

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

ОТЧЕТ
по учебной практике
Тема: Сильно связанные компоненты оргграфа

Студентка гр. 8383	_____	Аверина О.С.
Студентка гр. 8383	_____	Максимова А.А.
Студент гр. 8383	_____	Мирсков А.А.
Руководитель	_____	Фирсов М.А.

Санкт-Петербург
2020

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студентка Аверина О.С. группы 8383

Студентка Максимова А.А. группы 8383

Студент Мирсков А.А. группы 8383

Тема практики: **Сильно связанные компоненты орграфа**

Задание на практику:

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

Алгоритм: Косарайю.

Сроки прохождения практики: 29.06.2020 – 12.07.2020

Дата сдачи отчета: 00.07.2020

Дата защиты отчета: 00.07.2020

Студентка		Аверина О.С.
Студентка		Максимова А.А.
Студент		Мирсков А.А.
Руководитель		Фирсов М.А.

АННОТАЦИЯ

Целью данной практической работы является изучение и углубление теоретических знаний языка Java, закрепление материала при разработке собственного пошагового визуализатора алгоритма поиска сильно связанных компонент орграфа, обладающего удобным и понятным пользовательским интерфейсом и предусматривающего поведение пользователя, которое без обработки, может приводить к неопределенному поведению программы.

Данная практическая работа состоит из введения, в котором описана спецификация приложения, сопровождаемая макетом меню и диаграммой сценариев, требований, которые должны быть реализованы в программе, прототипа и промежуточных версий кода, плана разработки приложения и распределения ролей в бригаде, описания особенностей реализации, используемых в работе структур данных и разработанных методов, тестирования работы алгоритма и пользовательского интерфейса, а также заключения и списка источников, используемых при написании программы.

SUMMARY

The purpose of this practical work is to study and deepen the theoretical knowledge of the Java language, consolidate the material when developing your own step-by-step visualizer of the search algorithm for strongly connected components of a digraph, which has a convenient and intuitive user interface and provides user behavior that, without processing, can lead to undefined program behavior.

This practical work consists of an introduction, which describes the specification of the application, accompanied by a menu layout and a diagram of scenarios, requirements that must be implemented in the program, prototype and

intermediate versions of the code, an application development plan and role distribution in the team, description of the implementation features used in the work of data structures and developed methods, testing the operation of the algorithm and the user interface, as well as the conclusion and list of sources used when writing the program.

СОДЕРЖАНИЕ

	Введение	6
1.	Требования к программе	7
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 выполнения данной практической работы, является разработка программы, на базе высокоуровневого языка Java, реализующей пошаговую визуализацию алгоритма поиска сильно связанных компонент орграфа. Визуализатор алгоритма при этом должен обладать понятным и удобным пользовательским интерфейсом.

Реализация поиска сильно связанных компонент основана на алгоритме Косарайю, в котором ключевым аспектом является поиск в глубину с фиксированием времени выхода из каждой вершины орграфа.

Под сильно связными компонентами орграфа подразумевается его максимальные по включению сильно связанные подграфы. Сильно связные подграф - это граф, содержащий некое подмножество вершин данного графа и все ребра, инцидентные данному подмножеству, в котором между любыми двумя вершинами, включенными в него, существуют ориентированные пути из s в t и из t в s , где s и t - две любые вершины подграфа.

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1. Исходные требования к программе

1.1.1. Входные данные

Входные данные, а именно орграф, описываемый вершинами и ориентированными невзвешенными ребрами, может быть задан как список ребер, где указывается “Количество ребер”, а затем строки формата “Начальная вершина, конечная вершина” в текстовом файле, импортируемого в программу при нажатии клавиши “Импорт файла”.

