

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»

«Бінарні файли»

Варіант №32

Виконав студент ІП-14 Шляхтун Денис Михайлович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №2

Тема: бінарні файли.

Мета: вивчити особливості створення і обробки бінарних файлів.

Хід роботи

Задача. 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.

Постановка задачі. Для виконання задачі потрібно створити бінарний файл, що буде містити записи розкладу руху поїздів, введені користувачем. На основі цього файлу буде створений новий файл, який буде мати ранкові та вечірні записи з минулого файлу, тобто ті рейси, що відбуваються у проміжку між 18:00 та 10:00 наступного дня. Потім вміст цих двох файлів потрібно вивести в консоль. Ввідні дані, що вимагаються для виконання задачі: записи розкладу для першого файлу.

Випробування коду на C++.

Код:

Source.cpp

```
15     #include "lib.h"
16
17     int main()
18     {
19         string fileIn = "input.txt";    //назва файлу введених даних (№1)
20         string fileOut = "output.txt"; //назва кінцевого файлу (№2)
21         choice(fileIn);                 //введення даних у файл
22         fileOutput(fileIn);              //виведення файлу №1
23         fileCreate(fileIn, fileOut);     //створення файлу №2 відповідно до завдання
24         fileOutput(fileOut);             //виведення файлу №2
25         system("pause");
26     }
```

lib.h

```

1  #pragma once
2  #include <iostream>
3  #include <fstream>
4  #include <iomanip>
5  #include <string>
6  using namespace std;
7
8  //структура запису розкладу руху поїздів
9  struct TTimeTable
10 {
11     int number;          //номер рейсу
12     char direction[30];  //напрямок руху
13     int hourDep;         //година відправлення
14     int minuteDep;      //хвилина відправлення
15     int hourArr;        //година прибуття
16     int minuteArr;      //хвилина прибуття
17 };
18
19 void choice(string);      //вибір між створенням і доповненням файлу введених даних
20 void fileInput(ofstream&); //введення файлу
21 TTimeTable structInput(); //введення окремого запису розкладу руху
22 void fileOutput(string);  //виведення файлу
23 void fileCreate(string, string); //створення файлу відповідно до задачі
24 bool timeCheck(TTimeTable); //перевірка часу структури на відповідність умові задачі

```

lib.cpp

```

1      #include "lib.h"
2
3      //вибір між створенням і доповненням файлу введених даних
4      void choice(string fileName)
5      {
6          cout << "Enter 1 for new file or 2 to add to previous: ";
7          bool flag = true;
8          char s;
9          ofstream fileOut;    //відкриття потоку на запис файлу
10         while (flag)
11         {
12             flag = false;
13             cin >> s;
14             switch(s)
15             {
16                 case '1': //відкриття бінарного файлу
17                 {
18                     fileOut.open(fileName, ios::binary);
19                     fileInput(fileOut);
20                     break;
21                 }
22                 case '2': //відкриття бінарного файлу на дозапис в кінець файлу
23                 {
24                     fileOut.open(fileName, ios::binary | ios::app);
25                     fileInput(fileOut);
26                     break;
27                 }
28                 default: //обробка некоректного вводу
29                 {
30                     cout << "Wrong input. Please, try again: ";
31                     flag = true;
32                 }
33             }
34         }
35         fileOut.close();    //закриття потоку на запис файлу
36     }

```

```

38     //введення файлу
39     void fileInput(ofstream& fileOut)
40     {
41         char s;
42         bool flag = true;
43         while (flag)
44         {
45             TTimeTable data = structInput(); //введення окремого запису розкладу руху
46             fileOut.write((char*)&data, sizeof(TTimeTable)); //запис структури в файл
47             cout << "Enter + if you want to continue or any other symbol if don't: ";
48             cin >> s;
49             if (s != '+')
50                 flag = false;    //закінчення введення, якщо введений символ не плюс
51         }
52     }

```

```

54     //введення окремого запису розкладу руху
55     TTimeTable structInput()
56     {
57         cout << endl;
58         TTimeTable data;
59         string str;
60         cout << "Input the number of train: ";
61         cin >> data.number;
62         cin.ignore();
63         cout << "Input the direction of train: ";
64         cin >> data.direction;
65         cout << "Input time of departure in format \"12:34\": ";
66         cin.ignore();
67         cin >> str;
68         //переведення формату "12:34" в два окремих числа
69         data.hourDep = stoi(str.substr(0, str.find(':')));
70         data.minuteDep = stoi(str.substr(str.find(':') + 1, 2));
71         cout << "Input time of arrival in format \"12:34\": ";
72         cin.ignore();
73         cin >> str;
74         //переведення формату "12:34" в два окремих числа
75         data.hourArr = stoi(str.substr(0, str.find(':')));
76         data.minuteArr = stoi(str.substr(str.find(':') + 1, 2));
77
78         return data;
79     }

