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**

Направление подготовки «Прикладная математика и информатика»

**ОТЧЕТ**

по учебной практике

**Организация доступа к данным по ключу**

**Выполнила:** студентка группы 381603-1

Новикова Е.А.

**Проверила:** к.т.н.,доцент каф. МОСТ института ИТММ

Кустикова В.Д.

Нижний Новгород  
2018

**Содержание**

[Введение 3](#_Toc515895387)

[1 Постановка задачи 4](#_Toc515895388)

[2 Руководство пользователя 5](#_Toc515895389)

[3 Руководство программиста 6](#_Toc515895390)

[3.1 Описание структуры программы 6](#_Toc515895391)

[3.2 Описание структуры данных 6](#_Toc515895392)

[3.3 Описание программной реализация 7](#_Toc515895393)

[3.4 Описание программной реализация 7](#_Toc515895394)

[Заключение 9](#_Toc515895395)

[Список литературы 10](#_Toc515895396)

# Введение

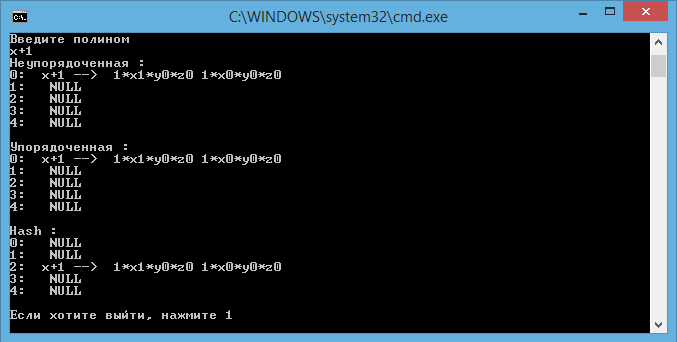
В современном мире люди часто используют таблицы для хранения различной информации. Таблицы сокращают время на поиски нужных данных, что очень важно для общества и разных сфер человеческой деятельности.

В данной лабораторной работе необходимо реализовать программу, которая будет осуществлять работу с таблицами трех видов, а именно: сортированными, несортированными, hash. С помощью данной программы можно выполнять вставку, удаление, поиск элемента.

# Постановка задачи

Формулировка задачи:

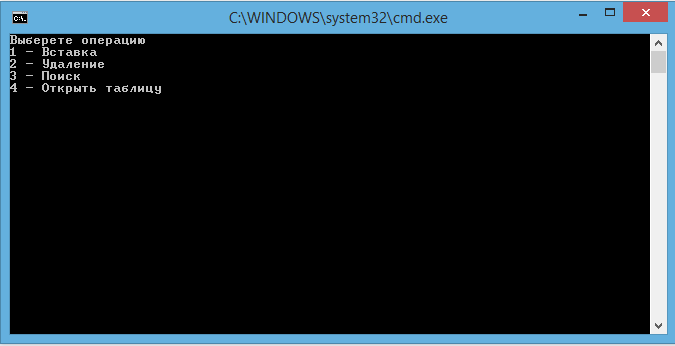
Необходимо разработать программу, которая будет работать с тремя видами таблиц: сортированные, несортированные, hash. Программа должна выполнять вставку, удаление или поиск элемента. В случае коллизии необходимо использовать два способа борьбы с ними: метод цепочек, метод открытого перемешивания. На вход программа должна получать размер таблицы, выбор действия (вставка, удаление, поиск), полиномы.

Пример работы программы:

1. Вставка элемента.

# Руководство пользователя

Для начала работы программы необходимо открыть файлsample.exe.После этого пользователю необходимо выбрать размер таблицы и операцию над элементом.

Пример работы программы:

1. Выбор операции

После этого пользователю выводится результат в зависимости от выбранной операции.

# Руководство программиста

## Описание структуры программы

Программа содержит в себе 5 проектов:

1. gtest (содержит в себе google тесты)
   1. gtest-h (заголовочный файл)
   2. gtest-all
2. polinom\_lib
   1. list.h (содержит в себе объявление и реализацию шаблонного класса list)
   2. monom.h (содержит в себе объявление шаблонного класса monom)
   3. node.h (содержит в себе объявление и реализацию шаблонного класса node)
   4. polinom.h (содержит в себе объявление шаблонного класса polinom)
   5. list.cpp
   6. node.cpp
   7. polinom.cpp (содержит в себе реализацию шаблонного класса polinom)
3. sample
   1. sample\_list.cpp (содержит в себе функцию main)
4. test
   1. test\_list.cpp (содержит в себе тесты для list)
   2. test\_main.cpp (содержит в себе функцию main, которая запускает google - тесты)
   3. test\_polinom.cpp (содержит в себе тесты для monom и polinom)
   4. test\_table.cpp (содержит в себе тесты для Tables)
5. Tables
   1. Tables.h (содержит объявление и реализацию виртуального класса)
   2. TablesHash.h (содержит объявление и реализацию класса hash-таблиц)
   3. TablesOrdered.h (содержит объявление и реализацию класса упорядоченных таблиц)
   4. TablesUnordered.h (содержит объявление и реализацию неупорядоченных таблиц)

