**Створення й аналіз різновидів статичних діаграм залежностей класів**

**в програмних проєктах**

Постановка задачі:

1. Запропонувати статичні реалізації мовою С++ наступних структур даних:
   1. Орграф, базований на constexpr-подібних засобах мови,
   2. Орграф, базований на засобах метапрограмуванння мовою шаблонів.
2. Екземпляри реалізованих структур даних мають конструюватися з наступними вхіднимии параметрами:
   1. Кількість вершин,
   2. Множина ребер.
3. Вершини графів мають бути представлені беззнаковими цілими числами
4. Запропоновані реалізації мають містити наступні (мета-)функції:
5. adjacent – повертає множину вершин, суміжних з вхідною вершиною,
6. vertex\_count – повертає кількість вершин у графі,
7. edge\_count – повертає кількість ребер у графі,
8. deg – повертає степінь вхідної вершини,
9. has\_edge – перевіряє наявність ребра між двома вхідними вершинами(в сторону другої)
10. Провести порівняльний аналіз отриманих реалізацій, враховуючи наступні критерії:
    1. Час компіляції,
    2. Складність синтаксису,
    3. Розмір файлу з об’єктним кодом,
    4. Обмеження розширення реалізації.
11. Реалізувати наступні алгоритми на отриманих структурах даних:
    1. Пошук в глибину(DFS):
       1. Вхідні дані: орграф, стартова вершина, функтор обробки пошукової ітерації.
       2. Вихідні дані: довільні(auto); визначені вихідним типом функтора обробки пошукової ітерації.
    2. Пошук в ширину(BFS):
       1. Вхідні дані: орграф, стартова вершина, функтор обробки пошукової ітерації.
       2. Вихідні дані: довільні(auto); визначені вихідним типом функтора обробки пошукової ітерації.
    3. Пошук циклів у орграфі:
       1. Вхідні дані: орграф.
       2. Вихідні дані: множину циклів.
12. Функтор обробки пошукової ітерації – довільний тип, що задовольняє наступні умови:
    1. Приймає на вхід вершину поточної ітерації та множину вершин, що акумульована з початку роботи алгоритму до поточної ітерації.
    2. Повертає optional довільного типу чи не повертає нічого.
13. Реалізувати алгоритм виявлення циклів залежностей у довільній ієрархії класів:
    1. Генерація орграфу залежностей на основі заданої ієрархії класів,
    2. Застосувати алгоритм пошуку циклів у орграфі,
    3. Друкування помилок компіляції з інформацією про виявлені цикли.

**ПЛАН**

1. Розроблення описаних у постановці задачі орграфів мовою шаблонів та constexpr-подібними засобами.
2. Тестування розроблених орграфів: написання статичних юніт-тестів.
3. Проведення порівняльного аналізу отриманих реалізацій.
4. Реалізація описаних у постановці задачі алгоритмів на орграфах.
5. Тестування розроблених алгоритмів: написання статичних юніт-тестів.
6. Реалізувати алгоритм виявлення циклів залежностей у довільній ієрархії класів.
7. Упровадження розробленої бібліотеки до проекту Зимовця, Гака, Сердюка.

**Пояснення:**

Будь-яку ієрархію класів можна представити у вигляді графа. А саме, визначимо граф наступним чином:

G = (V, E)

V - множина усіх класів

E = { {u, v} | u, v є V & (u залежить від v або навпаки) }

(між класами А і В є ребро, якщо клас А залежить від класу В або навпаки)

Якщо існує ланцюг, який починається та закінчується якимось класом А, то цей ланцюг є циклом залежностей.