

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по учебной практике**  
**Тема: Поиск кратчайших путей. Алгоритм Дейкстры**

Студент гр. 2383

\_\_\_\_\_

Сериков М.

Студент гр. 2303

\_\_\_\_\_

Мышкин Н.В.

Руководитель

\_\_\_\_\_

Фирсов М.А.

Санкт-Петербург

2024

## ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Сериков М. группы 2383

Студент Мышкин Н.В. группы 2303

Тема практики: Алгоритм Дейкстры поиска кратчайших путей в графе.

Задание на практику:

Командная итеративная разработка визуализатора алгоритма на Java с графическим интерфейсом.

Алгоритм: Алгоритм Дейкстры поиска кратчайших путей в графе.

Сроки прохождения практики: 26.06.2024 – 09.07.2024

Дата сдачи отчета: 00.07.2024

Дата защиты отчета: 00.07.2024

Студент	_____	Сериков М.
Студент	_____	Мышкин Н.В.
Руководитель	_____	Фирсов М.А.

## **АННОТАЦИЯ**

Цель практики — создание программы с поддержкой графического интерфейса для нахождения кратчайшего пути с помощью алгоритма Дейкстры на графе с неотрицательными весами ребер. Перед выполнением основного задания был составлен план разработки и спецификация программы согласно которым производилась работа.

## **SUMMARY**

The goal of the practice is to create a program with graphical interface support to find the shortest path using Dijkstra's algorithm on a graph with non-negative edge weights. Before completing the main task, a development plan and program specification were drawn up according to which the work was carried out.

## СОДЕРЖАНИЕ

Введение	5
1. Требования к программе	6
1.1. Исходные требования к программе*	6
1.1.1. Требования к вводу исходных данных	6
1.1.2. Требования к визуализации	6
1.1.3. Требования к интерфейсу	6
1.1.4. Требования к построению графа	10
1.1.5. Требования к работе алгоритма	10
1.2. Уточнение требований после сдачи прототипа	0
1.3. Уточнение требований после сдачи 1-ой версии	0
1.4. Уточнение требований после сдачи 2-ой версии	0
2. План разработки и распределение ролей в бригаде	12
2.1. План разработки	12
2.2. Распределение ролей в бригаде	12
3. Особенности реализации	0
3.1. Структуры данных	0
3.2. Основные методы	0
4. Тестирование	9
4.1. Тестирование графического интерфейса	9
4.2. Тестирование кода алгоритма	9
4.3. Тестирование кода графа	9
Заключение	0
Список использованных источников	0
Приложение А. Исходный код	0

## **ВВЕДЕНИЕ**

Главная цель практической работы — реализация графического представления работы алгоритма Дейкстры поиска кратчайших путей в графе. Для достижения поставленной цели необходимо реализовать рассматриваемый алгоритм, пользовательский интерфейс и визуализировать работу алгоритма, после чего произвести тестирование всех компонент проекта.

## **1. ТРЕБОВАНИЯ К ПРОГРАММЕ**

### **1.1. Исходные Требования к программе**

#### **1.1.1. Требования к вводу исходных данных**

На вход программе должен подаваться неотрицательно взвешенный неориентированный граф и исходная вершина. Название вершин состоят из одного символа латинского алфавита. Ввод начальных данных осуществляется двумя возможностями в зависимости от выбора пользователя: непосредственно в рабочей пространстве программы или посредством файла формата txt с данными о графе.

#### **1.1.2. Требования к визуализации**

Алгоритм в ходе работы сохраняет промежуточные решения в виде списка по которому будет пошаговая визуализация с контролем шагов со стороны пользователя. На каждом шагу вывод одного из следующих этапов работы алгоритма:

Проверка соседей текущей вершины: Для каждого соседа проверяется нахождение нового более оптимального пути. (цветом выделяется проверяемая вершина и ребро до соседней вершины).

Обновление данных соседней вершины: Если найден более оптимальный путь, то обновляется метка стоимости пути. (изменение цвета вершины)

Переход к следующей вершине: Из очереди выбирается новая вершина в качестве текущей с наименьшим весом (отрисовка пути с нуля путем перекрашивания соответствующих ребер).

На каждом шаге алгоритма сохраняется текущее состояние алгоритма и очередь вершин.

#### **1.1.3. Требования к интерфейсу**

Эскиз интерфейса представлен на рисунке 1.

