Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Омский государственный технический университет»

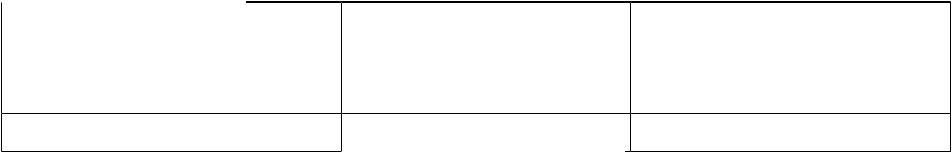
|  |  |
| --- | --- |
| Факультет (институт) | информационных технологий и компьютерных систем |
| Кафедра | «Математические методы и информационные технологии» |

|  |  |
| --- | --- |
|  | **КУРСОВАЯ РАБОТА** |
| **по дисциплине** | «Разработка программных приложений» |
| **на тему** |  |
|  | **Пояснительная записка** |

Шифр проекта (работы)



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Студента** | | | | Колычев И.К. | | | | |
|  |  |  |  |  | **фамилия, имя, отчество полностью** | | | | |
|  | **Курс** 4 | | | | **Группа** |  | ПИ-161 |  |  |
|  |  |  |  | | |  |  | |  |
|  | **Направление (специальность)** | | | | |  | 09.03.03 «Прикладная | | |
|  | информатика в экономике» | | | | |  |  |  |  |
|  |  |  |  |  | **код, наименование** | | |  |  |
|  | **Руководитель** | | | |  |  | доцент | | |
|  |  |  |  |  |  |  | **ученая степень, звание** | | |
|  |  |  |  | Юдин Евгений Борисович | | | | | |
|  |  |  |  |  | **фамилия, инициалы** | | |  |  |
|  | **Выполнил (а)** | | | |  |  |  |  |  |
|  |  |  |  |  |  | **дата, подпись студента (ки)** | | | |
|  | **К защите** | | | |  |  |  |  |  |
|  |  |  |  |  |  | **дата, подпись руководителя** | | | |
|  |  | | | **и** |  |  |  |  |  |
|  | **Выполнение** | | |  |  |  |  |  |
|  | **подготовка к защите** | | | | **Защита КП (КР)** | | | **Итоговый рейтинг** | |
|  | **КП (КР)** | | | |  |  |  |  |  |
|  |  |  |  | |  | | |  | |
|  |  |  | Работа защищена с оценкой | | | | | \_\_\_\_ | |



Омск 2019

Содержание

[Введение 2](#_Toc28301445)

[1 Постановка задачи курсового проектирования 4](#_Toc28301446)

[2 Теоретический анализ 4](#_Toc28301447)

[3 Описание алгоритма 6](#_Toc28301448)

[3.1 Аспекты реализации 6](#_Toc28301449)

[3.2 Диаграмма классов 6](#_Toc28301450)

[4 Описание разработанной системы 7](#_Toc28301451)

[4.1 Описание работы программы 7](#_Toc28301452)

[4.2 Аспекты реализации 8](#_Toc28301453)

[4.3 Сборка проекта с помощью Maven 8](#_Toc28301454)

[4.4 Сохранение проекта в wed сервисе github с помощью Git 9](#_Toc28301455)

[5 Результаты тестирования 11](#_Toc28301456)

[5.1 Результаты тестов 11](#_Toc28301457)

[5.2 Перечень достоинств 14](#_Toc28301458)

[5.3 Перечень недостатков 14](#_Toc28301459)

[Заключение 15](#_Toc28301460)

[Библиографический список 16](#_Toc28301461)

# Введение

Данная курсовая работа по дисциплине «Разработка программных приложений» посвящена приобретению навыков программирования на JAVA.

Целью данной курсовой работы является разработка графического приложения реализующее генерацию, визуализацию и сохранение графов Барабаши-Альберт.

Пояснительная записка включает в себя:

* постановку задачи курсового проекта,
* теоретический анализ,
* описание алгоритма,
* описание разработанной системы,
* результаты тестирования программы,
* заключение,
* библиографический список.

