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

ОТЧЕТ

по учебной практике

Тема: Визуализация алгоритмов на графах

Студентка гр. 8303 _____ Самойлова А.С.

Студент гр. 8381 _____ Сахаров М.С.

Студент гр. 8381 _____ Гоголев Е.Е.

Руководитель _____ Фирсов М.А.

Санкт-Петербург
2020

ЗАДАНИЕ
НА УЧЕБНУЮ ПРАКТИКУ

Студентка Самойлова А.С. группы 8303

Студент Сахаров М.С. группы 8381

Студент Гоголев Е.Е. группы 8381

Тема практики: визуализация алгоритмов на графах

Задание на практику:

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

Алгоритм: алгоритм Косарайю.

Сроки прохождения практики: 29.06.2020 – 12.07.2020

Дата сдачи отчета: 00.07.2020

Дата защиты отчета: 00.07.2020

Студентка	_____	Самойлова А.С.
Студент	_____	Сахаров М.С.
Студент	_____	Гоголев Е.Е.
Руководитель	_____	Фирсов М.А.

АННОТАЦИЯ

В отчете о практике собрана теоретическая и практическая информация о разработке графического приложения на языке Java. Данное приложение реализует построение графа с его визуальным представлением. Главная задача работы — реализовать визуализацию алгоритма Косарайю поиска компонент сильной связности графа. Отчет состоит из: введения, четырех глав, поделенных на параграфы, заключения, списка использованных источников и приложения с исходным кодом программы.

Во введении поставлена цель и задачи работы. В первой главе описаны требования к программе. Вторая глава содержит план разработки. В третьей главе описаны особенности реализации проекта. В четвертой главе представлены результаты тестирования программы.

В заключении приведены выводы, полученные в ходе работы.

SUMMARY

The practice report contains theoretical and practical information on the development of a graphical application in the Java language. This application implements the construction of a graph with its visual representation. The main task of the work is to implement the visualization of the Kosarayu algorithm for searching for components of a strongly connected graph. The introduction sets the goal and objectives of the work. The first chapter describes the program requirements. The second chapter contains a development plan. The third chapter describes the features of the project. The fourth chapter presents the results of testing the program.

In conclusion, the conclusions obtained in the course of the work are given.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель практики — построение готового приложения, реализующего работу с графическим представлением графа. Данное приложение должно визуализировать ход выполнения алгоритма Косарайю поиска компонент сильной связности. Компонентой сильной связности (strongly connected component) называется такое (максимальное по включению) подмножество вершин графа, что любые две вершины этого подмножества достижимы друг из друга. Важность данного алгоритма обусловлена широкой применимостью его результатов. Например, сильно связанные компоненты могут быть использованы для решения 2-выполнимости задач. Для выполнения заданной цели необходимо выполнить следующие задачи:

- определить архитектуру проекта
- реализовать представление графа, алгоритм
- создать графический интерфейс
- прописать логику визуализации пошагового выполнения алгоритма

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1.1 Требования к вводу-выводу исходных данных

Пользователь должен иметь возможность:

- Сохранить граф в файл .graph
- Загрузить граф из файла .graph

Формат хранения графа .graph (построчно):

- Число (N), означающее количество узлов графа.
- N строк вида $"{имя_узла} {позиция_x} {позиция_y}".
- Число (M), означающее количество дуг графа.
- M строк вида $"{имя_исходящего_узла} {имя_входящего_узла}".

Особенности хранения графа в файле представлены в разделе 3.2.

1.1.2 Требования редактору графов

На рис. 1 представлен эскиз будущего интерфейса программы.

Пользователь должен иметь возможность:

- Очистить граф (удалить все узлы и рёбра): при выборе в меню «Edit» пункта «Clear» будет вызван метод, который будет итеративно перебирать все существующие узлы и удалять все исходящие из него дуги, а затем и его самого.
- Создать узел на холсте двойным щелчком мыши: в функционале приложения будет предусмотрена возможность добавлять вершину к уже существующему и загруженному в приложение графу. Новая вершина будет определяться «именем» и координатами, определяющимися по щелчку мыши на холсте.
- Создать дугу между двумя узлами: в функционале приложения будет предусмотрена возможность добавлять дугу к уже существующему и загруженному в приложение графу. Новая дуга будет определяться вершиной-источником и вершиной-стоком выбранными пользователем.

