МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №5**

**з дисципліни «Основи програмування»**

**на тему: *«*Структурний тип даних в мові С++.»**

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: доц. Нежуміра О.І.

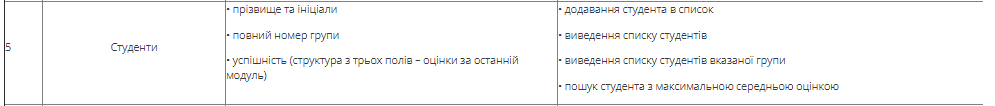
Дніпро, 2020

**Тема.** Структурний тип даних в мові С++.

**Мета.** Засвоїти поняття складеного типу даних. Отримати практичні навички роботи з інтегрованими типами даних – структурами і масивами структур – мови С++.

Варіант 5

Розробити програму для роботи з даними для обраної предметної області.



**Вимоги до програми:**

- вхідні дані вводяться з клавіатури;   
- перевірка вхідних даних на відповідність діапазону значень і некоректні символи;  
- введені дані зберігаються в масиві структур (масив складається з 20 елементів);  
- символьні поля структури оголошуються як масив символів;  
- інформація, яка зберігається у символьному полі структури, може складатися з декількох слів;  
- три перши елементи масиву необхідно ініціалізувати при оголошенні масиву;  
- управління виконанням програми здійснюється на основі текстового меню користувача;  
- дії програми реалізувати у вигляді функцій;  
- програму розробити як багатофайловий проект (в першому файлі заголовків розмістити опис структури елементів масиву, у другому - прототипи функцій програми, в файлі реалізації - опис функцій програми, функцію main() оформити окремим файлом);  
- результати роботи програми виводяться на екран, списки виводяться у вигляді таблиць.

**1. Зовнішні специфікації**

Вхідні дані:  
1) П.І.Б. студента.  
2) Група студента.  
3) Оцінки за модуль.

Вимоги до формату вхідних даних наведені у табл.1

**Формат вхідних даних**

Таблиця 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | П.І.Б. студента. | surname | Рядок символів | «Кулик С. В.» |
| 2 | Группа студента. | group | Натуральне число | 911 912 |
| 3 | Оцінки за модуль. | raiting | Натуральні числа більші за 0 та менші від 100 | 60 85 |

Вихідні дані:  
1) П.І.Б. студента.  
2) Група студента.  
3) Оцінки за модуль.

Вимоги до вихідних даних наведені в таблиці 2

**Формат вихідних даних**

Таблиця 2

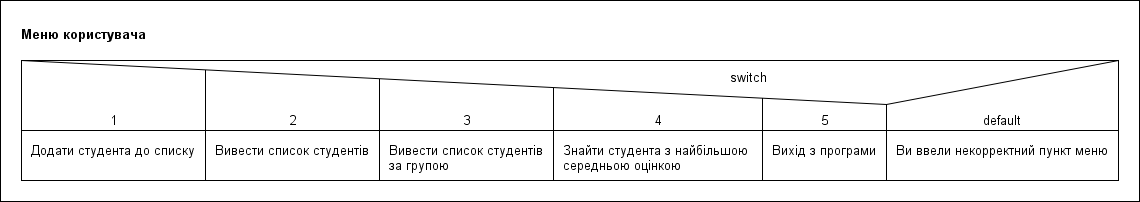
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | П.І.Б. студента. | surname | Рядок символів | «Кулик С. В.» |
| 2 | Группа студента. | group | Натуральне число | 911 912 |
| 3 | Оцінки за модуль. | raiting | Натуральні числа більші за 0 та менші від 100 | 60 85 |

**2. Методи рішення**

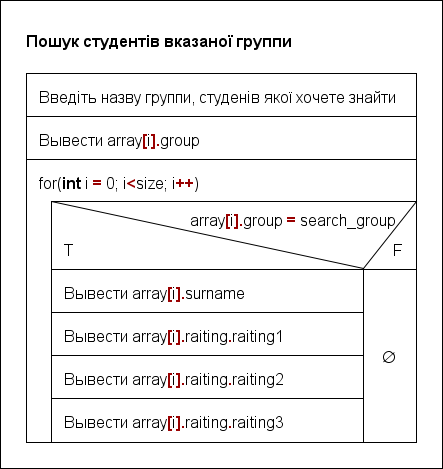
2.1. Для реалізації меню користувача, використовував оператори switch.  
2.2. Для органызації виводу студентів використав цикл та вивід полів.  
2.3. Для пошуку студентів за группою, використовував оператор if. Якщо номер введеноъ групи дорынюэ полю структури з групою, виводити інформацію з комірки масиву.  
2.4. Для пошуку студента з найбільшою середньою оцінкою, программа підраховує середнє арифметичне оцінок та повертає індекс студента в масиві.

**3. Алгоритм**

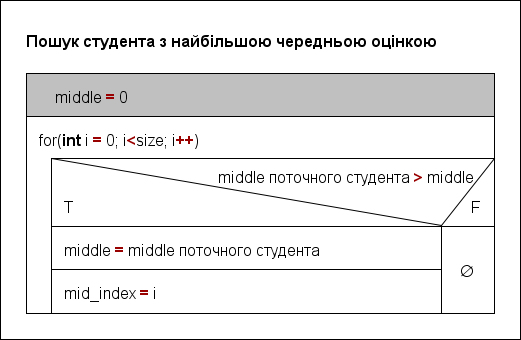
3.1. Меню користувача



3.2. Пошук студентів вказаної групи



3.3 Пошук студента з найбільшою середньою оцінкою



**4. Тести**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва тесту | Вхідні дані | Результат |
| 1. | Перший пункт меню | 1 П.І.Б. Кулик С.В. Група. 911 Оцінки за модуль 85 60 70 | Додавання студента в список |
| 2. | Другий пункт меню | 2 | Вивести список студентів з повною інформацією про них |
| 3. | Третій пункт меню | 3 Номер групи для пошуку : 911 | Вивести студентыв з 911 групи та повну інформацію про них |
| 4. | Четвертий пункт меню | 4 | Показати студента з найбільшою середньою оцінкою, в даному випадку буде Стругов. В.Є. |
| 5. | Вихід з программи | 5 | Програма завершить роботу |

**5. Текст програми**

Файл functions.h

#pragma once

#include <iostream>

#include "students.h"

void add\_student( Students array[], const int size, int &z );

void message();

void out\_student(Students array[], const int size, int& z);

void out\_group\_student(Students array[], const int size, int& z, int number\_group);

int middle(Students array[], const int size, int &mid\_index, int& z);

void search\_middle(Students array[], int mid\_index);

Файл functions.cpp

#include "functions.h"

#include <iostream>

using namespace std;

//Функция вывода пунктов меню на экран

void message()

{

cout << "=========================> МЕНЮ <==============================" << endl;

cout << "Выберете пункт меню, нажав на сответсвующую кнопку." << endl;

cout << "1. Добавить студента в список." << endl;

cout << "2. Вывести список студентов" << endl;

cout << "3. Вывести список студентов за группой" << endl;

cout << "4. Найти студента с наибольшей средней оценкой" << endl;

cout << "5. Выход" << endl;

cout << "===============================================================" << endl;

}

//функция вывода списка студентов на экран

void out\_student(Students array[], const int size, int &z)

{

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

{

cout << "Ф.И.О. студента: " << array[i].surname << endl;

cout << "Группа: " << array[i].group << endl;

cout << "Оценки за модуль: ";

cout << array[i].raiting.raiting1 << ' ';

cout << array[i].raiting.raiting2 << ' ';

cout << array[i].raiting.raiting2 << endl<<endl;

if (i == z-1)

break;

}

}

//Вывод студентов определённой групы

void out\_group\_student(Students array[], const int size, int& z, int number\_group)

{

cout << "Студенты группы: " << number\_group << endl;

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

{

if (array[i].group == number\_group)

{

cout << "Ф.И.О. студента: " << array[i].surname << endl;

cout << "Оценки за модуль: ";

cout << array[i].raiting.raiting1 << ' ';

cout << array[i].raiting.raiting2 << ' ';

cout << array[i].raiting.raiting2 << endl << endl;

}

}

}

//Поиск студента с наибольшей середней оценкой

int middle(Students array[], const int size, int &mid\_index, int& z)

{

float middle = 0;

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

{

if ((array[i].raiting.raiting1 + array[i].raiting.raiting2 + array[i].raiting.raiting3) / 3 > middle)

{

middle = (array[i].raiting.raiting1 + array[i].raiting.raiting2 + array[i].raiting.raiting3) / 3;

mid\_index = i;

}

}

return mid\_index;

}

void search\_middle(Students array[], int mid\_index)

