

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

**ОТЧЕТ**  
**по учебной практике**  
**Тема: Разработка визуализатора алгоритма на Java. Минимальное**  
**остовное дерево: алгоритм Прима**

Студентка гр. 0383	_____	Александрович В. П.
Студент гр. 0383	_____	Парфенов В. М.
Студентка гр. 0383	_____	Куртова К. А.
Руководитель	_____	Ефремов М. А.

Санкт-Петербург  
2022

## ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студентка Александрович В. П. группы 0383

Студент Парфенов В. М. группы 0383

Студентка Куртова К. А. группы 0383

Тема практики: Разработка визуализатора алгоритма на Java. Минимальное остовное дерево: алгоритм Прима.

Задание на практику:

Командная итеративная разработка визуализатора алгоритма на Java с графическим интерфейсом.

Алгоритм: алгоритм Прима для нахождения минимального остовного дерева.

Сроки прохождения практики: 29.06.2022 – 12.07.2022

Дата сдачи отчета: 00.07.2020

Дата защиты отчета: 00.07.2020

Студентка гр. 0383	_____	Александрович В. П.
Студент гр. 0383	_____	Парфенов В. М.
Студентка гр. 0383	_____	Куртова К. А.
Руководитель	_____	Ефремов М. А.

## **АННОТАЦИЯ**

Целью проекта является разработка итеративного визуализатора алгоритма Прима с графическим интерфейсом на языке Java. Главными задачами проекта являются создание плана работы и разработка приложения согласно этому плану. Разработка осуществляется в команде, каждый участник которой выполняет свою роль. Конечным продуктом является приложение с графическим интерфейсом, пошагово визуализирующее выполнение алгоритма Прима для поиска минимального остовного дерева на графе, заданном одним из нескольких способов: граф либо генерируется самой программой, либо считывается из текстового файла, либо вводится самим пользователем с использованием инструментов, предоставленных программой. Проект предполагает разработку программы на Java с JDK версии 17 с GUI, реализованным с помощью Swing. Для сборки проекта используется Maven, программа покрывается тестами JUnit.

## **SUMMARY**

The objective of the project is to develop an iterative visualizer of the Prim's algorithm with a graphical interface in Java language. The main tasks of the project are to create a work plan and develop applications according to this plan. Development is carried out in a team, each participant performing their role. The end result is a GUI application that visualizes the execution of the Prim's algorithm step by step. The graph is defined in one of several ways: it is either generated by the program itself, or read from a text file, or entered by the user themselves with the use of tools provided by the program. Is it presumed to build a Java program with JDK version 17 with a GUI implemented with Swing. Maven is used to build the project. The program is covered by JUnit tests.

## СОДЕРЖАНИЕ

	Введение	5
1.	Требования к программе	6
1.1.	Исходные требования к программе*	0
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**

Целью проекта является итеративная разработка приложения с графическим интерфейсом на Java, предоставляющего пользователю инструменты для взаимодействия с ним. Приложение должно пошагово визуализировать выполнение алгоритма Прима. Алгоритм Прима строит минимальное остовное дерево взвешенного связного неориентированного графа.

## 1. ТРЕБОВАНИЯ К ПРОГРАММЕ

### 1.1. Исходные требования к программе

#### 1.1.1. Требования к вводу исходных данных

Входные данные могут быть заданы одним из нескольких способов:

- 1) Загружены из текстового документа (файла с расширением .txt). Файл должен содержать данные в следующем формате: первая строка содержит число, обозначающее количество вершин, последующие строки вида  
*[вершина 1] [вершина 2] [вес ребра]*  
задают ребра графа. Вершины обозначаются их порядковыми номерами.
- 2) Введены пользователем вручную с помощью инструментов, предоставленных программой. Пользователю предоставляется несколько режимов: режим ввода вершин, режим ввода ребер и режим удаления элементов. Вершины добавляются пользователем щелчком мыши на свободную часть холста. Ребра добавляются выбором двух вершин в режиме добавления ребер, пользователя просят ввести вес создаваемого ребра. Пользователь может редактировать граф, удаляя элементы (ребра и вершины) в режиме удаления.
- 3) Сгенерированы самой программой. Данные о графе генерируются и визуализируются программой. Полученный граф должен удовлетворять условиям, требующимися для выполнения алгоритма (т. е. быть связным, неориентированным и взвешенным).

