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 20

Виконав студент ІП-15, Ликова Катерина Олександрівна
(шифр, прізвище, ім’я, по батькові)

Перевірила Вєчерковська Анастасія Сергіївна
(прізвище, ім’я, по батькові)

Київ 2022

Мета роботи: вивчити особливості створення і обробки бінарних файлів.

Постановка задачі:

20. Створити файл із списком телепередач: назва, час початку та час закінчення передачі. Визначити тривалість кожної передачі. Створити новий файл, що містить передачі, які відображаються в денний час (з 9:00 до 18:00).

Код

C++

main.cpp

```
#include "func.h"
int main()
{
	srand(time(NULL));
	int num = set_num_show();
	vector<TvShow> shows = set_shows(num);
	int mode = choose_mode();
	create_file("first", shows, mode);
	shows = read_file("first");
	cout << "Shows: " << endl;
	print_shows(shows);
	cout << "Durations of shows: " << endl;
	print_show_duration(shows);
	vector<TvShow> new_shows = set_day_shows(shows);
	create_file("second", new_shows, 1);
	new_shows = read_file("second");
	cout << "Day shows: " << endl;
	print_shows(new_shows);
	return 0;
}
```

func.h

```
#pragma once
#include <iostream>
#include <fstream>
#include <string>
#include <math.h>
#include <vector>
using namespace std;
struct Time
{
	int hours;
	int minutes;
};
struct TvShow
{
	char title[30];
	Time start;
```

```
	Time end;
};
int set_num_show();
string set_title();
Time set_show_time_start();
Time set_show_time_end(Time);
TvShow set_show();
vector<TvShow> set_shows(int);
void print_show(TvShow show, int num);
void print_time(Time time);
void print_shows(vector<TvShow> shows);
int choose_mode();
void create_file(string file_name, vector <TvShow> shows, int mode);
Time show_duration(Time start, Time end);
void print_show_duration(vector<TvShow> shows);
bool is_day_show(TvShow show);
vector<TvShow> set_day_shows(vector<TvShow> shows);
vector<TvShow> read_file(string file_name);
```

func.cpp

```
#include "func.h"
int set_num_show()
{
	string num;
	cout << "Enter number of TV shows: ";
	getline(cin, num);
	int i = 0;
	while (i < size(num))
	{
		if (!isdigit(num[i]) || num[0] == '0')
		{
			cout << "Your answer is incorrect. Please enter the correct answer: ";
			getline(cin, num);
			i = 0;
		}
		else
		{
			i++;
		}
	}
	return stoi(num);
}
string set_title()
{
	string text;
	cout << "Enter the title of the show: ";
	getline(cin, text);
	while (size(text) > 30)
	{
		cout << "The title is too long. Please enter title not more than 30 characters long: ";
		getline(cin, text);
	}
	return text;
```

```
}
Time set_show_time_start()
{
	Time num;
	num.hours = rand() % 24;
	num.minutes = rand() % 60;
	return num;
}
Time set_show_time_end(Time start)
{
	Time num;
	num.hours = start.hours + rand() % 9;
	if (num.hours >= 24)
	{
		num.hours -= 24;
	}
	num.minutes = start.minutes + rand() % 60;
	if (num.minutes >= 60)
	{
		num.minutes -= 60;
		num.hours++;
	}
	return num;
}
TvShow set_show()
{
	TvShow new_show;
	strcpy_s(new_show.title, set_title().c_str());
	new_show.start = set_show_time_start();
	new_show.end = set_show_time_end(new_show.start);
	return new_show;
}
vector<TvShow> set_shows(int num)
{
	vector<TvShow> shows;
	for (int i = 0; i < num; i++)
	{
		shows.push_back(set_show());
	}
	return shows;
}
void print_show(TvShow show, int num)
{
	cout << "Title of the show " << num << ": " << show.title << endl;
	cout << "Time of the start of the show " << num << ": ";
	print_time(show.start);
	cout << "Time of the end of the show " << num << ": ";
	print_time(show.end);
}
void print_time(Time time)
{
	cout << time.hours << ":" << time.minutes << endl;
```

```
}
void print_shows(vector<TvShow> shows)
{
	for (int i = 0; i < shows.size(); i++)
	{
		print_show(shows[i], i + 1);
	}
}
int choose_mode()
{
	string str;
	cout << "Choose writing mode: 1 - for overwriting the file, 2 - for adding to the file: ";
	cin >> str;
	while (str != "1" && str != "2")
	{
		cout << "You entered a wrong answer. Please enter '1' or '2': ";
		cin >> str;
	}
	return stoi(str);
}
enum File_mode {
	OVERWRITING = 1,
	ADDING = 2
};
void create_file(string file_name, vector <TvShow> shows, int mode)
{
	ofstream file;
	if (File_mode::OVERWRITING == mode) {
		file.open(file_name, ios::binary);
	}
	else {
		file.open(file_name, ios::binary | ios::app);
	}

	for (TvShow show : shows) {
		file.write((char*)(&show), sizeof(TvShow));
	}
	file.close();
}
Time show_duration(Time start, Time end)
{
	Time dur;
	if (end.hours == start.hours)
	{
		dur.hours = 0;
		dur.minutes = end.minutes - start.minutes;
	}
	else
	{
		dur.hours = end.hours - start.hours - 1;
		if (dur.hours < 0)
		{
```

```
                dur.hours = 24 - start.hours - 1 + end.hours;
            }
            dur.minutes = 60 - start.minutes + end.minutes;
            if (dur.minutes >= 60)
            {
                dur.minutes -= 60;
                dur.hours += 1;
            }
        }
        return dur;
}
void print_show_duration(vector<TvShow> shows)
{
        for (int i = 0; i < shows.size(); i++)
        {
            cout << "The duration of show " << i + 1 << " is: ";
            print_time(show_duration(shows[i].start, shows[i].end));
        }
}
bool is_day_show(TvShow show)
{
        bool res = false;
        if (show.start.hours >= 9 && show.start.hours <= 18 && show.end.hours <= 18 &&
show.end.hours >= 9)
        {
            if (show.end.hours == 18)
            {
                if (show.end.minutes == 0)
                {
                    res = true;
                }
            }
            else
            {
                res = true;
            }
        }
        return res;
}
vector<TvShow> read_file(string file_name)
{
        vector<TvShow> shows;
        TvShow show;
        ifstream file(file_name, ios::binary);
        while (file.read((char*)(&show), sizeof(TvShow)))
        {
            shows.push_back(show);
        }
        file.close();
        return shows;
}
vector<TvShow> set_day_shows(vector<TvShow> shows)
```

```
{
	vector<TvShow> new_shows;
	for (TvShow show : shows)
	{
		if (is_day_show(show))
		{
			new_shows.push_back(show);
		}
	}
	return new_shows;
}
```

