

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

**ОТЧЕТ**  
**по учебной практике**  
**Тема: Новые языки программирования**

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
1.2. 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	8
2.1. План разработки	8
2.2. Распределение ролей в бригаде	8
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 ориентированный 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.

### 1.1.2. Требования к визуализации

Интерфейс программы предоставляет собой диалоговое окно с кнопками для взаимодействия с пользователем. Состав диалогового окна:

1. Рабочая область внутри которой пользователь имеет возможность построить взвешенный граф и внутри которой демонстрируется пошаговая работа алгоритма

2. "Добавить вершину": Добавляет вершину графа в рабочую область по клику мыши. "Удалить": Удаляет выбранный элемент графа (ребро или вершину с инцидентными ребрами) при клике мышью. "Соединить вершины": Создает ребро между вершинами. "Перемещение": Перемещает объекты внутри рабочей области при помощи мыши. "Загрузить": Загружает граф в рабочую область при помощи файла. "Запустить основной алгоритм": Запускает выполнение алгоритма Дейкстры на графе, пользователь выбирает стартовую вершину. "Вперед": Переходит к следующей итерации алгоритма. "Назад": Возвращается к предыдущей итерации алгоритма. "Сброс": Очищает рабочую область.

За основу рассматриваемого алгоритма взят следующий псевдокод:

```
function Dijkstra(Graph, source):  
    dist[source]  $\leftarrow$  0  
    for each vertex v in Graph:  
        if v  $\neq$  source:  
            dist[v]  $\leftarrow$   $\infty$ 
```

```

        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[]

```

## **2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ**

### **2.1. План разработки**

<b>Дата</b>	<b>Этап проекта</b>	<b>Реализованные возможности</b>	<b>Выполнено</b>
<b>27.06.24</b>	Согласование спецификации		
<b>30.06.24</b>	Сдача прототипа		
<b>03.07.24</b>	Сдача версии 1		
<b>05.07.24</b>	Сдача версии 2		
08.09.24	Сдача отчёта		
08.09.24	Защита отчёта		

### **2.2. Распределение ролей в бригаде**

Сериков М. - пользовательский интерфейс программы, алгоритм

Дейкстры, визуализация алгоритма

Мышкин Н.В. - 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**