МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**КАФЕДРА ИИТ**

ОТЧЁТ

по лабораторной работе №11

**«Бинарные и текстовые файлы»**

Выполнил:

студент 1 курса

группы ПО-9

Харитонович Захар Сергеевич

Проверила:

Хацкевич М. В.

Брест 2022

**Цель работы:** Изучить принципы программирования с использованием бинарных файлов. Ознакомиться с основными функциями в Си для работы с бинарными файлами.

**Порядок выполнения работы**

**Текст задания**

В программу разработанную в лабораторной работе 10 добавить чтение и сохранение данных массива структур при помощи бинарных файлов следующим образом:

1. При первом запуске программы должен создаваться бинарный или текстовый файл на выбор пользователя для хранения данных из массива структур.

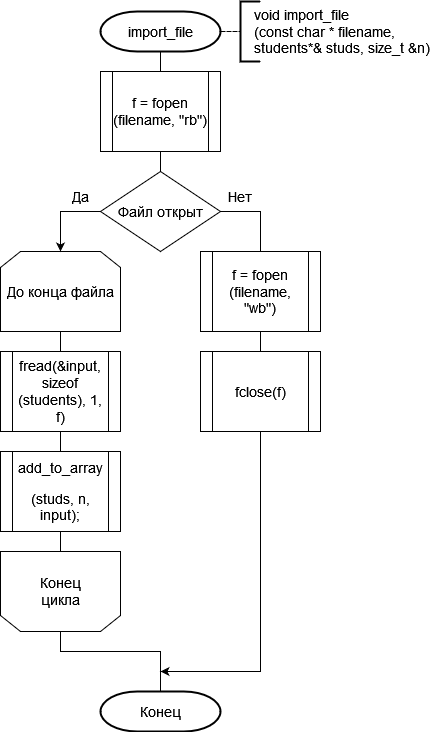
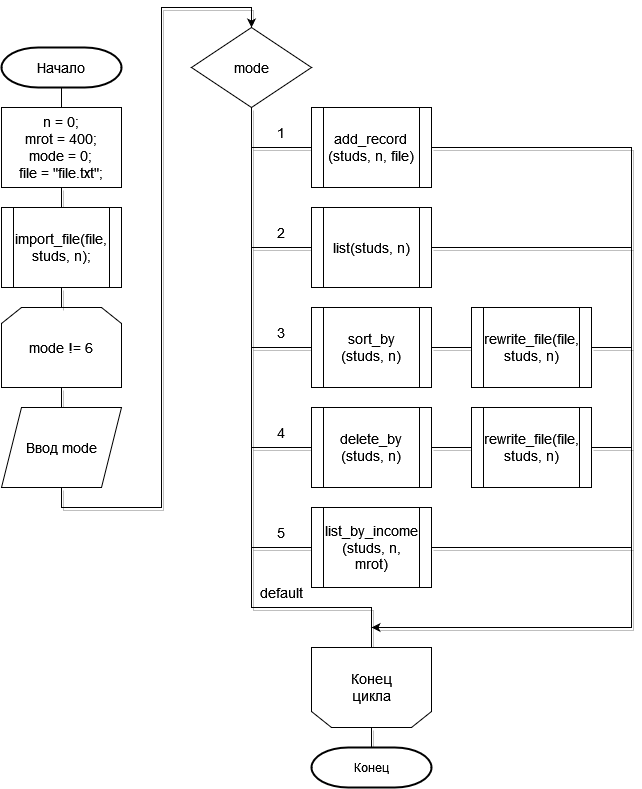
2. При добавлении новой записи в массив структур в файл должна дописываться новая запись, без изменения остальных записей.

3. При повторном запуске программы, если файл уже существует, то информация в массив структур должна читаться из этого файла. Если файл отсутствует, то он должен создаваться (см. Пункт 1).

4. Все изменения (сортировка, изменения полей записи, удаление записи) – сохраняются в файле при помощи полной перезаписи содержимого.

5. Сделать вывод о том, какие преимущества использования конкретного типа файлов (бинарные или текстовые) в решаемой вами задаче.

**Блок-схема**



#include <iostream>

#include <stdlib.h>

#include <string.h>

using namespace std;

struct students {

char surname[256];

char name[256];

char patronymic[256];

enum groups {po8, po9, ii21, ii22, unknown} grp;

double avrMark;

int income;

short bit : 4;

union {

int integer;

double real;

};

};

void display\_record(students);

void add\_record(students \*&, size\_t &, const char \*);

void add\_to\_array(students \*&, size\_t &, students);

void list(students \*, size\_t);

void sort\_by(students \*, size\_t);

void delete\_by(students \*&, size\_t &);

void list\_by\_income(students \*, size\_t, int);

void import\_file(const char \*, students \*&, size\_t &);

void rewrite\_file(const char \*, students \*, size\_t);

int main() {

students \* studs = new students[1];

size\_t n = 0;

char file[] = "file.txt";

import\_file(file, studs, n);

int mrot = 400;

cout << "Welcome to students database.\n";

short mode = 0;

while (mode != 6) {

cout << "Choose mode:\n1 - add a record. 2 - list records. 3 - sort by.\n4 - delete by. 5 - list records with small income. 6 - exit.\n> ";

cin >> mode;

switch (mode) {

case 1:

add\_record(studs, n, file);

break;

case 2:

list(studs, n);

break;

case 3:

sort\_by(studs, n);

rewrite\_file(file, studs, n);

break;

case 4:

delete\_by(studs, n);

rewrite\_file(file, studs, n);

break;

case 5:

list\_by\_income(studs, n, mrot);

case 6:

cout << "Exit\n";

break;

default:

cout << "Unknown.\n";

break;

}

}

delete[] studs;

return 0;

}

void import\_file(const char \* filename, students \*& studs, size\_t &n) {

FILE \*f;

if ((f = fopen(filename, "rb")) == NULL) {

f = fopen(filename, "wb");

fclose(f);

return;

}

for(;;) {

students input;

if (fread(&input, sizeof(students), 1, f) != 1) break;

add\_to\_array(studs, n, input);

}

fclose(f);

}

void rewrite\_file(const char \* filename, students \* studs, size\_t n) {

FILE \*f = fopen(filename, "wb");

for (size\_t i = 0; i < n; i++) fwrite(&studs[i], sizeof(students), 1, f);

fclose(f);

}

void add\_record(students \*&studs, size\_t &n, const char \* filename) {

cout << "Add a record. Fields:\nSurname. Name. Patronymic. Group. Average mark. Income per family member.\n";

char group[256];

students current;

cin >> current.surname >> current.name >> current.patronymic >> group >> current.avrMark >> current.income;

if (!strcmp(group, "po-8") || !strcmp(group, "PO-8")) {

current.grp = students::po8;

} else if (!strcmp(group, "po-9") || !strcmp(group, "PO-9")) {

current.grp = students::po9;

} else if (!strcmp(group, "ii-21") || !strcmp(group, "II-21")) {

current.grp = students::ii21;

} else if (!strcmp(group, "ii-22") || !strcmp(group, "II-22")) {

current.grp = students::ii22;

} else {

current.grp = students::unknown;

}

add\_to\_array(studs, n, current);

FILE \*f = fopen(filename, "ab");

fwrite(&current, sizeof(students), 1, f);

fclose(f);

}

**Вывод:** был изучен синтаксис и принципы работы с бинарными файлами. Реализована программа с использованием бинарных файлов.