Python

main.py

```
import func
def main():
    num = func.set_num_show()
    shows = func.set_shows(num)
    mode = func.choose_mode()
    func.create_file("first", shows, mode)
    shows = func.read_file("first")
    print("Shows:")
    func.print_shows(shows)
    print("Durations of shows:")
    func.print_show_duration(shows)
    new_shows = func.set_day_shows(shows)
    func.create_file("second", new_shows, "wb")
    new_shows = func.read_file("second")
    print("Day shows:")
    func.print_shows(new_shows)

if __name__ == '__main__':
    main()
```

structures.py

```
class Time:
    def __init__(time, hours, minutes):
        time.hours = hours
        time.minutes = minutes
class TvShow:
    def __init__(show, title, start, end):
        show.title = title
        show.start = start
        show.end  = end
```

func.py

```
import pickle
import random
import structures as struct
def set_num_show():
    num = input("Enter number of TV shows: ")
    while (not num.isdigit()) or int(num) < 0:
        num = input("Your answer is incorrect. Please enter the correct answer: ")
    return int(num)
def set_title():
```

```
    text = input("Enter the title of the show: ")
    return text
def set_show_time_start():
    hours = random.randint(0, 23)
    minutes = random.randint(0, 59)
    num = struct.Time(hours, minutes)
    return num
def set_show_time_end(start):
    hours = start.hours + random.randint(0, 8)
    if hours >= 24:
        hours -= 24
    minutes = start.minutes + random.randint(0, 59)
    if minutes >= 60:
        minutes -=60
        hours += 1
    num = struct.Time(hours, minutes)
    return num
def set_show():
    title = set_title()
    start = set_show_time_start()
    end = set_show_time_end(start)
    new_show = struct.TvShow(title, start, end)
    return new_show
def set_shows(num):
    shows = []
    for i in range(num):
        shows.append(set_show())
    return shows
def print_show(show, num):
    print("Title of the show", num, ": ", show.title)
    print("Time of the start of the show", num, ": ", end = "")
    print_time(show.start)
    print("Time of the end of the show", num, ": ", end = "")
    print_time(show.end)
def print_time(time):
    print(time.hours, ":", time.minutes)
def print_shows(shows):
    for i in range(len(shows)):
        print_show(shows[i], i + 1)
def choose_mode():
    str = input("Choose writing mode: 1 - for overwriting the file, 2 - for adding to the file: ")
    while str != "1" and str != "2":
        str = input("You entered a wrong answer. Please enter '1' or '2': ")
    if str == "1":
        return "wb"
    else:
        return "ab"
def create_file(file_name, shows, mode):
    with open(file_name, mode) as file:
        for show in shows:
            pickle.dump(show, file)
def print_file(file_name):
```

```
    i = 0
    with open(file_name, "rb") as file:
        while True:
            try:
                i += 1
                show = pickle.load(file)
                print_show(show, i)
            except EOFError:
                break
def show_duration(start, end):
    if end.hours == start.hours:
        hours = 0
        minutes = end.minutes - start.minutes
    else:
        hours = end.hours - start.hours - 1
        if hours < 0:
            hours = 24 - start.hours - 1 + end.hours
        minutes = 60 - start.minutes + end.minutes
        if minutes >= 60:
            minutes -= 60;
            hours += 1;
    dur = struct.Time(hours, minutes)
    return dur
def print_show_duration(shows):
    for i in range(len(shows)):
        print("The duration of show", i + 1, "is: ", end = "")
        print_time(show_duration(shows[i].start, shows[i].end))
def is_day_show(show):
    res = False
    if show.start.hours >= 9 and show.start.hours <= 18 and show.end.hours <= 18 and show.end.hours >=
9:
        if show.end.hours == 18:
            if show.end.minutes == 0:
                res = True
        else:
            res = True
    return res
def set_day_shows(shows):
    new_shows = []
    for show in shows:
        if is_day_show(show):
            new_shows.append(show)
    return new_shows
def read_file(file_name):
    shows = []
    with open(file_name, "rb") as file:
        while True:
            try:
                show = pickle.load(file)
                shows.append(show)
            except EOFError:
                break
```

