Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №8**

Дисциплина: «Информатика»

Тема: Структуры данных

Вариант 1

Выполнил:

Студент группы РИС-20-1б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

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

**Постановка задачи**

(Вариант 1)

Создание структуры "Абитуриент":

- фамилия, имя, отчество;

- год рождения;

- оценки вступительных экзаменов (3);

- средний балл аттестата.

Удалить элемент с указанным номером, добавить элемент после элемента с указанной фамилией.

**Анализ задачи**

1. Определить какие операции должны быть выполнены по заданию:

· Поиск по номеру элемента.

· Добавление элемента после элемента с заданной фамилией.

· Для даты рождения создано поле типа string.

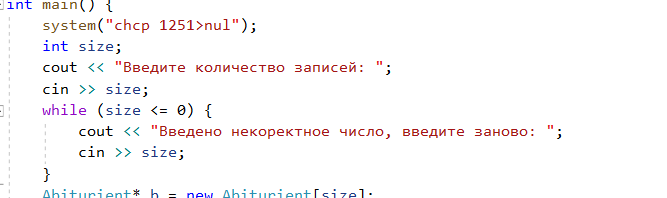
· Для ФИО созданы три поля типа string.

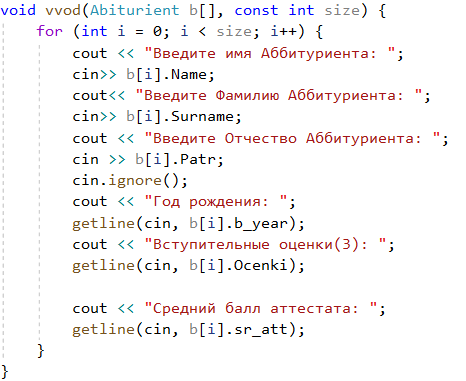
· Для оценок вступительных экзаменов создано поле типа string.

· Для среднего балла аттестата создано поле типа int.

· Важный момент:

2. Разработать структуру Abiturient и объявить её: Разработка структуры проводится без ограничения количества элементов :



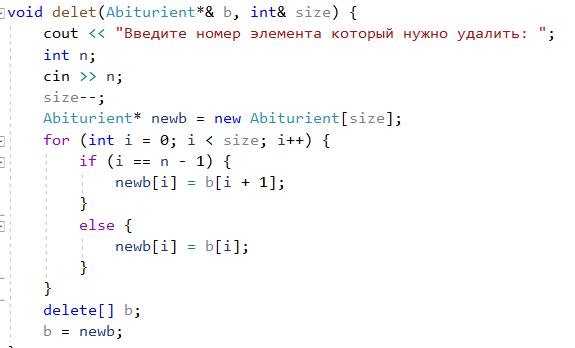
……

3. Данные в структуру введены через цикл while

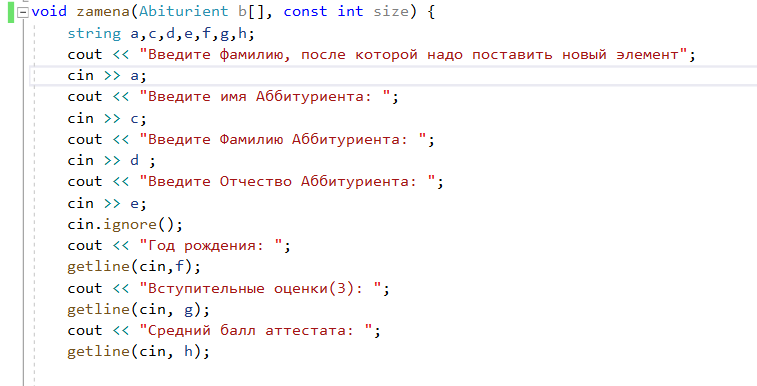
4. Вывод данных из разработанной структуры .

5. Удаляется элемент структуры с совпавшим номером множества .

