Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №17**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Методы быстрого поиска. Хеширование

Вариант 1

Выполнил:

Студент группы РИС-20-1б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Ознакомление с методами быстрого поиска на основе Хеширования

**Постановка задачи**

1. Создать динамический массив из записей (в соответствии с вариантом), содержащий не менее 100 элементов. Для заполнения элементов массива использовать ДСЧ.

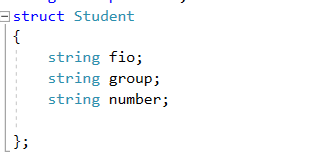
2. Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска использовать хеш-таблицу.

3. Подсчитать количество коллизий при размере хеш-таблицы 40, 75 и 90 элементов.

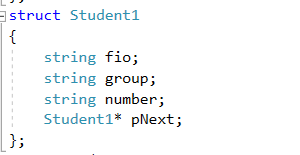
**Анализ задачи**

**1.** Для решения задачи необходимо:

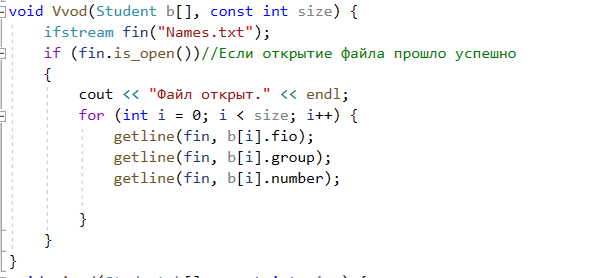
**1.1.** Реализовать структуру для дальнейшей работы с методами быстрыми поиска



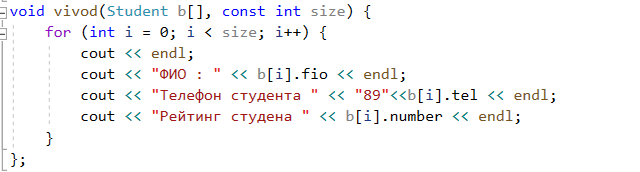
**1.2.** Реализовать новую структуру с указанием на предыдущую для удобства реализации в дальнейших функциях

:

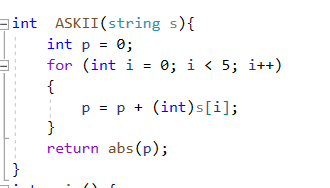
**1.3.** Реализовать прием данных с файла Names.txt



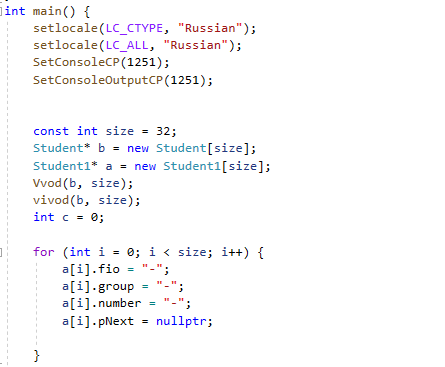
**1.4.** Реализовать функцию для вывода данных на консоль:



**1.3.** Реализовать функцию для перевода элементов типа string в числовое значение:

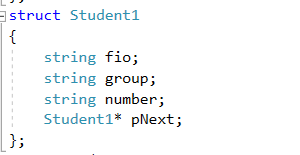
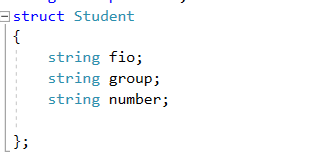