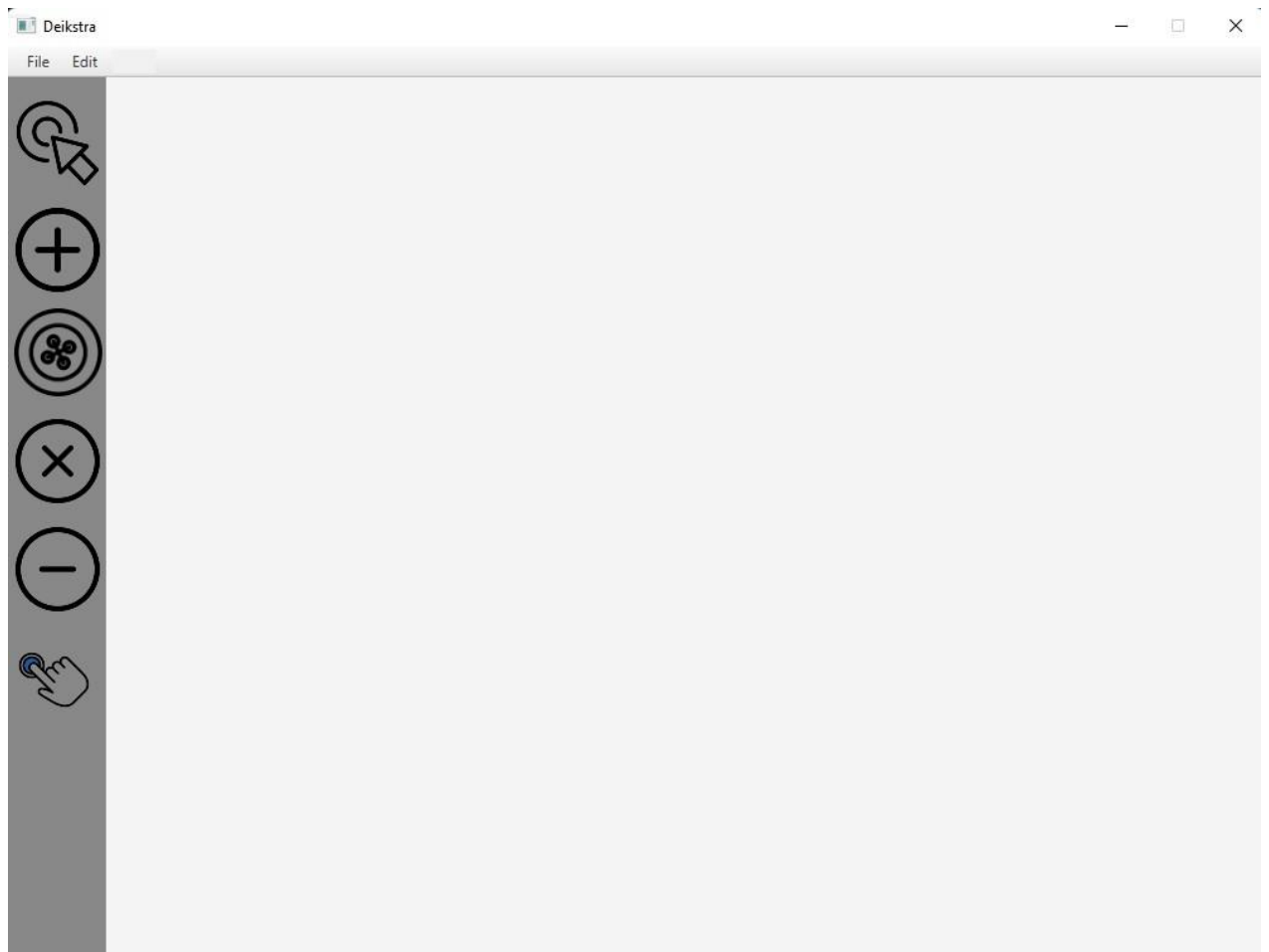


Рисунок 1 — интерфейс программы

В верхней панели программы находится меню с вкладками File, Edit.

Вкладка File: содержит в себе инструменты для загрузки графа из файла, кнопку выхода из программы

Вкладка Edit: содержит в себе инструменты для перехода к окну визуализации

Основная панель инструментов содержит в себе кнопки для построения графа, рассмотрим все кнопки.



Рисунок 2 — кнопка перемещения вершин

Кнопка на рис.2 необходима для перемещения вершин графа, пока активен режим каждую вершину можно перетащить посредством мыши.



