МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Хэш-таблица»**

**Выполнил:**студент группы 381706-2

Гущин Александр Владимирович

\_\_\_\_\_\_\_\_\_\_\_  Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_  Подпись

Нижний Новгород

2018

Оглавление

[1.Введение 3](#_Toc9373134)

[2.Постановка задачи 4](#_Toc9373135)

[3.Руководство пользователя 5](#_Toc9373136)

[4.Руководство программиста 6](#_Toc9373137)

[4.1.Описание структуры программы 6](#_Toc9373138)

[4.2.Описание структур данных 6](#_Toc9373139)

[4.3.Описание алгоритмов 7](#_Toc9373140)

[5.Эксперименты 9](#_Toc9373141)

[6.Заключение 10](#_Toc9373142)

[7.Литература 11](#_Toc9373143)

1.Введение

Хеш-таблица — абстрактный тип данных, представляющий собой массив элементов.

Поиск и вставка элементов производиться при помощи получения хеша.

Хеш зависит от ключа элемента.

2.Постановка задачи

Необходимо написать реализацию хеш-таблицы со всеми сопутствующими конструкторами, методами (положить элемент, найти элемент, удалить элемент). Вычисление хеша производится постоянным делением на 2 с добавлением текущего значения символа.

3.Руководство пользователя

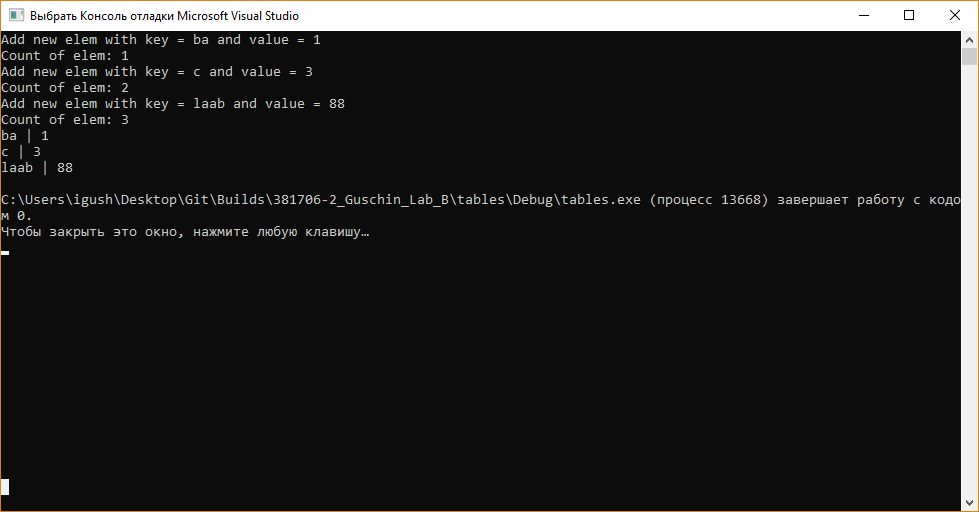


Рис.1 Пример использования.

Пример добавления элемента в таблицу, перегрузки сдвига влево.

4.Руководство программиста

4.1.Описание структуры программы

Программа состоит из таких модулей как:

* mString.h – класс строк.
* THElem.h – элемент таблицы.
* THTable.h – хеш-таблица.

4.2.Описание структур данных

1. char\* str – указатель на начало строки.
2. int count – длинна строки.
3. mString(char \* \_cstr) – конструктор с параметром.
4. mString(const mString& \_string) – конструктор копирования.
5. mString() – конструктор по умолчанию.
6. ~mString() – деструктор.
7. mString& operator=(const mString& \_string) – перегрузка оператора = для строк.
8. mString& operator=(char\* \_cstr) – перегрузка оператора = для массива символов.
9. bool operator==(const mString& \_string) – перегрузка оператора ==.
10. bool operator!=(const mString& \_string) – перегрузка оператора !=.
11. bool operator<=(const mString& \_string) – перегрузка оператора <=.
12. bool operator>=(const mString& \_string) – перегрузка оператора >=.
13. char\* GetCStr() const – метод, возвращает указатель на массив символов.
14. int GetCount() const – метод, возвращает длину сторки.
15. mString key – ключ элемента.
16. T value – значение элемента.
17. THElem(const mString& \_key, const T& \_value) – конструктор с параметрами.
18. THElem() - конструктор по умолчанию.
19. THElem(const THElem& elem) – конструктор копирования.
20. ~THElem() – деструктор.
21. T& GetValue() – метод, возвращает ссылку на значение элемента.
22. mString& GetKey() – метод, возвращает ссылку на ключ.
23. THElem<T>& operator=(const THElem& elem) – перегрузка оператора =.
24. bool operator==(const THElem& elem) – перегрузка оператора ==.
25. bool operator!=(const THElem& elem) – перегрузка оператора !=.
26. void SetKey(const mString& \_key) – метод, ставит новый ключ.
27. void SetValue(const T& \_value) – метод, ставит новое значение.
28. friend std::ostream& operator<<(std::ostream& out, THElem<T>& elem) – перегрузка сдвига влево.
29. int count – текущее кол-во элементов в таблице.
30. int size – размер таблицы.
31. int m – шаг при обходе массива.
32. static THElem<T> exemp – экземпляр класса THElem<T> по умолчанию.
33. THElem<T>\* pElem – указатель на начало массива.
34. int GetHash(const mString& elem) – метод, возвращает хеш от ключа.
35. THTable(const int \_size = 10) – конструктор с параметром.
36. THTable(const THTable<T>& table) – конструктор копирования.
37. void AddElem(THElem<T>& elem) – метод, добавляет элемент в таблицу.
38. void AddElem(const mString key ,const T & value) – метод, добавляет элемент в таблицу.
39. T& Find(const mString& key) – метод, возвращает значение элемента по ключу.
40. void DeleteElem(const mString& key) – удаление элемента.
41. int GetCount() – метод, возвращает кол-во элементов в таблице.
42. friend std::ostream& operator<<(std::ostream& out, const THTable<T>& table) – перегрузка сдвига влево.

4.3.Описание алгоритмов

void AddElem(THElem<T>& elem)

Если места нет, то бросаем исключение.

Находим хеш.

Пока не найдено свободное место, к хешу прибавляем константу и берем остаток от деления на размер массива.

Приравниваем свободному элементу заданный элемент.

Увеличиваем счетчик count на 1.

T & THTable<T>::Find(const mString & key)

Находим хеш.

Заводим счетчик итераций.

Пока ключи не совпадут и счетчик итераций меньше размера массива, к хешу прибавляем константу и берем остаток от деления на размер массива, также увеличиваем счетчик итераций на 1.

Если первый найденный элемент оказался пустой или счетчик итераций = количеству элементов в таблице, то бросаем исключение.

Возвращаем значение элемента.

5.Эксперименты

При самом удачном стечении обстоятельств вставка, нахождение и удаление элемента составляет O(1), в худшем, если все элементы имеют одинаковое значение хеша, O(n).

Характеристики компьютера: Amd fx-8350, 8gb оперативной памяти

6.Заключение

Благодаря этой работе у меня появился опыт взаимодействия с хеш-таблицами, улучшил понимание основ программирования базовых структур данных, осознал важность хеш-таблиц в жизни человека. Хеш-таблица значительно упрощает работу с информацией и помогают решать многие практические задания.

7.Литература

1. Википедия [https://ru.wikipedia.org/wiki/Хеш-таблица](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88-%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0) (Дата обращения 25.4.2019)
2. Сайт <http://algolist.manual.ru/ds/s_has.php> (Дата обращения 25.4.2019)