**«Пермский национальный исследовательский  
политехнический университет»**

**О Т Ч Ё Т**

**по лабораторной работе №8**

Тема: Структуры данных

Вариант 5

Выполнил работу

студент группы ИВТ-20-2б

Нефедов Л.В.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

Постановка задачи

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и до-бавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура "Человек":

- фамилия, имя, отчество;

- год рождения;

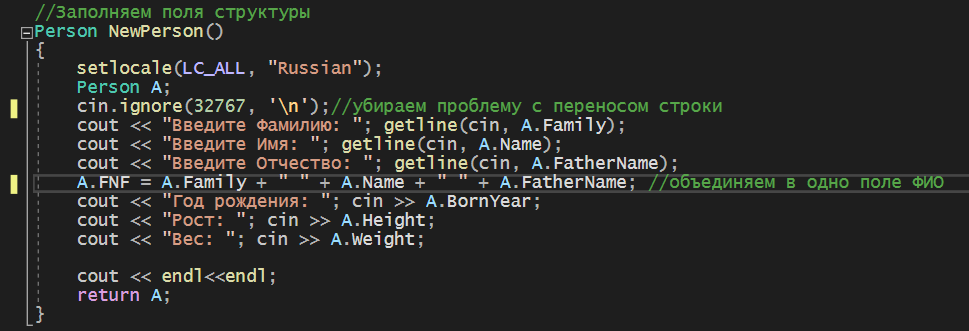
- рост;

- вес.

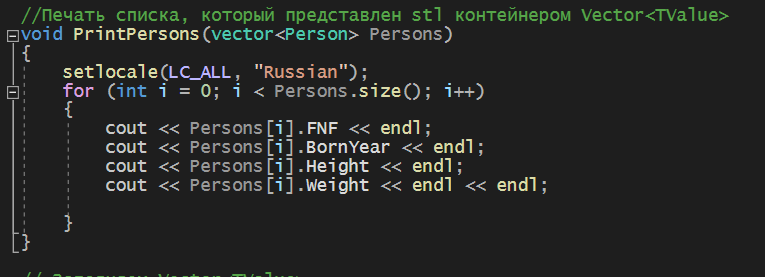
Удалить все элемент с указанным ростом и весом, добавить элемент после элемента с указанной фамилией.

Анализ задачи

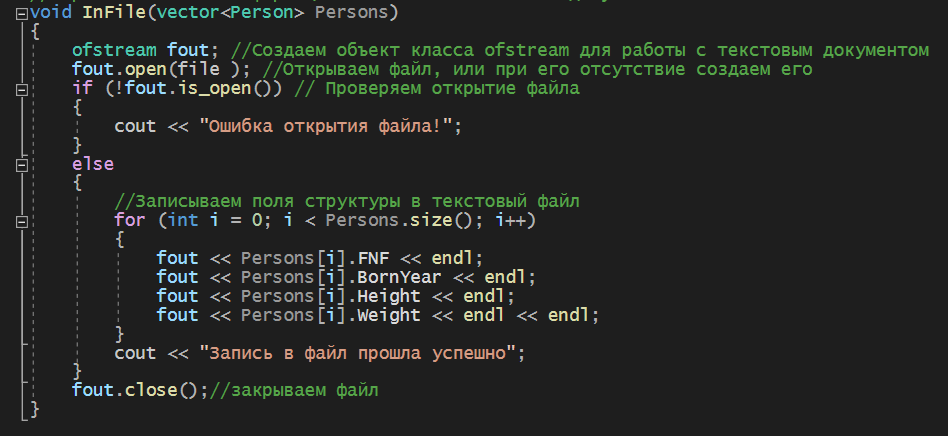
1. Для заполнения полей структуры и чтения с клавиатуры вводимых данных была разработана и применена функция Person



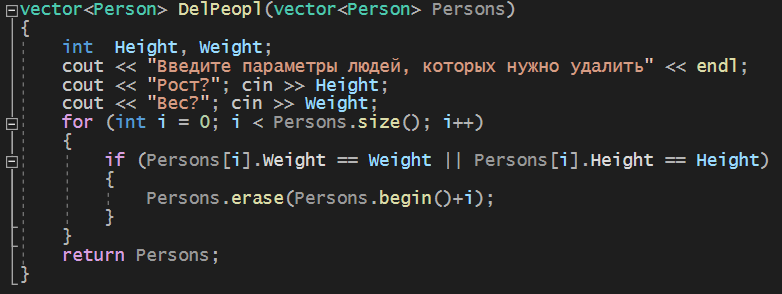
* 1. Печать получившегося списка осуществляется при помощи функции PrintPersons типа void



