

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»

«Бінарні файли»

Варіант 5

Виконав студент ІП-13 Вальчишен Ярослав Олександрович
(шифр, прізвище, ім'я, по батькові)

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

Київ 2022

Лабораторна робота

Варіант 13

5. Створити файл з інформацією про телефонні переговори: номер телефону, початок та кінець переговорів (за шаблоном - ГГ:ХХ). З вихідного файлу створити два нових, в одному з яких мають бути зафіксовані денні переговори (з 6:00 до 20:00), а в іншому – нічні.

Код програми

C#

```
using FileOperations;

namespace ProgLab2
{
    class Program
    {
        static void Main(string[] args)
        {
            string inputFilePath = @"Users/yaroslavvalchyshen/Desktop/Study/Labs/Programming/Lab2/CSharpProgLab2/InputFile.bat";
            string dayFilePath = @"Users/yaroslavvalchyshen/Desktop/Study/Labs/Programming/Lab2/CSharpProgLab2/DayFile.bat";
            string nightFilePath = @"Users/yaroslavvalchyshen/Desktop/Study/Labs/Programming/Lab2/CSharpProgLab2/NightFile.bat";

            List<Call> _dayCalls = new List<Call>();
            List<Call> _nightCalls = new List<Call>();

            bool isClearing = Input.GetFileMode();

            Call[] calls = Input.GetCalls().ToArray();

            MyFile.FillFile(inputFilePath, isClearing, calls);
            MyFile.DisplayFile(inputFilePath);

            for (int i = 0; i < calls.Length; i++)
            {
                if (calls[i].GetStartTime().IsDay() && calls[i].GetEndTime().IsDay())
                {
                    _dayCalls.Add(calls[i]);
                }
                else
                {
                    _nightCalls.Add(calls[i]);
                }
            }

            MyFile.FillFile(dayFilePath, isClearing, _dayCalls.ToArray());
            MyFile.DisplayFile(dayFilePath, "Day calls: ");
            MyFile.FillFile(nightFilePath, isClearing, _nightCalls.ToArray());
            MyFile.DisplayFile(nightFilePath, "Night calls: ");
        }
    }
}

using System;

namespace InputOperations
{
    public class Input
    {
        public static List<Call> GetCalls()
        {
            List<Call> calls = new List<Call>();
            Console.WriteLine("Enter information about phone calls. To stop enter empty line \n(format: ***0***** HH:MM HH:MM): ");

            string input = Console.ReadLine();
            while (input != "")
            {
                Call call = CreateCall(input);

                if (call != null)
                {
                    calls.Add(call);
                }
                else
                {
                    Console.WriteLine("Error! Wrong format. Try again");
                }
                input = Console.ReadLine();
            }
            return calls;
        }

        public static bool GetFileMode()
        {
            string rewriteKey = "r";
            Console.WriteLine($"Enter file mode (To rewrite file enter '{rewriteKey}' or anything else to append:");
            return Console.ReadLine() == rewriteKey;
        }

        private static Call CreateCall(string line)
        {
            string[] elements = line.Split(" ");

            if (elements.Length != 3)
            {
                return null;
            }
            if (!IsPhoneNumberFormatValid(elements[0]) || !IsTimeFormatValid(elements[1]) || !IsTimeFormatValid(elements[2]))
            {
                return null;
            }

            Time startTime = new Time(elements[1]);
            Time endTime = new Time(elements[2]);

            return new Call(elements[0], startTime, endTime);
        }
    }
}
```

```

52 private static bool IsPhoneNumberFormatValid(string number)
53 {
54     if (number.Length != 13 || number[0] != '+' || number[3] != '0')
55         return false;
56
57     for (int i = 1; i < number.Length; i++)
58         if (!Char.IsDigit(number[i]))
59             return false;
60
61     return true;
62 }
63
64 private static bool IsTimeFormatValid(string time)
65 {
66     if (time.Length != 5 || time[2] != ':')
67         return false;
68
69     for (int i = 0; i < time.Length; i++)
70         if (!Char.IsDigit(time[i]) && i != 2)
71             return false;
72
73     return true;
74 }
75
76 }
77
78 public class Time
79 {
80     private int _hh;
81     private int _mm;
82
83     public Time(string time)
84     {
85         string[] splitted = time.Split(":");
86
87         SetHours(Convert.ToInt32(splitted[0]));
88         SetMinutes(Convert.ToInt32(splitted[1]));
89     }
90
91     private void SetHours(int hours)
92     {
93         _hh = (hours > 23) ? 0 : hours;
94     }
95
96     private void SetMinutes(int minutes)
97     {
98         _mm = (minutes > 59) ? 0 : minutes;
99     }
100
101     public string GetTime()
102     {
103         string time = ((_hh < 10) ? "0" : "") + _hh + ":" + ((_mm < 10) ? "0" : "") + _mm;
104         return time;
105     }
106
107     public bool IsDay()
108     {
109         return (_hh > 6 && _hh < 20);
110     }
111 }
112
113
114 public class Call
115 {
116     private string _phoneNumber;
117     private Time _startTime;
118     private Time _endTime;
119
120     public Call(string phoneNumber, Time startTime, Time endTime)
121     {
122         _phoneNumber = phoneNumber;
123         _startTime = startTime;
124         _endTime = endTime;
125     }
126
127     public string GetPhoneNumber()
128     {
129         return _phoneNumber;
130     }
131
132     public Time GetStartTime()
133     {
134         return _startTime;
135     }
136
137     public Time GetEndTime()
138     {
139         return _endTime;
140     }
141 }
142
143 }
144