**1.4.** Реализовать главную функцию по демонстрации всех вышеперечисленных функций:



**2.**В ходе работы были использованы следующие типы данных:

**2.1.** Для составления структур данных использовались данные string,



**2.2.** Для работы с данными из файла использовалась постоянная целочисленная переменная



**2.3.** Для подсчета коллизий использовалась целочисленная переменная:

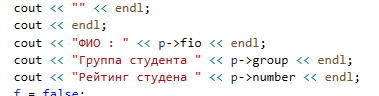


**3.**Для решения задачи данные были представлены в следующем виде:

**3.1.** Для удобной обработки данных для элементов использовалась переменная типа string.

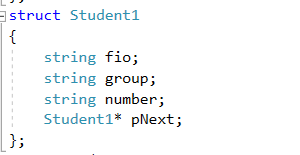
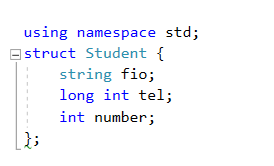
**4.**Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.**Вывод данных из массива реализован через функцию cout:

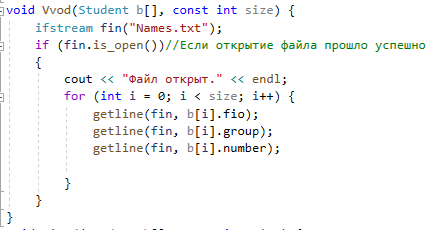


**5.**Поставленные задачи будут решены следующими действиями:

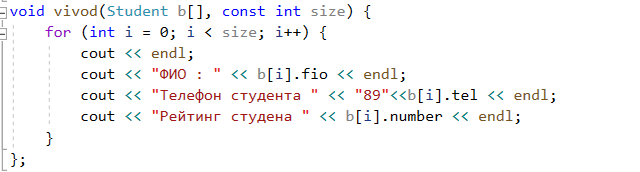
**5.1.** Для удобной обработки элементов было принято решение присвоить элементам структур тип данных string



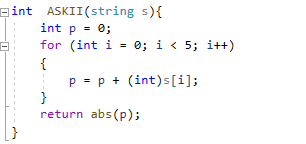
**5.2.** Для реализации ввода переменных в файл было принято решение по созданию файла Names.txt, где при помощи цикла for и функции getline считываются данные с заданного файла:



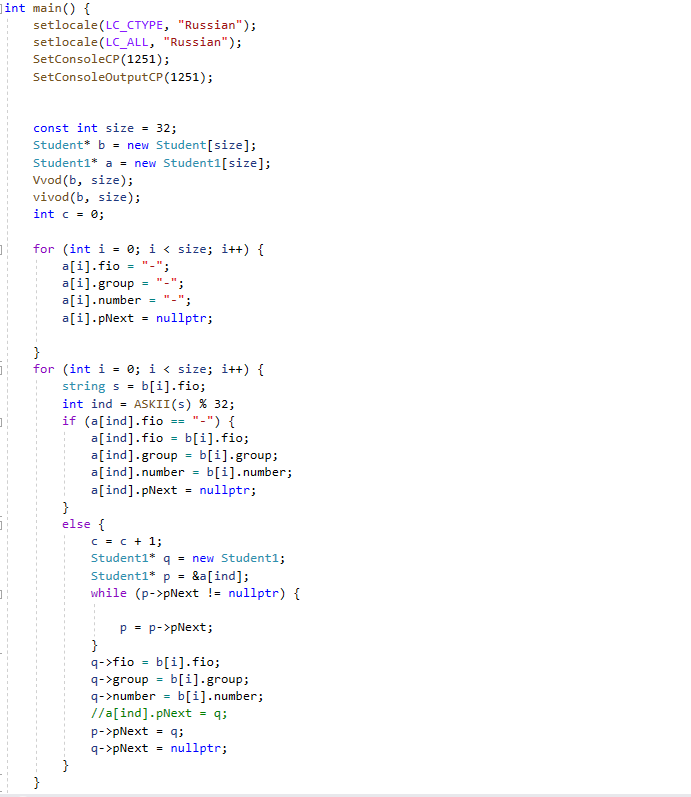
**5.3.** Для реализации вывода переменных используется цикл for, выводящий все элементы структуры через функцию cout:

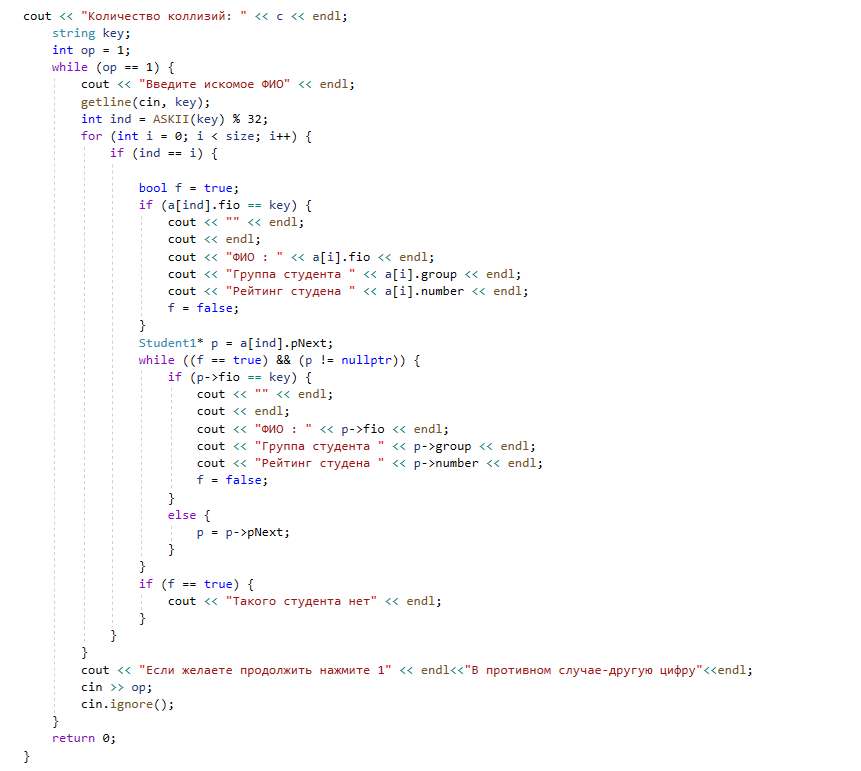


**5.4.** Для реализации функции по переводу элементов типа string в целочисленный элемент использовалась таблица ASKII, с дальнейшим суммированием всех переведённых символов строки в тип int

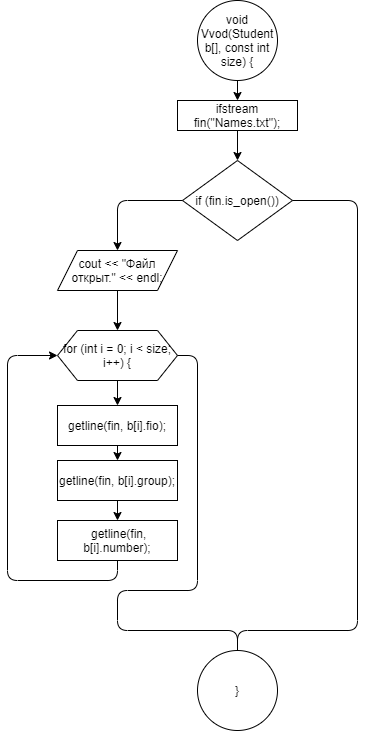


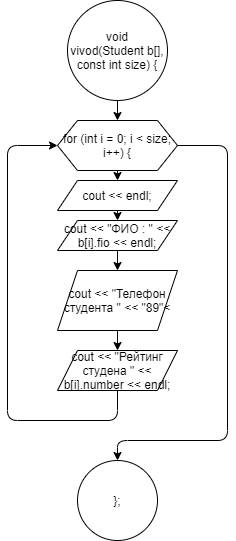
**5.5.** Для реализации метода хеширования всем элементам второй структуры присваивается пустое значение. Затем при переводе элементов структуры в целочисленный тип данных происходит обращение к элементу данных данного значения с дальнейшей обработкой:

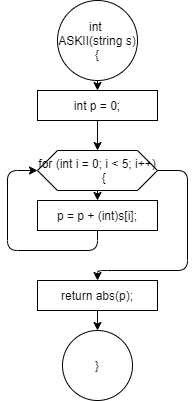


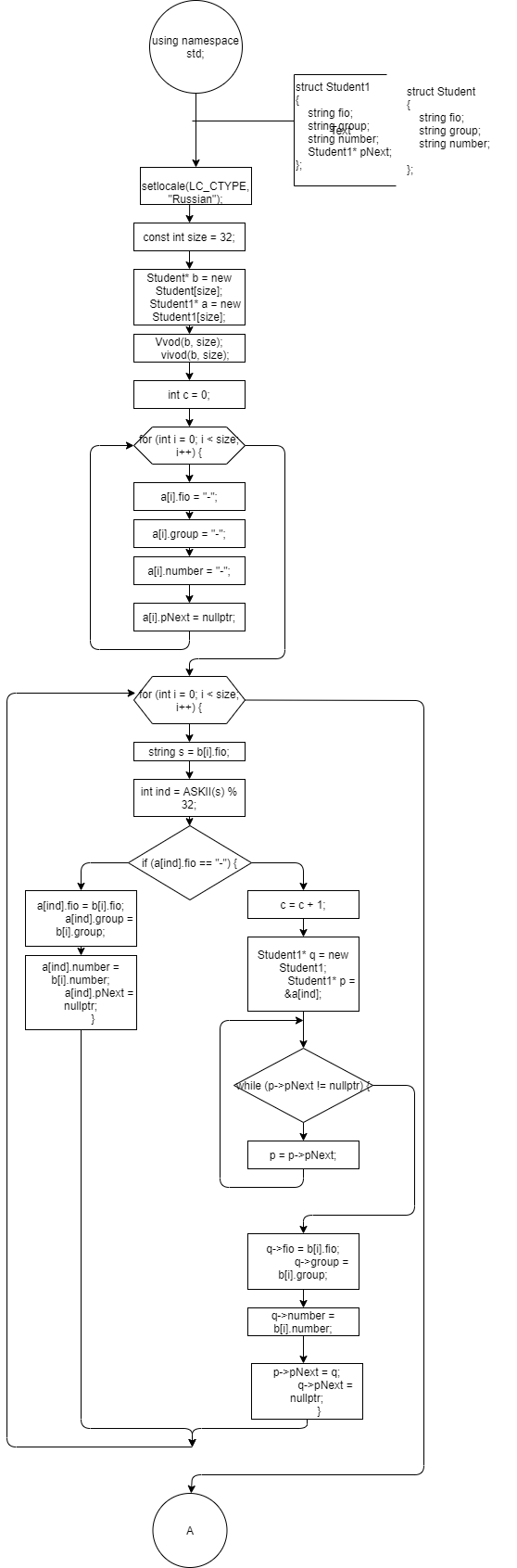


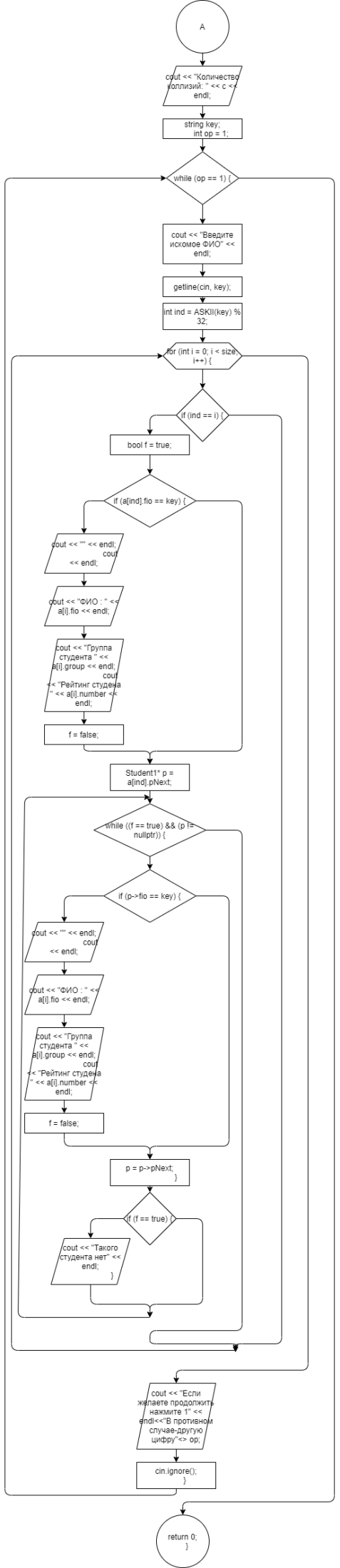
**Блок-схема программы**

****

****

****

****

****

**Решение**

#include <iostream>

#include <list>

#include <fstream>

#include <windows.h>

#include <math.h>

#include <string>

using namespace std;

struct Student

{

string fio;

string group;

string number;

};

struct Student1

{

string fio;

string group;

string number;

Student1\* pNext;

};

void Vvod(Student b[], const int size) {

ifstream fin("Names.txt");

if (fin.is\_open())//Если открытие файла прошло успешно

{

cout << "Файл открыт." << endl;

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

getline(fin, b[i].fio);

getline(fin, b[i].group);

getline(fin, b[i].number);

}

}

}

void vivod(Student b[], const int size) {

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

cout << endl;

cout << "ФИО : " << b[i].fio << endl;

cout << "Группа студента " << b[i].group << endl;

cout << "Рейтинг студена " << b[i].number << endl;

}

};

int ASKII(string s){

int p = 0;

for (int i = 0; i < 5; i++)

{

p = p + (int)s[i];

}

return abs(p);

}

int main() {

setlocale(LC\_CTYPE, "Russian");

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int size = 32;

Student\* b = new Student[size];

Student1\* a = new Student1[size];

Vvod(b, size);

vivod(b, size);

int c = 0;

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

a[i].fio = "-";

a[i].group = "-";

a[i].number = "-";

a[i].pNext = nullptr;

