

Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІІІ

ЗВІТ

з виконання лабораторної роботи № 2

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 22

Виконав:

студент 1-го курсу

гр. ІІІ-22 ФІОТ

Підпанюк Віталій Андрійович

Київ 2023

Завдання:

22. 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 програми:

main.cpp:

```
#include <iostream>
#include <istream>
#include <fstream>
#include <string>
#include "functions.h"
using namespace std;

int main() {
    file_write();
    return 0;
}
```

functions.h:

```
#pragma once
struct user;
struct Time;
void file_write();
void file_read();
void surname_write(char surname[20]);
void time_managing(Time *start_time, Time *end_time);
void time_write(int* start_hours, int* start_minutes);
bool time_check(Time *new_time);
int task_check();
void show_special_persons();
```

functions.cpp:

```
#include <iostream>
#include <istream>
#include <fstream>
#include <string>
#include "functions.h"
using namespace std;

struct Time
{
    int hours;
    int minutes;
};

struct user
{
    char surname[20];
    Time start;
    Time end;
};

void file_write()
{
    int ch;
    int i = 0;

    while(1){

        user user_info;
        surname_write(user_info.surname);

        time_managing(&user_info.start, &user_info.end);

        ofstream oFile("output.bin", ios::binary | ios::app);
        oFile.write((char*)&user_info, sizeof(user));
        oFile.close();
        cout<<"If you want to stop, press 0,\nif you want to continue press 1\nIf
you want to see result press 2\n";
        ch = task_check();
        ::fflush(stdin);
```

```

    if(ch == 0) break;
    if(ch == 1)
    {
        i++;
        ch = 300;
    }
    if(ch == 2)
    {
        file_read();
        show_special_persons();
        cout<<"If you want to stop, press 0,\nif you want to continue press 1\n";
        ch = task_check();
        if(ch == 0) break;
        if(ch == 1)
        {
            i++;
            ch = 300;
        }
    }
    ch = 300;

}

cout<<"\nall right\n";
}

void file_read()
{
    user user_info;
    int i = 1;
    ifstream iFile("output.bin", ios::binary);
    while(iFile.read((char*)&user_info, sizeof(user)))
    {
        cout<<"User"<<i<<": "<<user_info.surname<<endl;
        cout<<"Start time:
"<<user_info.start.hours<<": "<<user_info.start.minutes<<endl;
        cout<<"End time:
"<<user_info.end.hours<<": "<<user_info.end.minutes<<endl;
        cout<<"\n";
        i++;
    }
}

```

```

    }
    iFile.close();
}

void surname_write(char surname[20])
{
    cout << "Enter user`s surname:\n";
    cin>>surname;
}

void time_write(int* start_hours, int* start_minutes) {
    int &hours = *start_hours;
    int &minutes = *start_minutes;
    while (1) {
        cin >> hours;
        if (0 <= hours && hours < 24) break;
        else cout<< "*****\n-----Error-----\nHours
must be equal number between 0 and 24\nTry another
number\n*****\n";
    }
    cout << ":";
    while (1) {
        cin >> minutes;
        if (0 <= minutes && minutes < 60) break;
        else cout<< "*****\n-----Error-----
\nMinutes must be equal number between 0 and 60\nTry another
number\n*****\n";
    }
}

void time_managing(Time *start_time, Time *end_time)
{
    Time start = *start_time;
    Time end = *end_time;
    bool check = false;
    while(1)
    {
        cout << "Enter start time in format like hh:mm\n";
        while(1)
        {

```

```

        time_write(&start.hours, &start.minutes);
        check = time_check(&start);
        if(check) break;
    }
    cout << "Enter end time in format like hh:mm\n";
    check = false;
    while(1)
    {
        time_write(&end.hours, &end.minutes);
        check = time_check(&end);
        if(check) break;
    }

    if((start.hours*60+start.minutes)<(end.hours*60+end.minutes)) break;
    else cout<<"*****\nError\nStart time must be less value then
the end time\n*****\n";
    }
    *start_time = start;
    *end_time = end;
}

bool time_check(Time *new_time)
{
    Time cur_time = *new_time;
    bool check = false;
    int k = 1;
    int i = 0;
    user user_info;
    //int i = 1;
    ifstream iFile("output.bin", ios::binary);
    while (iFile.read((char *) &user_info, sizeof(user)))
    {
        check = false;
        if((cur_time.hours * 60 + cur_time.minutes) > (user_info.end.hours * 60
+ user_info.end.minutes))
            check = true;
        if((cur_time.hours * 60 + cur_time.minutes) < (user_info.start.hours *
60 + user_info.start.minutes))
            check = true;
        if(!check) {i++;}
    }
}

```

```

        k++;
    }
    if(i==0) {return true;}
    else
    {
        cout<<"-----\nError\nThis time is already registered\nTry again\n--
        -----\n";
    }
    iFile.close();
    return false;
}

```

```

int task_check()
{
    int ch = 300;
    while(1)
    {
        cin>>ch;
        if(ch != 0 && ch != 1 && ch != 2)
        {
            cout<<"Invalid indicator\nTry again\n";
            ch = 300;
        }
        if(ch == 0 || ch == 1 || ch == 2) break;
    }
    return ch;
}

```

```

void show_special_persons()
{
    user user_info;
    ifstream iFile("output.bin", ios::binary);
    cout<<"*****\nUsers, who
was served longer then 30 minutes\n";
    while(iFile.read((char*)&user_info, sizeof(user)))
    {
        if(((user_info.end.hours*60+user_info.end.minutes)-
        (user_info.start.hours*60+user_info.start.minutes))>=30)
        {
            cout<<"User"<<": "<<user_info.surname<<endl;

```

```

        cout<<"Start time:
"<<user_info.start.hours<<":"<<user_info.start.minutes<<endl;
        cout<<"End time:
"<<user_info.end.hours<<":"<<user_info.end.minutes<<endl;
        cout<<"\n";
    }
}
iFile.close();
}

```

Тестування:

Наочні результати успішного заповнення графіку прийому відвідувачів.

```

If you want to stop, press 0,
if you want to continue press 1
If you want to see result press 2
2
User1: Mazur
Start time: 10:10
End time: 10:45

User2: Grygorenko
Start time: 10:55
End time: 12:0

*****
Users, who was served longer then 30 minutes
User: Mazur
Start time: 10:10
End time: 10:45

User: Grygorenko
Start time: 10:55
End time: 12:0

If you want to stop, press 0,
if you want to continue press 1
0
all right

```

```

Enter user`s surname:
Mazur
Enter start time in format like hh:mm
10
:10
Enter end time in format like hh:mm
10
:45
If you want to stop, press 0,
if you want to continue press 1
If you want to see result press 2
1
Enter user`s surname:
Grygorenko
Enter start time in format like hh:mm
10
:55
Enter end time in format like hh:mm
12
:00
If you want to stop, press 0,
if you want to continue press 1
If you want to see result press 2

```

Випробування програми на хибні вхідні дані:

Enter user`s surname:

Viktor

Enter start time in format like hh:mm

15

:00

Enter end time in format like hh:mm

14

:30

Error

Start time must be less value then the end time

Enter start time in format like hh:mm