

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

**ОТЧЕТ**  
**по учебной практике**  
**Тема: Алгоритм Борувки**

Студент гр. 8303	_____	Абибулаев Э.Э.
Студент гр. 8303	_____	Рудько Д.Ю.
Студент гр. 8303	_____	Парфентьев Л.М.
Руководитель	_____	Ефремов М.А.

Санкт-Петербург  
2020

## ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Абибулаев Э.Э. группы 8303

Студент Рудько Д.Ю. группы 8303

Студент Парфентьев Л.М. группы 8303

Тема практики: Алгоритм Борувки

Задание на практику:

Командная итеративная разработка визуализатора алгоритма на Java с графическим интерфейсом.

Алгоритм: Борувки.

Сроки прохождения практики: 29.06.2020 – 12.07.2020

Дата сдачи отчета: 00.07.2020

Дата защиты отчета: 00.07.2020

Студент гр. 8303

\_\_\_\_\_

Абибулаев Э.Э.

Студент гр. 8303

\_\_\_\_\_

Рудько Д.Ю.

Студент гр. 8303

\_\_\_\_\_

Парфентьев Л.М.

Руководитель

\_\_\_\_\_

Ефремов М.А.

## **АННОТАЦИЯ**

Целью данной учебной практики является разработка графического приложения визуализации алгоритма Борувки поиска минимального остовного дерева в графе. Приложение пишется на языке Java с использованием фреймворка JavaFX.

Приложение разрабатывается бригадой из трех человек за несколько итераций.

## **SUMMARY**

The purpose of this educational practice is to develop a graphical application for visualizing the Boruvka algorithm for finding the minimum spanning tree in a graph. The application is written in Java using the JavaFX framework.

The application is developed by a team of three people for several iterations.

## ОГЛАВЛЕНИЕ

1. Требования к программе.....	5
1.1. Начальные требования.....	5
Диаграмма сценариев использования:.....	6
2. План разработки и распределение ролей в команде.....	7
2.1. План разработки.....	7
2.2. Распределение ролей в бригаде.....	7

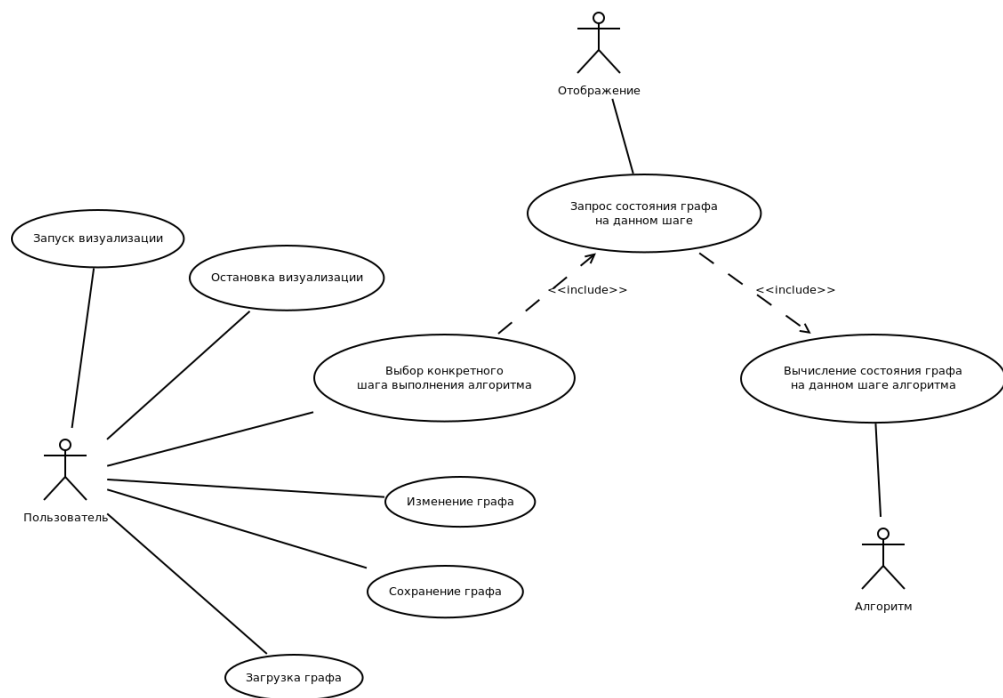
# 1. ТРЕБОВАНИЯ К ПРОГРАММЕ

## 1.1. Начальные требования

- Программа представляет собой визуализацию алгоритма Борувки нахождения наименьшего остовного дерева.
- Граф, на котором выполняется алгоритм, можно загружать из файла, а также создавать или модифицировать в самой программе.
- Визуализация пошаговая: на каждом шаге может происходить следующее:
  - Выбор очередного ребра;
  - Объединение компонент связности.
- Граф можно загрузить из файла и сохранить в файл.
  - При загрузке у части вершин могут быть указаны координаты. У остальных вершин координаты должны выставляться автоматически.
  - При сохранении графа координаты вершин записываются в файл.
- Перед запуском алгоритма граф можно изменять.
  - У вершин можно менять текст.
  - У ребер можно менять веса.
  - Можно добавлять новые ребра и вершины.
  - Можно удалять ребра и вершины.
- После запуска алгоритма граф становится неизменяемым. При выходе из режима визуализации граф снова становится изменяемым.

- При визуализации алгоритма отображается дополнительная информация.
  - Вершины и рёбра из разных компонент связности раскрашены разными цветами. Цвета выбираются случайным образом при запуске алгоритма.
  - При отображении каждого шага, на котором выбирается новое ребро, это ребро подсвечивается.
  - Не выбранные ребра внутри компонент связности (т.е. такие ребра, которые точно не будут выбраны на последующих шагах), будут отображаться специальным образом (тонкими светлыми линиями).

### Диаграмма сценариев использования:



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

### **2.1. План разработки**

- 2 июля: распределение ролей в бригаде; UML-диаграмма сценариев использования.
- 4 июля: графический интерфейс (не рабочий), проектирование классов программы, проектирование поведения программы.
- 6 июля: случайная генерация входных данных, обычная реализация алгоритма (до конца без промежуточных результатов) с отображением результата, план тестирования.
- 8 июля: прототип визуализации, тестирование, пошаговая реализация алгоритма.
- 10 июля: подготовка итогового “релиза”, завершение отчета.

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

- Абибулаев Э.Э.: разработка визуализации и графического интерфейса.
- Рудько Д.Ю.: реализация алгоритма.
- Парфентьев Л.М.: тестирование и сборка приложения.