Пример:

```
4
1 3
4 2
3 4
3 2
```

Также может быть построен вручную с помощью использования клавиш, располагаемых на панели управления и нажатия на экран с помощью курсора мыши для выбора местоположения вершин или построения ориентированного ребра между ними. Так, чтобы построить граф нужно выполнить следующую последовательность шагов:

1. Нажатие клавиши “Добавление вершины” с последующим щелчком на место, куда будет добавлена вершина
2. Для добавления ориентированного ребра: выбор клавиши “Добавить ребро”, поочередное нажатие на вершины, между которыми будет построено ориентированное ребро (первая вершина начальная, вторая - конечная)
3. Для отмены добавления вершины или ребра необходимо выбрать одну из кнопок “Удаление ребра”, “Удаление вершины” и щелкнуть на объект

1.1.2. Визуализация

Графический интерфейс представляет собой меню, состоящее из панели управления - набора клавиш, предназначенных для вызовов команд, реализующих построение графа, окна, в котором отображается пошаговая визуализация алгоритма или выводится результат работы программы - раскрашенный орграф, в зависимости от выбора пользователя, а также имеется окно, используемое для вывода текстовых пояснений и промежуточной информации.

Пошаговая визуализация алгоритма:

- 1) Построение графа, введенного пользователем.
- 2) При визуализации поиска в глубину, вершины и ребра, при посещении будут помечаться цветом. Рядом с вершиной будет написано время выхода.
- 3) Затем запуск поиска компонент сильной связности, окрашивание каждой компоненты в свой цвет.

*При этом на каждом шаге будет выводиться описание действия.

Эскиз интерфейса, реализующего визуализацию алгоритма поиска сильно связанных компонент орграфа, а также предоставляющего возможность ввода входных данных одним из нескольких предоставленных способов, изображен на рисунке 1.