# Постановка задачи курсового проектирования

Изучить возможности библиотеки Jung и, используя полученные знания, разработать приложение для создания графа Барабаши-Альберт.

# Теоретический анализ

JUNG - Java Universal Network / Graph Framework - это библиотека программного обеспечения, которая предоставляет общий и расширяемый язык для моделирования, анализа и визуализации данных, которые могут быть представлены в виде графика или сети. Он написан на Java, что позволяет приложениям на основе JUNG использовать обширные встроенные возможности Java API, а также других существующих сторонних библиотек Java.

Архитектура JUNG разработана для поддержки различных представлений объектов и их отношений, таких как ориентированные и ненаправленные графы, мультимодальные графы, графы с параллельными ребрами и гиперграфы. Он предоставляет механизм для аннотирования графиков, объектов и отношений с метаданными. Это облегчает создание аналитических инструментов для сложных наборов данных, которые могут исследовать отношения между объектами, а также метаданные, связанные с каждым объектом и отношением.

Текущее распределение JUNG включает в себя реализации ряда алгоритмов из теории графов, интеллектуального анализа данных и анализа социальных сетей, таких как процедуры кластеризации, декомпозиции, оптимизации, генерации случайных графов, статистического анализа и вычисления сетевых расстояний, потоков и показатели важности (центральность, PageRank, HITS и т. д.).

JUNG также предоставляет среду визуализации, которая позволяет легко создавать инструменты для интерактивного исследования сетевых данных. Пользователи могут использовать один из предоставленных алгоритмов компоновки или использовать платформу для создания своих собственных макетов. Кроме того, предусмотрены механизмы фильтрации, которые позволяют пользователям сосредоточить свое внимание или свои алгоритмы на определенных частях графика [1].

Maven - это инструмент для сборки Java проекта: компиляции, создания jar, создания дистрибутива программы, генерации документации. Простые проекты можно собрать в командной строке. Если собирать большие проекты с командной строки, то команда для сборки будет очень длинной, поэтому её иногда записывают в bat/sh скрипт. Но такие скрипты зависят от платформы. Для того чтобы избавиться от этой зависимости и упростить написание скрипта используют инструменты для сборки проекта.

Git-директория — это то место, где Git хранит метаданные и базу объектов вашего проекта. Это самая важная часть Git, и это та часть, которая копируется при клонировании репозитория с другого компьютера. Рабочая директория является снимком версии проекта. Файлы распаковываются из сжатой базы данных в Git-директории и располагаются на диске, для того чтобы их можно было изменять и использовать.

Модель Барабаши-Альберт (БА) — алгоритм генерации случайных без масштабных сетей с использованием принципа предпочтительного присоединения. Без масштабные сети широко распространены в природных сетях (пищевые цепочки) и сетях, созданных человеком (Интернет, всемирная паутина, сети цитирования, некоторые социальные сети). Многие исследуемые сети попадают под класс без масштабных сетей.

# Описание алгоритма

## Аспекты реализации

Для реализации данной программы было создано 2 класса. Рассмотрим каждый из поподробнее:

* Класс MainOld – главный класс. Отвечает за визуализацию приложения;
* Класс BAGenerator – создание графа;

## Диаграмма классов

Диаграмма классов созданного приложения приведена на рисунке 1.

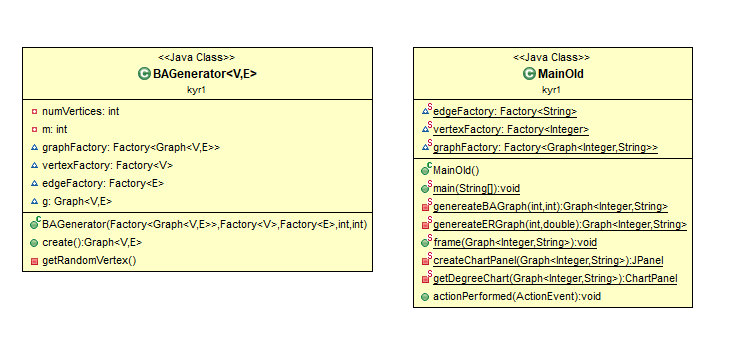


Рисунок 1 – Диаграмма классов

Элемент фона

Текстовая строчка

Маленький бант

Новогодняя шапка

Корона

Колпак

Текстовая строчка №2

Розовый бант

Смайл

Персонаж

Группа классов графических элементов

# Описание разработанной системы

## Описание работы программы

Визуализация программы представлена на рисунке 2.

