**Министерство высшего образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)**

**Электротехнический факультет**

**Дисциплина: основы алгоритмизации и программирования, 2 семестр**

**ОТЧЁТ**

**Тема: «HASH»**

Выполнил

Студент РИС-22-2б

Третьяков Н.А.

Проверил

Доцент кафедры ИТАС

Полякова О. А.

Пермь 2023

Содержание

Оглавление

[Логическое применение HASH 3](#_Toc134013770)

[Разбор видов списков 4](#_Toc134013771)

[Блок схема 5](#_Toc134013772)

[Программный код 6](#_Toc134013773)

[Приложение 11](#_Toc134013774)

# Логическое применение HASH

Хэши это таблица данных в которой доступ к элементам получается с помощью ключа.

Это необходимо для того чтобы определённым образом защищать данные

При работе с Хэш таблицами может появляется проблема что на определенном месте, соответствующем ключу, уже находится элемент это называется коллизии.

Хэш разделяются по способам реализации обработки коллизии, чтобы хранящиеся данные не накладывались друг на друга

# Разбор видов списков

Существуют два вида Хэшей с открытой адресацией и цепочками

Открытая адресация – суть метода заключается в том, что если в найденной ячейки есть элемент, то записать данные в следующую ячейку или идти по Хэшу пока не найдётся пустая ячейка.

Цепочки – суть метода заключается в том, что если в найденной ячейке есть элемент, то в имеющемся элементе сохранить адрес нового элемента если какой-то элемент уже сохранён, то искать по данной цепочки элемент, у который не хранит адрес на следующий элемент.

# Блок схема

Блок схема будет реализовывать Хэш с открытой адресацией, приложения находятся в файле из Приложения

Приложение 1 структура для хранения данных

Приложение 2 основной код программы с выбором действий

Приложение 3 Меню в котором описано какие действия можно делать

Приложение 4 Ввод элемента в Хэш

Приложение 5 Ввод в Хэш нескольких элементов

Приложение 6 Поиск элемента в Хэше по ключу

Приложение 7 Вывод Хэша на экран

Приложение 8 Функция для подсчёта суммы цифр числа и расчёт по ним ключа

# Программный код

#include <iostream>

#include <string>

#include <cmath>

#include <Windows.h>

using namespace std;

struct Element

{

string FIO = "";

string data;

string adress;

};

Element\* Mas;

int N = 0;

int Hash(int sum, int HashTable);

int Sum(string str);

void CinHash(int i);

void CinHashK(int i);

void CoutHash(int i);

void Chek(int i);

void Menu();

int main()

{

setlocale(0, "RUS");

SetConsoleCP(1251);

bool F = 1;

int i;

int n;

cout << "Enter how many elements can be stored in the HASH table" << endl;

cin >> i;

Mas = new Element[i];

while (F)

{

Menu();

cout << "Enter the action number" << endl;

cin >> n;

system("cls");

switch (n)

{

case 1:

{

CinHash(i);

break;

}

case 2:

{

CinHashK(i);

break;

}

case 3:

{

CoutHash(i);

break;

}

case 4:

{

Chek(i);

break;

}

case 5:

{

F = 0;

break;

}

default:

cout << "Enter the number of the item that is in the menu" << endl;

}

}

delete[] Mas;

}

void Menu()

{

cout << "\t Menu" << endl;

cout << "1) Enter 1 element into the table" << endl;

cout << "2) Enter several elements into the table" << endl;

cout << "3) Display the table on the screen" << endl;

cout << "4) Searching for an element by key" << endl;

cout << "5) Finish working with the table" << endl;

}

void CinHash(int i)

{

bool f = 1;

int k;

Element element;

cout << "Enter the data of the elements" << endl;

cout << "Enter your full name" << endl;

getline(cin >> ws, element.FIO);

cout << "Enter your date of birth" << endl;

getline(cin >> ws, element.data);

cout << "Enter your residential address" << endl;

getline(cin >> ws, element.adress);

int sum = Sum(element.FIO);

int key = Hash(sum, i);

if (Mas[key].FIO == "")

{

Mas[key] = element;

}

else

{

k = key + 1;

while ((k < i) && f)

{

if (Mas[k].FIO == "")

{

Mas[k] = element;

f = 0;

}

else

{

k++;

}

}

if (f)

{

k = 0;

while ((k < key) && f)

{

if (Mas[k].FIO == "")

{

Mas[k] = element;

f = 0;

}

else

{

k++;

}

}

}

f = 1;

}

}

void CinHashK(int i)

{

int n;

bool f = 1;

int k;

Element element;

cout << "Enter the number of items to be entered" << endl;

cin >> n;

while (n > 0)

{

cout << "Enter the data of the elements" << endl;

cout << "Enter your full name" << endl;

getline(cin >> ws, element.FIO);

cout << "Enter your date of birth" << endl;

getline(cin >> ws, element.data);

cout << "Enter your residential address" << endl;

getline(cin >> ws, element.adress);

int sum = Sum(element.FIO);

int key = Hash(sum, i);

if (Mas[key].FIO == "")

{

Mas[key] = element;

}

else

{

k = key + 1;

while ((k < i) && f)

{

if (Mas[k].FIO != "")

{

Mas[k] = element;

f = 0;

}

else

{

k++;

}

}

if (f)

{

k = 0;

while ((k < key) && f)

{

if (Mas[k].FIO != "")

{

Mas[k] = element;

f = 0;

}

else

{

k++;

}

}

}

f = 1;

}

n--;

}

}

void Chek(int i)

{

bool f = 1;

string a;

cout << "Enter your full name key:" << endl;

getline(cin >> ws, a);

int sum = Sum(a);

int key = Hash(sum, i);

if (Mas[key].FIO == a)

{

cout << Mas[key].FIO << endl;

cout << Mas[key].data << endl;

cout << Mas[key].adress << endl;

}

else

{

int k = key + 1;

while ((k < i) && (f))

{

if (Mas[k].FIO == a)

{

f = 0;

cout << Mas[k].FIO << endl;

cout << Mas[k].data << endl;

cout << Mas[k].adress << endl;

}

k++;

}

if (f)

{

k = 0;

while ((k < key) && (f))

{

if (Mas[k].FIO == a)

{

f = 0;

cout << Mas[k].FIO << endl;

cout << Mas[k].data << endl;

cout << Mas[k].adress << endl;

}

k++;

}

}

if (f)

{

cout << "The required element is not in the table or the key is entered incorrectly" << endl;

}

}

}

void CoutHash(int i)

{

int k = 0;

int m = 1;

while (i > 0)

{

if (Mas[k].FIO != "")

{

cout << m << ") " << Mas[k].FIO << endl;

cout << " " << Mas[k].data << endl;

cout << " " << Mas[k].adress << endl;

m++;

}

k++;

i--;

}

}

int Sum(string str)

{

int sum = 0;

for (int k = 0; k < str.size(); k++)

{

sum = sum + (int)str[k];

}

sum = abs(sum);

return sum;

}

int Hash(int sum, int HashTable)

{

int Key = sum % HashTable;

return Key;

}

# Приложение

Hash. drawio