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: Информатика

Тема: «Блоковый ввод-вывод»

Вариант №4

Выполнил работу

студент группы РИС-20-1б

Еске Вячеслав Сергеевич

Проверил

Доцент кафедры ИТАС

к.т.н. Полякова О.А.

Пермь, 2021

Цель задачи

Работа с двоичными файлами, организация ввода-вывода структурированной информации и ее хранение на внешних носителях.

Задача

Сформировать двоичный файл из элементов, заданной в вари-анте структуры, распечатать его содержимое, выполнить удаление и добавление 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;

- домашний адрес;

- номер телефона;

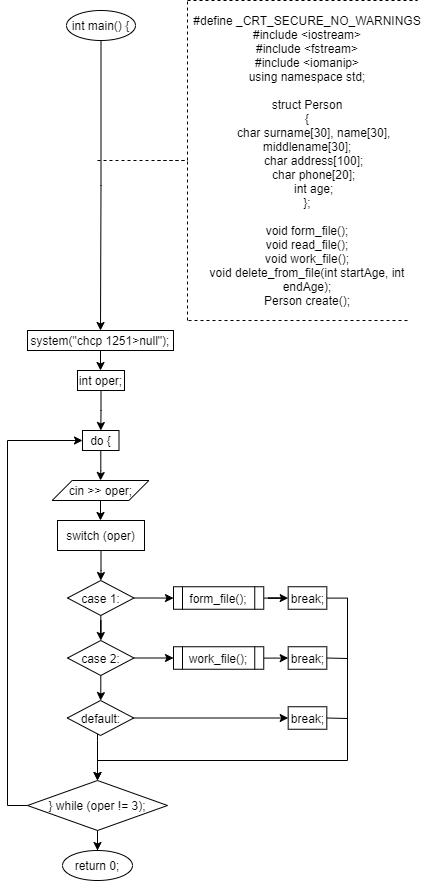
- возраст.

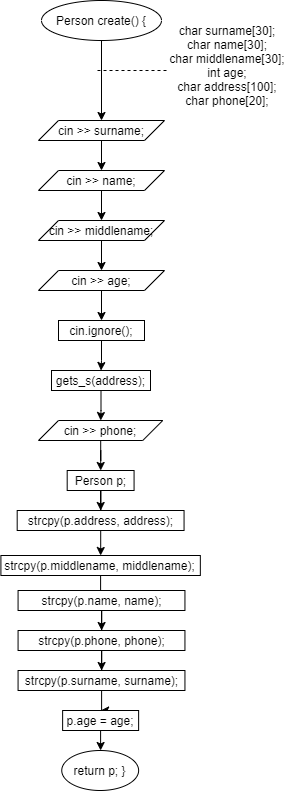
Удалить все элементы с заданным возрастом, добавить элемент после элемента с заданным номером.

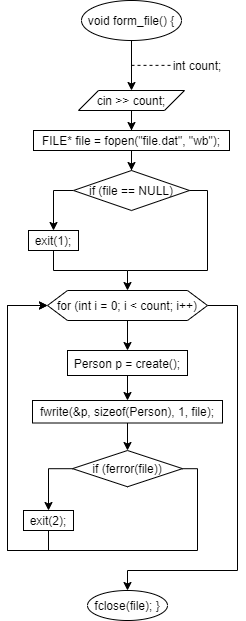
Анализ

1. Определить, какие должны быть выполнены действия:
   1. Поиск по возрасту.
   2. Добавление элемента после элемента с заданным номером.
   3. Для ФИО созданы три поля из массивов типа char, состоящие из 30 символов.
   4. Для адреса создано одно поле из массива типа char, состоящее из 100 символов.
   5. Для номера телефона создано одно поле из массива типа char, состоящее из 20 символов.
   6. Для возраста создано одно поле типа int.
2. Разработать структуру Person:
3. Данные в структуру вводятся через файл.
4. Вывод данных также осуществляется через файл.
5. В функции производится поиск всех элементов, возраст которых не входит в введенный интервал. Такие элементы записываются во временный файл и флажок меняет свое значение. Если после поиска флажок изменил свое значение, то старый файл удаляется, а новый переименовывается.  
   
6. Для добавления элемента в структуру данных вводится номер элемента, после которого необходимо поставить новый элемент. Функция создает новый файл и переносит все элементы до нужной позиции. В нужную позицию вводится новый элемент. После него в файл переносятся остальные элементы, файл переименовывается, а старый файл удаляется.  
   