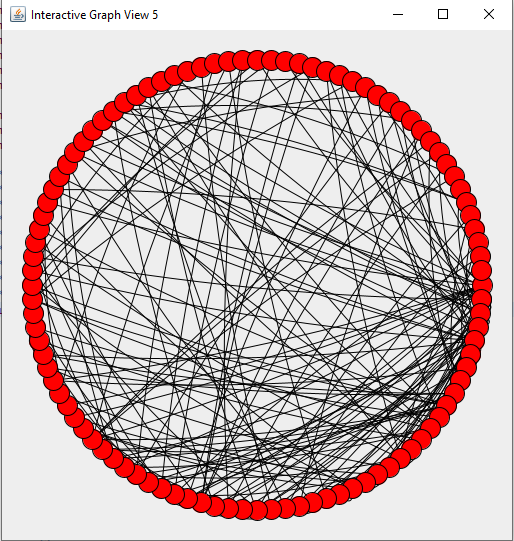


Рисунок 2 – Реализация программы

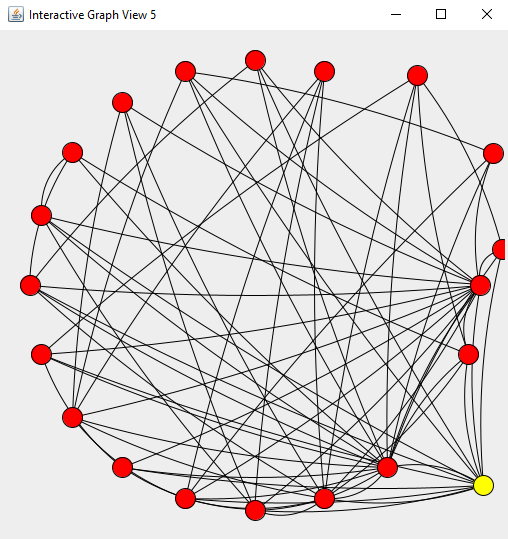
Данный проект способен реализовать различное число вершин и ребер, а так же перемещение вершин для детального рассмотрения графа, пример показан на рисунке 

Рисунок 3 – Подробное рассмотрение построенного графа

## Аспекты реализации

Для работы с графом была использована библиотека Jung, подключаемая дополнительно в Java.

Визуализация графа осуществлялась с помощью VisualizationViewer.

## Сборка проекта с помощью Maven

Для того, чтобы проект можно было открыть в любой среде разработки, создан проект на основе Maven.

