Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №8**

Дисциплина: основы алгоритмизации и программирования

Тема: “ Блоковый ввод-вывод ” в C++

Вариант 24

Выполнил работу

студент группы РИС-20-1б

Коваленко Н. А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель задачи**

Цель – написать программу с использованием структур данных и

выполнение операций с ними.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Понять, как создаются и объявляются структуры данных.
2. Понять, как работают структуры данных.
3. Понять, оптимальные методы ввода информации в структуру

данных.

1. Разработать программу по найденному материалу.

Постановка задачи

Структура "Студент":

- фамилия, имя, отчество;

- номер телефона;

- группа;

- оценки по 3 основным предметам.

Удалить элемент из группы с указанным номером, у которого среднее арифметическое оценок меньше заданного, добавить элемент после элемента с заданной фамилией.

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Поиск по номеру
* Добавление элемента после элемента с заданной фамилией
* Сравнение по среднему арифметическому оценок

1. Для решения поставленной задачи была создана структура Student со следующими полями:

Для ФИО созданы 3 поля типа string

Для номера телефона создано поле типа string

Для оценок создано 3 поля типа int

1. Разработать структуру Students и объявить ее

struct students

{

string name;

string surname1;

string surname;

string phonenumber;

int mark1; int mark2; int mark3;

};

1. Создание массива структур для простоты обращения к полям структуры:

students N1 = {"Георгий", "Андреевич", "Монин", "+7(902)45-67928", 5, 2, 2};

students N2 = { "Никита", "Алексеевич", "Коваленко", "+7(908)277-1780", 5, 5, 5 };

students N3 = { "Алексей", "Павлович", "Солдатов", "+7(902)45-32228", 3, 4, 2 };

students N4 = { "Иван", "Иванович", "Иванов", "+7(902)55-64238", 2, 4, 3 };

students N5 = { "Петр", "Алексеевич", "Романов", "+7(961)18-59348", 2, 5, 3 };

students N6 = { "Александр", "Мамонов", "Александрович", "+7(961)22-58651", 5, 5, 5 };

students a[6] = { N6, N1, N2, N3, N4, N5 };

1. Ввод данных осуществляется посредством функции

cout << " " << i << ") " << '\t' << a[i].name << endl << '\t' << a[i].surname1 << endl << '\t' << a[i].surname << endl << '\t' << a[i].phonenumber << endl << '\t' << a[i].mark1 << " " << a[i].mark2 << " " << a[i].mark3 << endl << endl;

1. Для решения задачи (поиска нужного номера, сравнения среднего арифметического оценок, поиска нужной фамилии) используются циклы и ветвления. Например, if (i == n && (a[i].mark1 + a[i].mark2 + a[i].mark3)/3 < m) i++; используется для сравнения среднего арифметического оценок.

Код

#include <iostream>

#include <string>

using namespace std;

struct students

{

string name;

string surname1;

string surname;

string phonenumber;

int mark1; int mark2; int mark3;

};

int main()

{

system("chcp 1251>nul");

int n, m, i;

string surnamecheck;

students N1 = {"Георгий", "Андреевич", "Монин", "+7(902)45-67928", 5, 2, 2};

students N2 = { "Никита", "Алексеевич", "Коваленко", "+7(908)277-1780", 5, 5, 5 };

students N3 = { "Алексей", "Павлович", "Солдатов", "+7(902)45-32228", 3, 4, 2 };

students N4 = { "Иван", "Иванович", "Иванов", "+7(902)55-64238", 2, 4, 3 };

students N5 = { "Петр", "Алексеевич", "Романов", "+7(961)18-59348", 2, 5, 3 };

students N6 = { "Александр", "Мамонов", "Александрович", "+7(961)22-58651", 5, 5, 5 };

students a[6] = { N6, N1, N2, N3, N4, N5 };

for (i = 1; i < 6; i++)

cout << " " << i << ") " << '\t' << a[i].name << endl << '\t' << a[i].surname1 << endl << '\t' << a[i].surname << endl << '\t' << a[i].phonenumber << endl << '\t' << a[i].mark1 << " " << a[i].mark2 << " " << a[i].mark3 << endl << endl;

cout << "Введи номер, который нужно удалить: "; cin >> n;

cout << "Введи среднее арифметическое оценок: "; cin >> m;

cout << "Введи фамилию, после которой добавить элемент: "; cin >> surnamecheck;

cout << "Изменения:" << endl;

i = 1;

while (i != n && a[i].surname != surnamecheck && i < 6)

{

cout << '\t' << a[i].name << endl << '\t' << a[i].surname1 << endl << '\t' << a[i].surname << endl << '\t' << a[i].phonenumber << endl << '\t' << a[i].mark1 << " " << a[i].mark2 << " " << a[i].mark3 << endl << endl;

i++;

}

if (i == n && (a[i].mark1 + a[i].mark2 + a[i].mark3)/3 < m) i++;

if (a[i].surname == surnamecheck)

