

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

**ОТЧЕТ**  
**по учебной практике**  
**ТЕМА: ВИЗУАЛИЗАЦИЯ АЛГОРИТМА ФЛОЙДА-УОРШЕЛЛА**

Студент гр. 9304	_____	Ковалёв П.Д.
Студент гр. 9304	_____	Борисовский В.Ю.
Студент гр. 9304	_____	Прокофьев М.Д.
Руководитель	_____	Фиалковский М.С.

Санкт-Петербург

2021

## ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Ковалёв П.Д. группы 9304

Студент Борисовский В.Ю. группы 9304

Студент Прокофьев М. Д. группы 9304

Тема практики: визуализация алгоритма Флойда-Уоршелла

Задание на практику:

Командная итеративная разработка визуализатора алгоритма(ов) на Java с графическим интерфейсом.

Алгоритм: алгоритм Флойда-Уоршелла.

Сроки прохождения практики: 01.07.2021 – 14.07.2021

Дата сдачи отчета: 00.07.2021

Дата защиты отчета: 00.07.2021

Студент

\_\_\_\_\_

Ковалёв П.Д.

Студент

\_\_\_\_\_

Борисовский В.Ю.

Студент

\_\_\_\_\_

Прокофьев М.Д.

Руководитель

\_\_\_\_\_

Фиалковский М.С.

## **АННОТАЦИЯ**

Цель практики – научиться работать в команде и улучшить умение писать код в объектно-ориентированном стиле на языке программирования Java. Изучить основы языка программирования Java, а также средства разработки приложений с графическим интерфейсом на данном языке. В рамках практики выполняется мини-проект в команде, суть которого – реализация визуализатора графового алгоритма средствами языка Java. В процессе работы предстоит разработать прототип интерфейса, реализовать сам алгоритм, а также при помощи средств тестирования отладить разработанную программу. Нашей командой был выбран алгоритм Флойда-Уоршелла.

## **SUMMARY**

The goal of the practice is to learn how to work in a team and improve the ability to write code in an object-oriented style in the Java programming language. Learn the basics of the Java programming language, as well as tools for developing applications with a graphical interface in this language. As part of the practice, a mini-project is carried out in a team, the essence of which is the implementation of a graph algorithm visualizer using the Java language. In the process of work, it is necessary to develop a prototype of the interface, implement the algorithm itself, and also use testing tools to debug the developed program. Our team chose the Floyd-Warshall algorithm.

## СОДЕРЖАНИЕ

	Введение	5
1.	Требования к программе	6
1.1.	Исходные требования к программе*	6
1.1.1	Требования к визуализации	7
1.1.2	Требования к входным данным	9
1.1.3	Требования к архитектуре	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	0
2.1.	План разработки	0
2.2.	Распределение ролей в бригаде	0
3.	Особенности реализации	0
3.1.	Структуры данных	0
3.2.	Основные методы	0
3.3		0
4.	Тестирование	0
4.1	Тестирование графического интерфейса	0
4.2	Тестирование итогового проекта	0
4.3	...	0
	Заключение	0
	Список использованных источников	0
	Приложение А. Исходный код – только в электронном виде	0

## **ВВЕДЕНИЕ**

Основная цель практики – реализовать визуализатор алгоритма Флойда-Уоршелла. Алгоритм предназначен для нахождения кратчайших путей между вершинами во взвешенном графе. Для реализации проекта, необходимо реализовать графический интерфейс, сам алгоритм и объединить данные наработки.

## **1. ТРЕБОВАНИЯ К ПРОГРАММЕ**

### **1.1. Исходные требования к программе**

Приложение будет иметь графический интерфейс, через который будет возможно вводить данные (вершины и рёбра графа), так же будет реализована поддержка ввода данных через файл. Программа будет иметь несколько режимов работы: многошаговый режим, суть которого – пошаговая демонстрация работы алгоритма и одношаговый – программа вычислит кратчайшие пути в переданном ей графе и перечислит стоимости переходов от одних вершин к другим в области “Матрица смежности” в окне программы.

