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

ОТЧЕТ

по учебной практике

Тема: Мосты

Студент гр. 7303 _____ Алексо А.А

Студент гр. 7303 _____ Бондарчук Н.Р.

Студент гр. 7304 _____ Овчинников Н.В.

Руководитель _____ Ефремов М.А.

Санкт-Петербург
2019

ЗАДАНИЕ

НА УЧЕБНУЮ ПРАКТИКУ

Студент Алексо А.А. группы 7303

Студент Бондарчук Н.Р. группы 7303

Студент Овчинников Н.В. группы 7304

Тема практики: выполнение мини-проекта на языке java.

Задание на практику:

Командная разработка приложения визуализации алгоритма «Поиска Мостов» на Java с графическим интерфейсом.

Алгоритм: Поиск Мостов.

Сроки прохождения практики: 01.07.2019 – 14.07.2019

Дата сдачи отчета: 09.07.2019

Дата защиты отчета: 10.07.2019

Студент	_____	Алексо А.А.
Студент	_____	Бондарчук Н.Р.
Студент	_____	Овчинников Н.В.
Руководитель	_____	Ефремов М.А.

АННОТАЦИЯ

В ходе выполнения данной работы был реализован алгоритм поиска мостов для автоматического поиска мостов в графе. Практическое задание состоит из 4 частей. В первой части расписаны требования к программе в течении выполнения практического задания. Во второй описано распределение работы на группу студентов и план выполнения работы. В третьей части рассматриваются методы решения задачи и используемые структуры данных. В четвертую часть включено тестирование элементов программы. Также сделано заключение, описаны используемая литература.

SUMMARY

In the course of this work, the bridge finder algorithm was implemented. The practical task consists of 4 parts. The first part describes the requirements for the program during the practical task. The second describes the responsibilities of students for work and the work plan. The third part deals with the methods of implementation of the task and the data structures used. The forth part includes testing of the program, a conclusion, describes the source literature.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель задачи – создание мини-проекта, который реализует алгоритм для автоматического поиска мостов в графе. В задаче требуется реализовать диаграмму всех использующихся классов, разработать алгоритм решения задачи и интерфейс к алгоритму. Также необходимо связать исходный код с интерфейсом и написать тесты к логике программы. Далее необходимо создать use-case диаграмму для данной программы в нотации UML.

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1. Исходные требования к программе

1.1.1. На рисунке 1 изображена use-case диаграмма. На старте пользователь может создать граф самостоятельно или считать его из файла. После выбора «Считать из файла» появляется окно с файловым менеджером. Выбрав файл, пользователь загрузит его содержимое в программу, где она автоматически отрисует граф на графической панели. На самой панели доступна кнопка «Поиск мостов».