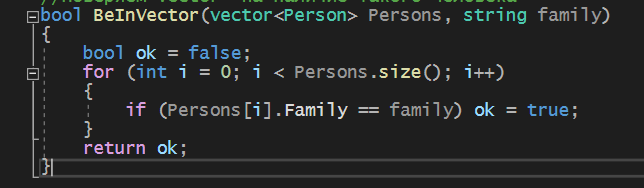
* 1. В функции InFile типа void открываем существующий файл, или же, если он отсутствует, то создаем его. Также идет проверка на открытие файла, а далее идет запись из полей структуры в файл



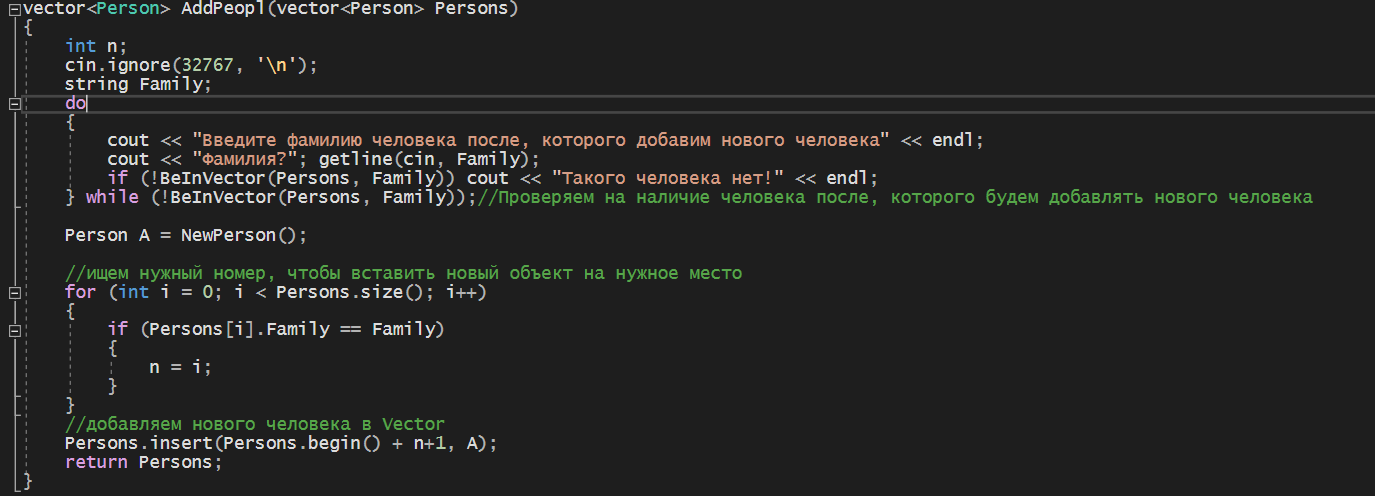
* 1. С помощью структуры данных запрашиваем данные людей, которых нужно удалить, и удаляем их.



* 1. С помощью логической функции проверяем структуру на наличие такого человека.



* 1. Здесь идет добавление людей в список, а также проверка на присутствие человека, после которого будем добавлять новые имена и вставка новых элементов.



1. В программе были использованы такие типы данных, как:
   1. Int для сохранения возраста, веса и года рождения людей;

