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

Гребенюк Д. О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

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

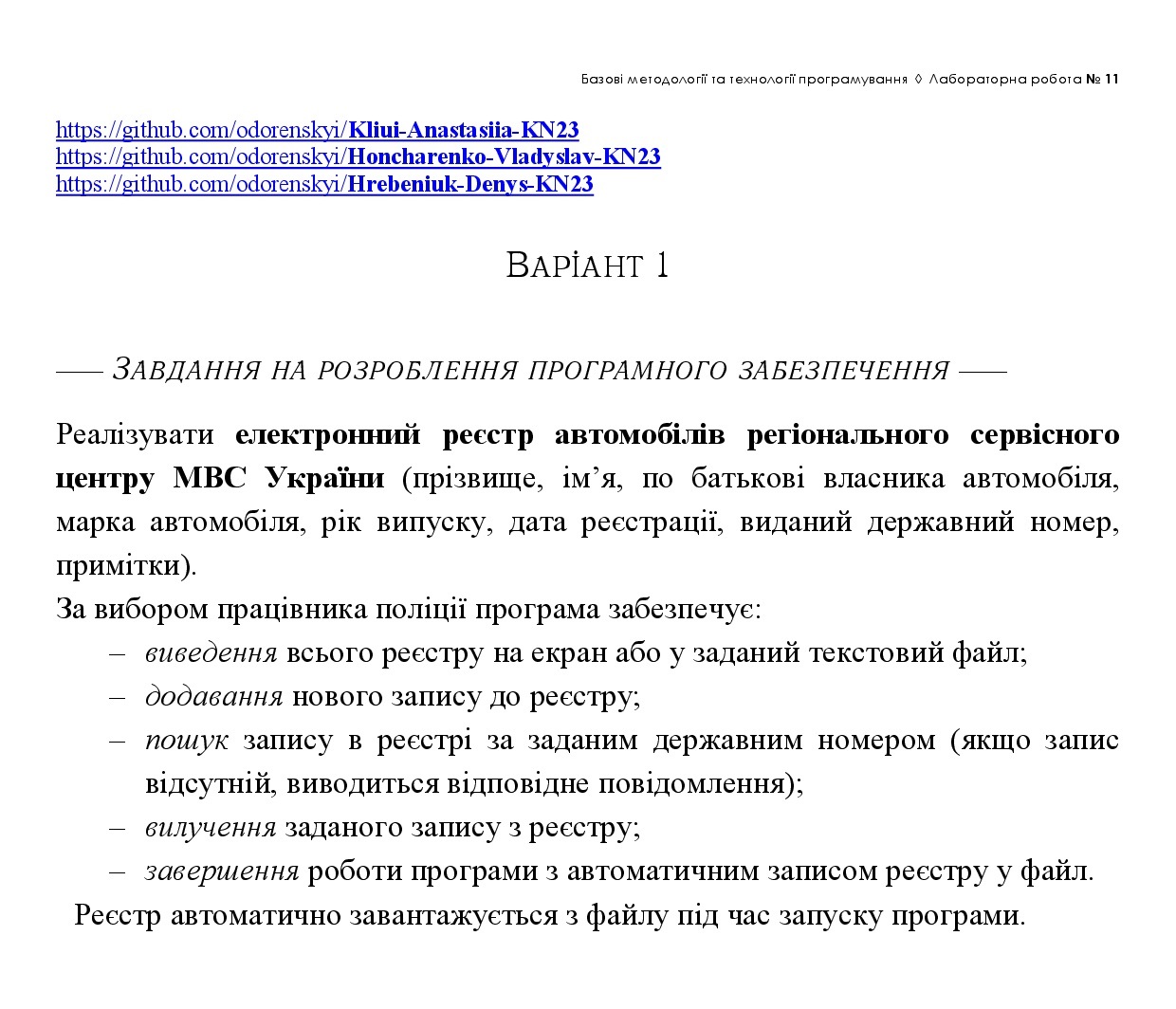
**Мета роботи:** Набуття ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

**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 1**



**Склад команди IT-проекта:** Гребенюк Денис, Клюй Анастасія, Гончаренко Владислав

**Розроблений план виконання IT-проекта**

**«Електронний реєстр автомобілів»**

**1. Аналіз задач ІТ-проекта та вимог до програмного забезпечення**

**● Ціль проекту:**

* розробка програми для автоматизації процесу ведення  електроного реєстру автомобілів у регіональному центрі МВС України.