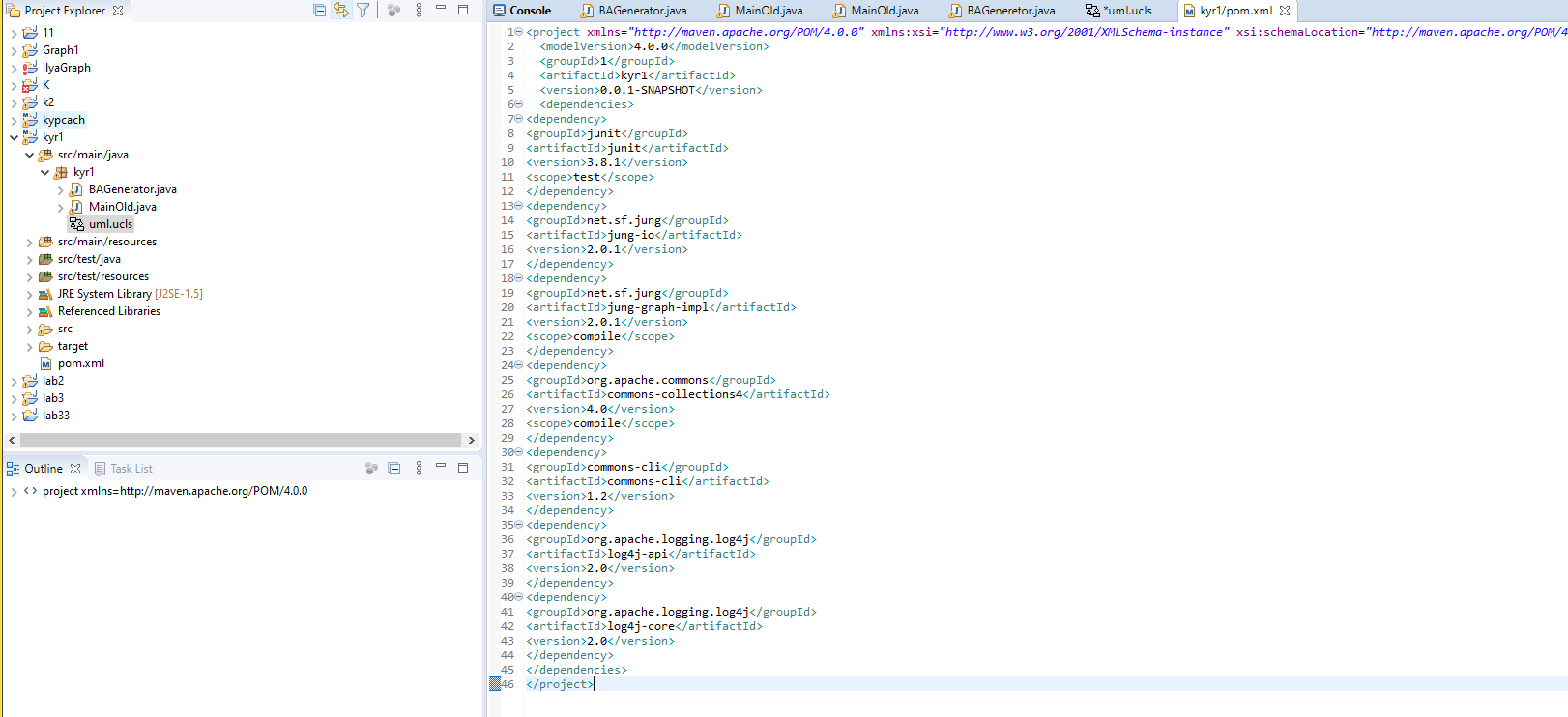


Рисунок 4 – Создание проекта Maven

## Сохранение проекта в wed сервисе github с помощью Git

При дальнейшей разработке проекта, возможно появится необходимость открыть проект на более ранней версии, для реализации подобной возможности проект загружен в wed сервис github.

