ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА

ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



ЗВІТ

про виконання лабораторної роботи № 9,1

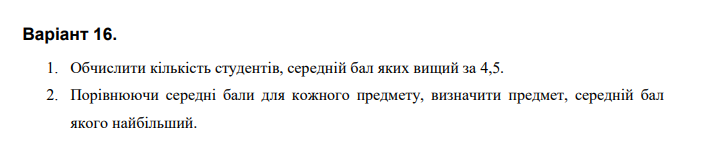
«Послідовний пошук в масиві структур»

з дисципліни «Алгоритмізація та програмування»

студентки групи ІТ-12

Прендкович Ірини Віталіївни

Умова завдання



Відповідь

#include <iostream>

#include <iomanip>

#include <string>

#include <Windows.h>

using namespace std;

enum Specialty { IT, IK, CA, KH, KB };

string strSpecialty[] = { "IT", "IK", "CA", "KH", "KB" };

struct Student

{

string prizv;

Specialty specialty;

short unsigned kurs;

short unsigned physics;

short unsigned math;

union

{ short unsigned programming;

short unsigned methods;

short unsigned pedagogika;

};

};

void Create(Student\* s, const int N);

void Print(Student\* s, const int N);

double BestAverage(Student\* s, const int N);

double GPA(Student\* s, const int N);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N;

cout << "enter the number of students ";

cin >> N;

Student\* s = new Student[N];

Create(s, N);

Print(s, N);

cout << "| the highest average score:" << fixed << setprecision(2) << BestAverage(s, N) << setw(82) << "|" << endl;

cout << "| GPA " << char(34) << "4,5" << char(34) << " or " << char(34) << "5" << char(34) << ": " << fixed << setw(6) << setprecision(2) << GPA(s, N) << "people" << setw(51) << "|" << endl; cout << "=================================================================================================================" << endl;

delete[] s;

}

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

{

int kurs, specialty;

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

{

cout << "Student# " << i + 1 << ":" << endl;

cin.get();

cin.sync();

cout << " name ";

getline(cin, s[i].prizv);

cout << " course ";

cin >> s[i].kurs;

cout << " special (0 - IT, 1 - IK, 2 - CA, 3 - KH, 4 - KB): "; cin >> specialty;

s[i].specialty = (Specialty)specialty;

cout << " grade in physics: "; cin >> s[i].physics;

cout << " grade in maths: "; cin >> s[i].math;

switch (specialty)

{ case 0:

cout << " grade in informatic: ";

cin >> s[i].programming;

break;

case 3:

cout << " grade in method"; cin >> s[i].methods;

break;

case 1:

case 2:

case 4:

cout << " grade in pedagogik: "; cin >> s[i].pedagogika;

break; }

}

}void Print(Student\* s, const int N)

{

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

<< endl;

cout << "| № | Surname | Course | Specialty | Physics | Mathematics | Programming| Method|Pedagogik|"

<< endl;

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

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

{

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

<< "| " << setw(13) << left << s[i].prizv

<< "| " << setw(3) << right << s[i].kurs << " "

<< "| " << setw(8) << right << strSpecialty[s[i].specialty] << " "

<< "| " << setw(4) << right << s[i].physics << " "

<< "| " << setw(6) << right << s[i].math << " ";

switch (s[i].specialty)

{

case 0:

cout << "| " << setw(7) << right << s[i].programming << " |" << setw(18) << "|" << setw(13) << "|" << endl; break; case 3: cout << "| " << setw(15) << "|" << setw(10) << right << s[i].methods << " |" << setw(13) << "|" << endl; break; case 1: case 2: case 4: cout << "| " << setw(15) << "|" << setw(18) << "|" << right << setw(7) << s[i].pedagogika << " |" << endl; break;

}

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

}

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

}double BestAverage(Student\* s, const int N)

{

double best = 0;

double F;

double M;

double I;

double Meth;

double Pedag;

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

{ switch (s[i].specialty)

{ case 0:

F = (s[i].physics) / N;

M = (s[i].math) / N;

I = (s[i].programming) / N;

Meth = (s[i].methods) / N;

Pedag = (s[i].pedagogika) / N;

break;

case 3:

F = (s[i].physics) / N;

M = (s[i].math) / N;

I = (s[i].programming) / N;

Meth = (s[i].methods) / N;

Pedag = (s[i].pedagogika) / N;

break;

case 1:

case 2:

case 4:

F = (s[i].physics) / N;

M = (s[i].math) / N;

I = (s[i].programming) / N;

Meth = (s[i].methods) / N;

Pedag = (s[i].pedagogika) / N;

break;

}

if(F > M && F > I && F > Meth && F > Pedag)

best = F;

if (M > F && M > I && M > Meth && M > Pedag)

best = M;

if (I > F && I > M && I > Meth && I > Pedag)

best = I;

if (Meth > M && Meth > I && Meth > F && Meth > Pedag)

best = Meth;

if (Pedag > F && Pedag > I && Pedag > Meth && Pedag > M)

best = Pedag;

}

return best;

}

double GPA(Student\* s, const int N)

{

int k = 0;

double best;

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

if (best >= 4,5)

