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 таблиц»**

**Выполнил:** студент группы 381706-1

Колесова К.Ю.

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

аспирант каф. МОСТ ИИТММ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г.

Нижний Новгород

2019.

Оглавление

[1. Введение 3](#_Toc10247841)

[2. Постановка задачи 4](#_Toc10247842)

[3. Руководство пользователя 5](#_Toc10247843)

[4. Разрыв страницы 6](#_Toc10247844)

[4.1. Описание структуры программы 6](#_Toc10247845)

[4.2. Описание структур данных 6](#_Toc10247846)

[4.3. Описание алгоритмов 8](#_Toc10247847)

[5. Заключение 9](#_Toc10247848)

[6. Список литературы 10](#_Toc10247849)

# 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. Каждая строка однозначно определяется одним или несколькими уникальными значениями, которые принимают её ячейки из определенного подмножества столбцов. Подмножество столбцов, которое уникально идентифицирует строку, называется первичным ключом. Программная реализация таблиц возможна на основе различных структур данных, например, с использованием статических или динамических одномерных массивов, линейных списков или деревьев.

Цель данной лабораторной работы – реализация структуры данных для хранения таблиц с использованием динамических массивов.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Реализация класса таблицы TTab.
2. Реализация класса элемента таблицы TElem.
3. Реализация примера использования данных классов.
4. Реализация тестов на базе Google Test.

# Руководство пользователя

В качестве примера использования предлагается следующая тестовая программа:

1. В начале программы создается таблица с целочисленными данными. В нее заносятся три элемента с ключами А, В и С. Полученная таблица выводится на экран.
2. Далее пытаемся ввести еще один элемент с ключом А, на что выводится предупреждение о повторяющихся ключах.
3. Затем программа предлагает пользователю внести собственную строку с данными в таблицу.
4. Результат выводится на экран.
5. Затем осуществляется проверка методов таблицы. Сначала мы проверяем метод Search для элемента с ключом С.
6. Затем проверяем оператор индексации, который по ключу элемента (В) возвращает его данные.

На этом программа заканчивается.

# Разрыв страницы

## Описание структуры программы

Проект “Tab ” состоит из следующих файлов:

* mainTab.cpp -пример использования

Проект “ TabLib ” состоит из следующих файлов:

* TabLib.h - описание класса “TTab” реализация методов класса “TTab” и перегрузка операций
* TabLib.cpp – исполняемый файл класса “TTab”
* TElenLib.h - описание класса “TElem” реализация методов класса “TElem” и перегрузка операций

Проект “ TabLibTest” состоит из следующих файлов:

* test\_tablib.cpp - реализация тестов для класса TTab
* test\_main.cpp

## Описание структур данных

* Класс TElem:

Поля:

string key; - размер стека;

T data; - элемент, расположенный на вершине стека;

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 элемегтов;

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 зполненность;

bool IsEmpty(); - проверка таблицы на пустоту;

Перегруженные операторы:

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

## Описание алгоритмов

* Метод Put:

Если таблица заполнена - бросается исключение.   
Если такой ключ уже существует - выводится предупреждение об этом.  
Иначе - заполняем последнюю строчку таблицы и увеличиваем количество элементов на 1.

* Метод Del:

Если таблица пустая - бросается исключение.   
Иначе - ищем значение элемента по ключу.   
Если такой элемент существует - присваиваем ему значение несуществующего элемента.

* Метод Search:

Выполняется проход по массиву элементов.   
Если такой ключ существует - возвращаем его.

## Заключение

В ходе лабораторной работы были реализованы классы TTab и TElem. Так же была реализована структура данных для хранения таблиц с использованием динамических массивов. Так же были реализованы тесты на базе Google Test.

# Список литературы

1. Лабораторный практикум. Составители: Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Под редакцией Гергеля В.П. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017. – 105с.

URL: <http://www.unn.ru/books/met_files/Pract_ADS.pdf>