Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №10

По дисциплине: «Основы алгоритмизации и программирования»

**Тема:** **«Структуры, перечисления, объединения»**

Вариант №11

Выполнил:

Студент 1-го курса

Группы ПО-7

Лобан К.Ю.

Проверила:

Войцехович О. Ю.

Брест 2021

**Цель работы:** Изучить синтаксис и правила работы со структурами. Реализовать программу с применением структур, перечислений и объединений.

**Задание**

Создать тип структуры согласно варианту, организовать поля этой структуры так, чтобы они содержали объединение, перечисление (можно добавить дополнительные поля) и битовое поле. Создать массив структур, содержащий информацию согласно варианту индивидуального задания. Реализовать работу с массивом структур через меню: ввод данных в массив, вывод содержимого массива на экран, сортировка по одному полю, удаления записи по заданному значению поля, выборка записей согласно индивидуального задания

**Вариант 11**

Ведомость абитуриентов содержит ФИО, адрес, оценки по трем предметам. Определить средний балл абитуриентов, проживающих в городе Минске.

**Текст программы:**

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

struct Enrollee

{

string name;

union

{

char city[50];

bool nonresident : 2;

} adress;

enum

{

Math,

English,

Physic

} subjects;

float marks[3];

};

int main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int choice, n, kol=0;

cout << "Введите количество абитуриентов: ";

cin >> n;

Enrollee \*en = new Enrollee[n];

string name;

char city[50];

float math, eng, phys, sum=0;

bool found=false;

do

{

cout << endl << "Меню: " << endl;

cout << "1-Ввод данных;" << endl;

cout << "2-Вывод данных;" << endl;

cout << "3-Сортировка по ФИО;" << endl;

cout << "4-Удаление по заданному значению поля;" << endl;

cout << "5-Средний балл абитуриентов из Минска;" << endl;

cout << "6-Выход." << endl;

cout << endl << "Ваш выбор: ";

cin >> choice;

switch (choice)

{

case 1:

{

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

{

cin.ignore(cin.rdbuf()->in\_avail());

cout << "Введите ФИО: ";

getline(cin, en[i].name);

cout << "Введите город: ";

cin.ignore(cin.rdbuf()->in\_avail());

cin.getline(en[i].adress.city,50);

if (en[i].adress.city != "Брест")

{

en[i].adress.nonresident = 1;

}

else

{

en[i].adress.nonresident = 0;

}

cout << "Введите оценки по предметам: " << endl;

cout << "Математика: ";

cin >> en[i].marks[en->Math];

cout << "Английский: ";

cin >> en[i].marks[en->English];

cout << "Физика: ";

cin >> en[i].marks[en->Physic];

cin.ignore(cin.rdbuf()->in\_avail());

}

}

break;

case 2:

{

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

{

cout << i+1 <<". ФИО: " << en[i].name << endl;

cout << "Город: " << en[i].adress.city << endl;

cout << "Оценки: " << endl;

cout << "Математика - " << en[i].marks[en->Math];

cout << ", Английский - " << en[i].marks[en->English];

cout << ", Физика - " << en[i].marks[en->Physic] << endl << endl;

}

}

break;

case 3:

{

for (int i = n-1; i >= 0; i--)

{

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

{

if (en[j].name > en[j + 1].name)

{

Enrollee temp = en[j];

\*(en + j) = \*(en + j + 1);

\*(en + j + 1) = temp;

}

}

}

}

break;

case 4:

{

char c;

cout << "По какому полю произвести удаление? (n-ФИО, c-город, m-оценка по математике, e-оценка по английскому, p-оценка по физике): ";

cin >> c;

switch (c)

{

case 'n':

{

cin.ignore(cin.rdbuf()->in\_avail());

cout << "Введите ФИО: ";

getline(cin, name);

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

{

if (name == en[i].name)

{

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

{

en[j] = en[j + 1];

}

n--;

}

}

}

break;

case 'c':

{

cin.ignore(cin.rdbuf()->in\_avail());

cout << "Введите город: ";

cin.getline(city, 50);

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

{

if (city == en[i].adress.city)

{

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

{

en[j] = en[j + 1];

}

n--;

}

}

}

break;

case 'm':

{

cout << "Введите оценку по математике: ";

cin>>math;

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

{

if (math == en[i].marks[en->Math])

{

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

{

en[j] = en[j + 1];

}

n--;

}

}

}

break;

case 'e':

