

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**



Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 6.1**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема: «Быстрый доступ к данным с помощью хеш-таблиц»

|  |  |  |
| --- | --- | --- |
| Выполнил студент: Лазаренко С.А.  Группа: ИКБО-10-23 |  |  |

Москва 2024

**СОДЕРЖАНИЕ**

[ЦЕЛЬ РАБОТЫ 3](#_Toc178190517)

[ХОД РАБОТЫ 4](#_Toc178190518)

[Формулировка задачи 4](#_Toc178190519)

[Описание подхода к решению 4](#_Toc178190520)

[Коды программы 6](#_Toc178190521)

[Результаты тестирования 13](#_Toc178190522)

[ВЫВОД 14](#_Toc178190523)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 15](#_Toc178190524)

# ЦЕЛЬ РАБОТЫ

Освоить приёмы хеширования и эффективного поиска элементов множества. Разработать программу для управления динамическим множеством данных с использованием хеш-таблицы, позволяющей осуществлять быстрый прямой доступ к элементам по ключу. Основной задачей является реализация базовых операций работы с множеством данных (вставка, удаление, поиск и вывод) с использованием хеш-таблицы и выбранной структуры данных, описанной в индивидуальном варианте. Программа должна быть организована в виде класса, который содержит массив данных и хеш-таблицу, а также реализует механизм автоматического расширения и рехеширования при необходимости.

Хеш-функция и метод разрешения коллизий подбираются самостоятельно, с соблюдением общепринятых правил их выбора. В процессе работы предусмотрена автоматическая инициализация хеш-таблицы с пятью-семью записями, а также возможность работы с таблицей через текстовый командный интерфейс, позволяющий пользователю выполнять основные операции в произвольной последовательности. Программа должна быть протестирована на корректность выполнения всех базовых операций, включая обработку коллизий и автоматическое рехеширование, чтобы обеспечить эффективное и надежное функционирование хеш-таблицы.

# ХОД РАБОТЫ

### Формулировка задачи

Разработать программу, которое использует хеш-таблицу для организации прямого доступа к элементам динамического множества полезных данных. Множество реализуйте на массиве, структура элементов (перечень полей) которого приведена в индивидуальном варианте.

Программа должна содержать класс с базовыми операциями: вставки, удаления, поиска по ключу, вывода. Включить в класс массив полезных данных и хеш-таблицу. Хеш-функцию подобрать самостоятельно, используя правила выбора функции.

Реализовать расширение размера таблицы и рехеширование, когда это требуется, в соответствии с типом разрешения коллизий.

Предусмотреть автоматическое заполнение таблицы 5-7 записями.

Реализовать текстовый командный интерфейс пользователя для возможности вызова методов в любой произвольной последовательности, сопроводите вывод достаточными для понимания происходящего сторонним пользователем подсказками.

Провести полное тестирование программы. Результаты тестирования включите в отчет по выполненной работе.

Индивидуальный вариант работы - 2

### Описание подхода к решению

Подход к решению задачи с использованием хеш-таблицы базируется на создании эффективной структуры данных для быстрого доступа к банковским счетам. В основе решения лежит применение метода цепного хеширования, позволяющего обрабатывать ситуации, когда несколько записей имеют одинаковый хеш. Для хранения данных о банковских счетах используется структура Data\_Record, которая содержит номер счета, ФИО владельца и адрес, а также указатель на следующий элемент, что позволяет организовать связные списки при возникновении коллизий.

Основная функциональность реализована в классе Hash\_Table, который управляет массивом указателей на объекты Data\_Record. Для вычисления индекса в массиве используется хеш-функция, рассчитывающая остаток от деления номера счета на размер таблицы. Эта функция обеспечивает равномерное распределение записей, минимизируя коллизии. Вставка новых записей происходит в начало связного списка по соответствующему индексу, и при достижении определенного уровня заполнения таблица автоматически расширяется, что сохраняет эффективность операций.

При удалении и поиске программа сначала определяет индекс с помощью хеш-функции, а затем перебирает элементы в соответствующем связном списке. В случае превышения порога заполнения запускается процедура рехеширования, которая удваивает размер таблицы и перераспределяет все записи. Пользователь взаимодействует с программой через текстовый интерфейс, что позволяет добавлять, удалять, искать и выводить записи.