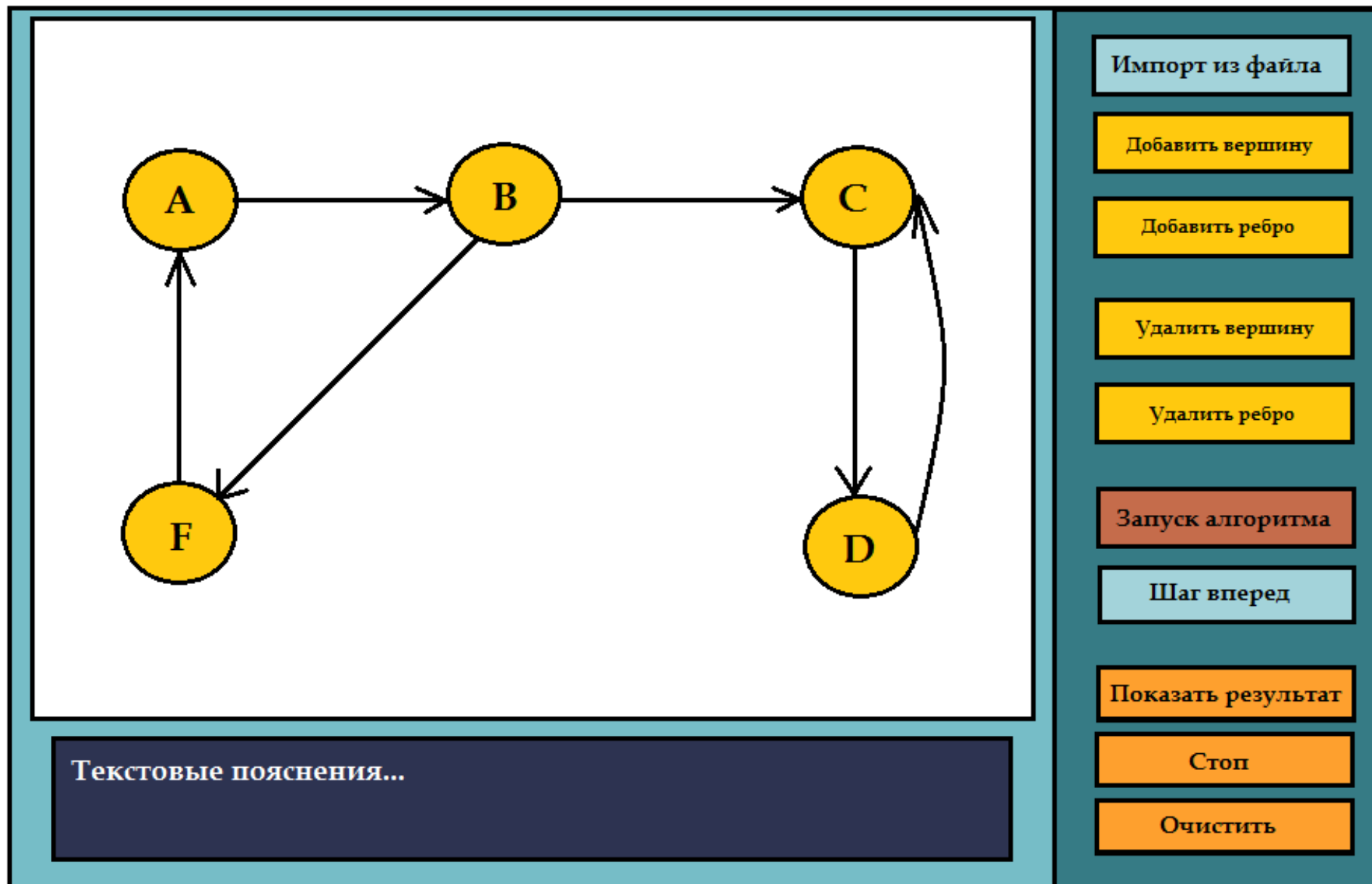


Рис. 1 - Макет меню

1.1.3. Взаимодействие пользователя с программой

Весь пользовательский интерфейс описан на панели управления меню - названия клавиш, являются одноименными методам, реализующим интерфейс. Назначение клавиш см. в таблице 1.

Клавиша	Назначение
“Импорт из файла”	Используется для ввода орграфа через файл.
“Добавить вершину”	Предназначена для добавления новой вершины.
“Добавить ребро”	Необходима для добавления нового ориентированного ребра.
“Удалить вершину”	Используется для удаления вершины.
“Удалить ребро”	Используется для удаления ребра.
“Запуск алгоритма”	После того, как пользователь ввел граф из файла или построил вручную, становится возможным нажатие данной клавиши, которая запускает работу самого алгоритма, сопровождаемую пошаговой визуализацией.
“Шаг вперед”	Переходит к следующему шагу алгоритма.
“Очистить”	Очищение экрана. Предоставляет возможность повторного использования программы без перезапуска, начиная с ввода исходного орграфа.
“Показать результат”	Используется для получения графа, после применения к нему алгоритма Косарайю, без пошаговой визуализации.
“Стоп”	При нажатии на эту клавишу выполнение алгоритма останавливается, введенный граф остается и может быть отредактирован перед повторным запуском алгоритма.

Таблица 1 - Клавиши панели управления

Возможные сценарии взаимодействия пользователя с интерфейсом разрабатываемой программы, представлены на рисунке 2.

Во время работы алгоритма в строке состояний снизу от окна графа будут поясняться действия алгоритма, производимые на соответствующем шаге, например: “Найдена компонента связности 1. Запуск поиска в глубину из вершины 5”. По возможности такие сообщения будут подробными, но, не

слишком нагроможденными и будут формулироваться из расчета, что пользователь ранее не был знаком с алгоритмом Косарайю.

