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”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ ДИНАМІЧНИХ СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КН-23

Горбенко К.О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

**Лабораторна робота №11**

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

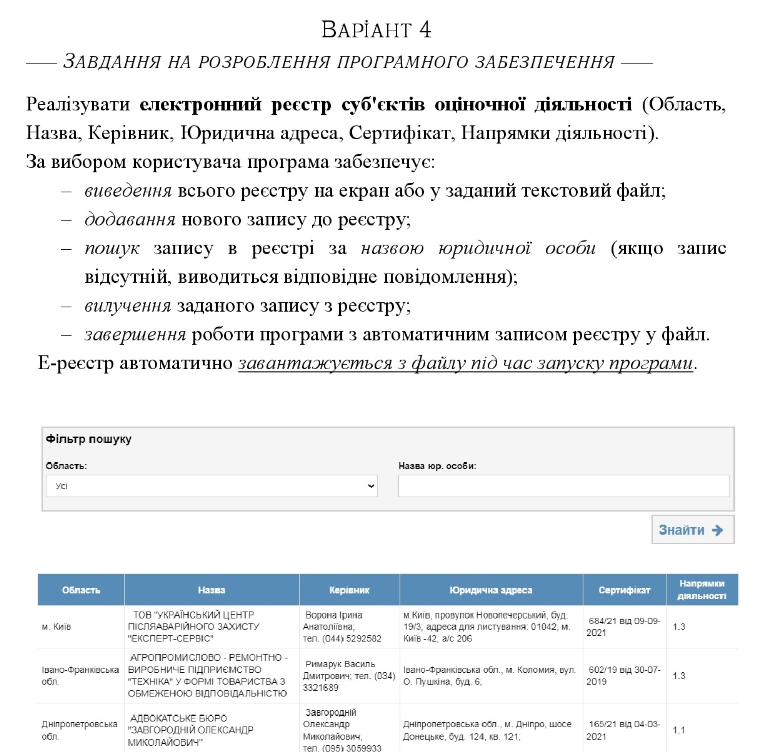
***Мета роботи:*** Набуття ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

**Завдання до лабораторної роботи:**

1. У складі команди ІТ-проекта розробити програмні модулі оброблення динамічної структури даних.

2. Реалізувати програмний засіб на основі розроблених командою ІТ-проекта модулів.

**Варіант 4**



**Склад команди IT-проекта:** Чубенко Ігор, Горбенко Катерина, Кобзар Миколa.

**Результати обговорення**

1. *Аналіз задач лабораторної роботи №11 та вимог до програмного забезпечення:*

* Програма повинна реалізовувати електронний реєстр суб'єктів оціночної діяльності з полями: Область, Назва, Керівник, Юридична адреса, Сертифікат, Напрямки діяльності.
* Програма має забезпечувати функції: виведення реєстру, додавання запису, пошук запису за назвою, вилучення запису, автоматичне збереження і завантаження реєстру з файлу.
* Програма має читатися українською мовою та вміти користуватися нею

1. *Обговорення специфікацій ПЗ, архітектури, алгоритмів та інтерфейсів:*

* Для реалізації е-реєстру доцільно використати векторну динамічну структуру даних.Ця структура визначена в стандартній бібліотеці C++ (<vector>), і вона містить об'єкти типу SubjActivity. Вектор дозволяє динамічно змінювати розмір інформаційного контейнера, що зберігає об'єкти SubjActivity, і автоматично керує пам'яттю для цих об'єктів.
* Основні модулі: структура даних для запису реєстру, функції для операцій над реєстром (додавання, пошук, вилучення, виведення тощо), інтерфейс користувача з меню вибору дій.
* Дані будуть зберігатися у текстовому файлі.

1. *Визначення динамічної структури даних для реалізації бази даних ПЗ:*

* Для реалізації е-реєстру доцільно використати структуру даних список (однозв'язний або двозв'язний).
* Кожен елемент списку буде представляти запис реєстру з полями: Область (рядок), Назва (рядок), Керівник (рядок), Юридична адреса (рядок), Сертифікат (рядок), Напрямки діяльності (масив рядків).

