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 – 3.

Методології програмування»

«Файли даних»

Варіант 32

Виконав студент ІП-11 Фукс Вікторія Ігорівна

Перевірив \_\_\_\_\_Вітковська Ірина Павлівна\_\_\_\_\_\_\_\_\_

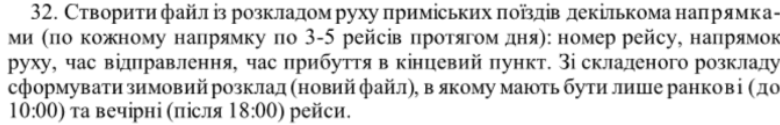
Київ 2022

**Лабораторна робота 2**

**ФАЙЛИ ДАНИХ**

**Мета** – вивчити особливості створення і обробки текстових файлів даних.

**Постановка задачі**:

****

**Код:**

**C#**

InfoWorker.cs:

using System;  
using System.Collections.Generic;  
using System.Linq;  
  
namespace csharp\_lab2  
{  
 internal class InfoWorker  
 {  
 public Info[] GetWinterTrips(Info[] wholeSchedule)  
 {  
 return wholeSchedule.Where(t => t.Departure.Hour < 10 || t.Departure.Hour > 18).ToArray();  
 }  
 public Info[] GetInput()  
 {  
 Console.Write("Enter number of trips:");  
 int count = int.Parse(Console.ReadLine());  
 List<Info> input = new List<Info>(count);  
 *//if (count >= 3 && count <= 5)* {  
 for (int i = 0; i < count; i++)  
 {  
 Console.Write("\nDestination: ");  
 string destination = Console.ReadLine();  
 string[] depart, arriv;  
 do  
 {  
 Console.Write("Departure time (HH:MM): ");  
 depart = Console.ReadLine().Split(':');  
 } while (!(depart.Length == 2 && int.TryParse(depart[0], out \_) && int.TryParse(depart[1], out \_) &&  
 int.Parse(depart[0]) <= 24 && int.Parse(depart[1]) <= 59));  
  
 do  
 {  
 Console.Write("Arrival time(HH:MM): ");  
 arriv = Console.ReadLine().Split(':');  
 } while (!(arriv.Length == 2 && int.TryParse(arriv[0], out \_) && int.TryParse(arriv[1], out \_) &&  
 int.Parse(arriv[0]) <= 24 && int.Parse(arriv[1]) <= 59));  
  
 input.Add(new Info(destination, (int.Parse(depart[0]), int.Parse(depart[1])),  
 (int.Parse(arriv[0]), int.Parse(arriv[1]))));  
 }  
 }  
 return input.ToArray();  
 }  
  
 public string InfoToString(Info inf)  
 {  
 return $"Destination:{inf.Destination}\n" +  
 $"Departure time:{string.Format("{0,0:D2}", inf.Departure.Hour)}:{string.Format("{0,0:D2}", inf.Departure.Minute)}\n" +  
 $"Arrival time:{string.Format("{0,0:D2}", inf.Arrival.Hour)}:{string.Format("{0,0:D2}", inf.Arrival.Minute)}\n";  
 }  
  
 public string InfoArrToString(Info[] inf)  
 {  
 string res = "";  
 foreach (var i in inf)  
 {  
 res += InfoToString(i) + "\n";  
 }  
  
 return res;  
 }  
 }  
}

Program.cs:

using System;  
  
namespace csharp\_lab2  
{  
 internal class Program  
 {  
 static void Main(string[] args)  
 {  
 string path1 = @"file1";  
 string path2 = @"file2";  
  
 FileWorker fileWorker = new FileWorker();  
 InfoWorker infoWorker = new InfoWorker();  
  
 fileWorker.CreateFile(path1, infoWorker.InfoArrToString(infoWorker.GetInput()));  
 Console.WriteLine("\nFile 1: ");  
 fileWorker.OutputFile(path1);  
 Info[] fromFile = fileWorker.StringToInfo(path1);  
 var winterTrips = infoWorker.GetWinterTrips(fromFile);  
 fileWorker.CreateFile(path2, infoWorker.InfoArrToString(winterTrips));  
  
