Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Hash-table(2)**

Дисциплина: «Основы алгоритмизации и программирования»

Тема : "Hash-таблицы"

Выполнил работу:

Студент группы РИС-22-1Б

Батин В.В.

Проверил:

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

**Постановка задачи**

Реализовать хэш-таблицу

Вариант 5:

ФИО, адрес, паспорт

H(k) = kmodM, где M - ФИО

Метод открытой адресации

**Код**

#include <iostream>

#include <string>

using namespace std;

const int DEFAULT\_SIZE = 100;

class Record {

public:

string key;

string address;

long int passport;

};

class HashTable {

private:

Record\* records;

int size;

int hashFunc(string key) {

int sum = 0;

for (int i = 0; i < key.length(); i++) {

sum += key[i];

}

return sum % size;

}

public:

HashTable(int size = DEFAULT\_SIZE) {

this->size = size;

records = new Record[size];

for (int i = 0; i < size; i++) {

records[i].key = "";

}

}

~HashTable() {

delete[] records;

}

void insert(string key, string address, long int passport) {

int index = hashFunc(key);

while (records[index].key != "") {

index = (index + 1) % size;

}

records[index].key = key;

records[index].address = address;

records[index].passport = passport;

}

Record\* find(string key) {

int index = hashFunc(key);

while (records[index].key != "") {

if (records[index].key == key) {

return &records[index];

}

index = (index + 1) % size;

}

return NULL;

}

void remove(string key) {

int index = hashFunc(key);

while (records[index].key != "") {

if (records[index].key == key) {

records[index].key = "";

records[index].address = "";

records[index].passport = 0;

return;

}

index = (index + 1) % size;

}

}

};

int main() {

setlocale(0, "");

HashTable ht;

ht.insert("Иванов Иван Иванович", "Москва, ул. Ленина, д.10, кв.5", 567890);

ht.insert("Петров Петр Петрович", "Санкт-Петербург, ул. Пушкина, д.5, кв.10", 654321);

ht.insert("Сидоров Сидор Сидорович", "Новосибирск, ул. Гагарина, д.15, кв.20", 543210);

Record\* rec1 = ht.find("Иванов Иван Иванович");

if (rec1 != NULL) {

cout << "Адрес: " << rec1->address << endl;

cout << "Номер паспорта: " << rec1->passport << endl;

}

else {

cout << "Запись не найдена" << endl;

}

ht.remove("Петров Петр Петрович");

Record\* rec2 = ht.find("Петров Петр Петрович");

if (rec2 != NULL) {

cout << "Адрес: " << rec2->address << endl;

cout << "Номер паспорта: " << rec2->passport << endl;

}

else {

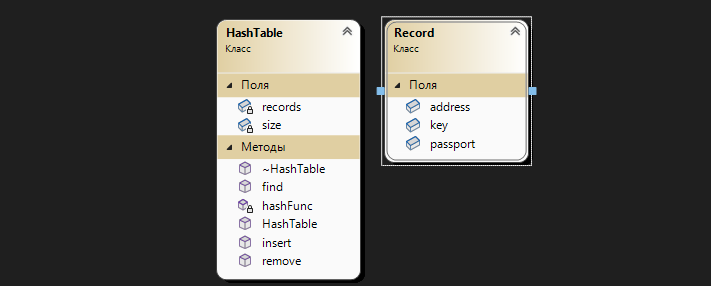
cout << "Запись не найдена" << endl;

}

return 0;

}

**UML диаграмма**

****