return shows

Тестування

C++

Microsoft Visual Studio Debug Console

```
Enter number of TV shows: j y
Your answer is incorrect. Please enter the correct answer: 15
Enter the title of the show: q
Enter the title of the show: w
Enter the title of the show: e
Enter the title of the show: r
Enter the title of the show: t
Enter the title of the show: y
Enter the title of the show: u
Enter the title of the show: i
Enter the title of the show: o
Enter the title of the show: p
Enter the title of the show: l
Enter the title of the show: k
Enter the title of the show: j
Enter the title of the show: h
Enter the title of the show: g
Choose writing mode: 1 - for overwriting the file, 2 - for adding to the file: f
You entered a wrong answer. Please enter '1' or '2': 1
Shows:
Title of the show 1: q
Time of the start of the show 1: 18:24
Time of the end of the show 1: 22:54
Title of the show 2: w
Time of the start of the show 2: 1:35
Time of the end of the show 2: 9:48
Title of the show 3: e
Time of the start of the show 3: 11:4
Time of the end of the show 3: 16:35
Title of the show 4: r
Time of the start of the show 4: 8:53
Time of the end of the show 4: 13:7
Title of the show 5: t
Time of the start of the show 5: 3:24
Time of the end of the show 5: 7:11
Title of the show 6: y
Time of the start of the show 6: 13:4
Time of the end of the show 6: 14:25
Title of the show 7: u
Time of the start of the show 7: 3:53
Time of the end of the show 7: 9:50
Title of the show 8: i
Time of the start of the show 8: 22:3
Time of the end of the show 8: 3:56
Title of the show 9: o
Time of the start of the show 9: 18:55
Time of the end of the show 9: 19:42
Title of the show 10: p
Time of the start of the show 10: 17:46
Time of the end of the show 10: 21:10
Title of the show 11: l
Time of the start of the show 11: 8:42
Time of the end of the show 11: 15:3
Title of the show 12: k
Time of the start of the show 12: 10:19
Time of the end of the show 12: 15:27
Title of the show 13: j
Time of the start of the show 13: 4:13
Time of the end of the show 13: 6:26
Title of the show 14: h
Time of the start of the show 14: 2:54
Time of the end of the show 14: 7:1
Title of the show 15: g
Time of the start of the show 15: 16:38
Time of the end of the show 15: 24:7
Durations of shows:
The duration of show 1 is: 4:30
The duration of show 2 is: 8:13
The duration of show 3 is: 5:31
The duration of show 4 is: 4:14
The duration of show 5 is: 3:47
The duration of show 6 is: 1:21
The duration of show 7 is: 5:57
The duration of show 8 is: 5:53
The duration of show 9 is: 0:47
The duration of show 10 is: 3:24
The duration of show 11 is: 6:21
The duration of show 12 is: 5:8
The duration of show 13 is: 2:13
The duration of show 14 is: 4:7
The duration of show 15 is: 7:29
Day shows:
Title of the show 1: e
Time of the start of the show 1: 11:4
Time of the end of the show 1: 16:35
Title of the show 2: y
```

