

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.
Модульне програмування»

«Файли даних. Бінарні файли»

Варіант 32

Виконав студент ІІ-13 Шевцова Анастасія Андріївна
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №2

ФАЙЛИ ДАНИХ.

БІНАРНІ ФАЙЛИ

Варіант 32

Створити файл із розкладом руху приміських поїздів декількома напрямками (по кожному напрямку по 3-5 рейсів протягом дня): номер рейсу, напрямок руху, час відправлення, час прибуття в кінцевий пункт. Зі складеного розкладу сформувати зимовий розклад (новий файл), в якому мають бути лише ранкові (до 10:00) та вечірні (після 18:00) рейси.

C++

lab_2_cpp.cpp

```
#include "lab_2_cpp_func.h"

int main()
{
    setlocale(LC_ALL, "");

    string fileName = "file2c.txt";
    string newFileName = "newfile2c.txt";

    inputFile(fileName);
    makeNewFile(fileName, newFileName);
    cout << "\nOriginal file:" << endl;
    outputFile(fileName);
    cout << "\nNew file:" << endl;
    outputFile(newFileName);

    system("pause");
    return 0;
}
```

lab_2_cpp_func.cpp

```
char fileMode()
{
    char choice;
    cout << "Press [0] if you want to write the file from the beginning or [1] if you want to overwrite it: ";
    cin >> choice;
    while (choice != '0' && choice != '1') {
        cout << "Wrong! Please try again." << endl;
        cout << "Press [0] if you want to write the file from the beginning or [1] if you want to overwrite it: ";
        cin >> choice;
    }
    return choice;
}

void checkTime(int& hour, int& minute)
{
    while (hour < 0 || hour > 23 || minute < 0 || minute > 59) {
        cout << "Wrong! Please try again." << endl;
        cout << "Time: "; cin >> hour; cin.get(); cin >> minute;
    }
}
```

```

int numOfRoutes()
{
    cout << "Enter the number of the routes: ";
    int num; cin >> num;
    while (num < 1) {
        cout << "Wrong! Please try again." << endl;
        cout << "Enter the number of the routes: "; cin >> num;
    }
    return num;
}

void inputFile(string filename)
{
    char choice = fileMode();
    ofstream file;
    if (choice == '0') {
        file.open(filename, ios::binary);
    }
    else if (choice == '1') {
        file.open(filename, ios::binary | ios::app);
    }

    if (!file.is_open()) {
        cout << "Failed to open file." << endl;
    }

    else {
        int num = numOfRoutes();
        timetable train;

        for (int i = 0; i < num; i++) {
            cout << "Input " << i + 1 << " route." << endl;

            cout << "Route number: "; cin >> train.number;

            cin.get();
            cout << "Route: "; cin.getline(train.way, LEN_STR);

            cout << "Departure time: ";
            cin >> train.departure_h; cin.get(); cin >> train.departure_m;
            checkTime(train.departure_h, train.departure_m);

            cout << "Arrival time: ";
            cin >> train.arrival_h; cin.get(); cin >> train.arrival_m;
            checkTime(train.arrival_h, train.arrival_m);

            file.write((char*)&train, sizeof(timetable));
        }
        file.close();
    }
}

void printTime(int hm)
{
    (hm < 10) ? (cout << "0" << hm) : (cout << hm);
}

void outputFile(string filename)
{
    ifstream file;
    file.open(filename, ios::binary);

    if (!file.is_open()) {
        cout << "Failed to open file." << endl;
    }
    else {
        timetable train;
        while (file.read((char*)&train, sizeof(timetable))) {

```

```

        cout << "-----" << endl;
        cout << "Route number: " << train.number << endl;
        cout << "Route: " << train.way << endl;
        cout << "Departure time: ";
        printTime(train.departure_h); cout << ":";
printTime(train.departure_m); cout << endl;
        cout << "Arrival time: ";
        printTime(train.arrival_h); cout << ":";
printTime(train.arrival_m); cout << endl;
        cout << "-----" << endl;
    }
}

```

```

void makeNewFile(string filename, string newfilename)
{
    ifstream file;
    ofstream newfile;
    file.open(filename, ios::binary);
    newfile.open(newfilename, ios::binary);

    if (!file.is_open() && !newfile.is_open()) {
        cout << "Failed to open file." << endl;
    }
    else {
        timetable train;
        while (file.read((char*)&train, sizeof(timetable))) {
            if (train.departure_h < 10 || (train.departure_h == 10 &&
train.departure_m == 0) || train.departure_h >= 18) {
                newfile.write((char*)&train, sizeof(timetable));
            }
        }

        file.close();
        newfile.close();
    }
}

```

lab_2_cpp_func.h

```

#pragma once
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;
#define LEN_STR 100

struct timetable
{
    int number;
    char way[LEN_STR];
    int departure_h;
    int departure_m;
    int arrival_h;
    int arrival_m;
};

char fileMode();
void checkTime(int&, int&);
int numOfRoutes();
void inputFile(string);
void printTime(int);
void outputFile(string);
void makeNewFile(string, string);

