Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 2 з
дисципліни «Основи програмування –
2. Методології програмування»

«Бінарні файли»

Варіант 14

Виконав студент ІП-13, Замковий Дмитро Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірив _Вечерковська Анастасія Сергіївна_____

(прізвище, ім'я, по батькові)

Лабораторна робота 2
Бінарні файли

Мета заняття: вивчити особливості створення і обробки бінарних файлів

Завдання: Варіант 14. Створити файл з розкладом руху міжміських автобусів: пункт призначення, час відправлення, та тривалість поїздки (у годинах та хвилинах). Видалити з файлу інформацію про рейси, в яких хоча б частину шляху потрапляє на нічний час (з 23:00 до 6:00). Виняток становлять маршрути з тривалістю шляху понад 17 годин. Визначити час відправлення останнього автобуса в заданий пункт призначення

Код програми:

На мові програмування C++:

Файл CppLab.cpp:

```
#include "Lib.h"

int main() {
	vector<Timetable> busses;
	vector<Timetable> updateBusses;
	Timetable bus;

	string fileName = "txt.bin";
	int h, m;

	writeFile(busses, bus, fileName);
	busses.clear();
	cout << endl;

	cout << "Verified records for compliance with the condition: ";
	readFile(busses, bus, fileName);


	for (int i = 0; i < busses.size(); i++)
	{
		busses[i].outputTimetable();
		updateBusses.push_back(busses[i]);
		h = busses[i].departure_time.hour + busses[i].duration_trip.hour;
		m = busses[i].departure_time.minute + busses[i].duration_trip.minute;
		if (m >= 60)
		{
			m = m - 60;
			h++;
		}
		if (h >= 23 || h < 6 || (h == 6 && m == 0))
		{
			if (busses[i].duration_trip.hour >= 17)
			{
				cout << " - night time, but more than 17 hours";
			}
			else
			{
				cout << " - night time, but less than 17 hours";
				updateBusses.pop_back();
			}
		}
	}
```

```
	ofstream fout;
	fout.open(fileName, ios::binary);

	for (int i = 0; i < updateBusses.size(); i++)
	{
		fout.write((char*)&updateBusses[i], sizeof(Timetable));
	}

	fout.close();
	cout << endl << "File after deletion:";

	for (int i = 0; i < updateBusses.size(); i++)
	{
		busses[i].outputTimetable();
	}
}
```

Файл Lib.h:

```
#pragma once
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Time
{
	int hour;
	int minute;
	Time();
	void setTime();
	string getTime();
};

struct Timetable
{
	string destination;
	Time departure_time;
	Time duration_trip;
	Timetable();
	void setTimetable();
	void outputTimetable();
};

void inputBus(vector<Timetable> &res);
void outputBusses(vector<Timetable> &busses);
void readFile(vector<Timetable>&, Timetable&, string fileName);
void writeFile(vector<Timetable>&, Timetable&, string fileName);
```

Файл Lib.cpp:

```
#include "Lib.h"

Time::Time() {
	hour = 0;
	minute = 0;
```

```
}

void Time::setTime() {
	cout << "Enter hour: ";
	cin >> hour;

	while (hour < 0 || hour > 23) {
		cout << "Please try again! Value of hour can range from 0 to 23 inclusive" <<
endl;
		cout << "Enter hour: ";
		cin >> hour;
	}

	cout << "Enter minute: ";
	cin >> minute;

	while (minute < 0 || minute > 59) {
		cout << "Please try again! Value of minute can range from 0 to 59 inclusive"
<< endl;
		cout << "Enter minute: ";
		cin >> minute;
	}
}

string Time::getTime()
{
	return to_string(hour) + ":" + to_string(minute);
}

Timetable::Timetable() {
	destination = "";
}

void Timetable::setTimetable() {
	cout << endl << "Enter destination: ";
	cin >> destination;
	while (destination.size() < 2)
	{
		cout << "Please try again! The number of characters in destination name is at
least 3" << endl;
		cin >> destination;

	}

	cout << "Set departure time" << endl;
	departure_time.setTime();

	cout << "Set duration of trip" << endl;
	duration_trip.setTime();
}

void Timetable::outputTimetable()
{
	cout << endl << "City: " + destination + "\tDeparture time: " +
departure_time.getTime() + "\tDuration of trip: " + duration_trip.getTime();
}

void inputBus(vector<Timetable> &res)
{
	unsigned int leng;

	cout << endl << "Enter how many buses you want to add: ";
	cin >> leng;

	for (int i = 0; i < leng; i++)
```

```
	{
		Timetable t;
		t.setTimetable();
		res.push_back(t);
	}
}

void writeFile(vector<Timetable>& busses, Timetable& bus, string fileName)
{
	char rewriteMod;
	ofstream fout;
	cout << "Do you want to rewrite the file (y/n)?";
	while (true)
	{
		cin >> rewriteMod;
		if (rewriteMod == 'y')
		{
			fout.open(fileName, ios::binary);
			break;
		}
		else if (rewriteMod == 'n')
		{
			readFile(busses, bus, fileName);
			cout << endl << "Old file:" << endl;
			outputBusses(busses);
			cout << endl;
			fout.open(fileName, ios::app || ios::binary);
			break;
		}
		else
		{
			cout << "ERROR: Enter y or n" << endl;
		}
	}

	if (!fout.is_open())
	{
		cout << "ERROR: Cannot open file";
		exit(0);
	}
	else
	{
		inputBus(busses);
		for (int i = 0; i < busses.size(); i++)
		{
			fout.write((char*)&busses[i], sizeof(Timetable));
		}

	}
	fout.close();
}

void readFile(vector<Timetable>& busses, Timetable& bus, string fileName)
{
	ifstream fin;

	fin.open(fileName, ios::binary);

	if (!fin.is_open())
	{
		cout << "ERROR: Unable to read file" << endl;
		exit(0);
	}
	else
	{
```

```
		while (fin.read((char*)&bus, sizeof(Timetable)))
		{
			busses.push_back(bus);
		}
	}

	fin.close();
}

void outputBusses(vector<Timetable>& busses)
{
	for (int i = 0; i < busses.size(); i++)
	{
		busses[i].outputTimetable();
	}
}
```