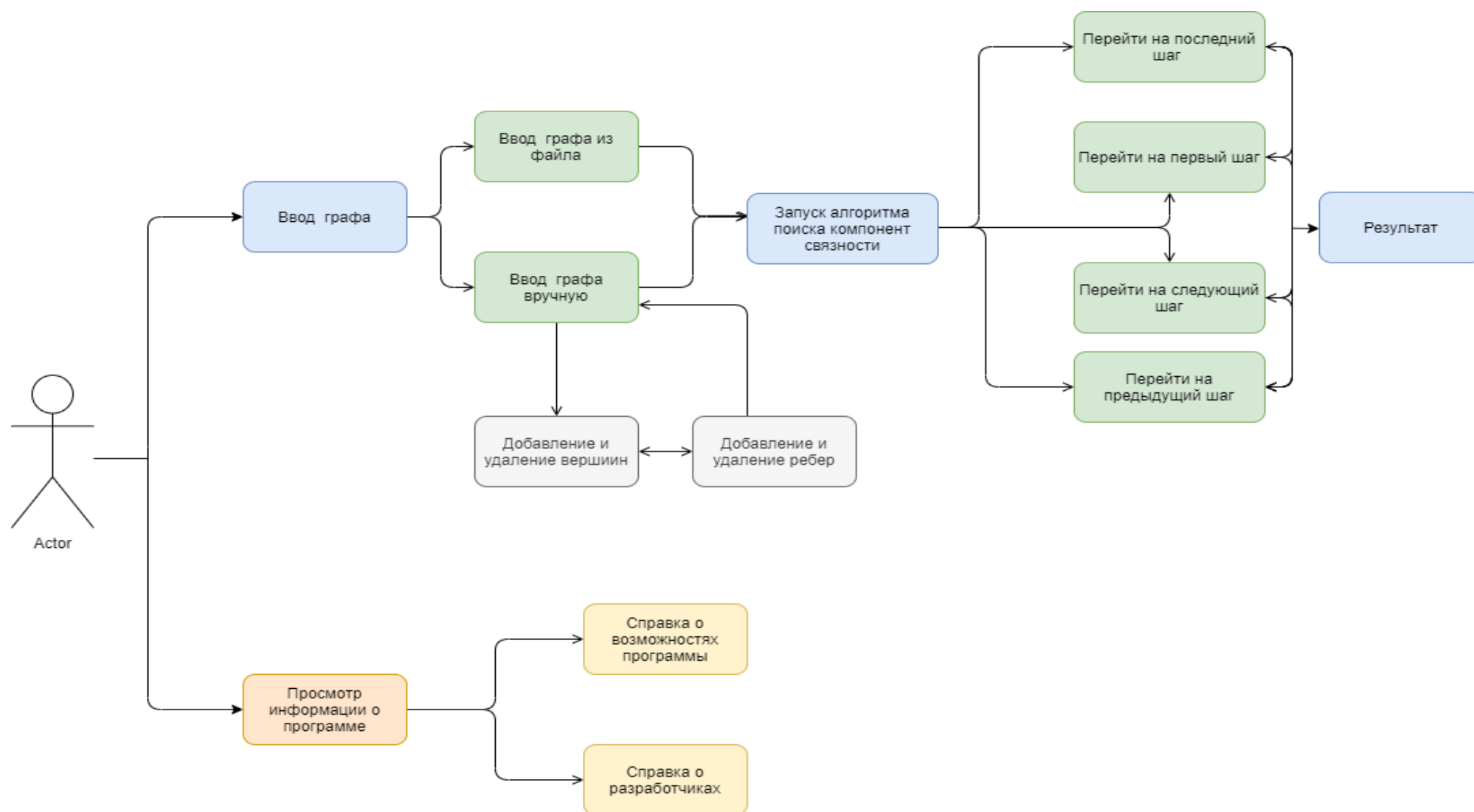


Рис. 2. Диаграмма сценариев

1.2. Уточнения требований в ходе разработки

Будут дополняться в процессе выполнения практической работы.

