

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

ОТЧЕТ
по учебной практике
Тема: Топологическая сортировка

Студент гр. 0303

Торопыгин А.С.

Студент гр. 0304

Шквиря Е.В.

Руководитель

Фирсов М.А.

Санкт-Петербург

2022

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Торопыгин А.С. группы 0303

Студент Шквиря Е.В. группы 0304

Тема практики: Топологическая сортировка

Задание на практику:

Командная итеративная разработка визуализатора алгоритма на Kotlin с графическим интерфейсом.

Алгоритм: Топологическая сортировка.

Сроки прохождения практики: 29.06.2020 – 12.07.2020

Дата сдачи отчета: 00.07.2020

Дата защиты отчета: 00.07.2020

Студент		Торопыгин А.С.
Студент		Шквиря Е.В.
Руководитель		Фирсов М.А.

АННОТАЦИЯ

Требуется разработать программу, визуализирующую выполнение алгоритма топологической сортировки на произвольном графе. В качестве дополнительного функционала выступает пошаговое выполнение и считывание и сохранение данных в файл. Алгоритм топологической сортировки используется для упорядочивания вершин в графе.

Целью работы является формирование навыков командной работы, изучение новых языков программирования и работы с фреймворками.

СОДЕРЖАНИЕ

	Введение	5
1.	Требования к программе	6
1.1.	Исходные требования к программе*	6
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	7
2.1.	План разработки	7
2.2.	Распределение ролей в бригаде	7
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 практической работы является разработка программы, визуализирующей работу алгоритма топологической сортировки на графе.

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1. Исходные Требования к программе

1.1.1. Требования к визуализации

Приложение должно иметь понятный графический интерфейс. Он должен полностью описывать функционал программы. Должна быть возможность при помощи мыши выбирать место создания и удаления элементов для редактирования графа. Должно быть пространство с выводом текстовой информации о выполнении алгоритма (этапы алгоритма, актуальное состояние, промежуточные шаги).

1.1.2. Требования к вводу исходных данных

Исходные данные поступают в приложение посредством считывания с файла или созданием графа вручную пользователем (при помощи графического интерфейса).

1.1.3. Требования к структуре программы

Явное разделение программы на несколько слоёв: слой данных, слой бизнес-логики, слой отображения.

1.1.4. Требования к языку

Написание программы на языке программирования Kotlin с использованием Kotlin Multiplatform и инструментов Jetpack Compose.

1.1.5. Требования к тестированию

Тестирование программы разделено на ручное и автоматическое. Вручную будут тестироваться элементы графического интерфейса (создание графа пользователем, передвижение по алгоритму), автоматически будет тестироваться часть, скрытая от пользователя (считывание из файла и сохранение в файл данных, создание объектов классов, алгоритм). Автоматическое тестирование будет реализовано при помощи UNIT-тестов.

1.2. Уточнение требований

1.2.1. Требования к визуализации

Для прототипа первой версии программы был составлен эскиз интерфейса (см. рис. 1).

