**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«Российский университет дружбы народов имени Патриса Лумумбы»**

Инженерная академия

*Кафедра механики и процессов управления*

**Курсовая работа**

По информатике и программированию

**Направление:** Прикладная математика и информатика

**Профиль:** Математические методы механики космического полёта и анализ геоинформационных данных

**Тема: Библиотека для работы с неориентированными графами на С++**

Выполнено студентом: Свитушкова Наталья

Группа: ИПМбд-02-23

№ студенческого: 1132233503

**Москва, 2025**

Эта библиотека реализует базовую структуру и алгоритмы для **неориентированных графов** на языке C++. Подходит для учебных проектов, визуализации графов и алгоритмической практики.

## Возможности

* ✅ Добавление **вершин**
* ✅ Добавление **рёбер** (неориентированных)
* ✅ Обход графа в **глубину (DFS)**
* ✅ Поиск **пути между двумя вершинами** с помощью DFS

## Как работает DFS (обход в глубину)

Алгоритм **Depth-First Search** (DFS) проходит граф, начиная с заданной вершины, и **уходит «вглубь»**, посещая соседей до тех пор, пока не упрётся в тупик. После этого возвращается назад и исследует другие возможные пути.

В реализации используется **итеративный** подход:

1. Создаётся стек (std::stack)
2. Добавляется стартовая вершина
3. Пока стек не пуст:
   * Извлекаем вершину
   * Если она не посещена — помечаем как посещённую
   * Добавляем всех её соседей в стек (если они ещё не были посещены)

## Как работает findPathDFS

Алгоритм похож на обычный DFS, но с дополнением:

* При посещении вершины запоминаем, **откуда в неё пришли** (parent)
* Когда достигается нужная вершина end, строим путь от неё к start, проходя по parent
* Полученный путь переворачиваем (так как он строится с конца)

1 — 2

| \

3 4 —— 5

\ /

6

Это граф, который мы создаем в main.go

Как и ожидается, путь будет 6 - 4 - 5