**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ Центр ускоренного обучения

ПОИСК КРАТЧАЙШЕГО ПУТИ В ОРИЕНТИРОВАННОМ ГРАФЕ

КУРСОВОЙ ПРОЕКТ

по дисциплине «Программирование»

Пояснительная записка 09.03.03 58.29.21 015 ПЗ

Руководитель ст. пр. Н.А. Архипов

Нормоконтpолер ст. пр. Н.А. Архипов

Студент гр. РИВ-120938у Т.В. Пряминин

**Екатеринбург 2023**

СОДЕРЖАНИЕ

[ВВЕДЕНИE 6](#_Toc136780881)

[1 ПОСТАНОВКА ЗАДАЧИ 7](#_Toc136780882)

[1.1 Формулировка задания 7](#_Toc136780883)

[1.2 Входные данные 7](#_Toc136780884)

[1.3 Выходные данные 7](#_Toc136780885)

[2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ 8](#_Toc136780886)

[2.1 Графы 8](#_Toc136780887)

[2.2 Алгоритм Дейкстры 9](#_Toc136780888)

[2.3 Описание алгоритма 10](#_Toc136780889)

[2.4 Схема алгоритма 11](#_Toc136780890)

[3 СТРУКТУРА ПРОГРАММЫ 12](#_Toc136780891)

[3.1 Диаграмма классов 13](#_Toc136780892)

[3.2 Диаграмма состояний 14](#_Toc136780894)

[4 ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС 15](#_Toc136780896)

[4.1 Снимок интерфейса 15](#_Toc136780897)

[4.2 Описание интерфейса 16](#_Toc136780899)

[ЗАКЛЮЧЕНИЕ 17](#_Toc136780900)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc136780901)

ВВЕДЕНИE

Данный курсовой проект посвящен изучению и реализации алгоритма поиска кратчайшего пути в ориентированном графе. В программе граф представлен в виде матрицы весов. Кратчайший путь представляет собой наименьший набор дуг, соединяющий две вершины в графе.

Актуальность данного проекта обусловлена большим практическим значением данного алгоритма так как задачи поиска кратчайшего пути широко применяются в различных областях, включая транспортную логистику, телекоммуникации, планирование маршрутов и другие.

Для поиска кратчайшего пути в ориентированном графе существует несколько алгоритмов такие как:

1. Алгоритм Беллмана-Форда;
2. Алгоритм Флойда-Уоршелла;
3. Алгоритм Дейкстры.

В рамках проекта реализован один алгоритм – алгоритм Дейкстры. Среди других алгоритмов его отличает меньшая асимптотическая сложность, но при этом он не способен работать с отрицательными весами дуг.

Для программной реализации алгоритма был выбран язык программирования Java. Пользовательский интерфейс создан при помощи бибилиотеки JavaFX.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Формулировка задания

Произвести разработку программы, реализующей алгоритм поиска кратчайшего пути в ориентированном графе – алгоритм Дейкстры. Ввод графа в программу должен осуществляться посредством матрицы весов. Программа должна обладать графическим пользовательским интерфейсом.

1.2 Входные данные

1) Количество вершин орграфа – countVertex;

2) Матрица весов – weightMatrix;

3) Начальная вершина пути – vertexFrom;

4) Конечная вершина пути – vertexTo.

1.3 Выходные данные

1) Вектор вершин, представляющий кратчайший путь от vertexFrom до vertexTo;

2) Общий вес кратчайшего пути;

3) Сообщение «Путь не найден», если путь из vertexFrom в vertexTo не сушествует.

2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

2.1 Графы

Граф является абстрактной структурой данных, используемой для моделирования отношений или связей между объектами. Он состоит из множества вершин (иногда также называемых узлами или точками) и множества ребер (иногда называемых дугами или связями), которые соединяют вершины.

Вершина графа представляет собой одиночный элемент в графе. Каждая вершина может иметь некоторые атрибуты или свойства, которые могут быть использованы для хранения информации о данной вершине.