### Коды программы

Реализуем код программы на языке программирования C++ :

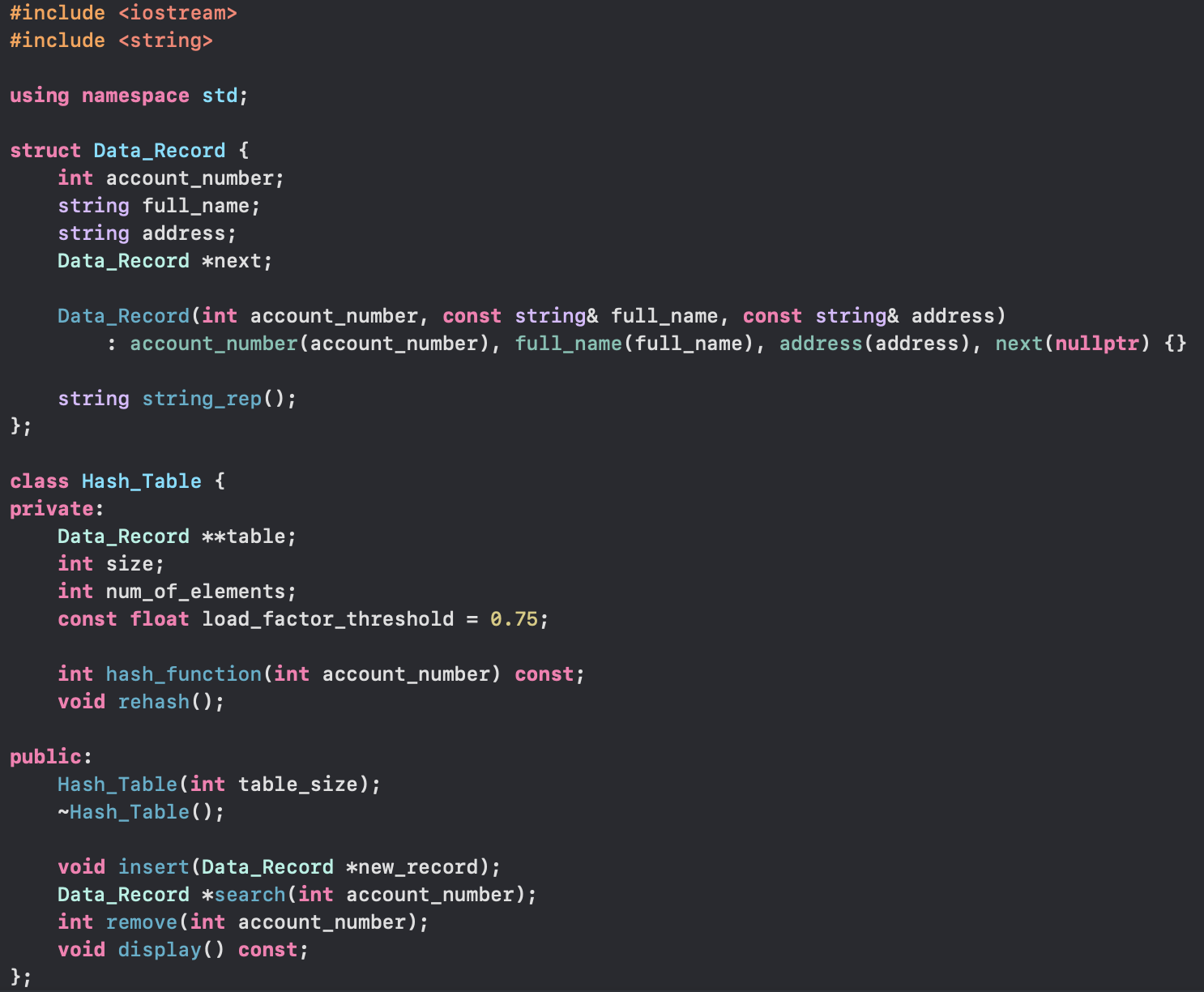


Рисунок 1 – код программы в hash.hpp



Рисунок 2 – Файл hash.cpp (часть 1)

