

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 No 2 з дисципліни
«Основи програмування 2.
Модульне програмування»
«Файли даних. Бінарні файли»

Варіант 7

Виконав студент ІП-1407 Грицина Діана Русланівна (шифр, прізвище,
ім'я, по батькові)

Перевірів _____ (прізвище, ім'я, по
батькові)

Київ 2022

Лабораторна робота No 2

Тема: Файли даних. Бінарні файли.

Мета: Вивчити особливості створення і обробки бінарних файлів.

Задача:

7. Створити файл з 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. На основі даного розкладу сформувати зимовий розклад (новий файл), в якому мають бути тільки ранкові (до 10:00) та вечірні (після 18:00) рейси.

Постановка задачі: для реалізації перезапису файлу або ж додавання інформації у нього, потрібно передбачити вибір режиму відкриття файлу користувачем, якщо у ньому вже є дані. Для зберігання даних у бінарному файлі використаємо структури. Щоб записати інформацію про поїзд у новий файл, перевіримо час відправлення рейсу відповідно до умови.

Код на C++

main.cpp

```
#include <iostream>
#include <fstream>
#include "func.hpp"
using namespace std;
int main() {
    string filename = "/Users/dianagr/Documents/lab22/lab22/trains.dat";
    string filename2 = "/Users/dianagr/Documents/lab22/lab22/result.dat";
    create_file(filename); // запис інформації у файл
    output_file(filename); // виведення першого створеного файлу
    create_new_file(filename, filename2); // створення 2 файлу на основі даних, які були введені
    cout<<"\n\nWINTER LIST";
    output_file(filename2); // виведення результату
    return 0;
}
```

func.hpp

```
#ifndef func_hpp
#define func_hpp

#include <stdio.h>
#include <fstream>
using namespace std;
void create_file(string filename);
void output_file(string filename);
void create_new_file(string file1, string file2);
#endif /* func_hpp */
```

func.cpp

```
#include "func.hpp"
#include <iostream>
#include <fstream>
struct Trains{
    string day,
    number,
    departure,
    arrival,
    dep_time,
    arr_time;
};
ofstream get_file(string filename){
    ifstream read(filename);
    if(read.peek() == EOF) { //перевірка чи файл пустий
        read.close();
        ofstream file_b(filename, ios::binary);
        return file_b;
    }else{
        read.close();
        cout<<"There is already some information in the file, to add something enter a,
        to rewrite it press enter ";
        string add;
        getline(cin, add); //вибір відкрити файл в режимі дозапису чи перезаписати
        if(add=="a"){
            ofstream file_b(filename, ios::app);
            return file_b;
        }else{
            ofstream file_b(filename, ios::binary);
            return file_b;
        }
    }
};

}

void create_file(string filename){
    ofstream file_b = get_file(filename); //відкриття файлу в обраному режимі
    Trains train;
    string days[7] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
    "Sunday"};
    for(int i=0; i<7; i++){ //створення руху поїздів на кожний день
        string skip="";
        cout<<"To add trains to "<<days[i]<<" enter a, to skip press enter ";
        getline(cin, skip);
        if(!empty(skip)){ //перевірка чи додається інформація
            int num;
            cout<<"Amount ";
            cin>>num; //кількість поїздів в 1 день
            for(int j=0; j<num; j++){
                train.day = days[i];
                cin.ignore();
                cout<<"Enter train number ";
                getline(cin, train.number);
                cout<<"Enter departure ";
                getline(cin, train.departure);
                cout<<"Enter arrival ";
                getline(cin, train.arrival);
                cout<<"Enter departure time ";
                getline(cin, train.dep_time);
                cout<<"Enter arrival time ";
                getline(cin, train.arr_time);
                file_b.write((char*)&train, sizeof(Trains)); //запис інформації про 1 поїзд у
                файл
            }
        }
    }
    file_b.close();
}
```

```

void output_file(string filename){
    Trains train;
    cout<<"\n\nList of trains\n";
    string days[7] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
        "Sunday"};
    for(int i=0; i<7; i++){
        cout<<"\nTrains for "<<days[i]<<"\n";
        ifstream file_b(filename, ios::binary);
        while(file_b.read((char*)&train, sizeof(Trains))){
            if(train.day==days[i]){//сортування поїздів по дням тижня
                cout<<train.number<<" "<<train.departure<<"-"<<train.arrival<<" :
                    "<<train.dep_time<<" - "<<train.arr_time<<"\n";
            }
        }
        file_b.close();
    }
}

void create_new_file(string file1, string file2){
    Trains train;
    ifstream file_b1(file1, ios::binary);//відкриття для читання з файла
    ofstream file_b2(file2, ios::binary);//відкриття для запису в файл
    int time;
    string dep_time;
    while(file_b1.read((char*)&train, sizeof(Trains))){
        dep_time = train.dep_time;//отримання інформації про час відправлення
        time = stoi(dep_time.erase(2, string::npos));
        if(time<=10||time>=18){//перевірка чи час задовольняє умову завдання
            file_b2.write((char*)&train, sizeof(Trains));//додавання поїзда в новий файл
        }
    }
    file_b1.close();
    file_b2.close();
}