```

```

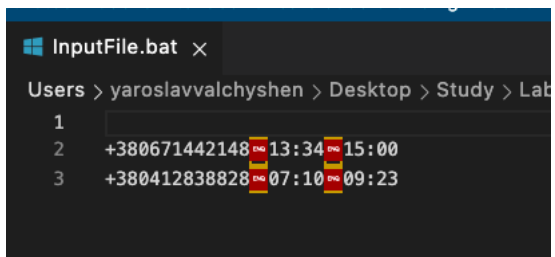
5
6 namespace FileOperations
7 {
8     public class MyFile
9     {
10         public static void FillFile(string path, bool isClearing, Call[] data)
11         {
12             if (isClearing)
13                 File.Delete(path);
14
15             BinaryWriter binaryWriter = new BinaryWriter(File.Open(path, FileMode.Append));
16             for (int i = 0; i < data.Length; i++)
17             {
18                 binaryWriter.Write(data[i].GetPhoneNumber());
19                 binaryWriter.Write(data[i].GetStartTime().GetTime());
20                 binaryWriter.Write(data[i].GetEndTime().GetTime());
21             }
22             binaryWriter.Close();
23         }
24
25         public static Call[] ReadFile(string path)
26         {
27             List<Call> calls = new List<Call>();
28             BinaryReader binaryReader = new BinaryReader(File.Open(path, FileMode.Open));
29             while (binaryReader.PeekChar() > -1)
30             {
31                 string phoneNumber = binaryReader.ReadString();
32                 Time startTime = new Time(binaryReader.ReadString());
33                 Time endTime = new Time(binaryReader.ReadString());
34
35                 Call call = new Call(phoneNumber, startTime, endTime);
36                 calls.Add(call);
37             }
38             binaryReader.Close();
39
40             return calls.ToArray();
41         }
42
43         public static void DisplayFile(string path, string caption = "File: ")
44         {
45             BinaryReader binaryReader = new BinaryReader(File.Open(path, FileMode.Open));
46
47             Console.WriteLine(caption);
48
49             while (binaryReader.PeekChar() > -1)
50             {
51                 string phoneNumber = binaryReader.ReadString();
52                 Time startTime = new Time(binaryReader.ReadString());
53                 Time endTime = new Time(binaryReader.ReadString());

```

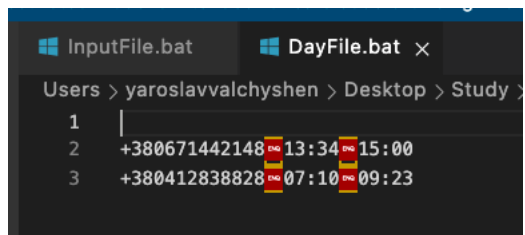
Тестування:

```
Enter file mode(To rewrite file enter 'r' or anything else to append):
r
Enter information about phone calls. To stop enter empty line
(format: +**0***** HH:MM HH:MM):
+380671442148 13:34 15:00
+389671553123 23:00 22222
Error! Wrong format. Try again
+380412838828 07:10 09:23

File:
+380671442148 13:34 15:00
+380412838828 07:10 09:23
Day calls:
+380671442148 13:34 15:00
+380412838828 07:10 09:23
Night calls:
█
```

