — saveToFile.

На рисунке 9 изображена функция, которая загружает записи из БД в программу.

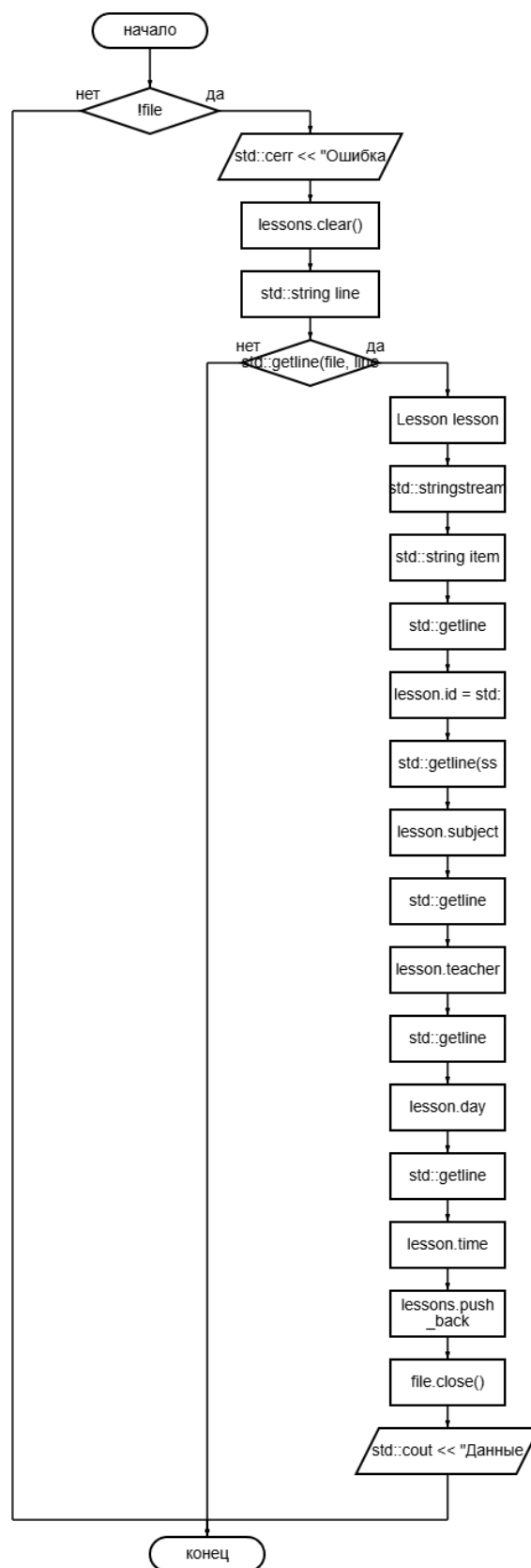


Рисунок 9 функции — loadFromFile.

На рисунке 10 изображена функция, которая фильтрует записи из БД.