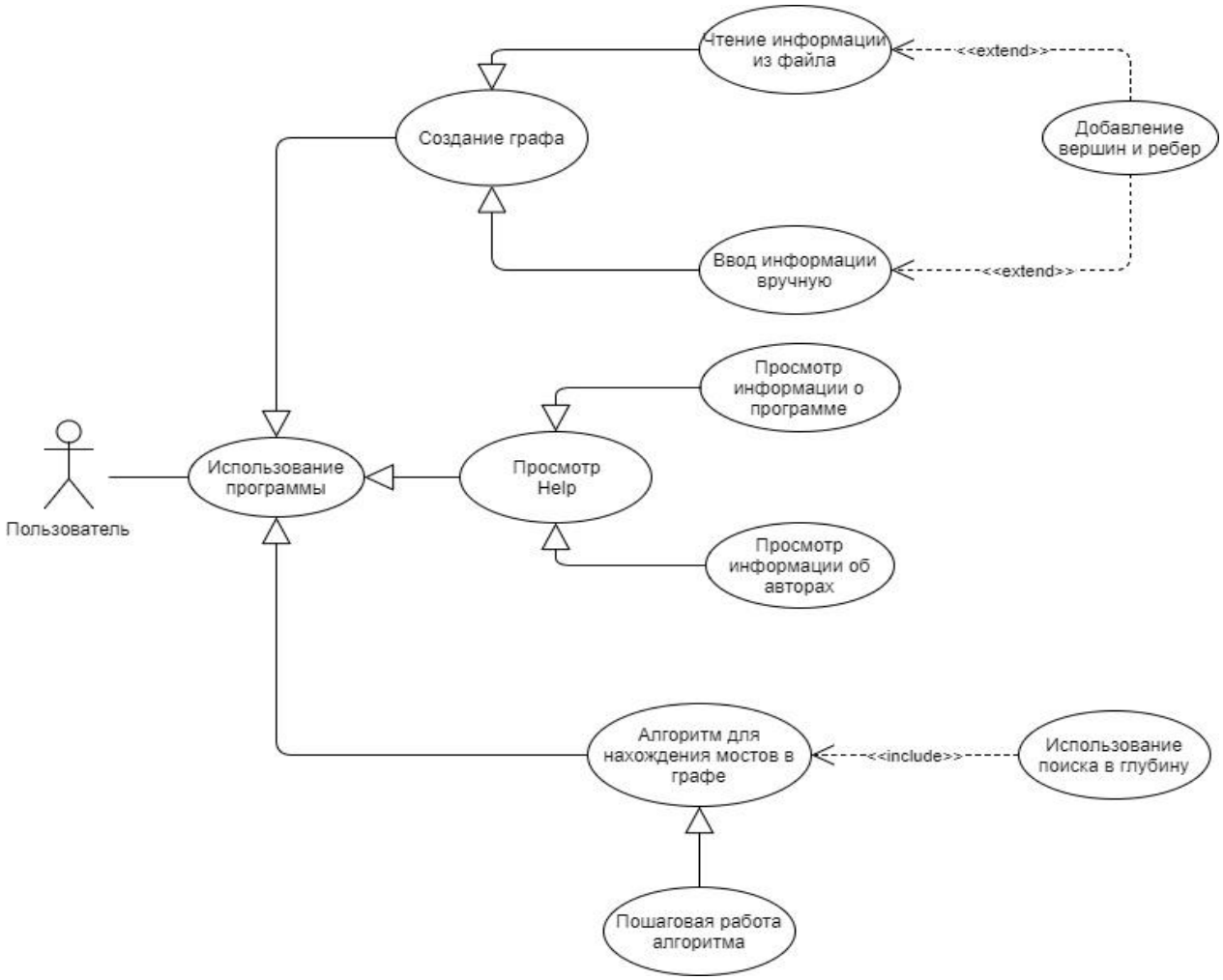

Рисунок 1 – Use-case диаграмма

1.2. Уточнение требований после сдачи прототипа.

1.2.1. Заполнить файл с темой задания

1.2.2. Сделать диаграмму классов.

1.2.3. Исправить недочеты в use-case диаграмме.

1.3. Уточнение требований после сдачи 1-ой версии.

1.3.1. Реализовать сборку проекта с помощью maven.

1.3.2. Сделать unit-тесты для модели.

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки

до 04.07: use case диаграмма, интерфейс без реализации логики.

до 06.07: прототип с демонстрацией функциональности программы.

до 08.07: рабочая версия со сборкой maven.

до 10.07: исправление недочетов первой версии.

2.2. Распределение ролей в бригаде

Алексо А.А.: базовые классы, логика алгоритма, отчет.

Бондарчук Н.Р.: юнит-тестирование, use-case диаграмма, сборка на maven.

Овчинников Н.В.: GUI, соединение интерфейса и логики программы, контроллер.

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

3.1. Использованные структуры данных

На рисунке 2 показана иерархия логики классов. Класс Drawing выполняет роль координатора между UI и состоянием приложения. В нем содержиться объект класса Graph, с которым идет взаимодействие программы.

