

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет

ЗВІТ  
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 11  
з навчальної дисципліни  
“Базові методології та технології програмування”

ЗАВДАННЯ ВИДАВ  
доцент кафедри кібербезпеки  
та програмного забезпечення  
Доренський О. П.  
<https://github.com/odorenskyi/>

ВИКОНАЛА  
студентка академічної групи  
КН-22  
Пилипенко В.С.

ПЕРЕВІРИВ  
викладач кафедри кібербезпеки  
та програмного забезпечення  
Олександр Собінов

**Тема:** Реалізація програмних засобів оброблення динамічних структур даних та бінарних файлів

**Мета роботи** полягає у набутті ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам'яттю та бінарними файловими потоками.

**Завдання:**

У складі команди ІТ-проекта розробити програмні модулі оброблення динамічної структури даних. Реалізувати програмний засіб на основі розроблених командою ІТ-проекта модулів.

**Склад команди:**

Кучер Денис КН-22

Скок Кирило КН-22

Пилипенко Вікторія КН-22

**Варіант № 12**

**ВАРІАНТ 12**

— ЗАВДАННЯ НА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ —

Створити **електронний довідник** пільг, наданих чинним законодавством по сплаті податків, зборів, інших обов'язкових платежів (Код пільги, Зміст пільги, Документ, № документа, Дата прийняття документа, Початок дії пільги, Кінець дії пільги, Ознака цільового призначення, Коментар).

За вибором користувача (оператора) програма забезпечує:

- пошук запису в довіднику за введеним кодом пільги;
- виведення всього довідника на екран або у текстовий файл;
- додавання нового запису в довідник;
- вилучення заданого запису з довідника;
- завершення роботи програми з автоматичним записом даних у файл.

Дані довідника автоматично завантажуються з файлу під час запуску програми.

## Лістинг головного файлу «main.cpp»

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
#include <Windows.h>

#include "../ModulesPylypenko.h"
#include "../ModulesKucher.h"
#include "../ModulesSkok.h"
#include "../struct_type_project_12.h"

using namespace std;

int main()
{
    vector <Benefits> note;
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    Benefits notes; // Створення нової структури Benefits для
збереження даних з рядка

    ifstream file("dividnuk.txt"); // Відкриття файлу
"dividnuk.txt" для зчитування
    if (file.is_open()) { // Перевірка, чи відкриття файлу
    пройшло успішно
        string line; // Змінна для збереження рядка зчитаного з
    файлу
        while (std::getline(file, line)) { // Цикл, який зчитує
    рядки з файлу по одному

            std::stringstream ss(line); // Створення потоку
    stringstream для подальшого розбиття рядка на окремі частини
            string item; // Змінна для збереження окремих частин
    рядка

            std::getline(ss, item, ','); // Розбиття рядка за
    допомогою ',' як роздільника і збереження першої частини в змінну item
            notes.code_ben = std::stoi(item); // Перетворення
    змінної item з рядкового типу в ціле число і збереження в поле
    code_ben структури notes

            std::getline(ss, notes.content_ben, ',');
            std::getline(ss, notes.document_ben, ',');
            std::getline(ss, notes.num_doc_ben, ',');
            std::getline(ss, notes.date_doc_ben, ',');
            std::getline(ss, notes.date_start_ben, ',');
            std::getline(ss, notes.date_end_ben, ',');
            std::getline(ss, notes.destination_sign_ben, ',');
            std::getline(ss, notes.coment_ben, ',');

            note.push_back(notes); // Додавання структури notes до
    вектору note
```

```

    }
    file.close();
} else {
    cout << "Файл не знайдено.\n";
    return 1;
}

int choice;
do {
    std::cout << "Оберіть запит:\n";
    std::cout << "1. Пошук по коду пільги\n";
    std::cout << "2. Виведення всього довідника\n";
    std::cout << "3. Додання нового запису в довідник\n";
    std::cout << "4. Видалення запису з довідника\n";
    std::cout << "5. Завершення роботи та запис даних у
файл\n";

    std::cin >> choice;

    switch (choice) {
        case 1:
            int code;
            cout << "Введіть код пільги: ";
            cin >> code;
            poshuk_kod(note, code);
            break;
        case 2:
            cout << "Натисніть 0 (увесь довідник у файл) чи 1 (на
екран) ";

            int button;
            cin >> button;
            if (button == 0) {
                cout << "Введіть ім'я в файл який хочете
записати довідник";

                string filename;
                cin >> filename;
                filename = filename + ".txt";
                File(note, filename);
            }
            else if (button == 1) {
                Screen(note);
            }

            break;
        case 3:
            add_new_data(note, "dividnuk.txt");
            break;
        case 4:
            delete_data(note, "dividnuk.txt");
            break;
        case 5:
            end_file(note, "dividnuk.txt");
            break;
        default:
            cout << "Неправильна опція. Спробуйте знову.\n";
    }
}

```

