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”

На тему:

“ Реалізація програмних модулів оброблення динамічних структур даних та бінарних файлів”

ВИКОНАВ

студент академічної групи

КН-24

Ковальова Єва

ПЕРЕВІРИВ

викладач Коваленко А. С.

м. Кропивницький 2025

**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 :**

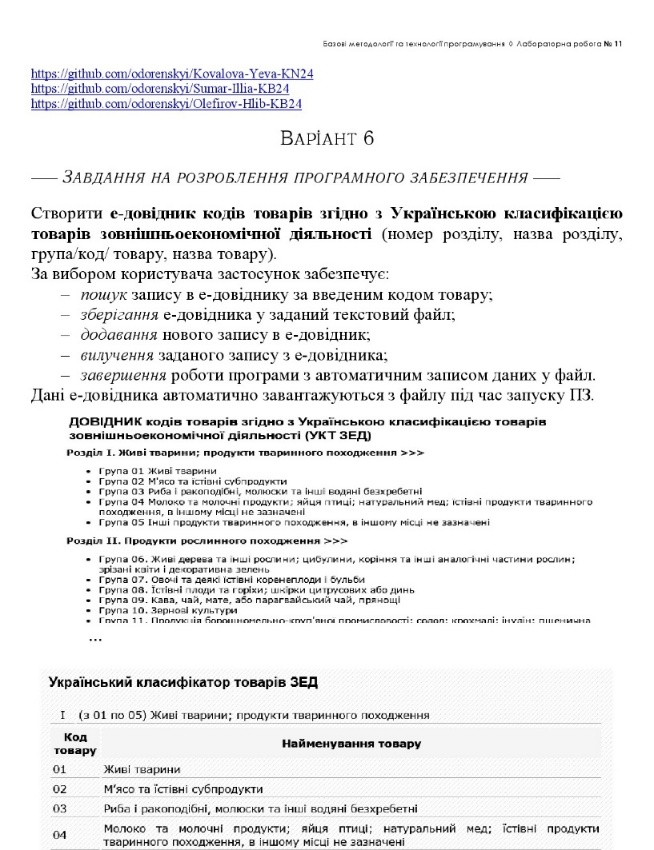
1. У складі команди ІТ-проєкта розробити програмні модулі оброблення динамічної структури даних.
2. Реалізувати програмний засіб на основі розроблених командою ІТ-проєкта модулів

**Склад команди IT-проєкта:**

1. Сумар Ілля
2. Олефіров Гліб
3. Ковальова Єва

**План роботи з виконання IT-проєкта:**

1. Завантажити Git-репозиторій - https://github.com/odorenskyi/student-name;
2. В файл README.md, записати: тему, мету, завдання та варіант.
3. Створити теки Report, prj, Software, TestSuite.
4. В текі \lab11\Report створити Документ Microsoft Word «Звіт до Лабораторної роботи № 11», в ньому написати титульну сторінку, тему, мету, завдання, варіант, склад команди.
5. Розробити детальний аналіз отриманих півзадач та проектування. Для розроблення даних півзадач потрібно використати створений на meeting, заголовковий файл struct\_type\_project\_6.h.
6. Розробити свої дві підзадачі та записати їх в модуль «ModulesПрізвище» підключивши заголовковий файл struct\_type\_project\_2.h, вміст файлу включити до звіту.
7. Завантажити фа**Не знайдено елементи покажчика.**йли на GitHub.
8. Коли всі учасники розроблять свої модулі та проведуть модульні тестування, при успішному тестування, завантажте Git-репозиторії інших учасників.
9. В Code::Blocks IDE створити проект консольного додатка prj\_2\_Прізвище.
10. Відповідно до стандарту ISO/IEC 12207 реалізувати програмний засіб згідно з завданням, використовуючи функції (модулі), розроблені учасниками команди ІТ-проекта.
11. Провести системне тестування консольного додатка prj\_2\_Прізвище та свого модуля. При успішному тестування переходи на наступний крок, якщо результат негативний повернутися до додатка, та виправити помилки.
12. Підготувати й зберегти у \Lab11\Report звіт про виконання лабораторної роботи, оформлений згідно з ДСТУ 3008:2015 “Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання”, зі змістом, визначеним цим порядком виконання лабораторної роботи.
13. Представити до захисту звіт з виконаної лабораторної роботи і проект у Git-репозиторії https://github.com/odorenskyi/student\_name.