1. *Створення заголовкового файлу з описом елемента структури даних:*

файл struct\_type\_project\_4.h  
#ifndef STRUCT\_TYPE\_PROJECT\_4\_H\_INCLUDED

#define STRUCT\_TYPE\_PROJECT\_4\_H\_INCLUDED

#include <string>

using namespace std;

struct SubjActivity {

string region;

string name;

string manager;

string address;

string certificate;

string activities;

};

#endif // STRUCT\_TYPE\_PROJECT\_4\_H\_INCLUDED

*5. Архітектура ПЗ*

* modulka
* ModulesHorbenko.h (Головний файл  Горбенко)
* ModulesKobzar.h(Головний файл Кобзаря)
* ModulesChubenko.h(Головний файл Чубенко)
* struct\_type\_project\_4.h(Заголовковий файл який містить структуру проекту)
* libModulesHorbenko.a
* libModulesKobzar.a
* libModulesChubenko.a
* ModulesHorbenko (Модулі розроблені Горбенко )
* ModulesKobzar (Модулі розроблені Кобзарем)
* ModulesChubenko (Модулі розроблені Чубенко)
* 4\_Horbenko (Наша программа в якій відбувається виклик модулів)

*6.    Розподіл підзадач з реалізації операцій над структурою даних. Кожен учасник команди реалізує дві функції для роботи зі списком :*

* Додавання нового запису в список (Чубенко Ігор)
* Пошук запису за назвою(Горбенко)
* Вилучення запису зі списку(Чубенко)
* Виведення списку на екран (Кобзар Микола)
* Збереження списку у файл (Горбенко)

*7.    Складання плану робіт з виконання лабораторної :*

* Розробка структури даних, функцій для роботи з нею
* Реалізація інтерфейсу користувача(меню с викликом функцій)
* Тестування і налагодження програми
* Оформлення документації

***Тестування:***

1. *Для searchRecord*

(Позитивні тести)

* Пошук існуючого запису
* Пошук запису з особливими символами

(Негативні тести)

* Пошук неіснуючого запису
* Пошук з порожнім веденням

1. *Для saveRegisrty*

(Позитивні тести)

* Збереження реєстру з кількома записами
* Збереження порожнього реєстру

(Негативні тести)

* Збереження з недоступною назвою файлу
* Збереження реєстру з некоректними даними

***Лістинг ModulesHorbenko.h:***

#ifndef MODULESHORBENKO\_H\_INCLUDED

#define MODULESHORBENKO\_H\_INCLUDED

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

#include "struct\_type\_project\_4.h"

void searchRecord();

void saveRegistry(const std::string& filename);

#endif // MODULESHORBENKO\_H\_INCLUDED

***Лістинг ModulesHorbenko:***

#include "ModulesHorbenko.h"

#include <fstream>

#include <iostream>

std::vector<SubjActivity> registry;

void searchRecord() {

std::string name;

std::cout << "Введіть назву юридичної особи для пошуку: ";

std::getline(std::cin, name);

auto it = std::find\_if(registry.begin(), registry.end(), [&](const SubjActivity& subj) {

return subj.name == name;

});

if (it != registry.end()) {

std::cout << "Область: " << it->region << std::endl;

std::cout << "Керівник: " << it->manager << std::endl;

std::cout << "Юридична адреса: " << it->address << std::endl;

std::cout << "Сертифікат: " << it->certificate << std::endl;

std::cout << "Напрямки діяльності: " << it->activities << std::endl;

}

else {

std::cout << "Запис не знайдено." << std::endl;

}

}

void saveRegistry(const std::string& filename) {

std::ofstream file(filename);

if (!file.is\_open()) {

std::cerr << "Помилка при відкритті файлу для запису: " << filename << std::endl;

return;

}

for (const auto& subj : registry) {

file << subj.region << ","

<< subj.name << ","

<< subj.manager << ","

<< subj.address << ","

<< subj.certificate << ","

<< subj.activities << "\n";

}

file.close();

}

***Лістинг 4\_Horbenko***

#include <iostream>

#include "ModulesHorbenko.h"

#include "ModulesKobzar.h"

#include "ModulesChubenko.h"

using namespace std;

extern std::vector<SubjActivity> registry;

int main()

{

system("chcp 1251");

int choice;

do {

cout << "\nМеню:\n";

cout << "1. Вивести весь реєстр\n";

cout << "2. Додати новий запис\n";

cout << "3. Шукати запис\n";

cout << "4. Видалити запис\n";

cout << "5. Завершити роботу програми\n";

cout << "Виберіть дію (1-5): ";

cin >> choice;

cin.ignore();

switch (choice) {

case 1:

displayRegistry("registry.txt");

break;

case 2:

addRecord();

bre

case 3:

searchRecord();

break;

case 4:

removeRecord();

break;

case 5:

cout << "Збереження реєстру у файл і завершення програми..." << endl;

saveRegistry("registry.txt");

break;

default:

cout << "Невірний вибір. Спробуйте ще раз." << endl;

}

} while (choice != 5);

return 0;

}

***Висновок***

Провівши лабораторну роботу, було досягнуто основну мету – набуття ґрунтовних вмінь і практичних навичок командної реалізації програмного забезпечення. Зокрема, команда успішно розробила програмні модулі для оброблення динамічних структур даних і роботи з бінарними файлами, використовуючи стандартні засоби мови програмування С++.