{

cout << "Студент с наибольшей средней оценкой" << endl;

cout << "Ф.И.О. студента: " << array[mid\_index].surname << endl;

cout << "Группа: " << array[mid\_index].group << endl;

cout << "Оценки за модуль: ";

cout << array[mid\_index].raiting.raiting1 << ' ';

cout << array[mid\_index].raiting.raiting2 << ' ';

cout << array[mid\_index].raiting.raiting2 << endl << endl;

}

//функция добавления студента

void add\_student(Students array[], const int size, int &z)

{

if (z < size)

{

setlocale(LC\_ALL, "ru");

cout << "Ф.И.О студента: ";

cin.get();

cin.getline(array[z].surname, 32);

cout << "Номер группы: ";

cin >> array[z].group;

while (cin.fail())

{

if (cin.fail())

{

cin.clear();

cin.ignore();

}

cin >> array[z].group;

cin.get();

}

cout << "Оценки за модуль" << endl;

cin >> array[z].raiting.raiting1;

while (cin.fail())

{

if (cin.fail())

{

cin.clear();

cin.ignore();

}

cin >> array[z].raiting.raiting3;

cin.get();

}

cin >> array[z].raiting.raiting2;

while (cin.fail())

{

if (cin.fail())

{

cin.clear();

cin.ignore();

}

cin >> array[z].raiting.raiting2;

cin.get();

}

cin >> array[z].raiting.raiting3;

while (cin.fail())

{

if (cin.fail())

{

cin.clear();

cin.ignore();

}

cin >> array[z].raiting.raiting3;

cin.get();

}

z++;

}

else

{

cout << "Список студентов полный" << endl;

}

}

Файл students.h

#pragma once

struct Progress

{

int raiting1;

int raiting2;

int raiting3;

};

struct Students

{

char surname[32];

int group;

Progress raiting;

};

Файл Op5.cpp

//Программа для роботи з даними в передметній області

//Програма повинна уміти додавати студента в меню.

//Шукати студентів за групою

//Шукати студента з найбільшим середнім балом

//Виводити весь список студентів

//Навігація виконується через меню користувача

//Вхідні дані: П.І.Б. студента, група, оцінки за модуль  
#include <istream>

#include <Windows.h>

#include "students.h"

#include "functions.h"

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//================Оглашение переменных=====================

int number;

int number\_group;

int mid\_index = 0;

bool ex = true;

int z = 3;

const int size = 20;

Students array[size] =

{

{"Іванісік Д.В.", 911, 60, 65, 78},

{"Стругов В.Є.",911, 90, 91, 90},

{"Ветлянчук Б.М.", 911, 50, 70, 59}

};

//==================Меню пользователя====================

while (ex)

{

message();

cin >> number;

switch (number)

{

case 1:

{

add\_student(array, size, z);

system("pause");

system("cls");

break;

}

case 2:

{

out\_student(array, size, z);

system("pause");

system("cls");

break;

}

case 3:

{

cout << "Введите номер группы, студентов которой хотите найти" << endl;

cin >> number\_group;

while (cin.fail())

{

if (cin.fail())

{

cin.clear();

cin.ignore();

}

cin >> number\_group;

cin.get();

}

out\_group\_student(array, size, z, number\_group);

system("pause");

system("cls");

break;

}

case 4:

{

middle(array, size, mid\_index, z);

search\_middle(array, mid\_index);

system("pause");

system("cls");

break;

}

case 5:

{

ex = false;

}

default:

while (cin.fail())

{

if (cin.fail())

{

cin.clear();

cin.ignore();

}

cin >> number;

cin.get();