**Завдання:**

**Аналіз та постановка задачі:**

**Аналіз виконання Task 11:**

На мітингу, який відбувся в Discord, ми з командою розділили завдання для кожного учасника із команди.

|  |  |
| --- | --- |
| Учасник команди | Функції для реалізації |
| 1. Сумар Ілля | addProduct(), deleteByCode() |
| 2. Ковалова Єва | searchByCode, saveToFile |
| 3. Олефіров Гліб | |  | | --- | |  |  |  | | --- | | loadFromFile, ShowMenu() | |

**Постановка виконання Task 11:**

**Вхідні дані:**

* **Функція для пошуку продукту (searchByCode)**:  
  Дані, що вводить користувач:
  + Код товару для пошуку
* **Функція для збереження в файл (saveToFile)**:
  + Динамічна структура даних (список продуктів), що зберігається у бінарний файл

**Вихідні дані:**

* **Функція пошуку запису:**
  + Якщо продукт з таким кодом знайдено — вивести всю інформацію про нього
  + Якщо продукт не знайдено — вивести відповідне повідомлення
* **Функція збереження у файл:**
  + Повідомлення про успішне збереження даних у бінарний файл

**Висновок:**  
Під час виконання лабораторної роботи № 11 на тему «Реалізація функцій пошуку та збереження даних у бінарний файл» з предмету «Базові Методології та Технології Програмування» я здобувла практичні навички роботи з пошуком у динамічних структурах даних, організацією та збереженням інформації у файлі.

**Вихідний код заголовкового файлу struct\_type\_project\_6:**

#ifndef STRUCT\_TYPE\_PROJECT\_6\_H\_INCLUDED

#define STRUCT\_TYPE\_PROJECT\_6\_H\_INCLUDED

#include <string>

using namespace std;

struct Product {

    string sectionNumber;

    string sectionName;

    string productCode;

    string productName;

};

struct Node {

    Product data;

    Node\* next;

};

#endif // STRUCT\_TYPE\_PROJECT\_6\_H\_INCLUDED

**Вихідний код модуля libModulesKovalova:**

#include "struct\_type\_project\_6.h"

#include <iostream>

using namespace std;

void searchByCode(Node\* head) {

    string code;

    cout << "\n🔍 Введіть код товару: ";

    cin >> code;

    Node\* current = head;

    while (current) {

        if (current->data.productCode == code) {

            cout << "\n✅ Знайдено:\n"

                 << "Номер розділу : " << current->data.sectionNumber << '\n'

                 << "Назва розділу : " << current->data.sectionName << '\n'

                 << "Код товару    : " << current->data.productCode << '\n'

                 << "Назва товару  : " << current->data.productName << '\n';

            return;

        }

        current = current->next;

    }

    cout << "❌ Запис не знайдено.\n";

}

void printMenu() {

    cout << "\n================ МЕНЮ =================\n"

         << "1. 🔍 Пошук запису\n"

         << "2. ➕ Додати запис\n"

         << "3. 🗑️ Видалити запис\n"

         << "4. 💾 Зберегти довідник\n"

         << "5. 🚪 Вихід (з автоматичним збереженням)\n"

         << "======================================\n"

         << "Оберіть дію (1-5): ";

}

**Вихідний код заголовкового файлу modulesKovalova.h:**

#ifndef KOVALOVA\_H\_INCLUDED

#define KOVALOVA\_H\_INCLUDED

#include "struct\_type\_project\_6.h"

void searchByCode(Node\* head);

void printMenu();

#endif // KOVALOVA\_H\_INCLUDED

**Вихідний код prj\_6\_Kovalova**

#include "struct\_type\_project\_6.h"

#include "modulesKovalova.h"

#include "modulesOlefirov.h"

#include "modulesSumar.h"

#include <iostream>

#include <cstdlib>

#include <windows.h>

// Прототипи функцій з модулів

using namespace std;

int main() {

     SetConsoleOutputCP(CP\_UTF8);

    SetConsoleCP(CP\_UTF8);

    setlocale(LC\_ALL, "uk\_UA.UTF-8");

    const string filename = "products.txt";

    Node\* head = loadFromFile(filename);

    int choice;

    do {

        printMenu();

        cin >> choice;

        cin.ignore(); // Очистити буфер після введення числа

        switch (choice) {

            case 1:

                searchByCode(head);

                break;

            case 2:

                addProduct(head);

                break;

            case 3:

                deleteByCode(head);

                break;

            case 4:

                saveToFile(filename, head);

                cout << "💾 Довідник збережено.\n";

                break;

            case 5:

                saveToFile(filename, head);

                cout << "👋 Програма завершується. Дані збережено.\n";

                break;

            default:

                cout << "❌ Невірний вибір. Спробуйте ще раз.\n";

        }

    } while (choice != 5);

    freeList(head);

    return 0;

