

Федеральное агентство по образованию Российской Федерации
Государственное образовательное учреждение
Высшего профессионального образования
Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий математики механики

ХЕШ-ТАБЛИЦЫ

Отчет по лабораторной работе

Выполнил:

студент ИИТММ гр. 381903-3

Алилуев А.О. _____

Проверил:

ассистент каф. МОСТ, ИИТММ

Лебедев И.Г. _____.

Содержание

1.Введение	3
2.Постановка целей и задач	4
3.Руководство пользователя	5
4.Руководство программиста	7
4.1.Описание структуры программы	7
4.2.Описание структур данных	8
4.3.Описание алгоритмов	9
5.Заключение	11
6.Литература	12

1. Введение

Еще одним видом таблиц, алгоритмы которых позволяют быстро вставлять и удалять элементы, являются хеш-таблицы. Они основаны на использовании хеш-функции, которая преобразует ключ элемента таблицы в целочисленное число, которое является индексом элемента в массиве. У хеш-функций есть один существенный недостаток, а именно: при передаче двух, казалось бы, разных ключей в хеш-функцию, она может вернуть одинаковые целочисленные значения. Таким образом встает проблема взаимнооднозначности или коллизии, для решения которой приходится придумывать отдельные алгоритмы. Один из таких алгоритмов будет рассмотрен в текущей работе.

2. Постановка целей и задач

Целью лабораторной работы является создание структуры хранения типа «Хеш-таблица» и методов работы с ним, таких как:

- Добавление элементов в таблицу;
- Удаление элементов из таблицы;
- Получение доступа к элементу.

Для реализации алгоритмов будет использоваться 3 класса:

- String;
- TElem;
- THashTable.

Классы TElem и THashTable являются шаблонными, и классы String и TElem уже были написаны в одной из предыдущих работ.

Для проверки правильности работы этих классов будут написаны тесты с использованием фреймворка Google Test, а также тестовый образец программы, которая будет использует класс список.

3.Руководство пользователя

После запуска программы пользователя встречает консольное окно (рис. 1):

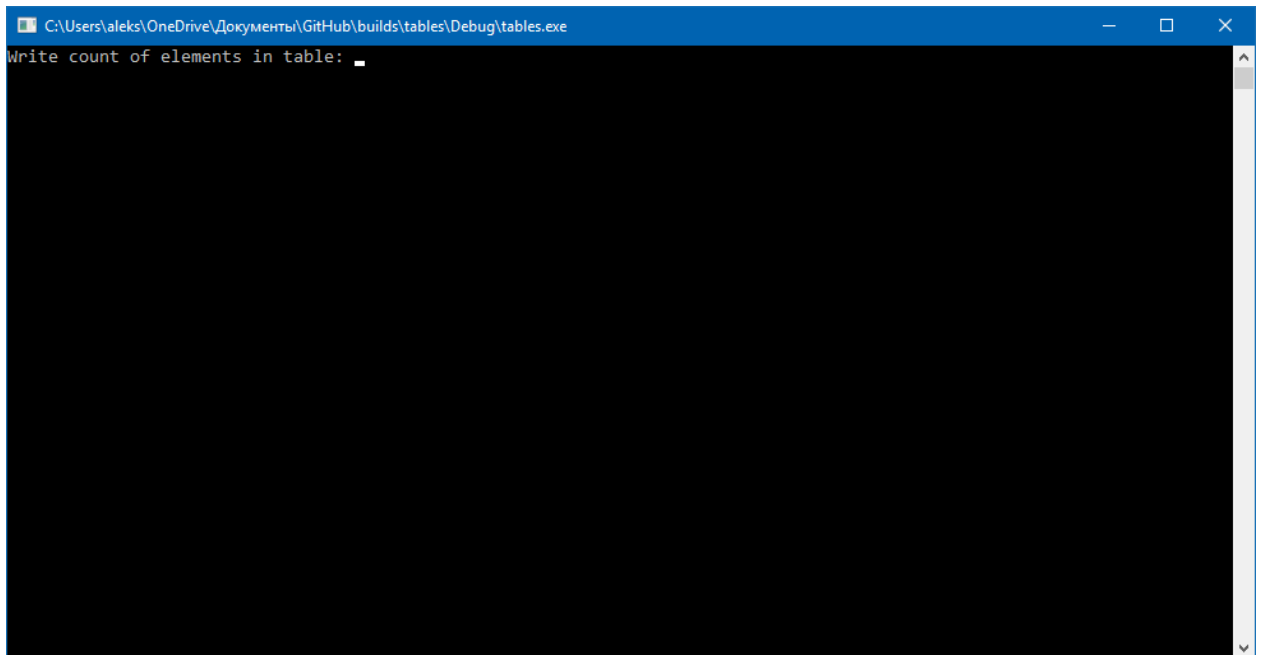


рис. 1 (вывод программы тестирования хеш-таблиц для пользователя)