Процес виконання завдань дозволив поглибити знання з управління динамічною пам'яттю та забезпечив розуміння важливих аспектів командної роботи у сфері програмування. Реалізація програми для електронного реєстру суб'єктів оціночної діяльності продемонструвала практичне застосування векторних динамічних структур даних та списків, а також навички з організації зберігання та обробки даних у бінарних файлах.

Моє завдання полягало в розробленні функцій пошуку запису за назвою та збереження реєстру у файл (registry.txt). Зокрема, функція пошуку запису (searchRecord()) дозволяє знайти юридичну особу за назвою та вивести відповідні дані: область, керівника, юридичну адресу, сертифікат та напрямки діяльності. Функція збереження реєстру (saveRegistry()) забезпечує збереження усіх записів у файл для подальшого відновлення даних.

Реалізація цих функцій включала:

* searchRecord(): пошук запису у векторі за назвою юридичної особи та виведення знайдених даних.
* saveRegistry(): збереження всіх записів з вектора до текстового файлу, зберігаючи дані про область, назву, керівника, юридичну адресу, сертифікат та напрямки діяльності.

Таким чином, виконана робота сприяла розвитку компетенцій у використанні мови С++ для створення ефективних та зручних у використанні програмних рішень. Це дає змогу впевнено сказати, що мета лабораторної роботи досягнута, а отримані знання і навички можуть бути успішно застосовані в подальших професійних проектах.

***Контрольні запитання і завдання***

*1. Що відповідно до міжнародного стандарту ISO/IEC 12207 розуміється під процесом комплексування (інтегрування) ПЗ?*

Процес комплексування (інтегрування) ПЗ згідно з ISO/IEC 12207 включає об'єднання компонентів програмного забезпечення в єдину систему і забезпечення їх сумісної роботи, виконання тестування інтегрованих компонентів та верифікації їх коректної роботи.

*2. Виконайте порівняльний аналіз вказівника та посилання С++.*

* Вказівник: Змінна, яка містить адресу іншої змінної. Вказівник може бути нульовим (nullptr), може змінювати адресу, на яку вказує, і підтримує арифметичні операції.
* Посилання: Альтернативне ім'я для вже існуючої змінної. Посилання не може бути нульовою і після ініціалізації повинна завжди вказувати на ту ж саму змінну.

*3. Перелічіть допустимі операції над вказівниками С++.*

* Присвоєння адреси.
* Дереферування (\*).
* Адресна арифметика (додавання/віднімання цілих чисел, різниця між двома вказівниками).
* Порівняння (==, !=, <, >, <=, >=).
* Інкрементування/декрементування (++/--).

*4. Яке призначення операції опосередкованої адресації та який синтаксис її запису мовою програмування С++?*

Члени (поля) елемента динамічної структури, що підлягають зберіганню, – це ті, що містять корисні дані, необхідні для відновлення стану об'єкта. Наприклад, для структури SubjActivity, яка містить інформацію про суб'єкти оціночної діяльності, такими членами будуть:

* region (рядок) – область.
* name (рядок) – назва юридичної особи.
* manager (рядок) – керівник.
* address (рядок) – юридична адреса.
* certificate (рядок) – сертифікат.
* activities (масив або список рядків) – напрямки діяльності.

Ці поля зберігаються у файл, оскільки вони є суттєвими для відновлення об'єкта при наступному завантаженні програми.

Члени, що не підлягають зберіганню:

Службові поля або вказівники, що використовуються для зв'язків між елементами, не підлягають зберіганню.

Це можуть бути, наприклад:

* Вказівники на наступний або попередній елемент у списку (у випадку однозв'язного або двозв'язного списку).

*5. Які функції-члени об’єктів fstream С++ забезпечують відкриття потоку, запису даних у потік та читання з нього?*

* Відкриття потоку: open().
* Запис даних: << (оператор зсуву вліво) або write().
* Читання даних: >> (оператор зсуву вправо) або read().

*6. Які члени (поля) елемента динамічної структури підлягають зберіганню у файл, а які ні? Відповідь обґрунтуйте.*

Зберігаються члени, що представляють дані (наприклад, поля region, name, manager, address, certificate, activities). Не зберігаються службові поля або вказівники, що використовуються для зв'язків між елементами (наприклад, вказівники на наступний або попередній елемент у списку), оскільки вони мають сенс лише в контексті поточного виконання програми.

*7. Чим бінарний файловий потік відрізняється від текстового з погляду читання/запису і/в нього інформації?*

* Бінарний потік: Дані зберігаються у вигляді сирих байтів, як вони є в пам'яті. Немає перетворення форматів даних (наприклад, число зберігається як є).
* Текстовий потік: Дані перетворюються в текстовий формат (наприклад, числа зберігаються як рядки символів). Може включати спеціальні символи для розділення даних (наприклад, новий рядок).