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.

Методології програмування»

«Бінарні файли»

Варіант 5

Виконав студент ІП-11 Гіжицький Даниїл Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2022

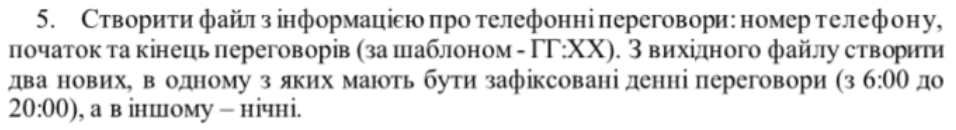
**Лабораторна робота №2**

**Бінарні файли**

**Мета**

**Вивчити можливість створення та обробуи бінарних файлів**

**Завдання(варіант 3):**



**Постановка задачі**

Для виконання поставленої задачі створимо допоміжні файли (та структуру на С#) з методами/функціями для роботи з файлами, текстом та консольним вводом.

**Програма на С#**

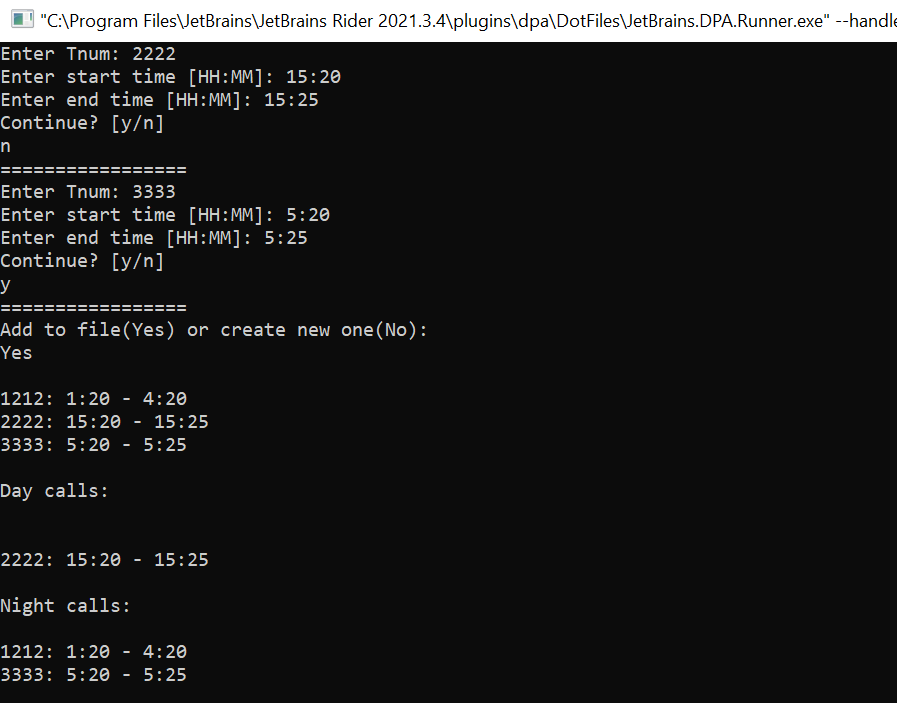
Файл 1:

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.IO;  
  
namespace OP2.\_2  
{  
 class Program  
 {  
   
 static void Main(string[] args)  
 {  
 Worker worker = new Worker();  
 string path1 = @"C:\Users\HP\source\repos\OPlab2.2\Tnum";  
 string path2 = @"C:\Users\HP\source\repos\OPlab2.2\day";  
 string path3 = @"C:\Users\HP\source\repos\OPlab2.2\night";  
 var calls = worker.TakeCalls();  
 var (day, night)=worker.DayOrNightCalls(calls);  
 worker.AppendOrCreate(path1, path2, path3);  
 worker.AppendToFile(path1, worker.PBookToStr(calls));  
 worker.PrintFile(path1);  
 worker.AppendToFile(path2, day);  
 worker.AppendToFile(path3, night);  
 Console.WriteLine("Day calls:");  
 worker.PrintFile(path2);  
 Console.WriteLine("Night calls:");  
 worker.PrintFile(path3);  
 Console.ReadKey();  
 }  
 }  
  
}

Файл 2:

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.IO;  
using System.Globalization;  
  
namespace OP2.\_2  
{  
 class CallInfo  
 {  
 public string Caller;  
 public (TimeOnly start, TimeOnly end) CallTime;  
   
 public CallInfo(string caller, TimeOnly start, TimeOnly end)  
 {  
 Caller = caller;  
 CallTime = (start, end);  
 }  
 public override string ToString()  
 {  
 return $"{Caller}: {CallTime.start} - {CallTime.end}";  
 }  
 }  
 internal class Worker  
 {  
 public void CreateFile(string path)  
 {  
 FileStream fstream = new FileStream(path, FileMode.**Create**);  
  
 fstream.Close();  
 }  
 public void AppendFile(string path)  
 {  
 FileStream fstream = new FileStream(path, FileMode.**OpenOrCreate**);  
  
 fstream.Close();  
 }  
 public void AppendToFile(string path, string calls)  
 {  
 FileStream fstream = new FileStream(path, FileMode.**Append**);  
 byte[] buffer;  
 buffer = System.Text.Encoding.Default.GetBytes('\n' + calls);  
 fstream.Write(buffer);  
 fstream.Close();  
 }  
  
 public List<CallInfo> TakeCalls()  
 {  
 List<CallInfo> phoneBook = new List<CallInfo>();  
 string exit;  
 do  
 {  
 Console.Write("Enter Tnum: ");  
 string caller = Console.ReadLine();  
 Console.Write("Enter start time [HH:MM]: ");  
 TimeOnly start = TimeOnly.Parse(Console.ReadLine());  
 Console.Write("Enter end time [HH:MM]: ");  
 TimeOnly end = TimeOnly.Parse(Console.ReadLine());  
 phoneBook.Add(new CallInfo(caller, start, end));  
 Console.WriteLine("Continue? [y/n]");  
 exit = Console.ReadLine();  
 Console.WriteLine("=================");  
 }  
 while(exit != "y");  
 return phoneBook;  
   
 }  
  
 public string PBookToStr(List<CallInfo> phoneBook)  
 {  
 List<string> calls = new List<string>();  
 foreach (var call in phoneBook)  
 {  
 calls.Add(call.ToString());  
 }  
 return string.Join('\n', calls);  
 }  
  
 private string ReadFile(string path)  
 {  
 FileStream fstream = new FileStream(path, FileMode.**Open**);  
 byte[] pnum = new byte[fstream.Length];  
 fstream.Read(pnum, 0, pnum.Length);  
 string phoneNum = Encoding.Default.GetString(pnum) + '\n';  
 fstream.Close();  
 return phoneNum;  
 }  
  
 public void PrintFile(string path)  
 {  
 Console.WriteLine(ReadFile(path));  
 }  
  
 public (string, string) DayOrNightCalls(List<CallInfo> phoneBook)  
 {  
 List<string> dayCalls = new List<string>();  
 List<string> nightCalls = new List<string>();  
 foreach (var call in phoneBook)  
 {  
 if (call.CallTime.end.Hour >= 6 && call.CallTime.end.Hour <= 20)  
 {  
 dayCalls.Add(call.ToString());  
 }  
 else  
 {  
 nightCalls.Add(call.ToString());  
 }  
   
 }  
  
 return (string.Join('\n', dayCalls), string.Join('\n', nightCalls));  
 }  
  
 public void AppendOrCreate(string path, string path2, string path3)  
 {  
 Console.WriteLine("Add to file(Yes) or create new one(No):");  
 bool add = false;  
 do  
 {  
 string answer = Console.ReadLine();  
 if (answer == "Yes")  
 {  
 AppendFile(path);  
 AppendFile(path2);  
 AppendFile(path3);  
 add = true;  
 }  
 else if (answer == "No")  
 {  
 CreateFile(path);  
 CreateFile(path2);  
 CreateFile(path3);  
 add = true;  
 }  
 } while (add == false);  
 }  
 }  
}

**Виконання на C#**

****

**Програма на python**

Файл 1:

import Worker  
  
  
def main():  
 path1 = "C:\\Users\\HP\\Desktop\\OP2.2p\\calls"  
 path2 = "C:\\Users\\HP\\Desktop\\OP2.2p\\daycalls"  
 path3 = "C:\\Users\\HP\\Desktop\\OP2.2p\\nightcalls"  
 answer = str(input("Add to file(Yes) or create new one(No):"))  
 Worker.append\_or\_create(path1, path2, path3, answer)  
 calls = Worker.take\_calls()  
 Worker.append\_to\_file(path1, Worker.list\_to\_str(calls))  
 Worker.print\_file(path1)  
 day, night = Worker.day\_or\_night(calls)  
 Worker.append\_to\_file(path2, day)  
 Worker.append\_to\_file(path3, night)  
 print("Day calls:")  
 Worker.print\_file(path2)  
 print("Night calls:")  
 Worker.print\_file(path3)  
  
main()

