

## Лабораторна робота 7

Тема: Опрацювання даних типу "дата-час" у текстових файлах

Мета роботи: набути практичних навиків програмного опрацювання даних типу "дата-час".

### Хід роботи

#### Завдання лабораторної роботи

1. Розглянути та виконати приклад 1, приклад 2 та приклад 3 опрацювання дат і часу
2. Написати програмний код мовою C++ для формування текстового файла даними (вміст файла зазначено у табл. 10.1), переглядання вмісту сформованого файла та розв'язання індивідуальних завдань з відбору тих записів файла, які задовольняють певній умові (завдання вибрati з табл. 7.1).

Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм.

5	<p>Журнал подій операційної системи: назва запущеної програми, рівень події (помилка, попередження тощо), дата події, час події</p>	<p>Вивести дані про помилки та визначити скільки днів пройшло від кожної з помилок на поточний момент часу</p>
---	---	--

#### Лістинг 7.1 – Код програми

```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
    setlocale(LC_ALL, "ukr");
    int day, month, year;
    cout << "Введіть дату кінцевого терміну придатності (спочатку день, потім місяць, рік): ";
    cin >> day >> month >> year;
    time_t t = time(0);
    tm* current = localtime(&t);
    tm expiry = {};
    expiry.tm_mday = day;
    expiry.tm_mon = month - 1;
    expiry.tm_year = year - 1900;
```

```

    time_t t_expiry = mktime(&expiry);
    double seconds_diff = difftime(t_expiry, t);
    int days = (int)(seconds_diff / (60 * 60 * 24));
    if (days > 0) {
        cout << "До закінчення терміну придатності залишилось: " <<
days << " днів.\n";
    }
    else if (days == 0) {
        cout << "Термін придатності закінчується сьогодні!\\n";
    }
    else {
        cout << "Термін придатності минув вже: " << -days << " днів
тому.\n";
    }

    system("pause");
    return 0;
}

```

Введіть дату кінцевого терміну придатності (день місяць рік): 20 11 2025  
До закінчення терміну придатності залишилось: 20 днів.

Рисунок 7.1 — Результат виконання прикладу 1

### Лістинг 7.2 – Код програми

```

#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
#include <climits>
#include <clocale>
#include <ctime>
using namespace std;
struct Friend {
    string surname;
    int day, month, year;
};

void createFile(const string& filename) {
    ofstream fout(filename);
    if (!fout.is_open()) {
        cerr << "Помилка відкриття файлу для запису: " << filename <<
endl;
        return;
    }
    fout << "Новак 12 5 2000" << endl;
    fout << "Коваль 21 3 2001" << endl;
    fout << "Войцік 28 3 2001" << endl;
    fout << "Адамські 28 3 2001" << endl;
    fout << "Римарчук 20 5 2000" << endl;
}

```

```

        fout << "Іванов 1 1 1995" << endl;
        fout.close();
    }

vector<Friend> readFromFile(const string& filename) {
    vector<Friend> list;
    ifstream fin(filename);
    if (!fin.is_open()) {
        cerr << "Помилка відкриття файлу для читання: " << filename <<
endl;
        return list;
    }
    Friend f;
    while (fin >> f.surname >> f.day >> f.month >> f.year) {
        list.push_back(f);
    }
    fin.close();
    return list;
}
bool isOlder(const Friend& a, const Friend& b) {
    if (a.year != b.year) {
        return a.year < b.year;
    }
    if (a.month != b.month) {
        return a.month < b.month;
    }
    return a.day < b.day;
}
int main() {
    setlocale(LC_ALL, "ukr");
    const string filename = "friends.txt";
    createFile(filename);
    vector<Friend> friends = readFromFile(filename);
    if (friends.empty()) {
        cout << "Список друзів порожній. Немає даних для обробки." <<
endl;
        return 1;
    }
    auto oldest_it = min_element(friends.begin(), friends.end(),
isOlder);
    cout << "## Найстаріші з народжених друзів: ##" << endl;
    cout << "-----"
<< endl;
    if (oldest_it != friends.end()) {
        const Friend& oldest = *oldest_it;
        cout << " " << oldest.surname << " * " << oldest.day << "."
<< oldest.month << "." << oldest.year << endl;
    }
    cout << "-----"
<< endl;
}