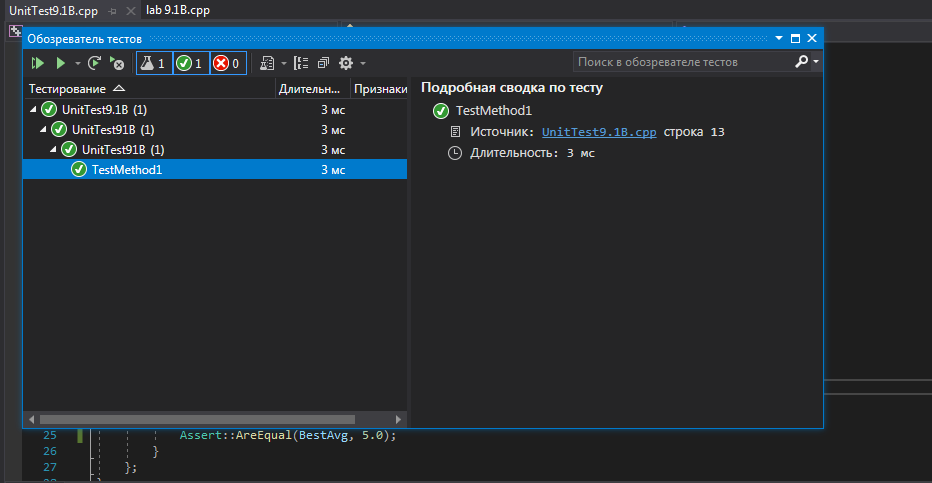
k++;

else

cout << "no >4.5";

return k / N;

}



#include "pch.h"

#include "CppUnitTest.h"

#include "../lab 9.1B/lab 9.1B.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest91B

{

TEST\_CLASS(UnitTest91B)

{

public:

TEST\_METHOD(TestMethod1)

{

int N = 1;

Student\* s = new Student[N];

s[0].prizv = "join";

s[0].kurs = 1;

s[0].specialty = IT;

s[0].physics = 5;

s[0].math = 5;

s[0].programming = 5;

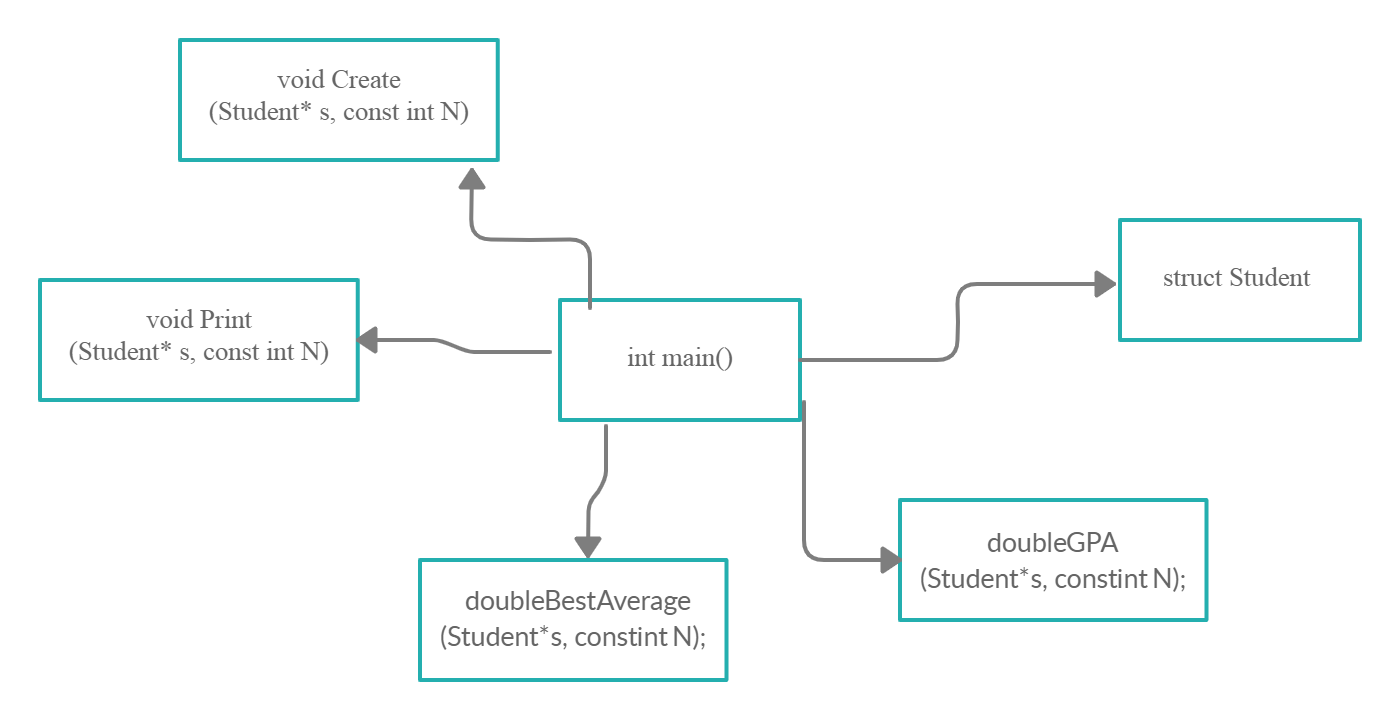
double BestAvg = BestAverage(s, N);

Assert::AreEqual(BestAvg, 5.0);

}

};

}



Висновок

На цій лабораторній я навчилася опрацьовувати масиви структур з об’єднаннями.