

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. 9303 | _____ | Лойконен М.Р. |
| Студент гр. 9303 | _____ | Микулик Д.П. |
| Студент гр. 9303 | _____ | Халилов Ш.А. |
| Руководитель | _____ | Ефремов М.А. |

Санкт-Петербург
2021

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Лойконен М.Р., гр. 9303

Студент Микулик Д.П., гр. 9303

Студент Халилов Ш.А., гр. 9303

Тема работы: АЛГОРИТМ БОРУВКИ

Задание на практику:

Разработка визуализатора алгоритма Борувки на Java.

Предполагаемый объем пояснительной записки:

Не менее 10 страниц.

Сроки проведения практики: 01.07.2021-14.07.2021

Дата сдачи отчета: 14.07.2021

| | | |
|--------------|-------|---------------|
| Студент | _____ | Лойконен М.Р. |
| Студент | _____ | Микулик Д.П. |
| Студент | _____ | Халилов Ш.А. |
| Руководитель | _____ | Ефремов М. А. |

АННОТАЦИЯ

Цель учебной практики заключается в разработке GUI-приложения, которое позволяет создать граф и визуализировать алгоритм Борувки для нахождения минимального остовного дерева в созданном графе.

Разработка программы происходит на языке Java командой из 3-х человек. Роли каждого из членов команды распределены в соответствии с имеющимися задачами. Сдача и демонстрация проекта происходит согласно плану разработки.

SUMMARY

The purpose of the training practice is to develop a GUI application that allows you to create a graph and visualize the Boruvka algorithm for finding the minimum spanning tree in the created graph.

The program is developed in Java by a team of 3 people. The roles of each of the team members are distributed in accordance with the existing tasks. Delivery and demonstration of the project takes place according to the development plan.

СОДЕРЖАНИЕ

| | | |
|-----|----------------------------------------------------------------|---|
| | Введение | 5 |
| 1. | Спецификация программы | 6 |
| 1.1 | Прототип интерфейса и USE-CASE-диаграмма | 8 |
| 2. | Организация работы | |
| 2.1 | План разработки | 7 |
| 2.2 | Распределение обязанностей в команде | ? |
| 3. | Описание кода программы | |
| 3.1 | Описание основных классов для реализации интерфейса приложения | ? |
| 3.2 | Описание основных классов для реализации логики приложения | ? |
| 4. | Тестирование | ? |
| | Заключение | ? |
| | Список использованных источников | ? |
| | Приложение А. Исходный код программы | ? |
| | Приложение Б. Результаты тестирования программы. | ? |

ВВЕДЕНИЕ

Целью практической работы является разработка графического приложения, выполняющего визуализацию работы алгоритма Борувки нахождения минимального остовного дерева графа.

При разработке приложения планируется реализация графического интерфейса для задания графа (напрямую, используя функционал приложения или из файла, где перечислены вершины и ребра графа). Также важной частью приложения является пошаговая визуализация алгоритма Борувки.

Разработка осуществляется на языке Java с использованием фреймворка Swing командой из трёх человек. В команде распределяются обязанности – разработка алгоритма, разработка интерфейса и визуализации, сборка и тестирование программы.

1. СПЕЦИФИКАЦИЯ ПРОГРАММЫ

1.1 ПРОТОТИП ИНТЕРФЕЙСА И USE-CASE ДИАГРАММА

Опишем изначальные требования к программе:

1. Программа представляет собой визуализацию алгоритма Борувки нахождения наименьшего остовного дерева.
2. Граф, на котором выполняется алгоритм, можно загружать из файла, а также создавать или модифицировать в самой программе.
3. Предполагается пошаговая визуализация алгоритма с возможностью отката на предыдущий шаг.
4. При визуализации алгоритма отображается дополнительная информация (например, разные компоненты связности окрашиваются в разные цвета).