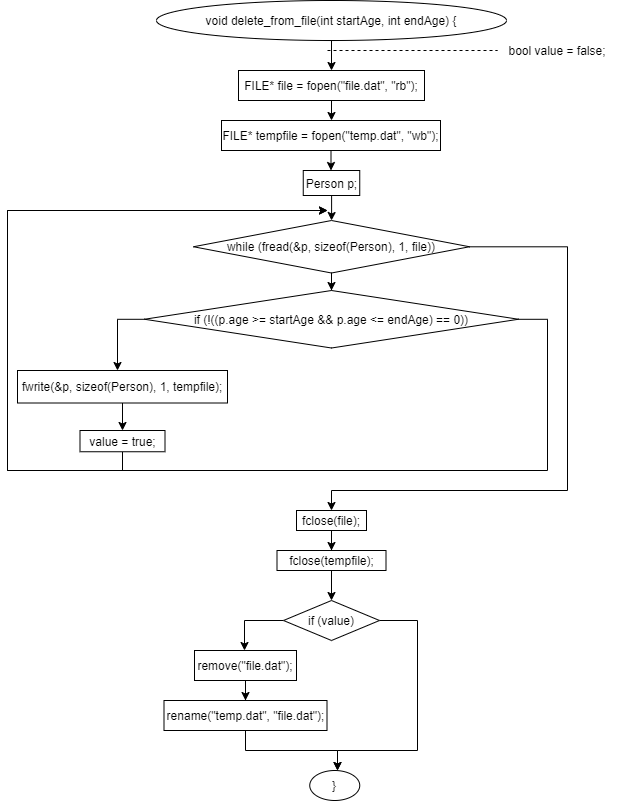
Блок-схема

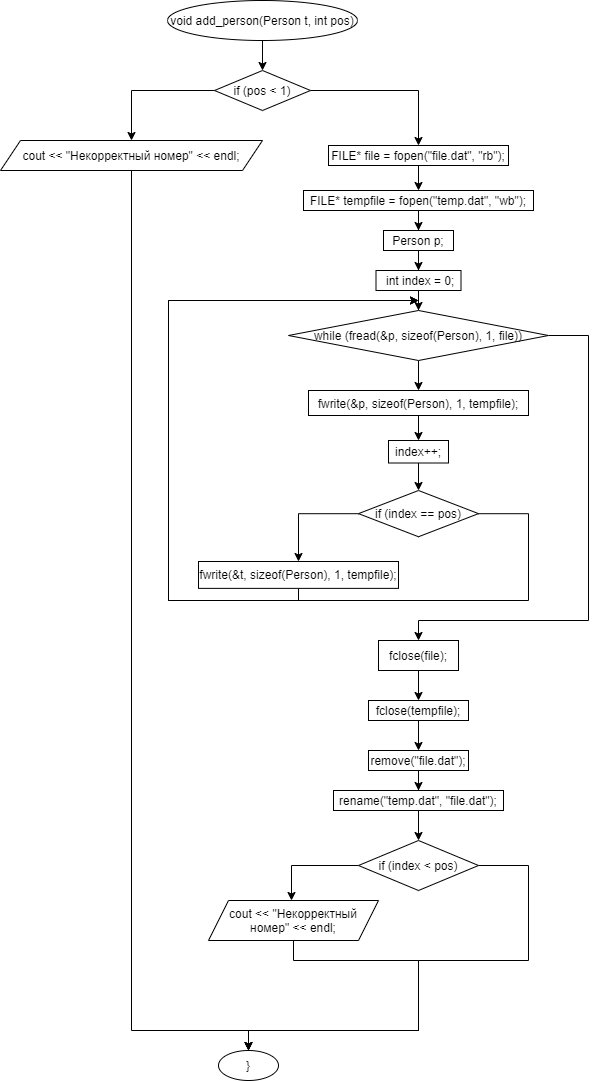


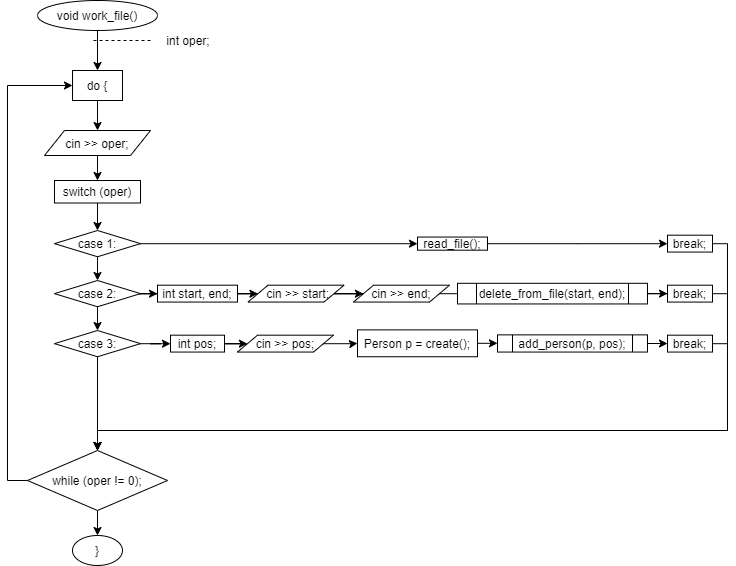












Программный код

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

struct Person

{

char surname[30], name[30], middlename[30];

char address[100];

char phone[20];

int age;