На мові програмування Python:

Файл main.py:

```
import modulLab2 as mod


def main():
    file_name = 'text.bin'
    new_file_name = 'new.text.bin'
    mode = ''
    while not (mode == 'a' or mode == 'w'):
        mode = input('Select the write mode to the file (a/w)? ')

    if mode == 'a':
        print('\nOld information in file:')
        mod.read_file(file_name)
        print('\n')

    mod.write_file(file_name, mode)
    print('\nInformation in the file without verification:')
    mod.read_file(file_name)

    mod.remove_unnecessary_items(new_file_name, file_name)
    print('\nInformation in the file after removing extra items:')
    mod.read_file(new_file_name)


if __name__ == '__main__':
    main()
```

Файл modulLab2.py:

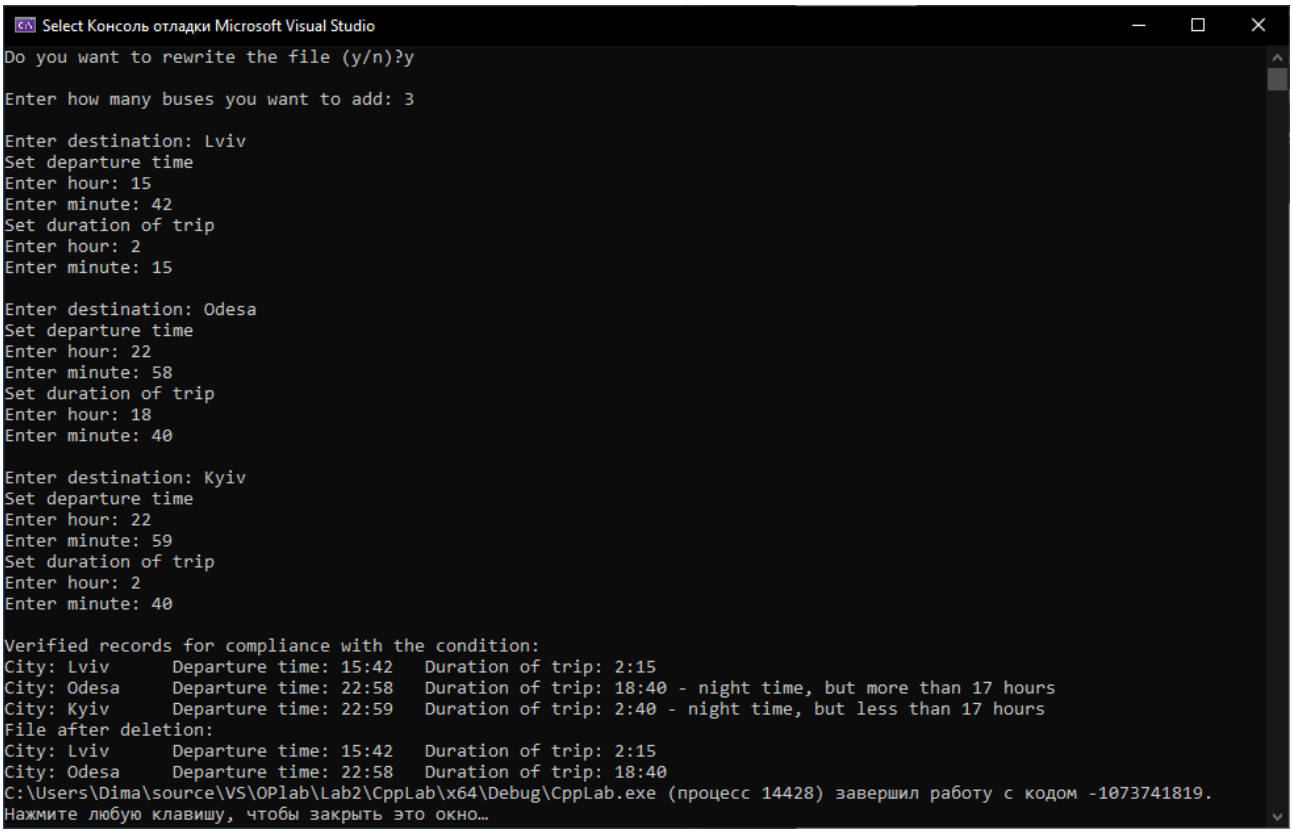
```
from pickle import dump, load
import time


def get_time(str_time: str):
    res = time.strptime(str_time, "%H:%M")
    return res
```

```
def get_arrival_time(t1, t2):
    if_night = False
    h1 = t1.tm_hour
    h2 = t2.tm_hour
    m1 = t1.tm_min
    m2 = t2.tm_min
    h = h1 + h2
    m = m1 + m2
    if m >= 60:
        h = h + 1
        m = m - 60
    if h >= 24:
        h = h - 24
        if_night = True
    else:
        if (h1 >= 23 or h1 < 6 or (h1 == 6 and m1 == 0)) or (h >= 23 or h < 6 or
(h == 6 and m == 0)):
            if_night = True
    return time.strptime(str(h) + ":" + str(m), "%H:%M"), if_night


def write_file(file_name: str, mode: str) -> None:
    bus = {}
    with open(file_name, f'{mode}b') as file:
        leng = int(input('Enter how many buses you want to add: '))
        if leng <= 0:
            print('ERROR: incorrect number specified')
        else:
            for i in range(leng):
                bus['destination'] = input('\nEnter the destination: ')
                bus['departure_time'] = input('Enter the time of departure in
the format <hh:mm>: ')
                bus['duration_trip'] = input('Enter the travel time in the
format <hh:mm>: ')
                dump(bus, file)


def read_file(file_name: str) -> None:
    with open(file_name, 'rb') as file:
        file.seek(0)

        while True:
            try:
                bus = load(file)
                departure_time = get_time(bus['departure_time'])
                duration_trip = get_time(bus['duration_trip'])
                arrival_time, if_night = get_arrival_time(departure_time,
duration_trip)

                print(f"\nDestination: {bus['destination']}")
                if if_night: print('Is NIGHT!')
                print(f"Time of departure:
{departure_time.tm_hour}:{departure_time.tm_min}"
                      f"\nTime of travel:
{duration_trip.tm_hour}:{duration_trip.tm_min}"
                      f"\nArrival time:
{arrival_time.tm_hour}:{arrival_time.tm_min}")
            except EOFError:
                break


def remove_unnecessary_items(new_file_name: str, file_name: str) -> None:
```

```
    with open(file_name, 'rb') as file:
        file.seek(0)
        with open(new_file_name, 'wb') as new_file:
            new_file.seek(0)
            while True:
                try:
                    bus = load(file)
                    departure_time = get_time(bus['departure_time'])
                    duration_trip = get_time(bus['duration_trip'])
                    arrival_time, if_night = get_arrival_time(departure_time,
duration_trip)
                    if_del = if_night and duration_trip.tm_hour < 17
                    if not if_del:
                        dump(bus, new_file)

                except EOFError:
                    break
```