Ребро графа представляет собой связь или отношение между двумя вершинами. Ребро может быть направленным (ориентированным) или ненаправленным (неориентированным). В случае ориентированного ребра есть определенное направление, указывающее, какая вершина является начальной, а какая конечной. В случае неориентированного ребра, связь между вершинами является двусторонней и не имеет определенного направления.

Ориентированный граф представляет собой граф, в котором каждое ребро имеет направление. Это означает, что ребра имеют определенные начальные и конечные вершины. Ориентированный граф используется для моделирования ситуаций, где существует направленный поток информации, зависимости или направление движения.

Ориентированные графы могут использоваться для моделирования различных ситуаций, таких как транспортные сети, телекоммуникационные сети, диаграммы состояний и другие, где направление имеет значение и важно учитывать поток информации или взаимодействие между вершинами.

2.2 Алгоритм Дейкстры

Алгоритм Дейкстры является одним из наиболее популярных алгоритмов для поиска кратчайшего пути в ориентированном графе. К его отличительным особенностям относят:

1. Предназначен для работы с ориентированными графами, в которых веса ребер являются неотрицательными числами.
2. Работает с одной вершиной графа в качестве начальной точки и находит кратчайшие пути от этой вершины до всех остальных вершин. Это делает алгоритм полезным при решении задач, где требуется найти оптимальный путь от одной конкретной точки до всех остальных точек в графе.
3. Работает пошагово, просматривая вершины графа и обновляя их расстояния от начальной вершины при нахождении более короткого пути. Этот пошаговый подход обеспечивает оптимальность найденных кратчайших путей и делает алгоритм эффективным даже на больших графах.
4. Является детерминированным, что означает, что он всегда будет давать одинаковый результат при одинаковых входных данных. Это делает его предсказуемым и надежным для использования в различных приложениях.

2.3 Описание алгоритма

Алгоритм Дейкстры работает по шагам для нахождения кратчайшего пути от одной начальной вершины до всех остальных вершин в графе. Ниже представлен алгоритм:

1. **Инициализация**: Устанавливаем начальную вершину и устанавливаем ее расстояние от начальной вершины равным 0. Устанавливаем расстояние до всех остальных вершин равным бесконечности.
2. **Выбор ближайшей вершины**: Выбираем вершину с наименьшим расстоянием от начальной вершины. Эта вершина становится текущей вершиной.
3. **Обновление расстояний**: Для каждой смежной вершины текущей вершины вычисляем временное расстояние, суммируя расстояние до текущей вершины и вес ребра, соединяющего текущую вершину с смежной вершиной. Если это временное расстояние меньше текущего расстояния до смежной вершины, обновляем текущее расстояние.
4. **Пометка посещенной вершины:** После обновления расстояний помечаем текущую вершину как посещенную.
5. **Повторение шагов 2-4:** Повторяем шаги 2-4 для всех оставшихся непосещенных вершин, выбирая каждый раз вершину с наименьшим текущим расстоянием.
6. **Завершение**: По завершении алгоритма все расстояния от начальной вершины до остальных вершин становятся оптимальными кратчайшими путями.

Алгоритм Дейкстры продолжает работать, пока все вершины не будут посещены или пока не будут найдены кратчайшие пути до всех вершин. В результате алгоритма мы получаем кратчайшие пути от начальной вершины до всех остальных вершин в графе.