Прототип интерфейса:

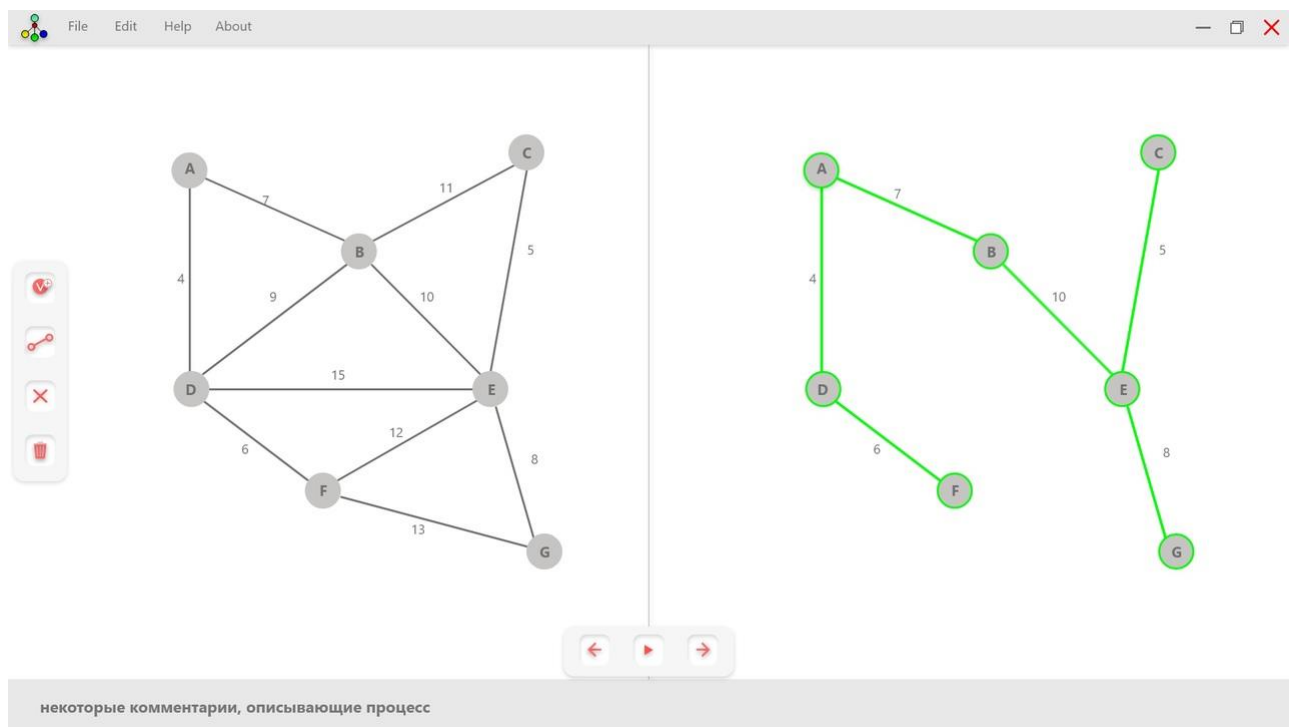


Рисунок 1 - Прототип интерфейса программы

USE-CASE диаграмма:

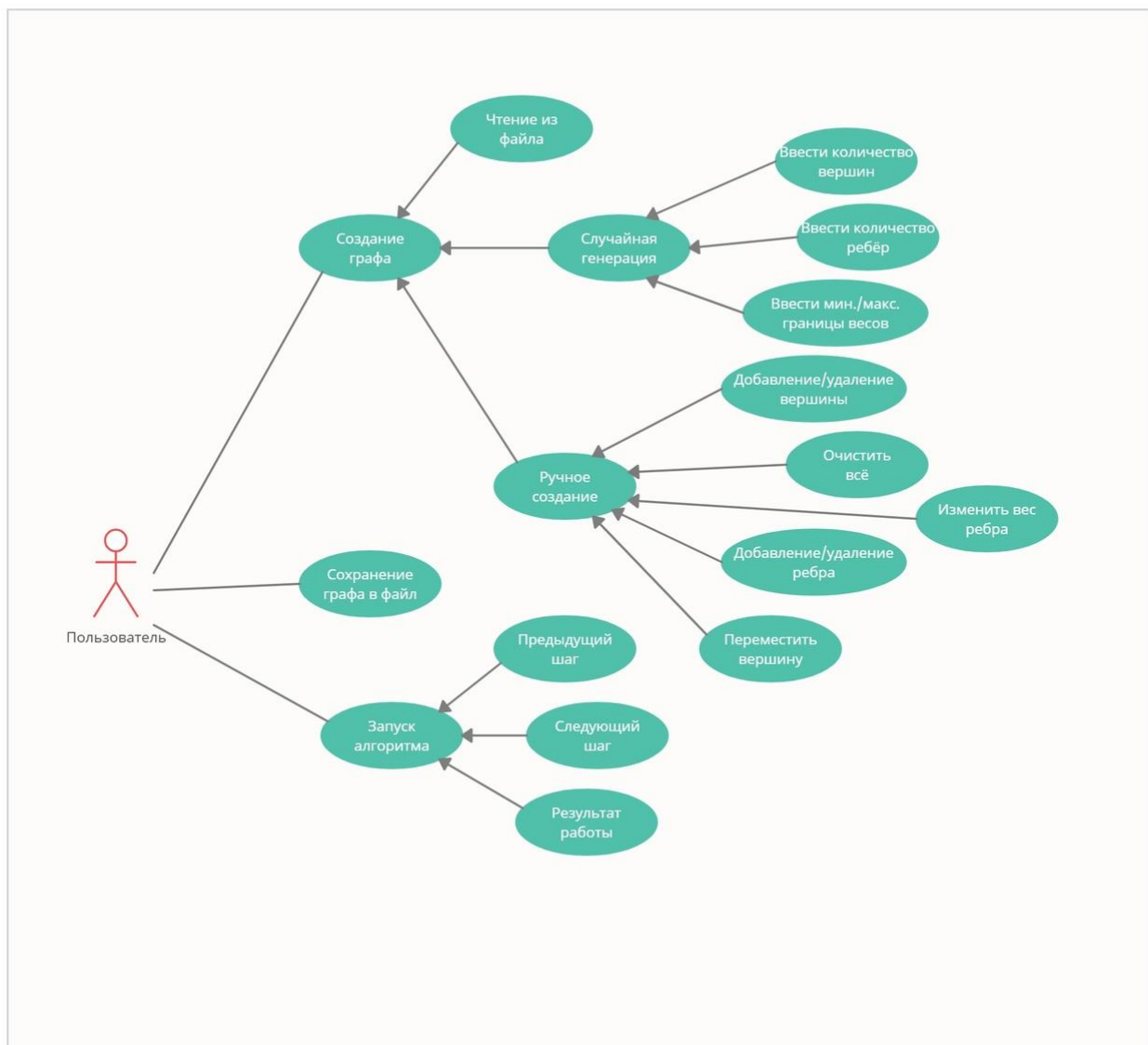


Рисунок 2- USE-CASE диаграмма

2. ОРГАНИЗАЦИЯ РАБОТЫ

2.1 ПЛАН РАЗРАБОТКИ

Задачи работы: изучение основ Java, создание приложения с графическим интерфейсом на Java (используя библиотеку Swing) для визуализации работы алгоритма, сборка (используя Maven) и тестирование программы.

В соответствии с имеющимися задачами был составлен план разработки приложения:

1. К 02.07.2021 изучить основы Java для дальнейшей работы над проектом, создать репозиторий проекта.
2. К 03.07.2021 обсудить спецификацию программы (представить прототип окна GUI приложения, представление USE-CASE диаграммы).
3. К 06.07.2021 разработать интерфейс приложения, начать реализацию базового функционала приложения (реализация алгоритма Борувки), окончательно проработать архитектуру приложения, построить UML-диаграмму классов.
4. К 08.07.2021 реализовать базовый функционал приложения, связать реализацию функционала (базового, т.е. реализация и визуализация алгоритма Борувки) и интерфейса.
5. К 10.07.2021 расширить имеющийся функционал приложения (например, реализовать разные способы задания графа), убрать критичные ошибки и баги.
6. К 12.07.2021 произвести тестирование и отладку программы, при необходимости этапу тестирования может предшествовать рефакторинг имеющегося кода. Сдача проекта.

2.2 РАСПРЕДЕЛЕНИЕ ОБЯЗАННОСТЕЙ В КОМАНДЕ

Обязанности между участниками команды распределены следующим образом:

Лойконен М.Р. – реализация алгоритма Борувки, тестирование программы.

Микулик Д.П. – реализация классов, связующих интерфейс и бизнес-логику (в данном случае сам алгоритм Борувки), написание отчета.

Халилов Ш.А. – реализация графического интерфейса, сборка программы, реализация прототипа графического интерфейса.

3. ОПИСАНИЕ КОДА ПРОГРАММЫ

3.1 ОПИСАНИЕ ОСНОВНЫХ КЛАССОВ ДЛЯ РЕАЛИЗАЦИИ ФУНКЦИОНАЛА ПРИЛОЖЕНИЯ

Для реализации функционала приложения были реализованы следующие классы:

- Graph, Edge, Node – классы, для хранения графа.
- Algorithm – общий интерфейс алгоритмов поиска МОД, BoruvkaAlg – класс, реализующий интерфейс Algorithm, МОД находится при помощи алгоритма Борувки.
- Facade – класс, обобщающий имеющуюся логику (граф и алгоритм).
- CareTaker, Memento, FacadeMemento – набор классов для реализации хранения промежуточных состояний при работе алгоритма (реализуют паттерн «Снимок»)
- Command, LoadCommand, GenerateCommand, VisualizeCommand – классы-команды, которые будут исполняться для загрузки, генерации и визуализации алгоритма Борувки.