```

Тестування:

```
Press [0] if you want to write the file from the beginning or [1] if you want to overwrite it: 0
Enter the number of the routes: 3
Input 1 route.
Route number: 1234
Route: jhgjhggc - hgvhjghg
Departure time: 09:00
Arrival time: 12:30
Input 2 route.
Route number: 4587
Route: gfcjgfhjgf - hgfcgfcgf
Departure time: 16:30
Arrival time: 19:00
Input 3 route.
Route number: 5695
Route: hgfcghf - gfgfhcg
Departure time: 19:15
Arrival time: 21:45

Original file:
-----
Route number: 1234
Route: jhgjhggc - hgvhjghg
Departure time: 09:00
Arrival time: 12:30
-----
Route number: 4587
Route: gfcjgfhjgf - hgfcgfcgf
Departure time: 16:30
Arrival time: 19:00
-----
Route number: 5695
Route: hgfcghf - gfgfhcg
Departure time: 19:15
Arrival time: 21:45
-----

New file:
-----
Route number: 1234
Route: jhgjhggc - hgvhjghg
Departure time: 09:00
Arrival time: 12:30
-----
Route number: 5695
Route: hgfcghf - gfgfhcg
Departure time: 19:15
Arrival time: 21:45
-----
```

Python

lab_2_python.py

```
from lab_2_py_functions import *

filename = 'file2p.txt'
newfilename = 'newfile2p.txt'

inputFile(filename)
makeNewFile(filename, newfilename)
print('\nOriginal file:')
outputFile(filename)
print('\nNew file:')
outputFile(newfilename)
```

lab_2_py_functions.py

```
import pickle

timetable = {
    "number": 0,
    "way": "",
    "departure_h": 0,
    "departure_m": 0,
    "arrival_h": 0,
    "arrival_m": 0
}

def fileMode():
    choice = input('Press [0] if you want to write the file from the beginning or [1] if you want to overwrite it: ')
    while choice != '0' and choice != '1':
        print('Wrong! Please try again.')
        choice = input('Press [0] if you want to write the file from the beginning or [1] if you want to overwrite it: ')
    return choice

def checkTime(hour, minute):
    while(hour < 0 or hour > 23 or minute < 0 or minute > 59):
        print('Wrong! Please try again.')
        time = input('Time: ')
        hour = int(time[:2])
        minute = int(time[3:])
    return hour, minute

def inputFile(filename):
    choice = fileMode()
    if choice == '0':
        file = open(filename, 'wb')
    elif choice == '1':
        file = open(filename, 'ab')

    num = int(input('Enter the number of the routes: '))
    while num < 1:
        print('Wrong! Please try again.')
        num = int(input('Enter the number of the routes: '))

    for i in range(num):
        timetable["number"] = int(input('Route number: '))

        timetable["way"] = input('Route: ')

        departure = input('Departure time: ')
        timetable["departure_h"] = int(departure[:2])
        timetable["departure_m"] = int(departure[3:])
        timetable["departure_h"], timetable["departure_m"] =
        checkTime(timetable["departure_h"], timetable["departure_m"])

        arrival = input('Arrival time: ')
        timetable["arrival_h"] = int(arrival[:2])
        timetable["arrival_m"] = int(arrival[3:])
        timetable["arrival_h"], timetable["arrival_m"] =
        checkTime(timetable["arrival_h"], timetable["arrival_m"])

        pickle.dump(timetable, file)
    file.close()

def printTime(hm):
    hmStr = str(hm)
    if hm < 10:
```

```
        return '0' + hmStr
    else:
        return hmStr

def outputFile(filename):
    file = open(filename, "rb")
    size = file.seek(0, 2)
    file.seek(0)

    while file.tell() < size:
        timetable = pickle.load(file)
        print("-----")
        print('Route number:', timetable["number"])
        print('Route:', timetable["way"])
        print('Departure time: ' + printTime(timetable["departure_h"]) + ':' +
printTime(timetable["departure_m"]))
        print('Arrival time: ' + printTime(timetable["arrival_h"]) + ':' +
printTime(timetable["arrival_m"]))
        print("-----")

    file.close()

def makeNewFile(filename, newfilename):
    file = open(filename, "rb")
    newfile = open(newfilename, "wb")
    size = file.seek(0, 2)
    file.seek(0)

    while file.tell() < size:
        timetable = pickle.load(file)
        if timetable["departure_h"] < 10 or (timetable["departure_h"] == 10 and
timetable["departure_m"] == 0) or timetable["departure_h"] >=18:
            pickle.dump(timetable, newfile)

    file.close()
    newfile.close()
```

Тестування:

```
Press [0] if you want to write the file from the beginning or [1] if you want to overwrite it: 0
Enter the number of the routes: 3
Route number: 1235
Route: jgfhghj - vghgvhg
Departure time: 08:25
Arrival time: 12:00
Route number: 2458
Route: jfcgfc - fcgfcgf
Departure time: 14:00
Arrival time: 16:00
Route number: 4956
Route: hgfhgfcgf - hgfchgfc
Departure time: 20:30
Arrival time: 22:00

Original file:
-----
Route number: 1235
Route: jgfhghj - vghgvhg
Departure time: 08:25
Arrival time: 12:00
-----
-----
Route number: 2458
Route: jfcgfc - fcgfcgf
Departure time: 14:00
Arrival time: 16:00
-----
-----
Route number: 4956
Route: hgfhgfcgf - hgfchgfc
Departure time: 20:30
Arrival time: 22:00
-----

New file:
-----
Route number: 1235
Route: jgfhghj - vghgvhg
Departure time: 08:25
Arrival time: 12:00
-----
-----
Route number: 4956
Route: hgfhgfcgf - hgfchgfc
Departure time: 20:30
Arrival time: 22:00
-----
Press any key to continue . . .
```