

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 – 2. Методології програмування»

Варіант 23

Виконав студент

ІП-15, Мочалов Дмитро Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірив

Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Завдання

23. Створити файл із списком клієнтів, обслужених менеджером протягом дня: прізвище, час приходу та час закінчення обслуговування. При введенні даних перевіряти їхню допустимість (чи не перетинаються клієнти з наявними). Створити новий файл, який містить інформацію про клієнтів, з якими менеджер спілкувався понад 30 хв.

Код C#

```
using System;
using System.Collections.Generic;
using System.Diagnostics.Contracts;
using static Lab2dotnet.functions;

namespace Lab2dotnet
{
    internal class Program
    {
        public static void Main(string[] args)
        {
            List<string[]> listOfClients = InputText();
            string path = "file1.dat";
            CreateAndWriteFile(path, listOfClients);
            string textFromFirstFile = ReadFile(path);
            Console.WriteLine("\nТекст з першого файлу:");
            Console.WriteLine(textFromFirstFile);
            List<string[]> resultList = FindSpecialClients(listOfClients);
            if (resultList.Count != 0)
            {
                string path2 = "resultFile.dat";
                CreateAndWriteFile(path2, resultList);
                Console.WriteLine("\nТекст з другого файлу:");
                string textFromResultFile = ReadFile(path2);
                Console.WriteLine(textFromResultFile);
            }
            else Console.WriteLine("\nНемає клієнтів з тривалістю прийому від 30 хвилин.");
        }
    }
}
```

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Text.RegularExpressions;

namespace Lab2dotnet
{
    [1 usage]
    public static class Functions
    {
        [1 usage]
        public static List<string[]> InputText()
        {
            List<string[]> list = new List<string[]>();
            Console.WriteLine("Якщо хочете закінчити ввід - натисніть delete, інакше будь-яку іншу клавішу.");
            do
            {
                Console.WriteLine("Введіть прізвище:");
                var a:ConsoleKeyInfo = Console.ReadKey();
                if (a.Key == ConsoleKey.Delete) break;

                string[] person = new string[3];

                person[0] = a.KeyChar + Console.ReadLine();
                do
                {
                    Console.WriteLine("Введіть час приходу в форматі (HH:mm):");
                    do
                    {
                        person[1] = Console.ReadLine();
                        if (ValidTime(person[1])) break;

                        Console.WriteLine("Ви ввели час в неправильному форматі, або недопустимі значення. Будь ласка введіть коректний час:");
                    } while (true);

                    Console.WriteLine("Введіть час закінчення обслуговування в форматі (HH:mm):");
                    do
                    {
                        person[2] = Console.ReadLine();

```

```

                        do
                        {
                            person[2] = Console.ReadLine();
                            if (ValidTime(person[2]) && isAdmissible(person[1], person[2])) break;

                            Console.WriteLine("Ви ввели час в неправильному форматі, або недопустимі значення. Будь ласка введіть коректний час:");
                        } while (true);

                        if (!isCrossing(list, time_start: person[1], time_end: person[2])) break;

                        Console.WriteLine("Час прийому який ви ввели неможливий, оскільки пересікається з вже існуючими клієнтами, спробуйте ще раз.");
                    } while (true);

                    list.Add(person);
                } while (true);

                return list;
            }

```

```

public static bool isAdmissible(string time1, string time2)
{
    DateTime time_start = DateTime.Parse(time1);
    DateTime time_end = DateTime.Parse(time2);
    if (time_end < time_start) return false;
    return true;
}

```

2 usages

```

public static bool ValidTime(string time)
{
    Regex regex = new Regex(pattern: @"^b((09|1[0-7]):([0-5][0-9]\b))|18:00\b");
    if (regex.IsMatch(time))
    {
        return true;
    }
    return false;
}

```

```

public static bool isCrossing(List<string[]> list, string time_start, string time_end)
{
    if (list.Count != 0)
    {
        DateTime date_start1 = DateTime.Parse(time_start);
        DateTime date_end1 = DateTime.Parse(time_end);
        for (int i = 0; i < list.Count; i++)
        {
            DateTime date_start2 = DateTime.Parse(list[i][1]);
            DateTime date_end2 = DateTime.Parse(list[i][2]);
            if (date_start1 > date_start2 && date_start1 < date_end2 ||
                date_end1 > date_start2 && date_end1 < date_end2)
            {
                return true;
            }
        }
    }

    return false;
}

```

2 usages

```

public static void CreateAndWriteFile(string path, List<string[]> list)
{
    using (BinaryWriter writer = new BinaryWriter(output: File.Open(path, FileMode.OpenOrCreate)))
    {
        string line = null;
        for (int i = 0; i < list.Count; i++)
        {
            line += (line == null ? "" : "\n") + $"Прізвище: {list[i][0]} \t Час приходу: {list[i][1]} \t Час закінчення: {list[i][2]}";
        }
        writer.Write(line);
    }
}

```

2 usages

```
public static string ReadFile(string path)
{
    using (BinaryReader reader = new BinaryReader(input: File.Open(path, FileMode.Open)))
    {
        string text = reader.ReadString();
        return text;
    }
}
```

1 usage