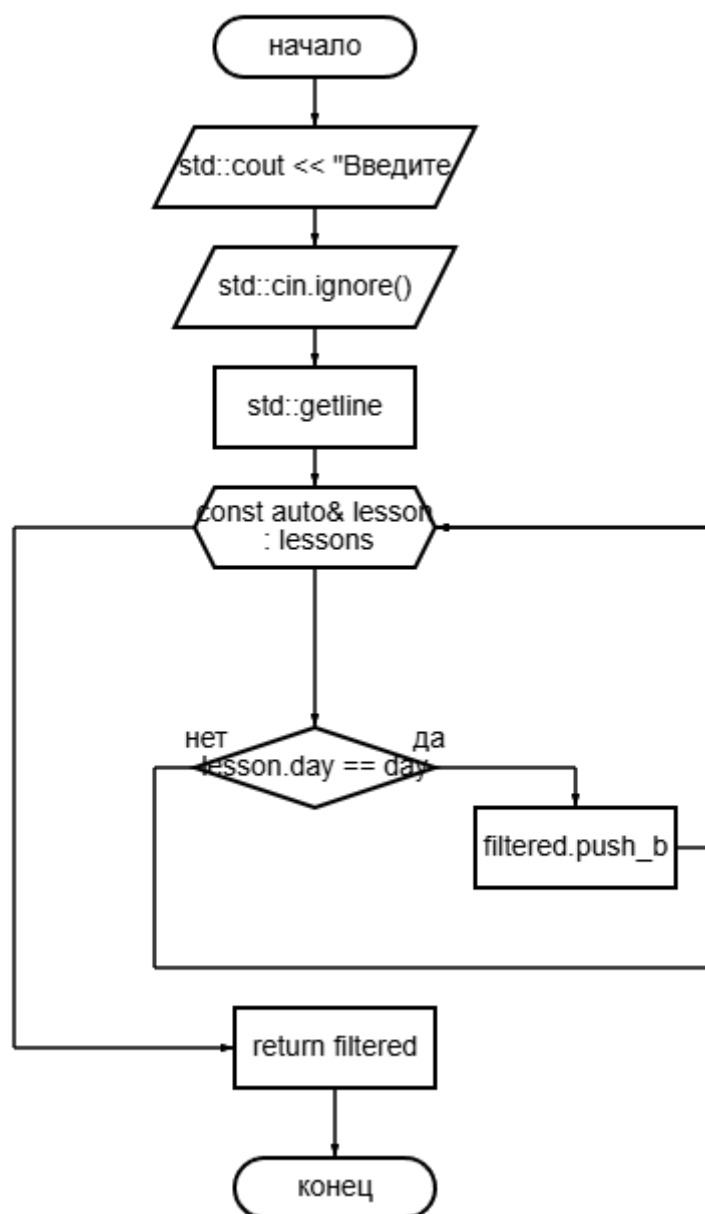


Рисунок 10 функции — `filterByDay`.

## **Руководство пользователя**

Main.exe представляет собой удобное приложение для управления учебным расписанием. В системе можно вести учет занятий, времени их проведения, дат и информации о преподавателях.

Функционал программы включает широкий набор инструментов для работы с данными:

1. Создание новых записей в базе.
2. Удаление неактуальной информации.
3. Редактирование существующих данных.
4. Поиск нужных записей по различным параметрам.
5. Экспорт базы данных в файл для резервного копирования.
6. Импорт данных из внешних файлов.
7. Сортировка записей по заданным критериям.
8. Фильтрация информации для быстрого поиска нужных данных.

Интерфейс приложения разработан с учетом удобства использования, что позволяет быстро освоить все возможности программы даже неопытным пользователям. Чтобы добавить занятие, нужно выбрать в меню «Добавить занятие» и нажать на клавиатуре цифру «1». Затем следует указать информацию (идентификатор, предмет, преподавателя, дату и время), как это показано на рисунке 11.

```
Меню:
1.   Добавить занятие
2.   Удалить занятие
3.   Редактировать занятие
4.   Найти занятие
5.   Сортировать занятия
6.   Фильтр по дням
7.   Сохранить в файл
8.   Загрузить из файла
9.   Показать все занятия
10.  Выход
Выберите действие: 1
Введите ID: 1
Введите предмет: math
Введите преподавателя: Ivanov I.I.
Введите день: 23 april
Введите время: 12:00
Занятие успешно добавлено.
```

Рисунок 11 — Добавление действия.

Если пользователь введет занятый ID программа предупредит о том, что такой ID уже существует, показано на рисунке 12.

```
Меню:
1.   Добавить занятие
2.   Удалить занятие
3.   Редактировать занятие
4.   Найти занятие
5.   Сортировать занятия
6.   Фильтр по дням
7.   Сохранить в файл
8.   Загрузить из файла
9.   Показать все занятия
10.  Выход
Выберите действие: 1
Введите ID: 1
Ошибка: Занятие с ID 1 уже существует.
```

Рисунок 12 — Ошибка при вводе существующего ID.

