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

ОТЧЕТ

по учебной практике

по дисциплине «Построение анализ алгоритмов»

Тема: Визуализация алгоритма Куна.

Студент гр. 9381 _____ Игнашов В. М.

Студент гр. 9381 _____ Семёнов А. Н.

Студент гр. 9381 _____ Матвеев А. Н.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2021

ЗАДАНИЕ НА

УЧЕБНУЮ ПРАКТИКУ

Студент Игнашов В. М. группы 9381

Студент Семёнов А. Н. группы 9381

Студент Матвеев А. Н. группы 9381

Тема практики: Визуализация алгоритма Куна.

Задание на практику:

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

Алгоритм: Алгоритм Куна.

Сроки прохождения практики: 29.06.2020 – 12.07.2020

Дата сдачи отчета:

Дата защиты отчета:

Студент гр. 9381	_____	Игнашов В. М.
Студент гр. 9381	_____	Семёнов А. Н.
Студент гр. 9381	_____	Матвеев А. Н.
Преподаватель	_____	Жангиров Т. Р.

АННОТАЦИЯ

Целью практической работы является итеративная разработка программы для визуализации работы алгоритма Куна. Пользователю программы должна быть предоставлена возможность самостоятельно задать входные данные для алгоритма с помощью графического интерфейса, а также возможность пошагового исполнения алгоритма с просмотром промежуточных состояний алгоритма (чередующейся цепи). Разработка осуществляется с использованием языка программирования Java командой из трех человек.

SUMMARY

The purpose of the practical work is the iterative development of the program to visualize the work of Kuhn's algorithm. The user of the program should be given the opportunity to independently specify the input data for the algorithm using the graphical interface, as well as the ability to step-by-step execution of the algorithm with viewing the intermediate states of the algorithm (alternating chain). Development is carried out using the Java programming language by a team of three people.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью практической работы является итеративная разработка программы для визуализации работы алгоритма Куна для нахождения наибольшего паросочетания в двудольном графе. Пользователю программы должна быть предоставлена возможность самостоятельно задать входные данные для алгоритма с помощью графического интерфейса, а также возможность пошагового исполнения алгоритма с просмотром промежуточных состояний алгоритма. В конечной версии программы должна быть предусмотрена возможность сохранения и загрузки исходных данных для алгоритма из файла.

Разработка осуществляется с использованием языка программирования Java командой из трех человек. Каждому участнику команды назначается роль и выполняемые им задачи (разработка интерфейса, логики алгоритма и тестирование программы). В результате выполнения работы должна быть представлена программа, без ошибок собирающаяся из исходных файлов и протестированная на корректность.

1. ТРЕБОВАНИЯ

Проектирование архитектуры:

1) Архитектура должна быть модульной, то есть разделен на разные слои абстракции;

2) Описание модулей проекта через UML диаграммы классов;

3) Описание работы программы и алгоритма через UML диаграммы состояний;

4) Описание работы программы и алгоритма через UML диаграммы последовательности;

5) Желательно соблюдать принципы SOLID;

6) Желательно использовать паттерны проектирования;

Требование к реализации алгоритма:

1) Алгоритм должен реализован так, чтобы можно было использовать любой тип данных. (Например через generic классы);

2) Алгоритм должен поддерживать возможность включения промежуточных выводов и пошагового выполнения.

Требование к проекту:

1) Возможность запуска через GUI и по желанию CLI (в данном случае достаточно вывода промежуточных выводов);

2) Загрузка данных из файла или ввод через интерфейс;

3) GUI должен содержать интерфейс управления работой алгоритма, визуализацию алгоритма, окно с логами работы;

4) Должна быть возможность запустить алгоритма заново на новых данных без перезапуска программы;

5) Должна быть возможность выполнить один шаг алгоритма, либо завершить его до конца. В данном случае должны быть автоматически продемонстрированы все шаги;

6) Должна быть возможность вернуться на один шаг назад;

7) Должна быть возможность сбросить алгоритма в исходное состояние.

На рисунке 1 представлен макет интерфейса программы:

Рис. 1. Макет интерфейса программы.

