Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІПІ

ЗВІТ

з виконання лабораторної роботи № 2

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 26

Виконав:

студент 1-го курсу

гр. ІП-21 ФІОТ

Скрипець Ольга Олександрівна

Київ 2023

1. **Посилання на репозиторій**

https://github.com/uhpolerr/lab2

1. **Код програми**

**Files.h**

#pragma once

#include <iostream>

#include <fstream>

#include <string>

#include <conio.h>

#include <cstdio>

#include <algorithm>

#include <regex>

#include <sstream>

using namespace std;

struct PhoneCall {

char date[11];

char startTime[9];

char endTime[9];

double duration;

float cost;

};

bool isLeapYear(int year);

bool isValidDate(int year, int month, int day);

bool isValidTime(int hour, int minute, int second);

void tariff();

void enterDay(PhoneCall& call, ofstream& outFile);

void enterStartTime(PhoneCall& call, ofstream& outFile);

void enterEndTime(PhoneCall& call, ofstream& outFile);

bool continueInputFunc();

void printHeaderTable();

double calculateCallDuration(PhoneCall call);

float function1(double startCallMinute, double allTimeCall, int index);

float calculateCallCost(PhoneCall call);

**Files.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

#include <string>

#include <ctime>

#include <ctime>

#include <Windows.h>

using namespace std;

static float tariffs[] = { 0, 1.19, 1.7, 1.19, 0 };

static float tariffsEndHou[] = { 420, 540, 1080, 1320, 1440};

struct PhoneCall {

char date[11];

char startTime[9];

char endTime[9];

double duration;

float cost;

};

// Функція для перевірки чи рік є високосним

bool isLeapYear(int year) {

return (year % 4 == 0 && year % 100 != 0) || year % 400 == 0;

}

// Функція для перевірки чи дата валідна

bool isValidDate(int year, int month, int day) {

if (month < 1 || month > 12) {

return false;

}

int daysInMonth;

if (month == 2) {

daysInMonth = isLeapYear(year) ? 29 : 28;

}

else if (month == 4 || month == 6 || month == 9 || month == 11) {

daysInMonth = 30;

}

else {

daysInMonth = 31;

}

if (day < 1 || day > daysInMonth) {

return false;

}

if (year < 1876) {

return false;

}

if (year > 2023 && month > 4 && day > 7) {

cout << "Ви що дзвонили у майбутнє?";

return false;

}

return true;

}

// Функція для перевірки чи час валідний

bool isValidTime(int hour, int minute, int second) {

if (hour < 0 || hour > 23 || minute < 0 || minute > 59 || second < 0 || second > 59) {

return false;

}

return true;

}

//Функція для вводу дати

void enterDay(PhoneCall& call, ofstream& outFile) {

int day, month, year;

bool isValid = false;

do {

cout << "Введіть дату у форматі DD/MM/YYYY: ";

cin >> day;

cin.ignore();

cin >> month;

cin.ignore();

cin >> year;

cin.ignore();

isValid = isValidDate(year, month, day);

if (!isValid) {

cout << "Введіть коректну дату!" << endl;

}

} while (!isValid);

sprintf(call.date, "%02d/%02d/%04d", day, month, year);

}

//Функція для вводу початкового часу

void enterStartTime(PhoneCall& call, ofstream& outFile) {

int hour, minute, second;

bool isValid = false;

do {

cout << "Введіть час початку розмови у форматі HH:MM:SS ";

cin >> hour;

cin.ignore();

cin >> minute;

cin.ignore();

cin >> second;

cin.ignore();

isValid = isValidTime(hour, minute, second);

if (!isValid) {

cout << "Введіть коректний час!" << endl;

}

} while (!isValid);

sprintf(call.startTime, "%02d:%02d:%02d", hour, minute, second);

}

//Функція для кінцевого часу

void enterEndTime(PhoneCall& call, ofstream& outFile) {

int hour, minute, second;

bool isValid = false;

do {

cout << "Введіть час закінчення розмови у форматі HH:MM:SS ";

cin >> hour;

cin.ignore();

cin >> minute;

cin.ignore();

cin >> second;

isValid = isValidTime(hour, minute, second);

if (!isValid) {

cout << "Введіть коректний час!" << endl;

}

} while (!isValid);

sprintf(call.endTime, "%02d:%02d:%02d", hour, minute, second);

}

//Функція для продовження вводу телефонних розмов

bool continueInputFunc() {

char answer;

cout << endl;

cout << "Продовжити ввід? (y/n): ";

cin >> answer;

cout << endl;

if (answer != 'y' && answer != 'Y') {

return false;

}

cin.ignore();

return true;

}

//Функція для виводу заголовку таблиці

void printHeaderTable() {

cout << "-------------------------------------------------------------------------------" << endl;

cout << "| Дата | Час початку | Час закінчення | Тривалість (хв) | Вартість (грн) |" << endl;

cout << "------------|-------------|----------------|-----------------|-----------------" << endl;

}

//Функція для виводу тарифу

void tariff() {

cout << "-------------------------------------------------------------------------------" << endl;

cout << "| Вартість розмов: |" << endl;

cout << "| 09:00-18:00 1.7 грн/хв 22:00-07:00 0 грн/хв |" << endl;

cout << "| 07:00-09:00 та 18:00-22:00 1.19 грн/хв |" << endl;

cout << "-------------------------------------------------------------------------------" << endl;

cout << endl;

