

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

**ОТЧЕТ**  
**по учебной практике**  
**Тема: Минимальное остовное дерево**

Студент гр. 0382	_____	Кондратов Ю. К.
Студент гр. 0382	_____	Литягин С. М.
Студент гр. 0382	_____	Сергеев Д. А.
Руководитель	_____	Фирсов М. А.

Санкт-Петербург

2022

## ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Кондратов Ю. К. группы 0382

Студент Литягин С. М. группы 0382

Студент Сергеев Д. А. группы 0382

Тема практики: Минимальное остовное дерево

Задание на практику:

Командная итеративная разработка визуализатора алгоритма на Kotlin и Java с графическим интерфейсом.

Алгоритм: алгоритм Краскала.

Сроки прохождения практики: 29.06.2022 – 12.07.2022

Дата сдачи отчета: 00.07.2020

Дата защиты отчета: 00.07.2020

Студент	_____	Кондратов Ю. К.
Студент	_____	Литягин С. М.
Студент	_____	Сергеев Д. А.
Руководитель	_____	Фирсов М. А.

## **АННОТАЦИЯ**

Кратко (в 8-10 строк) указать цель и основное содержание практики.

## **SUMMARY**

Briefly (8-10 lines) to describe the the purpose and main contents of the practice work.

## СОДЕРЖАНИЕ

	Введение	5
1.	Требования к программе	6
1.1.	Исходные требования к программе	0
1.1.1.	Требования к вводу исходных данных	0
1.1.2.	Требования к графическому интерфейсу	0
1.1.3.	Требования к работе алгоритма	
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 кода алгоритма	0
4.3	...	0
	Заключение	0
	Список использованных источников	0
	Приложение А. Исходный код – только в электронном виде	0

## **ВВЕДЕНИЕ**

### **Цель работы.**

Изучить новый язык программирования и реализовать визуализатор алгоритма Краскала на Kotlin и Java с графическим интерфейсом.

Реализуемый алгоритм применяется для построения минимального остовного дерева взвешенного связного неориентированного графа.

# 1. ТРЕБОВАНИЯ К ПРОГРАММЕ

## 1.1. Исходные требования к программе

### 1.1.1. Требования к вводу исходных данных

Возможность ввода исходных данных через файл;

Формат входных данных в случае двух вышеперечисленных вариантов должен иметь вид:

$N$  – число ребер графа;

В следующих  $N$  строках " $v_i v_j w_{ij}$ " – ребро графа  $v_i v_j$  с весом  $w_{ij}$ .

Возможность ввода исходных данных через графический интерфейс;

### 1.1.2. Требования к графическому интерфейсу

На рисунке 1 представлен эскиз графического интерфейса.