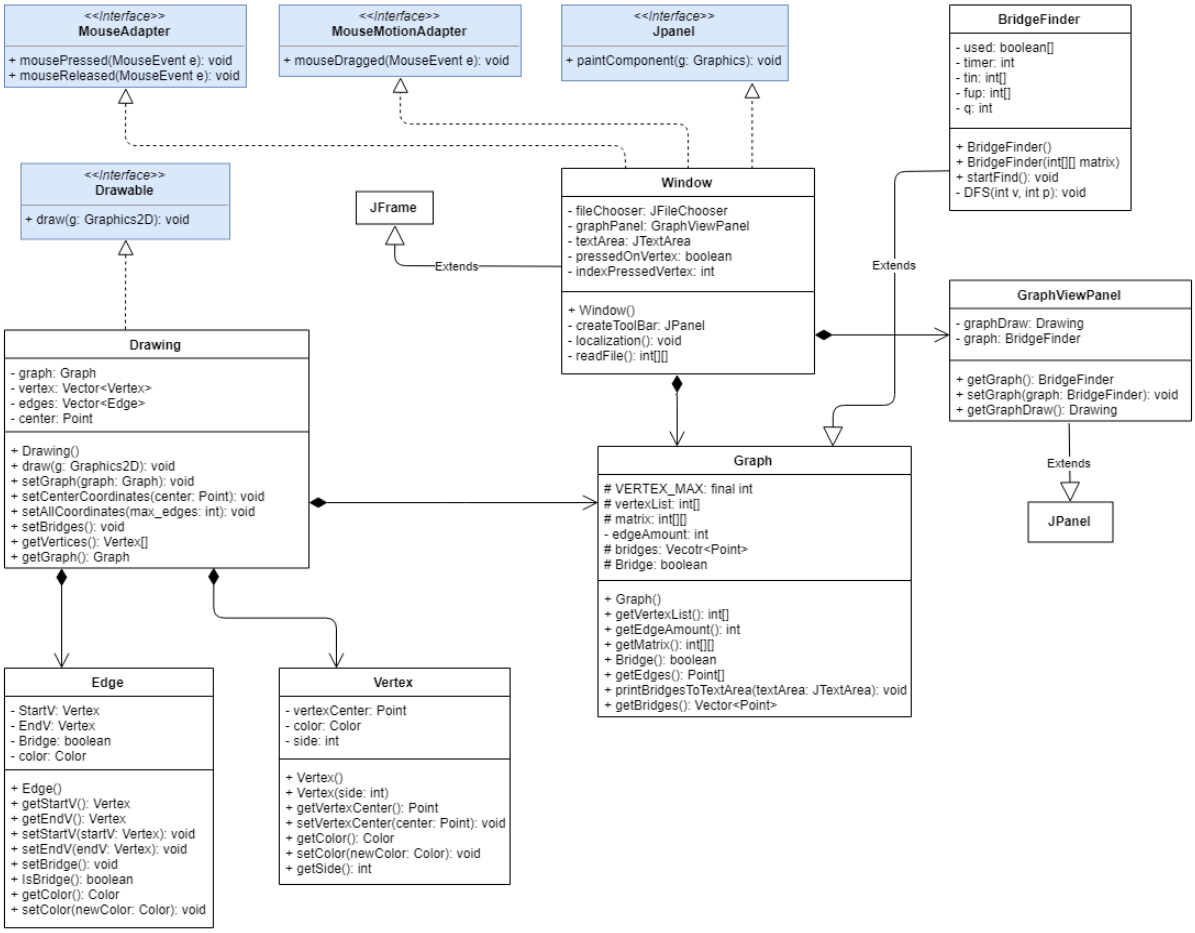

Рисунок 2 – классы логики программы.

3.3. Использование интерфейса

На следующих рисунках (рисунок 3, рисунок 4, рисунок5, рисунок6, рисунок7, рисунок8, рисунок9) представлен интерфейс программы. На рисунке 3 – главное мен. На рисунке 4 – меню выбора файла. На рисунке 5 – панель с графом. На рисунке 6, 7, 8, 9 – функционал программы и пример работы алгоритма поиска мостов.

