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

ОТЧЕТ

по учебной практике

Тема: Визуализатор алгоритма(ов)

Студент гр. 9304 _____ Силкин В.А.

Студент гр. 9304 _____ Боблаков Д.С.

Студент гр. 9304 _____ Атаманов С.Д.

Руководитель _____ Фиалковский М.С.

Санкт-Петербург

2021

ЗАДАНИЕ
НА УЧЕБНУЮ ПРАКТИКУ

Студент Силкин В.А. группы 9304

Студент Боблаков Д.С. группы 9304

Студент Атаманов С.Д. группы 9304

Тема практики: Визуализатор алгоритма(ов)

Задание на практику:

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

Алгоритм: жадный алгоритм построения минимального остовного дерева (алгоритм Краскала).

Сроки прохождения практики: 01.07.2020 – 14.07.2021

Дата сдачи отчета: 07.07.2020

Дата защиты отчета: 07.07.2020

Студент _____ Силкин В.А.

Студент _____ Боблаков Д.С.

Студент _____ Атаманов С.Д.

Руководитель _____ Фиалковский М.С.

АННОТАЦИЯ

Выполнение в бригаде визуализатора алгоритма на языке, использующем JVM (java virtual machine). В этом случае основным языком программирования был выбран Kotlin, использующим для визуализации фреймворк TornadoFX. Алгоритм для визуализации – алгоритм построения минимального остовного дерева (алгоритм Краскала). В ходе написания программы необходимо тестировать код, написанный в течении текущего этапа практики, а также составлять документацию к нему.

СОДЕРЖАНИЕ

	Введение	5
1.	Требования к программе	6
1.1.	Исходные требования к программе*	6
	1.1.1. Ввод данных о графе из текстового документа	6
	1.1.2. Реализация графического отображения графа	6
	1.1.3. Визуализация работы алгоритма.	7
	1.1.4. Работа остальных кнопок интерфейса	8
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	9
2.1.	План разработки	9
	2.1.1. План разработки алгоритма	9
	2.1.2. План разработки UI	9
2.2.	Распределение ролей в бригаде	10
3.	Особенности реализации	11
3.1.	Структуры данных	11
3.2.	Основные методы	11
3.3		0
4.	Тестирование	12
4.1	Тестирование графического интерфейса	12
4.2	Тестирование кода алгоритма	12
4.3	…	0
	Заключение	0
	Список использованных источников	0
	Приложение А. Исходный код – только в электронном виде	0

ВВЕДЕНИЕ

Цель практики – за счёт работы в команде, написать визуализатор алгоритма Краскала. Алгоритм Краскала преобразует граф в минимальное остовное дерево, рассматривая минимальные рёбра, которые ещё не были рассмотрены, добавляет их в граф, если они не образуют цикл, и заканчивает свою работу, когда в графе остаётся одна компонента связности. Минимальное остовное дерево может быть использовано для оптимизации работы других алгоритмах на графе, а также для связи всех вершин графа минимальным количиством рёбер (например, это может понадобиться для связи городов с помощью дорог).

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1. Исходные Требования к программе

Используемая архитектура – MVC.

Интерфейс программы должен содержать следующие опции: выход из программы и сворачивание окна, вывод окна помощи, ввод данных о графе из текстового документа, поле для графического представления графа, кнопки запуска, предыдущего шага, следующего шага, быстрого завершения алгоритма, очистки поля, сохранения графа в текстовый документ.

1.1.1. Ввод данных о графе из текстового документа

Документ для ввода данных всегда выбирает пользователь. Формат документа, который хранит данные о графе - .txt. Если не удается визуализировать граф из документа - вывести ошибку, понятную пользователю (неправильный формат данных в документе, не удалось открыть файл на запись, и т.д.).

Формат данных в текстовом файле - первая строка содержит названия вершин через пробел. Остальные строки записывает рёбра графа в виде - <Значение 1-ой вершины> <Значение 2-ой вершины> <Значение ребра>. Пример записи графа в текстовом документе:

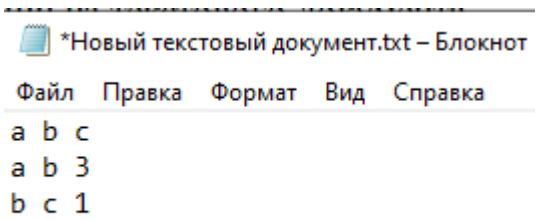

Рисунок 1 - Пример записи графа из 3 вершин и 2 рёбер в .txt документе

Значение ребра графа может быть только целочисленным или числом с плавающей точкой. Если нет - вывести ошибку при старте работы алгоритма.