Рисунок 3 — кнопка добавления вершин

Кнопка на рис.3 необходима для создания вершин графа, пока активен режим нажатием ЛКМ можно создавать вершины.

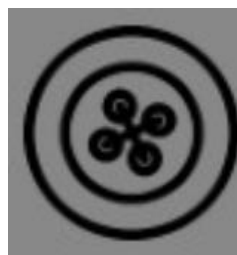


Рисунок 4 — кнопка добавления вершин

Кнопка на рис.4 необходима для добавления ребер графа, пока активен режим выделить две вершины создает ребро.



Рисунок 5 — кнопка удаления вершин

Кнопка на рис.5 необходима для удаления вершин и ребер графа, пока

активен режим выделить

вершины или ребра удаляет его.

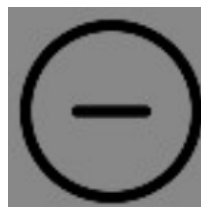


Рисунок 6 — кнопка очистки графа



Кнопка на рис.6 необходима для очистки графа, она очищает рабочую область.

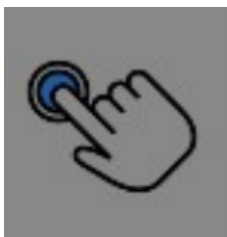


Рисунок 7 — кнопка выбора начальной вершины

Кнопка на рис.7 необходима для выбора начальной вершины для работы алгоритма, она выделяет одну вершину меняя её цвет.