## Описание структуры данных

Структура dataTable хранит два поля (данные и ключ)

Абстрактный, шаблонный класс таблиц – Table содержит реализацию методов навигации и автоматического перевыделения памяти, при котором данные копируются в массив большего размера.

unordtable - класс неупорядоченных таблиц (наследован от абстрактного класса Table)

Вставка:

После ввода данных его строка считается ключом. Если до заполнения таблицы осталось 30 процентов происходит увеличение размера массива в 1,65 раза. Старые данные копируются. Если в таблице нет совпадающих ключей, то вставляется в свободную ячейку, если ключи совпали, то срабатывает исключение.

Поиск:

Поиск производится по ключу. Если есть совпадения ключей, то возвращается адрес на структуру dataTable, иначе срабатывает исключение.

Удаление:

Удаление производится по ключу. Если есть совпадение ключей, то происходит его удаление и обнуление адреса данной ячейки, иначе срабатывает исключение.

ordertable - класс упорядоченных таблиц.

Вставка:

После ввода данных его строка считается ключом. Если до заполнения таблицы осталось 30 процентов происходит увеличение размера массива в 1,65 раза. Старые данные копируются. Для определения места в таблице, куда необходимо вставить полином, используя алгоритм бинарного поиска. Если ячейка не занята, то вставляем новый элемент, если занята, то происходит сдвиг всех элементов массива на одну ячейку, начиная с конца.

Поиск:

Поиск производится по ключу с помощью бинарного поиска. Если есть совпадения ключей, то возвращается адрес на структуру dataTable, иначе срабатывает исключение.

Удаление:

Удаление производится по ключу. Если есть совпадение ключей, то происходит его удаление и обнуление адреса данной ячейки, а так же сдвиг всех элементов на одну ячейку, иначе срабатывает исключение.

hashTable - класс хэш-таблиц

Вставка:

После ввода данных его строка считается ключом. Если до заполнения таблицы осталось 30 процентов происходит увеличение размера массива в 1,65 раза. Старые данные копируются. Хэш-функция принимает ключ, с его помощью генерирует число, это число будет использоваться для вставки в массив в качестве индекса. Если ячейка свободна – элемент вставляется, переменная, которая хранит количество вставленных элементов, увеличивает своё значение на единицу, затем ячейка помечается как занятая. Если ячейка была удалена происходит тоже самое, за исключение того, что переменная не увеличивает своё значение. Если ячейка занята и ключи равны, то срабатывает исключение.

Поиск:

Поиск производится по ключу. Ключ передаётся в хэш-функцию. Если есть совпадение ключей, то возвращается адрес на структуру dataTable, иначе срабатывает исключение.

Удаление:

Удаление производится по ключу. Ключ передаётся в хэш-функцию. Если есть совпадение ключей, то происходит его удаление и обнуление адреса данной ячейки, а так же сдвиг всех элементов на одну ячейку, иначе срабатывает исключение.

## Описание программной реализации

1. Table

virtual void insert(const typeK &k, const typeD &d) = 0; - вставка

virtual void delet(const typeK &k) = 0; - удаление

virtual dataTable<typeK, typeD> \* search(const typeK &k) const = 0; - поиск

void reset() - обнуление текущего

bool isended() const - проверка на конец

void getnext() - увеличение текущего индекса

typeD getcurdata() const - получить данные текущего элемента

1. unordtable

virtual void insert(const typeK &k, const typeD &d); - вставка в неупорядоченный таблице

virtual void delet(const typeK &k); - удаление элемента

virtual dataTable<typeK, typeD>\* search(const typeK &k) const; - поиск элемена по ключу

virtual void realocate(); - перераспределение памяти

1. ordertable

virtual void insert(const typeK &k, const typeD &d); - вставка в упорядоченную таблицу таблице

virtual void delet(const typeK &k); - удаление элемента

virtual dataTable<typeK, typeD>\* search(const typeK &k) const; - поиск элемента по ключу

virtual void realocate(); - перераспределение памяти

int Binarsearch(const typeK &key) const; - бинарный поиск

1. hashtable

void insert(const typeK &k, const typeD &d); - вставка в неупорядоченный таблице

void delet(const typeK &k); - удаление элемента

dataTable<typeK, typeD>\* search(const typeK &k) const; - поиск элемента по ключу

void realocate(); - перераспределение памяти

int Hashfunc(const typeK &k) const; - ищет место для вставки

int \*dm; - указатель на место вставки

# Заключение

В данной лабораторной работе была реализована программа, выполняющая работу с таблицами, а именно такие операции, как: вставку, поиск, удаления элементов. Данная программа поддерживает работу полиномов (введение полиномов пользователем, приведение подобных при записи в таблицу).

# Список литературы

1. *К* Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, Москва 2013
2. Гергель В.П. Рабочие материалы к учебному курсу «Методы программирования». 2002 г.