2.4 Схема алгоритма

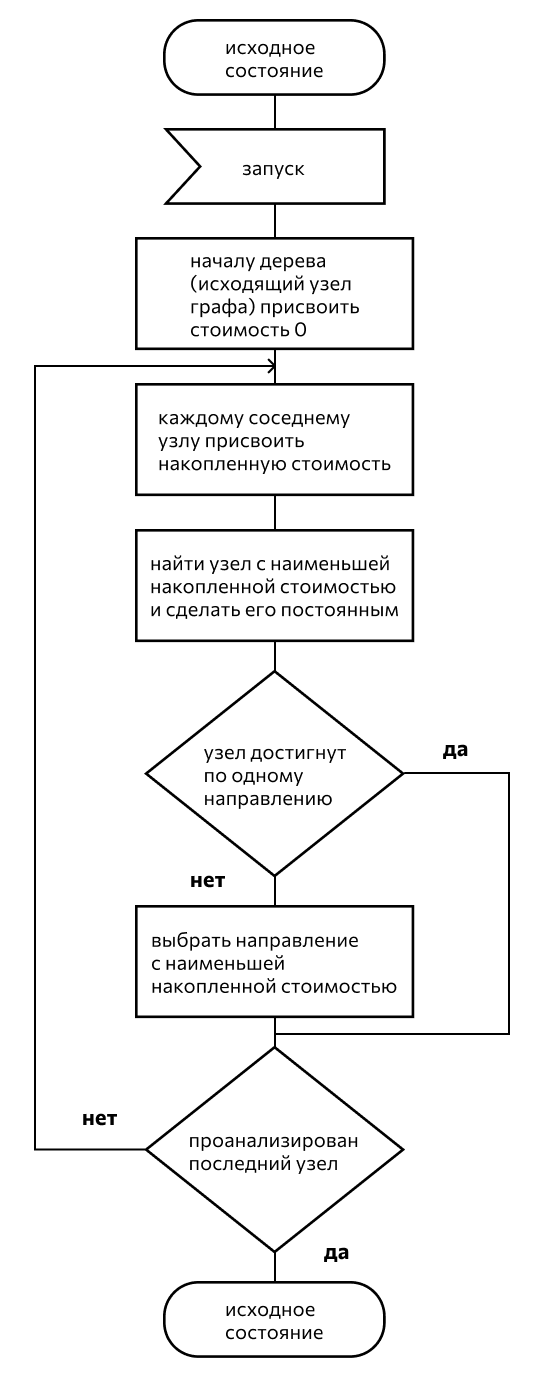


Рисунок 1 -Схема алгоритма

3 СТРУКТУРА ПРОГРАММЫ

Программу можно разделить на следующие логические модули:

Класс **MainApp** – точка входа в приложение, создает форму, загружает FXML;

Класс **MainController –** инкапсулирует JavaFX контроллеры. Определяет процедуры-обработчики событий;

Класс **MessageHandler –** предоставляет средства для осуществления логгирования и отображения сообщений пользователю;

Класс **WeightGraph –** представляет ориентированный граф. Создание графа осуществляется по матрице весов. Формирует список вершин, представляющий кратчайший путь до заданной вершины в соответствии с алгоритмом Дейкстры.

Управление зависимостями осуществляется посредством Maven.