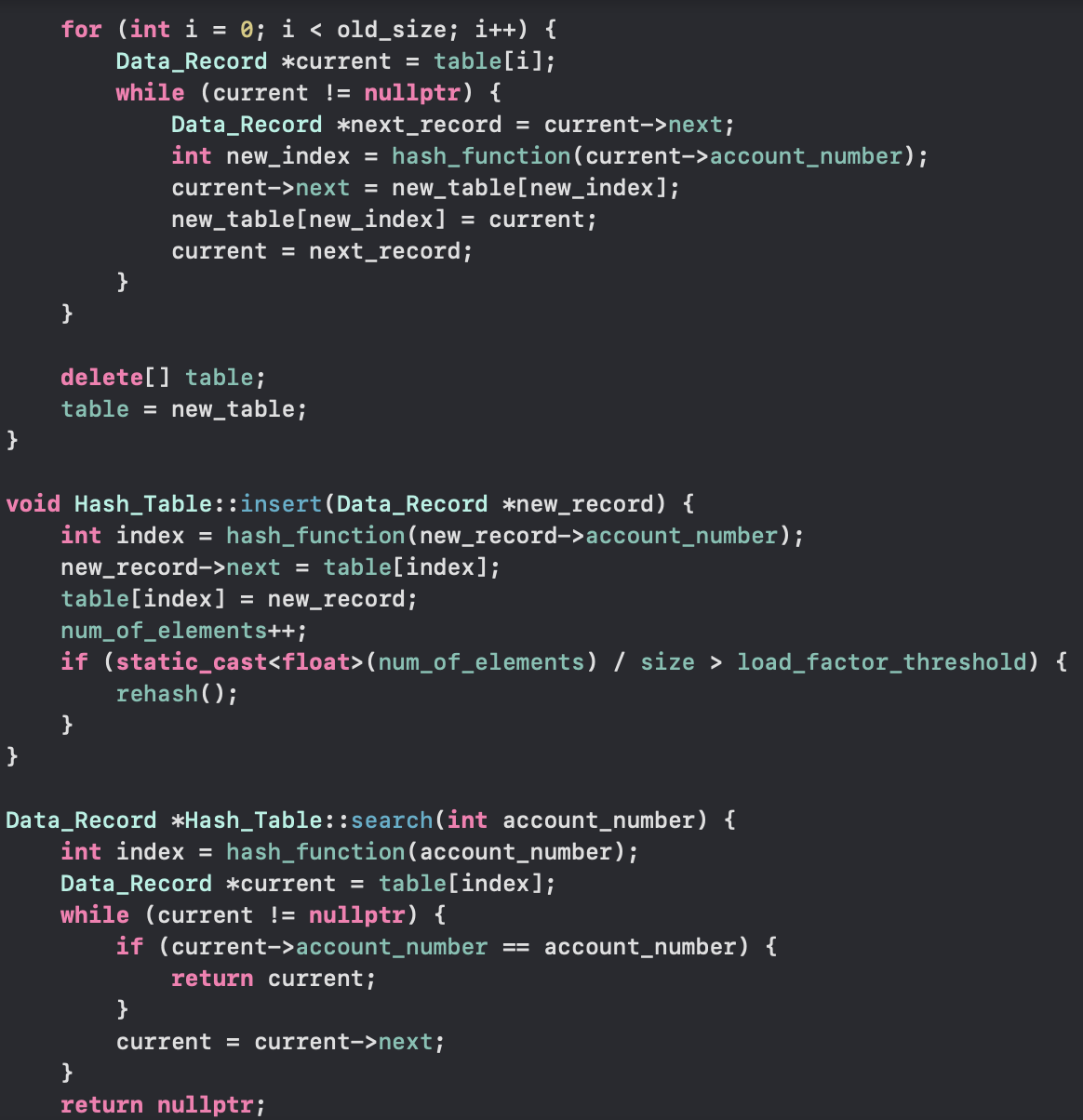


Рисунок 3 – Файл hash.cpp (часть 2)

