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**

**Отчет по учебной практике**

**«Структура хранения данных: просмотровые таблицы»**

**Выполнил:** студент группы 381706-1

Силенко Дмитрий Игоревич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2019.

**Оглавление**

[1. Введение. 2](#_Toc8318887)

[2. Постановка задачи. 3](#_Toc8318888)

[3. Руководство пользователя. 4](#_Toc8318889)

[4. Руководство программиста. 5](#_Toc8318890)

[4.1. Описание структуры программы. 5](#_Toc8318891)

[4.2. Описание структур данных. 5](#_Toc8318892)

[4.3. Описание алгоритмов. 6](#_Toc8318893)

[5. Эксперименты. 7](#_Toc8318894)

[6. Заключение. 8](#_Toc8318895)

[7. Литература. 9](#_Toc8318896)

# Введение.

**Основная цель данной работы** – разработка динамической структуры данных – просмотровой таблицы.

Но для начала необходимо разобраться, что такое таблица и просмотровая таблица и как они работают.

**Таблица** — двухмерный объект, состоящий из строк и столбцов, который используется для хранения данных.

**Просмотровая таблица** –этоодномерный массив элементов с заранее известным размером. Т.к. данные таблицы не упорядочены, то поиск элементов таблицы и удаление элемента из таблицы имеют сложность , а добавление элемента имеет сложность .

# Постановка задачи.

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация классов элемент таблицы и просмотровая таблица – TElem, TTable.
2. Пример программы, демонстрирующая работу класса TTable.
3. Написание набора автоматических тестов с использованием Google C++ Testing Framework и проверка работоспособности методов классов.

# Руководство пользователя.

При запуске программы создается таблица, состоящая из 6 элементов. Номер ключа соответствует номеру элемента в таблице. Выполняется поиск и вывод на кран элементов по ключу k1 и k2. После чего элемент с ключом k2 удаляется из таблицы и производится повторный поиск его, а также элемента, которого в таблице в принципе нет. И в том и в другом случае программа оповещает о том, что подобных элементов нет.

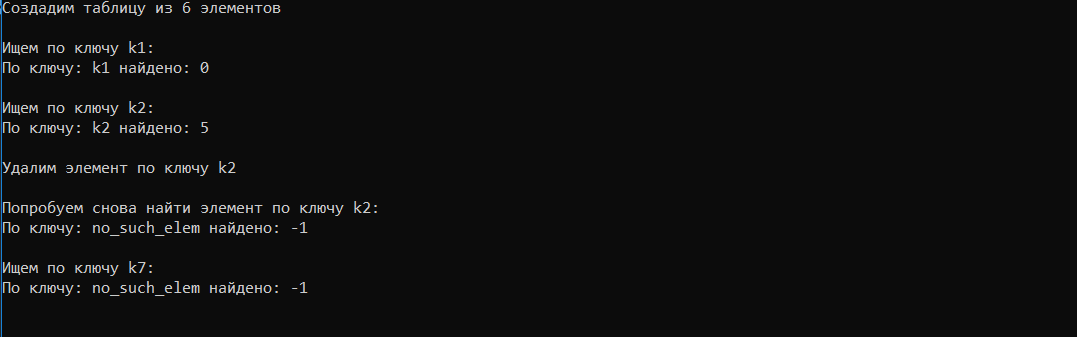


Рисунок 1 Поиск по разным ключам

# Руководство программиста.

# Описание структуры программы.

Программа состоит из следующих модулей:

* Модуль ViewingTableLib. Статическая библиотека. Включает в себя заголовочные файлы TableElem.h и ViewingTable.h, в которых описаны методы с реализаций шаблонных классов *TElem и TTable.*
* Модуль ViewingTableTest. Набор тестов для класса TTable. Включает в себя файл test\_viewing\_table*.cpp.* Разработаны они с помощью использования Google C++ Testing Framework.
* Модуль ViewingTable. Пример использования списка. Включает в себя файл с реализацией viewing\_table*\_main.cpp.*

