

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. Модульне
програмування»

«Файли даних. Бінарні файли»

Варіант __30__

Виконав студент _____ ІП-15 Розін Олексій Іванович _____
(шифр, прізвище, ім'я, по батькові)

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

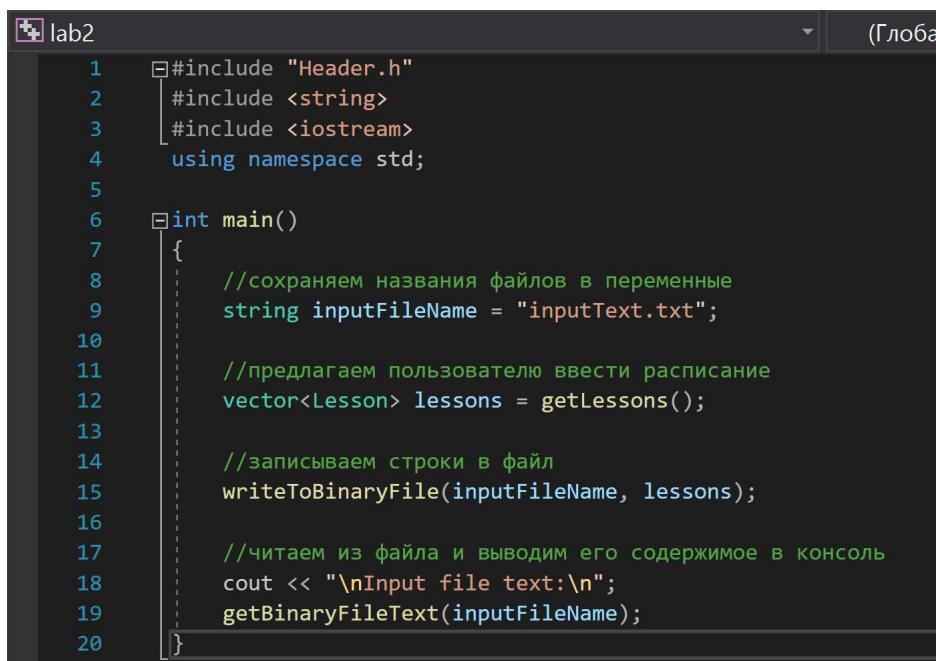
Київ 2021__

Завдання

30. Створити файл із розкладом занять у навчальному закладі протягом дня: назва предмета, час початку пари (за шаблоном ГГ:ХХ), час закінчення пари. Тривалість пари становить 1 год. 45 хв., а перерва між парами - не більше 45 хв. та не менше 5 хв. При введенні даних виконувати перевірку початку кожної пари (чи не накладається на попередню пару ти чи дотримані обмеження на тривалість перерви), а кінець пари розраховується автоматично.

Код C++

lab2.cpp



```
1 #include "Header.h"
2 #include <string>
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     //сохраняем названия файлов в переменные
9     string inputFileName = "inputText.txt";
10
11     //предлагаем пользователю ввести расписание
12     vector<Lesson> lessons = getLessons();
13
14     //записываем строки в файл
15     writeToBinaryFile(inputFileName, lessons);
16
17     //читаем из файла и выводим его содержимое в консоль
18     cout << "\nInput file text:\n";
19     getBinaryFileText(inputFileName);
20 }
```

Header.h

```
Header.h  Functions.cpp  lab2.cpp
lab2
1  #pragma once
2  #include <string>
3  #include <vector>
4  using namespace std;
5
6  struct Lesson {
7      char subjectName[21];
8      char startTime[6];
9      char endTime[6];
10 };
11
12 vector<Lesson> getLessons();
13 string getName();
14 string getStartTime();
15 bool isTime(string time);
16 void validateLessons(Lesson lesson, string startTime, vector<Lesson>& lessons);
17 int getMinutes(string time);
18 string getEndingTime(string time);
19 void writeToBinaryFile(string fileName, vector<Lesson> lessons);
20 void getBinaryFileText(string fileName);
```

