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 : "Перегрузка функций. Функции с переменным числом параметров"

Выполнила работу

Студентка группы РИС-22-1Б

Верхоланцева Екатерина Сергеевна

Проверила

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Г. Пермь-2023

**1. Постановка задачи.**

**Вариант 9:**

Структура "Пациент":

* фамилия, имя, отчество;
* домашний адрес;
* номер медицинской карты;
* номер страхового полиса.

Удалить элемент с заданным номером медицинской карты, добавить 2 элемента в начало файла.

**2. Анализ задачи.**

1. Создать структуру patient с 4-мя полями.

2. Создать функцию del\_el для удаления элемента с номером медицинской карты, совпадающим с введенным, в которую подается массив структур p[], введенный пользователем номер мед.карты и количество k элементов в массиве p. В цикле for сравниваются 2 номера, если есть совпадение, элемент удаляется из массива.

3. Создать функцию vvod\_str для ввода данных каждого элемента.

4. В функции main ввести количество пациентов k, объявить массив типа данных patient p[100].

5. Ввести данные k элементов.

6. Открыть файл Out.txt для записи, ввести в него данные из массива р. Закрыть файл.

7. Ввести номер мед.карты, элемент с которым нужно удалить.

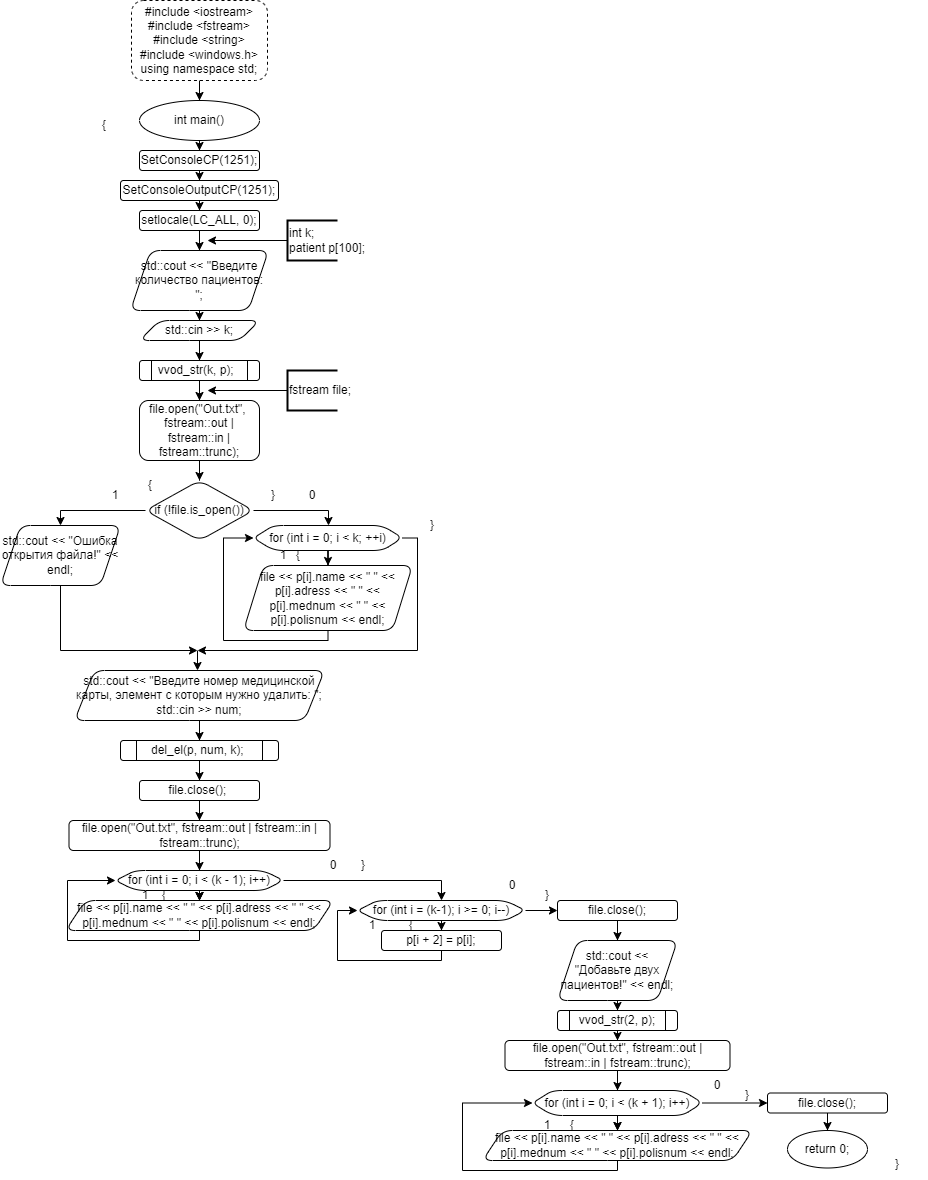
8. Удалить заданный элемент с помощью функции del\_el.

