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**

**«Реализация просмотровых таблиц»**

**Выполнил:** студент группы 381706-1

Власов Андрей Сергеевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2019.

Содержание

[1. Введение 3](#_Toc533027637)

[2. Постановка задачи 4](#_Toc533027638)

[3. Руководство пользователя 5](#_Toc533027639)

[4. Руководство программиста 6](#_Toc533027640)

4.[1. Описание структуры программы 6](#_Toc533027641)

4.[2. Описание структур данных 6](#_Toc533027642)

4.[3. Описание алгоритмов 7](#_Toc533027643)

[5. Заключение](#_Toc533027644) 9

[6. Литература](#_Toc533027645) 10

# **Введение**

Таблица –  это совокупность связанных данных, хранящихся в структурированном виде. Таблица использует модель вертикальных столбцов (имеющих уникальное имя) и горизонтальных строк. Ячейка — место, где строка и столбец пересекаются. Таблица содержит определенное число столбцов, но может иметь любое количество строк. Данные структурируются в виде таблиц практически повсеместно. В виде таблиц оформляются характеристики различных предметов и явлений, осуществляется отчетность. Более того, таблицы встречаются и в комплексах программного обеспечения, с помощью которых осуществляется управление аппаратурой компьютера и поддержка выполнения прикладных программ – операционных системах.

Цель данной лабораторной работы: реализация просмотровых таблиц.

# **Постановка задачи**

В данной лабораторной работе нам необходимо осуществить:

1. Разработку и реализацию класса таблицы TTab.
2. Разработку и реализацию класса элемента таблицы TElem.
3. Пример использования классов TTab и TElem.
4. Реализацию нескольких тестов на базе Google Test.

# **Руководство пользователя**

Для использования функционала библиотеки пользователю необходимо подключить заголовочный файл “TabLib.h”. После этого пользователь может создавать объекты класса TTab заданного размера и производить все необходимые действия с ними, а именно: добавлять элементы одного типа данных, производить поиск по ключу и удалять элементы.

# **Руководство программиста**

## **Описание структуры программы**

Программа состоит из следующих модулей:

* Модуль ScanTab. Содержит пример использования таблиц.
* Модуль ScanTabLib – статическая библиотека. Содержит файл TabLib.h, в котором описан интерфейс и реализация класса *TTab*. А также содержит файл TElemLib.h, в котором представлен интерфейс и реализация вспомогательного класса *TElem*.
* Модуль ScanTab Test. Содержит набор тестов, разработанных с помощью использования Google C++ Testing Framework.

## **Описание структур данных**

***Класс TElem:***

*Поля:*

* string key; - ключ записи;
* T data; - данные, содержащиеся в записи;

*Конструкторы и деструктор:*

* TElem (string \_key = “”, T \_data=0); - конструктор инициализации;
* TElem(TElem<T> &A); - конструктор копирования;

*Методы:*

* void SetKey(string \_key); - установить ключ;
* void SetData(T \_data); - установить данные;
* string GetKey(); - получить ключ;
* T GetData(); - получить данные;

*Перегруженные операторы:*

* TElem& operator = (TElem &A); - оператор присваивания;
* bool operator == (TElem &A); - оператор равно;
* bool operator != (TElem &A);- оператор неравно;

***Класс TTab:***

*Поля:*

* TElem<T> not\_found; - несуществующий элемент;
* TElem<T>\* mas; - массив элементов;
* int size, count; - размер количество элементов;

*Конструкторы и деструктор:*

* TTab (int \_size = 0); - конструктор инициализации;
* TTab (TTab& A); - конструктор копирования;

*Методы:*

* void SetSize (int \_size); - установить размер;
* void SetCount (int \_count); - установить количество заполненных элементов;
* TElem<T>\* GetMas(); - получить массив элементов;
* int GetSize(); - получить размер;
* int GetCount(); - получить количество элементов;
* void Put (string \_key, T \_data); - добавить строку к таблице;
* void Put (TElem<T> A); - добавить элемент к таблице;
* void Del (string \_key); - удалить элемент по ключу;
* TElem<T>& Search (string \_key); - найти элемент по ключу;
* bool IsFull (); - проверка таблицы на заполненность;
* bool IsEmpty(); - проверка таблицы на пустоту;

*Перегруженные операторы:*

* T operator[](string \_key); - оператор индексации;

## **Описание алгоритмов**

Метод **Put:**

Если таблица заполнена, бросаем исключение. Если такой ключ уже существует выводим предупреждение об этом. Иначе заполняем последнюю строчку таблицы и увеличиваем количество элементов на 1.

Метод **Del:**

Если таблица пустая, бросаем исключение. Иначе ищем значение элемента по ключу. Если такой элемент существует, то присваиваем ему значение несуществующего элемента.

Метод **Search**:

Проходимся по массиву элементов. Если такой ключ существует, то возвращаем его. Иначе возвращаем not\_found.

# **Заключение**

В ходе выполнения лабораторной работы была выполнена реализация такой структуры данных, как просмотровая таблица. А именно была разработана библиотека, реализующая класс таблиц со всем необходимым функционалом, показан пример использования, а также написан набор тестов с использованием Google C++ Testing Framework.

# **Литература**

Internet – ресурсы:

1. [https://ru.wikipedia.org/wiki/ Таблица\_(база\_данных)](https://ru.wikipedia.org/wiki/%20Таблица_(база_данных))

Книги:

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.: Пер. с англ. М.: Издат. дом «Вильямс», 2000. С. 58–76