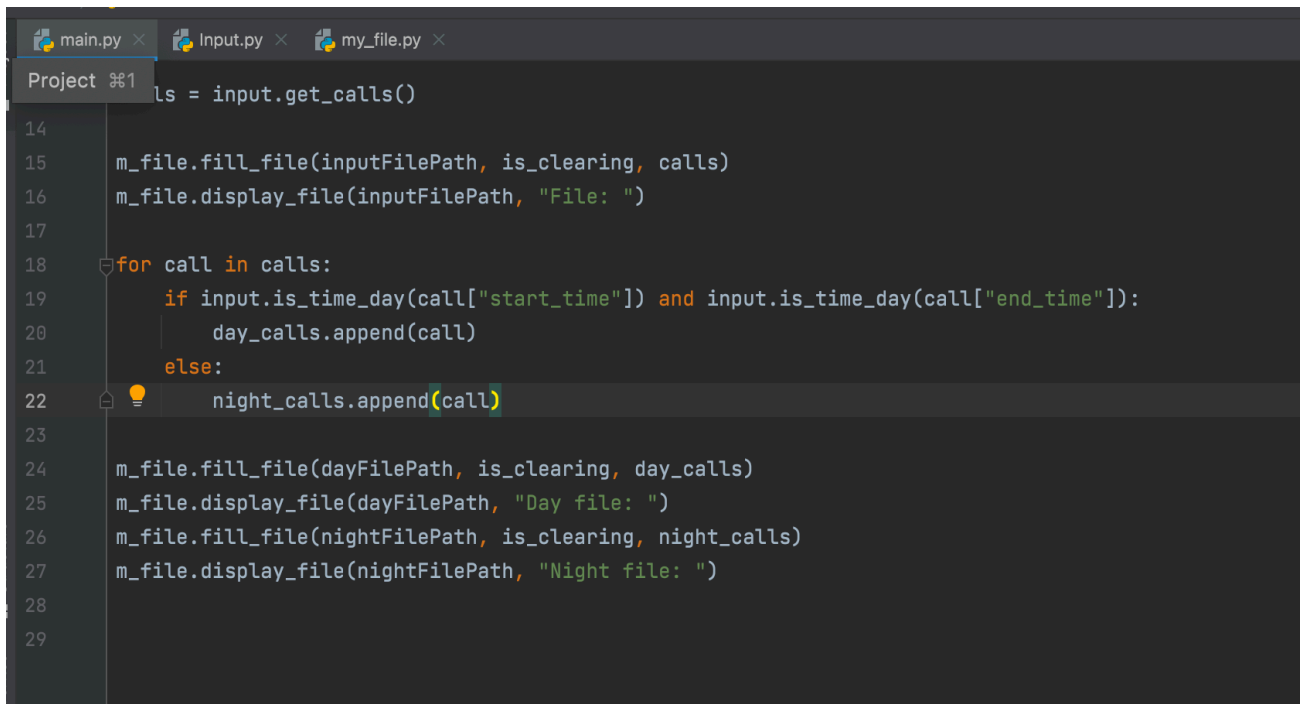


```
InputFile.bat x
Users > yaroslavvalchyshen > Desktop > Study > Lab
1
2 +380671442148 13:34 15:00
3 +380412838828 07:10 09:23
```

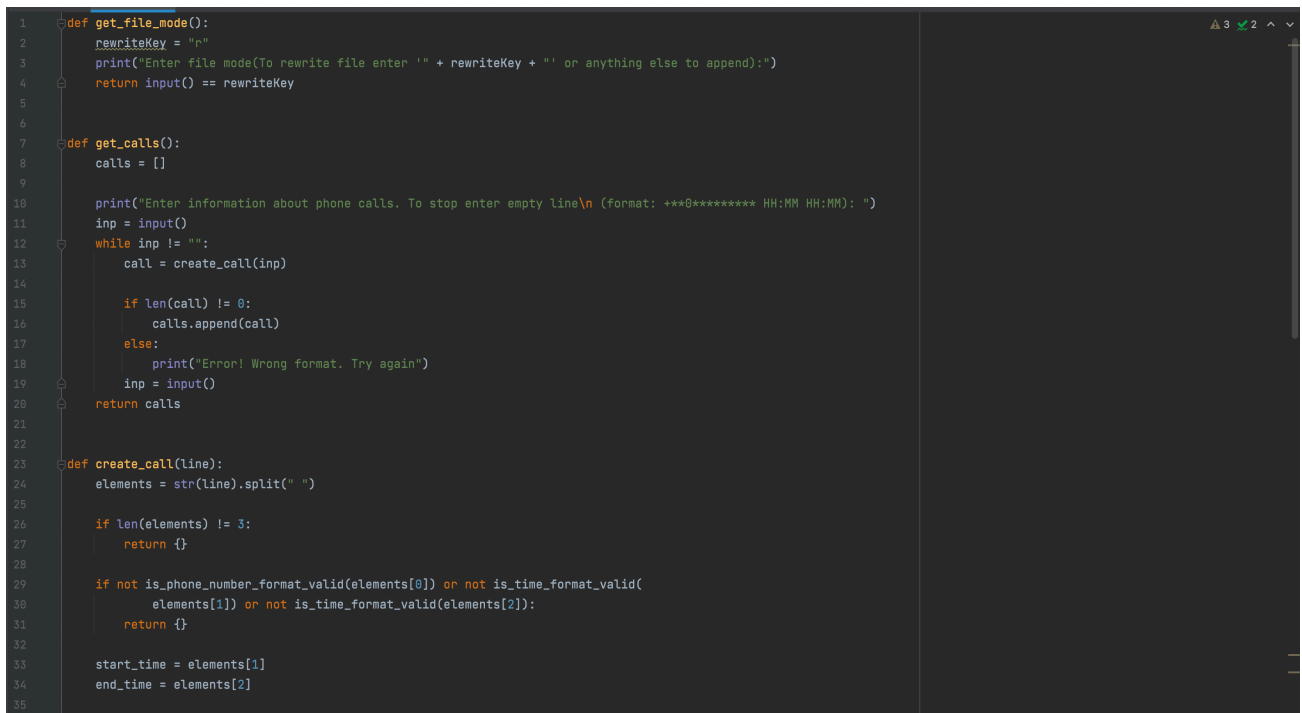


```
InputFile.bat DayFile.bat x
Users > yaroslavvalchyshen > Desktop > Study > Lab
1
2 +380671442148 13:34 15:00
3 +380412838828 07:10 09:23
```

