

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

Институт информационных технологий математики механики

# ОРГАНИЗАЦИЯ ДОСТУПА ПО ИМЕНИ (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          | 10 |
| 5.Заключение                     | 12 |
| 6.Литература                     | 13 |

## 1. Введение

Иногда для удобной организации доступа к памяти или элементам, которые лежат в этих ячейках памяти, прибегаю к организации доступа по имени элемента. Если приводить аналогию с массивом, то именем каждого элемента будет являться индекс этого элемента. Назовем такой способ доступа к данным табличным, а структуру хранения данных таблицей. Для реализации этой структуры данных нам необходим массив из элементов таблицы, которые в свою очередь состоят из хранимого элемента и его ключа. Существует множество реализаций таблиц, но в этой работе мы рассмотрим простейший их вид – просмотрные таблицы. Их особенность заключается в том, что у них нет особенностей и порядок хранения элементов определяется моментом их занесения в таблицу. Отсюда возникают закономерности, такие как поиск или удаление элемента из таблицы будет выполняться за линейное время. Тем не менее, такие таблицы достаточно легко преобразовать к отсортированным, о которых речь пойдет в следующий раз.

## 2. Постановка целей и задач

Целью лабораторной работы является создание структуры хранения типа «Просмотровая 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;
- TSeeTable.

Классы TElem и TSeeTable являются шаблонными.

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):

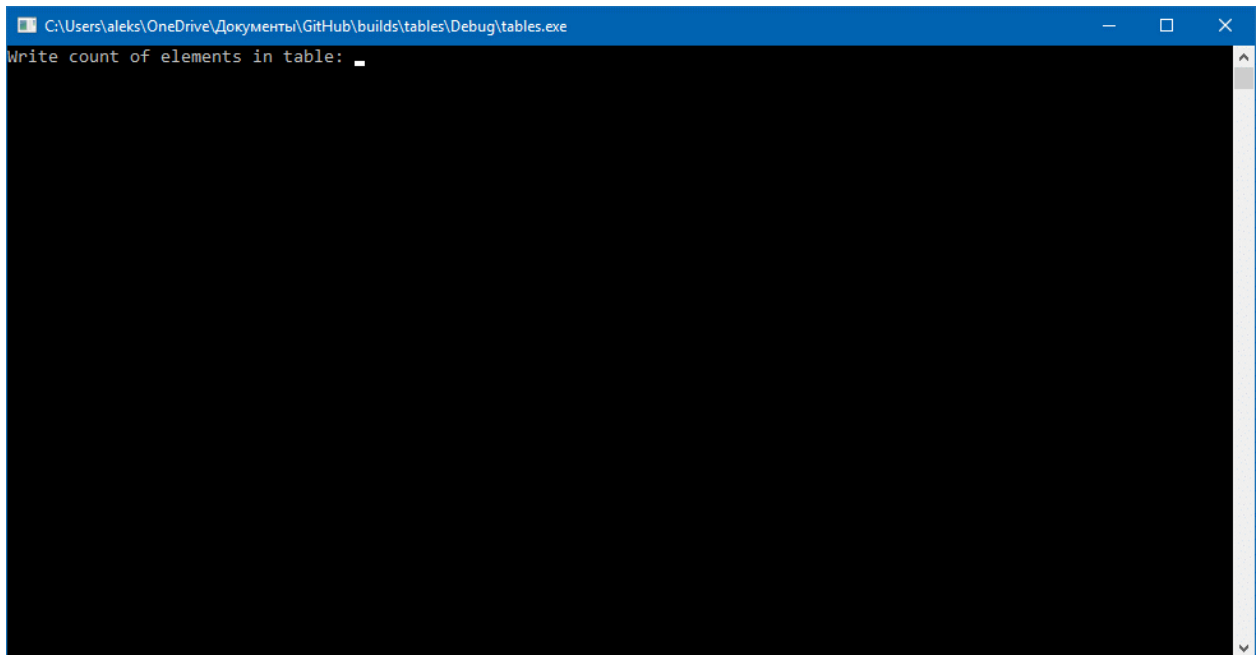


рис. 1 (вывод программы тестирования просмотревых таблиц для пользователя)

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).