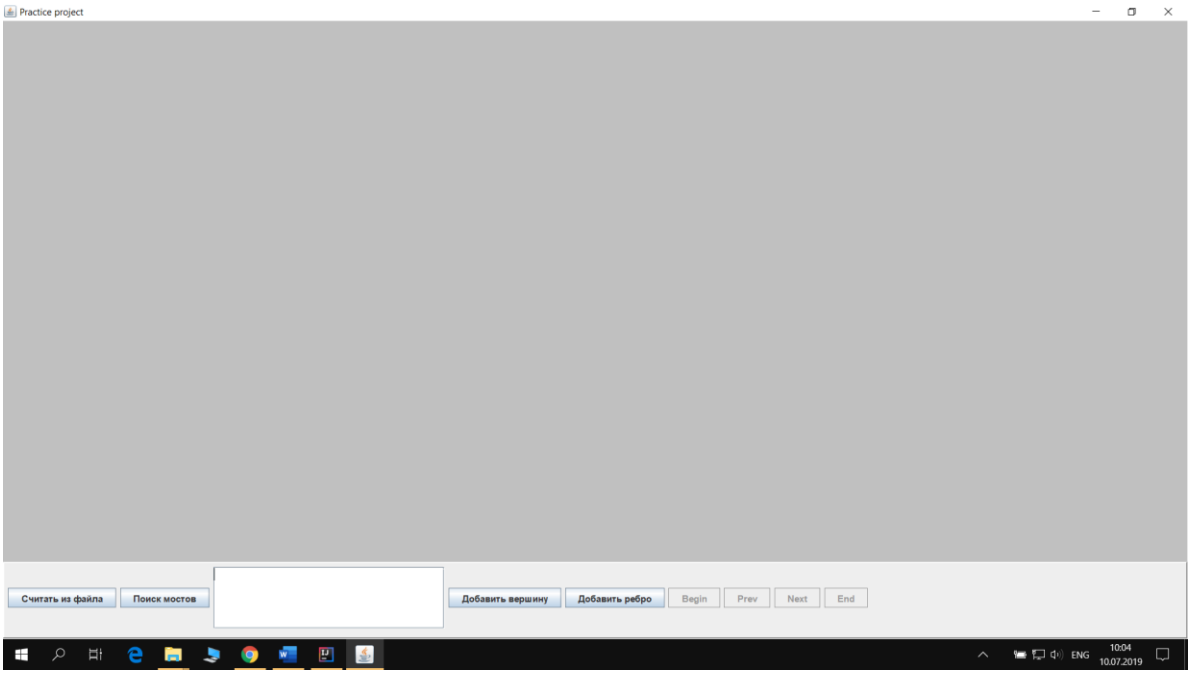

Рисунок 3 – интерфейс стартового меню программы

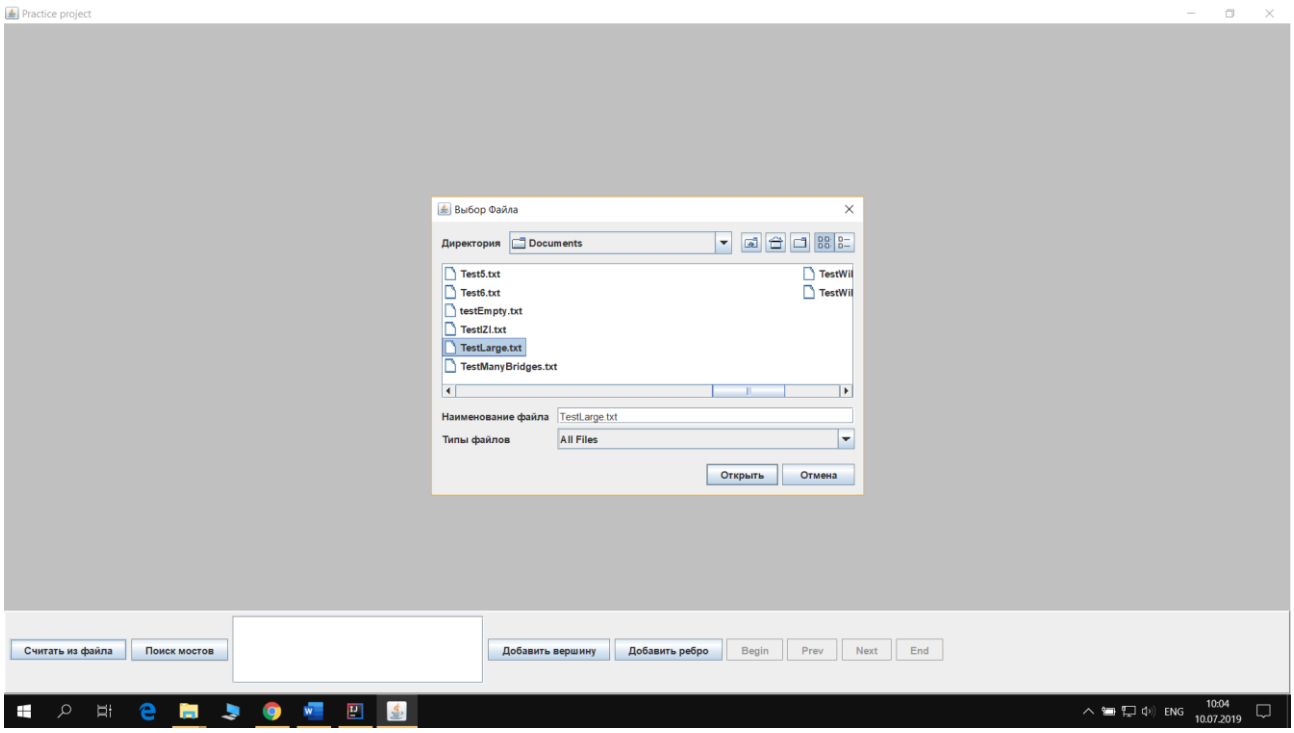

Рисунок 4 – меню выбора файла

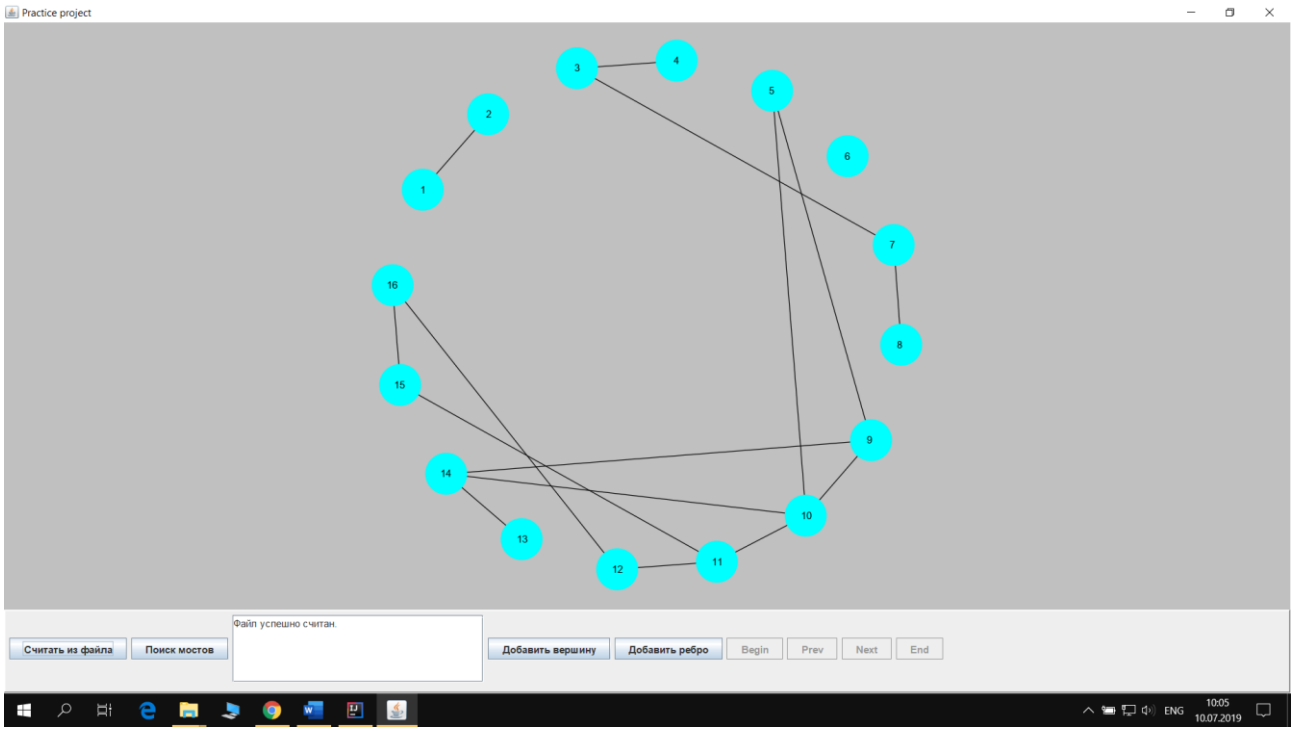

Рисунок 5 – графическая панель и граф

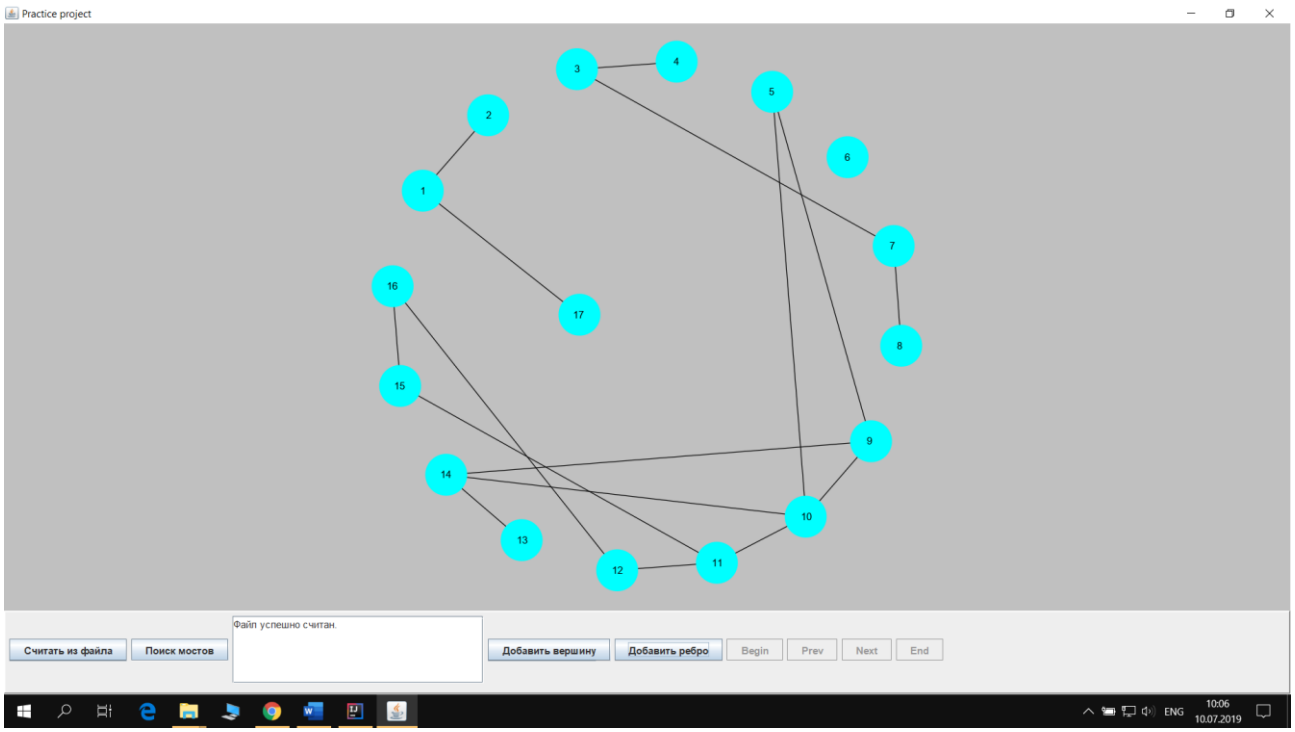

Рисунок 6 – добавление вершин и ребер

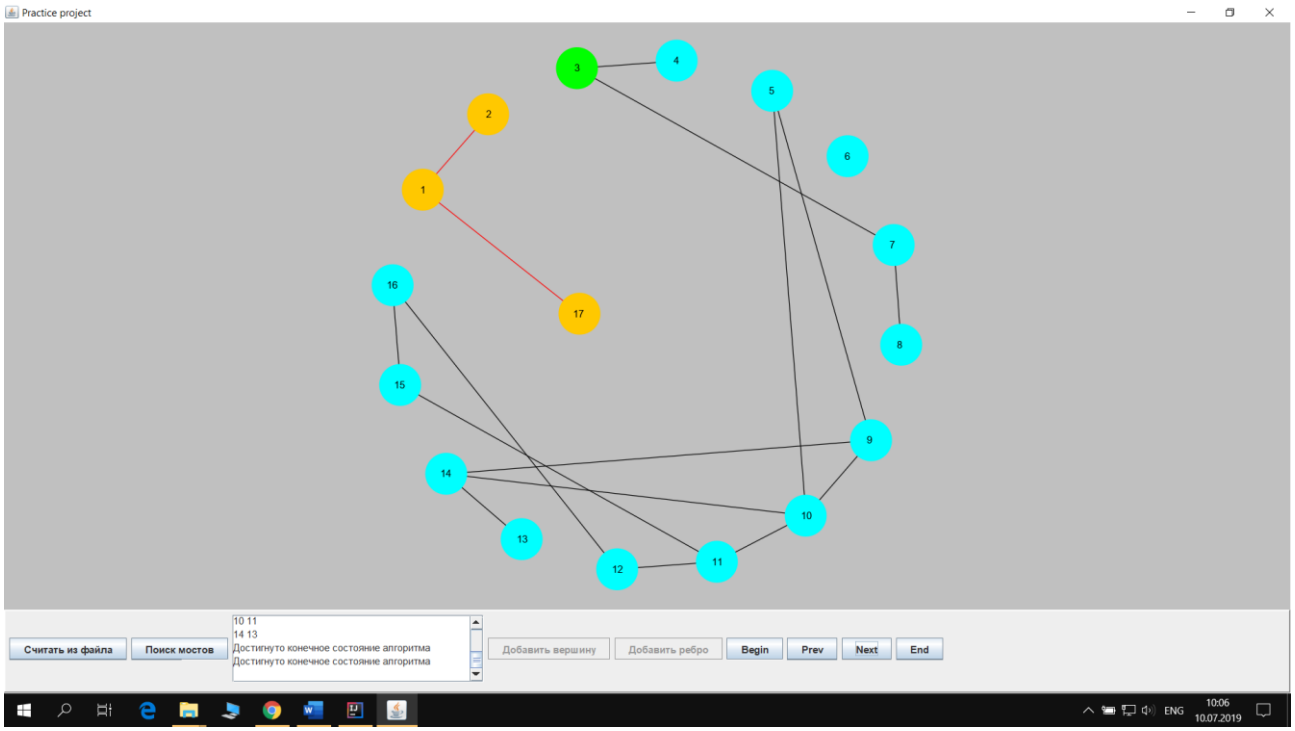

Рисунок 7 – пошаговая работа алгоритма

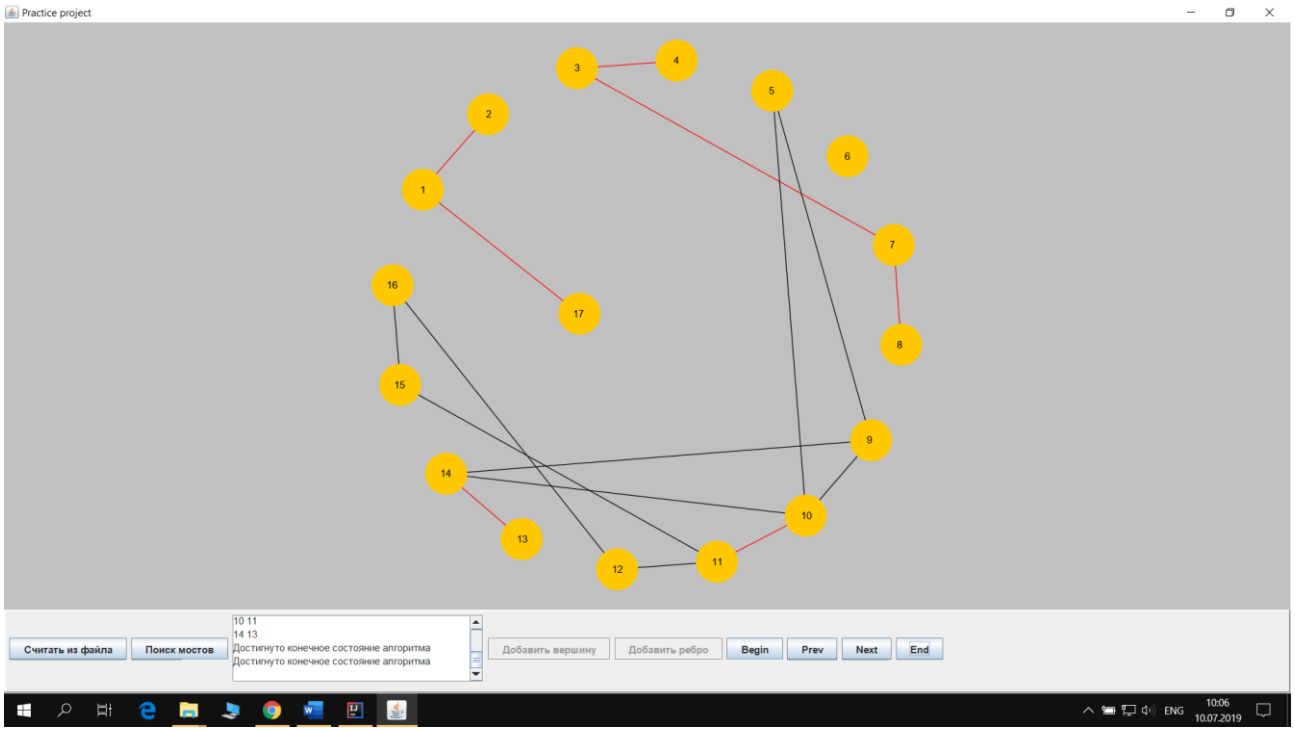