3.1 Диаграмма классов

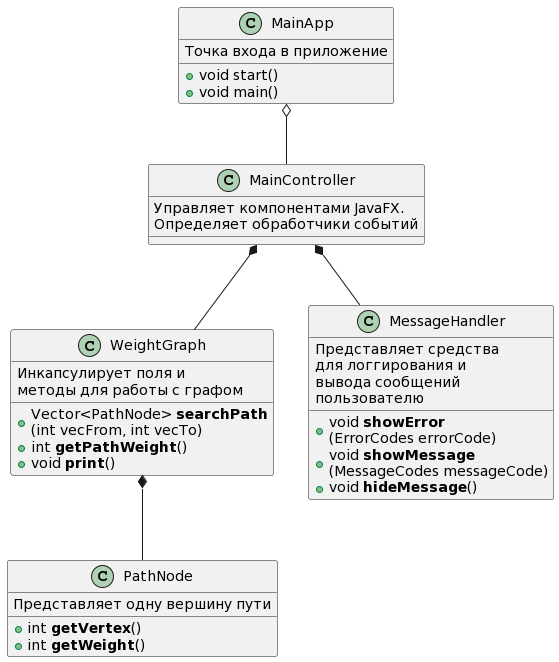


Рисунок 2 – Диаграмма классов

3.2 Диаграмма состояний

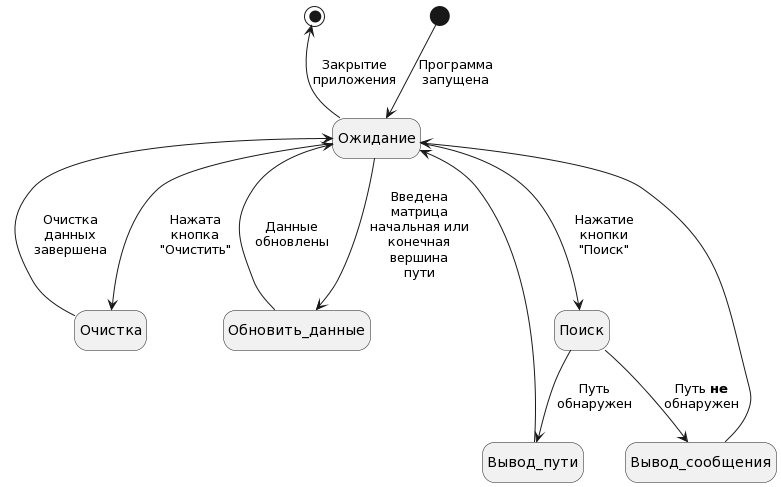


Рисунок 3 – Диаграмма состояний

4 ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

4.1 Снимок интерфейса

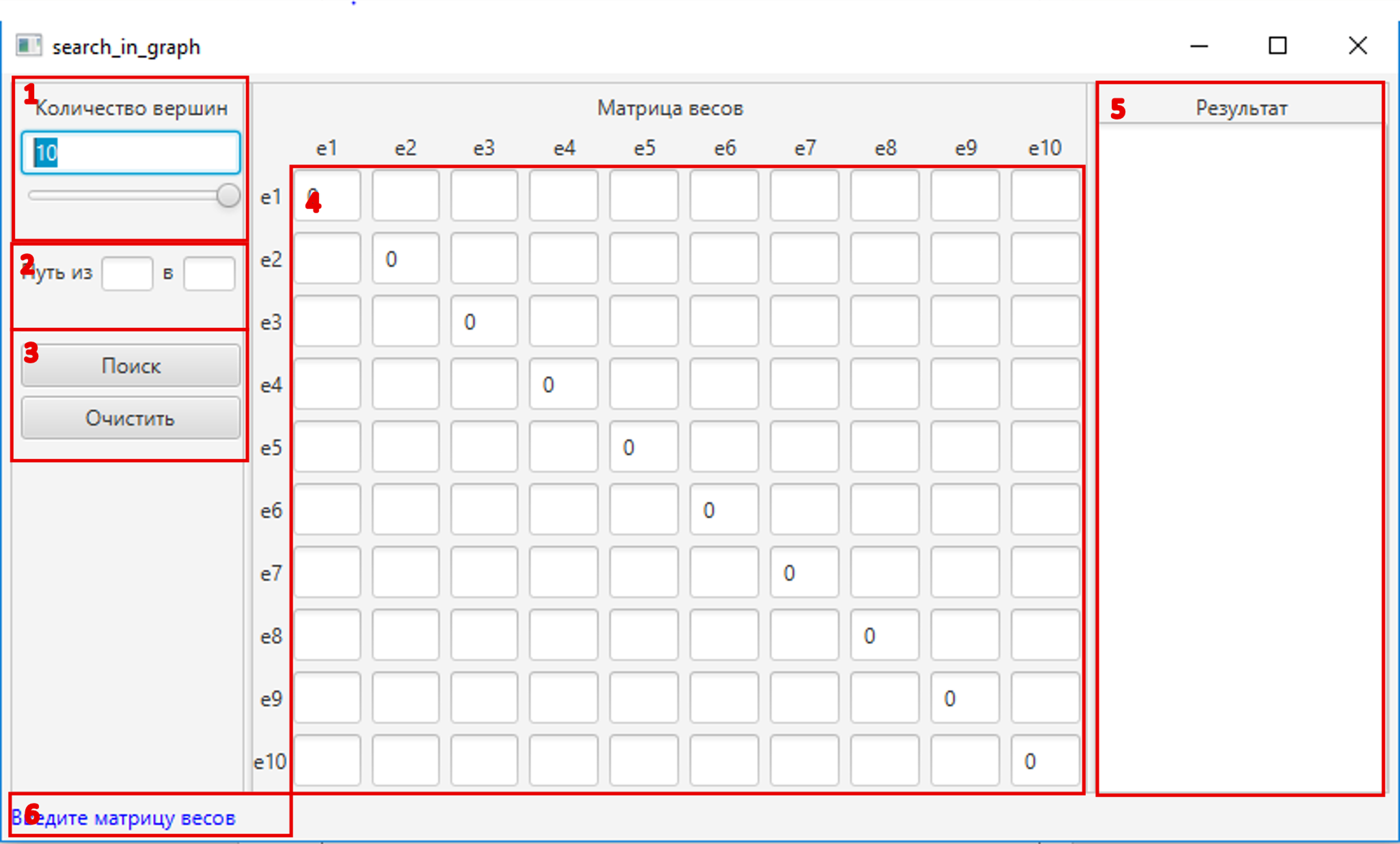


Рисунок 4 – Пользовательский интерфейс

4.2 Описание интерфейса

Ввод количества вершин осуществляется в **Поле 1.**

Ввод начальной и конечной вершины для поиска пути в графе осуществляется в **Поле 2.**

В **Поле 3** расположены кнопки управления программой.

**Поле 4** отвечает за ввод матрицы весов.

В **Поле 5** осуществляется вывод найденного пути.

Сообщения для пользователя, а так же ошибки в работе программы выводятся в **Поле 6**.

ЗАКЛЮЧЕНИЕ

В процессе выполнения данного курсового проекта были получены навыки в области алгоритмического мышления, программирования и анализа графовых структур. Был реализован алгоритм Дейкстры для поиска кратчайшего пути в ориентированных графах.

Алгоритмы поиска кратчайшего пути в ориентированном графе имеют широкое применение в различных областях, таких как транспортная логистика, телекоммуникационные сети, маршрутизация в компьютерных сетях и многое другое. Они играют важную роль в оптимизации путей и принятии решений в реальном времени. Поэтому знание этих алгоритмов и умение их применять актуальны и востребованы в современной индустрии и исследовательских сферах.