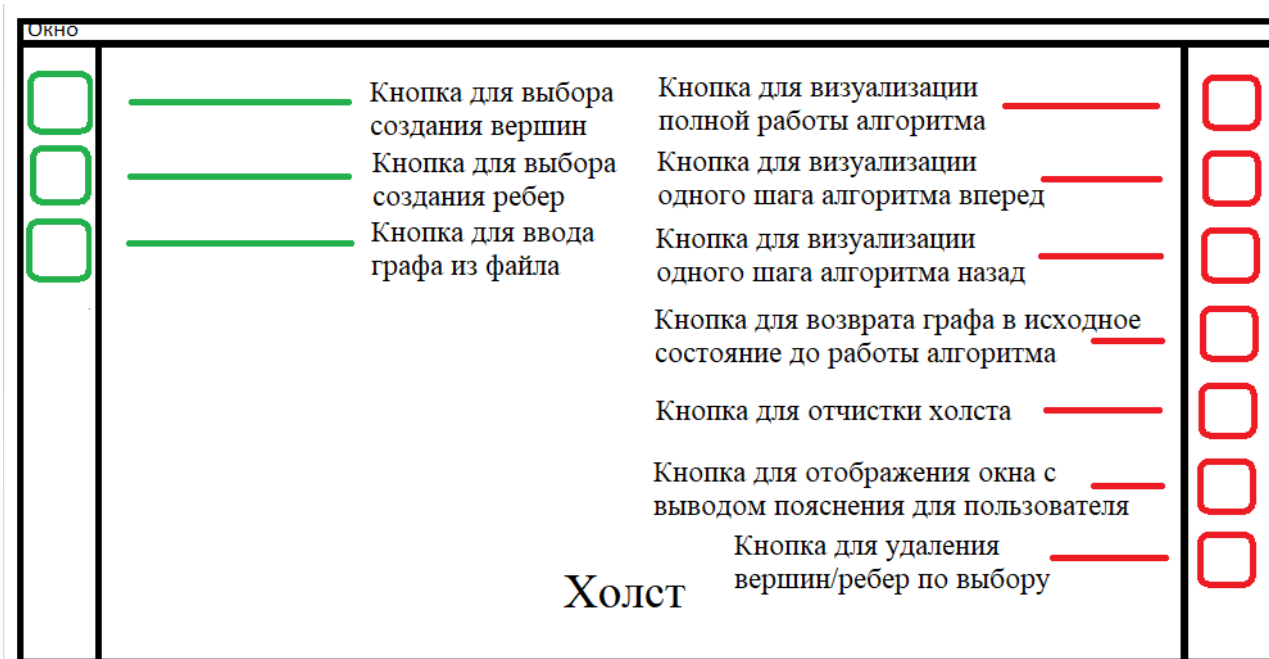


Рисунок 1 – Эскиз графического интерфейса

Возможность создания на холсте по клику мыши вершины с заданием ее наименования (должна быть активна соответствующая кнопка);

Возможность создания ребра между имеющимися вершинами с заданием его веса (должна быть активна соответствующая кнопка). Создание ребра производится левым кликом мыши по вершинам, которые соединяет создаваемое ребро, далее открывается контекстное меню с выбором веса;

Возможность отрисовки графа, ввод которого произведен через файл. При вводе через файл, узлы будут располагаться в вершинах углов правильного многоугольника с центром в середине экрана;

Изначальный граф на холсте имеет черный цвет;

Возможность выполнения полной пошаговой визуализации работы алгоритма по нажатию кнопки (на каждом шаге алгоритма будет выделяться серым цветом; если цикл образуется, то он будет выделен синим цветом, на следующем шаге возвращаются нужные цвета; если ребро не образует цикл, то оно будет перекрашено в красный цвет и вершины, инцидентные ему). При нажатии на кнопку запуска открывается контекстное меню с ползунком выбора скорости работы алгоритма;

Также вместе с полной визуализацией алгоритма будет заполняться файл-лог с пояснениями для пользователя. Выводимое пояснение будет иметь вид:

Описание алгоритма: на каждом шаге выбирается ребро с минимальным весом и не образующее цикл с уже выбранными ребрами.

Начало работы алгоритма;

Выбрано ребро AB с весом  $w$

...

Ребро CD не выбрано, т.к. образует цикл BCD

...

Конец работы алгоритма;

Возможность выполнения одного шага алгоритма вперед по нажатию кнопки (описание выводимых пояснений аналогично полной визуализации алгоритма);

Возможность вывода пояснений для пошаговой версии по нажатию соответствующей кнопки;

Возможность выполнения одного шага алгоритма назад по нажатию кнопки (аналогично выполнению полной визуализации, ребро, выбранное в последнем шаге алгоритма, будет перекрашено в изначальное состояние);

Возможность вернуться в исходное состояние по нажатию кнопки;

Возможность отчистки холста;

Возможность удаления вершин и ребер с холста (должна быть активна соответствующая кнопка). Для удаления нужно будет кликнуть по вершине или ребру, которое нужно удалить;

Возможность перемещения вершин по холсту (должна быть активна кнопка создания вершины). Для этого нужно зажать левую кнопку мыши на вершине и перетащить ее по холсту;