Если пользователь введет ID с отрицательным значением то программа предупредит о том, что ID не может быть отрицательным, показано на рисунке 13.

```
Меню:  
1.  Добавить занятие  
2.  Удалить занятие  
3.  Редактировать занятие  
4.  Найти занятие  
5.  Сортировать занятия  
6.  Фильтр по дням  
7.  Сохранить в файл  
8.  Загрузить из файла  
9.  Показать все занятия  
10. Выход  
Выберите действие: 1  
Введите ID: -1  
Ошибка: ID не может быть отрицательным.
```

Рисунок 13 — Ошибка при вводе отрицательного ID.

Если пользователь хочет удалить запись из базы данных, ему нужно выбрать пункт «Удалить занятие» в меню и нажать цифру «2». Затем следует ввести ID занятия, которое требуется удалить, показано на рисунке 14.

```
Меню:  
1.  Добавить занятие  
2.  Удалить занятие  
3.  Редактировать занятие  
4.  Найти занятие  
5.  Сортировать занятия  
6.  Фильтр по дням  
7.  Сохранить в файл  
8.  Загрузить из файла  
9.  Показать все занятия  
10. Выход  
Выберите действие: 2  
Введите ID для удаления: 1  
Занятие с ID 1 удалено.
```

Рисунок 14 — Удаление записи.

Если пользователь введет несуществующий ID, то программа выведет запись о том, что занятие с таким ID не найдено, показано на рисунке 15.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 2
Введите ID для удаления: 2
Занятие с таким ID не найдено.
```

Рисунок 15 — Ошибка при вводе несуществующего ID.

Если пользователь введет отрицательный ID, то программа выведет запись о том, что ID не может быть отрицательным, показано на рисунке 16.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 2
Введите ID для удаления: -1
ID не может быть отрицательным.
```

Рисунок 16 — Ошибка при вводе отрицательного ID.

Чтобы изменить параметры занятия, выберите в меню пункт «Редактировать занятие» (цифра «3»). Далее введите существующий ID и новые данные, такие как предмет, преподаватель, дата и время, показано на рисунке 17, 18.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 1
Введите ID: 1
Введите предмет: math
Введите преподавателя: Igor
Введите день: 11 may
Введите время: 12:10
Занятие успешно добавлено.
```

Рисунок 17 — Добавления занятия.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 3
Введите ID для редактирования: 1
Введите новый предмет: biology
Введите нового преподавателя: Shaman
Введите новый день: 9 mart
Введите новое время: 11:20
Занятие отредактировано.
```

Рисунок 18 — Редактирование занятия.

Если пользователь введет несуществующий ID, то программа выведет сообщение о том, что занятие с таким ID не найдено, показано на рисунке 19.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 3
Введите ID для редактирования: 3
Занятие с таким ID не найдено.
```

Рисунок 19 — Ошибка при вводе несуществующего ID.

Чтобы найти занятие, пользователю нужно нажать на цифру «4» с надписью «Найти занятие». Затем следует ввести ID и просмотреть результаты в таблице, показано на рисунке 20.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 4
Введите ID для поиска: 1
Найдено:

Результат поиска:
ID  Предмет          Преподаватель    День    Время
-----
1   biology          Shaman           9 mart  11:20
```

Рисунок 20 — Нахождение занятия по ID.

Если пользователь введет занятие с несуществующим ID, то программа выведет сообщение о том, что занятие не найдено, показано на рисунке 21.

```

Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 4
Введите ID для поиска: 12
Не найдено.

```

Рисунок 21 — Ошибка при вводе несуществующего ID.