```

```

81     //виведення файлу
82     void fileOutput(string fileName)
83     {
84         TTimeTable data;
85         ifstream fileOut(fileName, ios::binary); //відкриття бінарного файлу на читання
86         int i = 1;
87         cout << "\nFile " << fileName << '\n';
88         //поструктурне зчитування файлу
89         while (fileOut.read((char*)&data, sizeof(TTimeTable)))
90         {
91             cout << "#" << setw(5) << left << i << right
92                 << setw(5) << data.number
93                 << setw(30) << data.direction << " "
94                 << setw(2) << setfill('0') << data.hourDep
95                 << ':' << setw(2) << data.minuteDep << " "
96                 << setw(2) << data.hourArr
97                 << ':' << setw(2) << data.minuteArr << setfill(' ') << "\n";
98             i++;
99         }
100         fileOut.close();
101     }

```

```

103 //створення файлу відповідно до задачі
104 void fileCreate(string fileInName, string fileOutName)
105 {
106     TTimeTable data;
107     ifstream fileIn(fileInName, ios::binary); //відкриття файлу введених файлів на читання
108     ofstream fileOut(fileOutName, ios::binary); //відкриття файлу результату на запис
109     while (fileIn.read((char*)&data, sizeof(TTimeTable)))
110     {
111         if (timeCheck(data)) //якщо виконується умова відповідно до задачі
112         {
113             fileOut.write((char*)&data, sizeof(TTimeTable)); //запис структури у файл результатів
114         }
115     }
116     fileIn.close();
117     fileOut.close();
118 }

```

```

120 //перевірка часу структури на відповідність умові задачі
121 bool timeCheck(TTimeTable data)
122 {
123     bool flag;
124     float dep = data.hourDep + float(data.minuteDep) / 100;
125     float arr = data.hourArr + float(data.minuteArr) / 100;
126     if (dep < arr) //якщо відправлення і прибуття відбуваються в межах одного дня
127         flag = (dep <= 10 && arr <= 10) || (dep >= 18 && arr >= 18);
128     else //якщо відправлення і прибуття відбуваються в різні дні
129         flag = dep >= 18 && arr <= 10;
130     return flag;
131 }

```

Результат:

```
E:\university\II semester\basics of prog\Lab2\CPPLab2\D...
Enter 1 for new file or 2 to add to previous: 2

Input the number of train: 6062
Input the direction of train: Lviv-Khodoriv
Input time of departure in format "12:34": 09:05
Input time of arrival in format "12:34": 11:18
Enter + if you want to continue or any other symbol if don't: +

Input the number of train: 6064
Input the direction of train: Lviv-Khodoriv
Input time of departure in format "12:34": 11:20
Input time of arrival in format "12:34": 13:21
Enter + if you want to continue or any other symbol if don't: -

File input.txt
#1      6009      Lviv-Truskavets  06:51  09:42
#2      6011      Lviv-Truskavets  13:36  16:12
#3      6013      Lviv-Truskavets  17:05  19:48
#4      6015      Lviv-Truskavets  20:22  22:56
#5      6019      Lviv-Sianky    07:42  12:09
#6      6021      Lviv-Sianky    09:25  13:50
#7      6023      Lviv-Sianky    12:40  17:26
#8      6025      Lviv-Sianky    17:40  22:45
#9      6027      Lviv-Sianky    20:47  01:56
#10     6062      Lviv-Khodoriv  09:05  11:18
#11     6064      Lviv-Khodoriv  11:20  13:21

File output.txt
#1      6009      Lviv-Truskavets  06:51  09:42
#2      6015      Lviv-Truskavets  20:22  22:56
#3      6027      Lviv-Sianky    20:47  01:56
Press any key to continue . . .
```

Випробування коду на Python.

Код:

source.py

```
import module
from class1 import TTimeTable

fileInName = "input.txt"    #назва файлу введених даних (№1)
fileOutName = "output.txt"  #назва кінцевого файлу (№2)
module.fileInput(fileInName, module.choice(fileInName)) #введення даних у файл
module.fileOutput(fileInName)                             #виведення файлу №1
module.fileCreate(fileInName, fileOutName)                 #створення файлу №2 відповідно до завдання
module.fileOutput(fileOutName)                             #виведення файлу №2
```

class1.py

```

#клас запису розкладу руху поїздів
class TTimeTable():
    def __init__(self,number,direction,hourDep,minuteDep,hourArr,minuteArr):
        self.number = number           #номер рейсу
        self.direction = direction      #напрямок руху
        self.hourDep = hourDep          #година відправлення
        self.minuteDep = minuteDep      #хвилина відправлення
        self.hourArr = hourArr          #година прибуття
        self.minuteArr = minuteArr     #хвилина прибуття

