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**

Дисциплина: «Основы алгоритмизации и программирования»

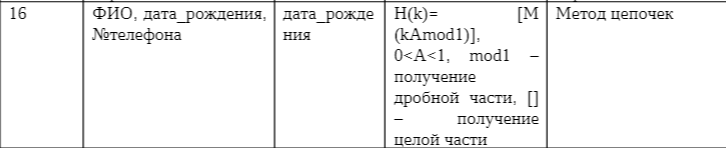
Тема: Хеш таблицы

Выполнил работу  
Студент группы РИС-22-1Б  
Ишемцева Мария Андреевна  
Проверил  
Доцент кафедры ИТАС  
Полякова Ольга Андреевна

Постановка задачи

Реализовать хэш таблицу.





Код:

#include <iostream>

#include <string>

#include <windows.h>

#include <ctime>

using namespace std;

int collis = 0;

string names[10] = { "Борислав", "Василий", "Ярослав", "Эрнест", "Архип", "Спиридон", "Илларион", "Федот", "Роберт", "Александр" };

string surnames[10] = {"Михеев", "Карпов", "Самсонов", "Королёв", "Гришин", "Третьяков", "Громов", "Юдин", "Муравьёв", "Ермаков" };

string last\_names[10] = { "Авенирович", "Вонифатович", "Игнатьевич", "Лаврентьевич", "Памфильевич", "Протасович", "Феофилактович", "Антипович", "Валериевич", "Арсениевич" };

string phones[10] = { "89820568317", "89522750080", "89230830754", "89943366719", "89629427448", "89871106348", "89038496205", "89829922085", "89322008547", "89064204539"};

string dates[10] = { "1990.12.03", "1984.09.10", "2003.10.06", "2000.03.31", "2001.02.22", "2016.11.17", "1992.02.09", "1974.04.02", "1991.02.02", "2002.12.12" };

struct human {

human() {

fio = "NULL";

date = "NULL";

number = "NULL";

}

string fio;

string date;

string number;

};

int random\_num() {

return rand() % 9;

}

struct hash\_table {

human\* arr;

hash\_table(int size) {

arr = new human[size];

}

~hash\_table() {

delete[] arr;

}

void add(human temp, int size);

void pop(string date, int size);

void find\_index(string date, int n);

};

int get\_date(string str) {

return stoi(str.substr(0, 4)) + stoi(str.substr(5, 2)) + stoi(str.substr(8, 2));

}

string randon\_fio() {

return surnames[random\_num()] + " " + names[random\_num()] + " " + last\_names[random\_num()];

}

string random\_phone() {

return phones[random\_num()];

}

string random\_date() {

return dates[random\_num()];

}

void humans(human\* arr, int count) {

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

arr[i].fio = randon\_fio();

arr[i].date = random\_date();

arr[i].number = random\_phone();

}

}

int hash\_function(string str, int size) {

double f = 0.344 \* get\_date(str);

double g = (f - int(f)) \* size;

return int(g);

}

void hash\_table::add(human tmp, int n) {

int idx = hash\_function(tmp.date, n);

int hash = idx;

if (arr[idx].fio == "NULL") {

arr[idx] = tmp;

return;

}

else {

while (idx < n) {

if (arr[idx].fio == "NULL") {

arr[idx] = tmp;

return;

}

idx++;

collis++;

}

if (idx >= n) {

idx = 0;

collis++;\

while (idx < hash) {

if (arr[idx].fio == "NULL") {

arr[idx] = tmp;

return;

}

idx++;

collis++;

}

}

}

}

void hash\_table::find\_index(string date, int n) {

int idx = hash\_function(date, n);

int hash = idx;

while (arr[idx].date != date && idx < n) {

idx++;

}

if (idx >= n) {

idx = 0;

while (arr[idx].date != date && idx < hash) {

idx++;

}

if (idx >= hash) {

cout << "Человек с датой рождения - " << date << " нет.\n";

}

}

else {

cout << "Человек с датой рождения - " << date << " содержится по индексу " << idx << endl;

}

}

void show\_info(human temp) {

cout << "ФИО: " << temp.fio << endl;

cout << "Дата рождения: " << temp.date << endl;

cout << "Телефон: " << temp.number << endl;

}

void show(hash\_table\* table, int n) {

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

show\_info(table->arr[i]);

}

}

void arr\_hum(human\* arr, int n) {

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

show\_info(arr[i]);

}

}

void hash\_table::pop(string date, int n) {

int idx = hash\_function(date, n);

if (arr[idx].date == date) {

arr[idx] = human();

return;

}

else {

while (idx < n) {

if (arr[idx].date == date) {

arr[idx] = human();

return;

}

idx++;

}

if (idx >= n) {

idx = 0;

while (idx < n) {

if (arr[idx].date == date) {

arr[idx] = human();

return;

}

idx++;

}

}

}

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

int n;

cout << "Введите количество элементов(не меньше ста): ";

cin >> n;

human\* arr = new human[n];

hash\_table table(n);

humans(arr, n);

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

table.add(arr[i], n);

}

show(&table, n);

table.find\_index("1990.12.03", n);

table.pop("1990.12.03", n);

show(&table, n);

cout << "\nКоличество коллизий для " << n << " записей: " << collis << endl;

delete[] arr;

return 0;

}

Блок схема:

