ЗВІТ

про виконання лабораторної роботи № 9.1.B

«Послідовний пошук в масиві структур»

з дисципліни

«Алгоритмізація та програмування»

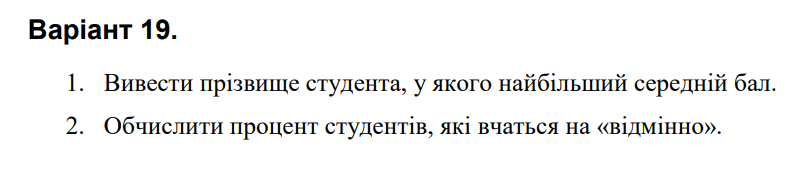
студентки групи РІ-11

Пятницької Вікторії Володимирівни

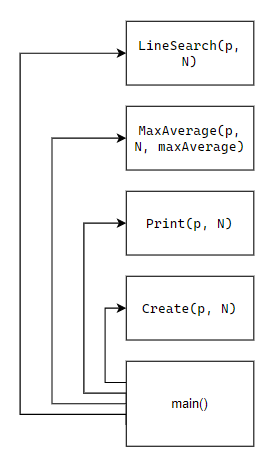
**Мета роботи**

Навчитися опрацьовувати масиви структур з об’єднаннями.

**Умова завдання:**



**Структурна схема програми:**

****

**Текст програми:**

#include <iostream>

#include <iomanip>

#include <string>

#include <Windows.h>

using namespace std;

enum Specialnist { KN, IT, MATH, PHYSIC, WORK };

enum Curs { I, II, III, IV };

string cursStr[] = { "I", "II", "III", "IV" };

string specialnistStr[] = { "Комп'ютерні науки", "Інформатика", "Математика та економіка", "Фізика та інформатика", "Трудове навчання" };

struct Student

{

string prizv;

Curs curs;

Specialnist specialnist;

struct

{

int physic;

int math;

};

union

{

int IT;

int nummet;

int pedagogy;

};

};

void Create(Student\* p, const int N)

{

int specialnist;

int curs;

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

{

cout << "Студент № " << i + 1 << ":" << endl;

cin.get();

cin.sync();

cout << " Прізвище: "; getline(cin, p[i].prizv);

cout << " Курс (0 - I, 1 - II, 2 - III, 3 - IV): ";

cin >> curs;

cout << " Спеціальність (0 - Комп’ютерні науки, 1 - Інформатика, 2 - Математика та економіка, 3 - Фізика та інформатика, 4 - Трудове навчання): ";

cin >> specialnist;

cout << " Оцінка з фізики : "; cin >> p[i].physic;

cout << " Оцінка з математики : "; cin >> p[i].math;

p[i].specialnist = (Specialnist)specialnist;

p[i].curs = (Curs)curs;

switch (p[i].specialnist)

{

case KN:

cout << " Оцінка з програмування : "; cin >> p[i].IT;

break;

case IT:

cout << " Оцінка з чисельних методів : "; cin >> p[i].nummet;

break;

case MATH:

case PHYSIC:

case WORK:

cout << " Оцінка з педагогіки : "; cin >> p[i].pedagogy;

break;

}

cout << endl;

}

}

void Print(Student\* p, const int N)

{

cout << "======================================================================================================================================================================="

<< endl;

cout << "| № | Прізвище | Курс | Cпеціальність | Оцінка з фізики | Оцінка з математики | Оцінка з програмування | Оцінка з чисельних методів | Оцінка з педагогіки |"

<< endl;

cout << "-----------------------------------------------------------------------------------------------------------------------------------------------------------------------"

<< endl;

cout << fixed;

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

{

cout << "| " << setw(2) << i + 1;

cout << "| " << setw(9) << p[i].prizv;

cout << "| " << setw(5) << cursStr[p[i].curs];

cout << "| " << setw(25) << specialnistStr[p[i].specialnist];

cout << "| " << setw(16) << p[i].physic;

cout << "| " << setw(20) << p[i].math;

switch (p[i].specialnist)

