

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.  
Метидології програмування»

«Бінарні файли»

Варіант 9

Виконав студент ІП-13 Григоренко Родіон Ярославович  
(шифр, прізвище, ім'я, по батькові)

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

## Лабораторна робота 2

### Бінарні файли

#### Варіант 9

9. Створити файл із списком клієнтів перукарні на день: прізвище та ім'я клієнта, час (у форматі ГГ:ХХ) та передбачувана тривалість процедури. При створенні файлу перевіряти, чи не зайнятий час і чи достатньо у майстра вільного часу для виконання необхідної процедури. Вивести список усіх клієнтів, які прийдуть після 16:30.

#### Код програми

C++

#### Lab2.cpp

```
#include <iostream>
#include "Header.h";
using namespace std;
int main()
{
    string file = "file.txt";
    input_file(file);
    show_late_clients(file);
}
```

#### Header.h

```
#pragma once
#include <iostream>
#include <string>
#include <fstream>
#include <vector>
#include <ctime>
#include <algorithm>

using namespace std;

struct client;

void input_file(string);

vector<string> split(string, char);
void input_file(string);
bool check_time(string, client);
string get_line(string, int);
void show_late_clients(string filePath);
```

## Source.cpp

```
#include "Header.h";

struct client {
    string surname;
    string name;
    int hrs;
    int mins;
    int dur;
    bool state;
    void save(string line) {
        vector <string> arr = split(line, ' ');
        if (arr.size() > 3) {
            state = true;
            surname = arr[0];
            name = arr[1];
            dur = stoi(arr[3]);
            if (count(arr[2].begin(), arr[2].end(), ':')) {
                vector <string> tm = split(arr[2], ':');
                hrs = stoi(tm[0]);
                mins = stoi(tm[1]);
            }
        }
        else {
            state = false;
            cout << "Wrong time format\n";
        }
    }
    else {
        state = false;
        cout << "Wrong format\n";
    }
};

void input_file(string filePath) {
    string line;
    cout << "Enter client info in format: Surname Name Time(HH:MM) Duration(in minutes)\n";
    getline(cin, line);
    while (line!="") {
        client person;
        person.save(line);
        if (person.state) {
            if (check_time(filePath, person) == true) {
                ofstream file(filePath, ios::app);
                if (!file.is_open()) {
                    cout << "ERROR: could not open";
                }
            }
            else {
                file << line + '\n';
            }
            file.close();
        }
        else {
            cout << "No time";
        }
    }
    getline(cin, line);
}
```

```

}

bool check_time(string filePath, client person) {
    string line = " ";
    bool flag = true;
    ifstream File(filePath);
    bool chk;
    while (!File.eof()) {
        getline(File, line);
        client person1;
        if (line != "") {
            person1.save(line);
        }
        chk = ((person1.hrs * 60 + person1.mins + person1.dur > person.hrs * 60 +
            person.mins) && (person1.hrs * 60 + person1.mins + person1.dur < person.hrs *
            60 + person.mins + person.dur)) || ((person.hrs * 60 + person.mins + person.dur
            > person1.hrs * 60 + person1.mins) && (person.hrs * 60 + person.mins +
            person.dur < person1.hrs * 60 + person1.mins + person1.dur));
        if (chk) {
            flag = false;
        }
    }
    File.close();
    return flag;
}

void show_late_clients(string filePath) {
    ifstream file(filePath);
    string line = "";
    while (!file.eof()) {
        getline(file, line);
        client person;
        if (line != "") {
            person.save(line);
        }
        if (person.hrs*60 + person.mins > 16*60 + 30) {
            cout << line << "\n";
        }
    }
    file.close();
}

vector<string> split(string str, char separator)
{
    vector<string> res;
    string slice = "";
    str += " ";
    for (int i = 0; i < str.length(); i++) {
        if (str[i] == separator)
        {
            if (slice.length() > 0) {
                res.push_back(slice);
                slice = "";
            }
        }
        else {
            slice += str[i];
        }
    }
}

