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”

КОМАНДНА 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 КБ-24

Заріцкий В. А.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А. С.

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.**Варіант №2**

**Завдання до лабораторної роботи**

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 IT-команди**

* Заріцкий Віктор (<https://github.com/odorenskyi/Zaritskyi-Viktor-KB24>)
* Кондратенко Дмитро (<https://github.com/odorenskyi/Kondratenko-Dmytro-KB24>)
* Колесник Віктор(<https://github.com/odorenskyi/Kolesnyk-Viktor-KB24>)

# Аналіз задач ІТ‑проєкту та вимог до ПЗ

1. Завантаження бази з файлу при старті програми.
2. Відображення всієї бази на екрані або вивід у текстовий файл (за вибором користувача).
3. Додавання нового запису («особова картка»).
4. Пошук запису за прізвищем працівника.
5. Видалення запису за табельним номером.
6. Збереження бази у файл при завершенні роботи.

# Специфікації ПЗ, концептуальні рішення, архітектура

— Головний файл (main.cpp) — єдиний «оркестратор» з консольним меню.

— Три незалежні модулі (по 2 функції кожний), усі залежать тільки від спільного заголовка struct\_type\_project\_2.h.

Схема взаємодії:

main.cpp → Module\_File

main.cpp → Module\_CRUD

main.cpp → Module\_SearchDel

Module\_File → struct\_type\_project\_2.h

Module\_CRUD → struct\_type\_project\_2.h

Module\_SearchDel → struct\_type\_project\_2.h

Алгоритми викликів із main.cpp:

1. Завантажити: loadDatabase(head, filename)
2. Меню оператора: виклик addRecord, printAllRecords, findRecordByName або deleteRecord
3. На виході: saveDatabase(head, filename)

# Вибір динамічної структури даних

Тип: однозв’язний список.

Обґрунтування:

* Динамічність (будь-яка кількість записів).
* Простота вставки/видалення та проходження лінійно (O(n) — прийнятно для кількох сотень записів).

Вузол списку (Node):

struct Node {

EmployeeRecord record;

Node\* next;

};

Таблиця 1 – Розподіл підзадач (по модулях)

|  |  |  |
| --- | --- | --- |
| **Модуль** | **Студент** | **Функції** |
| Module\_File | Заріцкий Віктор | loadDatabase(), saveDatabase() |
| Module\_CRUD | Кондратенко Дмитро | addRecord(), printAllRecords() |
| Module\_SearchDel | Колесник Віктор | findRecordByName(), deleteRecord() |

Таблиця 2 – План робіт за ISO/IEC 12207

|  |  |
| --- | --- |
| **Етап** | **Відповідальні** |
| 1. Планування | Усі |
| 2. Реалізація модулів | За модулями |
| 3. Інтеграція та тестування | Усі |
| 4. Верифікація відповідності вимогам | Усі |
| 5. Валідація, демонстрація викладачу | Усі + викладач |