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки

Таблица 2 - План разработки программы

02.07	03.07 Показ прототипа	04.07	05.07	06.07 Показ 1 версии
<ul style="list-style-type: none"> + Сдать вводное задание. + Написать спецификацию. 	<ul style="list-style-type: none"> + Сдать спецификацию. + Создать окно интерфейса, добавить кнопки управления. + Сделать план тестирования + Сдать прототип программы. 	<ul style="list-style-type: none"> + Реализовать алгоритм + Создать класс вершины + Сделать функцию добавления вершины на поле. + Сделать импорт графа из файла в виде списка ребер и преобразование в список смежности. 	<ul style="list-style-type: none"> + Создать класс ребра. + Реализовать добавление ориентированного ребра для двух вершин. + Добавить возможность нумерации вершин в алфавитном порядке 	<ul style="list-style-type: none"> + Подключить клавиши: “Добавление ребра”, “Добавление вершины”, “Очистить” + Провести тестирования + Исправить недочеты + Добавить вывод текстовых пояснений для пользователя, печатающихся в специальное окно на панели меню.
07.07	08.07 Показ 2 версии	09.07	10.07 Показ финальной версии	11.07 Резервный день
<ul style="list-style-type: none"> + Реализовать отображение графа на экран + Добавить функцию окрашивания вершин, используемую в визуализации + Добавить пометки времени выхода вершин при обходе в глубину. + Связать алгоритм с визуализацией + Подключить клавишу “Запуск алгоритма” 	<ul style="list-style-type: none"> + Реализовать пошаговую визуализацию алгоритма + Подключить клавишу “ Шаг вперед”, “Показать результат” + Провести тестирования + Исправить недочеты 	<ul style="list-style-type: none"> + Подключить клавишу “Стоп” + Добавить удаление ребра. + Добавить удаление вершины. + Подключить клавиши: “Удаление ребра”, “Удаление вершины” + Добавить возможно добавления объектов на экран с помощью нажатий курсора 	<ul style="list-style-type: none"> + Провести финальное тестирование + Исправить недочеты + Доделать отчет 	

2.2. Распределение ролей в бригаде

Распределение ролей в процессе разработки представлено в таблице 3.

Таблица 3 - Распределение ролей

Роль	Ответственный(е)
Лидер	Ольга Аверина
Алгоритмист	Анастасия Максимова
Фронтенд	Андрей Мирсков, Ольга Аверина, (Анастасия Максимова)
Тестировщик	Андрей Мирсков
Документация	Анастасия Максимова

Пояснение:

- ☐ **Лидер** - имеет решающее право голоса, занимается управлением репозитория, помогает фронтенду
- ☐ **Алгоритмист** - реализует основной алгоритм программы (поиска компонент сильной связности), отвечает за его работу и обеспечивает его безопасность от пользователя (обрабатывает ошибки, которые возможно предвидеть заранее)
- ☐ **Фронтенд** - реализация пользовательского интерфейса, отвечает за его наполнение, реализацию, включая пошаговую визуализацию алгоритма
- ☐ **Тестировщик** - организует тестирование функционала, создает наборы тестовых данных, знает, что тестировалось, а что нет и по какой причине
- ☐ **Документация** - создание документации проекта в виде отчета, выбор формата комментариев, используемых в программе, следит за их написанием

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

Для разработки графического интерфейса используется библиотека JavaFX.

3.1. Структуры данных

3.2. Основные методы

4. ТЕСТИРОВАНИЕ

4.1. План тестирования

1. В процессе разработки предстоит протестировать следующее:
 - a. Ручной ввод графа.
 - b. Ввод графа из файла.
 - c. Корректность работы алгоритма
 - d. Пошаговая работа алгоритма и раскраска графа.
 - e. Работа кнопок.
2. Для этого будут использоваться следующие методы тестирования:
 - a. Ручной ввод графа.
 - i. Добавление вершин, ребер
 - ii. Удаление вершин, ребер
 - iii.
 - b. Ввод графа из файла.
 - i. Ввод в файл некорректных данных отслеживание реакции программы. Некорректные данные — это, например, ввод из файла с одним числом в файле, или когда в строке с “ребром” введен один номер вершины.
 - c. Корректность работы алгоритма
 - i. Проверка работы алгоритма на графе из одной вершины, из двух несмежных вершин.
 - ii. Проверка работы алгоритма на пустом графе.
 - iii. Проверка работы на входных данных и проверка корректности результата
 - d. Пошаговая работа алгоритма и раскраска графа.
 - i. Запуск и пошаговый “прогон” алгоритма с отслеживанием шагов и окраски соответствующих вершин.
 - e. Работа кнопок.