```

```

cout << "\n## Народжені друзі, які народилися у березні місяці
(3): ##" << endl;
cout << "-----"
<< endl;
bool foundMarch = false;
for (const auto& f : friends) {
    if (f.month == 3) {
        cout << " " << f.surname << " *" << f.day << "." <<
f.month << "." << f.year << endl;
        foundMarch = true;
    }
}
if (!foundMarch) {
    cout << " Друзів, народжених у березні, не знайдено." <<
endl;
}
cout << "-----"
<< endl;
return 0;
}

```

```

## Найстар?ш? з народжених друз?в: ##
-----
?ванов * 1.1.1995
-----

## Народжен? друз?, як? народилися у березн? м?сяц? (3): ##
-----
Коваль * 21.3.2001
Войц?к * 28.3.2001
Адамськ? * 28.3.2001
-----
```

Рисунок 7.2 — Результат виконання прикладу 3

## Лістинг 7.3 – Код програми

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <ctime>
#include <iomanip>
#include <sstream>
#include <climits>
using namespace std;
struct Event {
    string program;
    string level;
```

```

        int day, month, year;
        int hour, minute;
    };
    void createFile(const string& filename) {
        ofstream fout(filename);
        if (!fout.is_open()) return;
        fout << "Explorer Info 05.11.2025 10:30\n";
        fout << "Chrome Error 01.11.2025 09:15\n";
        fout << "Word Warning 28.10.2025 16:50\n";
        fout << "Discord Error 25.10.2025 21:00\n";
        fout << "Updater Info 06.11.2025 11:05\n";
        fout << "System Error 30.10.2025 08:00\n";
        fout.close();
        cout << "Файл '" << filename << "' створено.\n";
    }
    vector<Event> readFromFile(const string& filename) {
        vector<Event> events;
        ifstream fin(filename);
        if (!fin.is_open()) return events;

        string line;
        while (getline(fin, line)) {
            stringstream ss(line);
            Event e;
            string dateStr, timeStr;
            if (!(ss >> e.program >> e.level >> dateStr >> timeStr)) continue;
            if (sscanf(dateStr.c_str(), "%d.%d.%d", &e.day, &e.month, &e.year) != 3) continue;
            if (sscanf(timeStr.c_str(), "%d:%d", &e.hour, &e.minute) != 2) continue;

            events.push_back(e);
        }
        fin.close();
        return events;
    }
    int daysSince(int d, int m, int y) {
        tm eventDate = {};
        eventDate.tm_mday = d;
        eventDate.tm_mon = m - 1;
        eventDate.tm_year = y - 1900;

        time_t t_event = mktime(&eventDate);
        time_t t_now = time(nullptr);
        if (t_event == (time_t)-1) {
            return -1;
        }

        double diff_sec = difftime(t_now, t_event);
        return static_cast<int>(diff_sec / (60.0 * 60.0 * 24.0));
    }
}

```

```

int main() {
    setlocale(LC_ALL, "ukr");
    const string filename = "events.txt";
    createFile(filename);
    vector<Event> events = readFromFile(filename);
    if (events.empty()) {
        cerr << "Файл порожній або не прочитано жодного запису.\n";
        return 1;
    }
    cout << "\n## Дані про помилки (Error): ##\n";
    cout << "-----\n";
    for (const auto& e : events) {
        if (e.level == "Error") {
            int days = daysSince(e.day, e.month, e.year);
            cout << " | " << left << setw(15) << e.program
                << " | " << setw(7) << e.level
                << " | " << setfill('0') << setw(2) << e.day << "."
                << setw(2) << e.month << "." << e.year
                << " " << setw(2) << e.hour << ":" << setw(2) << e.minute
                << setfill(' ') << " | Пройшло: " << days << " днів\n";
        }
    }
    cout << "-----\n";

    return 0;
}

```

```

Файл events.txt створено.

Дані про помилки (Error):
Chrome | Error | 01.11.2025 09:15 | Пройшло: 7 днів
Discord | Error | 25.10.2025 21:00 | Пройшло: 14 днів
System | Error | 30.10.2025 08:00 | Пройшло: 9 днів
Press any key to continue . . .

```

Рисунок 7.9 — Результат виконання завдання

Висновок: на цій лабораторній роботі я набувал практичних навиків програмного опрацювання даних типу