```
public static List<string[]> FindSpecialClients(List<string[]> list)
{
    List<string[]> resultList = new List<string[]>();
    for (int i = 0; i < list.Count; i++)
    {
        string[] time_start = list[i][1].Split(separator: new[] { ':' });
        string[] time_end = list[i][2].Split(separator: new[] { ':' });
        if (isSpecial(time_start, time_end))
        {
            resultList.Add(list[i]);
        }
    }

    return resultList;
}
```

```
public static bool isSpecial(string[] time_start, string[] time_end)
{
    int hours_start = Int32.Parse(time_start[0]);
    int hours_end = Int32.Parse(time_end[0]);
    int minutes_start = Int32.Parse(time_start[1]);
    int minutes_end = Int32.Parse(time_end[1]);
    if (hours_end - hours_start == 1)
    {
        if (minutes_start > 30 && minutes_end < 30 && 60 - minutes_start + minutes_end < 30)
        {
            return false;
        }
    }

    int delta = minutes_end - minutes_start;
    if (hours_start == hours_end && delta < 30)
    {
        return false;
    }

    return true;
}
```

Код Python

```
from functions import *
import functions

listOfClients = InputText()

fileName1 = 'file1'

CreateFile(fileName1,listOfClients)

textFromFile = ReadFile(fileName1)

print('Текст з першого файлу: ')

print(textFromFile)

resultList = FindSpecialClients(listOfClients)

if len(resultList) != 0:
    fileName2 = 'file2'
    CreateFile(fileName2,resultList)
    print('Текст з другого файлу:')
    textFromResultFile = ReadFile(fileName2)
    print(textFromResultFile)
else:
    print('Немає клієнтів з тривалістю прийому понад 30 хвилин')
```

```

import math
import keyboard
import re
import datetime
import pickle

def InputText():
    listOfClients = []
    print('Введіть список клієнтів, якщо хочете закінчити ввід, натисніть ctrl+space+enter після введення часу закінчення обслуговування.')
    while not keyboard.is_pressed('ctrl+space'):
        person = []
        person.append(input('Введіть прізвище:'))
        while True:
            time_start = input('Введіть час приходу в форматі (HH:mm):')
            while not isValid(time_start):
                time_start = input('Ви ввели час в неправильному форматі, або недопустимі значення. Будь ласка введіть коректний час:')
            time_end = input('Введіть час закінчення обслуговування в форматі (HH:mm):')
            while not (isValid(time_end) and isAdmissible(time_start, time_end)):
                time_end = input('Ви ввели час в неправильному форматі, або недопустимі значення. Будь ласка введіть коректний час:')
            if not isCrossing(listOfClients, time_start, time_end):
                break
            print('Час прийому який ви ввели неможливий, оскільки пересікається з вже існуючими клієнтами, спробуйте ще раз')
        person.append(time_start)
        person.append(time_end)
        listOfClients.append(person)
    return listOfClients

```

```

def isValid(time):
    match = re.match(r'\b((09|1[0-7]):([0-5][0-9]\b))|18:00\b', time)
    if match:
        return True
    return False

def isAdmissible(time1, time2):
    time_start = datetime.datetime.strptime(time1, "%H:%M")
    time_end = datetime.datetime.strptime(time2, "%H:%M")
    if time_end > time_start:
        return True
    return False

def isCrossing(listOfClients, time1, time2):
    if len(listOfClients) != 0:
        time_start1 = datetime.datetime.strptime(time1, "%H:%M")
        time_end1 = datetime.datetime.strptime(time2, "%H:%M")
        for i in range(len(listOfClients)):
            time_start2 = datetime.datetime.strptime(listOfClients[i][1], "%H:%M")
            time_end2 = datetime.datetime.strptime(listOfClients[i][2], "%H:%M")
            if time_start1 > time_start2 and time_start1 < time_end2 or time_end1 > time_start2 and time_end1 < time_end2:
                return True
    return False

```

```

def CreateFile(fileName,listOfClients):
    with open(fileName + '.dat','wb') as file:
        text = ''
        for i in range(len(listOfClients)):
            line = 'Прізвище: ' + listOfClients[i][0] + '\tЧас приходу: ' + listOfClients[i][1] + '\tЧас закінчення: ' + listOfClients[i][2]
            text+= ('' if text==' ' else '\n') + line
        pickle.dump(text,file)

def ReadFile(fileName):
    with open(fileName + '.dat','rb') as file:
        result = pickle.load(file)
        return result
def FindSpecialClients(listOfClients):
    resultlist = []
    for i in range(len(listOfClients)):
        time_start = listOfClients[i][1].split(':')
        time_end = listOfClients[i][2].split(':')
        if isSpecial(time_start,time_end):
            resultlist.append(listOfClients[i])
    return resultlist

def isSpecial(time_start,time_end):
    hours_start = int(time_start[0])
    hours_end = int(time_end[0])
    minutes_start = int(time_start[1])
    minutes_end = int(time_end[1])

    if hours_end - hours_start == 1:
        if minutes_start>30 and minutes_end<30 and 60-minutes_start+minutes_end < 30:
            return False
    if hours_start == hours_end and minutes_end - minutes_start < 30:
        return False

    return True

```

Робота програми

C#

Якщо хочете зак?нчити вв?д - натисн?сть delete, ?накше любую ?ншу клав?шу.