Microsoft Visual Studio Debug Console

```
Enter the title of the show: h
Enter the title of the show: g
Choose writing mode: 1 - for overwriting the file, 2 - for adding to the file: f
You entered a wrong answer. Please enter '1' or '2': 1
Shows:
Title of the show 1: q
Time of the start of the show 1: 18:24
Time of the end of the show 1: 22:54
Title of the show 2: w
Time of the start of the show 2: 1:35
Time of the end of the show 2: 9:48
Title of the show 3: e
Time of the start of the show 3: 11:4
Time of the end of the show 3: 16:35
Title of the show 4: r
Time of the start of the show 4: 8:53
Time of the end of the show 4: 13:7
Title of the show 5: t
Time of the start of the show 5: 3:24
Time of the end of the show 5: 7:11
Title of the show 6: y
Time of the start of the show 6: 13:4
Time of the end of the show 6: 14:25
Title of the show 7: u
Time of the start of the show 7: 3:53
Time of the end of the show 7: 9:50
Title of the show 8: i
Time of the start of the show 8: 22:3
Time of the end of the show 8: 3:56
Title of the show 9: o
Time of the start of the show 9: 18:55
Time of the end of the show 9: 19:42
Title of the show 10: p
Time of the start of the show 10: 17:46
Time of the end of the show 10: 21:10
Title of the show 11: l
Time of the start of the show 11: 8:42
Time of the end of the show 11: 15:3
Title of the show 12: k
Time of the start of the show 12: 10:19
Time of the end of the show 12: 15:27
Title of the show 13: j
Time of the start of the show 13: 4:13
Time of the end of the show 13: 6:26
Title of the show 14: h
Time of the start of the show 14: 2:54
Time of the end of the show 14: 7:1
Title of the show 15: g
Time of the start of the show 15: 16:38
Time of the end of the show 15: 24:7
Durations of shows:
The duration of show 1 is: 4:30
The duration of show 2 is: 8:13
The duration of show 3 is: 5:31
The duration of show 4 is: 4:14
The duration of show 5 is: 3:47
The duration of show 6 is: 1:21
The duration of show 7 is: 5:57
The duration of show 8 is: 5:53
The duration of show 9 is: 0:47
The duration of show 10 is: 3:24
The duration of show 11 is: 6:21
The duration of show 12 is: 5:8
The duration of show 13 is: 2:13
The duration of show 14 is: 4:7
The duration of show 15 is: 7:29
Day shows:
Title of the show 1: e
Time of the start of the show 1: 11:4
Time of the end of the show 1: 16:35
Title of the show 2: y
Time of the start of the show 2: 13:4
Time of the end of the show 2: 14:25
Title of the show 3: k
Time of the start of the show 3: 10:19
Time of the end of the show 3: 15:27

D:\docs\univer\on\on2\lab2\lab2\Debug\lab2.exe (process 12180) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Python

G:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe

```
Enter number of TV shows: 10
Enter the title of the show: q
Enter the title of the show: w
Enter the title of the show: e
Enter the title of the show: r
Enter the title of the show: t
Enter the title of the show: y
Enter the title of the show: u
Enter the title of the show: i
Enter the title of the show: o
Enter the title of the show: p
Choose writing mode: 1 - for overwriting the file, 2 - for adding to the file: 1
Shows:
Title of the show 1 :  q
Time of the start of the show 1 : 13 : 40
Time of the end of the show 1 : 22 : 31
Title of the show 2 :  w
Time of the start of the show 2 : 21 : 55
Time of the end of the show 2 : 24 : 25
Title of the show 3 :  e
Time of the start of the show 3 : 23 : 48
Time of the end of the show 3 : 8 : 37
Title of the show 4 :  r
Time of the start of the show 4 : 6 : 59
Time of the end of the show 4 : 13 : 42
Title of the show 5 :  t
Time of the start of the show 5 : 18 : 39
Time of the end of the show 5 : 23 : 4
Title of the show 6 :  y
Time of the start of the show 6 : 17 : 48
Time of the end of the show 6 : 21 : 34
Title of the show 7 :  u
Time of the start of the show 7 : 22 : 27
Time of the end of the show 7 : 7 : 17
Title of the show 8 :  i
Time of the start of the show 8 : 12 : 51
Time of the end of the show 8 : 13 : 6
Title of the show 9 :  o
Time of the start of the show 9 : 3 : 10
Time of the end of the show 9 : 9 : 54
Title of the show 10 :  p
Time of the start of the show 10 : 19 : 33
Time of the end of the show 10 : 20 : 56
Durations of shows:
The duration of show 1 is: 8 : 51
The duration of show 2 is: 2 : 30
The duration of show 3 is: 8 : 49
The duration of show 4 is: 6 : 43
The duration of show 5 is: 4 : 25
The duration of show 6 is: 3 : 46
The duration of show 7 is: 8 : 50
The duration of show 8 is: 0 : 15
The duration of show 9 is: 6 : 44
The duration of show 10 is: 1 : 23
Day shows:
Title of the show 1 :  i
Time of the start of the show 1 : 12 : 51
Time of the end of the show 1 : 13 : 6
Press any key to continue . . .
```