

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

«Национальный исследовательский

Нижегородский государственный университет им. Н.И. Лобачевского»  
(ННГУ)

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных  
технологий

## **ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

**«Реализация класса Просмотровая таблица»**

**Выполнил:** студент группы  
381706-2

Крюков Дмитрий Алексеевич

\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

\_\_\_\_\_ Подпись

Нижний Новгород

2019

## Содержание

<u>Содержание.....</u>	<u>2</u>
<u>1. Введение.....</u>	<u>3</u>
<u>2. Постановка задачи.....</u>	<u>4</u>
<u>3. Руководство пользователя.....</u>	<u>5</u>
<u>4. Руководство программиста.....</u>	<u>6</u>
<u>4.1 Описание структуры программы.....</u>	<u>6</u>
<u>4.2 Описание структур данных.....</u>	<u>6</u>
<u>4.3 Описание алгоритмов.....</u>	<u>7</u>
<u>5. Заключение.....</u>	<u>8</u>
<u>6. Литература.....</u>	<u>9</u>

# 1. Введение

Таблица – динамическая структура данных. Базисное множество – семейство линейных структур из записей, базисное отношение включения определяется операциями вставки и удаления записей

Таблица - последовательность строк (записей)

Запись может состоять из нескольких полей

Одно из полей должно задавать имя записи (ключ), остальные поля образуют тело записи

Операции под таблицей

Поиск записи по ключу

Вставка новой записи

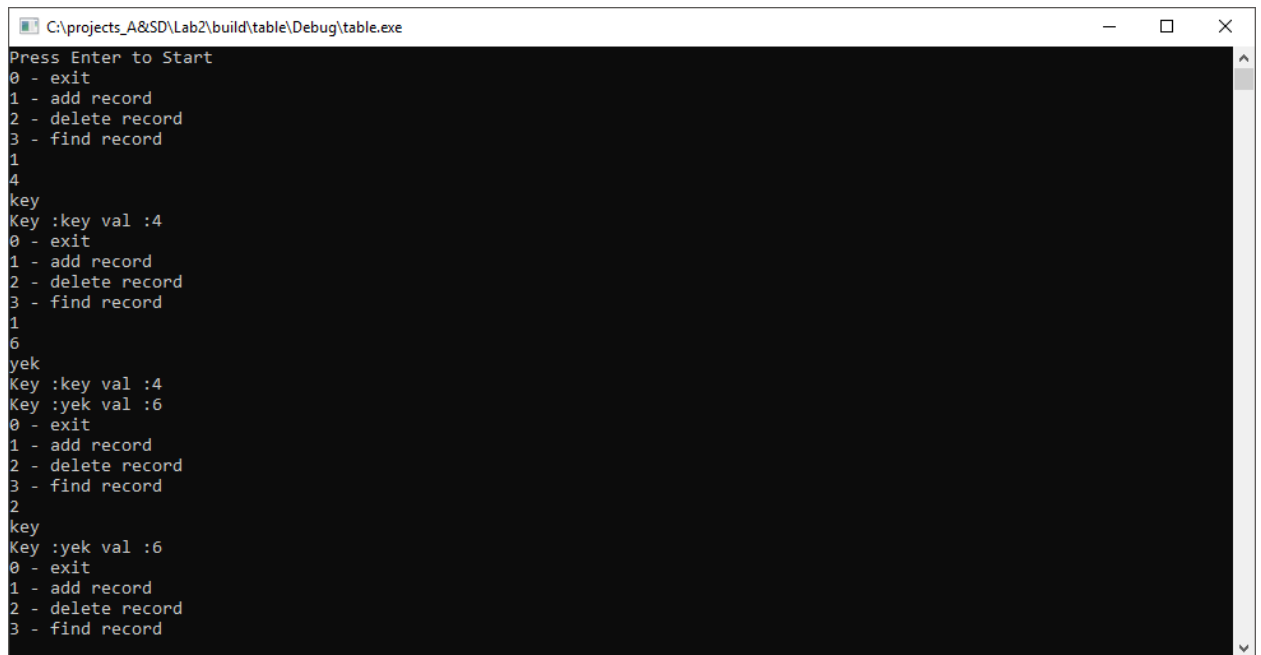
Удаление записи

## 2. Постановка задачи

- Организация доступа по имени для управления информацией в привычной для человека форме
- Представление данных во многих задачах из разных областей приложений (таблицы идентификаторов, номенклатура изделий, словари и т.п.)
- Абстрагирование от проблем распределения памяти при размещении данных
- Реализация концепции *ассоциативной памяти* (память с доступом к данным по содержимому в этих данных)
- Отображение на ЭВМ такого важного математического понятия как *множества*

### 3. Руководство пользователя

Данная программа предназначена для тестирования динамической структуры просмотрная таблица



```
C:\projects_A&SD\Lab2\build\table\Debug\table.exe
Press Enter to Start
0 - exit
1 - add record
2 - delete record
3 - find record
1
key
Key :key val :4
0 - exit
1 - add record
2 - delete record
3 - find record
1
6
yey
Key :yey val :6
0 - exit
1 - add record
2 - delete record
3 - find record
2
key
Key :key val :6
0 - exit
1 - add record
2 - delete record
3 - find record
```

пользователю предлагается набор команд :

- вставка записи в таблицу
- удаление записи из таблицы
- поиск записи в таблице

## 4. Руководство программиста

### 4.1 Описание структуры программы

1. Модуль `tablelib` (`TScanTable.h`, `TTabRecord.h`, `TKey.h`, `TKey.cpp`) – реализация класса таблица, запись таблицы, ключ
2. Модуль `table(main.cpp)` – реализация программы для тестирования динамической структуры просмотревая таблица
3. Модуль `tabletest(scantable_test.cpp, tabrecord_test.cpp, key_test.cpp,)` - тестирование класса просмотревая таблица при помощи Google C++ Testing Framework.

### 4.2 Описание структур данных

#### Структура

Поля:

`dataCount` — число записей в таблице

`tabSize` — максимальное число записей в таблице

`pRecs` – массив записей таблицы

Методы:

`Add(TTabRecord<ValType> *tr)` — добавляет запись в таблицу

`Delete(TKey k)` — удаляет запись из таблицы

`Search(TKey k)` — поиск по ключу

`operator[] (TKey k)` — обращение по ключу, в случае отсутствия записи добавляет

пустую запись с заданным ключом

### 4.3 Описание алгоритмов

Добавление:

если (tabSize == dataCount - 1)

ошибка

pRecs[dataCount] = tr

вернуть dataCount++

Поиск:

для i от 0 до dataCount

если pRecs[i]->key == k

вернуть pRecs[i]

вернуть NULL

## **5. Заключение**

В ходе работы был реализован класс просмотрная таблица, в нем реализованы функции удаления, вставки и поиска, а так же добавлена индексация по ключу



## **6. Литература**

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