**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

**ОТЧЕТ**

**по учебной практике**

**“Визуализация алгоритма Флойда - Уоршелла в ориентированном графе на языке Java.”**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6383 |  | Тимофеев Д.А. |
| Студентка гр. 6383 |  | Михеева Е.Е. |
| Руководитель |  | Чайка К.В. |

Санкт-Петербург

2018

**ЗАДАНИЕ**

**на учебную практику**

|  |  |  |
| --- | --- | --- |
| Студент группы 6383 Тимофеев Д.А. | | |
| Студентка группы 6383 Михеева Е.Е. | | |
| Тема практики: визуализация алгоритмов на языке Java | | |
| Задание на практику:  Командная итеративная разработка визуализатора алгоритма на Java с графическим интерфейсом.  Алгоритм: алгоритм Флойда – Уоршелла | | |
| Сроки прохождения практики: 27.06.2018 – 10.07.2018 | | |
| Дата сдачи отчета: 00.06.2018 | | |
| Дата защиты отчета: 00.06.2018 | | |
|  | | |
| Студент гр.6383 |  | Тимофеев Д.А. |
| Студентка гр.6383 |  | Михеева Е.Е. |
| Руководитель |  | Чайка К.В. |

# Аннотация

Темой данной учебной практики является командная итеративная разработка визуализатора алгоритма на языке программирования Java. Цель учебной практики – получить практические навыки в визуализации алгоритмов, изучить и получить навыки использования языка программирования Java, получить навыки работы в команде. В работе представлена визуализация алгоритма Флойда – Уоршелла с пользовательским интерфейсом.

# Summary

## The subject of this training practice is the command iterative development of the algorithm visualizer in Java programming language. The goal of the training practice is to gain practical skills in visualizing Java algorithms and team skills. The work shows the visualization of the Floyd –Warshall algorithm in an oriented graph with the user interface.

## **Содержание**

[ВВЕДЕНИЕ 5](#_Toc486433905)

[1. ТРЕБОВАНИЯ К ПРОГРАММЕ 6](#_Toc486433906)

[1.1 Исходные Требования к программе 6](#_Toc486433907)

[1.2 Изменения к спецификации 7](#_Toc486433908)

[1.2.1 Изменения к спецификации после сдачи прототипа 7](#_Toc486433909)

[1.2.2 Изменения к спецификации после сдачи первой версии 7](#_Toc486433910)

[1.3 Описание входных и выходных данных 8](#_Toc486433911)

[1.4 Примерная работа программы 8](#_Toc486433912)

[2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ 12](#_Toc486433913)

[2.1. План разработки 12](#_Toc486433914)

[2.2. Распределение ролей в бригаде 13](#_Toc486433915)

ВЕДЕНИЕ

Алгоритм Флойда — Уоршелла — алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного [графа](http://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) без циклов с отрицательными весами с использованием метода динамического программирования. В нашей работе будет реализован визуализатор данного алгоритма, с использованием графического интерфейса.

Этот алгоритм был одновременно опубликован в статьях Роберта Флойда ([Robert Floyd](http://en.wikipedia.org/wiki/Robert_Floyd)) и Стивена Уоршелла ([Stephen Warshall](http://en.wikipedia.org/wiki/Stephen_Warshall)) в 1962 г., хотя в 1959 г. Бернард Рой ([Bernard Roy](http://en.wikipedia.org/wiki/Bernard_Roy)) опубликовал практически такой же алгоритм, но это осталось незамеченным.

В нашей работе будет реализован визуализатор данного алгоритма, с использованием графического интерфейса.

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1 Исходные Требования к программе

При запуске проекта будет создаваться отдельное окно. Данное окно будет иметь 3 поля:

1. Поле графического представления графа и визуализации алгоритма. Граф строится по данным, введенными пользователем с помощью графического интерфейса. Для удобства пользователя предусмотрено перемещение вершин графа с помощью мыши.
2. Поле с элементами управления:

1) Следующий шаг

2) Предыдущий шаг

При нажатии кнопки следующий шаг происходят следующие действия: на графе состояние вершины отображается соответствующим цветом (1 цвет – промежуточная вершина, 2 цвет – конечная вершина) и номером порядка посещения и использования. При рассмотрении ребра, соединяющего вершины, оно будет выделено цветом.

