# 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 № 1.2 з дисципліни

«Основи програмування»

Варіант 7

Виконав студент ІП-11 Головня Олександр Ростиславович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

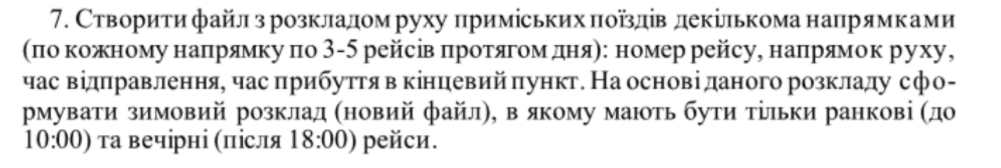
Київ 2021

Лабораторна робота No 1.2

**Тема:**

**Мета заняття:** Вивчити особливості створення і обробки бінарних файлів.

**Завдання:**



**Постановка задачі**

Створюємо перший файл у який записуємо усі рейси, для зручності використовуємо структуру.За певною математичною формулою вираховуємо які рейси нам потрібні для зимового розкладу, записуємо в інший файл.

(Є можливість вибору операції: чи новий файл, чи допис.

**Код на пайнтон**

**import pickle**

**def FirstFile(file1, oper):**

**count=int(input("Введіть кількість поїздів: "))**

**print("Формат часу: xx:xx ")**

**with open(file1,oper) as text:**

**for i in range(count):**

**direction=input("Введіть напрямок: ")**

**number=input("Введіть номер поїзда: ")**

**timeP=input("Введіть час відправлення: ")**

**timeK=input("Введіть час прибуття: ")**

**A=train(number,direction,timeP,timeK)**

**pickle.dump(A, text)**

**def SecondFile(array, file2):**

**with open(file2,'wb') as text:**

**min\_time=1000**

**max\_time=1800**

**for i in range (len(array)):**

**tp=int(array[i].timeP)**

**tk=int(array[i].timeK)**

**if ((tp <= min\_time and tk <= min\_time) or (tp >= max\_time and tk >= max\_time) or (tp >= max\_time and tk <= min\_time)):**

**pickle.dump(array[i], text)**

**class train:**

**def \_\_init\_\_(TRAIN, number=0, direction=None, timeP="00:00", timeK="00:00"):**

**TRAIN.number=number**

**TRAIN.direction=direction**

**TRAIN.timeP=timeP**

**TRAIN.timeK=timeK**

**def inf(TRAIN):**

**print("Номер поїзда", TRAIN.number, "напрямок",TRAIN.direction, ", відправляється о",TRAIN.timeP, "приїжджає о", TRAIN.timeK)**

**def Output(file):**

**array=[]**

**with open(file,'rb') as text:**

**while True:**

**try:**

**A=pickle.load(text)**

**except:**

**break**

**else:**

**array.append(A)**

**A.inf()**

**return array**

**print("print wb - new file!")**

**print("print ab - add to file!")**

**oper = input()**

**file1="Trains.bin"**

**file2="NewTrains.bin"**

**array=[]**

**FirstFile(file1,oper)**

**print("Літній розклад: ")**

**array=Output(file1)**

**SecondFile(array, file2)**

**print("Зімній розклад: ")**

**Output(file2)**

**Код на С++**

#include <iostream>

#include <stdlib.h>

#include <string.h>

#include <windows.h>

#include <string>

#include <fstream>

using namespace std;

const int n = 2;

struct TRAIN

{

char item[15];

int number;

int timeP;

int timeK;

};

void firstFile(int& count, TRAIN inf[n], int oper);

void Output(TRAIN inf[n], int j);

void SecondFile(TRAIN inf[n], int tp, int tk, int oper);

void OUTPUT(string name);

int operation();

ofstream Makefile(int oper, string name);

void main()

{

setlocale(LC\_ALL, "ru");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

TRAIN inf[n];

int count,oper;

char otv[5];

char zn[3] = "да";

count = 0;

int tp = 1000;

int tk = 1800;

oper = operation();

while (count < n)

{

firstFile(count, inf,oper);