в которой сначала от пользователя требуется ввести количество элементов в таблице, а затем заполнить поля этой таблицы (1 поле – ключ, 2 поле – целочисленное число, которое будет хранить эта ячейка) (рис. 2).

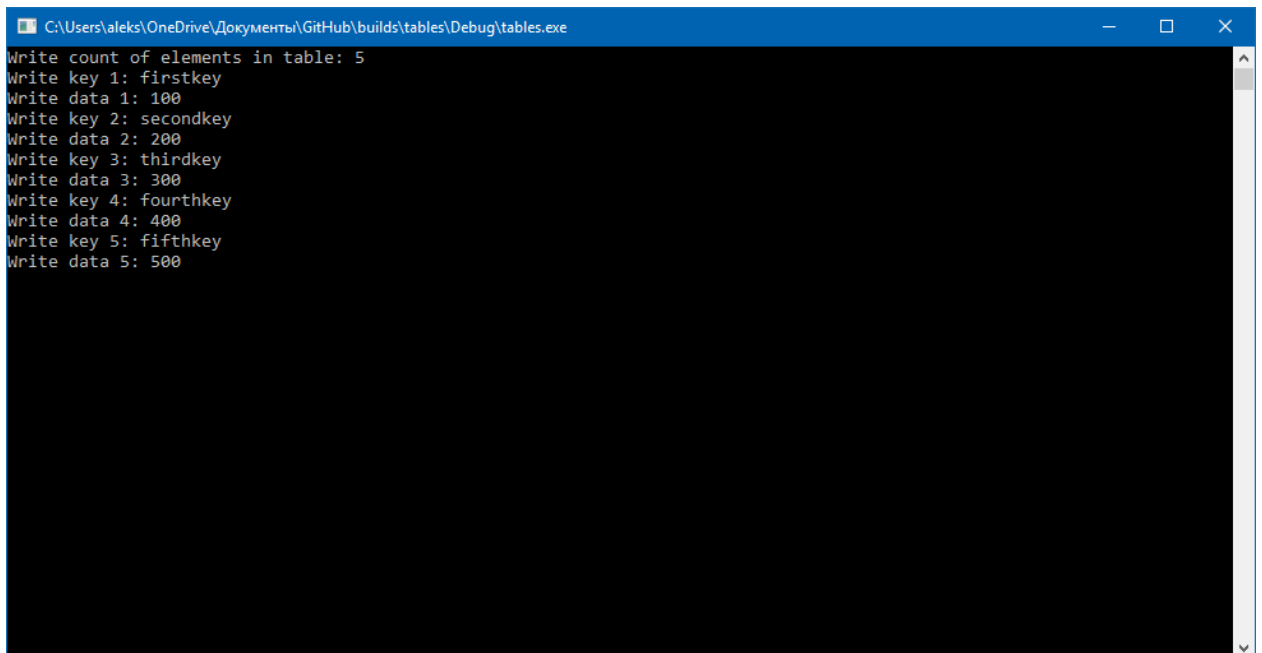
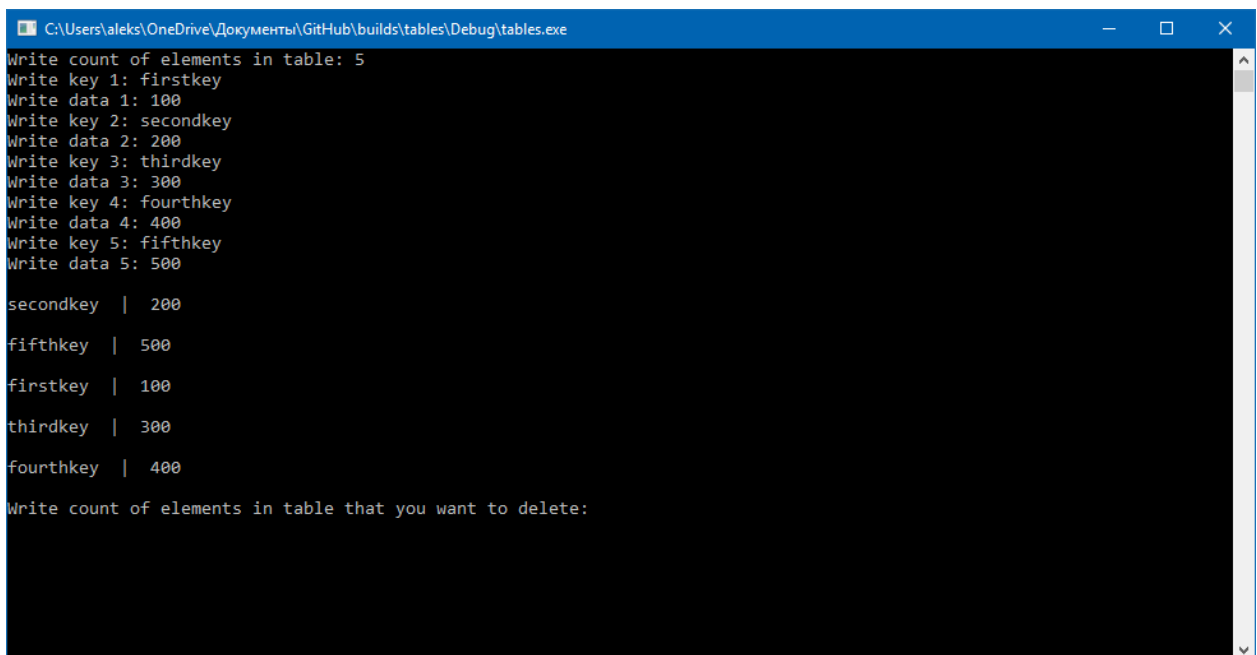