**● Функціональні вимоги:**

1. ***Виведення*** всього реєстру на екран або у заданий текстовий файл

2. ***Додавання*** нового запису до реєстру автомобілів з такою інформацією: прізвище, ім'я, по батькові власника, марка автомобіля, рік випуску, дата реєстрації, виданий державний номер та примітки;

3. ***Пошук***запису в реєстрі за даним державним номером;

4. ***Вилучення*** заданого запису з реєстру.

5. ***Завершення***роботи програми з автоматичним записом реєстру у файл.

6.  ***Автоматичне завантаження*** з файлу під час запуску програми

**● Нефункціональні вимоги:**

1. програма повинна бути ефективною та швидкою у роботі;

2. надійність та стійкість програми до помилок та відмов.

**● Дані:**

- прізвище, ім'я, по батькові власника автомобіля;

- марка автомобіля, рік випуску, дата реєстрації;

- виданий державний номер;

- примітки.

***Вимоги до програмного забезпечення***

**●Інтерфейс користувача:**

- інтуїтивно зрозумілий та зручний для введення та виведення інформації.

**● Продуктивність:**

- швидке виконання запитів, оптимізація для великих обсягів даних.

**● Сумісність:**

- підтримка різних операційних систем.

**● Масштабованість:**

- легкість оновлення та розширення функціоналу.

**● Технічні обмеження:**

- врахування можливостей використовуваного обладнання та програмного середовища.

**● Документація:**

- повна документація коду

**2. Специфікації ПЗ, концептуальні проектні рішення, архітектура програмного засобу, загальні алгоритми функціонування та інтерфейси модулів**

***Специфікація ПЗ***

**Обговорення щодо:**

1) вимог до програмного забезпечення;

2) вибрання типу динамічної структури даних (список, стек, черга, дерево) для реалізації бази даних ПЗ;

3) створення заголовочного файлу;

4) розподілити підзадачі реалізації операції над динамічною структурою даних;

5) складання плану робіт проекту з урахуванням стандарту ISO/IEC 12207

***Концептуальні проектні рішення***

**1) Модульна архітектура**

• розподіл системи на незалежні модулі для спрощення розробки та тестування:

• модуль виведення реєстру - відповідає за виведення всього реєстру на екран або у заданий текстовий файл.

• модуль додавання нового запису - відповідає за додавання нового запису про автомобіль до реєстру.

• модуль пошуку за державним номером - відповідає за пошук запису в реєстрі за даним державним номером. Якщо запис не буде знайдено, модуль виведе відповідне повідомлення.

• модуль вилучення запису - відповідає за вилучення заданого запису з реєстру.

• модуль автоматичного запису реєстру у файл - відповідає за автоматичний запис у файл під час завершення роботи програми.

**2) Використання списку для реалізації електронного реєстру автомобілів**

• оптимізація швидкості пошуку, вставки та видалення даних

• поліпшення впорядкувння та доступної інформації

***Архітектура програмного засобу***

**1)Модуль управління даними**

**●** Основною структурою даних, є структура **CarRecord**

**●** Операції:

- виведення реєстру: функція для виведення всього реєстру на екран або у заданий текстовий файл. Ця операція дозволяє користувачу переглядати всі дані, що зберігаються у реєстрі.

- додавання нового запису: функція для додавання нового запису про автомобіль до реєстру. Ця операція дозволяє користувачу ввести дані про новий автомобіль та додати його до реєстру.

- пошук за державним номером: функція для пошуку запису в реєстрі за заданим державним номером. Ця операція дозволяє знайти інформацію про автомобіль за його номером.

- вилучення  запису: функція для вилучення заданого запису з реєстру. Ця операція дозволяє користувачу вилучити інформацію про автомобіль з реєстру.

- завершення: функція для завершення роботи програми з автоматичним записом реєстру у файл. Під час завершення роботи програми всі дані про автомобілі автоматично записуються у файл, щоб забезпечити збереження інформації.

- автоматичне завантаження: під час запуску програми, автоматично завантажує реєстр автомобілів з файлу. Ця операція дозволяє користувачу продовжувати роботу з програмою без необхідності вручного завантаження даних.

**2) *Інтерфейс користувача***

• Консольний інтерфейс: простий текстовий інтерфейс, що дозволяє користувачу вводити команди для управліня записами;

• Взаємодія з користувачем: чіткі інструкції та запити для забезпечення правильного введення даних;

***Загальні алгоритми функціонування***

1.**Виведення всього реєстру на екран або у заданий текстовий файл:**

- проходження через всі записи в реєстрі;

- виведення кожного запису на екран або збереження у файл.

**2. Додавання нового запису до реєстру:**

- перевірка наявності даних про автомобіль;

- введення нових даних;

- збереження нового запису в реєстрі.