```

Результат:

```

To add trains to Monday enter a, to skip press enter a
Amount 1
Enter train number 788
Enter departure Odessa
Enter arrival Kyiv
Enter departure time 07:35
Enter arrival time 17:20
To add trains to Tuesday enter a, to skip press enter a
Amount 1
Enter train number 768
Enter departure Lviv
Enter arrival Ternopil
Enter departure time 14:30
Enter arrival time 17:00
To add trains to Wednesday enter a, to skip press enter
To add trains to Thursday enter a, to skip press enter a
Amount 2
Enter train number 567
Enter departure Kharkiv
Enter arrival Kyiv
Enter departure time 19:30
Enter arrival time 01:28

Enter train number 676
Enter departure Kherson
Enter arrival Odessa
Enter departure time 08:15
Enter arrival time 13:10
To add trains to Friday enter a, to skip press enter
To add trains to Saturday enter a, to skip press enter

```

```
To add trains to Sunday enter a, to skip press enter a
Amount 1
Enter train number 877
Enter departure Kyiv
Enter arrival Ternopil
Enter departure time 16:50
Enter arrival time 22:20
```

Файл 1:

```
List of trains

Trains for Monday
788 Odessa-Kyiv : 07:35 - 17:20

Trains for Tuesday
768 Lviv-Ternopil : 14:30 - 17:00

Trains for Wednesday

Trains for Thursday
567 Kharkiv-Kyiv : 19:30 - 01:28
676 Kherson-Odessa : 08:15 - 13:10

Trains for Friday

Trains for Saturday

Trains for Sunday
877 Kyiv-Ternopil : 16:50 - 22:20
```

Файл 2:

```
WINTER LIST

List of trains

Trains for Monday
788 Odessa-Kyiv : 07:35 - 17:20

Trains for Tuesday

Trains for Wednesday

Trains for Thursday
567 Kharkiv-Kyiv : 19:30 - 01:28
676 Kherson-Odessa : 08:15 - 13:10

Trains for Friday

Trains for Saturday

Trains for Sunday
Program ended with exit code: 0
```

Python: main.py

```
import trains

print()
trains.create_file("input.dat")#запис інформації у файл
trains.out_file("input.dat")#виведення першого створеного файлу
trains.new_file("input.dat", "result.dat")#створення 2 файлу на основі даних, які були введені
print()
print("WINTER LIST")
trains.out_file("result.dat")#виведення результату
```