Эскиз окна визуализации работы алгоритма представлен на рисунке 8

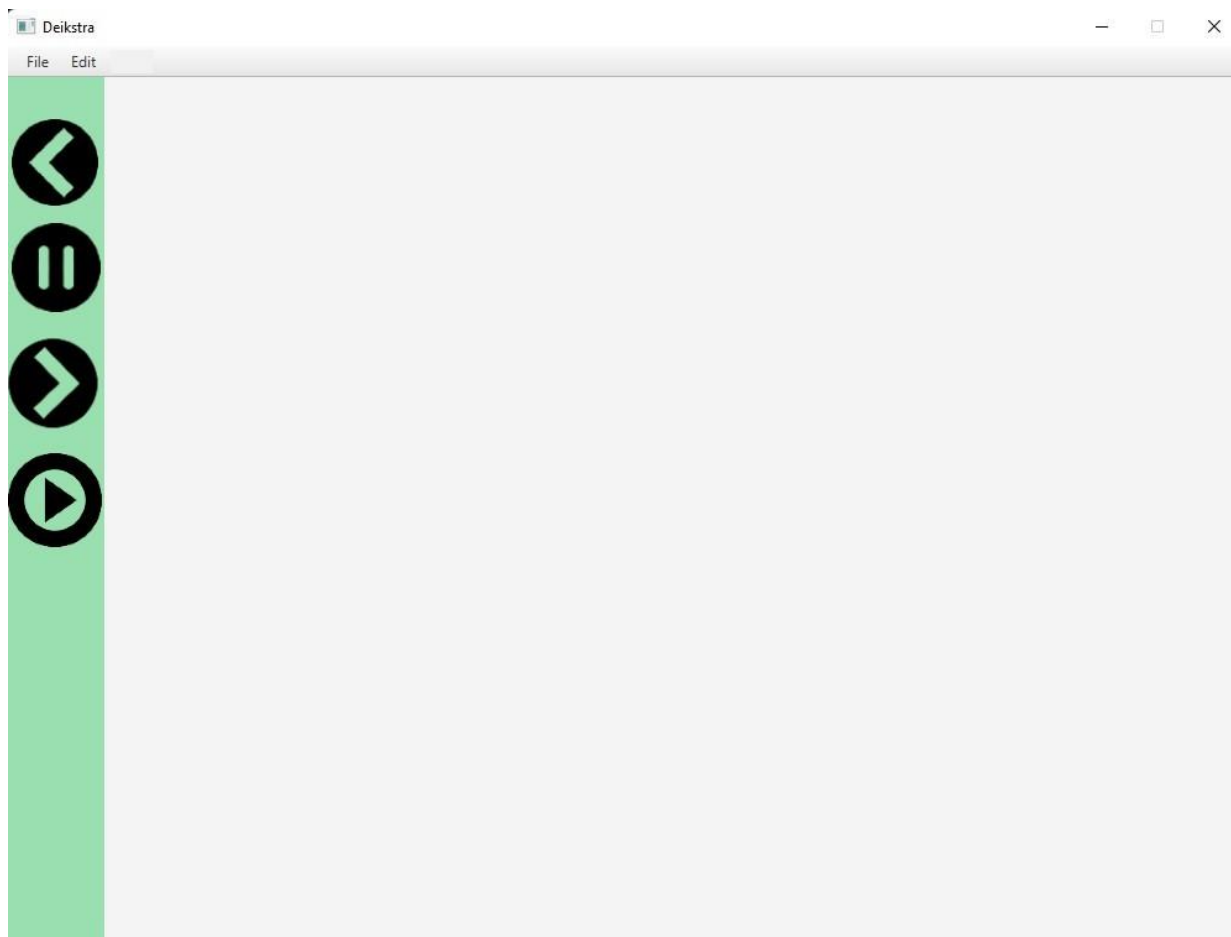


Рисунок 8 — окно визуализации программы

В данном окне представлены инструменты для контроля визуализации работы алгоритма, в случае если выбрана вершина старта алгоритма, то кнопки становятся активными, для начала запуска визуализации алгоритма используется нижняя кнопка, также имеются кнопки паузы и переключения

между следующим и предыдущим шагами визуализации. Кнопка начала визуализации запускает автоматическое воспроизведение шагов с небольшой задержкой, при нажатии паузы можно перемещаться между шагами работы алгоритма.

#### **1.1.4. Требования к построению графа**

Для построения будут использоваться кнопки:

Режим перетаскивания вершин — нажимая ЛКМ и удерживая курсор на вершине возможно её перемещение внутри рабочей области программы (перемещение вершины вслед за курсором)

Режим добавления вершины — нажатие ЛКМ по свободному пространству создаст вершину, которой можно будет присвоить название.

Режим добавления ребра — Необходимо выбрать пару вершин, выделение будет демонстрироваться как изменение цвета вершины, после выделения второй вершины будет построено ребро с возможностью ввода его веса, являющимся неотрицательным числом.

Режим удаления объекта — выделения ЛКМ вершины или ребра удаляет его из графа.

Кнопка очистки рабочей области — удаление всего графа.

#### **1.1.5. Требования к работе алгоритма**

За основу рассматриваемого алгоритма взят следующий псевдокод:

```
function Dijkstra(Graph, source):  
    dist[source] ← 0  
    for each vertex v in Graph:  
        if v ≠ source:  
            dist[v] ← ∞  
            prev[v] ← undefined  
    Q ← the set of all nodes in Graph  
    while Q is not empty:  
        u ← vertex in Q with min dist[u]  
        remove u from Q  
        for each neighbor v of u:  
            alt ← dist[u] + length(u, v)  
            if alt < dist[v]:
```

```
dist[v] ← alt
prev[v] ← u
return dist[], prev[]
```

Алгоритм должен уметь работать с неориентированными графами, с ограничением на не более чем 1 ребро между двумя вершинами и без петель (ребра имеющего начало и конец в одной вершине), он должен реализовывать алгоритм Дейкстры, поиск кратчайших путей с заданной вершины. Алгоритм должен сохранять промежуточные результаты работы для пошаговой визуализации.

## 2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

### 2.1. План разработки

Дата	Этап проекта	Реализованные возможности	Выполнено
27.06.24	Согласование спецификации		
02.06.24	Сдача прототипа	Графический интерфейс программы	
04.07.24	Сдача версии 1	Работа алгоритма без контроля шагов, возможность построения графа, обработка нажатий кнопок рабочей среды.	
06.07.24	Сдача версии 2	Пошаговый контроль работы алгоритма, вывод пояснений, обработка исключений, полное функционирование всех частей программы.	
08.09.24	Сдача отчёта		
08.09.24	Защита отчёта		

### 2.2. Распределение ролей в бригаде

Сериков М. - пользовательский интерфейс программы, визуализация алгоритма, связь графической составляющей программы с основной логикой.

Мышкин Н.В. - реализация алгоритма, структур данных, тестирование

### **3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ**

#### **3.1. Структуры данных**

#### **3.2. Основные методы**

## **4. ТЕСТИРОВАНИЕ**

### **4.1. Первый подраздел четвёртого раздела**

### **4.2. Второй подраздел четвёртого раздела**

## **ЗАКЛЮЧЕНИЕ**

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**



**ПРИЛОЖЕНИЕ А**  
**НАЗВАНИЕ ПРИЛОЖЕНИЯ**