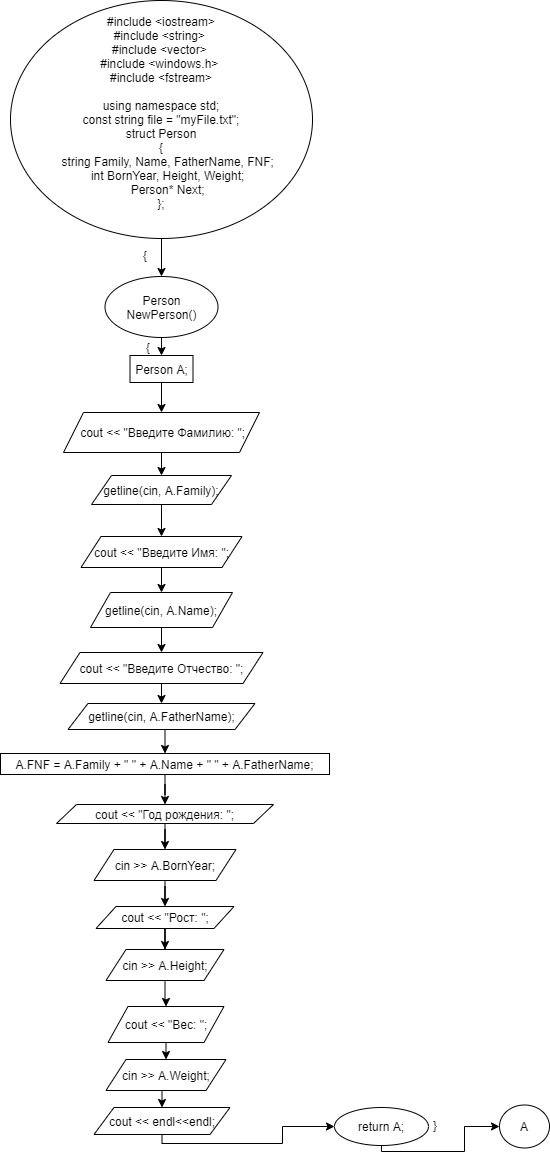


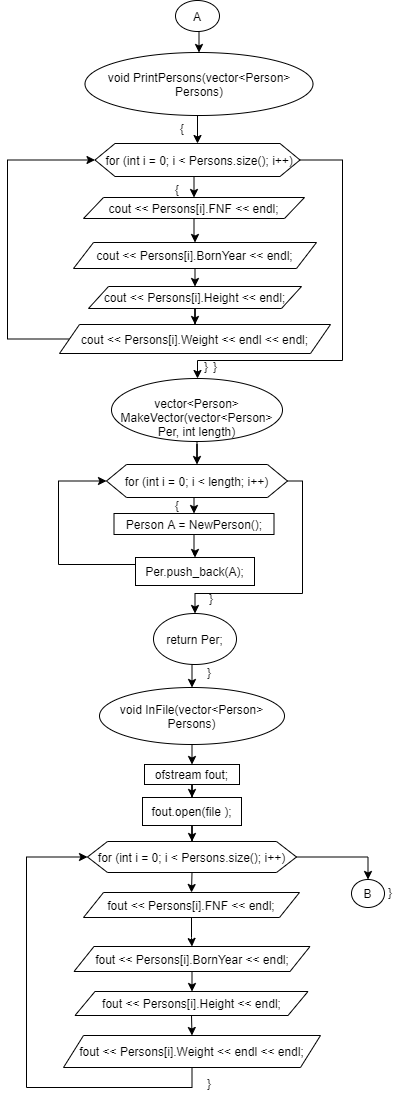
* 1. String для сохранения данных о человеке;