{

case KN:

cout << "| " << setw(23) << p[i].IT << "|" << setw(29) << "|" << setw(22) << "|";

break;

case IT:

cout << "| " << setw(24) << "|" << setw(28) << p[i].nummet << "|" << setw(22) << "|";

break;

case MATH:

case PHYSIC:

case WORK:

cout << "| " << setw(24) << "|" << setw(29) << "|" << setw(21) << p[i].pedagogy << "|"<< endl;

break;

}

cout << endl;

cout << "------------------------------------------------------------------------------------------------------------------------------------------------------------------------";

cout << endl;

}

}

void MaxAverage(Student\* p, const int N, double& maxAverage)

{

int i = 0;

int index = 0;

double average = (p[i].physic + p[i].math + p[i].IT) / 3.;

maxAverage = average;

for (int j = 1; j < N; j++)

{

switch (p[i].specialnist)

{

case KN:

average = (p[j].physic + p[j].math + p[j].IT) / 3.;

break;

case IT:

average = (p[j].physic + p[j].math + p[j].nummet) / 3.;

default:

average = (p[j].physic + p[j].math + p[j].pedagogy) / 3.;

}

if (maxAverage <= average)

{

maxAverage = average;

index = j;

}

}

cout << "Прізвище студента: " << p[index].prizv << ", середній бал: " << maxAverage << endl;

}

double LineSearch(Student\* p, const int N)

{

int k = 0, n = 0;

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

{

if (p[i].specialnist == KN || p[i].specialnist == IT || p[i].specialnist == MATH || p[i].specialnist == PHYSIC)

{

n++;

if (p[i].physic == 5 && p[i].math == 5 && (p[i].IT == 5 || p[i].pedagogy == 5 || p[i].nummet == 5))

k++;

}

}

return 100. \* k / n;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N;

cout << "Введіть число студентів: "; cin >> N;

Student\* p = new Student[N];

Create(p, N);

Print(p, N);

double maxAverage = 0.;

MaxAverage(p, N, maxAverage);

cout << "Процент студентів, які вчаться на «відмінно»: " << LineSearch(p, N) << "%" << endl;

return 0;

}

**Посилання на git-репозиторій з проектом:**

**Результати unit-тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "../lab\_9.1.2/lab\_9.1.2.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest91B

{

TEST\_CLASS(UnitTest91B)

{

public:

TEST\_METHOD(TestMethod1)

{

Student\* p = new Student;

p->prizv = "A";

p->curs = II;

p->specialnist = MATH;

p->physic = 5;

p->math = 5;

p->pedagogy = 5;

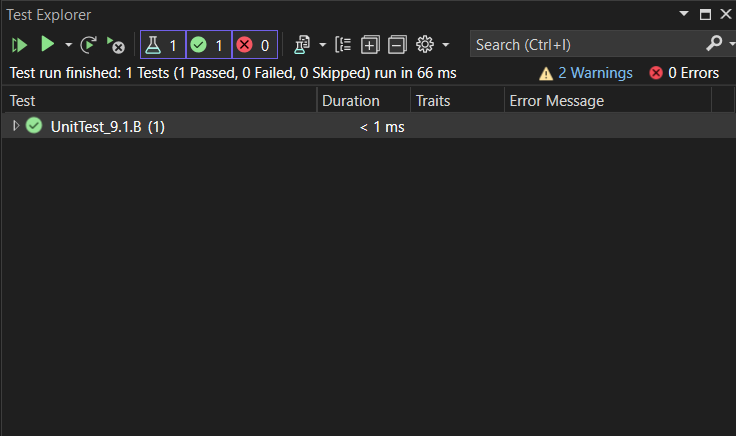
double t = LineSearch(p, 1);

Assert::AreEqual(t, 100.);

}

};

}

****

**Висновки:**

Отже, виконавши цю лабораторну роботу я навчилася опрацьовувати масиви структур з об’єднаннями.