trains.py

```
import pickle
import os
def create_file(filename):
    train_data = []
    days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
    if os.stat(filename).st_size == 0: f = open(filename, 'wb')#перевірка чи файл пустий
    else:
        #вибір відкрити файл в режимі дозапису чи перезаписати
        add = input("There is already some information in the file, to add something enter a, to rewrite it press enter ")
        if add: f = open(filename, 'ab')
        else: f = open(filename, 'wb')
    for day in days:#створення руху поїздів на кожний день
        add = input("To add trains for " + day + " press a, to skip press enter ")
        if add:#перевірка чи додається інформація
            num = int(input("Amount "))#кількість поїздів в 1 день
            for x in range(num):
                train_day = day
                number = input("Enter train number ")
                departure = input("Enter departure ")
                arrival = input("Enter arrival ")
                dep_time = input("Enter departure time ")
                arr_time = input("Enter arrival time ")
                train = [train_day, departure, arrival, dep_time, arr_time, number]
                train_data.append(train)#додавання інформації про рейс до усіх даних

    pickle.dump(train_data, f)#запис даних у файл
    f.close
```

```
def out_file(filename):
    print("\nList of trains\n")
    days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
    f = open(filename, 'rb')
    train_data = []
    while True:
        try:
            train_data.append(pickle.load(f))#завантаження усіх рейсів з файлу
        except EOFError:
            break
    f.close
    for day in days:
        print("Trains for " + day)
        for trains in train_data:
            for train in trains:
                if(day==train[0]):#виведення інформації по дням
                    print(train[5] + " " + train[1] + "-" + train[2] + " : " + train[3] + " - " + train[4])
    print()
```

```
def new_file(filename1, filename2):
    f1 = open(filename1, 'rb')
    train_data = []
    while True:
        try:
            train_data.append(pickle.load(f1))
        except EOFError:
            break
    f1.close
    f2 = open(filename2, 'wb')
    train_data2 = []
    for trains in train_data:
        for train in trains:
            dep_time = int(train[3][0:2])#час відправлення
            if dep_time<=10 or dep_time>=18:#перевірка чи час задовольняє умову завдання
                train_data2.append(train)#додавання поїзда в до усіх даних

    pickle.dump(train_data2, f2)#запис перевірених даних у новий файл
    f2.close
```

Результат:

```
There is already some information in the file, to add something enter a, to rewrite it press enter
To add trains for Monday press a, to skip press enter a
Amount 1
Enter train number 767
Enter departure Lviv
Enter arrival Ternopil
Enter departure time 07:20
Enter arrival time 10:05
To add trains for Tuesday press a, to skip press enter a
Amount 1
Enter train number 787
Enter departure Kyiv
Enter arrival Odessa
Enter departure time 17:40
Enter arrival time 23:50
To add trains for Wednesday press a, to skip press enter a
Amount 1
Enter train number 540
Enter departure Kharkiv
Enter arrival Kyiv
Enter departure time 06:40
Enter arrival time 13:35
To add trains for Thursday press a, to skip press enter
To add trains for Friday press a, to skip press enter
To add trains for Saturday press a, to skip press enter a
Amount 1
Enter train number 678
Enter departure Kyiv
Enter arrival Lviv
Enter departure time 18:20
Enter arrival time 02:30
To add trains for Sunday press a, to skip press enter
```

Файл 1:

```
List of trains

Trains for Monday
767 Lviv-Ternopil : 07:20 - 10:05

Trains for Tuesday
787 Kyiv-Odessa : 17:40 - 23:50

Trains for Wednesday
540 Kharkiv-Kyiv : 06:40 - 13:35

Trains for Thursday

Trains for Friday

Trains for Saturday
678 Kyiv-Lviv : 18:20 - 02:30

Trains for Sunday
```

Файл 2:

```
WINTER LIST

List of trains

Trains for Monday
767 Lviv-Ternopil : 07:20 - 10:05

Trains for Tuesday

Trains for Wednesday
540 Kharkiv-Kyiv : 06:40 - 13:35

Trains for Thursday

Trains for Friday

Trains for Saturday
678 Kyiv-Lviv : 18:20 - 02:30

Trains for Sunday
```


Висновок: у лабораторній роботі досліджено особливості створення і обробки бінарних файлів на мовах C++ та Python. Доступ до фізичних файлів здійснено за допомогою файлових змінних(об'єктів). Дані, введені користувачем, можуть бути записані у файл, або ж додані до вже існуючої інформації, в залежності від обраного режиму відкриття. Інформація про рейси зберігається у структурах. Програми доповнені комплексом функцій для обробки файлів відповідно до умови задачі.