{

cout << "Введите оценку по английскому: ";

cin >> eng;

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

{

if (eng == en[i].marks[en->English])

{

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

{

en[j] = en[j + 1];

}

n--;

}

}

}

break;

case 'p':

{

cout << "Введите оценку по физике: ";

cin >> phys;

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

{

if (phys == en[i].marks[en->Physic])

{

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

{

en[j] = en[j + 1];

}

n--;

}

}

}

break;

}

}

break;

case 5:

{

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

{

if (en[i].adress.city == "Минск")

{

found = true;

kol++;

sum += en[i].marks[en->Math] + en[i].marks[en->English] + en[i].marks[en->Physic];

}

}

if (found)

{

cout << "Средний балл минских абитуриентов: " << sum / (3 \* kol) << endl;

}

else

{

cout << "Ничего не найдено!" << endl;

}

}

break;

case 6:

{

cout << "Выход из программы..." << endl;

}

break;

default:

{

cout << "Неверный ввод!" << endl;

}

}

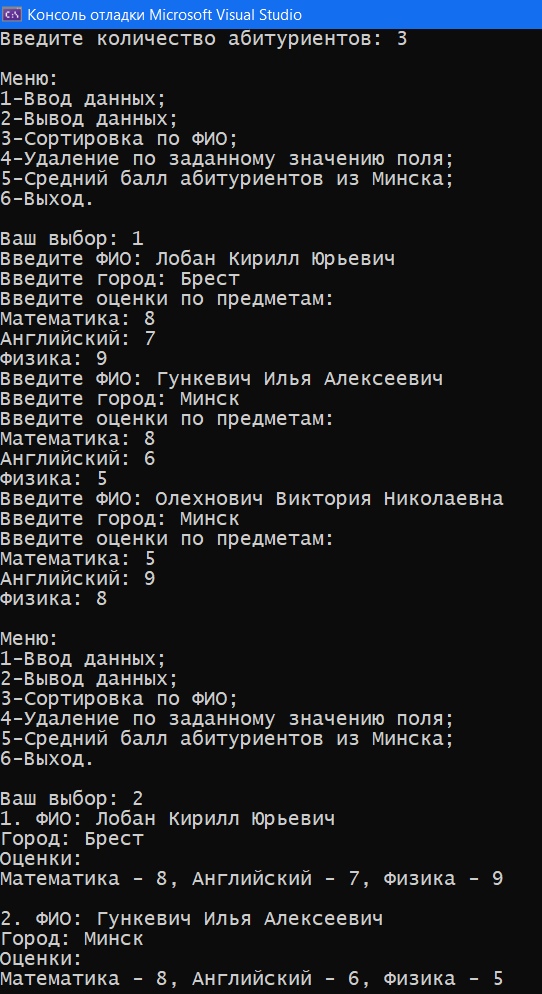
} while (choice!=6);

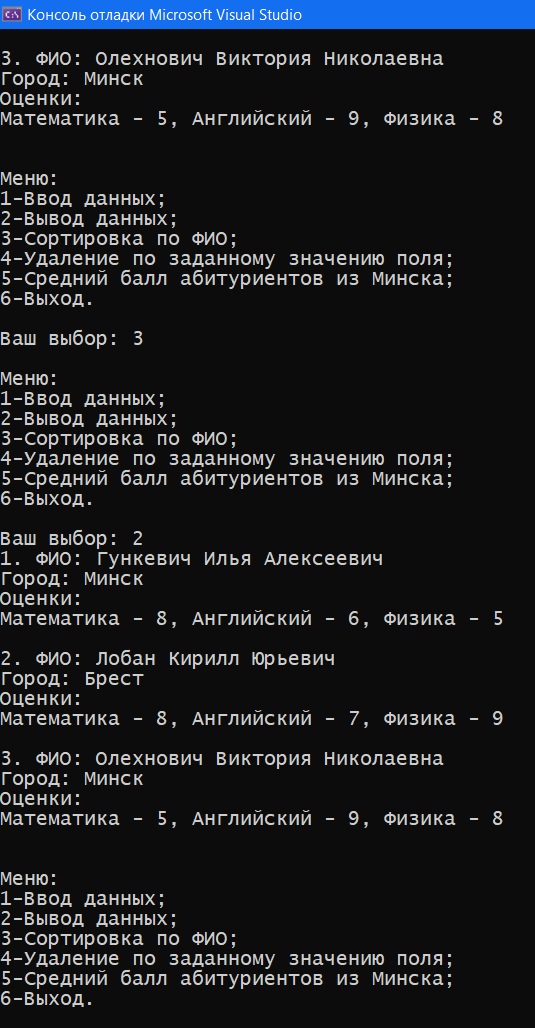
delete[]en;

