**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по учебной практике**

Тема: Кратчайшие пути в графе. Алгоритм Дейкстры.

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1304 |  | Чернякова В.А. |
| Студентка гр. 1304 |  | Ярусова Т.В. |
| Студент гр. 1303 |  | Беззубов Д.В. |
| Руководитель |  | Шестопалов Р.П. |

Санкт-Петербург

2023

**ЗАДАНИЕ**

**на учебную практику**

|  |  |  |
| --- | --- | --- |
| Студентка Чернякова В.А. группы 1304 | | |
| Студентка Ярусова Т.В. группы 1304 | | |
| Студент Беззубов Д.В. группы 1303  Тема практики: Кратчайшие пути в графе. Алгоритм Дейкстры. | | |
| Задание на практику:  Командная итеративная разработка визуализатора алгоритма на Kotlin с графическим интерфейсом.  Алгоритм: Дейкстры. | | |
| Сроки прохождения практики: 30.06.2020 – 13.07.2020 | | |
| Дата сдачи отчета: 13.07.2020 | | |
| Дата защиты отчета: 13.07.2020 | | |
|  | | |
| Студентка |  | Чернякова В.А. |
| Студентка |  | Ярусова Т.В. |
| Студент |  | Беззубов Д.В. |
| Руководитель |  | Шестопалов Р.П. |

**Аннотация**

Создание программы с поддержкой графического интерфейса для нахождения кратчайшего пути в графе с помощью алгоритма Дейкстры. Главное преимущество алгоритма Дейкстры – значительно низкая сложность, которая является почти линейной. Реализация алгоритма основана на использовании очереди с приоритетом для перехода в близлежащие вершины. Графический интерфейс дает возможность как пошагово отследить нахождение кратчайшего пути, так и сразу же получить результат. Предусмотрена возможность загрузки графа из файла, а также его создание в рабочей области по щелчку мыши. Приложение поддерживает обработку некорректно введенных данных пользователем - вывод диалогового окна с сообщением об ошибке, что предотвращает возможность некорректной работы. Программа написана на языке программирования Kotlin в среде разработки IntelliJ IDEA c использованием инструментария с графическим интерфейсом swing.

**Summary**

Creating a program with GUI support for finding the shortest path in a graph using Dijkstra's algorithm. The main advantage of Dijkstra's algorithm is a significantly low complexity, which is almost linear. The implementation of the algorithm is based on using a priority queue to go to nearby vertices. The graphical interface makes it possible to track the finding of the shortest path step by step, and immediately get the result. It is possible to load a graph from a file, as well as create it in the workspace at the click of the mouse. The application supports the processing of incorrectly entered data by the user - the output of a dialog box with an error message, which prevents the possibility of incorrect operation. The program is written in the Kotlin programming language in the IntelliJ IDEA development environment using swing GUI toolkit.

**содержание**

[ВВЕДЕНИЕ 5](#_Toc139215259)

[1. ТРЕБОВАНИЯ К ПРОГРАММЕ 6](#_Toc139215260)

[1.1. Исходные требования к программе 6](#_Toc139215261)

[1.1.1. Требования к вводу исходных данных 6](#_Toc139215262)

[1.1.2. Требования к визуализации 6](#_Toc139215264)

[1.2. Шаблоны архитектуры 7](#_Toc139215279)

[1.3. Требования к тестированию 10](#_Toc139215296)

[1.3.1. Тестирование работы алгоритма 10](#_Toc139215297)

[1.3.2. Тестирование визуализации 10](#_Toc139215298)

[2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ 12](#_Toc139215300)

[2.1. План разработки 12](#_Toc139215302)

[2.2. Распределение ролей в бригаде 12](#_Toc139215322)

|  |
| --- |
|  |
|  |
|  |
|  |

# ВВЕДЕНИЕ

Kotlin — это язык программирования, который позволяет писать программы для разных платформ. Сейчас на Kotlin пишут приложения для Android, кроссплатформенные и веб-приложения.

У языка Kotlin есть особенности, которые делают его привлекательным для разработчиков. Защита от ошибок: в Kotlin есть несколько инструментов, которые предотвращают случайные ошибки при написании кода. Поддержка ООП и функционального программирования: язык Kotlin умеет работать как с объектами, так и с функциями — в зависимости от того, что требуется для проекта или конкретной задачи. И другие.

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

## Исходные требования к программе

### Требования к вводу исходных данных

Алгоритм должен получать на вход взвешенный ориентированный граф с положительными весами ребер. Именем вершины может быть следующее: символ, строка, число. Данные могут задаваться, как и в рабочем пространстве программы в соответствии с нажатием соответствующих кнопок, так и в виде файла формата *txt*. В файле данные задаются в следующем порядке через пробел: каждое с новой строки – вершина из которой исходит ребро, вершина куда направлено ребро, вес ребра между этими вершинами.