```

Введ?ть пр?звище:wewh
Введ?ть час приходу в формат? (НН:мм):10:00
Введ?ть час зак?нчення обслуговування в формат? (НН:мм):10:30
Введ?ть пр?звище:aga
Введ?ть час приходу в формат? (НН:мм):10:20
Введ?ть час зак?нчення обслуговування в формат? (НН:мм):11:00
Час прийому який ви ввели неможливий, оск?льки перес?кається з вже ?снуючими кл?єнтами, спробуйте ще раз.
Введ?ть час приходу в формат? (НН:мм):10:40
Введ?ть час зак?нчення обслуговування в формат? (НН:мм):11:40
Введ?ть пр?звище:lagdgd
Введ?ть час приходу в формат? (НН:мм):12:00
Введ?ть час зак?нчення обслуговування в формат? (НН:мм):12:20
Введ?ть пр?звище:asga
Введ?ть час приходу в формат? (НН:мм):12:30
Введ?ть час зак?нчення обслуговування в формат? (НН:мм):13:00
Введ?ть пр?звище:adga
Введ?ть час приходу в формат? (НН:мм):13:00
Введ?ть час зак?нчення обслуговування в формат? (НН:мм):13:10
Введ?ть пр?звище:dgdh
Введ?ть час приходу в формат? (НН:мм):13:20
Введ?ть час зак?нчення обслуговування в формат? (НН:мм):13:15
Ви ввели час в неправильному формат?, або недопустим? значення. Будь ласка введ?ть коректний час:14:00
Введ?ть пр?звище:
Текст з першого файлу:
Пр?звище:wewh      Час приходу: 10:00      Час зак?нчення:10:30
Пр?звище:aga      Час приходу: 10:40      Час зак?нчення:11:40
Пр?звище:lagdgd      Час приходу: 12:00      Час зак?нчення:12:20
Пр?звище:asga      Час приходу: 12:30      Час зак?нчення:13:00
Пр?звище:adga      Час приходу: 13:00      Час зак?нчення:13:10
Пр?звище:dgdh      Час приходу: 13:20      Час зак?нчення:14:00

```


Текст з першого файлу:

Пр?звище:wewh	Час приходу: 10:00	Час зак?нчення:10:30
Пр?звище:aga	Час приходу: 10:40	Час зак?нчення:11:40
Пр?звище:laggagd	Час приходу: 12:00	Час зак?нчення:12:20
Пр?звище:asga	Час приходу: 12:30	Час зак?нчення:13:00
Пр?звище:adga	Час приходу: 13:00	Час зак?нчення:13:10
Пр?звище:dgdh	Час приходу: 13:20	Час зак?нчення:14:00

Текст з другого файлу:

Пр?звище:wewh	Час приходу: 10:00	Час зак?нчення:10:30
Пр?звище:aga	Час приходу: 10:40	Час зак?нчення:11:40
Пр?звище:asga	Час приходу: 12:30	Час зак?нчення:13:00
Пр?звище:dgdh	Час приходу: 13:20	Час зак?нчення:14:00

Process finished with exit code 0.

Python

```
Введіть список клієнтів, якщо хочете закінчити ввід, натисніть ctrl+space+enter після введення часу закінчення обслуговування.
Введіть прізвище:shs
Введіть час приходу в форматі (HH:mm):10:00
Введіть час закінчення обслуговування в форматі (HH:mm):10:20
Введіть прізвище:wewe
Введіть час приходу в форматі (HH:mm):10:20
Введіть час закінчення обслуговування в форматі (HH:mm):11:00
Введіть прізвище:dfhdhf
Введіть час приходу в форматі (HH:mm):11:00
Введіть час закінчення обслуговування в форматі (HH:mm):12:00
Введіть прізвище:weweh
Введіть час приходу в форматі (HH:mm):12:00
Введіть час закінчення обслуговування в форматі (HH:mm):12:20
Введіть прізвище:erheh
Введіть час приходу в форматі (HH:mm):13:00
Введіть час закінчення обслуговування в форматі (HH:mm):15:00
Текст з першого файлу:
Прізвище: shsЧас приходу: 10:00Час закінчення: 10:20
Прізвище: weweЧас приходу: 10:20Час закінчення: 11:00
Прізвище: dfhdhfЧас приходу: 11:00Час закінчення: 12:00
Прізвище: wewehЧас приходу: 12:00Час закінчення: 12:20
Прізвище: efhehЧас приходу: 12:30Час закінчення: 13:00
Прізвище: erhehЧас приходу: 13:00Час закінчення: 15:00
Текст з другого файлу:
Прізвище: weweЧас приходу: 10:20Час закінчення: 11:00
Прізвище: dfhdhfЧас приходу: 11:00Час закінчення: 12:00
Прізвище: efhehЧас приходу: 12:30Час закінчення: 13:00
Прізвище: erhehЧас приходу: 13:00Час закінчення: 15:00
```