**3. Пошук запису в реєстрі за даним державним номером:**

- введення державного номера для пошуку;

- пошук в реєстрі за державним номером;

- виведення результату пошуку або відповідного повідомлення, якщо запис не знайдено.

**4. Вилучення заданого запису з реєстру:**

- введення даних для ідентифікації запису, який потрібно вилучити;

- пошук запису в реєстрі;

- вилучення знайденого запису.

**5. Завершення роботи програми з автоматичним записом реєстру у файл:**

- збереження всіх змін у реєстрі у файл.

-завершення виконання програми.

**6. Автоматичне завантаження реєстру з файлу під час запуску програми:**

- перевірка наявності файлу з реєстром;

- якщо файл існує, завантаження даних з файлу у пам'ять програми;

- якщо файл відсутній, створення порожнього реєстру.

**3.  Обгрунтування вид динамічної структури даних для реалізації бази даних ПЗ, складових її елементів (поля структури і їх типи).**

Для реалізації бази даних програмного забезпечення «Електронний реєстр автомобілів» у мові програмування C++ було обрано двонапрямлений (двозв’язний)  список.

**Обґрунтування цього вибору наступне:**

**1) Швидкий доступ:**

Двонапрямлені списки дозволяють швидше переміщатися як вперед, так і назад по списку. Це важливо для операцій, таких як пошук та видалення записів за державним номером, де може знадобитися звернення до попередніх записів.

**2) Ефективність додавання та вилучення:**

Додавання та видалення елементів у двонапрямленому списку може бути ефективними, оскільки не потрібно переміщатися через весь список для зміни посилань.

**3) Простота реалізації:**

Двонапрямлені списки є достатньо простими для реалізації та керуються досить зрозумілими правилами, що полегшує їх розуміння та використання.

**4) Зручність використання:**

Для операцій, таких як виведення всього реєстру на екран або у заданий текстовий файл, двонапрямлений список також може бути зручним, оскільки його можна ітерувати у будь-якому напрямку.

**5) Навчальний аспект:**

Часто використовується у курсах зі структур даних. Відповідно, він ідеально підходить для проектів, спрямованих на вивчення студентами фундаментальних принципів роботи структур даних.

**Складові елементи структури**

**1) Прізвище власника автомобіля:**

- поле структури: lastName

- тип даних: string

**2) Ім'я власника автомобіля:**

- поле структури: firstName

- тип даних: string

**3) По батькові власника автомобіля:**

- поле структури: middleName

- тип даних: string

**4)  Марка автомобіля:**

- поле структури: carBrand

- тип даних: string

**5) Рік випуску автомобіля:**

- поле структури: year

- тип даних: int

**6) Дата реєстрації автомобіля:**

- поле структури: registrationDate

- тип даних: string

**7) Державний номер автомобіля:**

- поле структури: licensePlate

- тип даних: string

**8) Примітки:**

- поле структури: notes

- тип даних: string

**4. Створення заголовочного файлу**

**Задача:** створити заголовочний файл struct\_type\_project\_N.h (N ― номер варіанта завдання) з описом елементів динамічної структури даних мовою С++

**5. Розподіл завдань між учасниками команди підзадачі з реалізації операцій над динамічною структурою даних відповідно до розробленої архітектури програмного засобу.**

**Клюй Анастасія**

1) ***виведення*** всього реєстру на екран або у заданий текстовий файл: реалізація функції для виведення вмісту дерева на екран чи у текстовий файл.

2) ***додавання*** нового запису до реєстру: розробка методу, який дозволяє користувачу вводити новий запис, і додає цей запис до двійкового дерева пошуку. Це включає збір даних від користувача і створення нового вузла в дереві.

**Гончаренко Владислав**

1) ***пошук*** запису в реєстрі за даним державним номером: розробка функції пошуку, яка дозволяє користувачу ввести державний номер і знаходить відповідний запис у дереві, виводячи інформацію про автомобіль або повідомлення про те, що автомобіль не знайдено.

2) ***автоматичне*** завантаження реєстру з файлу під час запуску програми:  розробка механізму для читання даних з файлу і відновлення структури дерева при запуску програми, забезпечуючи коректне відновлення стану реєстру.

**Гребенюк Денис**

1) ***вилучення***заданого запису з реєстру:  реалізація методу видалення запису за введеним користувачем державним номером, що вимагає впевненого видалення вузла з дерева та балансування його після видалення.

2) ***завершення*** роботи програми з автоматичним записом реєстру у файл:  розробка методу, що забезпечує збереження всіх даних у файл при закритті програми та їх відновлення при наступному запуску.