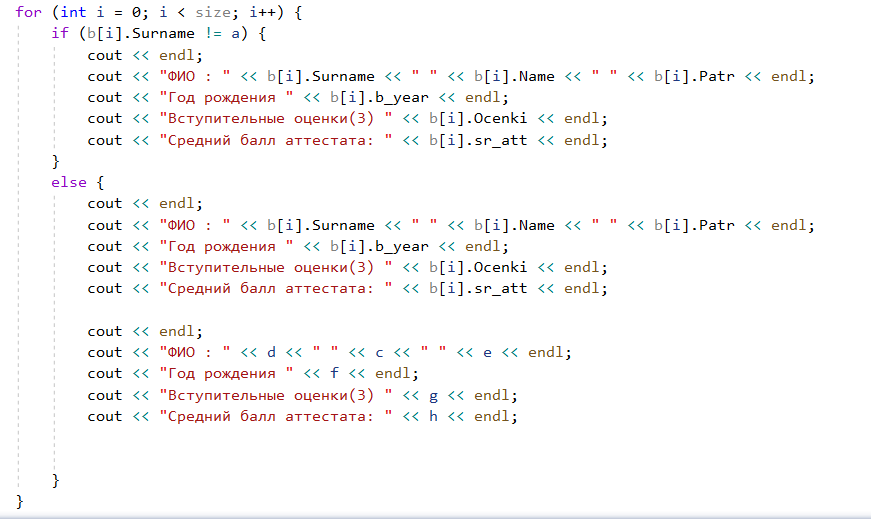
Для реализации иллюзии удаления элементов структуры с нулевыми полями даты рождения используется дополнительная структура newb[]



7. Для добавления элемента в структуру данных вводится фамилия, перед которой необходимо поставить новый элемент, а также под части новой структуры

****

8. Для вставки нового элемента в найденное место структуры(см. пункт 7) реализован цикл for с выводом всех элементов структур до заданной фамилии, с дальнейшим выводом нового элемента структуры и добавлением замыкающих структур:

****

**Код**

#include <iostream>

#include <stdio.h>

#include <string>

using namespace std;

struct Abiturient {

string Name;

string Surname;

string Patr;

string b\_year;

string Ocenki;

string sr\_att;

};

void vvod(Abiturient b[], const int size) {

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

cout << "Введите имя Аббитуриента: ";

cin>> b[i].Name;

cout<< "Введите Фамилию Аббитуриента: ";

cin>> b[i].Surname;

cout << "Введите Отчество Аббитуриента: ";

cin >> b[i].Patr;

cin.ignore();

cout << "Год рождения: ";

getline(cin, b[i].b\_year);

cout << "Вступительные оценки(3): ";

getline(cin, b[i].Ocenki);

cout << "Средний балл аттестата: ";

getline(cin, b[i].sr\_att);

}

}

void vivod(Abiturient b[], const int size) {

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

cout << endl;

cout << "ФИО : " << b[i].Surname << " " << b[i].Name << " " << b[i].Patr << endl;

cout << "Год рождения " <<b[i].b\_year << endl;

cout <<"Вступительные оценки(3) " <<b[i].Ocenki<< endl;

cout << "Средний балл аттестата: " <<b[i].sr\_att << endl;

}

}

void delet(Abiturient\*& b, int& size) {

cout << "Введите номер элемента который нужно удалить: ";

int n;

cin >> n;

size--;

Abiturient\* newb = new Abiturient[size];

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

if (i == n - 1) {

newb[i] = b[i + 1];

}

else {

newb[i] = b[i];

}

}

delete[] b;

b = newb;

}

void zamena(Abiturient b[], const int size) {

string a,c,d,e,f,g,h;

cout << "Введите фамилию, после которой надо поставить новый элемент";

cin >> a;

cout << "Введите имя Аббитуриента: ";

cin >> c;

cout << "Введите Фамилию Аббитуриента: ";

cin >> d ;

cout << "Введите Отчество Аббитуриента: ";

cin >> e;

cin.ignore();

cout << "Год рождения: ";

getline(cin,f);

cout << "Вступительные оценки(3): ";

getline(cin, g);

cout << "Средний балл аттестата: ";

getline(cin, h);