# Описание структур данных.

*Класс TElem:*

Поля:

String key; - строка-ключ для доступа к элементам таблицы

T data; - сами данные

Конструкторы:

TElem(string \_key =””, T \_data =0);

TElem(TElem &elem);

Методы:

void SetData(T \_data); - вставить данные в таблицу

void SetKey(string \_key); - установить ключ

string GetKey(); - получить ключ

T GetData(); - получить данные

T& GetDataAddress(); - получить адрес данных

TElem<T>& operator=(TElem &elem);

bool operator==(TElem &elem); - поверка на равенство

*Класс TTable:*

Поля:

TElem<T> notFound; - элемент не найден

TElem<T> \*mas; - массив элементов таблицы

int length; - длина таблицы

int count; - количество элементов в таблице

Конструкторы:

TTable(int \_length = 10);

TTable(TTable &table);

Методы:

void SetCount(int \_count); - задать количество элементов в таблице

void SetLength(int \_length); задать длину таблицы

int GetLength(); - получить длину таблицы

int GetCount(); - получить количество элементов в таблице

void Put(string \_key, T \_data); - вставить элемент в таблицу

void Del(string \_key); - удалить элемент из таблицы

TElem<T>& Search(string \_key); - поиск элемента в таблице

T& operator[](string \_key); - доступ к элементу по ключу

# Описание алгоритмов.

**Добавление элемента в таблицу.**

Сначала проверяем таблицу на полноту. Если в ней есть свободные места, то новый элемент помещается в конец таблицы, а количество элементов таблицы увеличивается на единицу.

**Удаление элемента из таблицы.**

Сначала проверяем таблицу на пустоту. Если она не пуста, то выполняем поиск элемента по ключу. Если элемент найден, то ему присваивается значение константы **notFound**. При этом количество элементов в таблице не уменьшается.

**Поиск элемента в таблице.**

Поиск выполняется по ключу. Совершается обход таблица по количеству элементов и ключ каждого элемента сравнивается с искомым ключом. Если элемент с искомым ключом найден, то возвращается содержимое этого элемента, если же элемент не найден, то возвращается константа **notFound**.

# Эксперименты.

Эксперименты проводились на ПК с следующими параметрами:

1. Операционная система: Windows 10 Домашняя
2. Процессор: Intel(R) Core™ i5-8250U CPU @ 1.60 GHz
3. Версия Visual Studio: 2017

|  |  |  |  |
| --- | --- | --- | --- |
| Количество элементов в таблице | Время работы метода **Put(string \_key, T data)** (в млс) | Время работы метода **Del(string \_key)** (в млс) | Время работы метода **Search(string \_key)** (в млс) |
| 1000 | 0 | 1 | 2 |
| 10000 | 0 | 249 | 261 |
| 20000 | 0 | 5219 | 5431 |

*Таблица 1. Результаты экспериментов.*

Таким образом, можно увидеть, что сложность работы метода **Put(string \_key, T data)** – . А сложность работы методов **Search(string \_key)** и **Del(string \_key)** – .

# Заключение.

Эта лабораторная работа дала возможность более детально разобраться с устройством работы просмотровой таблицы. В ходе выполнения, был реализован вспомогательный шаблонный класс элемента таблицы TElem и основной шаблонный класс таблицы TTable, описанный в специально разработанной библиотеке ViewingTableLib. Все это позволило нам реализовать базовую работу с просмотровой таблицей: поиск элемента по ключу, добавление и удаление элемента.

Помимо этого, для проверки работоспособности всех методов, были написаны автоматические тесты, реализованные с использованием Google C++ Testing Framework. Пример использования списка для пользователя также написан и успешно работает.

# Литература.

1. Ссылка из Википедии про таблицы:

<https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0_(%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)>

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», 2015.