**ЗВІТ**

**до лабораторної роботи № < 11.4 >**

**« Опрацювання бінарного файлу »**

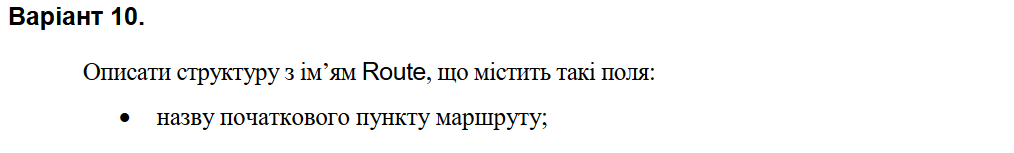
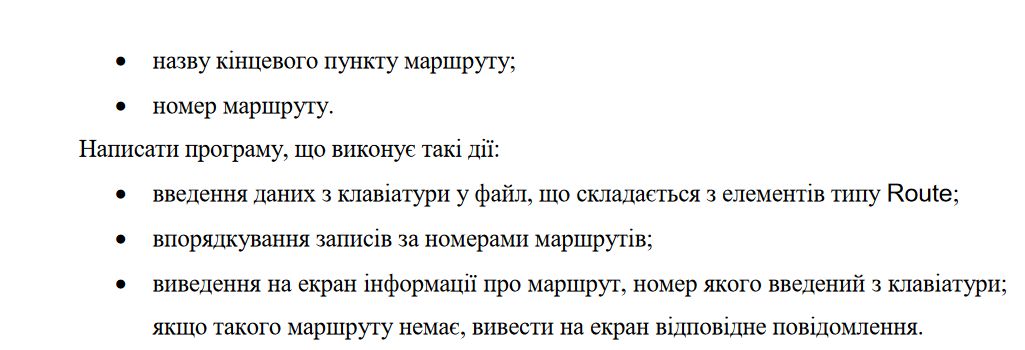
**з дисципліни**

**«Алгоритмізація та програмування»**

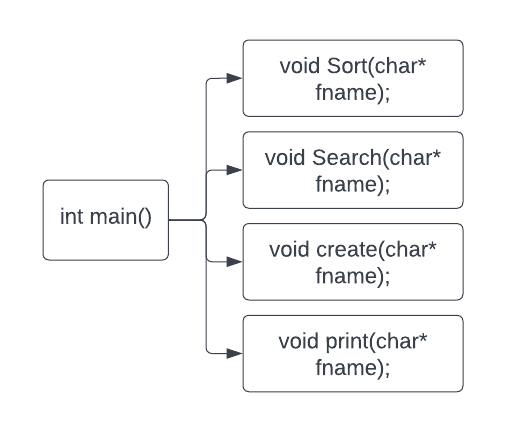
**студента групи ІН-105Б**

**Горанова Анастаса**

**Умова завдання:**



**Структурна схема:**



**Текст програми:**

// 11.4.cpp

// < Горанов Анастас >

// Лабораторна робота № 11.4

// Опрацювання бінарного файлу

// Варіант 10

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <iomanip>

#include <fstream>

using namespace std;

struct route

{

char start[101];

char end[101];

int nomer;

};

void Sort(char\* fname);

void Search(char\* fname);

void create(char\* fname);

void print(char\* fname);

int main() {

char fname[101];

int choice;

do {

cout << "--------------------------\n";

cout << "Menu\n";

cout << "[1] - Sorting\n";

cout << "[2] - Search\n";

cout << "[3] - Entering and saving\n";

cout << "[4] - Download and display\n";

cout << "[0] - Quit\n";

cout << "--------------------------\n";

cout << "Choose: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Sorting:\n";

cin.get();

cin.sync();

cout << "Enter file name: "; cin.getline(fname, sizeof(fname));

Sort(fname);

break;

case 2:

{

cout << "Search:\n";

cin.get();

cin.sync();

cout << "Enter file name: "; cin.getline(fname, sizeof(fname));