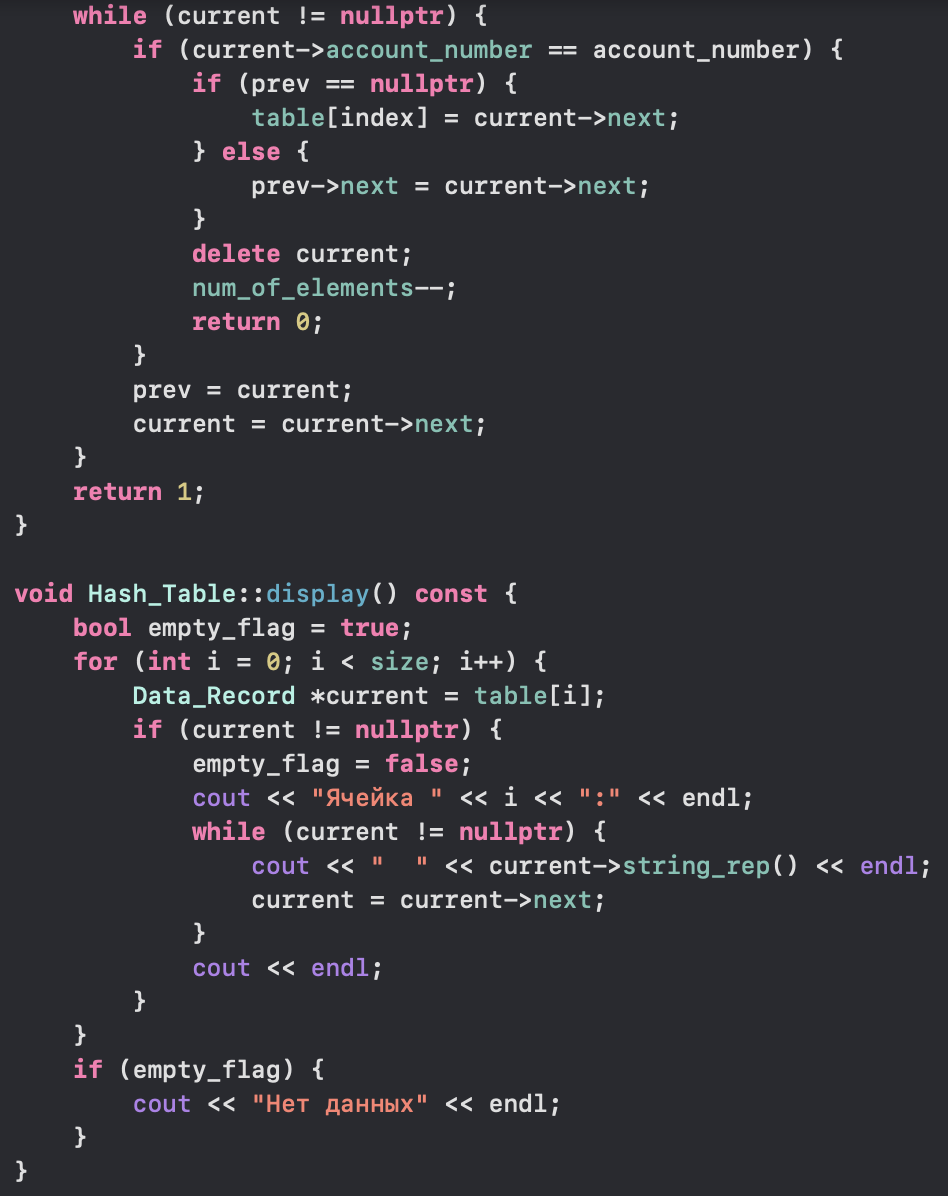


Рисунок 4 – Файл hash.cpp (часть 3)



Рисунок 5 – Файл main.cpp (часть 1)