```

        break;
    }
} while (choice != 5);

return 0;
}

```

### Лістинг файлу «ModulesPylypenko.h»:

```

#ifndef MODULESPYLYPENKO_H_INCLUDED
#define MODULESPYLYPENKO_H_INCLUDED

#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
#include <Windows.h>
#include "struct_type_project_12.h"

void add_new_data(vector<Benefits>& note, const string& file_name)
{
    Benefits new_data;
    cout << "Введіть дані нового запису:\n";
    cout << "Код пільги: ";
    cin >> new_data.code_ben;
    cout << "Зміст пільги: ";
    cin.ignore(); // Ignore the newline character
    getline(cin, new_data.content_ben);
    cout << "Тип документу: ";
    getline(cin, new_data.document_ben);
    cout << "Номер документу: ";
    getline(cin, new_data.num_doc_ben);
    cout << "Дата документу: ";
    getline(cin, new_data.date_doc_ben);
    cout << "Дата початку дії: ";
    getline(cin, new_data.date_start_ben);
    cout << "Дата закінчення дії: ";
    getline(cin, new_data.date_end_ben);
    cout << "Ознака цільового призначення: ";
    getline(cin, new_data.destination_sign_ben);
    cout << "Коментар: ";
    getline(cin, new_data.coment_ben);
    note.push_back(new_data);

    // Add the new data to the file
    ofstream file("dividnuk.txt", ios::app); // Open the file in
append mode
    if (file.is_open()) {
        file << new_data.code_ben << ","
        << new_data.content_ben << ","
        << new_data.document_ben << ","
        << new_data.num_doc_ben << ","
        << new_data.date_doc_ben << ","

```

```

        << new_data.date_start_ben << ", "
        << new_data.date_end_ben << ", "
        << new_data.destination_sign_ben << ", "
        << new_data.coment_ben << "\n";
    cout << "Новий запис успішно додано до файлу " <<
"dividnuk.txt" << ".\n";
    file.close();
} else {
    cout << "Помилка при відкритті файлу для запису.\n";
}
}

void delete_data(vector<Benefits>& note, const string& file_name)
{
    if (note.empty()) {
        cout << "Немає даних для видалення.\n";
        return;
    }

    cout << "Список записів:\n";
    for (size_t i = 0; i < note.size(); i++) {
        cout << i + 1 << ". " << note[i].code_ben << " - " <<
note[i].content_ben << "\n";
    }

    int choice;
    cout << "Виберіть номер запису для видалення: ";
    cin >> choice;

    if (choice < 1 || choice > note.size()) {
        cout << "Невірний номер запису.\n";
        return;
    }

    note.erase(note.begin() + choice - 1);
}

#endif MODULESPYLYPENKO_H_INCLUDED

```

### Лістинг файлу «ModulesKucher.h»:

```

#ifndef MODULESKUCHER_H_INCLUDED
#define MODULESKUCHER_H_INCLUDED

#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
#include <Windows.h>
#include "struct_type_project_12.h"

using namespace std;

```

```

void poshuk_kod(const vector<Benefits>& note, int code) {
    for (const auto& notes : note) {
        if (notes.code_ben == code) {
            // Вивід
            cout << "Пільга знайдена:\n";
            cout << "Код пільги: " << notes.code_ben << "\n";
            cout << "Зміст пільги: " << notes.content_ben << "\n";
            cout << "Документ пільги: " << notes.document_ben <<
"\n";
            cout << "Номер документа: " << notes.num_doc_ben <<
"\n";
            cout << "Дата документа: " << notes.date_doc_ben <<
"\n";
            cout << "Дата початку: " << notes.date_start_ben <<
"\n";
            cout << "Дата кінця дії: " << notes.date_end_ben <<
"\n";
            cout << "Ознака змісту: " <<
notes.destination_sign_ben << "\n";
            cout << "Коментарій: " << notes.coment_ben << "\n";
            return;
        }
    }
    cout << "Пільга з таким кодом " << code << " не знайдена.\n";
}

#endif // MODULESKUCHER_H_INCLUDED

```

### Лістинг файлу «ModulesSkok.h»:

```

#ifndef MODULESKOK_H_INCLUDED
#define MODULESKOK_H_INCLUDED
#include<iostream>
#include<fstream>
#include "struct_type_project_12.h"

void one_benefit(const Benefits& benefit)
{
    cout<<"Code:"<<benefit.code_ben<<"\n";
    cout<<"Content:"<<benefit.content_ben<<"\n";
    cout<<"Document:"<<benefit.document_ben<<"\n";
    cout<<"Number Document:"<<benefit.num_doc_ben<<"\n";
    cout<<"Date document:"<<benefit.date_doc_ben<<"\n";
    cout<<"Date start:"<<benefit.date_start_ben<<"\n";
    cout<<"Date end:"<<benefit.date_end_ben<<"\n";
    cout<<"Destination sign:"<<benefit.destination_sign_ben<<"\n";
    cout<<"Coment:"<<benefit.coment_ben<<"\n";
}

void Screen(const vector<Benefits>& note)
{
    for (const Benefits& benefit : note)
    {
        one_benefit(benefit);
    }
}

```

```

void File(const vector<Benefits>& note,string filename)
{
    ofstream outFile(filename);
    if (outFile.is_open())
    {
        for (const Benefits& benefit : note)
        {
            outFile << "Code: " << benefit.code_ben << "\n";
            outFile << "Content: " << benefit.content_ben <<
"\n";
            outFile << "Document: " << benefit.document_ben <<
"\n";
            outFile << "Number Document: " <<
benefit.num_doc_ben << "\n";
            outFile << "Date document: " <<
benefit.date_doc_ben << "\n";
            outFile << "Date start: " <<
benefit.date_start_ben << "\n";
            outFile << "Date end: " << benefit.date_end_ben <<
"\n";
            outFile << "Destination sign: " <<
benefit.destination_sign_ben << "\n";
            outFile << "Comment: " << benefit.coment_ben <<
"\n\n";
        }
        outFile.close();
    }
    else
    {
        cout << "Помилка при відкритті файла.\n";
    }
}

void end_file(const vector<Benefits>& note,string filename)
{
    ofstream outFile(filename);
    if (outFile.is_open())
    {
        for (const Benefits& benefit : note)
        {
            outFile << "Code: " << benefit.code_ben << "\n";
            outFile << "Content: " << benefit.content_ben <<
"\n";
            outFile << "Document: " << benefit.document_ben <<
"\n";
            outFile << "Number Document: " <<
benefit.num_doc_ben << "\n";
            outFile << "Date document: " <<
benefit.date_doc_ben << "\n";
            outFile << "Date start: " <<
benefit.date_start_ben << "\n";
            outFile << "Date end: " << benefit.date_end_ben <<
"\n";
            outFile << "Destination sign: " <<
benefit.destination_sign_ben << "\n";

```



```

        outFile << "Comment: " << benefit.coment_ben <<
"\n\n";
    }
    outFile.close();
}
else
{
    cout << "Помилка при відкритті файла.\n";
}
}
#endif // MODULESKOK_H_INCLUDED

```

### **Лістинг файлу «struct\_type\_project\_12.h»:**

```

#ifndef STRUCT_TYPE_PROJECT_12_H_INCLUDED
#define STRUCT_TYPE_PROJECT_12_H_INCLUDED

#include <string>
#include <vector>
#include <iostream>
#include <fstream>

using namespace std;

struct Benefits
{
    int code_ben;
    string content_ben;
    string document_ben;
    string num_doc_ben;
    string date_doc_ben;
    string date_start_ben;
    string date_end_ben;
    string destination_sign_ben;
    string coment_ben;
};