Рисунок 8 – конечное состояние графа

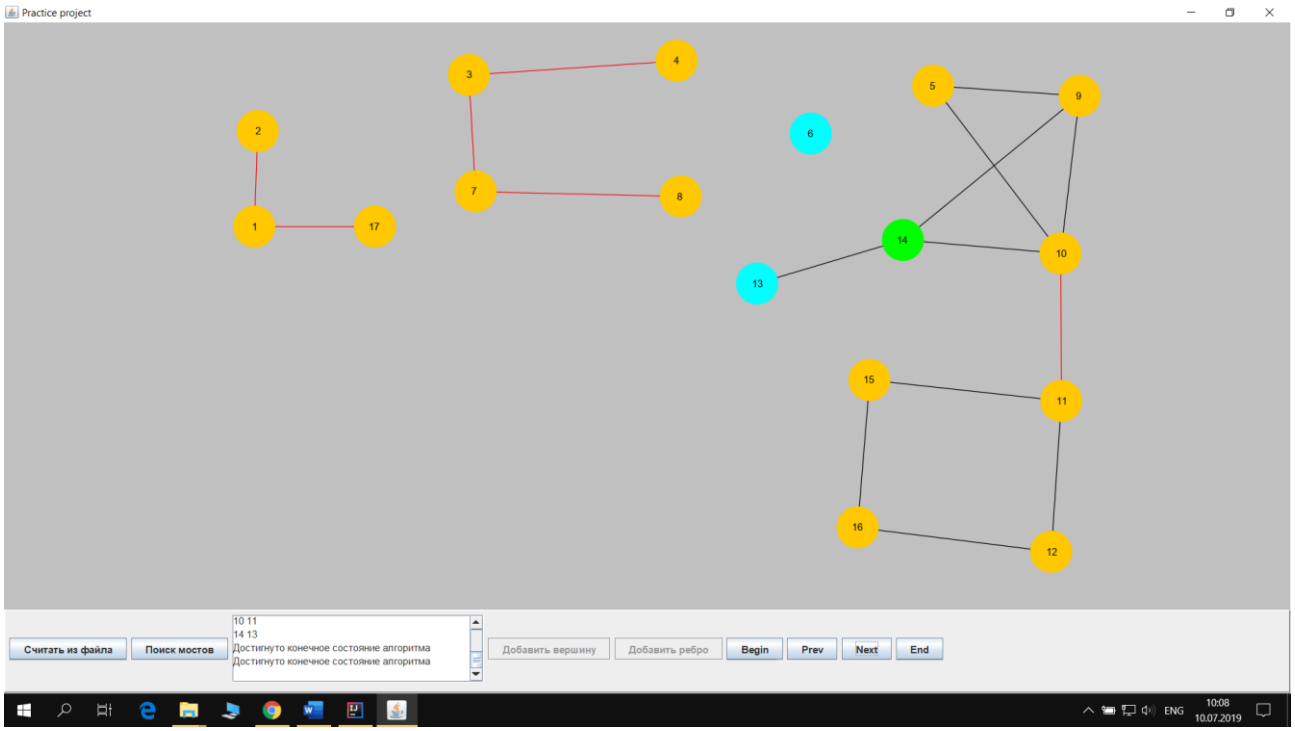

Рисунок 9 – возможность свободного перемещения вершин

4. ТЕСТИРОВАНИЕ

4.1. Тестирование кода алгоритма

Создается классы BridgeFinderTest и GraphTest для создания и хранения модульных тестов.

В методе *FindBridgeT()* создается объект класса BridgeFinder для использования во всех модульных тестах.

5 методов FindBridge() проверяют работу алгоритма.