В первых двух случаях обязательна проверка на связность графа.

#### 1.1.2. Требования к визуализации

Графический интерфейс приложения должен содержать панель инструментов и холст, на котором будет представлен граф.

На панели инструментов содержится несколько кнопок, отвечающих за следующие действия:

- 1) Кнопка “Очистить” удаляет все элементы с холста.
- 2) Кнопка “Загрузить” позволяет загрузить файл формата .txt с информацией о графе.
- 3) Кнопка “Сгенерировать” добавляет на холст граф, сгенерированный самой программой.
- 4) Кнопка “Добавить вершину” включает режим добавления вершин. Вершины добавляются щелчком мыши на свободную часть холста.
- 5) Кнопка “Добавить ребро” включает режим добавления ребер. Ребра добавляются выбором двух вершин и вводом значения их веса.
- 6) Кнопка “Добавить вершину” включает режим удаления элементов. Элементы (ребра и вершины) удаляются щелчком мыши по ним.
- 7) Кнопка “Назад” показывает предыдущий шаг алгоритма.
- 8) Кнопка “Вперед” показывает следующий шаг алгоритма.

Визуализация работы алгоритма происходит следующим образом. Каждый шаг алгоритма показывается после нажатия кнопки “Вперед”, с возможностью отката на предыдущий шаг с помощью кнопки “Назад”. Красным цветом на графе выделяются ребра и вершины, входящие в остовное дерево, голубым – рассматриваемые на данном шаге вершины и ребра, т. е. вершины, инцидентные последней вершине, вошедшей в минимальное остовное дерево, и соответствующие им ребра, зеленым — вершина и ребро с минимальным весом, которые войдут в минимальное остовное дерево на следующем шаге. В отдельной строке на каждом шаге алгоритма показывается текущий вес минимального остовного дерева.

Макет графического интерфейса представлен на рис. 1.

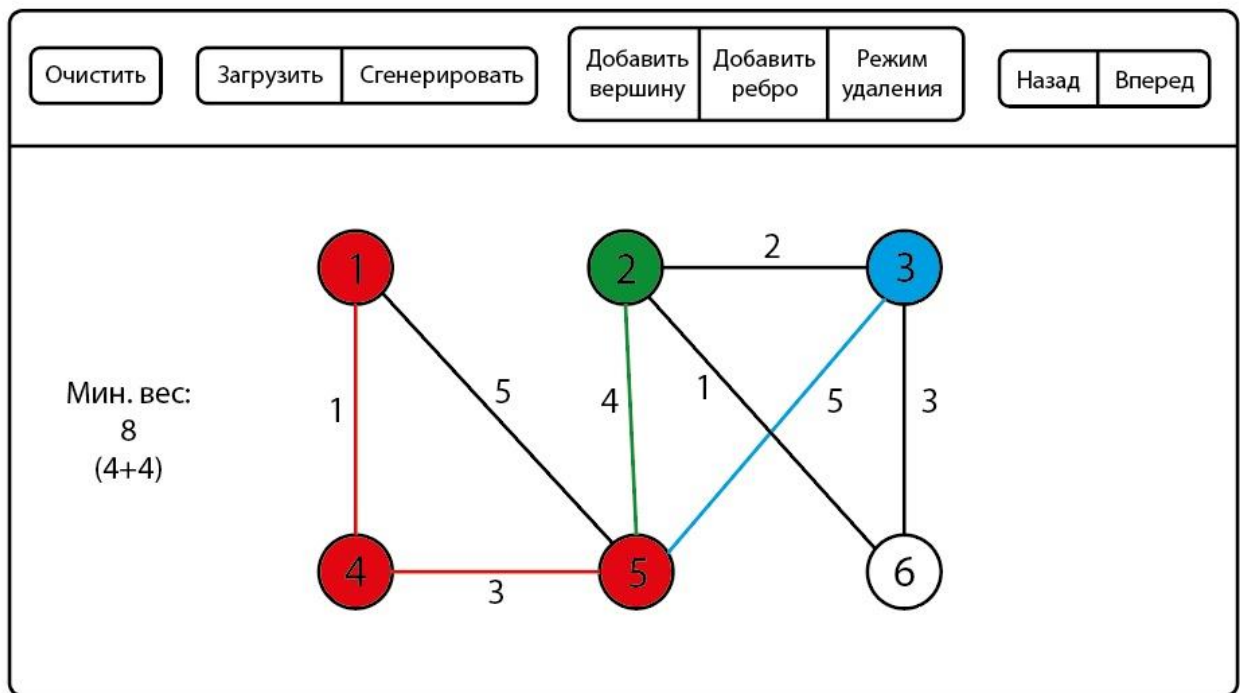


Рисунок 1 — Макет графического интерфейса приложения  
 Диаграмма сценариев использования представлена на рис. 2.