рис. 2 (заполнение полей таблицы)

Далее полученная таблица будет выведена для пользователя (порядок вывода элементов такой же, как порядок их хранения в памяти) и программа предложит удалить некоторое количество элементов, нужно ввести их количество (рис. 3).



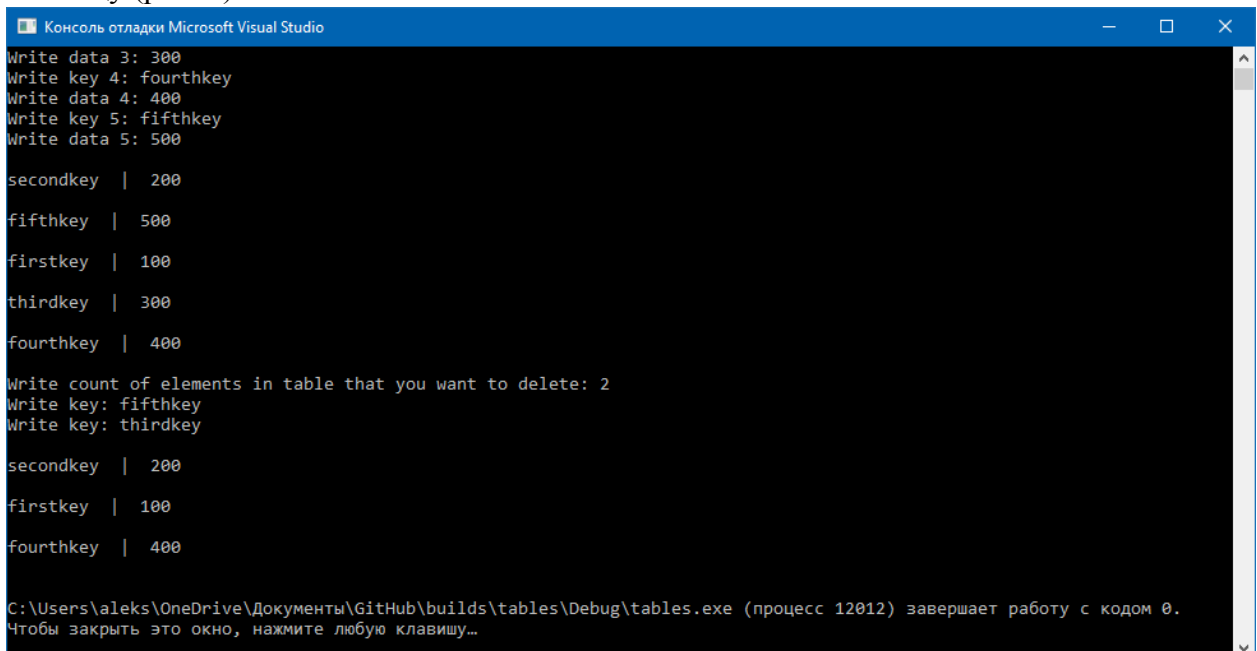
```
C:\Users\aleks\OneDrive\Документы\GitHub\builds\tables\Debug\tables.exe
Write count of elements in table: 5
Write key 1: firstkey
Write data 1: 100
Write key 2: secondkey
Write data 2: 200
Write key 3: thirdkey
Write data 3: 300
Write key 4: fourthkey
Write data 4: 400
Write key 5: fifthkey
Write data 5: 500

secondkey | 200
fifthkey | 500
firstkey | 100
thirdkey | 300
fourthkey | 400

Write count of elements in table that you want to delete:
```

