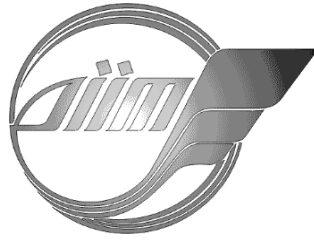


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

**Лабораторна робота №1**

**з дисципліни «Професійна практика програмної інженерії»**

**на тему: “Аналіз вимог та визначення специфікацій програмного забезпечення”**

Виконав:  
студент гр.ПЗ1911  
Сафонов Д.Є.  
Прийняла:  
Шаповал І.В.

Дніпро, 2021

**Тема.** Аналіз вимог та визначення специфікацій програмного забезпечення.

**Мета.** Отримання навичок спільної роботи над проектом, збору вимог та розробки специфікацій різного рівня. Оцінити складність проекту та визначити ризики і терміни розробки.

**Завдання.** 10. Бібліотека з обробки графів. Пошук в ширину, глибину, Ейлерові шляхи.

### **Функціональні вимоги до програмного забезпечення.**

#### **Data Structures(Different representations of Graph):**

- **AdjacencyList<Directed>** implements **Graph**
  - Directed- directed/undirected/bidirectional.
- **AdjacencyMatrix<Directed>** implements **Graph**
  - Directed- directed/undirected/bidirectional.

#### **Algorithms:**

- range **Bfs(Graph g, Graph::Vertex root)** – breadth first search
- range **Dfs(Graph g, Graph::Vertex vertex)** – depth first search
- range **Euler(Graph g)** – eulerian path

#### **Functions:**

- range **AdjacentVertices(Graph g)** – required to implement algorithms

Вказаний у функціях перший аргумент скоріш за все буде передаватися як `this`, а самі функції будуть методами класів.

### **Експлуатаційні вимоги до програмного забезпечення.**

- **правильність** – усі методи мають надавати правильний результат.
- **універсальність** – методи застосовні до усіх графів.
- **надійність (завадостійкість)** – бібліотека повністю безпечна через те, що усі некоректні вхідні дані будуть відловлені ще на стадії компіляції програми, яка буде використовувати бібліотеку.
- **перевіреність** – будуть розроблені тести для усіх алгоритмів та їх складових.
- **точність результатів** – через те, що результати дискретні, точність абсолютна.
- **програмна сумісність** – Програмне забезпечення є бібліотекою розробленою на C++20 із використанням стандартної бібліотеки та компілятора g++10.2, тож бібліотека сумісна з усіма програмами, які розроблюються із використанням цих інструментів, або новіших. Також будуть передбачені прив'язки(bindings) для мови Python.
- **ефективність** – для кожної задачі можливо підібрати клас, який реалізує інтерфейс графу, або навіть розробити свій(бібліотека надає публічний інтерфейс).

## Архітектура програмного забезпечення та розподіл завдань між розробниками.

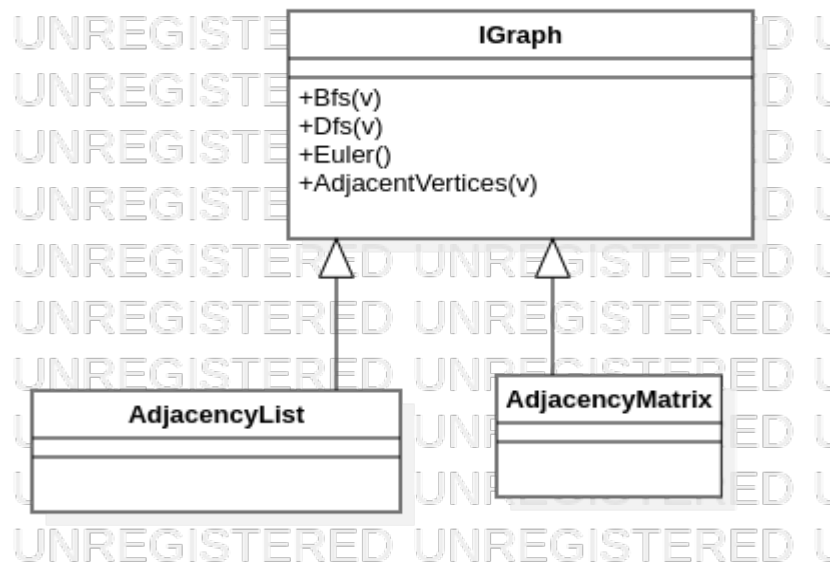


Рисунок 1 Діаграма класів

Обов'язки розподілені навпіл:

Дорогокупля – class AdjacencyMatrix, methods Bfs(v), Dfs(v).

Сафонов – class AdjacencyList, methods Euler(), AdjacentVertices().

## Внутрішні специфікації до внутрішніх складових програмного забезпечення.

- range **Bfs**(**Graph** g, **Graph::Vertex** root) – breadth first search
- range **Dfs**(**Graph** g, **Graph::Vertex** vertex) – depth first search
- range **Euler**(**Graph** g) – eulerian path
- range **AdjacentVertices**(**Graph** g) — повертає вершини, сусідні до даної.
- **AdjacencyMatrix<Directed>** implements **Graph**
  - Представляє граф за допомогою матриці суміжності.
- **AdjacencyList<Directed>** implements **Graph**
  - Представляє граф за допомогою списку суміжності.
- **Graph**
  - Клас-інтерфейс для використання із методами з бібліотеки.

## Висновки.

Були зібрані вимоги, на їх основі була розроблена діаграма класів та прописані повні функціональні та експлуатаційні вимоги. Були розподілені задачі між розробниками.