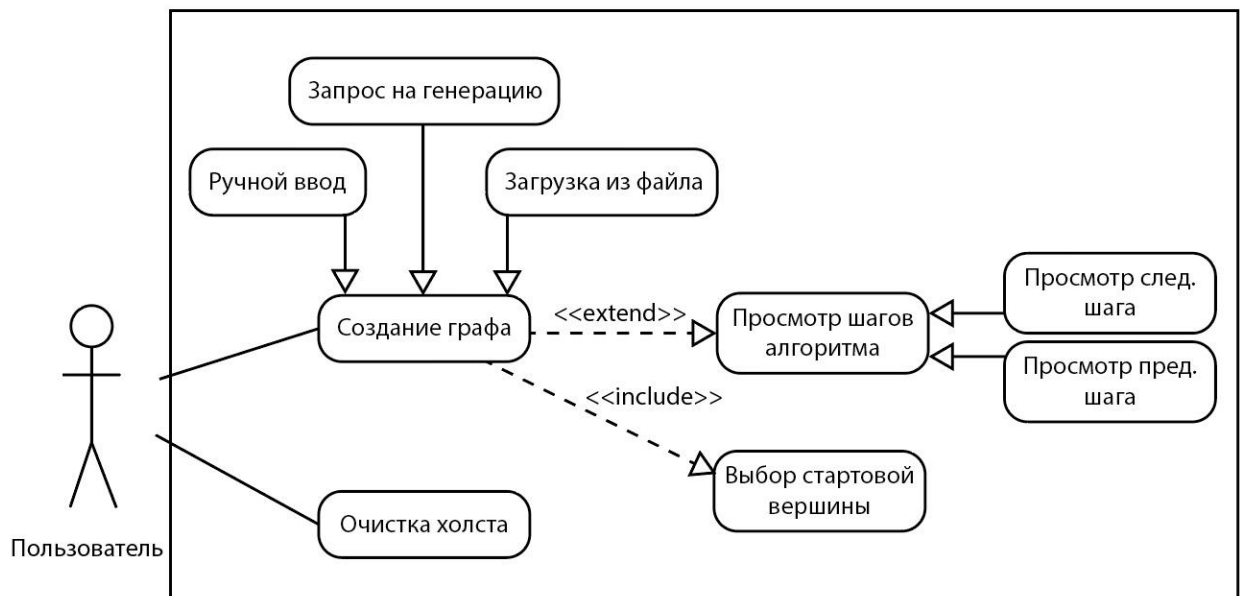


Рисунок 2 — Диаграмма сценариев использования



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

### **2.1. План разработки**

02.07 — создание файла отчёта, заполнение титульного листа с постановкой задачи и сроками, написание раздела спецификаций, описание ролей участников команды; согласование спецификаций; создание Maven-проекта.

04.07 — Создание прототипа интерфейса: интерфейс программы на заглушках, реализация структур данных и алгоритма, тесты для структур данных и алгоритма; создание диаграмм классов и описание сущностей.

06.07 — Рабочий прототип: реализация всех типов генерации данных, выполнение и отображение работы алгоритма; представление диаграмм последовательности, описание тестовых случаев.

08.07 – Корректная работа кнопок, отвечающих за пошаговое исполнение алгоритма, реализация структур данных, отвечающих за пошаговое выполнение алгоритма, реализация тестов для этих структур; создание диаграммы состояний для описания процесса пошагового исполнения алгоритма, пояснения к диаграмме, описание тестовых случаев, описание интерфейса взаимодействия с пошаговым выполнением алгоритма.

10.07 — Сборка проекта в jar-архив, представление итогового отчёта.

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

Александрович Валерия — покрытие программы тестами, связь интерфейса и внутренних структур данных

Парфенов Владислав — визуализация, создание интерфейса программы

Куртова Карина — реализация алгоритма и структур данных

### **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**

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