### 1.1.1 Требования к визуализации

Окно программы будет выглядеть следующим образом:

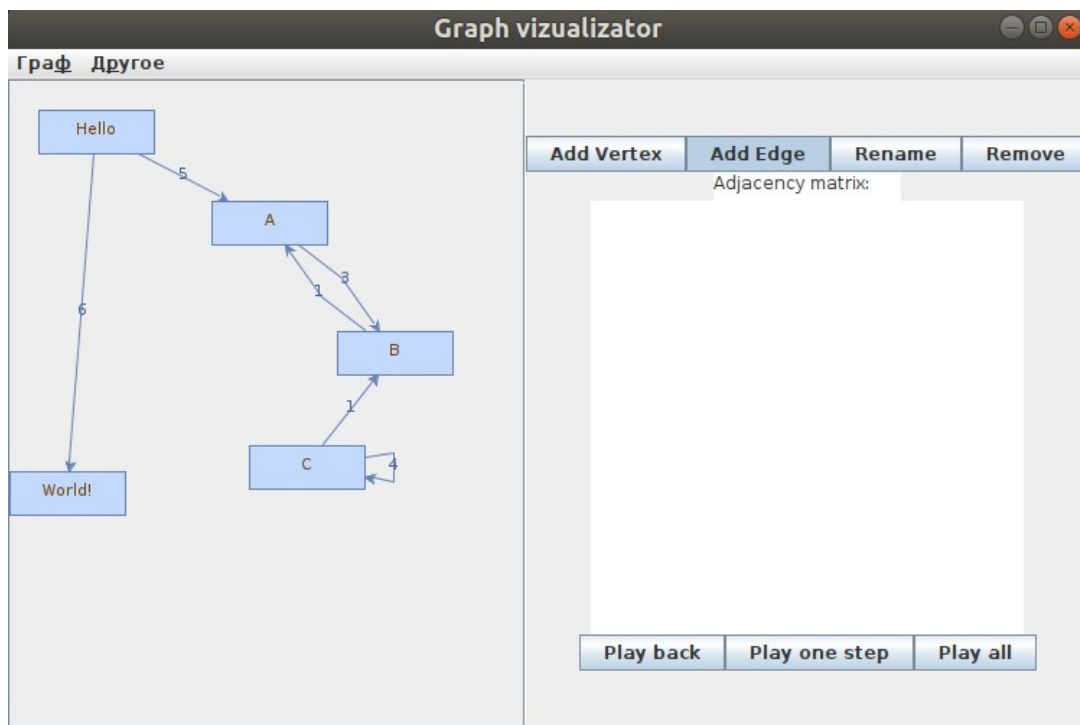


Рисунок 1 – Окно программы

Из рисунка 1 видно, что программа будет реализовывать следующий функционал:

Добавление/удаление вершин и рёбер. После нажатия соответствующих кнопок, клик левой кнопки мыши будет создавать или удалять вершину графа, а также добавлять ребро между вершинами. При добавлении ребра будет всплывать окошко, в котором нужно будет ввести вес ребра. Это видно на рисунке 2.

Переименование. Данная кнопка позволяет переименовать вершину, нажатием на нее. При нажатии появляется окошко, в котором пользователю будет предложено ввести новое название вершины.