Поиск кратчайшего пути - лишь одна из многих задач, связанных с графовыми алгоритмами. Изучение данной темы может стать отправной точкой для более глубокого изучения графовых алгоритмов, таких как алгоритмы топологической сортировки, поиска циклов или нахождения минимального остовного дерева. Это открывает возможности для дальнейшего исследования и применения в реальных проектах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 На каком языке программирования пишут игры // Skypro URL: https://sky.pro/media/na-kakom-yazyke-programmirovaniya-pishut-igry/ (дата обращения: 26.05.2023).

2 Современные языки программирования // Stormnet URL: https://www.it-courses.by/na-kakih-yazykah-programmirovaniya-pishut-igry/ (дата обращения: 26.05.2023).

3 Д. Э. Кнут. "Искусство программирования".

4 А. А. Ахо, Д. Дж. Гусфилд. "Алгоритмы и структуры данных".

5 Мясников, И. Е. Спиричева, Н. Р. Тимошенко, С. И. Государственная итоговая аттестация : учебно-методическое пособие для студентов, обучающихся по направлениям 09.03.01, 09.04.01 «Информатика и вычислительная техника», 09.03.02, 09.04.02 «Информационные системы и технологии», 09.03.03, 09.04.03 « Прикладная информатика», 09.03.04, 09.04.04 «Программная инженерия». - Издательство Уральского университета, 2017. - 108 с.