```

module.py

```

import pickle
from class1 import TTimeTable

#вибір між створенням і доповненням файлу введених даних
def choice(fileName):
    while True:
        s = input("Enter 1 for new file or 2 to add to previous: ")
        if s == '1':      #відкриття бінарного файлу на запис
            return "wb"
        elif s == '2':    #відкриття бінарного файлу на дозапис в кінець файлу
            return "ab"
        print("Wrong input. Try again.")

#введення окремого запису розкладу руху
def structInput():
    number = int(input("\nInput the number of train: "))
    direction = input("Input the direction of train: ")
    str = input("Input time of departure in format \"12:34\": ")
    #переведення формату "12:34" в два окремих числа
    hourDep = int(str[:str.find(':')])
    minuteDep = int(str[str.find(':')+1:])
    str = input("Input time of arrival in format \"12:34\": ")
    #переведення формату "12:34" в два окремих числа
    hourArr = int(str[:str.find(':')])
    minuteArr = int(str[str.find(':')+1:])
    return TTimeTable(number, direction, hourDep, minuteDep, hourArr, minuteArr)

```



```
#введення файлу
def fileInput(fileName, param):
    fileIn = open(fileName, param)
    flag = True
    while flag:
        #введення та запис окремого елемента розкладу в файл
        pickle.dump(structInput(), fileIn)
        if input("Enter + to continue or any other symbol to stop: ") != '+':
            flag = False #закінчення введення, якщо введений символ не плюс
    fileIn.close()

#зчитування файлу
def fileRead(fileName):
    fileOut = open(fileName, 'rb')
    list = []
    flag = True
    while flag:
        try:
            #покласове перенесення даних з файлу до списку
            list.append(pickle.load(fileOut))
        except EOFError:
            #доки не наступить кінець файлу
            flag = False
    fileOut.close()
    return list
```

```

#виведення файлу
def fileOutput(fileName):
    print("\nFile {}".format(fileName))
    i = 1
    for data in fileRead(fileName):
        print("#{:<5}{:<5}{:>30}  {:0>2}:{:0>2}  {:0>2}:{:0>2}".format(i,
            data.number, data.direction, data.hourDep, data.minuteDep,
            data.hourArr, data.minuteArr))
        i+=1

#створення файлу відповідно до задачі
def fileCreate(fileInName, fileOutName):
    fileOut = open(fileOutName, "wb") #відкриття файлу результату на запис
    for data in fileRead(fileInName): #зчитування файлу введених даних
        if timeCheck(data):          #якщо виконується умова відповідно до задачі
            pickle.dump(data, fileOut) #запис у файл результатів
    fileOut.close()

#перевірка часу на відповідність умові задачі
def timeCheck(data):
    dep = data.hourDep + float(data.minuteDep) / 100
    arr = data.hourArr + float(data.minuteArr) / 100
    if dep < arr: #якщо відправлення і прибуття відбуваються в межах одного дня
        flag = (dep <= 10 and arr <= 10) or (dep >= 18 and arr >= 18)
    else:        #якщо відправлення і прибуття відбуваються в різні дні
        flag = dep >= 18 and arr <= 10
    return flag

```

Результат:

```
C:\Program Files\WindowsApps\PythonSoftwar...  —  □  ×
Enter 1 for new file or 2 to add to previous: 2

Input the number of train: 6062
Input the direction of train: Lviv-Khodoriv
Input time of departure in format "12:34": 09:05
Input time of arrival in format "12:34": 11:18
Enter + to continue or any other symbol to stop: +

Input the number of train: 6064
Input the direction of train: Lviv-Khodoriv
Input time of departure in format "12:34": 11:20
Input time of arrival in format "12:34": 13:21
Enter + to continue or any other symbol to stop: -

File input.txt:
#1    6009          Lviv-Truskavets    06:51    09:42
#2    6011          Lviv-Truskavets    13:36    16:12
#3    6013          Lviv-Truskavets    17:05    19:48
#4    6015          Lviv-Truskavets    20:22    22:56
#5    6019          Lviv-Sianky       07:42    12:09
#6    6021          Lviv-Sianky       09:25    13:50
#7    6023          Lviv-Sianky       12:40    17:26
#8    6025          Lviv-Sianky       17:40    22:45
#9    6027          Lviv-Sianky       20:47    01:56
#10   6062          Lviv-Khodoriv     09:05    11:18
#11   6064          Lviv-Khodoriv     11:20    13:21

File output.txt:
#1    6009          Lviv-Truskavets    06:51    09:42
#2    6015          Lviv-Truskavets    20:22    22:56
#3    6027          Lviv-Sianky       20:47    01:56
Press any key to continue . . . █
```

Висновок: При виконанні лабораторної роботи було вивчено особливості створення і обробки бінарних файлів даних у мовах програмування C++ та Python. На основі набутих навичок було написано лабораторну роботу, при виконанні якої було використано засоби запису, зчитування та доповнення бінарних файлів.