

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский политехнический
университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные
системы» направление подготовки: 09.03.01– «Информатика и
вычислительная техника»

Лабораторная работа

**по дисциплине «Основы алгоритмизации и программирования» на
тему «Разработка автоматизированного рабочего места кухни
ресторана. Задача коммивояжера»**

Выполнила студентка группы

ИВТ-23-1б Долганова Диана Евгеньевна

Проверил:

доцент кафедры ИТАС

Яруллин Денис Владимирович

(оценка)

(дата)

(подпись)

г. Пермь, 202

Постановка задачи

Реализовать алгоритмы для собственного, придуманного самим автором, варианта двунаправленного графа, имеющего не менее 6 вершин.

Алгоритмы:

1. Обход в ширину.
2. Обход в глубину.
3. Алгоритм Дейкстры.
4. Выполнить задание своего варианта из методички:
Laby_Chast_3.docx (лаб работа по графам)

Анализ задач

Метод factorial(int i):

1. Рекурсивно вычисляет факториал числа i .
2. Если число равно 0, возвращается 1. В противном случае вызывается рекурсивно функция с аргументом $i - 1$.

Метод Graph::addNode():

1. Добавляет новый узел в граф.
2. Если граф пустой, выделяется память под матрицу и связанные массивы.
3. Иначе создается новая матрица большего размера, данные из старой копируются в нее, а старая матрица удаляется.

Метод Graph::delNode(int num):

1. Удаляет узел из графа по заданному номеру.
2. Создается новая матрица меньшего размера, данные из старой копируются с учетом удаленного узла, а старая матрица освобождается.

Метод `Graph::searchInWidth(int start)`:

1. Выполняет обход графа в ширину, начиная с конкретной вершины. Использует очередь для хранения вершин для обработки.
2. Возвращает строку с порядком посещения вершин.

Метод `Graph::searchInDepth(int start)`:

1. Выполняет обход графа в глубину, начиная с указанной вершины. Использует стек для хранения вершин для обработки.
2. Возвращает строку с порядком посещения вершин.

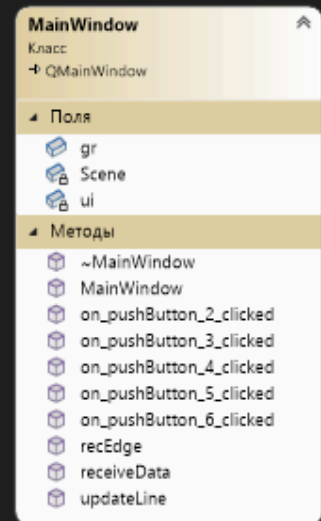
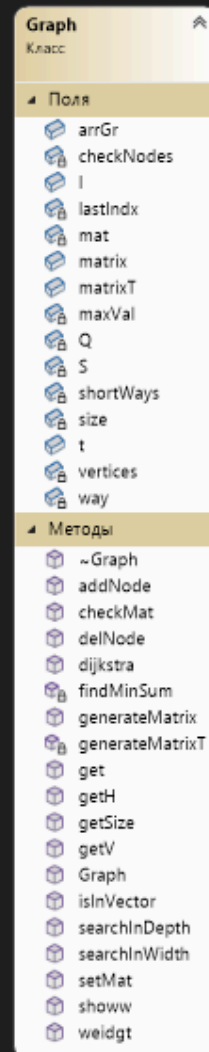
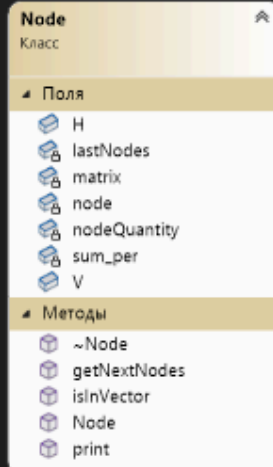
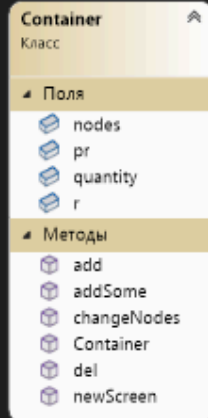
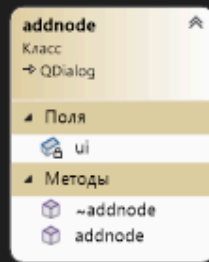
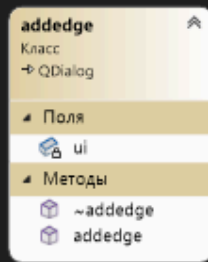
Метод `Graph::dijkstra(int start)`:

1. Выполняет алгоритм Дейкстры для поиска кратчайшего пути в графе от стартовой вершины.
2. Запоминает информацию о кратчайшем пути и его весе.

Метод `Graph::showw()`: Возвращает строку, представляющую кратчайший путь, который был найден алгоритмом Дейкстры.