- Удалить выделенный узел: после выделения узла и нажатия клавиши Delete будет вызван метод, удаляющий все связанные с этой вершиной дуги, а затем и саму вершину
- Удалить выделенную дугу: после выделения дуги и нажатия клавиши Delete будет вызван метод, удаляющий данную дугу из всех контейнеров, хранящих дуги в графе.
- Переименовать узел: после выделения узла и нажатия клавиши “R” будет выведено окно для ввода нового «имени» узла графа и вызвана функция изменяющее существующие название на введённое пользователем

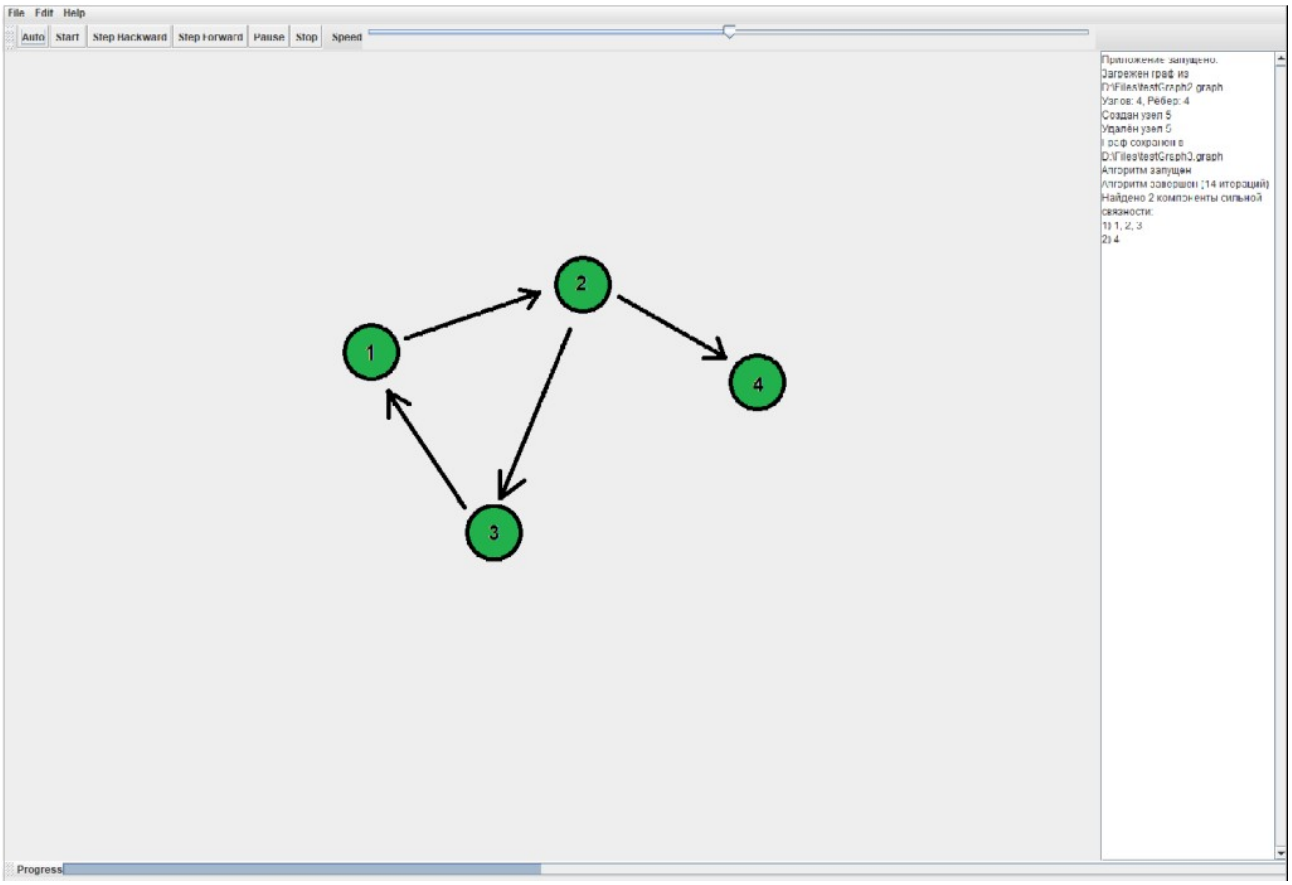

Рисунок 1 — Эскиз интерфейса

1.1.3 Требования к визуализации

На центральной панели программы будут изображены узлы графа в виде окружностей с «именами» внутри согласно координатам, заданным

пользователям, и соединённые стрелочками, которые указывают направления рёбер.

Необходимо реализовать следующие функции:

- Запуск алгоритма
- Запуск алгоритма пошагово
- Выполнить один шаг вперёд
- Выполнить один шаг назад
- Пауза выполнения алгоритма
- Остановка и сброс выполнения алгоритма
- Настройка скорости визуализации

Визуализация алгоритма будет заключаться в выводе шагов обхода графа (путём окрашивания вершин в разные цвета) и изменения направления рёбер графа. Найденные компоненты сильной связности будут окрашиваться в различные цвета. Пользователь сможет останавливать визуализацию алгоритма, вызывать и просматривать предыдущий и следующий шаги алгоритма, отслеживать изменения и пояснения к ним в текстовой форме, выводимые в правой части окна приложения (сообщения