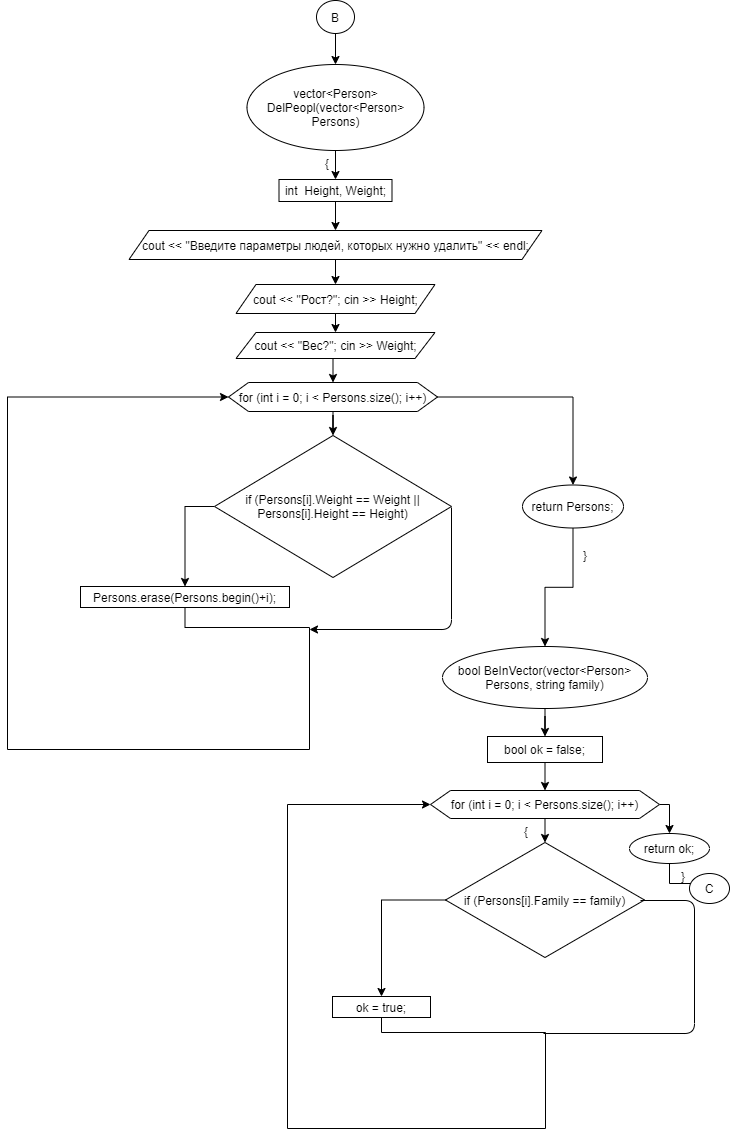


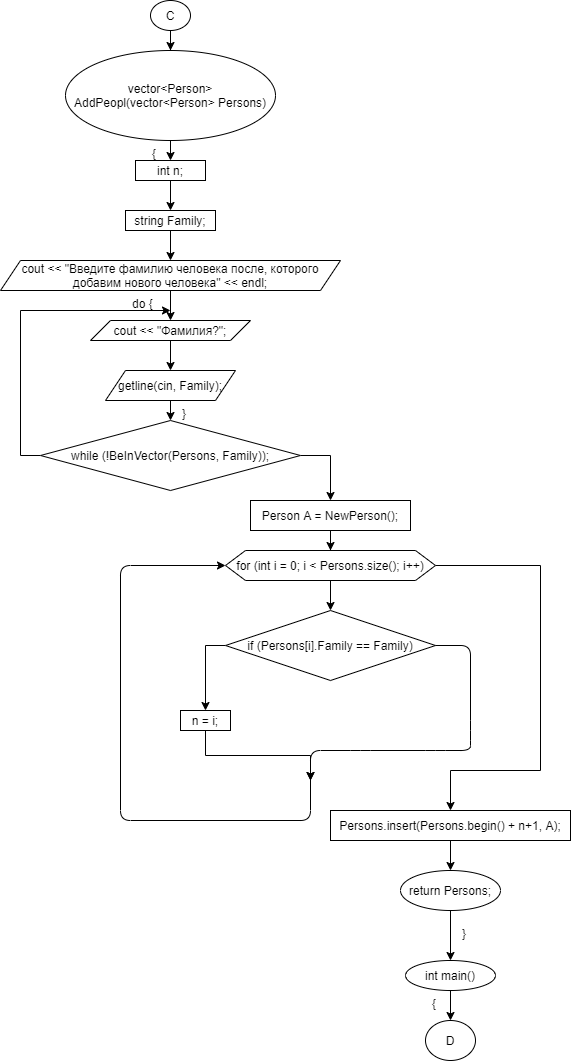
2.3.