Python



```
13 calls = input.get_calls()
14
15 m_file.fill_file(inputFilePath, is_clearing, calls)
16 m_file.display_file(inputFilePath, "File: ")
17
18 for call in calls:
19     if input.is_time_day(call["start_time"]) and input.is_time_day(call["end_time"]):
20         day_calls.append(call)
21     else:
22         night_calls.append(call)
23
24 m_file.fill_file(dayFilePath, is_clearing, day_calls)
25 m_file.display_file(dayFilePath, "Day file: ")
26 m_file.fill_file(nightFilePath, is_clearing, night_calls)
27 m_file.display_file(nightFilePath, "Night file: ")
28
29
```



```
1 def get_file_mode():
2     rewriteKey = "r"
3     print("Enter file mode (To rewrite file enter 'r' + rewriteKey + 'a' or anything else to append):")
4     return input() == rewriteKey
5
6
7 def get_calls():
8     calls = []
9
10    print("Enter information about phone calls. To stop enter empty line\n (format: +*0***** HH:MM HH:MM): ")
11    inp = input()
12    while inp != "":
13        call = create_call(inp)
14
15        if len(call) != 0:
16            calls.append(call)
17        else:
18            print("Error! Wrong format. Try again")
19            inp = input()
20    return calls
21
22
23 def create_call(line):
24     elements = str(line).split(" ")
25
26     if len(elements) != 3:
27         return {}
28
29     if not is_phone_number_format_valid(elements[0]) or not is_time_format_valid(
30         elements[1]) or not is_time_format_valid(elements[2]):
31         return {}
32
33     start_time = elements[1]
34     end_time = elements[2]
35
```

```

def is_phone_number_format_valid(number):
    if len(number) != 13 or number[0] != "+" or number[3] != "0":
        return False
    i = 1
    while i < len(number):
        if not str(number[i]).isdigit():
            return False
        i += 1
    return True

def is_time_format_valid(time):
    if len(time) != 5 or time[2] != ":":
        return False

    i = 0
    while i < len(time):
        if not str(time[i]).isdigit() and i != 2:
            return False
        i += 1

    splitted = str(time).split(":")
    if int(splitted[0]) > 23 or int(splitted[1]) > 59:
        return False

    return True

def is_time_day(time):
    splitted = str(time).split(":")
    return 6 < int(splitted[0]) < 20

```

Тестування:

Enter file mode(To rewrite file enter 'r' or anything else to append):

r

Enter information about phone calls. To stop enter empty line

(format: +*0***** HH:MM HH:MM):

+380671554128 00:13 12:00

+380612345566 11:22 23:00

=323

Error! Wrong format. Try again

+380671554128 11:00 23:11

File:

+380671554128 00:13 12:00

+380612345566 11:22 23:00

+380671554128 11:00 23:11

Day file:

Night file:

+380671554128 00:13 12:00

+380612345566 11:22 23:00

+380671554128 11:00 23:11

Process finished with exit code 0

|

Висновки:

У лабораторній роботі я застосував на практиці знання щодо створення та обробки бінарних файлів даних