Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІПІ

ЗВІТ

з виконання лабораторної роботи № 3

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 26

Виконав:

студент 1-го курсу

гр. ІП-21 ФІОТ

Скрипець Ольга Олександрівна

Київ 2023

1. **Посилання на репозиторій**

https://github.com/uhpolerr/op\_lab3

1. **Код програми**

**func.h**

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Train {

public:

int number;

string destination;

string departureTime;

Train() {}

Train(int num, string dest, string depTime) {

number = num;

destination = dest;

departureTime = depTime;

}

};

string menu();

bool isValidTrainNumber(int number);

bool isValidDestination(string destination);

bool isValidDepartureTime(string departureTime);

int validationNumber();

string validationDestination();

string validationDepartureTime();

void addTrain(Train trains[], int& numTrains, const int max);

void findLatestTrainToDestination(Train trains[], int numTrains);

**func.cpp**

#include <iostream>

#include <string>

#include <Windows.h>

#include "func.h"

using namespace std;

string menu() {

string choice;

cout << "----Enter 1 to add a train" << endl;

cout << "----Enter 2 to determine the latest train that leaves for the specified destination" << endl;

cout << "----Click something else to finish" << endl;

cin >> choice;

return choice;

}

bool isValidTrainNumber(int number) {

if (number < 1 || number > 999) {

return false;

}

return true;

}

bool isValidDestination(string destination) {

if (destination.empty()) {

return false;

}

for (char c : destination) {

if (!isalpha(c)) {

return false;

}

}

return true;

}

bool isValidDepartureTime(string departureTime) {

if (departureTime.size() != 5) {

return false;

}

if (departureTime[2] != '-') {

return false;

}

int hour = stoi(departureTime.substr(0, 2));

int minute = stoi(departureTime.substr(3, 2));

if (hour < 0 || hour > 23 || minute < 0 || minute > 59) {

return false;

}

return true;

}

int validationNumber() {

int number;

bool isValid = false;

do {

cout << "Enter the train number: ";

cin >> number;

isValid = isValidTrainNumber(number);

if (!isValid) {

cout << "The train number must be from 1 to 999" << endl;

}

} while (!isValid);

return number;

}

string validationDestination() {

string destination;

bool isValid = false;

do {

cout << "Enter your destination: ";

cin >> destination;

isValid = isValidDestination(destination);

if (!isValid) {

cout << "The destination is not entered correctly" << endl;

}

} while (!isValid);

return destination;

}

string validationDepartureTime() {

string departureTime;

bool isValid = false;

do {

cout << "Enter the time of departure in YY-XX format: ";

cin >> departureTime;

isValid = isValidDepartureTime(departureTime);

if (!isValid) {

cout << "Incorrect entry of departure time" << endl;

}

} while (!isValid);

return departureTime;

}

void addTrain(Train trains[], int& numTrains, const int max) {

if (numTrains < max) {

int number = validationNumber();

string destination = validationDestination();

string departureTime = validationDepartureTime();

trains[numTrains] = Train(number, destination, departureTime);

numTrains++;

cout << "Train added successfully" << endl;

}

else {

cout << "The list of trains is full, it is not possible to add more" << endl;

}

}

void findLatestTrainToDestination(Train trains[], int numTrains) {

string destination;

bool isValid = false;

do {

cout << "Enter your destination: ";

cin >> destination;

isValid = isValidDestination(destination);

if (!isValid) {

cout << "The destination is not entered correctly" << endl;

}

} while (!isValid);

int latestTrainIndex = -1;

for (int i = 0; i < numTrains; i++) {

if (trains[i].destination == destination) {

if (latestTrainIndex == -1 || trains[i].departureTime > trains[latestTrainIndex].departureTime) {

latestTrainIndex = i;

}

}

}

if (latestTrainIndex == -1) {

cout << "There are no trains to " << destination << endl;

}

else {

cout << "The train that goes to " << destination << " at the latest, has a number " << trains[latestTrainIndex].number << endl;

}

}

**Lab3.cpp**

#include <iostream>

#include <string>

#include <Windows.h>

#include "func.h"

using namespace std;

int main() {

setlocale(LC\_CTYPE, "ukr");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int max = 100;

Train trains[max] = {

Train(1, "Kyiv", "01-30"),

Train(2, "Kyiv", "16-15"),

Train(3, "Kyiv", "14-23"),

Train(4, "Lviv", "20-45"),

Train(5, "Lviv", "18-30"),

Train(6, "Lviv", "20-17"),

Train(7, "Lviv", "22-15"),

Train(8, "Odessa", "23-45"),

Train(9, "Odessa", "01-06"),

Train(10, "Dnipro", "03-50")

};

int numTrains = 10;

while (true) {

string choice = menu();

if (choice == "1") {

addTrain(trains, numTrains, max);

}

else if (choice == "2") {

findLatestTrainToDestination(trains, numTrains);

}

else {

break;

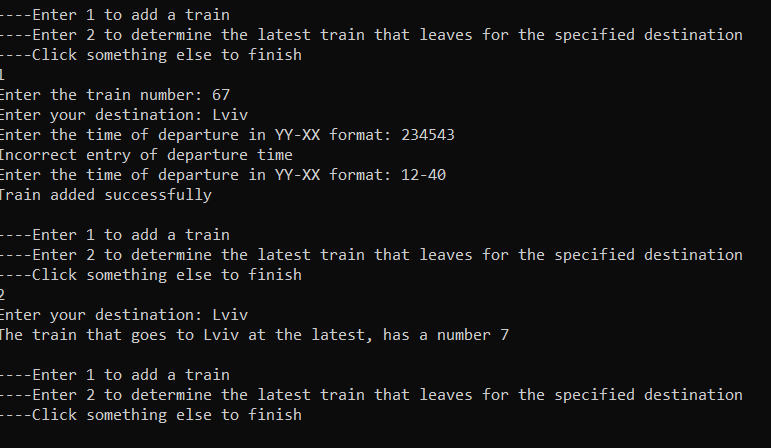
}

}

return 0;

}

**Перевірки**

****