Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Информатика»

Тема: «Графы»

Семестр 2

Выполнил работу

Студент группы ИВТ-22-2б

Мифтахов Марат Ринатович

Проверил

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Г. Пермь-2023

**Постановка задачи**

Реализовать следующие алгоритмы для собственного варианта графа, имеющего не менее 6 вершин: обход в ширину, обход в глубину, алгоритм Дейкстры. Необходимо реализовать функции для редактирования графов: создание новой вершины, удаление вершины, добавление и удаление ребра, редактирование весов рёбер.

**Анализ задачи**

Для реализации графа необходимо реализовать следующие компоненты:

- Обход в глубину (метод depth)

Обход начинается с первой указанной вершины и двигается только в глубь до тех пор, пока для посещения неотмеченных вершин не потребуется вернуться назад к уже пройденным. Алгоритм продолжается до тех пор, пока все вершины не будут посещены.

- Обход в ширину (метод width)

Обход начинается с посещения определённой вершины (для обхода всего графа часто выбирается произвольная вершина). Затем алгоритм посещает соседей этой вершины. За ними - соседей соседей, и так далее.

Алгоритм:

Поддерживается очередь из вершин для посещения. При посещении очередной вершины в очередь добавляются все её соседи, которые ещё не были посещены и ещё не находятся в очереди. Для проверки, была ли вершина уже посещена, используется массив меток. Изначально всем вершинам, не считая первой, присваивается значение false, после прохождения - true. Данный процесс повторяется до тех пор, пока в очереди существуют вершины для посещения.

- Алгоритм Дейкстры (метод Deixstra)

Алгоритм Дейкстры является алгоритмом поиска минимального пути от заданной вершины до всех остальных. Изначально все вершины графа получают некоторую метку, которая характеризует расстояние до исходной

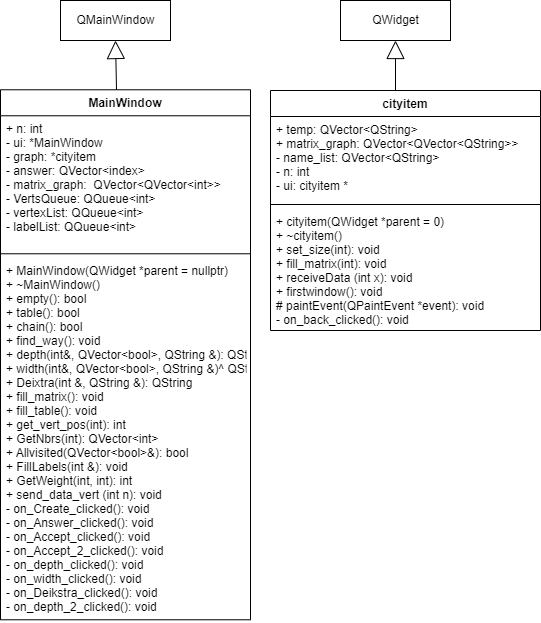
вершины. Это расстояние не известно, поэтому расстояние пока считается как очень большое число.

Вершина, от которой начинается путь получает метку нуль. Далее посещённые вершины также будут помечаться.

Алгоритм:

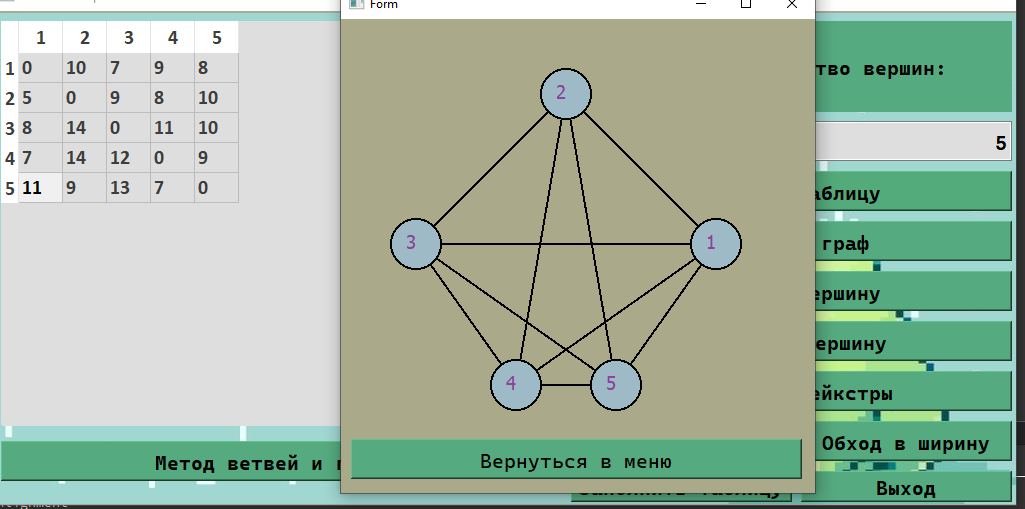
Алгоритм обходит соседей обрабатываемой вершины. Вычисляется расстояние до каждого из соседей как сумма метки обрабатываемой и веса ребра до соседней. Полученное значение сравнивается с меткой соседней вершины и, если полученное значение меньше, то это значение присваивается соседней вершине. После прохода по соседям, обрабатываемая вершина отмечается как обработанная. Ищется новая обрабатываемая вершина, как ближайшая к уже обработанной. При этом, новая обрабатываемая вершина не должна быть обработанной. Обход продолжается до тех пор, пока все вершины не будут обработаны.