рис. 3 (вывод таблицы на экран и запрос на ввод количества удаляемых элементов)

В конце программа попроси ввести ключи удаляемых элементов и выведет конечную таблицу (рис. 4).



```
Консоль отладки Microsoft Visual Studio
Write data 3: 300
Write key 4: fourthkey
Write data 4: 400
Write key 5: fifthkey
Write data 5: 500

secondkey | 200
fifthkey | 500
firstkey | 100
thirdkey | 300
fourthkey | 400

Write count of elements in table that you want to delete: 2
Write key: fifthkey
Write key: thirdkey

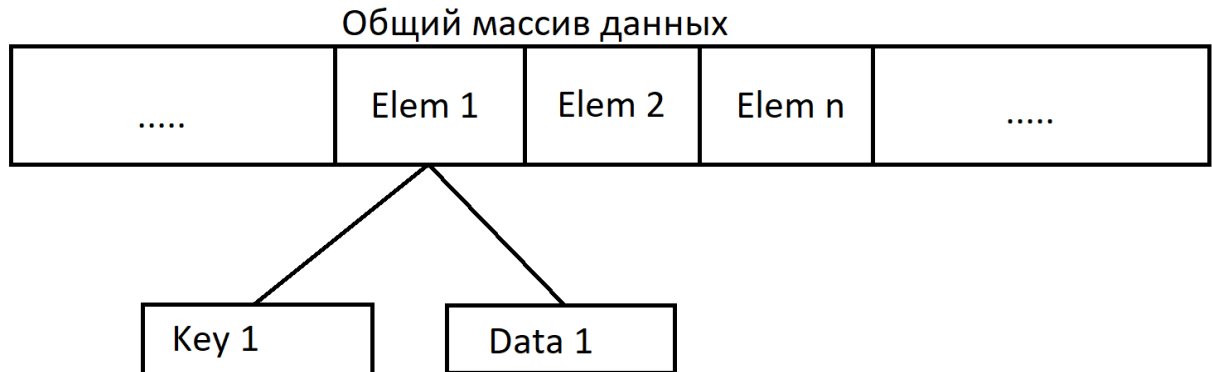
secondkey | 200
firstkey | 100
fourthkey | 400

C:\Users\aleks\OneDrive\Документы\GitHub\builds\tables\Debug\tables.exe (процесс 12012) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу...
```

рис. 4 (ввод ключей и удаление элементов таблицы)

4.Руководство программиста

4.1.Описание структуры программы



Хеш-таблица будет реализована как массив элементов таблицы, каждый элемент включает в себя ключ и значение, которое хранит этот элемент:

То есть для реализации алгоритмов будет использовано 3 класса:

- Класс «Строка» (String).
- Класс «Элемент Таблицы» (TElem), который будет использовать класс String.
- Класс «Таблица» (TTable), который использует класс TElem.

Проблема коллизии будет решена с помощью простых чисел, которые будут использованы в качестве размера массива и постоянной переменной, которая будет использована, как шаг для обхода по массиву. Если при использовании, хеш-функция вернула неправильное значение, то к возвращённому значению будет прибавляться постоянная переменная и проверяться ключ текущего элемента, пока не найдется нужный элемент.

А также проект использующий фреймворк Google Test, для проверки правильности работы этих классов и тесовый проект, который будет показываться пользователю.