2. ПЛАН РАЗРАБОТКИ

2.1. План разработки

- ☐ Составление спецификации
- ☐ Распределение ролей
- ☐ Разработка прототипа пользовательского интерфейса
- ☐ Возможность построения графа в пользовательском интерфейсе
- ☐ Просмотр начального и конечного состояний алгоритма
- ☐ Чтение/сохранение графа в файл
- ☐ Возможность пошагового исполнения алгоритма
- ☐ Возможность возврата алгоритма к предыдущим состояниям
- ☐ Тестирование корректности работы программы
- ☐ Реализация дополнительного функционала

Распределение ролей.

Семёнов: Разработка и отладка алгоритма

Матвеев: Тестирование + связь логики и графического интерфейса

Игнашов: Разработка и отладка графического интерфейса

3. ОПИСАНИЕ АРХИТЕКТУРЫ.

Пользователь может вводить данные графа как из файла, так и через интерфейс в виде списков смежности ребер.

Визуализация алгоритма Куна пошаговая: на каждом шаге можно видеть расширение увеличивающейся цепи в паросочетании при помощи поиска в глубину и перенаправление рёбер в том случае, если 2 ребра оказались инцидентные одной вершине. Обрабатываемые рёбра подсвечиваются. В конце алгоритма будет выведен граф и максимальное паросочетание также будет выделено.

При пошаговом исполнении алгоритма пользователь может сделать шаг назад.

Пользователь может сохранять результаты в файл.

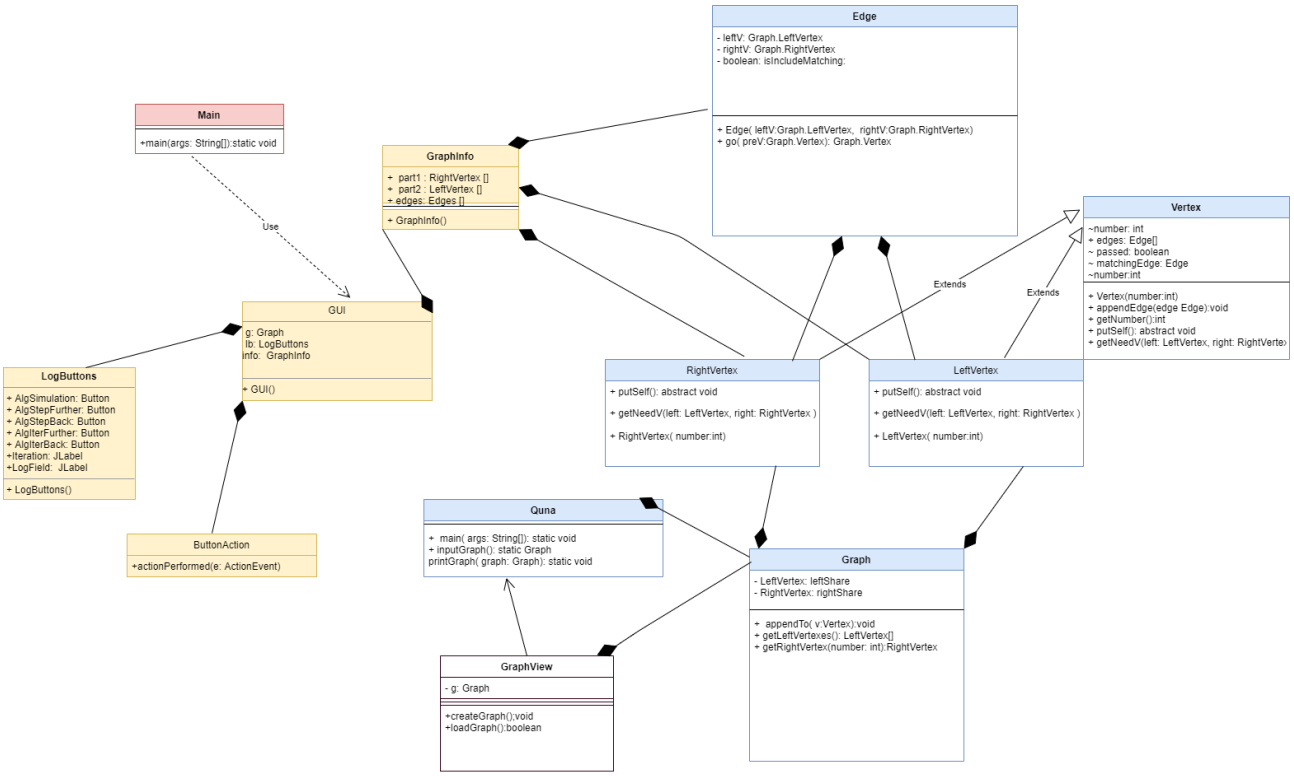

Рис. 2. Диаграмма классов.