}

cout << "Вивести iнформацiю про поїзда що введені?" << endl;

cout << "Вивести iнформацiю про поїзда що введені?" << endl;

cin >> otv;

if (\*otv == \*zn) {

for (int i = 0; i < n; i++)

{

Output(inf,i);

}

}

SecondFile(inf, tp, tk,oper);

//OUTPUT("Trains.dat");

cout << endl;

cout << "===================================" << endl;

//OUTPUT("NewTrains.dat");

}

void firstFile(int& count, TRAIN inf[n],int oper) {

ofstream fOut = Makefile(oper, "Trains.dat");

if (!fOut.is\_open()) {

cout << "ERROR!!!!!!!!!!!! File cannot opened!" << endl;

}

else

{

for (int i = 0; i < n; i++)

{

cout << "------ Информация для " << (1 + count++) << " поезда ----------------------" << endl;

cout << "Введіть пункт призначення : ";

cin >> (&inf[i])->item;

cout << "Введіть номер поїзда : ";

cin >> (&inf[i])->number;

cout << "Введіть час відправлення ( ЧЧ : ММ ) : ";

cin >> (&inf[i])->timeP;

cout << "Введіть час прибуття ( ЧЧ : ММ ) : ";

cin >> (&inf[i])->timeK;

cout << "----------------------------" << endl;

fOut.write((char\*)&inf[i], sizeof(TRAIN));

}

}

fOut.close();

}

void Output(TRAIN inf[n], int j) {

cout << "----------------------------" << endl;

cout << "Пункт призначення : " << inf[j].item << endl;

cout << "Номер поїзда : " << inf[j].number << endl;

cout << "Час відправлення ( ЧЧ : ММ ): " << inf[j].timeP << endl;

cout << "Час прибуття ( ЧЧ : ММ ): " << inf[j].timeK << endl;

cout << "----------------------------" << endl;

}

void SecondFile(TRAIN inf[n], int tp, int tk,int oper) {

ofstream fOut = Makefile(oper, "NewTrains.dat");

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

if (inf[i].timeP <= tp && inf[i].timeK <= tp || inf[i].timeP >= tk && inf[i].timeK >= tk || inf[i].timeP >= tk && inf[i].timeK <= tp) {

if (!fOut.is\_open()) {

cout << "ERROR!!!!!!!!!!!! File cannot opened!" << endl;

}

else

{

cout << "-----------ДОБАВЛЕНО В НОВИЙ РОЗКЛАД-----------------" << endl;

cout << "Пункт призначення : " << inf[i].item << endl;

cout << "Номер поїзда : " << inf[i].number << endl;

cout << "Час відправлення ( ЧЧ : ММ ): " << inf[i].timeP << endl;

cout << "Час прибуття ( ЧЧ : ММ ): " << inf[i].timeK << endl;

cout << "----------------------------" << endl;

fOut.write((char\*)&inf[i], sizeof(TRAIN));

}

}

}

fOut.close();

}

void OUTPUT(string name) {

fstream fs;

fs.open(name, ios::app | ios::out | ios::in | ios::binary);

while (!fs.eof())

{

char msg;

fs.read((char\*)&msg, sizeof msg);

cout << msg;

}

}

ofstream Makefile(int oper, string name) {

if (oper == 1) {

ofstream file(name, ios::out | ios::binary); //ств пустого файлу

return file;

}

else if (oper == 2) {

ofstream file(name, ios::app | ios::binary); //Доповнити файл

return file;

}

}

int operation() {

int oper; //Вибір операції

cout << "1. NEW" << endl;