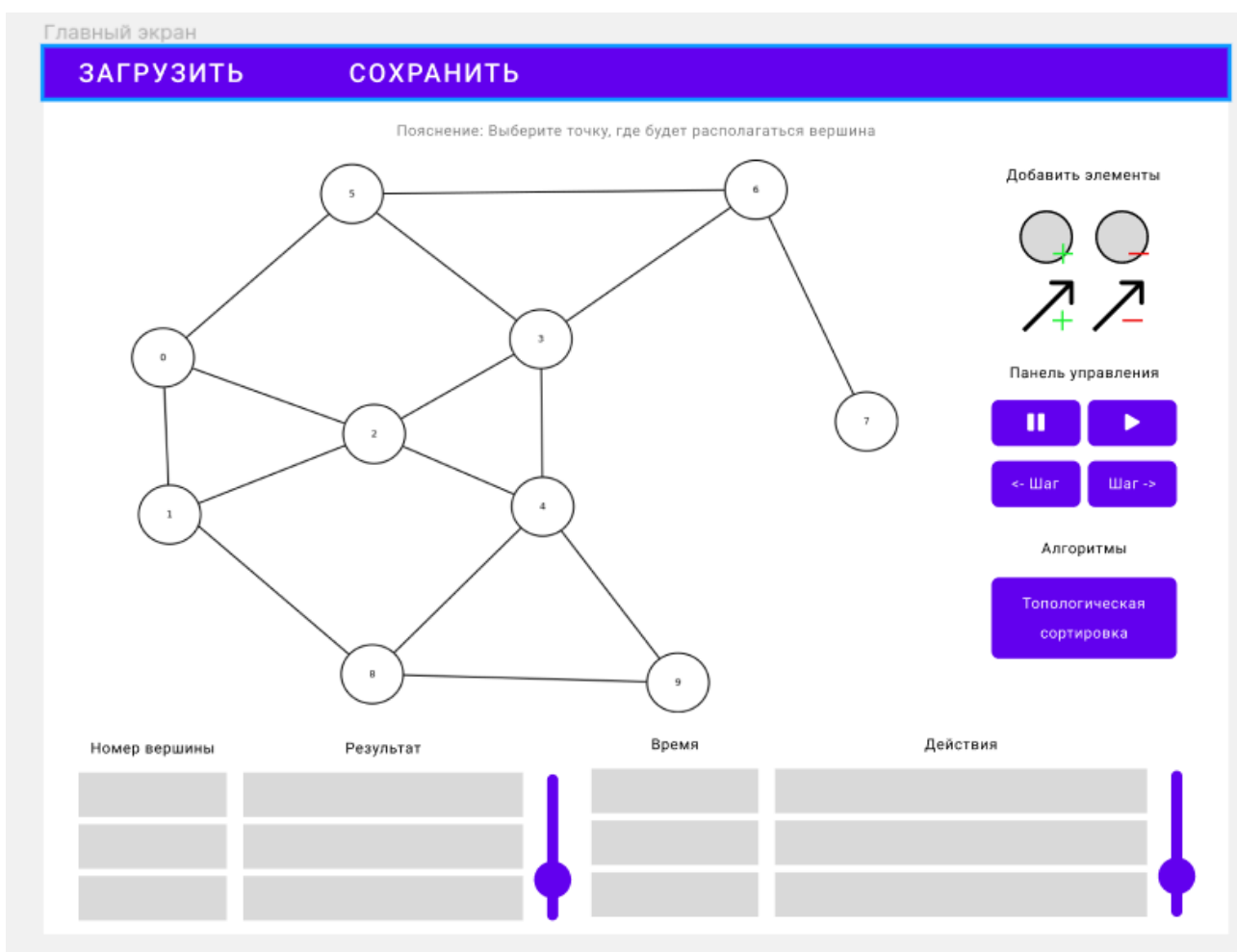


Рис. 1 - Эскиз прототипа программы

На верхней панели инструментов расположены две кнопки - загрузить и сохранить. Первая позволяет открывать для пользователя окно выбора файла, и в случае, если формат файла поддерживается, начинает процедуру считывания информации. В качестве допустимого формата представления данных был выбран JSON. Вторая сохраняет информацию о графе в файл. Способ представления будет описан ниже.

В поле “Пояснение” будет выводиться информация о необходимом действии, которое ожидается от пользователя.

Главная часть экрана приложения будет содержаться холст для работы с графом. На нём будет графическое представление данных в формате отображения вершин и рёбер. Справа от холста располагается панель инструментов для работы с графом - добавление/удаление вершины, добавление/удаление ориентированного ребра. При добавлении вершины на холст сканируется пространство для определения, не закроет ли эта вершина другую вершину. При добавлении ребра происходит выбор двух вершин, после чего происходит соединение. При удалении вершины все связанные с ней рёбра тоже удаляются. При удалении ребра происходит разрыв связи между двумя вершинами. Также на панели инструментов находится кнопка для старта работы алгоритма топологической сортировки.

Кнопка “Топологическая сортировка” запускает алгоритм топологической сортировки, который работает в автоматическом режиме с некоторой задержкой между шагами алгоритма. При нажатии кнопки “Шаг вперёд” или “Шаг назад” управление программой переходит пользователю и он может перемещаться между состояниями алгоритма самостоятельно. При нажатии кнопки “Пауза” программа останавливается на текущем шаге алгоритма и ждёт дальнейших указаний к работе. При нажатии кнопки “Продолжить” алгоритм продолжает свою работу в автоматическом режиме.

Ниже холста располагается таблица с текстовой информацией о ходе работы. В левой части таблицы располагается время, в которое произошло действие. В правой части - описание действия.

1.2.2. Требования к вводу входных данных.