};

void form\_file();

void read\_file();

void work\_file();

void delete\_from\_file(int startAge, int endAge);

Person create();

int main()

{

system("chcp 1251>null");

int oper;

do

{

cout << "1.Сформировать файл" << endl;

cout << "2.Работа с файлом" << endl;

cout << "3.Выход" << endl;

cin >> oper;

switch (oper)

{

case 1:

form\_file();

break;

case 2:

work\_file();

break;

default:

break;

}

} while (oper != 3);

return 0;

}

Person create()

{

cout << "Фамилия: ";

char surname[30];

cin >> surname;

cout << "Имя: ";

char name[30];

cin >> name;

cout << "Отчество: ";

char middlename[30];

cin >> middlename;

cout << "Возраст: ";

int age;

cin >> age;

cout << "Адрес: ";

char address[100];

cin.ignore();

gets\_s(address);

cout << "Телефон: ";

char phone[20];

cin >> phone;

Person p;

strcpy(p.address, address);

strcpy(p.middlename, middlename);

strcpy(p.name, name);

strcpy(p.phone, phone);

strcpy(p.surname, surname);

p.age = age;

return p;

}

void form\_file()

{

cout << "Кол-во людей = ";

int count;

cin >> count;

FILE\* file = fopen("file.dat", "wb");

if (file == NULL)

exit(1);

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

{

Person p = create();

fwrite(&p, sizeof(Person), 1, file);

if (ferror(file))

exit(2);

}

fclose(file);

}

void read\_file()

{

FILE\* file = fopen("file.dat", "rb");

Person p;

cout << setw(15) << "ФИО" << setw(23) << "Возраст" << setw(15) << "Телефон" << setw(20) << "Адрес" << endl;

while (fread(&p, sizeof(Person), 1, file))

{

char fio[30] = "";

strcat(fio, p.surname);

strcat(fio, " ");

strcat(fio, p.name);

strcat(fio, " ");

strcat(fio, p.middlename);

cout << setw(25) << fio << setw(10) << p.age << setw(20) << p.phone << setw(20) << p.address << endl;

}

fclose(file);

}

void delete\_from\_file(int startAge, int endAge)

{

FILE\* file = fopen("file.dat", "rb");

FILE\* tempfile = fopen("temp.dat", "wb");

Person p;

bool value = false;

while (fread(&p, sizeof(Person), 1, file))

{

if (!((p.age >= startAge && p.age <= endAge) == 0))

{

fwrite(&p, sizeof(Person), 1, tempfile);

value = true;

}

}

fclose(file);

fclose(tempfile);

if (value)

{

remove("file.dat");

rename("temp.dat", "file.dat");

}

}

void add\_person(Person t, int pos)

{

if (pos < 1)

cout << "Некорректный номер" << endl;

else

{

FILE\* file = fopen("file.dat", "rb");

FILE\* tempfile = fopen("temp.dat", "wb");

Person p;

int index = 0;

while (fread(&p, sizeof(Person), 1, file))

{

fwrite(&p, sizeof(Person), 1, tempfile);

index++;

if (index == pos)

fwrite(&t, sizeof(Person), 1, tempfile);

}

fclose(file);

fclose(tempfile);

remove("file.dat");

rename("temp.dat", "file.dat");

if (index < pos)

cout << "Некорректный номер" << endl;

}

}

void work\_file()

{

int oper;

do

{

cout << "1.Чтение файла" << endl;

cout << "2.Удаление из файла" << endl;

cout << "3.Добавление в файл" << endl;

cout << "0.Назад" << endl;

cin >> oper;

switch (oper)

{

case 1:

read\_file();

break;

case 2:

{

int start, end;

cout << "Введете начальный возраст: ";

cin >> start;

cout << "Введете конечный возраст: ";

cin >> end;

delete\_from\_file(start, end);

break;

}

case 3:

{

int pos;

cout << "После какого элемента добавить? ";

cin >> pos;

Person p = create();

add\_person(p, pos);

break;

}

}

} while (oper != 0);

}

Выполненная работа