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

if (b[i].Surname != a) {

cout << endl;

cout << "ФИО : " << b[i].Surname << " " << b[i].Name << " " << b[i].Patr << endl;

cout << "Год рождения " << b[i].b\_year << endl;

cout << "Вступительные оценки(3) " << b[i].Ocenki << endl;

cout << "Средний балл аттестата: " << b[i].sr\_att << endl;

}

else {

cout << endl;

cout << "ФИО : " << b[i].Surname << " " << b[i].Name << " " << b[i].Patr << endl;

cout << "Год рождения " << b[i].b\_year << endl;

cout << "Вступительные оценки(3) " << b[i].Ocenki << endl;

cout << "Средний балл аттестата: " << b[i].sr\_att << endl;

cout << endl;

cout << "ФИО : " << d << " " << c << " " << e << endl;

cout << "Год рождения " << f << endl;

cout << "Вступительные оценки(3) " << g << endl;

cout << "Средний балл аттестата: " << h << endl;

}

}

}

int main() {

system("chcp 1251>nul");

int size;

cout << "Введите количество записей: ";

cin >> size;

while (size <= 0) {

cout << "Введено некоректное число, введите заново: ";

cin >> size;

}

Abiturient\* b = new Abiturient[size];

vvod(b, size);

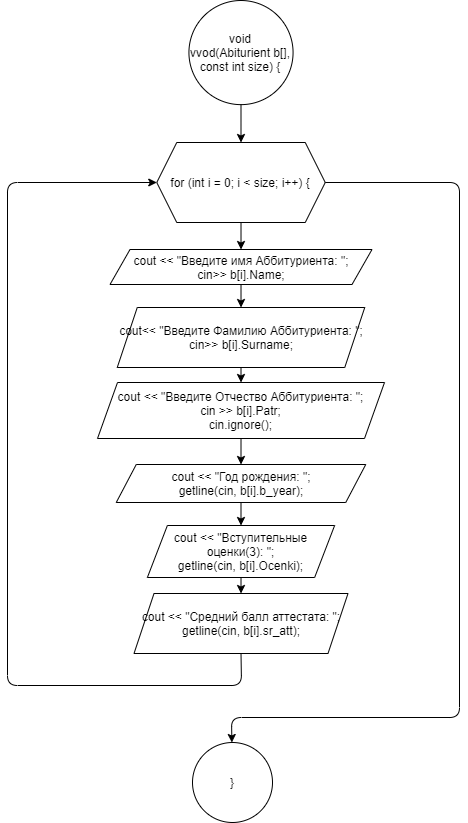
vivod(b, size);

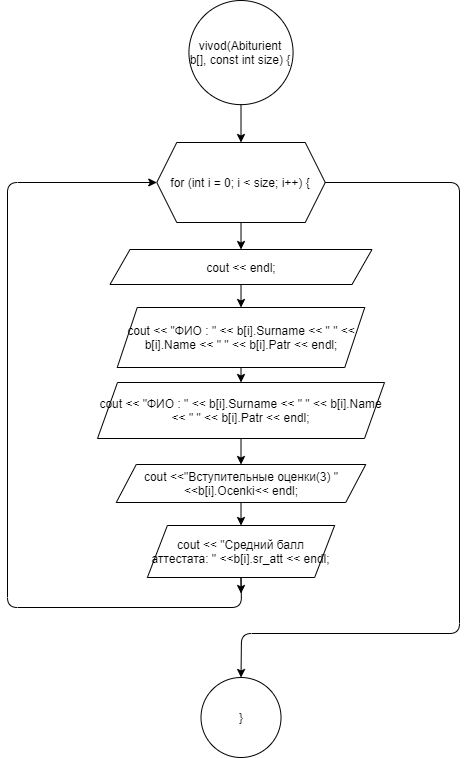
delet(b, size);