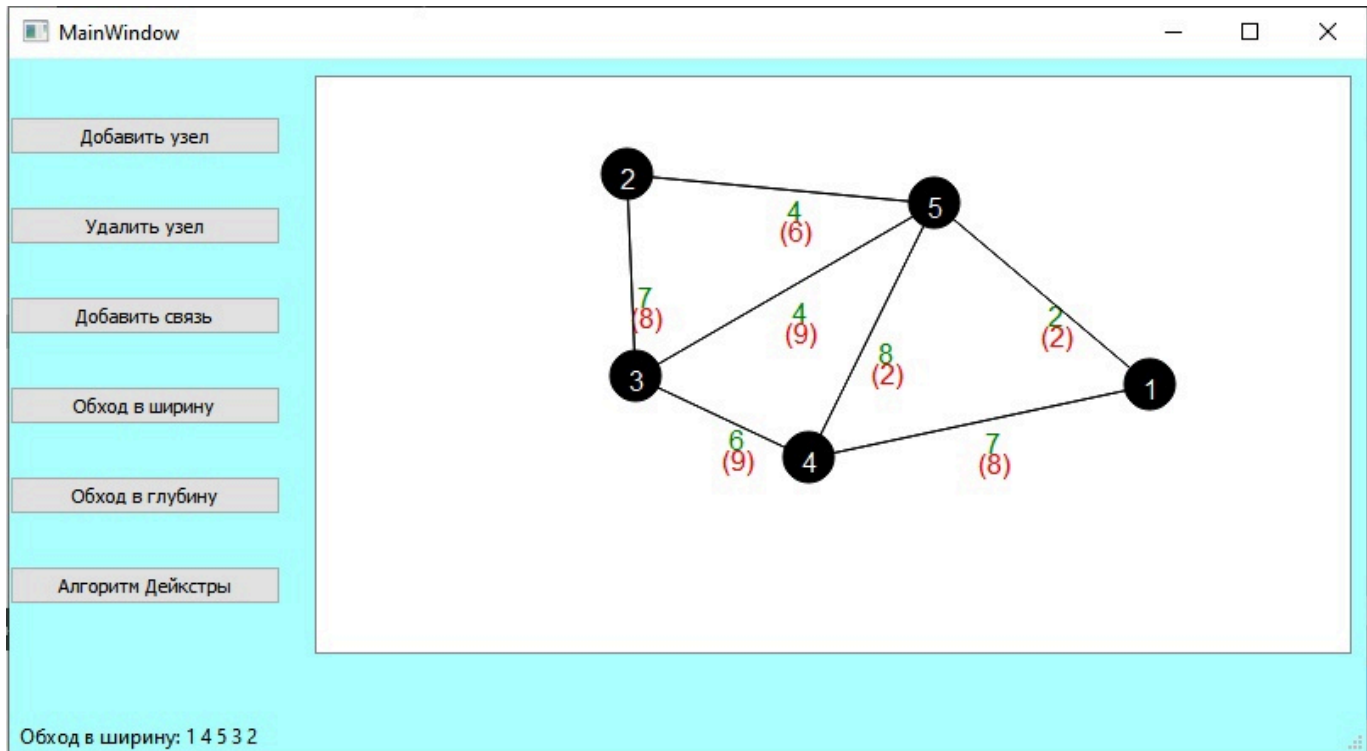
Метод `Graph::setMat(int firstVert, int secondVert, int weight)`: Устанавливает вес ребра между двумя вершинами в графе.