### **1.1.3. Требования к работе алгоритма**

Возможность полного выполнения алгоритма (по завершении работы полученное дерево будет выведено в отдельный файл в формате: " $v_i v_j w_{ij}$ " – ребро графа  $v_i v_j$  с весом  $w_{ij}$ );

Возможность выполнения одного шага алгоритма вперед;

Возможность выполнения одного шага алгоритма назад;

Возможность вывода промежуточных данных;

## **1.2. Уточнение требований после сдачи прототипа**

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

Реализация автоматического добавления нумерации к имени вершины в случае, если такое имя уже существует или в соответствующем поле ничего не введено;

Реализация изменения веса ребра (для этого должна быть активна кнопка создания ребра; при нажатии на уже существующее ребро его вес будет изменен на указанный в соответствующем поле);

Реализация запроса на полный путь к файлу с входными данными (при нажатии на кнопку ввода графа через файл);

Реализация обработки случая, что во время выполнения визуализации алгоритма (пошаговой/полной) может быть изменен граф (возможность перемещения вершин должна быть сохранена);



Реализация изменения размеров окна и холста (можно реализовать в 3-ей версии);

Реализация добавления ребер после перемещения вершин по холсту

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

- Сделать более внятные пояснения к выбору скорости визуализации;
- Выделять выбранную кнопку;
- Исправить некорректную расцветку после цикла;
- Пояснения при наведении мыши;
- Перемещение вершин не должно останавливать алгоритм;
- Исправить баг, что после перемещения вершины удаление не приводит к остановке визуализации;
- Шкала скорости от 1 до 20, где 19 – почти мгновенно (50 мс), 20 – сразу результат шага алгоритма или его полной визуализации.

## 2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

### 2.1. План разработки

План разработки представлен в таблице 1.

Таблица 1 – План разработки

№ п/п	Цель	Срок выполнения
1	Распределение ролей	01.07.2022
2	Реализация приложения, демонстрирующего интерфейс (создание кнопок, реализация создания/удаления вершин и ребер на холсте, реализация отчистки холста). Сдача прототипа	04.07.2022
3	Реализация структуры данных дерева, реализация ввода исходных данных через файл и через GUI	05.07.2022
4	Реализация алгоритма Краскала (полностью все шаги алгоритма и выполнение отдельного шага алгоритма), реализация полной пошаговой визуализации выполнения алгоритма. Сдача 1-ой версии	06.07.2022
5	Исправление замечаний и внесение правок. Реализация визуализации шага алгоритма назад, реализация выбора скорости визуализации полного алгоритма, реализация вывода пояснений для полной и пошаговой визуализаций, реализация выделения циклов другим цветом на шаге. Сдача 2-ой версии	08.07.2022
6	Исправление замечаний и внесение правок. Сдача финальной версии	11.07.2022

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

В данной работе Кондратов Ю. А. реализовывает следующие задачи на языке Java:

Структура данных дерева, алгоритм Краскала (все шаги, один шаг вперед, один шаг назад), вывод пояснений для пошаговой версии алгоритма, ввод исходных данных через файл, вывод остовного дерева в файл.

В данной работе Литягин С. М. реализовывает следующие задачи на языке Kotlin:

Реализация интерфейса приложения, ввод исходных данных через GUI, удаление вершин/ребер с холста, отчистка холста, возврат графа в исходное состояние, перемещение вершин по холсту.

В данной работе Сергеев Д. А. реализовывает следующие задачи на языке Kotlin:

Реализация визуализации полной/пошаговой версии алгоритма, выбор скорости работы визуализации алгоритма.

### **3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ**

#### **3.1. Структуры данных**

#### **3.2. Основные методы**

## **4. ТЕСТИРОВАНИЕ**

### **4.1. Первый подраздел третьего раздела**

### **4.2. Второй подраздел третьего раздела**

## **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. Опубл. 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**  
**НАЗВАНИЕ ПРИЛОЖЕНИЯ**

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