### Требования к визуализации

Пользовательский интерфейс должен представлять собой диалоговое окно, которое содержит набор кнопок, предназначенных для управления программой пользователем

Диалоговое окно должно состоять из:

* Рабочей области. Она предназначена для построения графа и отображения пошаговых действий в процессе работы алгоритма или его результата.
* Окна логирования. Позволяет отслеживать работу алгоритма на каждом шаге.
* Кнопки «Добавить вершину». Создание вершины графа в рабочей области после клика мышью. При создании пользователь задает вершине имя, соответствующее требованиям к входным данным программы.
* Кнопки «Удалить». Удаление выбранного щелчком мыши элемента графа: ребра или вершины с инцидентными ей ребрами.
* Кнопки «Соединить вершины». Проведение направленного ребра. Задаются параметры начальной и конечной вершины, а также вес ребра.
* Кнопки «Сохранить». Сохранение созданного в рабочей области графа в формате *txt* файла.
* Кнопки «Загрузить». Загрузить в рабочую область граф, представленный формате *txt*.
* Кнопки «Старт». Запуск работы алгоритма Дейкстры на графе. Выбор стартовой вершины.
* Кнопки «Вперед». Отображение следующей итерации алгоритма.
* Кнопки «Назад». Отображение предыдущей итерации алгоритма.
* Кнопки «Результат». Отображение результата работы алгоритма Дейкстры на графе.
* Кнопки «Сброс». Очищение рабочей области.

## Шаблоны архитектуры

Для реализации приложения принято решение использовать архитектуру MVC. Это позволяет разбить реализацию на следующие модули: модель, визуализация и контроллер. Модели соответствует представление графа, контроллер отвечает за взаимодействие пользователя с графическим интерфейсом и реализацию работы алгоритма, визуализация, соответственно, графический интерфейс. Соответствующая схема MVC представлена на рисунке 1.

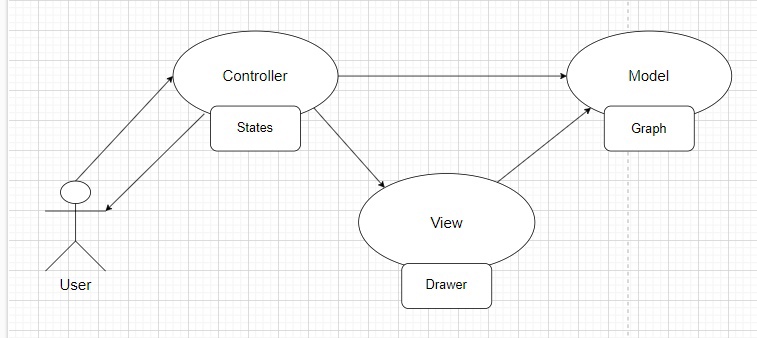


Рисунок 1 – схема архитектуры приложения.

На рисунке 2 представлена UML диаграмма с описанием устройства компонента Model.

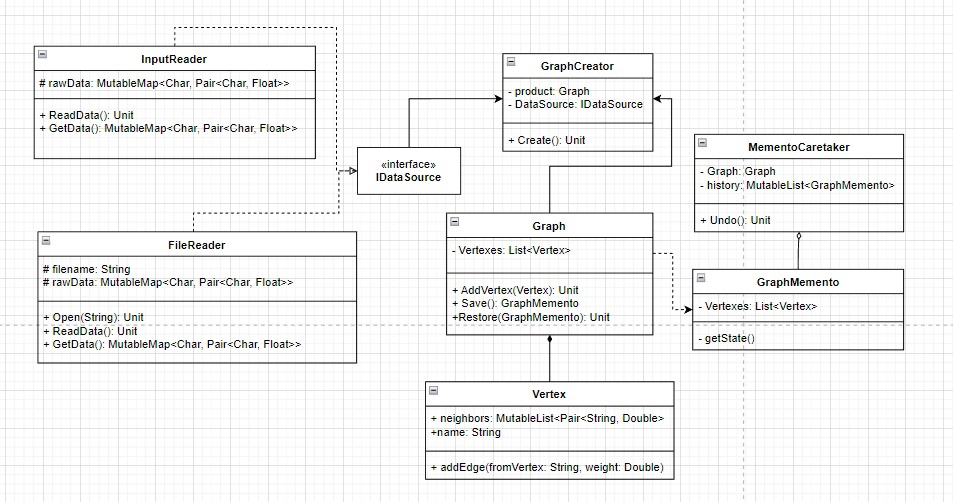


Рисунок 2 – UML диаграмма хранения информации о графе.

На рисунке 3 представлена UML диаграмма модуля View.

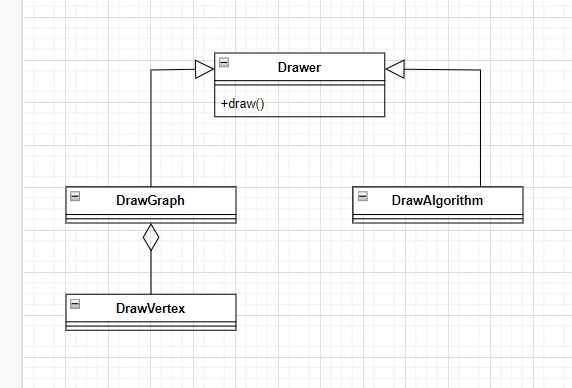


Рисунок 3 – UML диаграмма реализации визуала.