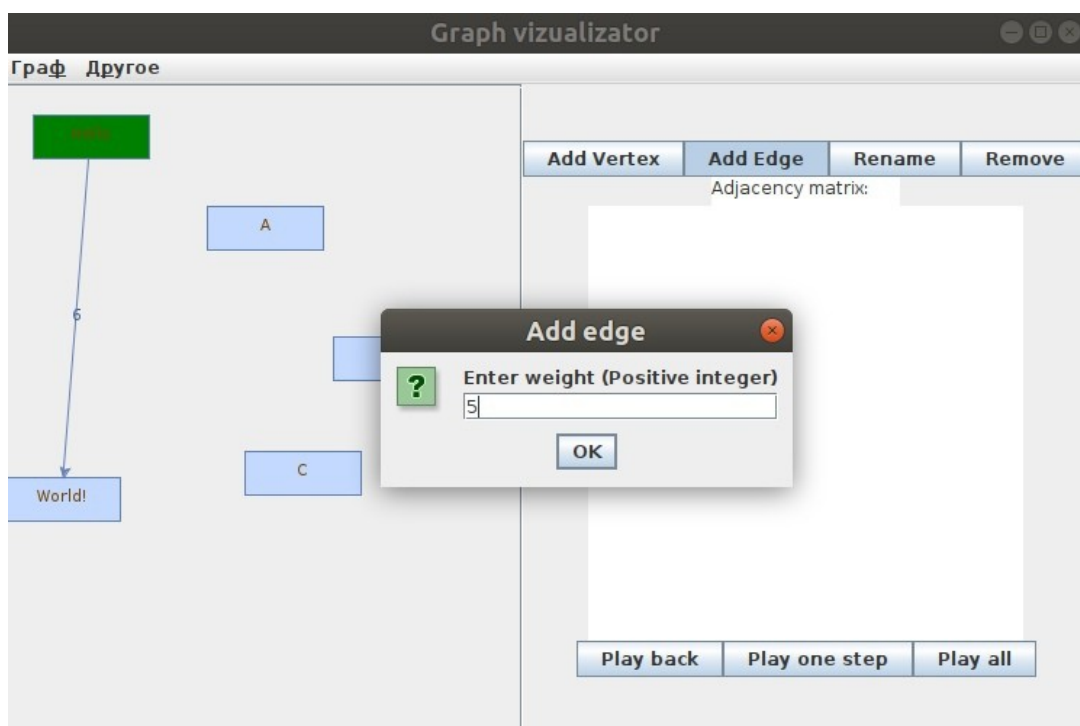


Рисунок 2 – Добавление ребра

При нажатии на “Граф” доступны две опции: загрузить граф из файла – появится окошко, в котором можно будет перемещаться по файловой системе для нахождения файла с графом и создать новый граф (подразумевается, что данное действие очистит рабочую область).

При нажатии на “Другое”, можно будет просмотреть вкладки “справка” и “о программе”.

Сделать шаг. Данная опция позволит шаг за шагом увидеть работу алгоритма.

Сделать шаг назад. Данная опция дает возможность откатиться на один шаг назад в визуализации алгоритма.

Запустить полностью. Данная опция позволит запустить алгоритм на графе без демонстрации пошаговой обработки графа.

На рисунке 1 также видно поле “Adjacency matrix”, которое представляет собой матрицу кратчайших путей, которая будет изменяться в процессе работы алгоритма.

Также, слева видна рабочая область, в которой будет находиться граф.



### **1.1.2 Требования к входным данным**

Входные данные можно задать двумя способами: вручную создать граф, добавив вершины и рёбра в окне программы или загрузить текстовый файл, содержащий матрицу смежности графа, при помощи кнопки “Loading”. Если пользователь хочет загрузить текстовый файл, то он должен поместить туда данные следующим образом: на первой строке находится количество строк матрицы смежности, на второй – количество столбцов, после чего располагается сама матрица смежности графа. В случае неверных данных в текстовом файле, программа получит исключение об ошибке и уведомит об этом пользователя, после чего завершится.

Исходя из алгоритма Флойда-Уоршелла, требования к входным данным следующие: веса рёбер следует подавать неотрицательными (алгоритм не предназначен для работы с отрицательными рёбрами) и целыми, т.к. при работе с вещественными весами, высокая точность вычислений не гарантируется.

### 1.1.3 Требования к архитектуре

Проект выполняется средствами языка программирования Java. Для визуализации используется библиотека Swing. Данная библиотека была выбрана потому, что предоставляет большой набор связанных с ней библиотек, которые в свою очередь позволяют визуализировать графы. В частности, в данном проекте используется библиотека jgraphx (<https://github.com/vlsi/jgraphx-publish>).

В реализации алгоритма Флойда-Уоршелла граф хранится в матрице смежности и все действия производятся с ней. Тип данных элементов матрицы – целые числа. Результатом работы алгоритма является набор кратчайших путей между вершинами.

## **1.2. Уточнение требований после попытки сдачи спецификации**

После сдачи спецификации был добавлен следующий функционал: кнопка “Play back”, которая позволяет откатиться на один шаг назад в пошаговом выполнении алгоритма, вкладки “Граф” и “Другое”. Первая вкладка позволяет очистить рабочую область и загрузить граф из файла, а вторая - просмотреть вкладки “справка” и “о программе” соответственно.

### **1.3. Уточнение требований после сдачи 1-ой версии**

#### **1.4. Уточнение требований после сдачи 2-ой версии**

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

### **2.1. План разработки**

До 01.07.2021 – Распределение по бригадам и выбор темы мини-проекта

До 07.07.2021 – Сдача вводного задания

До 07.07.2021 – Согласование спецификации. Создание прототипа графического интерфейса.