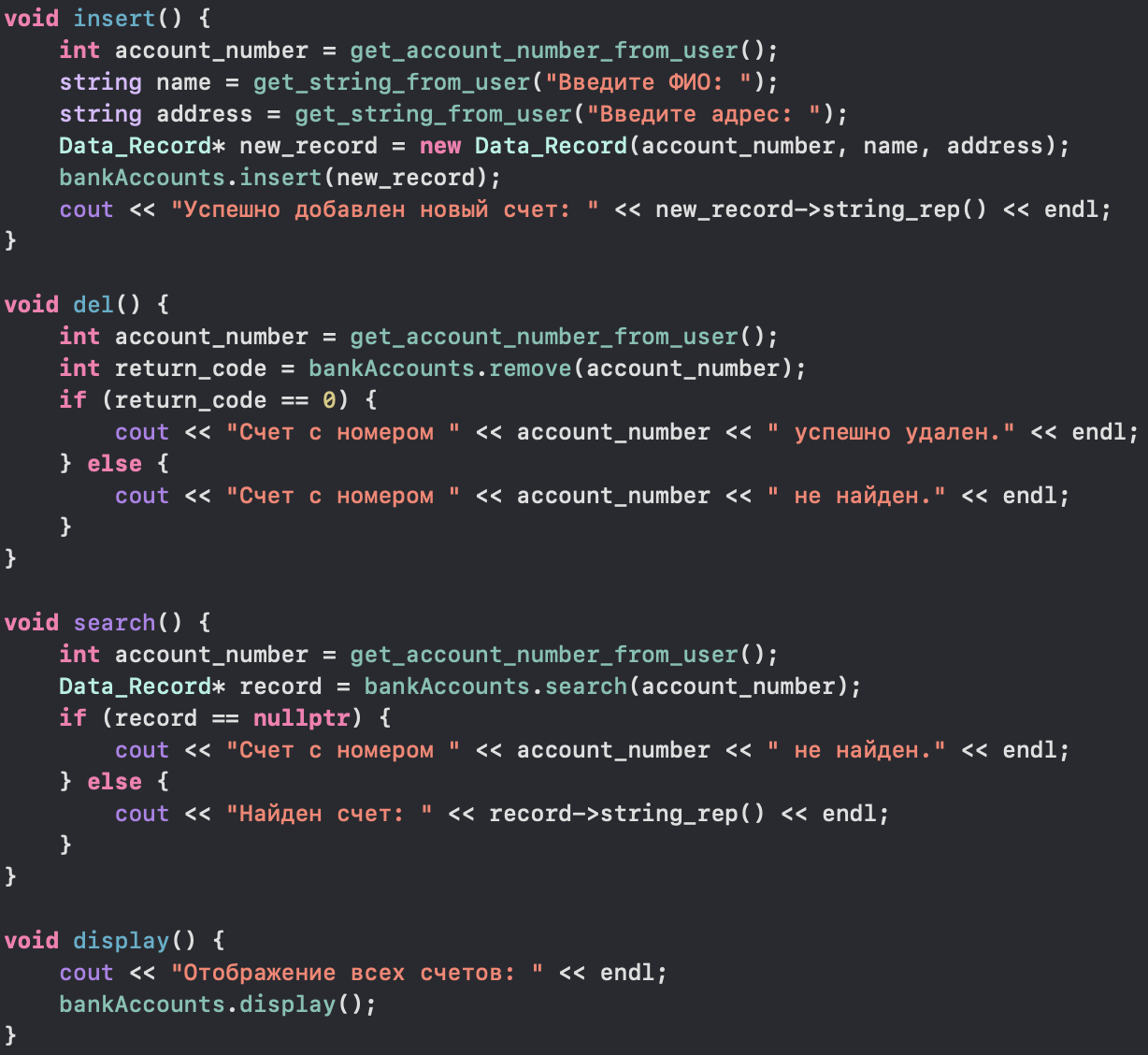


Рисунок 6 – Файл main.cpp (часть 2)



Рисунок 7 – Файл main.cpp (часть 3)

### Результаты тестирования

Выполним тестирование программы:

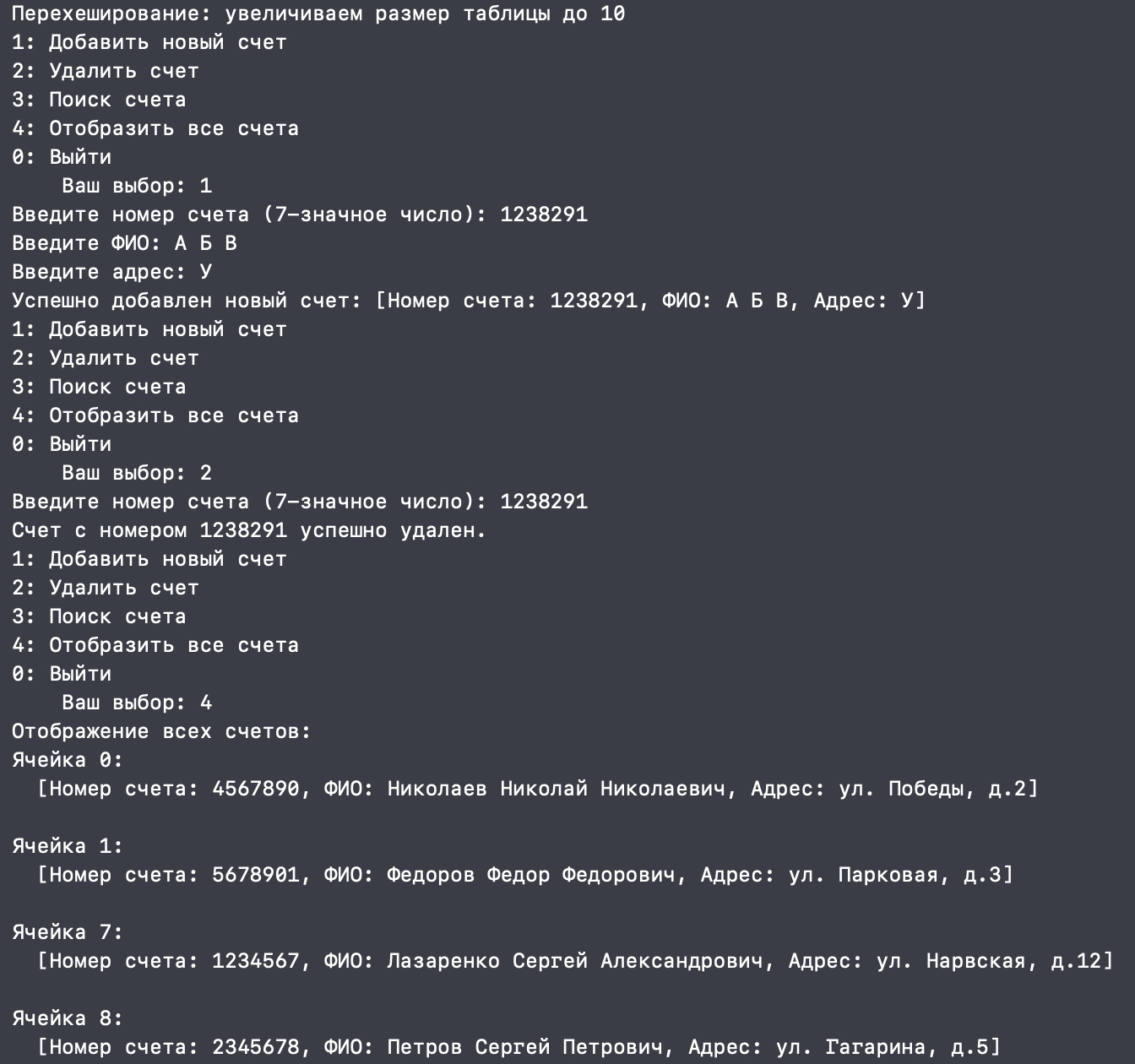


Рисунок 8 – Тестирование

Тестирование показало, что программа работает корректно.

# ВЫВОД

В результате выполнения работы были освоены приёмы хеширования и эффективного поиска элементов множества. Был получен опыт реализации хэш-таблицы с цепным хешированием.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Страуструп Б. Программирование. Принципы и практика с использованием C++. 2-е изд., 2016.

2. Документация по языку С++ [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/cpp/cpp/ (дата обращения 08.09.2024).

3. Курс: Структуры и алгоритмы обработки данных. Часть 2 [Электронный ресурс]. https://online-edu.mirea.ru/course/view.php?id=4020 (дата обращения 04.09.2024)