о загрузке и сохранении графа, количество вершин и ребер в нем, текущее состояние алгоритма (запущен, остановлен, на паузе, а также текущее количество пройденных, найденных, добавленных в список вершин, количество итераций и сообщение об инвертировании дуг при переходе между этапами алгоритма). Будет указан текущий этап алгоритма, так как алгоритм состоит из двух отличных этапов), какие компоненты были найдены и какие вершины в них входят). Также будет реализована возможность перемещать вершины зажатой левой кнопкой мыши, что позволит пользователю располагать вершины в удобном порядке.

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки

1. К 6 июля — прототип (реализован графический интерфейс, классы, представляющие граф, базовая отрисовка элементов (отображается сам граф)). Написан совместно составленный план тестирования готового проекта.
2. К 8 июля — первая версия проекта. Реализован алгоритм, частичная реализация отображения выполнения алгоритма (отображение шага выполнения вперед/назад, запуск/пауза). Улучшение интерфейса (добавление горячих клавиш). Проведено частичное тестирование программы, исправлены замечания и баги.
3. К 10 июля — итоговая версия проекта. Полная реализация логики визуализации (выбор между пошаговым и автоматическим ходом алгоритма, настройка скорости анимации), рефакторинг исходного кода, проведено тестирование.

2.2. Распределение ролей в бригаде

- Самойлова Анна (8303) — логика алгоритма
- Гоголев Евгений (8381) — логика визуализации
- Сахаров Виктор (8381) — графический интерфейс