 Console.WriteLine("File 2:\t[WINTER SCHEDULE]");  
 fileWorker.OutputFile(path2);  
  
  
 Console.ReadLine();  
 Console.ReadLine();  
 }  
 }  
  
}

double minPersent = text[0].CheckPersent();  
 int minInd = 0;  
   
 for (int i = 0; i < numOfText; i++)  
 {  
 if (minPersent > text[i].CheckPersent())  
 {  
 minPersent = text[i].CheckPersent();  
 minInd = i;  
 }  
 }  
 Console.WriteLine($"Text with smallest persent {minInd+1}");  
 }  
 }  
}

FileWorker.cs:

using System;  
using System.Collections.Generic;  
using System.Text;  
using System.IO;  
using csharp\_lab2;  
  
internal class FileWorker  
 {  
 public void CreateFile(string path, string text) *//створення файлу* {  
 FileStream fileStream = new FileStream(path, FileMode.**Create**);  
 byte[] arr = Encoding.Default.GetBytes(text);  
 fileStream.Write(arr, 0, arr.Length);  
 fileStream.Close();  
 }  
  
 public string ReadFile(string path) *//зчитування з файлу* {  
 FileStream fileStream = new FileStream(path, FileMode.**Open**);  
 byte[] arr = new byte[fileStream.Length];  
 fileStream.Read(arr, 0, arr.Length);  
 string textFromFile = Encoding.Default.GetString(arr);  
 fileStream.Close();  
 return textFromFile;  
 }  
  
 public Info[] StringToInfo(string path) *// текст файлу* {  
 string[] fromFile = ReadFile(path).Split("\n", StringSplitOptions.**RemoveEmptyEntries**);  
 List<Info> list = new List<Info>();  
 string dest = "";  
 (int, int) depart = (0, 0), arrive = (0, 0);  
 for (int i = 0; i < fromFile.Length; i++)  
 {  
 string[] splitted = fromFile[i].Split(':', StringSplitOptions.**RemoveEmptyEntries**);  
 if ((i + 1) % 3 == 1)  
 dest = splitted[1];  
 else if ((i + 1) % 3 == 2)  
 depart = (Convert.ToInt32(splitted[1]), Convert.ToInt32(splitted[2]));  
 else  
 {  
 arrive = (Convert.ToInt32(splitted[1]), Convert.ToInt32(splitted[2]));  
 list.Add(new Info(dest, depart, arrive));  
 }  
 }  
  
 return list.ToArray();  
 }  
  
 public void OutputFile(string path) => Console.WriteLine(ReadFile(path)); *//вивід з файлу* public void AppendFile(string pathToFile, string text) *//доповнення файлу* {  
 FileStream fileStream = new FileStream(pathToFile, FileMode.**Append**);  
 byte[] arr = Encoding.Default.GetBytes(text + "\n");  
 fileStream.Write(arr, 0, arr.Length);  
 fileStream.Close();  
 }  
 }

Info.cs

internal struct Info  
{  
 public string Destination;  
 public (int Hour, int Minute) Departure;  
 public (int Hour, int Minute) Arrival;  
  
 public Info(string dist, (int, int) depart, (int, int) arriv)  
 {  
 Destination = dist;  
 Departure = depart;  
 Arrival = arriv;  
 }  
}

**Python:**

main.py

from FileWorker import\*  
  
def main():  
 path1 = "D:\\code\\C#\\Лабы\\second\\py\\1";  
 path2 = "D:\\code\\C#\\Лабы\\second\\py\\2";  
  
 create\_file(path1)  
 print("\nTrips: ")  
 read\_file(path1)  
 print("\nWinter Schedule: ")  
  
 winter\_schedule(path1, path2)  
 read\_file(path2)  
  
  
main()