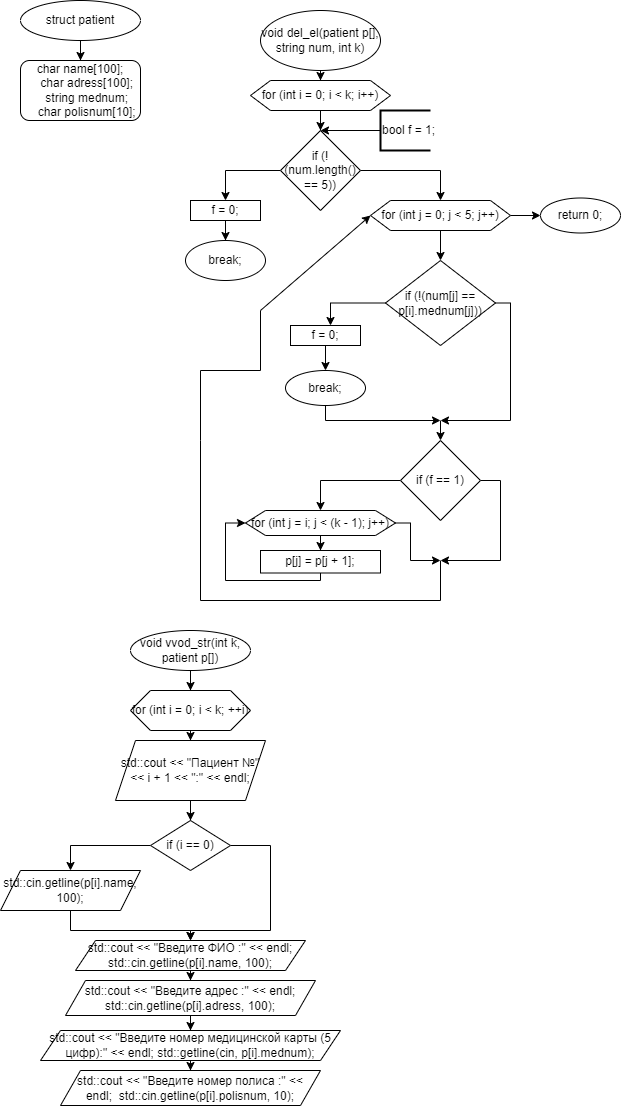
9. Освободить место в начале массива р для двух новых элементов.

10. Ввести данные двух новых пациентов с помощью функции vvod\_str.

11. Открыть файл Out.txt для записи новых данных. Ввести данные в файл. Закрыть файл.

**3. Блок-схема.**





**4. Код.**

#include <iostream>

#include <fstream>

#include <string>

#include <windows.h>

using namespace std;

/\*

Удалить элемент с заданным номером медицинской карты, добавить 2 элемента в начало файла.

\*/

struct patient

{

char name[100];

char adress[100];

string mednum;

char polisnum[10];

};

void del\_el(patient p[], string num, int k)

{

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

bool f = 1;

if (!(num.length() == 5)) {

f = 0;

break;

}

else {

for (int j = 0; j < 5; j++) {

if (!(num[j] == p[i].mednum[j])) {

f = 0;

break;

}

if (f == 1) {

for (int j = i; j < (k - 1); j++) {

p[j] = p[j + 1];

}

}

}

}

}

}

void vvod\_str(int k, patient p[]) {

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

{

std::cout << "Пациент №" << i + 1 << ":" << endl;

if (i == 0)

{

std::cin.getline(p[i].name, 100);

}

std::cout << "Введите ФИО :" << endl; std::cin.getline(p[i].name, 100);

std::cout << "Введите адрес :" << endl; std::cin.getline(p[i].adress, 100);

std::cout << "Введите номер медицинской карты (5 цифр):" << endl; std::getline(cin, p[i].mednum);

std::cout << "Введите номер полиса :" << endl; std::cin.getline(p[i].polisnum, 10);

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, 0);

int k; //кол-во экземпляров структуры

std::cout << "Введите количество пациентов: ";

std::cin >> k;

patient p[100]; // массив структур

vvod\_str(k, p); // ввод данных структуры

fstream file;

file.open("Out.txt", fstream::out | fstream::in | fstream::trunc); // открытие файла

if (!file.is\_open())

{

std::cout << "Ошибка открытия файла!" << endl;

}

else

{

for (int i = 0; i < k; ++i) {// ввод данных в файл

file << p[i].name << " " << p[i].adress << " " << p[i].mednum << " " << p[i].polisnum << endl;

}

}

string num;

std::cout << "Введите номер медицинской карты, элемент с которым нужно удалить: "; std::cin >> num;

std::cout << "Данные из файла до удаления: " << endl;

string s;

file.seekg(0, file.beg);

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

getline(file, s, '\n');

cout << s << endl;

}

del\_el(p, num, k); // удаление элемента с заданным номером мед карты

std::cout << endl;

file.close(); // закрытие файла

file.open("Out.txt", fstream::out | fstream::in | fstream::trunc); // открываем файл для записи без удаленного элемента

for (int i = 0; i < (k - 1); i++) { // ввод данных в файл

file << p[i].name << " " << p[i].adress << " " << p[i].mednum << " " << p[i].polisnum << endl;

}

std::cout << "Данные из файла после удаления:" << endl;

file.seekg(0, file.beg);

/\*for (int i = 0; i < (k - 1); i++)\*/

while (!file.eof()){

getline(file, s, '\n');

cout << s << endl;

}

// --------------------------------------------------------------------------------------------------------------

for (int i = (k-1); i >= 0; i--) { // освобождаем место в начале массива для двух новых элементов

p[i + 2] = p[i];

}

file.close();

std::cout << endl;

std::cout << "Добавьте двух пациентов!" << endl;

vvod\_str(2, p);

file.open("Out.txt", fstream::out | fstream::in | fstream::trunc);

std::cout << "Данные с двумя новыми пациентами:" << endl;

for (int i = 0; i < (k + 1); i++) { // ввод данных в файл

file << p[i].name << " " << p[i].adress << " " << p[i].mednum << " " << p[i].polisnum << endl;

}

file.seekg(0, file.beg);

while (!file.eof()) {

getline(file, s, '\n');

cout << s << endl;

}

file.close();

return 0;

}

**5. Результат программы.**