}

**Аргументи виконання лабораторної роботи:**

1. **Засвоєння принципів пошуку в динамічних структурах.**
2. **Реалізація алгоритму пошуку за ключем (кодом товару).**
3. **Обробка випадків, коли запис не знайдено.**
4. **Навички створення функцій із перевіркою результату пошуку.**
5. **Робота з вказівниками у пов’язаних списках.**
6. **Розуміння принципу лінійного пошуку.**
7. **Практика повернення результатів через функції.**
8. **Використання умовних конструкцій (if, else) для логіки пошуку.**
9. **Створення повідомлень для користувача про успішність операції.**
10. **Формування звички повідомляти про помилки та винятки.**
11. **Застосування інкапсуляції для обробки пошуку всередині класу.**
12. **Реалізація збереження у файл за допомогою потоків ofstream.**
13. **Ознайомлення з форматом бінарного збереження.**
14. **Відпрацювання відкриття та закриття файлів.**
15. **Перевірка на успішне відкриття файлу перед записом.**
16. **Захист програми від збереження порожніх даних.**
17. **Створення функцій серіалізації даних.**
18. **Практика циклічного обходу списку для збереження.**
19. **Вивчення методів запису об’єктів у файл.**
20. **Вивчення різниці між текстовим і бінарним збереженням.**
21. **Застосування модульного принципу для окремих функцій.**
22. **Відокремлення логіки пошуку від логіки збереження.**
23. **Навички форматування даних для читабельного запису.**
24. **Створення гнучкої структури для підтримки масштабування.**
25. **Робота з константними параметрами у функціях.**
26. **Підвищення безпеки обробки даних через захист від помилок.**
27. **Побудова зручного інтерфейсу для взаємодії з користувачем.**
28. **Створення інформативних повідомлень про успішність дій.**
29. **Усвідомлення важливості правильного порядку запису у файл.**
30. **Робота з циклами для перегляду списку.**
31. **Поглиблення знань про структуру вузлів (Node) у списках.**
32. **Практика передачі вказівників у функції.**
33. **Відпрацювання повторного використання коду.**
34. **Дотримання принципу DRY (Don’t Repeat Yourself).**
35. **Підготовка до реалізації сортування та фільтрації даних.**
36. **Навички побудови простого алгоритму перевірки наявності коду.**
37. **Застосування об’єктно-орієнтованого підходу до пошуку.**
38. **Підвищення обізнаності у файловій системі.**
39. **Формування розуміння структури файлів на диску.**
40. **Оптимізація алгоритмів збереження.**
41. **Підготовка до реалізації завантаження даних із файлу.**
42. **Вивчення ролі та призначення буфера в файловому вводі/виводі.**
43. **Побудова навичок перевірки коректності даних перед збереженням.**
44. **Розуміння важливості збереження актуального стану програми.**
45. **Закріплення знань роботи з методами класу.**
46. **Застосування принципів абстрагування.**
47. **Побудова практичного прикладу роботи з базою даних товарів.**
48. **Підготовка до розробки справжніх програм із збереженням стану.**
49. **Створення стабільної та відмовостійкої логіки запису.**
50. **Застосування методів перевірки існування файлу.**
51. **Вивчення небезпеки втрати даних при неправильному збереженні.**
52. **Навички розділення логіки на методи: writeNode, saveToFile, searchByCode.**
53. **Створення гнучкої структури коду для розширення.**
54. **Вивчення варіантів оптимізації обходу списку.**
55. **Формування навички логічного мислення.**
56. **Підвищення дисципліни структурування функціоналу.**
57. **Підготовка до роботи з великими обсягами інформації.**
58. **Навички читання документації до бібліотек потоків вводу/виводу.**
59. **Усвідомлення важливості тестування функцій.**
60. **Практика створення простих юніт-тестів вручну.**
61. **Звичка перевіряти повернені значення функцій.**
62. **Застосування логічних операцій у процесі пошуку.**
63. **Побудова інтуїтивно зрозумілого інтерфейсу пошуку.**
64. **Усвідомлення ролі пошуку в структурах даних.**
65. **Підготовка до реалізації складніших структур (наприклад, дерев).**
66. **Закріплення роботи зі структурами типу Node\* head.**
67. **Практика ручного обходу зв’язного списку.**
68. **Розуміння важливості унікальності ключа (коду товару).**
69. **Вивчення варіантів збереження структур в один файл.**
70. **Використання ітеративного підходу до обробки списків.**
71. **Побудова логіки безпечного запису у файл без втрати даних.**
72. **Розвиток самостійності у вирішенні задач.**
73. **Підвищення рівня впевненості у володінні C++.**
74. **Підготовка до більш просунутих тем (STL, бази даних).**
75. **Створення завершеного фрагменту функціонального застосунку.**