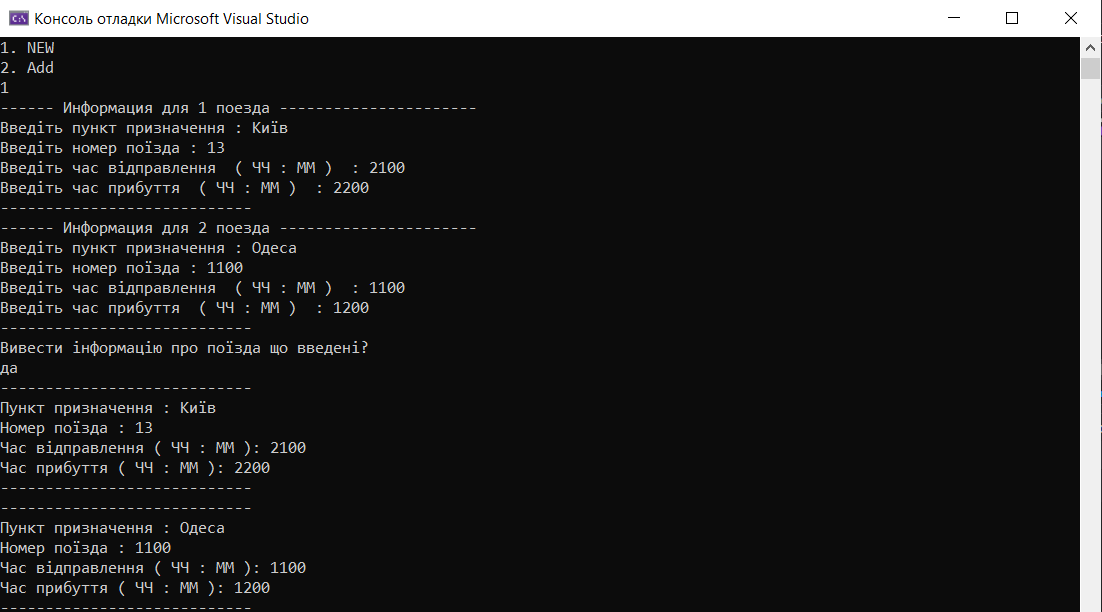
cout << "2. Add" << endl;

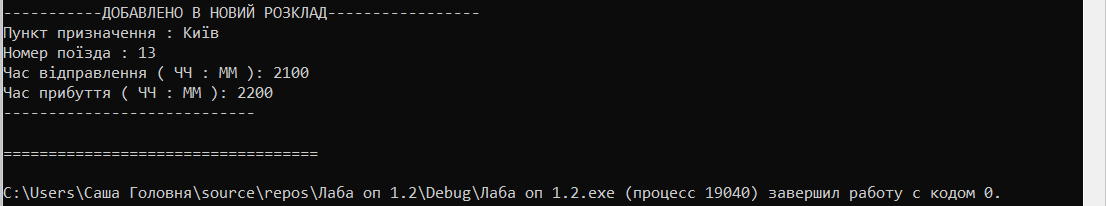
cin >> oper;

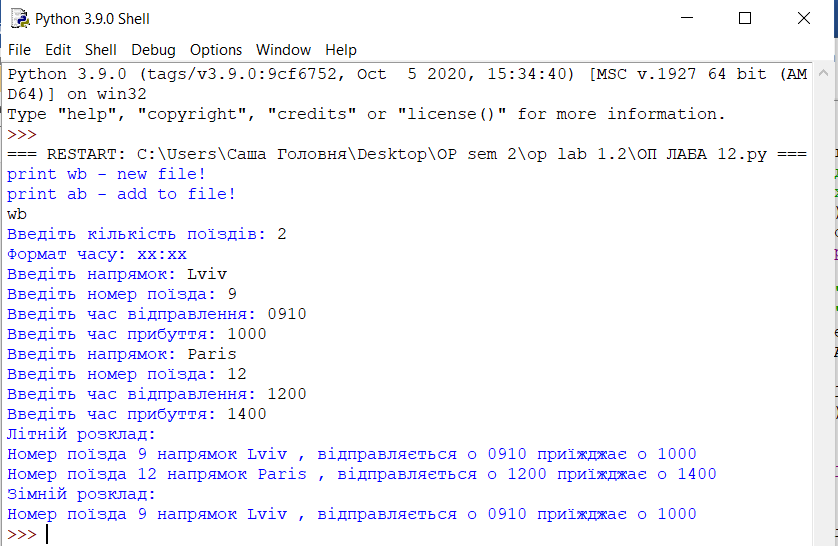
return oper;

}

**Результати**







**Висновок:** Ми вивчили особливості створення бінарних файлів, навчилися записувати та считувати. Працюючи над лабораторною, вивчили особливості обробки бінарних файлів.