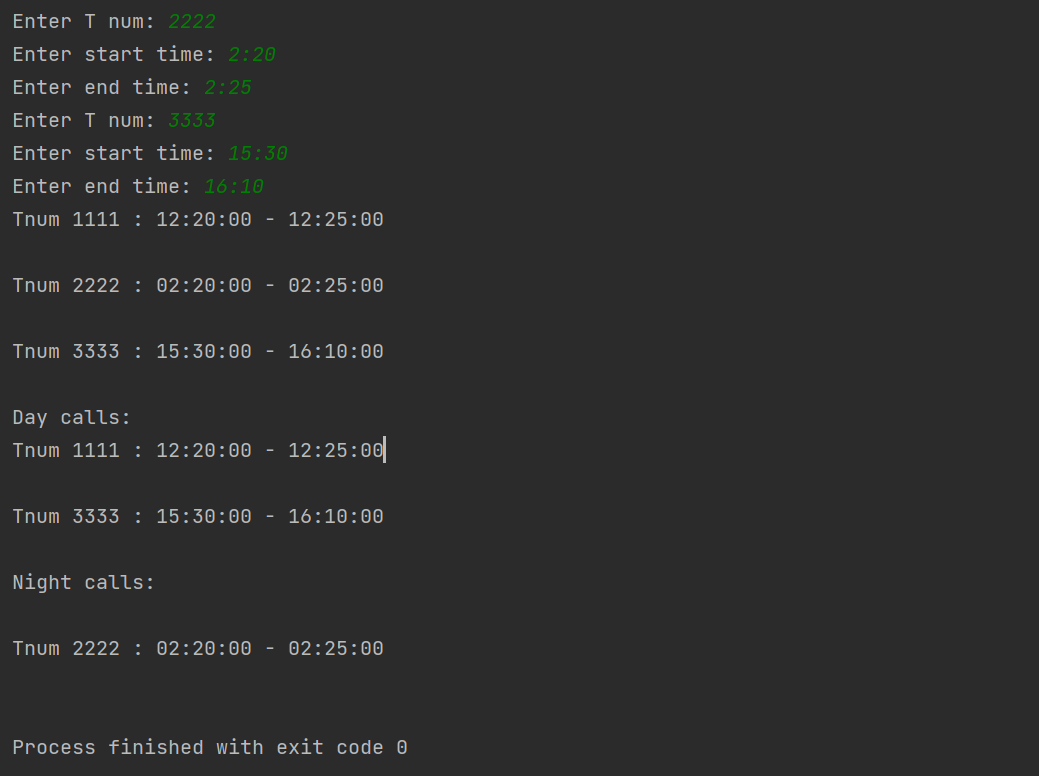
Файл 2:

import datetime  
from datetime import timedelta  
from Caller import call\_info  
  
  
def create\_file(path):  
 file\_hander = open(path, "wb")  
 file\_hander.close()  
  
  
def append\_to\_file(path, text: str):  
 with open(path, "ab") as file:  
 calls = str.encode(text)  
 file.write(calls)  
  
  
def read\_file(path) -> str:  
 calls = []  
 with open(path, "rb") as file:  
 calls = file.read()  
 return calls.decode()  
  
  
def print\_file(path):  
 print(read\_file(path))  
  
  
def take\_calls() -> list[call\_info]:  
 count\_calls = int(input("Enter num of calls: "))  
 calls\_and\_time = {}  
 calls = []  
 for i in range(count\_calls):  
 call = call\_info(  
 input("Enter T num: "),  
 datetime.datetime.strptime(input("Enter start time: "), "%H:%M").time(),  
 datetime.datetime.strptime(input("Enter end time: "), "%H:%M").time(),  
 )  
 calls.append(call)  
  
 return calls  
  
  
def list\_to\_str(calls: list):  
 calls\_str = []  
 for call in calls:  
 calls\_str.append(str(call))  
 return '\n'.join(calls\_str)  
  
  
def day\_or\_night(calls: list):  
 day = []  
 night = []  
 for call in calls:  
 if 6 <= call.start.hour <= 20:  
 day.append(call)  
  
 else:  
 night.append(call)  
 return '\n'.join([str(x) for x in day]), '\n'.join([str(x) for x in night])  
  
  
def append\_or\_create(path1, path2, path3, answer):  
 if answer == "Yes":  
 append\_to\_file(path1, '\n')  
 append\_to\_file(path2, '\n')  
 append\_to\_file(path3, '\n')  
 else:  
 create\_file(path1)  
 create\_file(path2)  
 create\_file(path3)

Файл 3

import datetime  
from datetime import timedelta  
class call\_info:  
 def \_\_init\_\_(self, caller: str, start: datetime, end: datetime):  
 self.caller = caller  
 self.start = start  
 self.end = end  
 def \_\_str\_\_(self):  
 return (f'Tnum {self.caller} : {self.start} - {self.end}\n')

**Виконання на python**



**Висновок:**

На лабораторній роботі я працював з технологію роботи з бінарними файлами. Програмного їх створення, запису та зчитування тексту з них, поставлене завдання було виконано на мовах C# та python.