Ниже для конкретного понимания реализованных классов и их взаимосвязи представлена UML диаграмма классов.

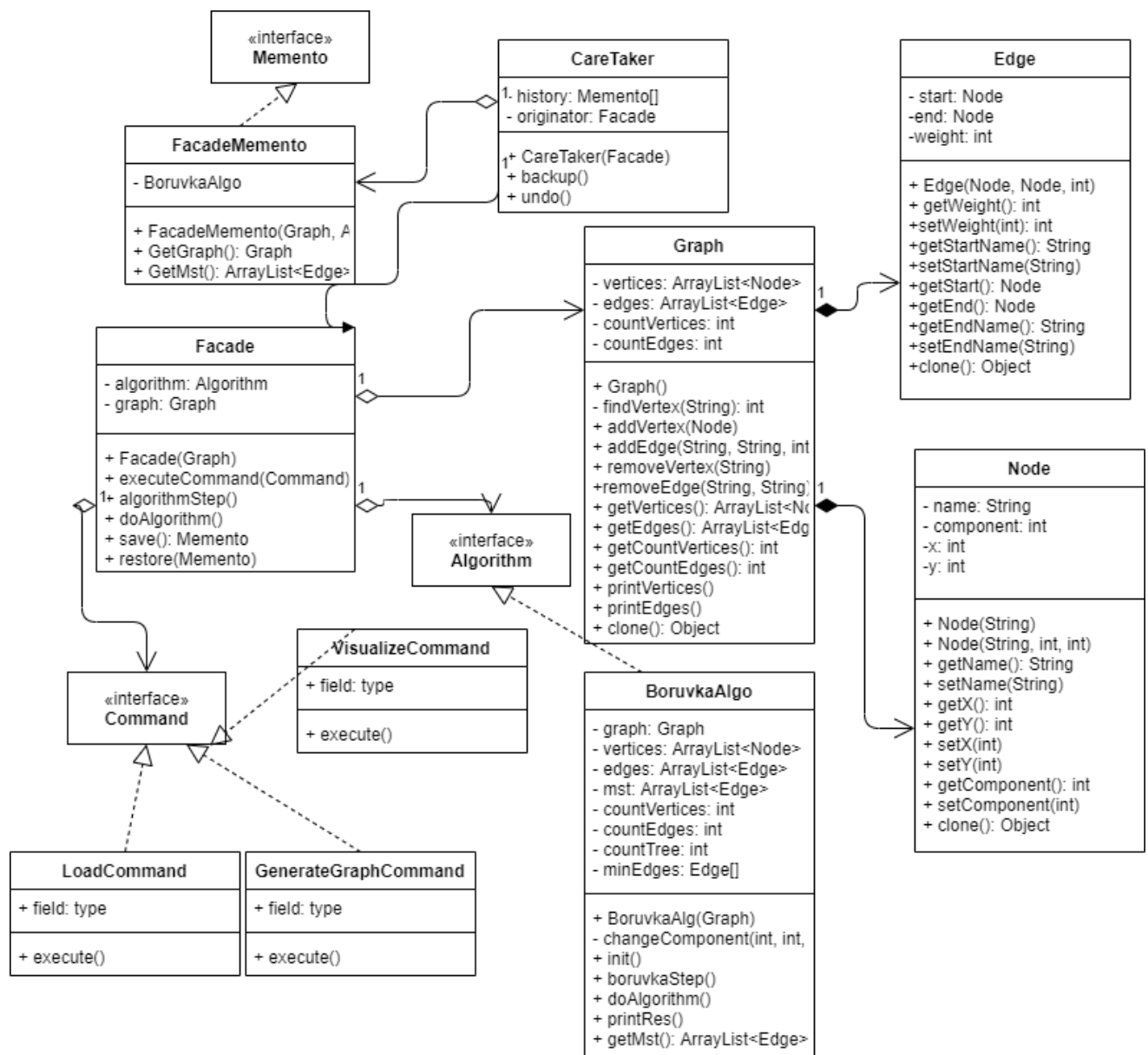


Рисунок 3 - UML диаграмма классов

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