**UML - Диаграмма**

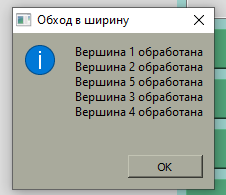
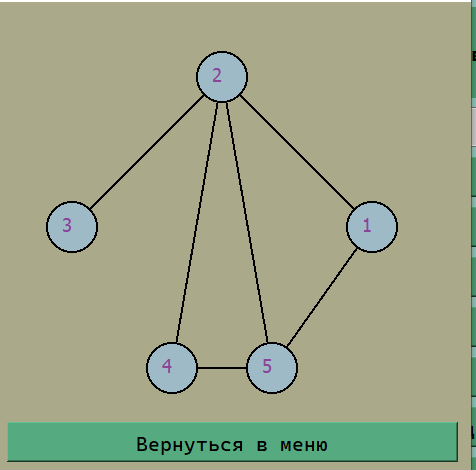


**Рис. 1 – UML-диаграмма классов MainWindow и cityitem**

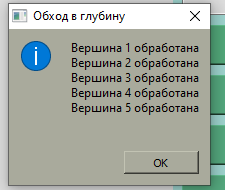
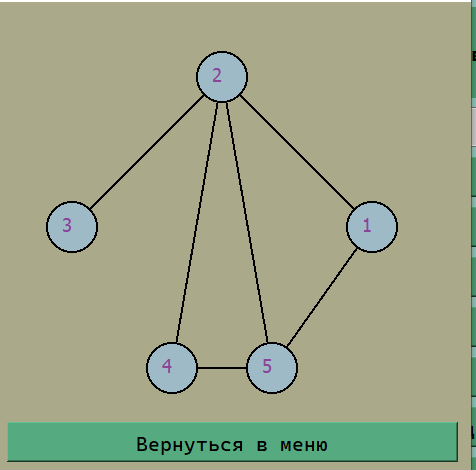
**Скриншоты работы программы**



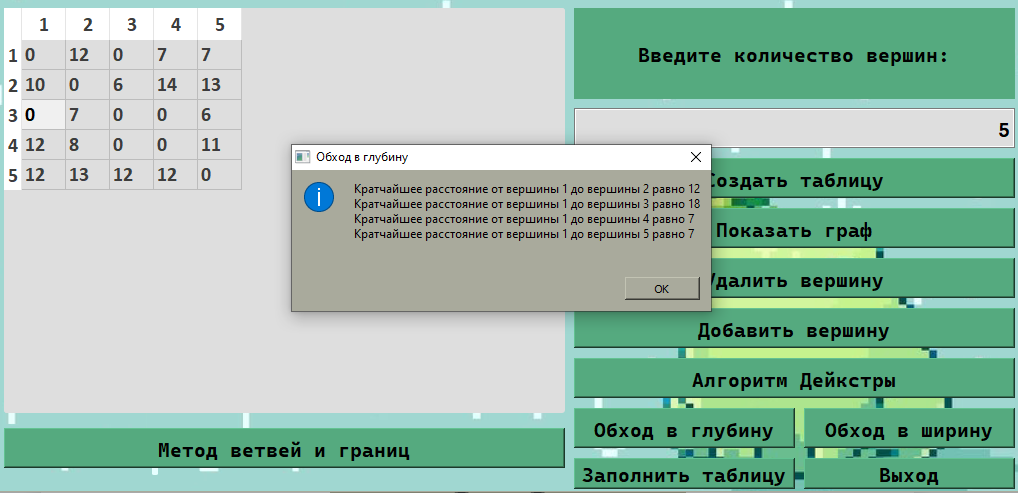
**Рис. 2 – Матрица смежности**



**Рис. 3 – Обход в глубину**



**Рис. 4 – Обход в ширину**



**Рис. 5 – Алгоритм Дейкстры**