Министерство науки и высшего образования РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа киберфизических систем и управления

УДК 004.421

УТВЕРЖДАЮ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_г.

Отчет

по дисциплине «Теория и технология программирования»

Лабораторная работа №8 «Методы хэширования», вариант 17

Выполнил:

Студент

гр.3530902/90001 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Непушкин С.А. подпись, дата

Доцент ВШКФСиУ,

Кандидат технических наук

С. В. Хлопин \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Санкт-Петербург

2020

**1. Цель работы.**

Цель задания - овладеть методами хэширования

**2. Задание:**

Составить программу для поиска по хэшам данных. Хэширование проводить в соответствии с индивидуальными заданиями. В модуле поиска, предусмотреть реализацию обработки случая, при котором хэш-коды различных данных совпадают

**3. Текст программы**

#include<iostream>

#include<fstream>

#include<string>

#include<list>

using namespace std;

struct Text {

list<pair<int, string>>words;

//const int p = 137; // если и большие и маленькие

//const int p = 31; если только маленькие

Text(string filename)

{

ifstream ifs(filename);

while (!ifs.eof()) // ifs-input file stream

{

string str;

ifs >> str; // только 1 слово

// getline(ifs,str); всю строку

words.push\_back(pair<int, string>(hash(str), str));

}

ifs.close();

}

void print()

{

for (list<pair<int, string>>::iterator it = words.begin(); it != words.end(); it++)

{

cout << "\"(" << (\*it).first << ")" << (\*it).second << "\"" << endl;

//cout << (\*it).second << " " << endl;

}

}

int compareStrings(string s1, string s2)

{

int result = 0;

for (int i = 0; i < s1.length() && i < s2.length(); i++)

{

result += s1[i] - s2[i];

if (result != 0)

{

break;

}

}

return result;

}

int hash(string str)

{

int result = 0;

//int base = 1;

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

{

result += str[i];

//base \*= p;

}

return result;

}

//моя исходная функция

/\*void findWord(string word)

{

int h = hash(word);

int count = 0;

int i = 0;

for (list<pair<int, string>>::iterator it = words.begin(); it != words.end(); it++, i++)

{

if (((\*it).first == h) && (compareStrings((\*it).second, word) == 0))

{

cout << i + 1 << " ";

count++;

}

}

cout << endl << "found " << count << " words" << endl;

}\*/

//моя доработанная функция

void findWord(string word)

{

int h = hash(word);

int count1 = 0;

int count2 = 0;

int i = 0;

for (list<pair<int, string>>::iterator it = words.begin(); it != words.end(); it++, i++)

{

if ((\*it).first == h)

{

count1++;

if (compareStrings((\*it).second, word) == 0)

{

count2++;

}

}

}

cout << endl << "using only hash we found " << count1 << " words" << endl;

cout << "using also check hash we found " << count2 << " words" << endl;

}

};

int main()

{

Text text("in.txt");

text.print();

cout << "input word to find" << endl;

string str;

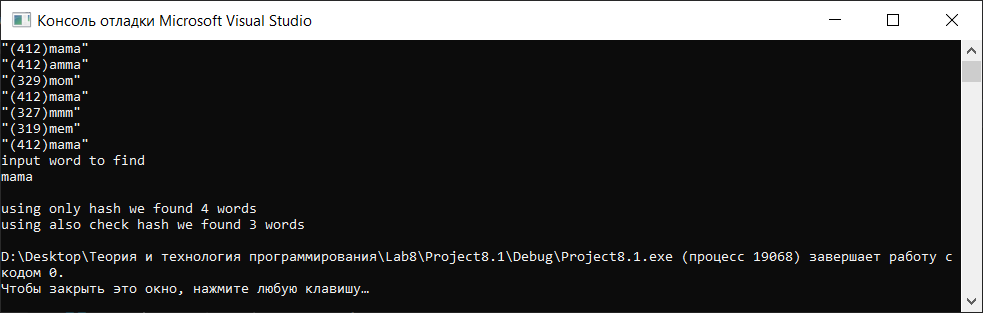
cin >> str;

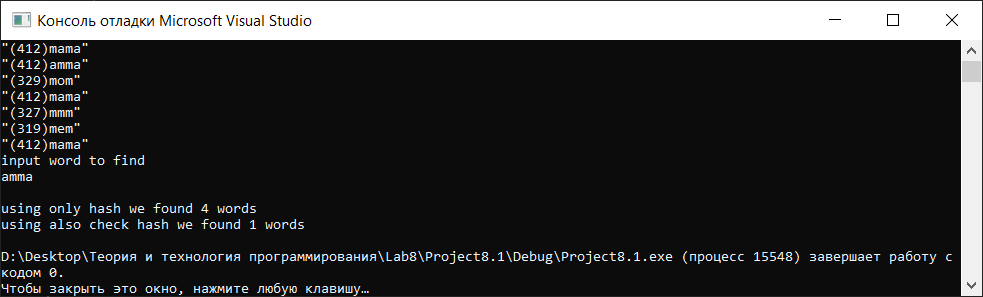
text.findWord(str);

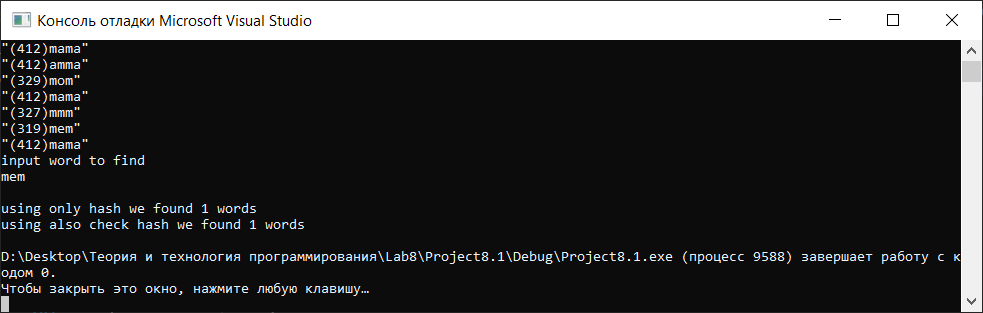
return 0;

}

**4. Пример работы программы**







**Вывод**

В ходе лабораторной работы я:

-узнал о классе “list”

-узнал о такой структуре данных, как “iterator”

-узнал о том, как производится поиск по хэшу

-узнал о библиотеке “list”

-столкнулся с проблемой присваивания хэша слову, но успешно решил ее