- i. Попытка нажать на кнопки “Удалить вершины/ребра” при отсутствии на поле вершин/ребер, соответственно.
 - ii. Попытка во время работы алгоритма нажать кнопки “Импорт из файла”, “Добавить вершины/ребра”, “Удалить вершины/ребра”, “Запуск”.
- 3. Тестирование проводит на следующих этапах разработки:
 - a. Ручной ввод графа: между сдачей 1 и 2 версий программы.
 - b. Ввод графа из файла: перед сдачей 1 версии.
 - c. Корректность работы алгоритма: перед сдачей 1 версии.
 - d. Пошаговая работа алгоритма и раскраска графа: перед сдачей 2 версии.
 - e. Работа кнопок: перед сдачей финальной версии.

4.2. Второй подраздел третьего раздела

ЗАКЛЮЧЕНИЕ

Кратко подвести итоги, проанализировать соответствие поставленной цели и полученного результата.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Ниже представлены примеры библиографического описания, В КАЧЕСТВЕ НАЗВАНИЯ ИСТОЧНИКА в примерах приводится вариант, в котором применяется то или иное библиографическое описание.

1. Иванов И. И. Книга одного-трех авторов. М.: Издательство, 2010. 000 с.
2. Книга четырех авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров, В. В. Васильев. СПб.: Издательство, 2010. 000 с.
3. Книга пяти и более авторов / И. И. Иванов, П. П. Петров, С. С. Сидоров и др.. СПб.: Издательство, 2010. 000 с.
4. Описание книги под редакцией / под ред. И.И. Иванова СПб., Издательство, 2010. 000 с.
5. Иванов И.И. Описание учебного пособия и текста лекций: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.
6. Описание методических указаний / сост.: И.И. Иванов, П.П. Петров. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010. 000 с.
7. Иванов И.И. Описание статьи с одним-тремя авторами из журнала // Название журнала. 2010, вып. (№) 00. С. 000–000.
8. Описание статьи с четырьмя и более авторами из журнала / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название журнала. 2010, вып. (№) 00. С. 000–000.
9. Иванов И.И. Описание тезисов доклада с одним-тремя авторами / Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПб, 2010, С. 000–000.
10. Описание тезисов доклада с четырьмя и более авторами / И. И. Иванов, П. П. Петров, С. С. Сидоров и др. // Название конференции: тез. докл. III международной науч.-техн. конф., СПб, 00–00 янв. 2000 г. / СПбГЭТУ «ЛЭТИ», СПб, 2010, С. 000–000.
11. Описание электронного ресурса // Наименование сайта. URL: <http://east-front.narod.ru/memo/latchford.htm> (дата обращения: 00.00.2010).

12. ГОСТ 0.0–00. Описание стандартов. М.: Изд-во стандартов, 2010.
13. Пат. RU 000000000. Описание патентных документов / И. И. Иванов, П. П. Петров, С. С. Сидоров. Оpubл. 00.00.2010. Бюл. № 00.
14. Иванов И.И. Описание авторефератов диссертаций: автореф. дисс. канд. техн. наук / СПбГЭТУ «ЛЭТИ», СПб, 2010.
15. Описание федерального закона: Федер. закон [принят Гос. Думой 00.00.2010] // Собрание законодательств РФ. 2010. № 00. Ст. 00. С. 000–000.
16. Описание федерального постановления: постановление Правительства Рос. Федерации от 00.00.2010 № 00000 // Опубликовавшее издание. 2010. № 0. С. 000–000.
17. Описание указа: указ Президента РФ от 00.00.2010 № 00 // Опубликовавшее издание. 2010. № 0. С. 000–000.

ПРИЛОЖЕНИЕ А
НАЗВАНИЕ ПРИЛОЖЕНИЯ

полный код программы должен быть в приложении, печатать его не надо