**ЗВІТ**

**до лабораторної роботи № < 11.2 >**

**« Послідовний пошук у бінарному файлі »**

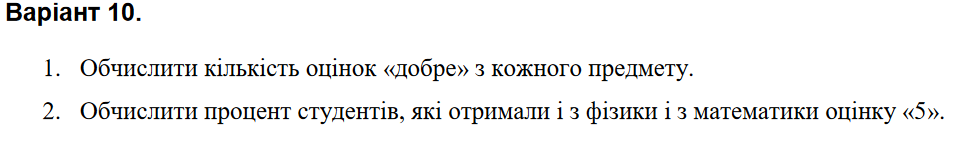
**з дисципліни**

**«Алгоритмізація та програмування»**

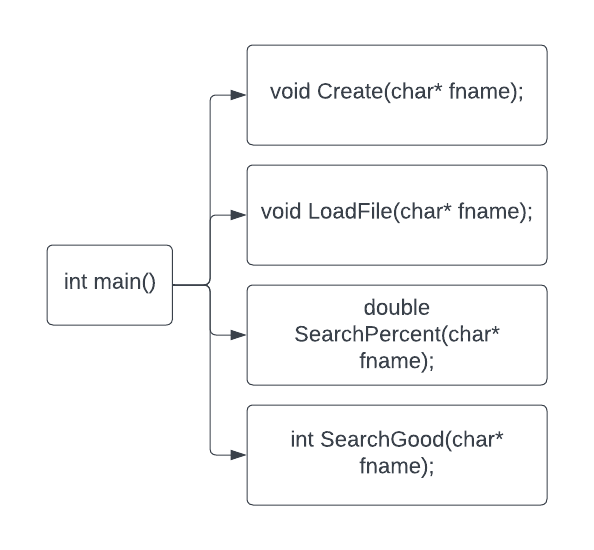
**студента групи ІН-105Б**

**Горанова Анастаса**

**Умова завдання:**



**Структурна схема:**



**Текст програми:**

// 11.2.cpp

// < Горанов Анастас >

// Лабораторна робота № 11.2

// Послідовний пошук у бінарному файлі

// Варіант 10

#include <iostream>

#include <stdio.h>

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

enum Specialnist { KN, ME, IN, FI, TN };

string specialnistList[] = { "KN", "ME", "IN", "FI", "TN" };

struct Student

{

char surname[100];

int course;

Specialnist spec;

int fiz;

int mat;

union

{

int prog;

int chis\_metod;

int pedagog;

};

};

void Create(char\* fname);

void LoadFile(char\* fname);

int SearchGood(char\* fname);

double SearchPercent(char\* fname);

int main()

{

char fname[61];

char ch;

do

{

cout << endl;

cout << "Choose:" << endl;

cout << "[1] - Entering and saving information" << endl;

cout << "[2] - Loading and output of information" << endl;

cout << "[3] - Get the number of 'good' grades among students" << endl;

cout << "[4] - Find students who got 5 in both physics and mathematics" << endl;

cout << "[0] - Quit" << endl << endl;

cout << "Your choice: "; cin >> ch;

switch (ch)

{

case '0':

break;

case '1':

cin.get(); // очищуємо буфер клавіатури – щоб не було символу

cin.sync(); // "кінець рядка", який залишився після вводу числа

cout << endl;

cout << "File Name: "; cin.getline(fname, sizeof(fname));

Create(fname);

break;

case '2':

cin.get(); // очищуємо буфер клавіатури – щоб не було символу

cin.sync(); // "кінець рядка", який залишився після вводу числа

cout << endl;

cout << "File Name: "; cin.getline(fname, sizeof(fname));

cout << endl;

LoadFile(fname);

break;

case '3':

cin.get(); // очищуємо буфер клавіатури – щоб не було символу

cin.sync(); // "кінець рядка", який залишився після вводу числа

cout << endl;

cout << "File Name: "; cin.getline(fname, sizeof(fname));

cout << endl;

cout << "Number of 'good' grades among students: " << SearchGood(fname) << endl;

cout << endl;

break;

case '4':

cin.get(); // очищуємо буфер клавіатури – щоб не було символу

cin.sync(); // "кінець рядка", який залишився після вводу числа

cout << endl;

cout << "File Name: "; cin.getline(fname, sizeof(fname));

cout << endl;

cout << "Percentage of students with '5' in physics and mathematics: " << SearchPercent(fname) << endl;

cout << endl;

break;

default:

cout << "ERROR ";

}

} while (ch != '0');

return 0;

}

void Create(char\* fname)

{

ofstream f(fname, ios::binary);

if (!f)

{

cerr << "Error opening file '" << fname << "'" << endl;

f.close();

return;

}

int spec;

Student student;

char ch;

int i = 0;

do

{

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

cout << "Surname: "; cin >> student.surname;

cout << "Course: "; cin >> student.course;

cout << "Grades in physics: "; cin >> student.fiz;

cout << "Grades in math: "; cin >> student.mat;

cout << "A specialty (0 - KN, 1 - ME, 2 - IN, 3 - FI, 4 - TN): "; cin >> spec;

if (spec > 4) {

cout << endl;

cout << "There is no such specialty.Selected 2 (IN)";

spec = 0;

}

student.spec = (Specialnist)spec;

switch (student.spec)

{

case KN:

cout << "Grades in programming: "; cin >> student.prog;

break;

case IN:

cout << "Grades in num.methods: "; cin >> student.chis\_metod;

break;

case ME:

case FI:

case TN:

cout << "Grades in pedagogy: "; cin >> student.pedagog;

break;