Чтобы узнать количество всех занятий в базе данных, пользователю необходимо нажать на клавиатуре цифру «9», рядом с которой написано «Показать все занятия». После нажатия на эту клавишу будет показана таблица со всеми вводными данными, показано на рисунке 22.

| Все занятия: |         |               |        |       |
|--------------|---------|---------------|--------|-------|
| ID           | Предмет | Преподаватель | День   | Время |
| 1            | biology | Shaman        | 9 mart | 11:20 |
| 3            | russian | Sasha         | 30 feb | 20:00 |
| 2            | math    | Nikita        | 10 may | 19:20 |

```

Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход

```

Рисунок 22 — Все занятия в БД.

Если же пользователь нажмет цифру “9”, но БД будет пуста, то выйдет сообщение о том, что нет данных для отображения, показано на рисунке 23.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 9
Все занятия:Нет данных для отображения.
```

Рисунок 23 — Ошибка при выводе несуществующей БД.

Чтобы отсортировать занятия, пользователю нужно нажать на цифру «5» с надписью «Сортировать занятия». Сортировка будет выполнена по возрастанию идентификаторов (ID). Для проверки результата следует нажать на клавишу «9», показано на рисунке 24, 25, 26.

```
Все занятия:
ID  Предмет      Преподаватель  День      Время
-----
1   biology      Shaman         9 mart    11:20
3   russian      Sasha          30 feb    20:00
2   math         Nikita         10 may    19:20

Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: |
```

Рисунок 24 – Вывод всех занятий.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 5
Занятия отсортированы.
```

Рисунок 25 — Сортировка занятий.

| Все занятия: |         |               |        |       |
|--------------|---------|---------------|--------|-------|
| ID           | Предмет | Преподаватель | День   | Время |
| 1            | biology | Shaman        | 9 mart | 11:20 |
| 2            | math    | Nikita        | 10 may | 19:20 |
| 3            | russian | Sasha         | 30 feb | 20:00 |

Меню:

1. Добавить занятие
2. Удалить занятие
3. Редактировать занятие
4. Найти занятие
5. Сортировать занятия
6. Фильтр по дням
7. Сохранить в файл
8. Загрузить из файла
9. Показать все занятия
10. Выход

Выберите действие: |

Рисунок 26 — Отсортированные занятия.

Чтобы отфильтровать занятия по дням, пользователю нужно нажать на цифру «6» с надписью «Фильтр по дням». Затем следует выбрать нужный день. После этого будут показаны только те занятия, которые соответствуют выбранному дню, показано на рисунке 27.

| Меню:                               |                       |               |        |       |
|-------------------------------------|-----------------------|---------------|--------|-------|
| 1.                                  | Добавить занятие      |               |        |       |
| 2.                                  | Удалить занятие       |               |        |       |
| 3.                                  | Редактировать занятие |               |        |       |
| 4.                                  | Найти занятие         |               |        |       |
| 5.                                  | Сортировать занятия   |               |        |       |
| 6.                                  | Фильтр по дням        |               |        |       |
| 7.                                  | Сохранить в файл      |               |        |       |
| 8.                                  | Загрузить из файла    |               |        |       |
| 9.                                  | Показать все занятия  |               |        |       |
| 10.                                 | Выход                 |               |        |       |
| Выберите действие: 6                |                       |               |        |       |
| Введите день для фильтрации: 9 mart |                       |               |        |       |
| Фильтрованные занятия:              |                       |               |        |       |
| ID                                  | Предмет               | Преподаватель | День   | Время |
| 1                                   | biology               | Shaman        | 9 mart | 11:20 |

Рисунок 27 — Фильтрация занятий по дням.

Если пользователь введет день фильтрации которого не существует, то программа выведет сообщение о том, что нет данных для отображения, показано на рисунке 28.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 6
Введите день для фильтрации: 3 may
Фильтрованные занятия:Нет данных для отображения.
```

Рисунок 28 — Ошибка при вводе дня, которого нет в БД.

Чтобы сохранить базу данных, пользователю нужно нажать на цифру «7» с надписью «Сохранить в файл». Затем следует указать имя файла для сохранения. Файл будет размещен в каталоге с программой, показано на рисунке 29.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 7
Введите имя файла для сохранения: enotik
Данные сохранены в файл.
```

Рисунок 29 — Сохранение БД в отдельный файл.