#endif // STRUCT_TYPE_PROJECT_12_H_INCLUDED

```

### **Висновок:**

В ході виконання лабораторної роботи я більше попрактикувалася з використанням функцій і створення бібліотеки модулів, що дозволить ефективніше вирішувати поставлені задачі для створення програми. Бібліотеки модулів дають змогу винести функції, які використовує програма у окремий файл, таким чином різні люди можуть додавати елементи програми, як функції.

Ця лабораторна робота була пов'язана з читанням та записом даних у файли. Завдання цієї теми були орієнтовані на читання та запис у файли. Тому я змогла більше зрозуміти запис та зчитування файлів у чому дуже допомогли завдання різного

виду, які дозволили більше ретельно дослідити та попрактикуватися в цьому. Крім того, я дізналася про різні види кодування тексту та про те, які проблеми можуть виникнути при роботі з файлами, якщо не враховувати кодування. Окрім того, я зрозуміла, що використання функцій і бібліотек модулів дозволяє зберігати код більш структурованим і організованим способом, що покращує роботу над проектами та спрощує їх підтримку.

Також ця робота була особливою тим, що вона була командно спрямована, тож необхідно було зібрати команду і вже з нею пропрацьовувати складові програми, та реалізацію. Це є дуже корисним, дивлячись, що працюють програмісти у команді.

Загалом, ця тема дала мені можливість практично відчувати, як працюють різні аспекти роботи з файлами та як ефективно використовувати функції і бібліотеки модулів для створення більш структурованого та ефективного коду.

## Додаток:

Назва тестового набору Test Suite Description	TestSuite 11
Назва проекту / ПЗ Name of Project / Software	Prj_Pylypenko_12.exe
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сюїта Test Suite Author	Пилипенко Вікторія Кучер Денис Скок Кирило
Виконавець Implementer	Пилипенко Вікторія Кучер Денис Скок Кирило

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
1.	<ol style="list-style-type: none"> <li>Обираємо варіант з запропонованих</li> <li>Обрали «Додати до файлу»</li> <li>Заповнюємо поле</li> <li>Завершуємо програму</li> </ol>	<p>Введіть дані нового запису:</p> <p>Код пільги: 1</p> <p>Зміст пільги: На пологи</p> <p>Тип документу: Документ</p> <p>Номер документу: 45</p> <p>Дата документу: 12.06.2020</p> <p>Дата початку дії: 20.06.2023</p> <p>Дата закінчення дії: 31.12.2023</p> <p>Ознака цільового призначення: -</p> <p>Коментар: Дякую.</p> <p>Новий запис успішно додано до файлу dividnuk.txt.</p>	Passed
2.	<ol style="list-style-type: none"> <li>Обираємо варіант з запропонованих</li> <li>Обрали «Видалити з документу»</li> <li>Написали що саме хочемо видалити</li> <li>Завершили програму</li> </ol>	<p>Список записів:</p> <ol style="list-style-type: none"> <li>123 - Пільга на освіту</li> <li>456 - Пільга на лікарняний</li> <li>789 - Пільга на проїзд</li> <li>124 - 124</li> <li>1 - На пологи</li> </ol> <p>Виберіть номер запису для видалення: 4</p>	Passed
3.	<ol style="list-style-type: none"> <li>Обираємо варіант з запропонованих</li> <li>Обрали «Пошук по коду»</li> <li>Написали код пільги</li> <li>Знайшли</li> <li>Завершили роботу</li> </ol>	<p>Введіть код пільги: 1</p> <p>Пільга знайдена:</p> <p>Код пільги: 1</p> <p>Зміст пільги: На пологи</p> <p>Документ пільги: Документ</p> <p>Номер документа: 45</p> <p>Дата документа: 12.06.2020</p> <p>Дата початку: 20.06.2023</p> <p>Дата кінця дії: 31.12.2023</p> <p>Ознака змісту: -</p>	Passed

		Коментарій: Дякую.	
4.	1. Обираємо варіант з запропонованих 2. Обрали «Виведення всього довідника» 3. Написали код пільги 4. Знайшли 5. Завершили роботу	2 Натисніть 0(увесь довідник у файл) чи 1(на екран)0 Введіть ім'я в файл який хочете записати довідник 34	Passed
5.	1. Обираємо варіант з запропонованих 2. Обрали «Виведення всього довідника» 3. Написали код пільги 4. Знайшли 5. Завершили роботу	Натисніть 0(увесь довідник у файл) чи 1(на екран)1 Code:123 Content:Пільга на освіту Document:Документ: Освіта Number Document: 123456 Date document: 2023-04-01 Date start: 2023-04-01 Date end:2023-04-30 Destination sign: Ознака 1 Coment: Інвалідність Code:456 Content:Пільга на лікарняний Document:Документ: Лікарня Number Document: 789012 Date document: 2023-05-01 Date start: 2023-05-01 Date end:2023-05-31 Destination sign: Ознака 2 Coment: ВПО Code:789 Content:Пільга на проїзд Document:Документ: Проїзд Number Document: 345678 Date document: 2023-06-01 Date start: 2023-06-01 Date end:2023-06-30 Destination sign: Ознака 3 Coment: Літній вік Code:1 Content:На пологи Document:Документ Number Document:45 Date document:12.06.2020 Date start:20.06.2023 Date end:31.12.2023 Destination sign:- Coment:Дякую.	Passed