}

//Функція для розрахунку хвилин

double calculateCallDuration(PhoneCall call) {

int startHour, startMinute, startSecond, endHour, endMinute, endSecond;

double durationInSeconds;

sscanf(call.startTime, "%d:%d:%d", &startHour, &startMinute, &startSecond);

sscanf(call.endTime, "%d:%d:%d", &endHour, &endMinute, &endSecond);

if (startHour < endHour) {

durationInSeconds = (endHour \* 3600 + endMinute \* 60 + endSecond) - (startHour \* 3600 + startMinute \* 60 + startSecond);

}

else {

durationInSeconds = (endHour \* 3600 + endMinute \* 60 + endSecond) + ((23-startHour) \* 3600 + (59-startMinute) \* 60 + (60-startSecond));

}

double durationInMinutes = durationInSeconds / 60;

return durationInMinutes;

}

//Рекурсивна функція для розрахунку вартості

float function1(double startCallMinute, double allTimeCall, int index) {

if (tariffsEndHou[index]- startCallMinute < allTimeCall) {

double sum = (tariffsEndHou[index] - startCallMinute) \* tariffs[index];

allTimeCall -= tariffsEndHou[index] - startCallMinute;

if (index == 4) {

index = 0;

startCallMinute = 0;

}

else {

startCallMinute = tariffsEndHou[index];

index += 1;

}

return sum + function1(startCallMinute, allTimeCall, index);

}

return allTimeCall\* tariffs[index];

}

//Функція для розрахунку вартості

float calculateCallCost(PhoneCall call) {

float startHour, startMinute, startSecond, endHour, endMinute, endSecond;

float cost=0;

double minute = calculateCallDuration(call);

sscanf(call.startTime, "%f:%f:%f", &startHour, &startMinute, &startSecond);

sscanf(call.endTime, "%f:%f:%f", &endHour, &endMinute, &endSecond);

double startCallMinute = startHour \* 60 + startMinute + startSecond / 60;

if (startHour < 7 || startHour>22) {

if (startHour < 7) {

return (double)function1(startCallMinute, minute, 0);

}

else {

return (double)function1(startCallMinute, minute, 4);

}

}

else if ((startHour>7 && startHour<9) || (startHour>18 && startHour<22)) {

if (startHour > 7 && startHour < 9) {

return (double)function1(startCallMinute, minute, 1);

}

else {

return (double)function1(startCallMinute, minute, 3);

}

}

else {

double cost = (double)function1(startCallMinute, minute, 2);

return cost;

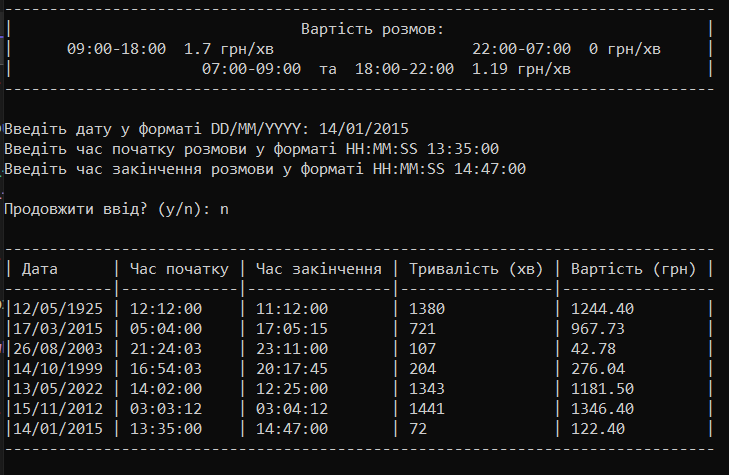
}

}

**Main.cpp**

1. #define \_CRT\_SECURE\_NO\_WARNINGS
2. #include <iostream>
3. #include <fstream>
4. #include <string>
5. #include <Windows.h>
6. #include "Files.h"
7. using namespace std;
8. int main() {
9. setlocale(LC\_CTYPE, "ukr");
10. SetConsoleCP(1251);
11. SetConsoleOutputCP(1251);
12. tariff();
13. ofstream outFile("phonecalls.bin", ios::app | ios::binary);
14. if (!outFile) {
15. cerr << "Не вдалося відкрити файл!" << endl;
16. return 1;
17. }
18. PhoneCall call;
19. bool continueInput = true;
20. while (continueInput) {
21. int year{}, month{}, day{}, hour{}, minute{}, second{};
22. enterDay(call, outFile);
23. enterStartTime(call, outFile);
24. enterEndTime(call, outFile);
25. call.duration = calculateCallDuration(call);
26. call.cost = calculateCallCost(call);
27. outFile.write(reinterpret\_cast<const char\*>(&call), sizeof(call));
28. continueInput = continueInputFunc();
29. }
30. outFile.close();
31. ifstream inFile("phonecalls.bin", ios::in | ios::binary);
32. if (!inFile) {
33. cerr << "Не вдалося відкрити файл!" << endl;
34. return 1;
35. }
36. printHeaderTable();
37. while (inFile.read(reinterpret\_cast<char\*>(&call), sizeof(PhoneCall))) {
38. printf("|%-11s| %-11s | %-14s | %-15.0f | %-12.2f |\n", call.date, call.startTime, call.endTime, call.duration, call.cost);
39. }
40. inFile.close();
41. cout << "-------------------------------------------------------------------------------" << endl;
42. cin.ignore();
44. return 0;
45. }

**Перевірки**

****