2 метода getVertexList() проверяют правильность ввода матрицы.

2 метода getEdgeAmount() проверяет правильность ввода ребер в матрицу.

2 метода getMatrixTest() правильность передачи матрицы.

2 метода getEdgesTest() правильность метода getEdges() в графе.

2 метода getBridgesTest() правильность метода getBridges() в графе.

ЗАКЛЮЧЕНИЕ

В ходе разработки мини-проекта в нотации UML были созданы use-case диаграмма и диаграмма классов, по которым была построена модель решения задачи - класс, хранящий данные о состоянии головоломки и класс, вносящий изменения в эти данные по сигналам от UI. Пользовательский интерфейс "Мостов" был реализован при помощи библиотеки Swing. Сборка проекта осуществлялась фреймворком Apache Maven. В итоге после тестирования были получены полноценная реализация алгоритма "Поиска Мостов" с выбором файла, а также опыт разработки на языке Java.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кэти Сьерра и Берт Бейтс Изучаем Java: Москва, 2016, 383-428с.

2. Документация Java™ Platform, Standard Edition 7 API Specification [Электронный ресурс] // Copyright © 1993, 2018, Oracle and/or its affiliates. URL: https://docs.oracle.com/javase/7/docs/api/index.html 04.07.2019

3. Руководство по maven – что такое maven [Электронный ресурс], Apache Maven Project. URL: www.apache-maven.ru 03.07.2019