1. Диалог создания графа:
2. Ввод количества вершин
3. добавить ребро
4. старт

1.2 Примерная работа программы

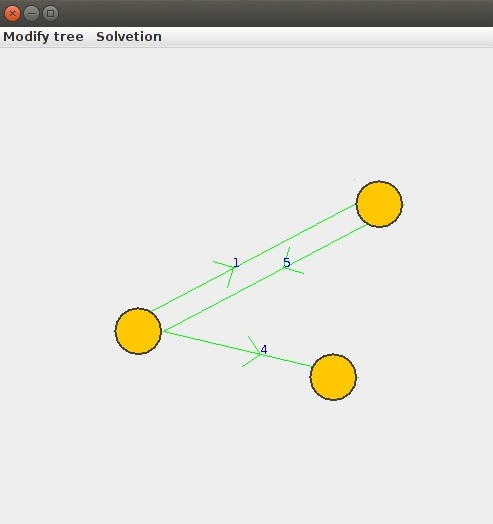


Рисунок 1. Интерфейс поля визуализации алгоритма.

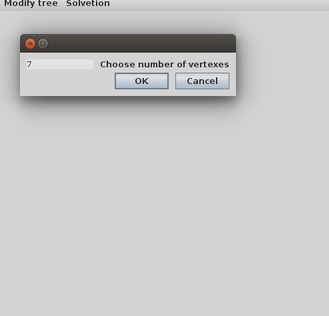


Рисунок 2. Диалог задания количества вершин в графе.



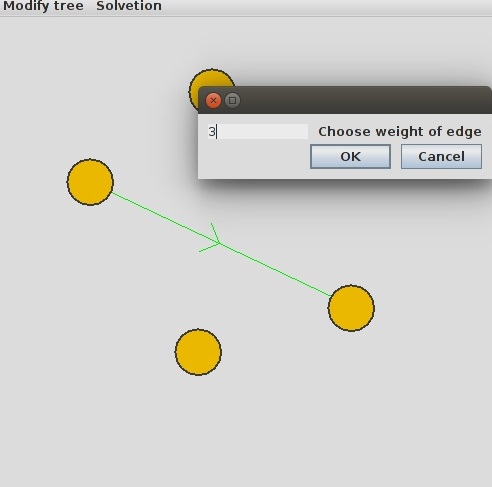
Рисунок 3. Задание ребер графа

Рисунок 4. Диалог задания веса ребра.

1.3 Описание входных и выходных данных

Входные данные:

Граф, введенный пользователем с помощью графического интерфейса.

Выходные данные:

Графическое представление графа по шагам с подсветкой всех посещенных вершин и соответствующих рассматриваемых ребер. Вывод кратчайшего пути между заданной парой вершин.

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки

29.06.2018 (пятница) – разработка спецификации

30.06.2018 (суббота) – согласование спецификации с руководителем

(31.06 -03.07). 2018 – разработка интерфейса

04.07.2018 (среда) - предоставление планируемого интерфейса (прототип), с отсутствующей функциональностью.

05.07.2018 (четверг)– реализация графического представления графа, на основе входных данных и осуществление работы алгоритма с выводом результата в соответствующее окно (частичная функциональность)

06.07.2018 (среда) - сдача 1-ой версии с добавлением частичной функциональности к пользовательскому интерфейсу;

07.07.2018 – реализация пошаговой работы алгоритма, тестирование программы. Оформление отчета.

10 июля (пятница) - сдача финальной версии. Представление проекта с полной функциональностью.

2.2. Распределение ролей в бригаде

В таблице 1 представлен состав бригады и распределение обязанностей.

Таблица 1.

|  |  |
| --- | --- |
| Имя студента | Обязанность |
| Тимофеев Дмитрий | Реализация отдельных графических компонентов (интерактивные вершины, ребра графа). Реализация алгоритма Флойда – Уоршелла с учетом специфики пошаговой визуализации. |
| Михеева Екатерина | Реализация диалога ввода графа пользователем. Пошаговая визуализация алгоритма. Тестирование. |