}

if (!f.write((char\*)&student, sizeof(Student)))

{

cerr << "Error writing to file." << endl;

f.close();

}

cout << "Continue? (Y/N) "; cin >> ch;

} while (ch == 'Y' || ch == 'y');

f.close();

}

void LoadFile(char\* fname)

{

ifstream f(fname, ios::binary);

if (!f)

{

cerr << "Error opening file '" << fname << "'" << endl;

f.close();

return;

}

Student student;

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

<< endl;

cout << "| # | Surname | Course | Specilization | Physic | Mathematic | Programming | Numerical Methods | Pedagogy |"

<< endl;

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

<< endl;

int i = 0;

while (f.read((char\*)&student, sizeof(Student)))

{

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

cout << "| " << setw(8) << left << student.surname

<< "| " << setw(4) << right << student.course << " "

<< " | " << setw(14) << left << specialnistList[student.spec]

<< setw(4) << "| " << setw(3) << left << student.fiz << " "

<< setw(6) << "| " << setw(6) << left << student.mat << " ";

switch (student.spec)

{

case KN:

cout << setw(8) << "| " << setw(5) << left << student.prog << setw(21)

<< " | " << setw(10) << left

<< "| " << left << " |" << endl;

break;

case ME:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << student.pedagog << " |" << endl;

break;

case IN:

cout << "" << setw(13) << "| " << setw(10) << " |" << " "

<< setw(9) << student.chis\_metod << " |" << " "

<< setw(8) << "" << " |" << endl;

break;

case FI:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << student.pedagog << " |" << endl;

break;

case TN:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << student.pedagog << " |" << endl;

break;

}

}

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

<< endl;

cout << endl;

f.close();

}

double SearchPercent(char\* fname)

{

ifstream f(fname, ios::binary);

if (!f)

{

cerr << "Error opening file '" << fname << "'" << endl;

f.close();

return 0;

}

Student student;

double percent;

int k = 0; int kilk = 0;

while (f.read((char\*)&student, sizeof(Student)))

{

if (student.mat == 5 && student.fiz == 5)

k++;

kilk++;

}

percent = static\_cast<double>(k) / kilk \* 100;

return percent;

f.close();

}

int SearchGood(char\* fname)

{

ifstream f(fname, ios::binary);

if (!f)

{

cerr << "Error opening file '" << fname << "'" << endl;

f.close();

return 0;

}

Student student;

int k = 0, n = 0;

while (f.read((char\*)&student, sizeof(Student)))

{

if (student.mat == 4) {

k++;

n++;

}

if (student.fiz == 4) {

k++;

n++;

}

switch (student.spec) {

case KN:

if (student.prog == 4) {

k++;

n++;

}

break;

case ME:

if (student.pedagog == 4) {

k++;

n++;

}

break;

case IN:

if (student.chis\_metod == 4) {

k++;

n++;

}

break;

case TN:

if (student.pedagog == 4) {

k++;

n++;

}

break;

case FI:

if (student.pedagog == 4) {

k++;

n++;

}

break;

}

cout << "A student " << student.surname << " has " << k << " good grades." << endl;

k = 0;

}

cout << endl;

f.close();

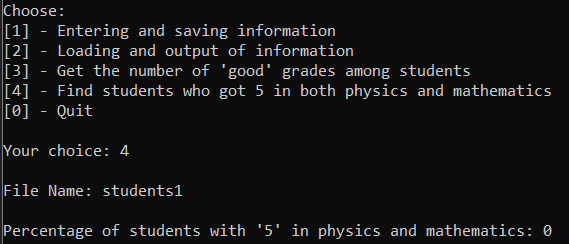
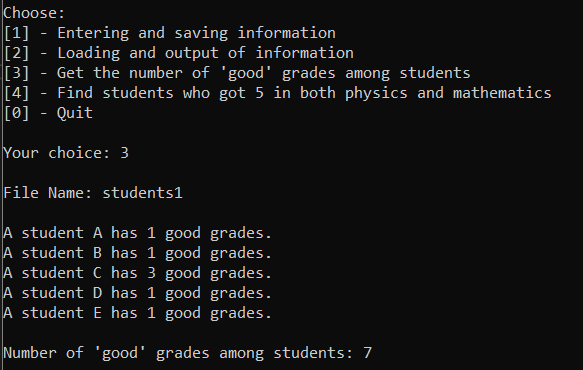
return n;

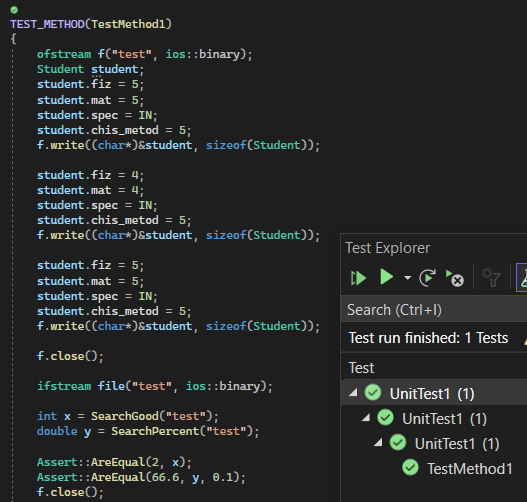
}

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

*https://github.com/StassNG/11.2*

**Результати програми та unit-тесту:**

****

****

**Висновок:** я навчився послідовному пошуку в бінарному файлі.