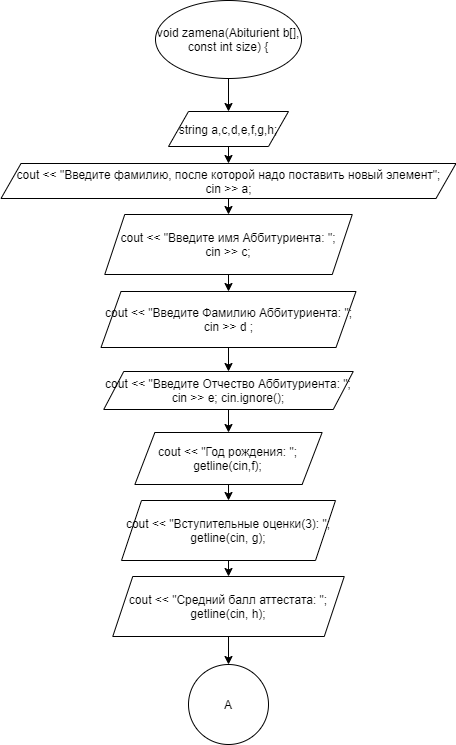
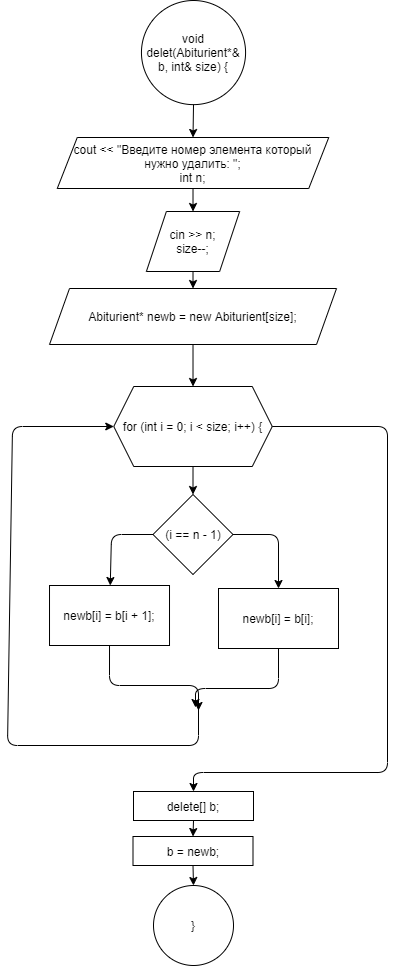
vivod(b, size);

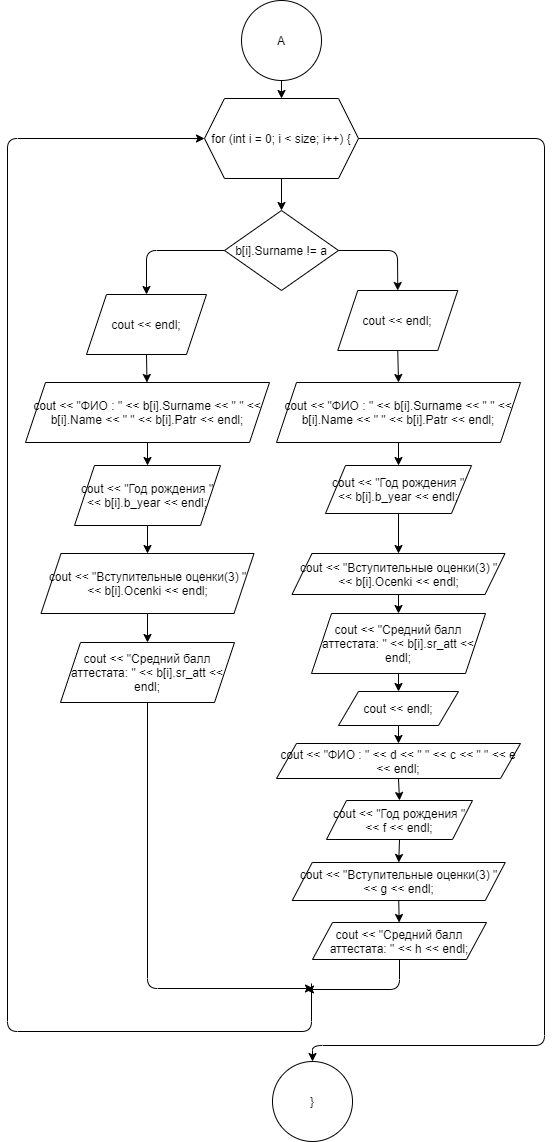
zamena(b, size);