1.1.2. Реализация графического отображения графа

Граф отображается в виде кругов или прямоугольников с данными в центре фигуры. Стандартный цвет рёбер – синий или чёрный. Над серединой ребра отображается его значение. Граф неориентированный, потому направление рёбер графически отображать не нужно.

В поле для графического представления графа необходимо реализовать добавление вершины графа, её переименование (задание значения), удаление, и те же самые операции с рёбрами графа.

1.1.3. Визуализация работы алгоритма.

Шаг визуализации - действие выбора необработанного ребра (выделить его жёлтым цветом), добавление ребра в минимальное остовное дерево (выделить его зелёным цветом) или отбрасывание ребра без его дальнейшего рассмотрения (выделить его красным цветом).

После старта алгоритма пользователь может перейти на предыдущий (если шаг не первый) или следующий шаг визуализации. При быстром завершении алгоритма, выводится только последний шаг визуализации. Все кнопки шагов активны только до последнего шага алгортима. После достижения последнего шага, алгоритм завершается, все красные рёбра отбрасываются, зелёные рёбра перекрашиваются в основной цвет.

1.1.4. Работа остальных кнопок интерфейса

Очистка поля сбрасывает все текущие настройки графа. Активна только вне работы алгоритма.

Сохранение графа в текстовый документ - если была вызвана до работы алгоритма, сохраняет граф, необработанный алгоритмом, если после - сохраняет обработанный граф. Нельзя вызвать во время работы алгоритма.

1.2. Уточнение требований после 1-го этапа

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки

2.1.1. План разработки алгоритма

1) Реализовать считывание ребер из файла.

2) Реализован собственно сам алгоритм, возвращающий минимальное остовное дерево.

3) К данному функционалу написать юнит-тесты

4) Реализовать логирование.

5) Реализовать просмотр работы алгоритма по шагам.

2.1.2. План разработки UI

1) Разработка окна GUI, расположение указанных кнопок. – До первого этапа

2) Добавление отображения графа, построение, удаление графа внутри программы

3) Добавление сохранения и загрузки графа из файла

4) Возможность вызвать алгоритм из GUI – До второго этапа

5) Реализация кнопок пошагового выполнения алгоритма

6) Отладка алгоритма на тестовых данных

Окно UI выглядит следующим образом:

Рисунок 2 – Прототип UI

Поле графа находится в центре, остальные кнопки находятся справа сверху и слева сверху.

2.2. Распределение ролей в бригаде

Роли в бригаде распределены следующим образом:

Боблаков Д.С. – ответственный за реализацию алгоритма на языке программирования kotlin (алгоритмист).

Атаманов С.Д. – реализация интерфейса на kotlin с помощью фреймворка TornadoFX (фронтенд).

Силкин В.А. – лидер, ответственный за документацию, фронтенд, тестировщик.

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

3.1. Структуры данных

Основная структура хранит в себе граф в виде HashMap, и называется graphHashMap. Вид структуры – HashMap<char, HashMap<char, double>>, где char внутри основной структуры – вершина, из которой исходят рёбра, а HashMap хранит в себе все рёбра, которые исходят из этой вершины в виде вершины, в которую входит ребро, и значение ребра.

3.2. Основные методы

1) readGraph – метод, считывающий ввод графа, и записывающий его в структуру graphHashMap.

2) findMinSpanningTree – метод, реализующий логику алгоритма Краскала, и записывающий его результат в структуру graphHashMap.

3) switchStep – метод, который выводит промежуточную версию графа для отображения в UI

4. ТЕСТИРОВАНИЕ

4.1. Тестирование графического интерфейса

Графический интерфейс тестируется вручную – проверка работы всех кнопок, проверка исключений, описанных в пункте 1, проверка корректности отображения графа, а также инструментов его редактирования.

4.2. Тестирование кода алгоритма

Тестирование будет осуществляться с помощью Unit-тестов для каждой функции. Для readGraph будут применяться тесты следующего вида: некорректный/пустой вес рёбер, подача на вход несвязного графа, обычный граф для работы алгоритма. Для тестирования findMinSpanningTree будут применяться тесты вида: обычный граф для работы алгоритма, граф, имеющий более одной компоненты связности, несвязный граф, граф, имеющий рёбра с отрицательным весом. Функция switchStep будет тестироваться вместе с тестированием графического интерфейса.

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