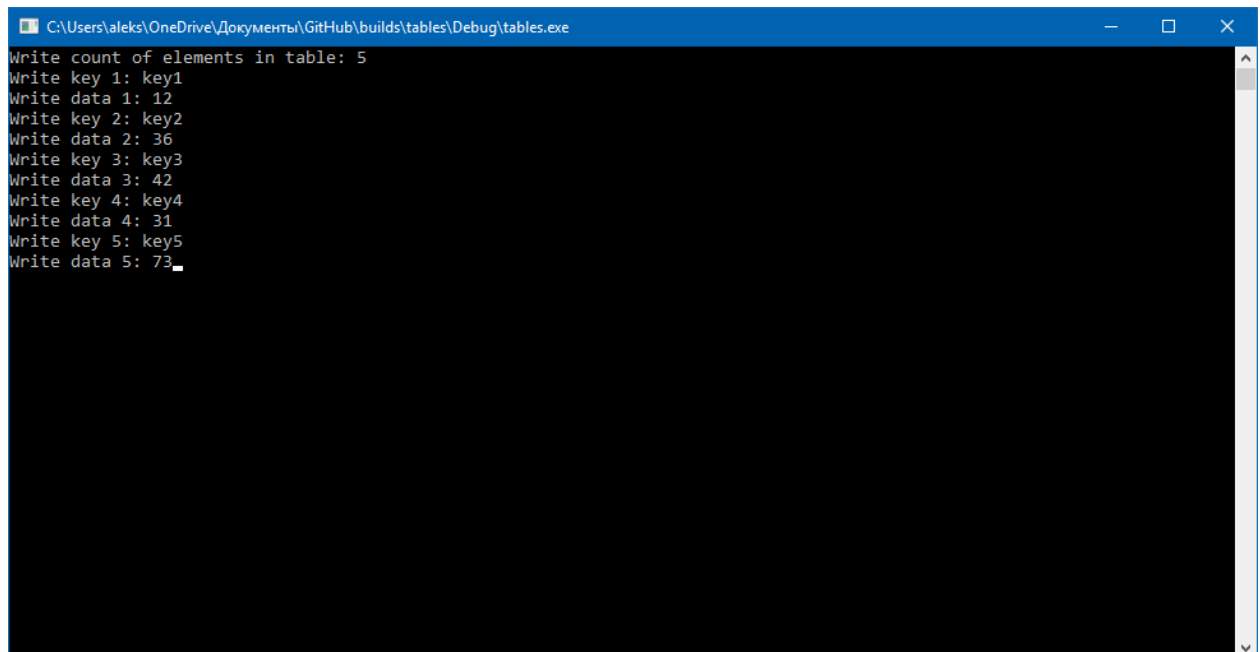


рис. 2 (заполнение полей таблицы)

Далее полученная таблица будет выведена для пользователя 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: key1
Write data 1: 12
Write key 2: key2
Write data 2: 36
Write key 3: key3
Write data 3: 42
Write key 4: key4
Write data 4: 31
Write key 5: key5
Write data 5: 73

key1 | 12
key2 | 36
key3 | 42
key4 | 31
key5 | 73

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: 42
Write key 4: key4
Write data 4: 31
Write key 5: key5
Write data 5: 73

key1 | 12
key2 | 36
key3 | 42
key4 | 31
key5 | 73

Write count of elements in table that you want to delete: 2
Write key: key2
Write key: key4