Functions.cpp

```
Header.h  Functions.cpp  lab2.cpp
lab2
1  #include "Header.h"
2  #include <string>
3  #include <iostream>
4  #include <fstream>
5  #include <vector>
6  #include <iomanip>
7  using namespace std;
8
9  vector<Lesson> getLessons() {
10     int count;
11     vector<Lesson> lessons;
12
13     cout << "Enter how many lessons do you wanna add: "; cin >> count;
14     cin.ignore();
15     cout << "\n";
16
17     for (int i = 0; i < count; i++) {
18         Lesson mySubject;
19         strcpy_s(mySubject.subjectName, getName().c_str());
20         string startTime = getStartTime();
21         strcpy_s(mySubject.startTime, startTime.c_str());
22         size_t currentSize = lessons.size();
23         validateLessons(mySubject, startTime, lessons);
24         if (lessons.size() == currentSize) i--;
25     }
26     return lessons;
27 }
28
29 string getName() {
30     string name;
31     cout << "Enter name of the subject: "; cin >> name;
32     while (name.length() > 20 || name.length() < 1) {
33         cout << "Incorrect length of name, try one more time: "; cin >> name;
34     }
35     return name;
36 }
37
38 bool isTime(string time) {
39     if (time.length() > 5 || time.length() < 5) return false;
40     else if (time[2] != ':') return false;
41     else if (!isdigit(time[0]) || !isdigit(time[1]) || !isdigit(time[3]) || !isdigit(time[4])) return false;
42     else if (stoi(time.substr(0, 2)) > 23 || stoi(time.substr(0, 2)) < 0) return false;
43     else if (stoi(time.substr(3, 2)) > 59 || stoi(time.substr(3, 2)) < 0) return false;
44     return true;
45 }
46
47 string getStartTime() {
48     string startTime;
49     cout << "Enter starting time of the lesson in this format [HH:MM]: "; cin >> startTime;
50     while (!isTime(startTime)) {
51         cout << "Incorrect time format, try one more time: "; cin >> startTime;
52     }
53     return startTime;
54 }
```

```

56 void validateLessons(Lesson lesson, string startTime, vector<Lesson>& lessons) {
57     if (lessons.size() == 0) {
58         string ending = getEndingTime(startTime);
59         strcpy_s(lesson.endTime, ending.c_str());
60         lessons.push_back(lesson);
61     }
62     else {
63         int currMinutes = getMinutes(lesson.startTime);
64         int prevMinutes = getMinutes(lessons[lessons.size() - 1].endTime);
65         int minutesDiff = currMinutes - prevMinutes;
66         if (minutesDiff < 5 || minutesDiff > 45) {
67             cout << "Turn cannot be more than 45 minutes and less than 5 minutes!\n";
68         }
69         else {
70             string ending = getEndingTime(startTime);
71             strcpy_s(lesson.endTime, ending.c_str());
72             lessons.push_back(lesson);
73         }
74     }
75 }
76 }
77
78 int getMinutes(string time) {
79     size_t semicolonIndex = time.find(';');
80     int hours = stoi(time.substr(0, semicolonIndex));
81     int minutes = stoi(time.substr(semicolonIndex + 1));
82     return hours * 60 + minutes;
83 }
84
85 string getEndingTime(string time) {
86     int totalMinutesStart = getMinutes(time);
87     int totalMinutesEnd = totalMinutesStart + 105;
88     int hoursEnd = totalMinutesEnd / 60;
89     int minutesEnd = totalMinutesEnd % 60;
90     string newTime = to_string(hoursEnd) + ":";
91     newTime += minutesEnd == 0 ? "00" : to_string(minutesEnd);
92     return newTime;
93 }
94

```

```

98 void writeToBinaryFile(string fileName, vector<Lesson> lessons) {
99     ofstream file;
100     string ans;
101     cout << "Do you want to rewrite information (Y/N)? ";
102     cin >> ans;
103     while (ans != "Y" && ans != "y" && ans != "N" && ans != "n") {
104         cout << "Wrong, enter 'Y' or 'N': ";
105         cin >> ans;
106     }
107     if (ans == "Y" or ans == "y") file.open(fileName, ios::binary);
108     else file.open(fileName, ios::binary | ios::app);
109
110     for (size_t i = 0; i < lessons.size(); i++)
111     {
112         file.write((char*)&lessons[i], sizeof(Lesson));
113     }
114     file.close();
115 }
116
117
118 void getBinaryFileText(string fileName) {
119     Lesson subject{};
120     ifstream file(fileName, ios::binary);
121     while (file.read((char*)&subject, sizeof(Lesson))) {
122         string name = subject.subjectName;
123         string start = subject.startTime;
124         string end = subject.endTime;
125         cout << name + ", " + start + ", " + end << endl;
126     }
127     file.close();
128 }
129
130

```

Код Python

lab2Py.py

```
funcs.py lab2Py.py ↵ ✕  
  
# импортируем файл с функциями  
import funcs  
# сохраняем имя файла в переменную  
inputFileName = "inputTextPy.txt"  
  
# создаем список классов и преобразовываем его в список строк  
lessons = funcs.getLessons()  
  
# записываем строки в файл  
funcs.writeToFile(inputFileName, lessons)  
  
# читаем текст из файла и выводим его в консоль  
print('\nInput file text:')  
funcs.getFileText(inputFileName)
```