FileWorker.py

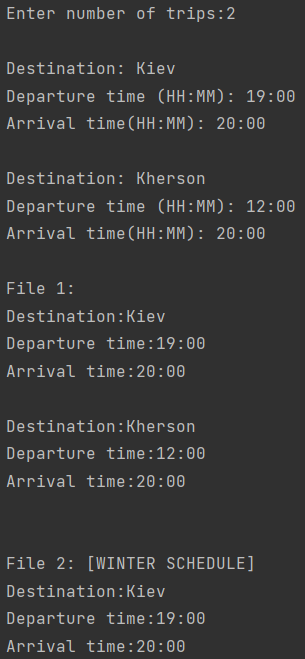
from itertools import count  
  
from Trips import Trips  
import pickle  
  
  
def get\_info(name):  
 with open(name, "rb") as file\_read:  
 text = pickle.load(file\_read)  
 return text  
  
  
def read\_file(name):  
 file\_read = open(name, "rb")  
 text = pickle.load(file\_read)  
 for trip in text:  
 trip.write()  
  
  
def create\_file(name):  
 write\_file = open(name, "wb")  
 count = int(input("Enter number of trips: "))  
 text = []  
 if count >= 3 and count <=5:  
 for i in range(count):  
 trips = Trips()  
 trips.number = str(i + 1)  
 print(f"\nTrip number: {trips.number}")  
 trip\_dest = str(input("Destination: "))  
 trips.destination = trip\_dest  
 dep\_time = str(input("Departure time (HH:MM): "))  
 if dep\_time[:2] < "0" or dep\_time[:2] > "24" or dep\_time[3:] < "0" or dep\_time[3:] > "59":  
 print("\n!!!Wrong time!!!")  
 exit(0)  
 else:  
 trips.departure = dep\_time  
 arrival\_time = str(input("Arrival time (HH:MM): "))  
 if arrival\_time[:2] < "0" or arrival\_time[:2] > "24" or arrival\_time[3:] < "0" or arrival\_time[3:] > "59":  
 print("\n!!!Wrong time!!!")  
 exit(0)  
 else:  
 trips.arrival = arrival\_time  
 text.append(trips)  
 pickle.dump(text, write\_file)  
  
  
def winter\_schedule(name1, name2):  
 file\_read = open(name1, "rb")  
 file\_write = open(name2, "wb")  
  
 text\_read = pickle.load(file\_read)  
 text\_write = []  
 for trip in text\_read:  
 if trip.departure < "10" or trip.departure > "18":  
 text\_write.append(trip)  
 pickle.dump(text\_write, file\_write)

Trips.py

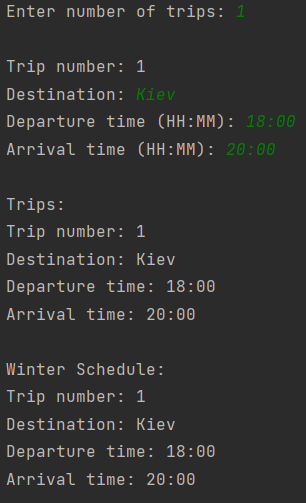
class Trips:  
 def \_\_init\_\_(self, number="", destination="", departure="", arrival=""):  
 self.number = number  
 self.destination = destination  
 self.departure = departure  
 self.arrival = arrival  
  
 def write(self):  
 print("Trip number: " + self.number +  
 "\nDestination: " + self.destination +  
 "\nDeparture time: " + self.departure +  
 "\nArrival time: " + self.arrival + "\n")

**Випробовування алгоритму:**

**C#:**

****

**Python:**

****

**Висновок:**

Ми вивчили особливості створення і обробки текстових файлів даних.

У результаті лабораторної роботи ми написали код на двух мовах програмування С# та Python. Зробили випробування алгоритму.