Результат виконання:

C++



Python:

```
C:\Users\Dima\source\PC\OP\venv\Scripts\python.exe C:/Users/Dima/source/PC/OP/Lab2/main.py
Select the write mode to the file (a/w)? a

Old information in file:

Destination: Kyiv
Is NIGHT!
Time of departure: 13:19
Time of travel: 20:16
Arrival time: 9:35

Destination: Lviv
Is NIGHT!
Time of departure: 5:14
Time of travel: 0:37
Arrival time: 5:51

Destination: Odesa
Time of departure: 14:15
Time of travel: 1:35
Arrival time: 15:50

Destination: Kriviy Rig
Is NIGHT!
Time of departure: 21:38
Time of travel: 16:36
Arrival time: 14:14


Enter how many buses you want to add: 1

Enter the destination: Kharkiv
Enter the time of departure in the format <hh:mm>: 12:50
Enter the travel time in the format <hh:mm>: 10:15

Information in the file without verification:

Destination: Kyiv
Is NIGHT!
Time of departure: 13:19
```

```
Information in the file without verification:

Destination: Kyiv
Is NIGHT!
Time of departure: 13:19
Time of travel: 20:16
Arrival time: 9:35

Destination: Lviv
Is NIGHT!
Time of departure: 5:14
Time of travel: 0:37
Arrival time: 5:51

Destination: Odesa
Time of departure: 14:15
Time of travel: 1:35
Arrival time: 15:50

Destination: Kriviy Rig
Is NIGHT!
Time of departure: 21:38
Time of travel: 16:36
Arrival time: 14:14

Destination: Kharkiv
Is NIGHT!
Time of departure: 12:50
Time of travel: 10:15
Arrival time: 23:5
```

```
Information in the file after removing extra items:

Destination: Kyiv
Is NIGHT!
Time of departure: 13:19
Time of travel: 20:16
Arrival time: 9:35

Destination: Odesa
Time of departure: 14:15
Time of travel: 1:35
Arrival time: 15:50

Process finished with exit code 0
```