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

3.1. Архитектура проета

UML-диаграмма проекта представлена ниже на рис. 2

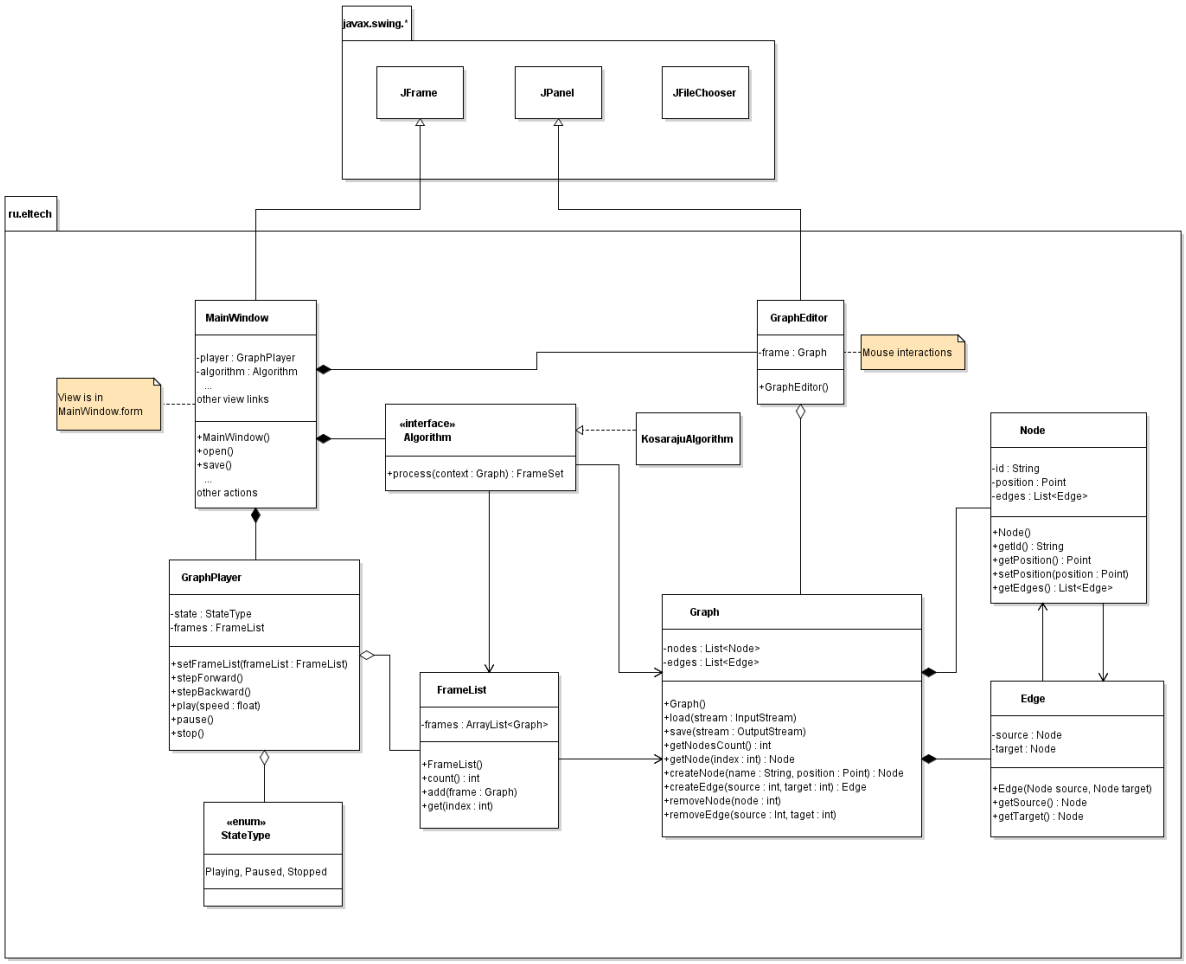

Рисунок 2 — UML-диаграмма проекта

3.2. Особенности хранения графа

Хранение графа выполняется в текстовом файле в кодировке UTF8.

Стандартная сериализация java не используется в пользу читабельности файла без использования программы, а также переносимости данных в программы, написанные на других языках программирования.

При загрузке (Graph.load) считывается весь файл целиком и создаётся объект класса Graph, при сохранении (Graph.save) объект класса Graph

преобразуется в текстовое представление и записывается в файл с расширением .graph. Если файл уже существует, то происходит перезапись с удалением старых данных.

3.3 Отображение графа

Отображение графа в окне приложения будет осуществляться с использованием стандартных средств библиотеки swing для рисования.

4. ТЕСТИРОВАНИЕ

4.1. Первый подраздел третьего раздела

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 00000000. Описание патентных документов / И. И. Иванов, П. П. Петров, С. С. Сидоров. Опубл. 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