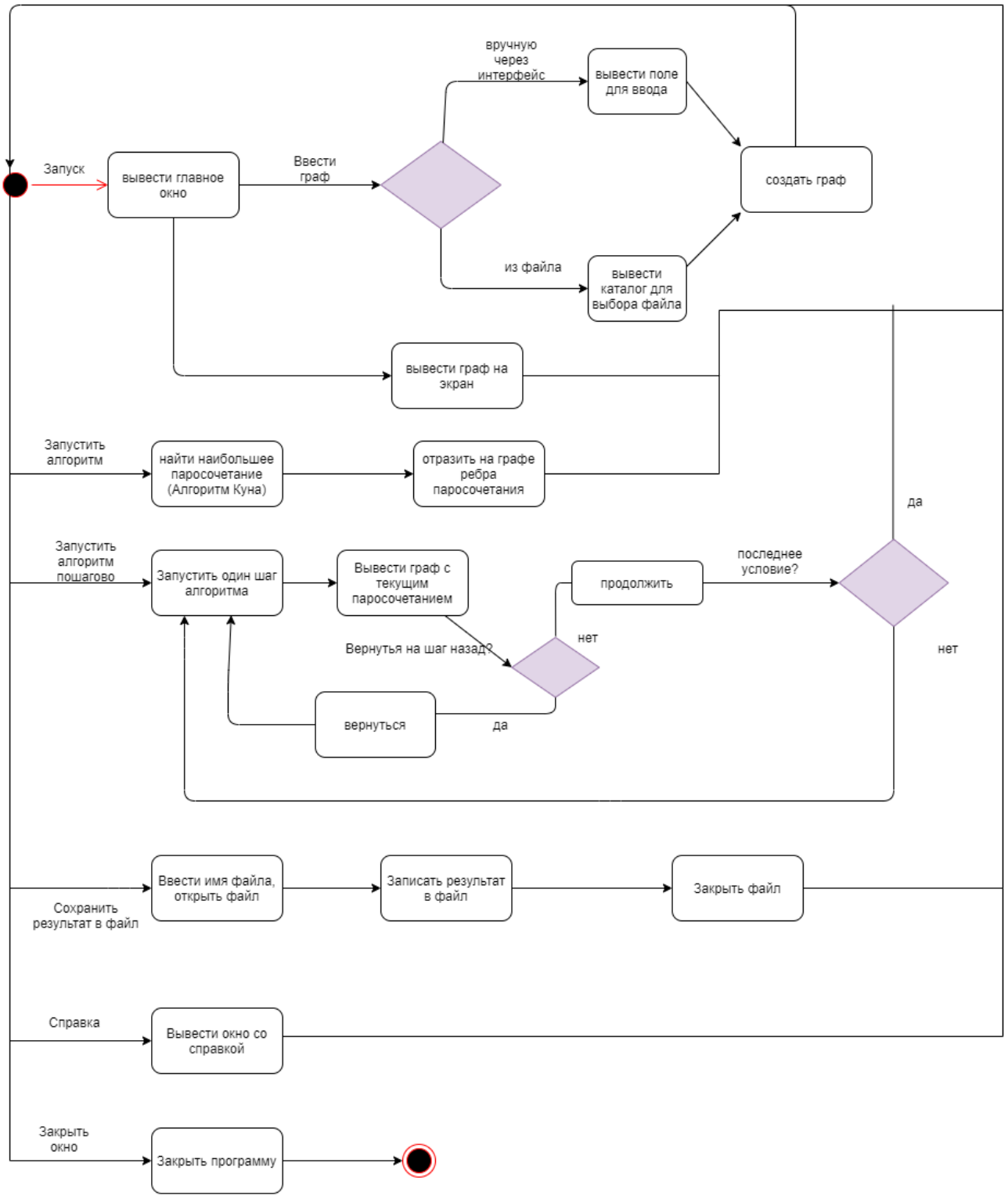

Рис. 3. Диаграмма состояний.