Чтобы пользователь смог загрузить БД в программу, ему требуется нажать цифру “8”, где напротив будет написано “Загрузить из файла”. Для того чтобы данные перенеслись из файла, нужно ввести существующее имя

файла, который находится в одной папке с программой, показано на рисунке 30.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 8
Введите имя файла для загрузки: enotik
Данные загружены из файла.
```

Рисунок 30 — Загрузка файла из существующей БД.

Если пользователь введет несуществующее имя файла, то программа выведет сообщение о том, что при открытии файла произошла ошибка, показано на рисунке 31.

```
Меню:
1.  Добавить занятие
2.  Удалить занятие
3.  Редактировать занятие
4.  Найти занятие
5.  Сортировать занятия
6.  Фильтр по дням
7.  Сохранить в файл
8.  Загрузить из файла
9.  Показать все занятия
10. Выход
Выберите действие: 8
Введите имя файла для загрузки: sasa
Ошибка открытия файла для чтения.
```

Рисунок 31 — Ошибка при вводе несуществующей БД.

Чтобы пользователь мог завершить работу с БД, ему следует нажать на цифру “10”, где напротив написано “Выход”, показано на рисунке 32.

```
Меню:  
1.  Добавить занятие  
2.  Удалить занятие  
3.  Редактировать занятие  
4.  Найти занятие  
5.  Сортировать занятия  
6.  Фильтр по дням  
7.  Сохранить в файл  
8.  Загрузить из файла  
9.  Показать все занятия  
10. Выход  
Выберите действие: 10  
  
C:\Users\sasar\pr\tt\Debug\tt.exe (процесс 24892) завершил работу с кодом 0 (0x0).  
Нажмите любую клавишу, чтобы закрыть это окно:|
```

Рисунок 32 — Окончание работы с программой.

## **Заключение**

В процессе выполнения курсовой работы был приобретен ценный опыт в области разработки программного обеспечения. Были освоены следующие ключевые компетенции:

1. Многомодульное программирование - создание сложных программных решений из отдельных взаимосвязанных модулей.
2. Работа с файлами - освоение методов обработки и хранения данных.
3. Консольное программирование - изучение функций взаимодействия с командной строкой.
4. Управление памятью - освоение принципов работы с оперативной памятью.
5. Отладка и тестирование - приобретение навыков поиска и исправления ошибок в программах.
6. Программирование - освоение языков C/C++ и Assembler в среде разработки Visual Studio 2022.

Результатом работы стала функциональная программа для управления базой данных сотовых операторов, которая включает весь базовый набор инструментов для эффективной работы с данными.

Перспективы развития программы включают:

1. Модернизацию интерфейса - внедрение графического пользовательского интерфейса.
2. Улучшение навигации - добавление поддержки управления с помощью мыши.
3. Повышение удобства - создание более интуитивного пользовательского взаимодействия.

Эти улучшения сделают программу более современной и удобной для конечных пользователей.

## Список используемых источников

1. Керниган, Б. Язык программирования С / Б. Керниган, Д. Ритчи. – 1985. – URL: [https://www.r5.org/files/books/computers/languages/c/kr/Brian\\_Kernighan\\_Dennis\\_Ritchie-The\\_C\\_Programming\\_Language-RU.pdf](https://www.r5.org/files/books/computers/languages/c/kr/Brian_Kernighan_Dennis_Ritchie-The_C_Programming_Language-RU.pdf) (дата обращения: 10.04.2025).
2. MSDN : [сайт]. – URL: <https://search.msdn.microsoft.com/> (дата обращения: 22.04.2025).
3. Прага, С. Язык программирования С++ / С. Прага. – 2019.
4. Тюгашев, А. А. Языки программирования : учебное пособие / А. А. Тюгашев. – 2018.
5. Аблязов, Р. Программирование на ассемблере на платформе x86-64 / Р. Аблязов. – 2016.