funcs.py

```
funcs.py ↵ ✕ lab2Py.py  
  
import pickle  
  
def getMinutes(time):  
    semicolonIndex = time.find(':')  
    hours = int(time[:semicolonIndex])  
    minutes = int(time[semicolonIndex+1:])  
    return hours * 60 + minutes  
  
def getEndingTime(time):  
    totalMinutesStart = getMinutes(time)  
    totalMinutesEnd = totalMinutesStart + 105  
    hoursEnd = totalMinutesEnd // 60  
    minutesEnd = totalMinutesEnd % 60  
    newTime = str(hoursEnd) + ':'  
    newTime += '00' if minutesEnd == 0 else str(minutesEnd)  
    return newTime  
  
def validateLesson(lesson, lessons):  
    if len(lessons) == 0:  
        lesson["endTime"] = getEndingTime(lesson["startTime"])  
        lessons.append(lesson)  
        return lessons  
    else:  
        currMinutes = getMinutes(lesson["startTime"])  
        prevMinutes = getMinutes(lessons[-1]["endTime"])  
        minutesDiff = currMinutes - prevMinutes  
        if minutesDiff < 5 or minutesDiff > 45:  
            print('\nTurn cannot be more than 45 minutes and less than 5 minutes!')  
            return lessons  
        else:  
            lesson["endTime"] = getEndingTime(lesson["startTime"])  
            lessons.append(lesson)  
            return lessons  
  
def getLessons():  
    lessons = []  
    count = int(input('Enter how many lessons do you wanna add: '))  
    i = 0  
    while i < count:  
        mySubject = {  
            "subjectName": getName(),  
            "startTime": getStartTime(),  
            "endTime": ""  
        }  
        currentLen = len(lessons)  
        lessons = validateLesson(mySubject, lessons)  
        if len(lessons) != currentLen: i += 1  
    return lessons  
  
def getName():  
    name = input('\nEnter name of the subject: ')  
    while len(name) < 1:  
        name = input('\nIncorrect length of name, try one more time: ')  
    return name
```

```

def isTime(time):
    time = time.strip()
    if len(time) > 5 or len(time) < 5: return False
    if time[2] != ':': return False
    elif not time[0].isdigit or not time[1].isdigit or not time[3].isdigit or not time[4].isdigit: return False
    elif int(time[:2]) > 23 or int(time[:2]) < 0: return False
    elif int(time[3:]) > 59 or int(time[3:]) < 0: return False
    return True

def getStartTime():
    startTime = input('Enter starting time of the lesson in this format [HH:MM]: ')
    while not isTime(startTime):
        startTime = input('Incorrect time format, try one more time: ')
    return startTime

def writeToFile(fileName, lessons):
    ans = input("Do you want to rewrite information (Y/N)? ")
    while ans != 'Y' and ans != 'y' and ans != 'N' and ans != 'n':
        ans = input("Wrong, enter 'Y' or 'N': ")

    if ans == 'Y' or 'y': file = open(fileName, 'wb')
    else: file = open(fileName, 'ab')
    pickle.dump(lessons, file)
    file.close()

def getFileText(fileName):
    file = open(fileName, 'rb')
    lessons = pickle.load(file)
    for lesson in lessons:
        res = lesson["subjectName"] + ", " + lesson["startTime"] + ", " + lesson["endTime"]
        print(res)
    file.close()

```

Консоль

C++

 Консоль отладки Microsoft Visual Studio

```

Enter how many lessons do you wanna add: 3

Enter name of the subject: math
Enter starting time of the lesson in this format [HH:MM]: 09;00
Incorrect time format, try one more time: 0900
Incorrect time format, try one more time: 09:00
Enter name of the subject: english
Enter starting time of the lesson in this format [HH:MM]: 11:15
Enter name of the subject: pe
Enter starting time of the lesson in this format [HH:MM]: 13:40
Do you want to rewrite information (Y/N)? Y

Input file text:
math, 09:00, 10:45
english, 11:15, 13:00
pe, 13:40, 15:25

```

Python

C:\WINDOWS\system32\cmd.exe

```
Enter how many lessons do you wanna add: 3

Enter name of the subject: pe
Enter starting time of the lesson in this format [HH:MM]: 09:30

Enter name of the subject: english
Enter starting time of the lesson in this format [HH:MM]: 12:000
Incorrect time format, try one more time: 12
Incorrect time format, try one more time: 12:00

Enter name of the subject: art
Enter starting time of the lesson in this format [HH:MM]: 13:555
Incorrect time format, try one more time: 13:55
Do you want to rewrite information (Y/N)? Y

Input file text:
pe, 09:30, 11:15
english, 12:00, 13:45
art, 13:55, 15:40
Для продолжения нажмите любую клавишу . . .
```

inputText.txt

inputText.txt – Блокнот

Файл Правка Формат Вид Справка

math юююююююююююююююю09:00 10:45 english юююююююююююююююю11:15 13:00 pe юююююююююююююююююююююю13:40 15:25

inputTextPy.txt

inputTextPy.txt – Блокнот

Файл Правка Формат Вид Справка

{}(\\subjectName"ь pe"ь startTime"\\09:30"\\endTime"\\11:15"u}"(h \\english"\\12:00"\\13:45"u}"(h \\art"\\13:55"\\15:40"ue.

Висновок

Я навчився особливостям створення і обробки бінарних файлів даних і набув практичних навичок під час виконання лабораторної роботи, а саме створив бінарний файл програмних шляхом з можливістю користувача записати туди розклад уроків, при цьому контролюючи, щоб розклад уроків не перетинався один з одним.