Search(fname);

break;

}

case 3:

cout << "Entering and saving:\n";

cin.get();

cin.sync();

cout << "Enter file name: "; cin.getline(fname, sizeof(fname));

create(fname);

break;

case 4:

cout << "Display information:\n";

cin.get();

cin.sync();

cout << "Enter file name: "; cin.getline(fname, sizeof(fname));

print(fname);

break;

case 0:

break;

default:

cout << "--------------------------\n";

cout << "Меню\n";

cout << "[1] - Sorting\n";

cout << "[2] - Search\n";

cout << "[3] - Loading and display\n";

cout << "[4] - Entering and saving\n";

cout << "[0] - Вихід\n";

cout << "--------------------------\n";

cout << "Choose:";

cin >> choice;

}

} while (choice != 0);

}

void Sort(char\* fname) {

ifstream f(fname, ios::binary);

if (!f)

{

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

f.close();

return;

}

f.seekg(0, ios::end);

int size = f.tellg() / sizeof(route);

f.seekg(0, ios::beg);

route\* r = new route[size];

route tmp;

f.read((char\*)r, size \* sizeof(route));

for (int i0 = 0; i0 < size - 1; i0++) {

for (int i1 = 0; i1 < size - i0 - 1; i1++) {

if (r[i1].nomer > r[i1 + 1].nomer)

{

tmp = r[i1];

r[i1] = r[i1 + 1];

r[i1 + 1] = tmp;

}

}

}

f.close();

ofstream file(fname, ios::binary);

file.write((char\*)r, size \* sizeof(route));

file.close();

}

void Search(char\* fname) {

int nomer;

bool n = false;

cout << "Enter route number for search: ";

cin.sync(); cin >> nomer;

ifstream f(fname, ios::binary);

if (!f)

{

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

return;

}

route r;

while (f.read((char\*)&r, sizeof(route)))

{

if (r.nomer == nomer)

{

n = true;

break;

}

}

if (n == true)

{

cout << " Route number " << nomer << " found: " << endl;

cout << " Start: " << r.start << endl;

cout << " End: " << r.end << endl;

}

else

{

cout << " No route with this number was found." << endl;

}

}

void create(char\* fname)

{

ofstream fout(fname, ios::binary);

char ch; route r;

do

{

cout << "Route number: "; cin >> r.nomer;

cout << "Start: "; cin >> r.start;

cout << "End: "; cin >> r.end;

if (!fout.write((char\*)&r, sizeof(route)))

{

cerr << "Write error." << endl;

}

cout << "Continue? (y/n): "; cin >> ch;

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

cout << endl;

fout.close();

}

void print(char\* fname)

{

ifstream f(fname, ios::binary);

if (!f)

{

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

return;

}

route r;

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

<< endl;

cout << "| Start | End | Number" << setw(3) << "|"

<< endl;

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

<< endl;

while (f.read((char\*)&r, sizeof(route)))

{

cout << "| " << setw(7) << left << r.start << " "

<< "| " << setw(6) << right << r.end << " "

<< "| " << setw(7) << left << r.nomer << " |" << endl;

}

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

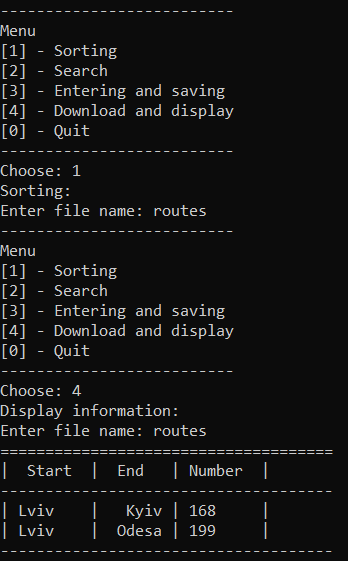
f.close();

}

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

*https://github.com/StassNG/11.4*

**Результати програми та unit-тесту:**

****

**Висновок:** я навчився опрацьовувати бінарні файли.