}

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

string s = b[i].fio;

int ind = ASKII(s) % 32;

if (a[ind].fio == "-") {

a[ind].fio = b[i].fio;

a[ind].group = b[i].group;

a[ind].number = b[i].number;

a[ind].pNext = nullptr;

}

else {

c = c + 1;

Student1\* q = new Student1;

Student1\* p = &a[ind];

while (p->pNext != nullptr) {

p = p->pNext;

}

q->fio = b[i].fio;

q->group = b[i].group;

q->number = b[i].number;

//a[ind].pNext = q;

p->pNext = q;

q->pNext = nullptr;

}

}

cout << "Количество коллизий: " << c << endl;

string key;

int op = 1;

while (op == 1) {

cout << "Введите искомое ФИО" << endl;

getline(cin, key);

int ind = ASKII(key) % 32;

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

if (ind == i) {

bool f = true;

if (a[ind].fio == key) {

cout << "" << endl;

cout << endl;

cout << "ФИО : " << a[i].fio << endl;

cout << "Группа студента " << a[i].group << endl;

cout << "Рейтинг студена " << a[i].number << endl;

f = false;

}

Student1\* p = a[ind].pNext;

while ((f == true) && (p != nullptr)) {

if (p->fio == key) {

cout << "" << endl;

cout << endl;

cout << "ФИО : " << p->fio << endl;

cout << "Группа студента " << p->group << endl;

cout << "Рейтинг студена " << p->number << endl;

f = false;

}

else {

p = p->pNext;

}

}

if (f == true) {

cout << "Такого студента нет" << endl;

}

}

}

cout << "Если желаете продолжить нажмите 1" << endl<<"В противном случае-другую цифру"<<endl;

cin >> op;

cin.ignore();

}

return 0;

}

**Скриншоты**