Входные данные хранятся в файле JSON, в котором есть поля:

- name - имя вершины (строка);
- id - номер вершины (десятичное число);
- point - координата центра вершины (два вещественных числа);
- order - номер порядка (десятичное число);

- edges - список названий вершин, в которые идут рёбра из текущей вершины (список строк).

Было принято решение хранить имя и номер вершины. Это даёт несколько преимуществ:

- Это позволит программе работать с графом, в котором некоторые вершины могут иметь одинаковое название;
- В ходе разработки удобнее будет работать с номером вершины, а не с её именем.

1.2.3. Требования к структуре программы

В программе будет присутствовать разбиение архитектуры на несколько слоёв: слой данных, слой бизнес-логики и слой отображения. Явное разбиение будет выглядеть таким образом: Data, Models и UI. Взаимодействие между слоями приведено на рисунке 2.

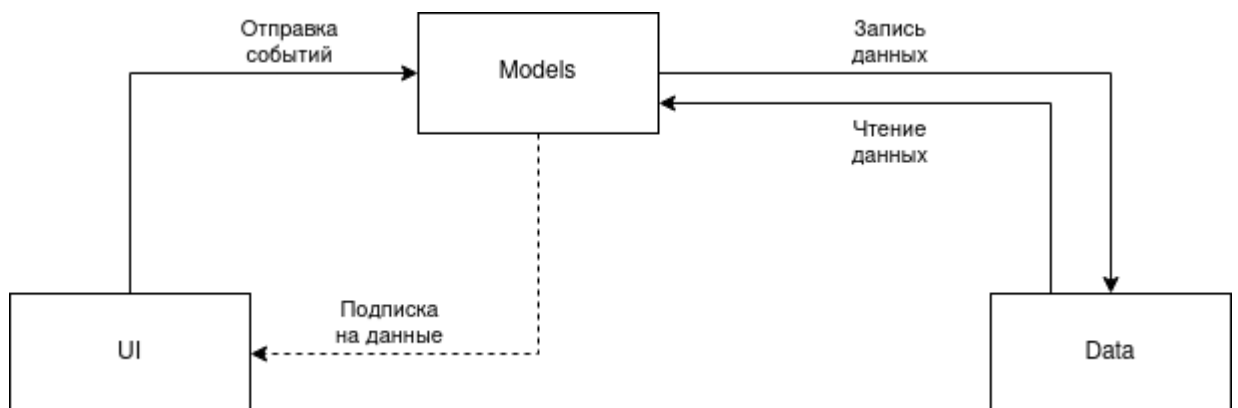


Рис. 2 - Слои архитектуры программы

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки

Таблица 1. Поэтапный план разработки приложения

Дата	Задача	Статус выполнения
04.07.2022	Согласование спецификации	
05.07.2022	Добавление классов для хранения данных. Загрузка информации о графе из файла. Функции для расположения на холсте.	
06.07.2022	Добавление функционала для UI-элементов (добавление элементов графа). Сохранение состояния работы программы в отдельный файл (логи). Сдача прототипа	
07.07.2022	Добавление функционала для UI-элементов (удаление элементов графа). Реализация алгоритма топологической сортировки.	
08.07.2022	Сохранение информации о графе. Сохранение результата топологической сортировки как изображения. Сдача 1-й версии программы	
09.07.2022	Реализация сохранения состояний работы топологической сортировки. Передача информации о шагах в таблицу логирования.	
10.07.2022	Передача информации о шагах в UI. Добавление функционала шагов вперёд/назад. Проверка валидности графа и данных при считывании с файла.	

	Сдача 2-й версии программы	
11.07.2022	Подготовка релизной версии проекта. Внесение корректировок. Тестирование.	
12.07.2022	Сдача финальной версии программы	

2.2. Распределение ролей в бригаде

Шквиря Е.В.:

- Создание структур данных (ViewObject)
- Проектирование и реализация графического интерфейса.
- Составление архитектуры приложения, разделение приложения на логические слои.
- Реализация графической части работы алгоритма топологической сортировки, в том числе касающейся отрисовки и получения информации об этапах работы алгоритма.
- Тестирование своей части работы.

Торопыгин А.С.:

- Создание структур данных (Data)
- Реализация алгоритма топологической сортировки.
- Разбиение алгоритма на логические части для возможности пошагового графического отображения работы алгоритма, в том числе логирование этапов работы алгоритма.
- Реализация сохранения и загрузки данных.
- Тестирование своей части работы.

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

3.1. Структуры данных

3.2. Основные методы

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 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