Класс String:

Класс строка реализует функции работы с массивом символов, такие как: сравнение, присвоение, доступ к элементам массива.

Класс TElem:

Класс элемент таблицы содержит реализацию работы с элементами. В нем реализованы такие методы, как: сравнение элементов, доступ к ключам и данным.

Класс THashTable:

Класс просмотревые таблицы содержит реализацию работы с таблицами. В нем реализованы такие методы, как: положить элемент в таблицу, удалить элемент из таблицы, получить значение по ключу и др.

Класс gtest:

Класс gtest реализует тестирование классов String, TElem и THashTable, по средствам фреймворка Google Test. Тесты пишутся для каждого метода классов, каждого ветвления этих методов и для всех возможных исключений этих методов.

Проект table:

В данном проекте реализован примет использования таблиц, показанный пользователю.

4.2.Описание структур данных

Реализация классов String и TElem не изменилась с предыдущей работы, поэтому их описание пропустим.

Класс TTable:

`template< class T >` - шаблон класса T

`static TElem<T> st` – статический пустой элемент таблицы, которым первоначально заполняем таблицу;

`TElem<T>* node` – указатель на массив элементов таблицы;

`int size` – текущий размер таблицы;

`int count` – текущее количество занятых элементов таблицы.

`const int m = 2` – шаг для обхода массива.

Описание методов:

Метод:	Описание:
<code>TElem<T> THashTable<T>::st;</code>	Инициализация статического поля класса
<code>THashTable<T>::THashTable(const int _size)</code>	Конструктор таблицы, который принимает максимальное количество элементов этой таблицы.

<code>THashTable<T>::THashTable(const THashTable<T>& table)</code>	Конструктор копирования для хеш-таблицы.
<code>THashTable<T>::~~THashTable()</code>	Деструктор таблицы.
<code>int THashTable<T>::GetCount() const</code>	Возвращает текущее количество элементов в таблице.
<code>void THashTable<T>::Add(String& key, const T& data)</code>	Метод, который позволяет добавить элемент в таблицу, принимая ключ, который будет иметь этот элемент и само значение элемента.
<code>void THashTable<T>::Add(TElem<T>& elem)</code>	Метод, который позволяет добавить элемент в таблицу, принимая ссылку на уже существующий элемент.
<code>bool THashTable<T>::Del(String& key)</code>	Метод, который позволяет удалить элемент из таблицы по его ключу.
<code>T& THashTable<T>::Search(String& key)</code>	Метод, осуществляющий поиск элемента в таблице по его ключу и возвращает его значение.
<code>bool THashTable<T>::IsSimple(const int number)</code>	Метод, который проверяет простое ли число, пришедшее ему как параметр.
<code>int THashTable<T>::Hash(String& key)</code>	Хеш-функция, которая возвращает целочисленное число по ключу.
<code>void THashTable<T>::Expansion(int newsize)</code>	Метод, который позволяет увеличить максимальный размер таблицы.

4.3. Описание алгоритмов

Подробное описание некоторых методов

Добавление элемента по значению и ключу:

- Проверка не закончилось ли память в массиве, если да, то выделение дополнительной;
- Применение хеш-функции для вычисления места нового элемента.
 - Если число, которое выдала хеш-функция больше текущего размера таблицы, то ее размер увеличивается;
- Пока не найдется первый свободный элемент в массиве будем прибавлять к числу, которое вернула хеш-функция постоянную для шага;
- Как только найдено первое свободное место, то записываем туда данные и увеличиваем количество элементов в таблице на единицу.

Удаление элемента таблицы по ключу:

- Применяем хеш-функцию для нахождения нужного элемента;

- Если ключ элемента не совпал с параметром, то в цикле проходи по массиву от числа, возвращенного хеш-функцией, с шагом постоянной, прибавляя ее к числу;
- Если элемент не нашли, то возвращаем false;
- Если удалось найти, то ставим вместо него пустой элемент, уменьшаем количество элементов в таблице на единицу и возвращаем true.
-

5. Заключение

В заключении можно сказать, что все поставленные цели и задачи были выполнены, а именно: созданы классы «String», «TElem» и «THashTable» с реализованными методами добавления, удаления и доступа к элементам таблицы, а также написаны к ним тесты, и они успешно пройдены.

6.Литература

- Учебные материалы к учебному курсу «Методы программирования» - Гергель В.П.