{

cout << '\t' << a[i].name << endl << '\t' << a[i].surname1 << endl << '\t' << a[i].surname << endl << '\t' << a[i].phonenumber << endl << '\t' << a[i].mark1 << " " << a[i].mark2 << " " << a[i].mark3 << endl << endl;

i++;

cout << '\t' << a[0].name << endl << '\t' << a[0].surname1 << endl << '\t' << a[0].surname << endl << '\t' << a[0].phonenumber << endl << '\t' << a[0].mark1 << " " << a[0].mark2 << " " << a[0].mark3 << endl << endl;

}

while (i != n && a[i].surname != surnamecheck && i<6)

{

cout << '\t' << a[i].name << endl << '\t' << a[i].surname1 << endl << '\t' << a[i].surname << endl << '\t' << a[i].phonenumber << endl << '\t' << a[i].mark1 << " " << a[i].mark2 << " " << a[i].mark3 << endl << endl;

i++;

}

if (i == n && (a[i].mark1 + a[i].mark2 + a[i].mark3) / 3 < m) i++;

if (a[i].surname == surnamecheck)

{

cout << '\t' << a[i].name << endl << '\t' << a[i].surname1 << endl << '\t' << a[i].surname << endl << '\t' << a[i].phonenumber << endl << '\t' << a[i].mark1 << " " << a[i].mark2 << " " << a[i].mark3 << endl << endl;

i++;

cout << '\t' << a[0].name << endl << '\t' << a[0].surname1 << endl << '\t' << a[0].surname << endl << '\t' << a[0].phonenumber << endl << '\t' << a[0].mark1 << " " << a[0].mark2 << " " << a[0].mark3 << endl << endl;

}

while (i < 6)