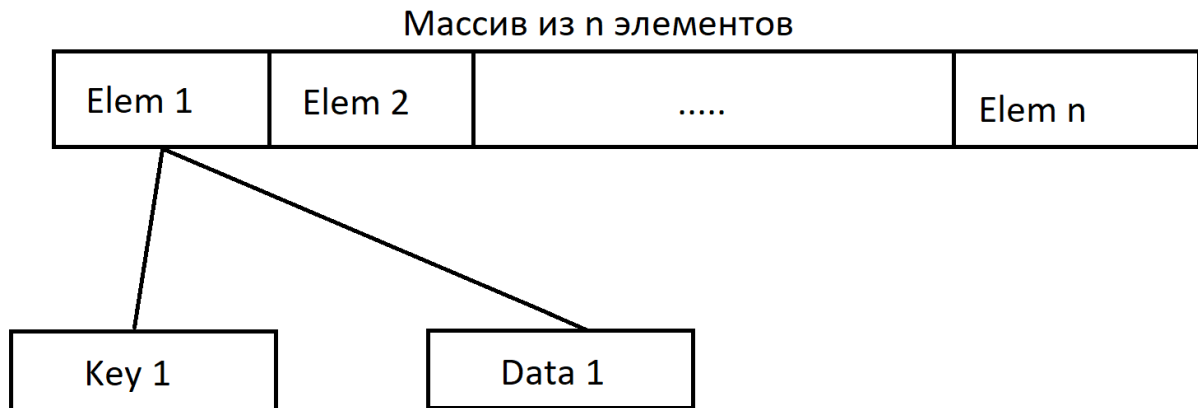
key1 | 12
key3 | 42
key5 | 73

C:\Users\aleks\OneDrive\Документы\GitHub\builds\tables\Debug\tables.exe (процесс 16352) 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



Просмотровая 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.

А также проект использующий фреймворк Google Test, для проверки правильности работы этих классов и тестовый 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 TTable:**

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 TTable, 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

### Класс String:

`char*` str – указатель на массив для хранения строки;

`int` count – переменная для хранения числа элементов в строке.

### Класс TElem:

`template< class T >` - шаблон класса T

`T data` – хранимое значение элемента;

`String key` – ключ элемента.

### 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.

### Описание методов:

| Метод:  | Описание:   |
|---|---|
| <code>String::String()</code>                 | Конструктор по умолчанию для класса String.   |
| <code>String::String(const char* _str)</code> | Конструктор с параметром для класса String, который принимает указатель на массив символов. |



|  |  |
|--|--|
| <code>String::String(const String&amp; _str)</code>  | Конструктор копирования класса String.   |
| <code>String::~~String()</code>  | Деструктор класса String.  |
| <code>int String::GetCount() const</code>  | Метод, который возвращает количество символов в строке.                                |
| <code>String&amp; String::operator=(const String&amp; _str)</code>                         | Перегрузка оператора присваивания.   |
| <code>bool String::operator==(const String&amp; _str) const</code>                         | Перегрузка оператора равно для строки.   |
| <code>bool String::operator!=(const String&amp; _str) const</code>                         | Перегрузка оператора не равно для строки.  |
| <code>bool String::operator&lt;(const String&amp; _str) const</code>                       | Перегрузка оператора меньше для строки.  |
| <code>bool String::operator&gt;(const String&amp; _str) const</code>                       | Перегрузка оператора больше для строки.  |
| <code>char&amp; String::operator[](const int pos) const</code>                             | Возвращение элемента массива символов по его номеру.                                   |
| <code>TElem&lt;T&gt;::TElem()</code>   | Конструктор по умолчанию для элемента таблицы.   |
| <code>TElem&lt;T&gt;::TElem(const T&amp; _data, const String&amp; _key)</code>             | Конструктор с параметрами, который принимает хранимый элемент и его ключ.              |
| <code>TElem&lt;T&gt;::~~TElem()</code>   | Деструктор для элемента таблицы.   |
| <code>TElem&lt;T&gt;&amp; TElem&lt;T&gt;::operator=(const TElem&lt;T&gt;&amp; elem)</code> | Перегрузка оператора присваивания для элемента.  |
| <code>bool TElem&lt;T&gt;::operator==(const TElem&lt;T&gt;&amp; elem)</code>               | Перегрузка оператора равно для элемента таблицы.                                       |
| <code>bool TElem&lt;T&gt;::operator!=(const TElem&lt;T&gt;&amp; elem)</code>               | Перегрузка оператора не равно для элемента таблицы.                                    |
| <code>T&amp; TElem&lt;T&gt;::GetData()</code>  | Возвращает значение, которое хранится в элементе таблицы.                              |
| <code>void TElem&lt;T&gt;::SetData(const T&amp; _date)</code>                              | Метод, который позволяет изменить значение, которое хранится в элементе таблицы.       |
| <code>String&amp; TElem&lt;T&gt;::GetKey()</code>  | Возвращает ключ элемента таблицы.  |
| <code>void TElem&lt;T&gt;::SetKey(const String&amp; _key)</code>                           | Позволяет изменить ключ элемента таблицы.  |
| <code>TElem&lt;T&gt; TTable&lt;T&gt;::st;</code>   | Инициализация статического поля класса   |
| <code>TTable&lt;T&gt;::TTable(const int _size)</code>                                      | Конструктор таблицы, который принимает максимальное количество элементов этой таблицы. |
| <code>TTable&lt;T&gt;::TTable(const TTable&lt;T&gt;&amp; table)</code>                     | Конструктор копирования для просмотрной таблицы.                                       |
| <code>TTable&lt;T&gt;::~~TTable()</code>   | Деструктор таблицы.  |