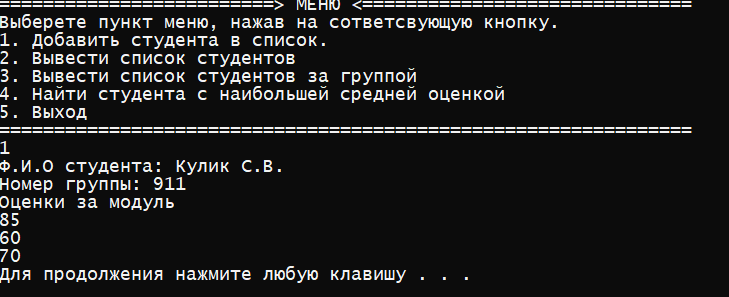
}

}

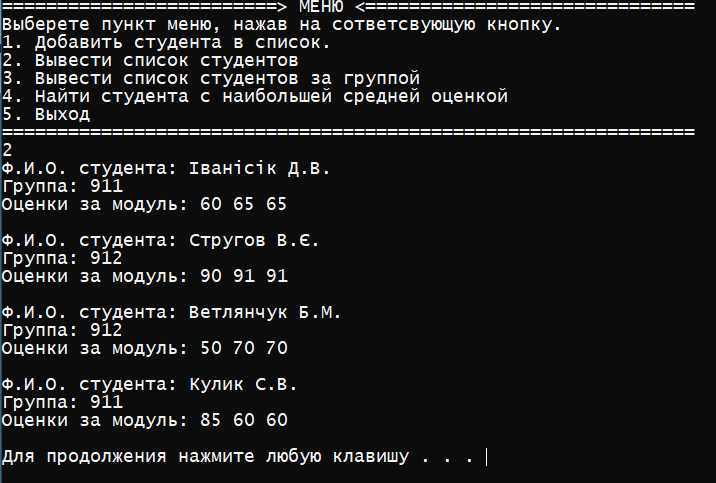
}

}

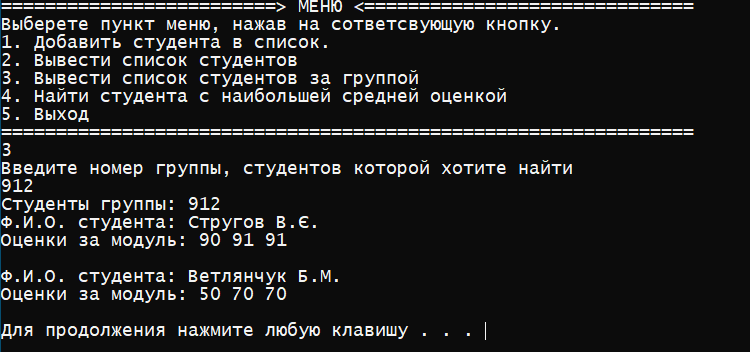
**6. Результати тестувань**  
6.1. Перший пункт меню



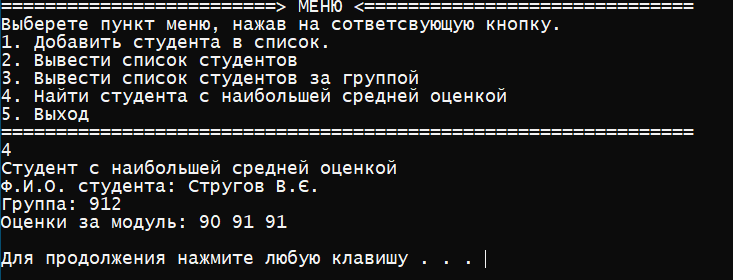
6.2. Другий пункт меню



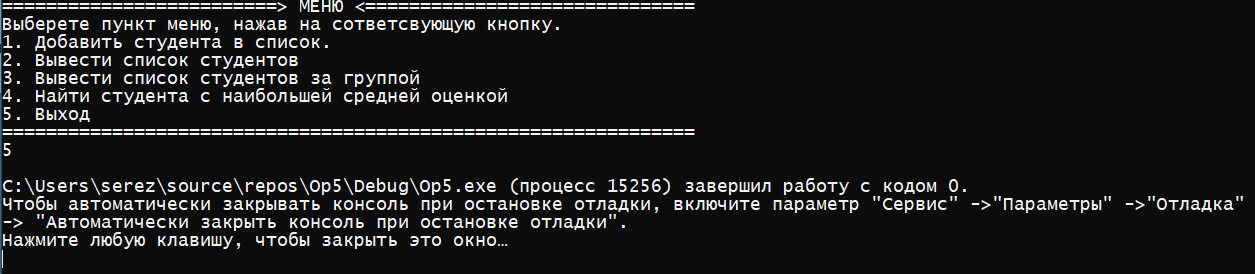
6.3. Третій пункт меню



6.4. Четвертий пункт меню



6.5. П’ятий пункт меню



Усі результати тестувань збіглися з очікуваннями. Програма правильно виконує кожен пункт меню.

**7. Висновок**

На мою думку використання структур є дуже зручним, адже сворюється новий тип данних, в я кому можна зберігати декілька різновидів інших данних, замість того, щоб оголошувати велику кількість змінних, можна оголисити структуру з відповідними полями, а в поєднанні з массивом структур можна зручно зберігати та редагувати інформацію необхідну вам.