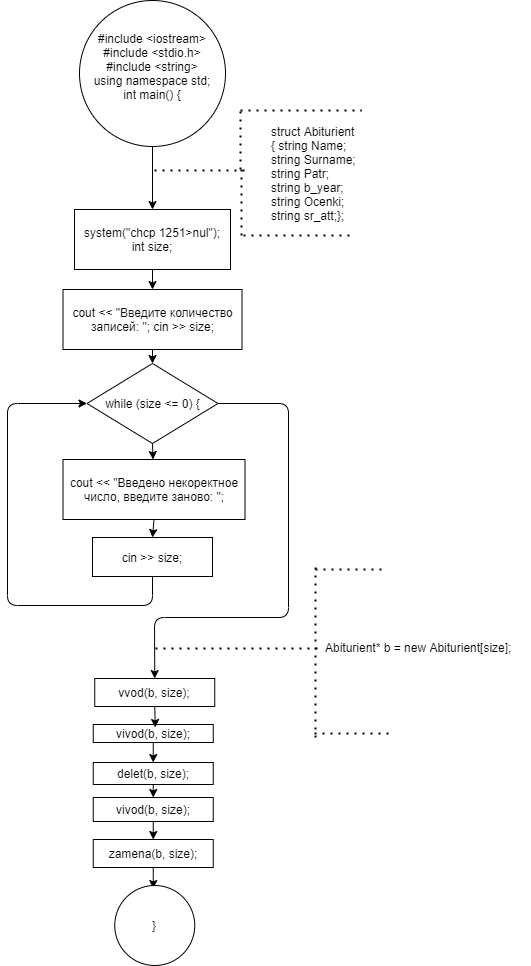
}

****

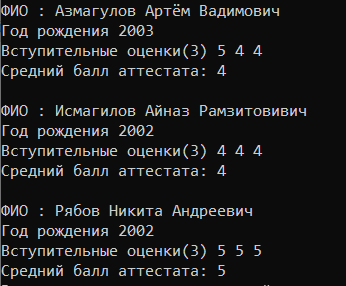
****

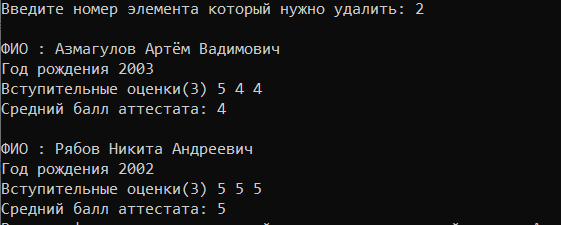
****

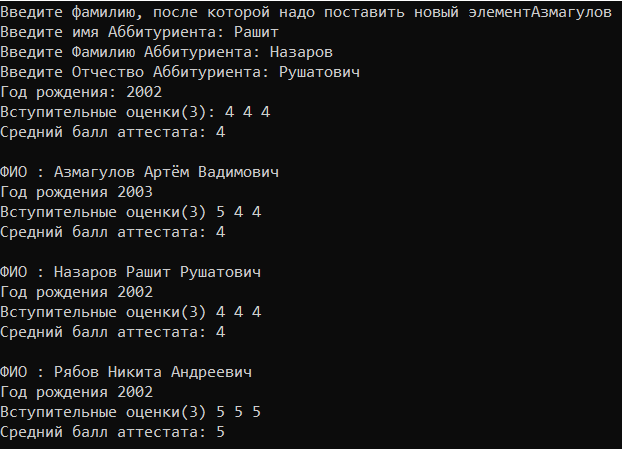
****

****

**Скриншоты результата работы**

****

****

****