|  |   |
|--|---|
| <code>int TTable&lt;T&gt;::GetCount()<br/>const</code>   | Возвращает текущее количество элементов в таблице.  |
| <code>bool TTable&lt;T&gt;::Add(const<br/>TElem&lt;T&gt;&amp; elem)</code>                     | Метод, который позволяет добавить элемент в таблицу, принимая ссылку на уже существующий элемент.                                   |
| <code>String&amp; TTable&lt;T&gt;::Add(const<br/>T&amp; _data)</code>                          | Метод, который позволяет добавить элемент в таблицу, принимая значение для этого элемента, а ключ будет сгенерирован автоматически. |
| <code>bool<br/>TTable&lt;T&gt;::Del(TElem&lt;T&gt;&amp;<br/>elem)</code>                       | Метод, который позволяет удалить элемент из таблицы по такому же элементу.  |
| <code>bool TTable&lt;T&gt;::Del(const<br/>String&amp; _key)</code>                             | Метод, который позволяет удалить элемент из таблицы по ключу.   |
| <code>TElem&lt;T&gt;&amp;<br/>TTable&lt;T&gt;::Search(const<br/>String&amp; _key) const</code> | Метод, осуществляющий поиск элемента в таблице по его ключу.  |
| <code>T&amp;<br/>TTable&lt;T&gt;::operator[] (const<br/>String&amp; _key) const</code>         | Метод, который возвращает значение элемента по ключу.   |
| <code>void<br/>TTable&lt;T&gt;::Expansion(const int<br/>newsize)</code>                        | Метод, который позволяет увеличить максимальный размер таблицы.   |

### 4.3. Описание алгоритмов

#### Подробное описание некоторых методов

Добавление элемента по значению:

- Проверка не закончилось ли место в таблице, если да, то расширяем ее в 2 раза;
- Кладем значение элемента в нужный элемент массива и создаем строку со стандартным именем;
- Если это первый элемент массива, то присваиваем ключу этого элемента значение стандартного имени, увеличиваем количество элементов в таблице и возвращаем ключ.
- Если нет, строка со стандартным ключом переходит в строку из одного символа – первой буквы предыдущего элемента.

Удаление элемента таблицы по ключу:

- Проверка на пустоту таблицы, если пуста, то возвращаем false;
- Проходим в цикле по всем элементам таблицы, и если находим нужный, то сдвигаем все оставшиеся элементы таблицы на 1 вперед, уменьшаем количество элементов на 1 и возвращаем true;

- Если элемент не нашли, то возвращаем false.
-

## 5. Заключение

В заключении можно сказать, что все поставленные цели и задачи были выполнены, а именно: созданы классы «String», «TElem» и «TTable» 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.