До 10.07.2021 – Сдача второго этапа

До 14.07.2021 – Сдача финальной версии мини-проекта

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

Ковалёв Павел – разработка алгоритма и структуры данных, хранящей граф, ведение документации, тестирование.

Борисовский Виктор – разработка и дизайн графического интерфейса

Прокофьев Михаил – лидер, объединение алгоритма и GUI, координация работы алгоритмиста и разработчика интерфейса.

### **3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ**

#### **3.1. Структуры данных**

В качестве структуры данных используется вектор векторов, который представляет из себя матрицу смежности графа. Данная структура данных используется в алгоритме для вычисления кратчайших путей в графе.

### **3.2. Основные методы**



## 4. ТЕСТИРОВАНИЕ

### 4.1. Тестирование алгоритма

Для тестирования алгоритма был написан класс *ReleaseFloydTester*, который содержит юнит-тесты методов класса *ReleaseFloyd*. Тестировались следующие методы:

*fillMatrix(Vector <Vector<Integer>>)*

*run\_step(Vector<Vector<Integer>> , mxGraph )*

*run\_all(Vector <Vector<Integer>>)*

На *fillMatrix()* был написан один тест, на *run\_step()* и *run\_all()* по 4 теста. В каждом из них методу подавался граф в виде матрицы смежности, после проверялась сумма элементов матрицы смежности со значением, которое должно было получиться.

## **4.2. Тестирование итогового проекта**

## **ЗАКЛЮЧЕНИЕ**

Кратко подвести итоги, проанализировать соответствие поставленной цели и полученного результата.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

*Ниже представлены примеры библиографического описания, В КАЧЕСТВЕ НАЗВАНИЯ ИСТОЧНИКА в примерах приводится вариант, в котором применяется то или иное библиографическое описание.*

1. Иванов И. И. Книга одного-трех авторов. М.: Издательство, 2010. 000 с.
2. Книга четырех авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров, В. В. Васильев. СПб.: Издательство, 2010. 000 с.
3. Книга пяти и более авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров и др.. СПб.: Издательство, 2010. 000 с.
4. Описание книги под редакцией / под ред. И.И. Иванова СПб., Издательство, 2010. 000 с.
5. Иванов И.И. Описание учебного пособия и текста лекций: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.
6. Описание методических указаний / сост.: И.И. Иванов, П.П. Петров. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.
7. Иванов И.И. Описание статьи с одним-тремя авторами из журнала // Название журнала. 2010, вып. (№) 00. С. 000–000.
8. Описание статьи с четырьмя и более авторами из журнала / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название журнала. 2010, вып. (№) 00. С. 000–000.
9. Иванов И.И. Описание тезисов доклада с одним-тремя авторами / Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПб, 2010, С. 000–000.
10. Описание тезисов доклада с четырьмя и более авторами / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПб, 2010, С. 000–000.

11. Описание электронного ресурса // Наименование сайта. URL: <http://east-front.narod.ru/memo/latchford.htm> (дата обращения: 00.00.2010).
12. ГОСТ 0.0–00. Описание стандартов. М.: Изд-во стандартов, 2010.
13. Пат. RU 000000000. Описание патентных документов / И. И. Иванов, П. П. Петров, С. С. Сидоров. Оpubл. 00.00.2010. Бюл. № 00.
14. Иванов И.И. Описание авторефератов диссертаций: автореф. дисс. канд. техн. наук / СПбГЭТУ «ЛЭТИ», СПб, 2010.
15. Описание федерального закона: Федер. закон [принят Гос. Думой 00.00.2010] // Собрание законодательств РФ. 2010. № 00. Ст. 00. С. 000–000.
16. Описание федерального постановления: постановление Правительства Рос. Федерации от 00.00.2010 № 00000 // Опубликовавшее издание. 2010. № 0. С. 000–000.
17. Описание указа: указ Президента РФ от 00.00.2010 № 00 // Опубликовавшее издание. 2010. № 0. С. 000–000.

## **ПРИЛОЖЕНИЕ А**

### **ИСХОДНЫЙ КОД ПРОГРАММЫ**

полный код программы должен быть в приложении, печатать его не надо