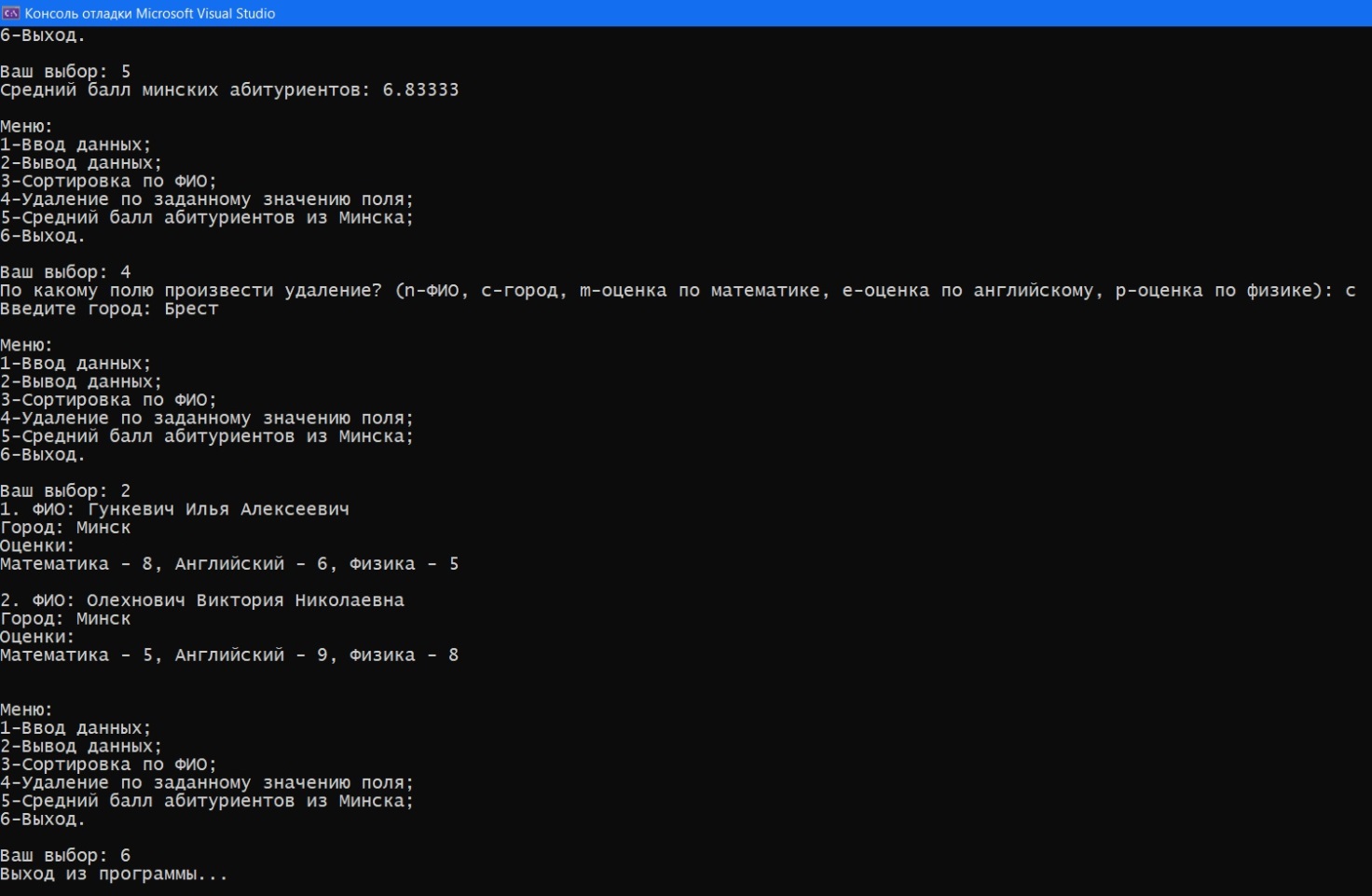
return 0;

}

**Результаты:**







**Вывод:** я изучил синтаксис и правила работы со структурами. Реализовал программу с применением структур, перечислений и объединений.