```

```
}  
}  
res.push_back(slice);  
return res;  
}
```

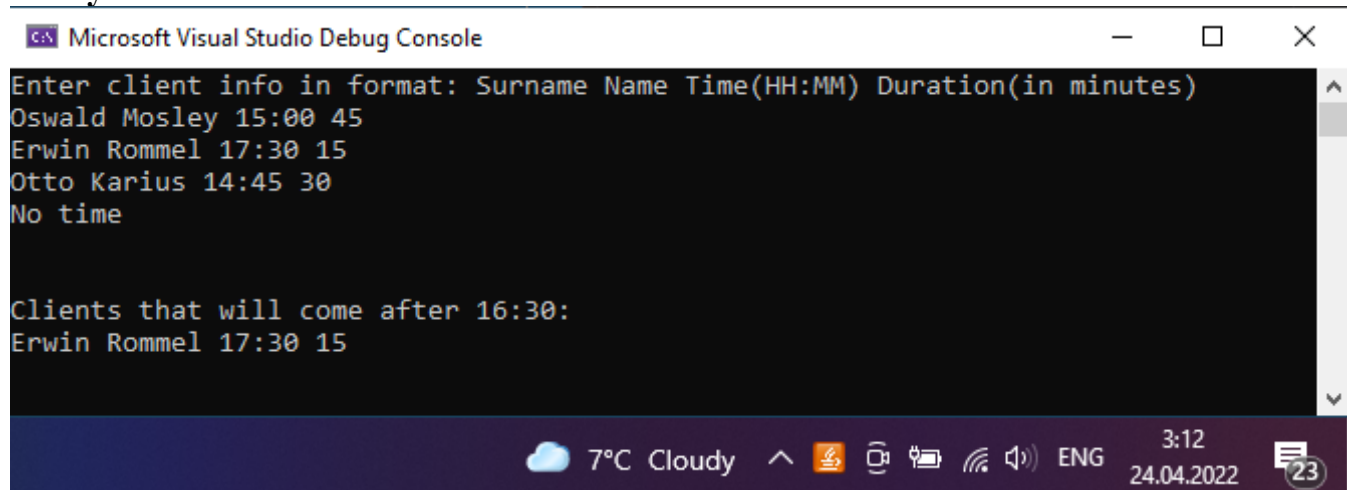








## Тестування:



The screenshot shows the Microsoft Visual Studio Debug Console with a dark background. The text is as follows:

```
Enter client info in format: Surname Name Time(HH:MM) Duration(in minutes)
Oswald Mosley 15:00 45
Erwin Rommel 17:30 15
Otto Karius 14:45 30
No time

Clients that will come after 16:30:
Erwin Rommel 17:30 15
```

At the bottom of the console window, there is a status bar showing weather information (7°C Cloudy), system icons, language (ENG), and the date/time (3:12 24.04.2022).

## Python

### main.py

```
from Functions import *
if __name__ == '__main__':
    file = "file.txt"
    input_file(file)
    show_late_clients(file)
```

### Functions.py

```
class Client:
    surname = ""
    name = ""
    hrs = 0
    mins = 0
    dur = 0
    state = False
    def save(self,line):
        arr = line.split(' ')
        if (len(arr) > 3):
            self.state = True
            self.surname = arr[0]
            self.name = arr[1]
            self.dur = int(arr[3])
            if arr[2].count(':'):
                tm = arr[2].split(':')
                self.hrs = int(tm[0])
                self.mins = int(tm[1])
```

```

        else:
            self.state = False
            print("Wrong time format\n")
    else:
        self.state = False
        print("Wrong format\n")

def input_file(filePath):
    #print("Enter client info in format: Surname Name
    Time(HH:MM) Duration(in minutes)\n")
    line = input("Enter client info in format: Surname Name
    Time(HH:MM) Duration(in minutes)\n")
    while line != "":
        person = Client()
        person.save(line)
        if person.state == True:
            if check_time(filePath, person) == True:
                with open(filePath, 'a') as file:
                    file.write(line + '\n')
                file.close()
            else:
                print("No time\n")
        line = input()

def check_time(filePath, person):
    flag = True
    with open(filePath, 'r') as file:
        lines = file.read().split("\n")
        chk = True
        for line in lines:
            person1 = Client()
            if line != "":
                person1.save(line)
                chk = ((person1.hrs * 60 + person1.mins +
                person1.dur > person.hrs * 60 + person.mins) and (person1.hrs
                * 60 + person1.mins + person1.dur < person.hrs * 60 +
                person.mins + person.dur)) or ((person.hrs * 60 + person.mins
                + person.dur > person1.hrs * 60 + person1.mins) and
                (person.hrs * 60 + person.mins + person.dur < person1.hrs *
                60 + person1.mins + person1.dur))
            if chk == True:
                flag = False
        file.close()
    return flag

```

```
def show_late_clients(filePath):
    print("\nClients that will come after 16:30:\n")
    with open(filePath, 'r') as file:
        lines = file.read().split("\n")
        for line in lines:
            person = Client()
            if line != "":
                person.save(line)
            if person.hrs*60 + person.mins > 16*60 + 30:
                print(line + "\n")
    file.close()
```



```
Enter client info in format: Surname Name Time(HH:MM) Duration(in minutes)
```

```
Oswald Mosley 15:00 45
```

```
Erwin Rommel 17:30 15
```

```
Otto Karius 14:45 30
```

```
No time
```

```
Clients that will come after 16:30:
```

```
Erwin Rommel 17:30 15
```

### **Тестування:**

### **Висновки:**

Я вивчив особливості створення і обробки бінарних файлів даних. Застосував ці навички на практиці.