UML - диаграмма

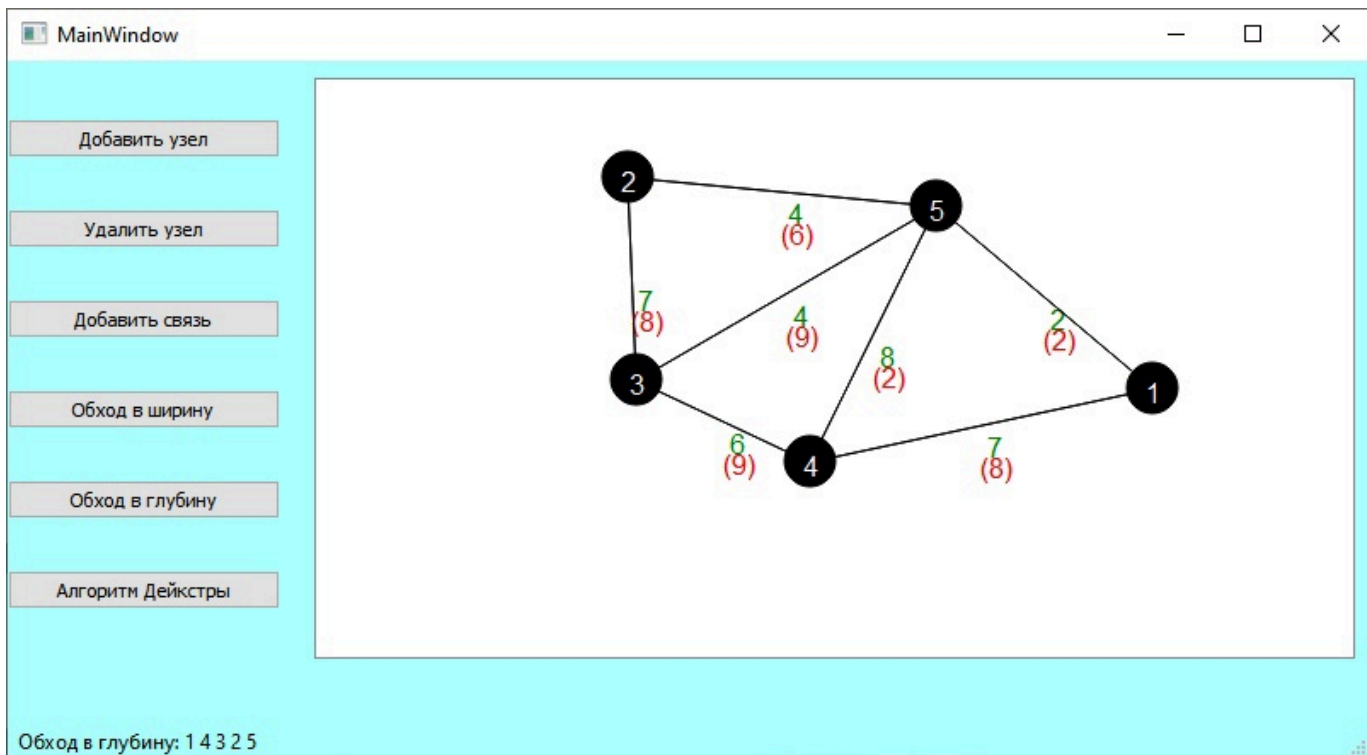


Демонстрация работы

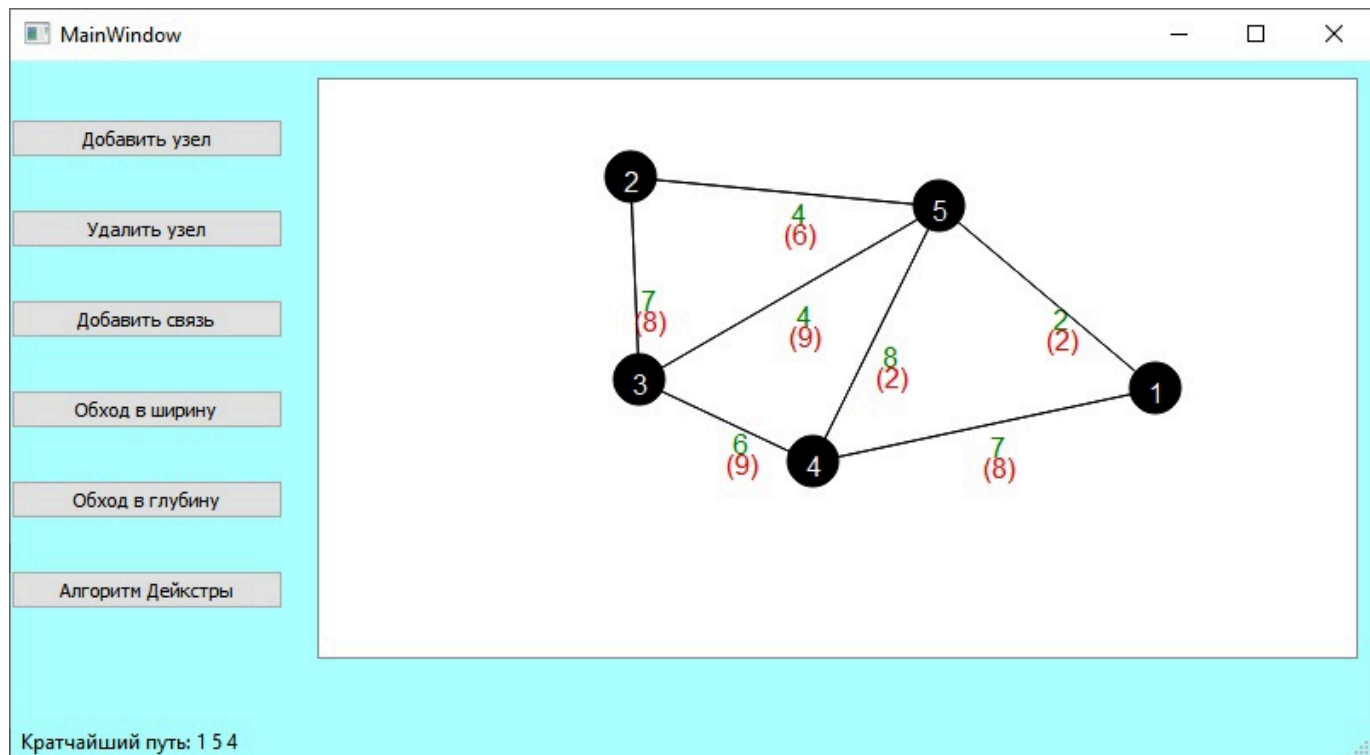
Обход в ширину



Обход в глубину



Алгоритм Дейкстры



Ссылка на GitHub

https://github.com/dolganovadd/LABS_PSTU/tree/main/sem_2.gitkeep/labs.gitkeep/creative%20works.gitkeep