На рисунке 4 представлена UML диаграмма взаимодействия пользователя с UI.

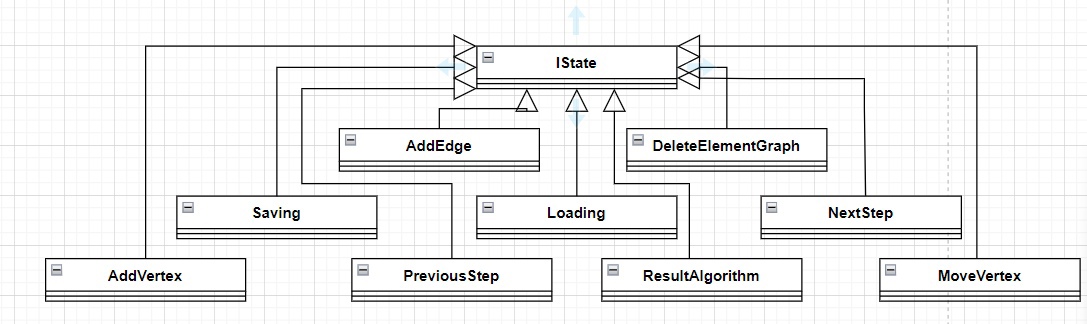


Рисунок 4 – UML диаграмма взаимодействия пользователя с UI.

В работе реализуется алгоритм Дейкстры для поиска кратчайших путей в графе. За основу взят следующий псевдокод:

|  |
| --- |
| func dijkstra(s):  for v∈V  d[v] = ∞  used[v] = false  d[s] = 0  for i∈V  v = null  for j∈V *// найдем вершину с минимальным расстоянием* if !used[j] and (v == null or d[j] < d[v])  v = j  if d[v] == ∞  break  used[v] = true  for e : исходящие из v ребра *// произведем релаксацию по всем ребрам, исходящим из v* if d[v] + e.len < d[e.to]  d[e.to] = d[v] + e.len |

На рисунке 5 представлена диаграмма, отражающая сценарии взаимодействия пользователя с программой.

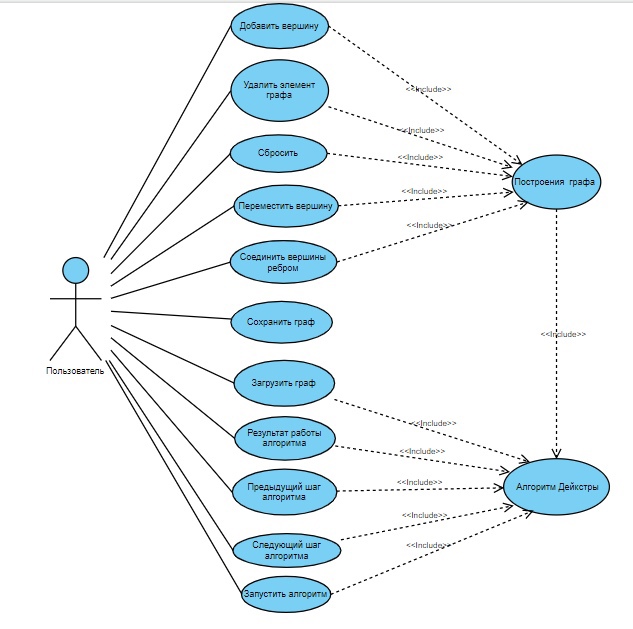


Рисунок 5 – user case диаграмма.

## Требования к тестированию

### Тестирование работы алгоритма

Написание Unit-тестов, позволяющих проверить корректность работы алгоритма как на стандартных, так и на исключительных случаях, таких как: отсутствие пути в графе, граф состоит из одной вершины, граф не является связным.

### Тестирование визуализации

Разработка Unit-тестов, которые позволяют проверить корректность работы конкретных состояний, смены между состояниями, работу с графом, а именно добавление и удаление вершин, ребер.

# ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

## План разработки

Для выполнения поставленных задач был прописан примерный план разработки проекта, представленный на таблице 1.

Таблица 1 - План разработки

|  |  |  |
| --- | --- | --- |
| **№**  **п/п** | **Наименование работ** | **Срок**  **выполнения** |
| **1** | Создание спецификации и плана разработки. Построение UML диаграмм с архитектурными решениями. | 01.07-03.07 |
| **2** | Написание кода алгоритма, разработка обработчиков кнопок интерфейса. | 03.07-05.07 |
| **3** | Написание кода для демонстрации пошаговой работы алгоритма. | 05.07-07.07 |
| **4** | Исправление недочетов проекта. | 07.07-10.07 |
| **5** | Написание тестов к проекту. | 10.07-12.07 |

## Распределение ролей в бригаде

Беззубов Д. – архитектура программы.

Ярусова Т. – реализация алгоритма.

Чернякова В. – визуализация работы алгоритма.

**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 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