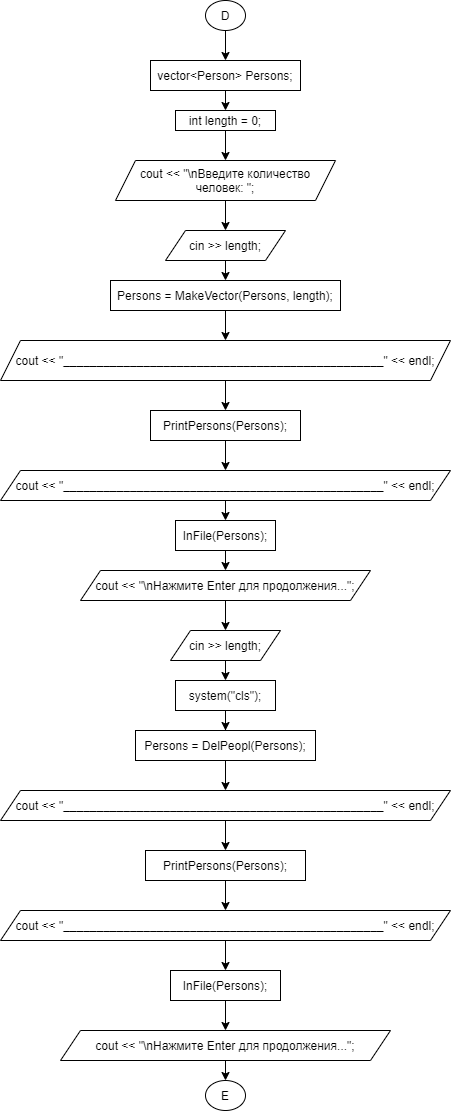
Блок—схема программы

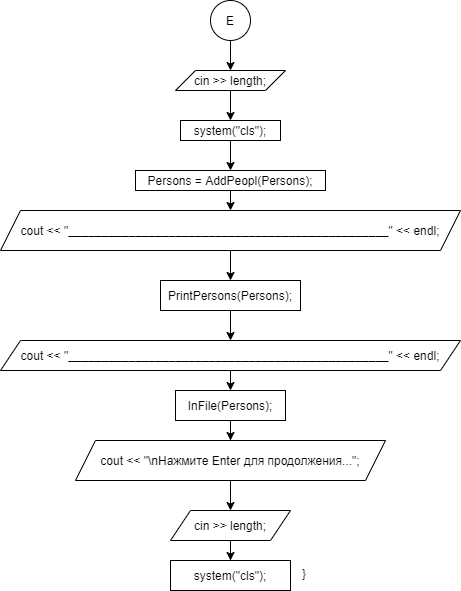












Решение

#include <iostream>

#include <string>

#include <vector>

#include <windows.h>

#include <fstream>

using namespace std;

const string file = "myFile.txt";

struct Person

{

string Family, Name, FatherName, FNF;

int BornYear, Height, Weight;

Person\* Next;

};

Person NewPerson()

{

setlocale(LC\_ALL, "Russian");

Person A;

cin.ignore(32767, '\n');

cout << "Введите Фамилию: "; getline(cin, A.Family);

cout << "Введите Имя: "; getline(cin, A.Name);

cout << "Введите Отчество: "; getline(cin, A.FatherName);

A.FNF = A.Family + " " + A.Name + " " + A.FatherName;

cout << "Год рождения: "; cin >> A.BornYear;

cout << "Рост: "; cin >> A.Height;

cout << "Вес: "; cin >> A.Weight;

cout << endl << endl;

return A;

}

void PrintPersons(vector<Person> Persons)

{

setlocale(LC\_ALL, "Russian");

for (int i = 0; i < Persons.size(); i++)

{

cout << Persons[i].FNF << endl;

cout << Persons[i].BornYear << endl;

cout << Persons[i].Height << endl;

cout << Persons[i].Weight << endl << endl;

}

}

vector<Person> MakeVector(vector<Person> Per, int length)

{

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

{

Person A = NewPerson();

Per.push\_back(A);

}

return Per;

}

void InFile(vector<Person> Persons)

{

ofstream fout;

fout.open(file);

if (!fout.is\_open())

{

cout << "Ошибка открытия файла!";

}

else

{

for (int i = 0; i < Persons.size(); i++)

{

fout << Persons[i].FNF << endl;

fout << Persons[i].BornYear << endl;

fout << Persons[i].Height << endl;

fout << Persons[i].Weight << endl << endl;

}

cout << "Запись в файл прошла успешно";

}

fout.close();

}

vector<Person> DelPeopl(vector<Person> Persons)

{

int Height, Weight;

cout << "Введите параметры людей, которых нужно удалить" << endl;

cout << "Рост?"; cin >> Height;

cout << "Вес?"; cin >> Weight;

for (int i = 0; i < Persons.size(); i++)

{

if (Persons[i].Weight == Weight || Persons[i].Height == Height)

{

Persons.erase(Persons.begin() + i);

}

}

return Persons;

}

bool BeInVector(vector<Person> Persons, string family)

{

bool ok = false;

for (int i = 0; i < Persons.size(); i++)

{

if (Persons[i].Family == family) ok = true;

}

return ok;

}

vector<Person> AddPeopl(vector<Person> Persons)

{

int n;

cin.ignore(32767, '\n');

string Family;

do

{

cout << "Введите фамилию человека после, которого добавим нового человека" << endl;

cout << "Фамилия?"; getline(cin, Family);

if (!BeInVector(Persons, Family)) cout << "Такого человека нет!" << endl;

} while (!BeInVector(Persons, Family));

Person A = NewPerson();

for (int i = 0; i < Persons.size(); i++)

{

if (Persons[i].Family == Family)

{

n = i;

}

}

Persons.insert(Persons.begin() + n + 1, A);

return Persons;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vector<Person> Persons;

setlocale(LC\_ALL, "Russian");

int length = 0;

cout << "\nВведите количество человек: "; cin >> length;

Persons = MakeVector(Persons, length);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

PrintPersons(Persons);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

InFile(Persons);

cout << "\nВведите любое число для продолжения...";

cin >> length;

system("cls");

Persons = DelPeopl(Persons);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

PrintPersons(Persons);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

InFile(Persons);

cout << "\nВведите любое число для продолжения...";

cin >> length;

system("cls");

Persons = AddPeopl(Persons);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

PrintPersons(Persons);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

InFile(Persons);

cout << "\nВведите любое число для продолжения...";

cin >> length;

system("cls");

}

Скриншоты