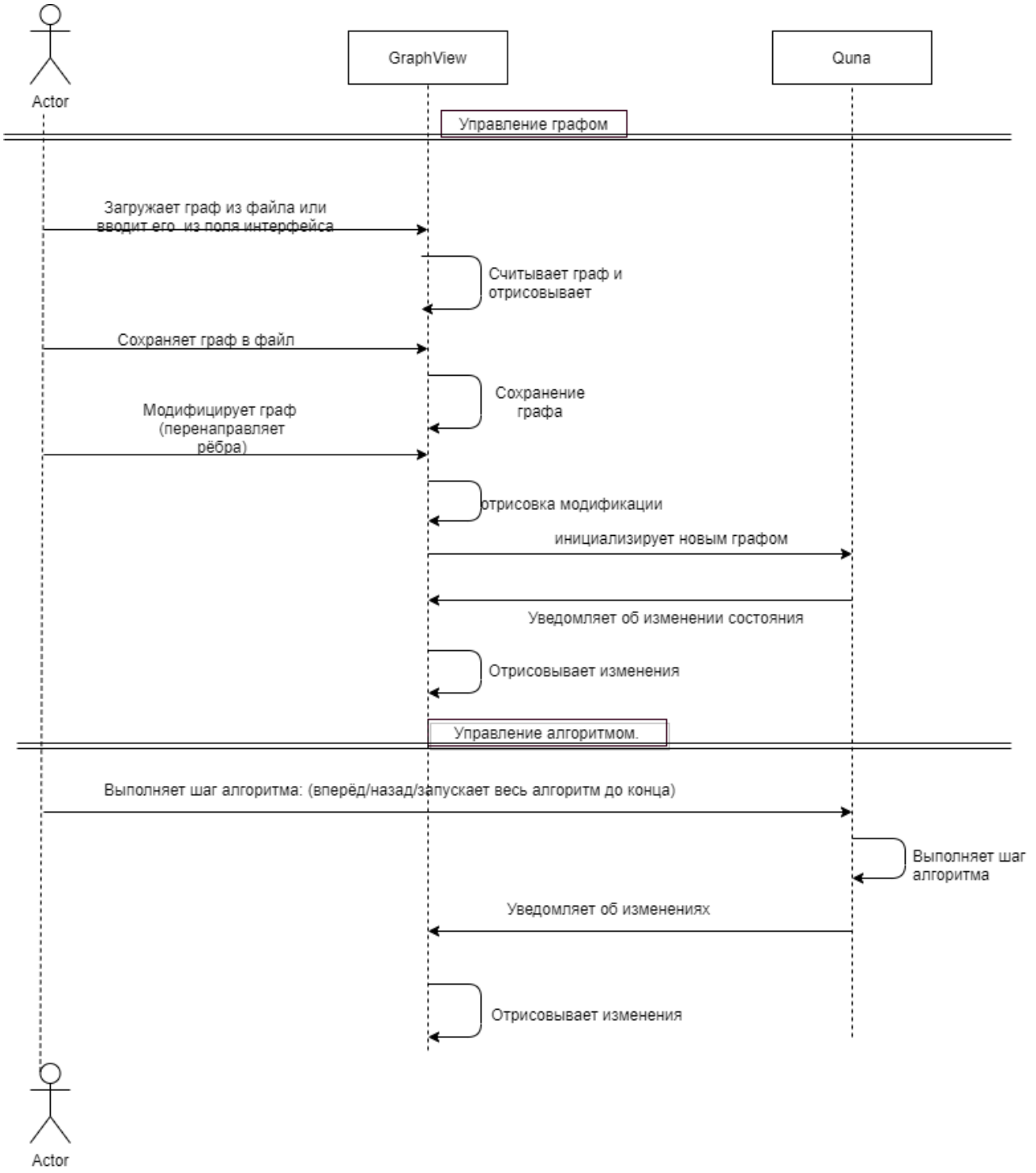

Рис. 4. Диаграмма последовательностей.