{

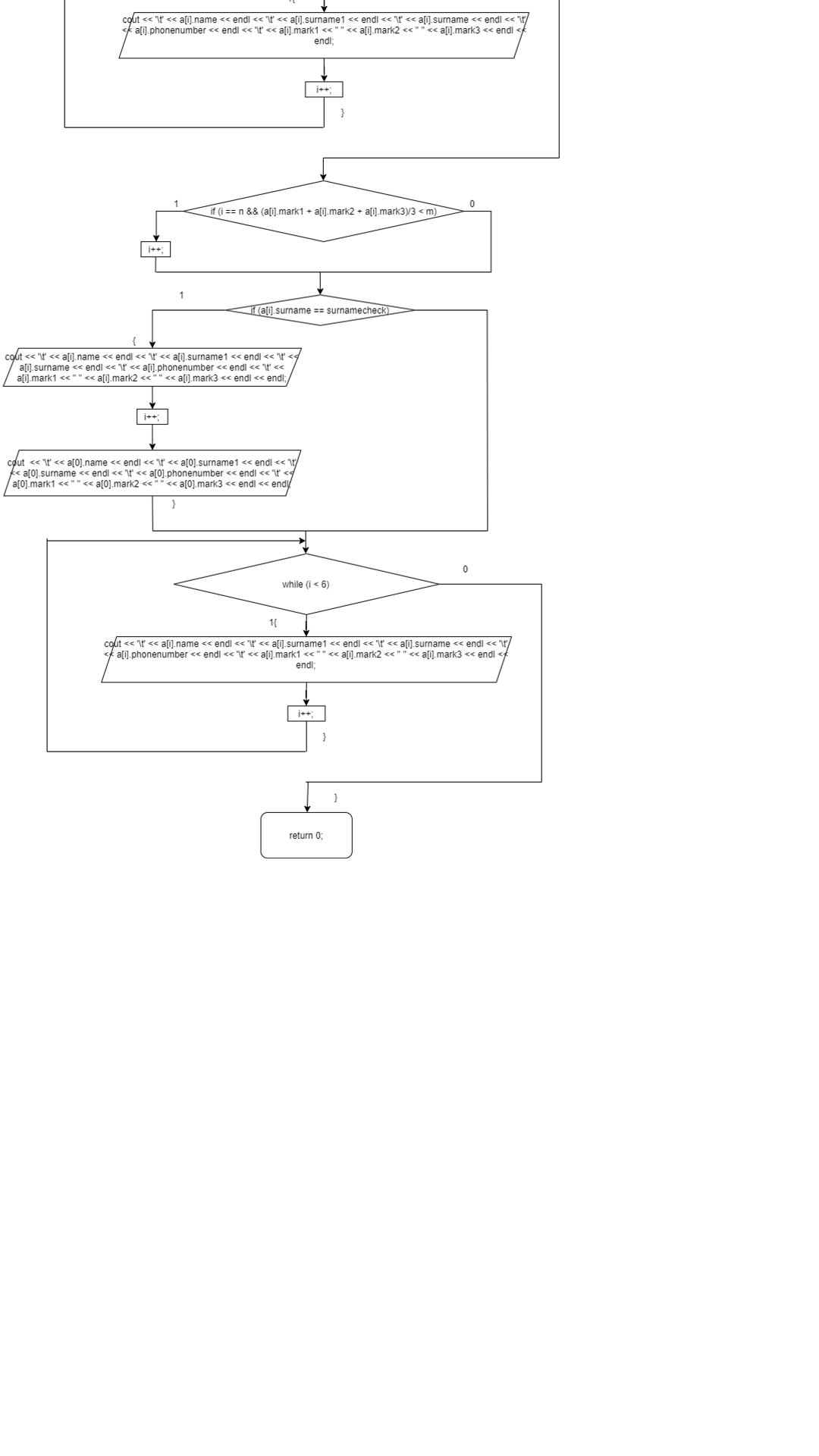
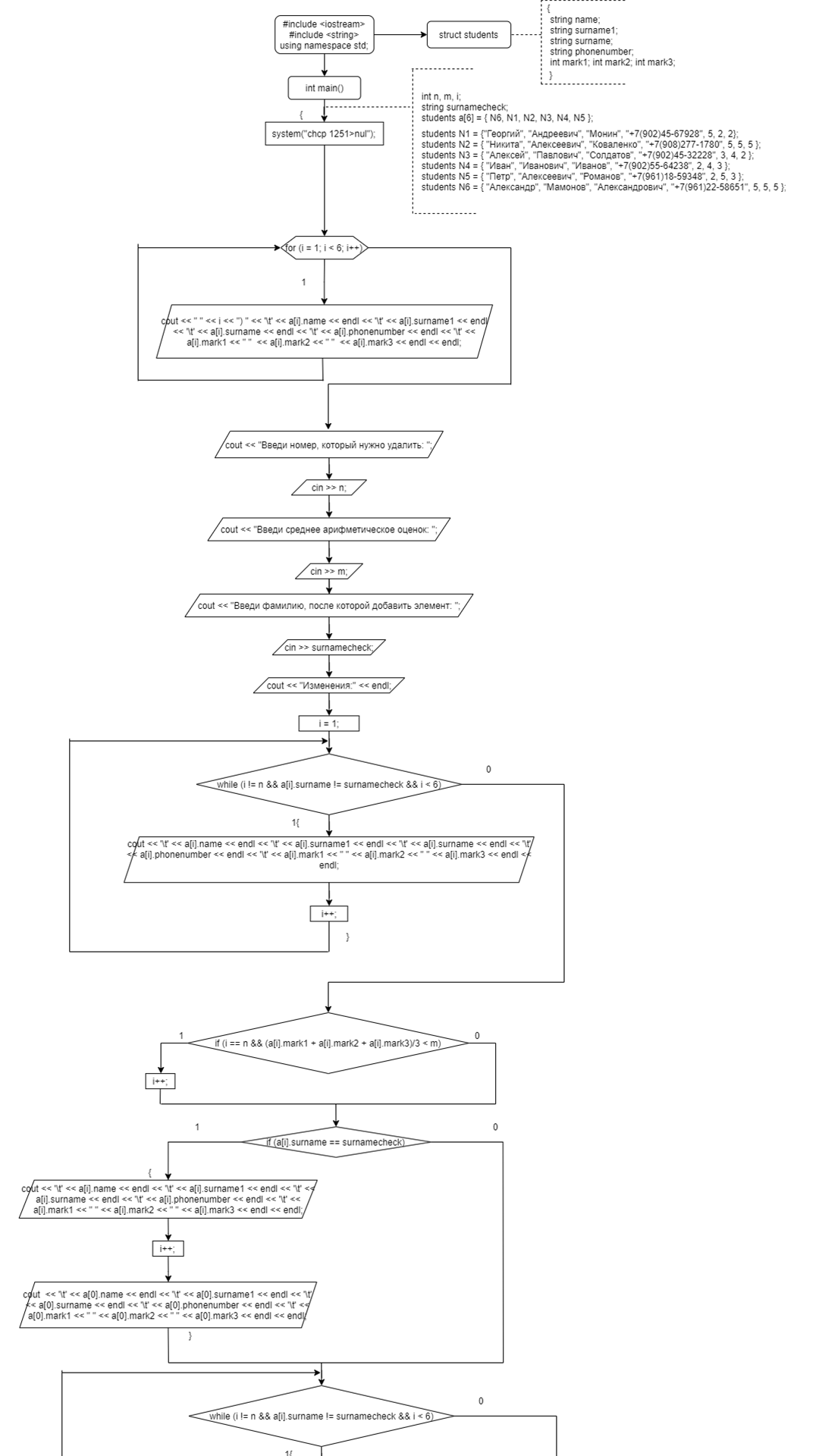
cout << '\t' << a[i].name << endl << '\t' << a[i].surname1 << endl << '\t' << a[i].surname << endl << '\t' << a[i].phonenumber << endl << '\t' << a[i].mark1 << " " << a[i].mark2 << " " << a[i].mark3 << endl << endl;

i++;

}

}

Блок-схема:



Работа кода