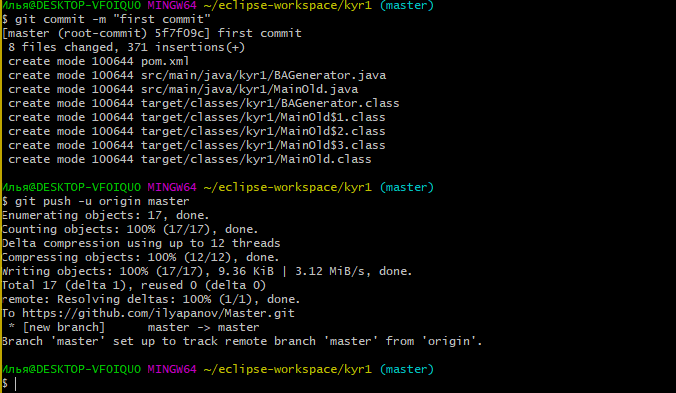


Рисунок 5 – Загрузка проекта в wed сервис

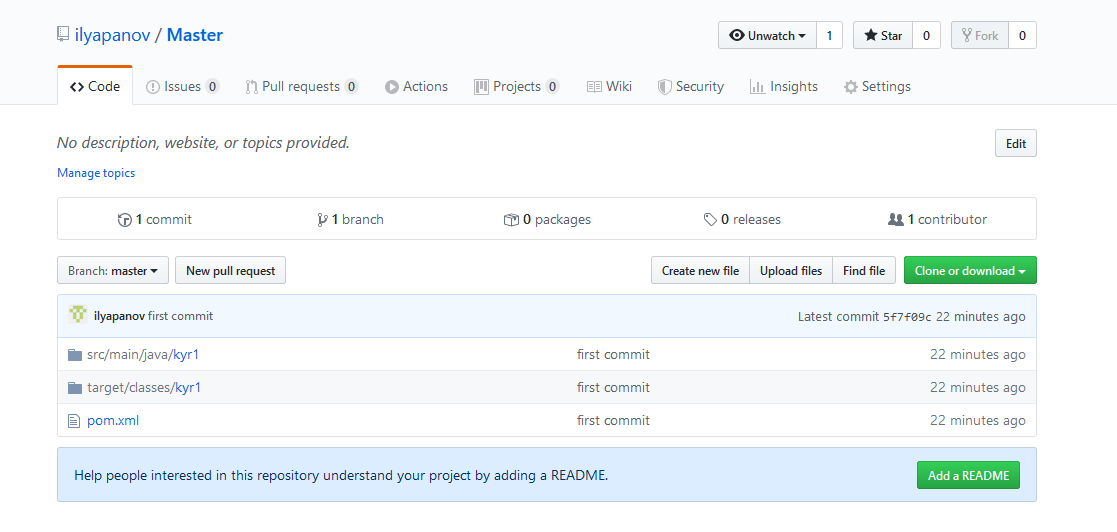


Рисунок 6 – Загрузка проекта в wed сервис

Так же был создан файл ReadMe для того, чтобы будущее разработчики проекта могли узнать все тонкости и особенности его реализации.

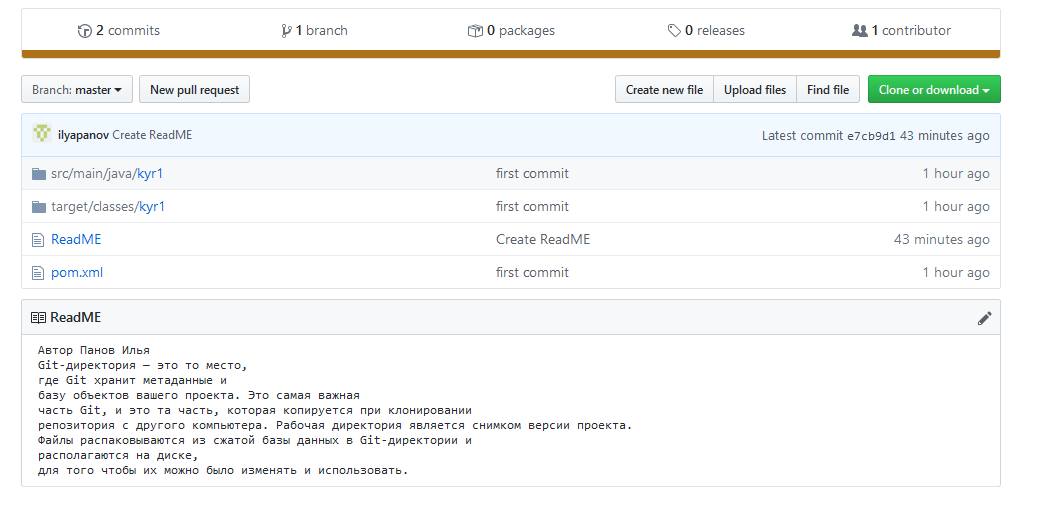


Рисунок 7 – файл ReadMe

# Результаты тестирования

## Результаты тестов

По результатам тестов программа выполняет корректный расчёт графа вне зависимости от количества ребер и вершин. Пример работы программы можно увидеть на рисунках 8, 9 и 10.

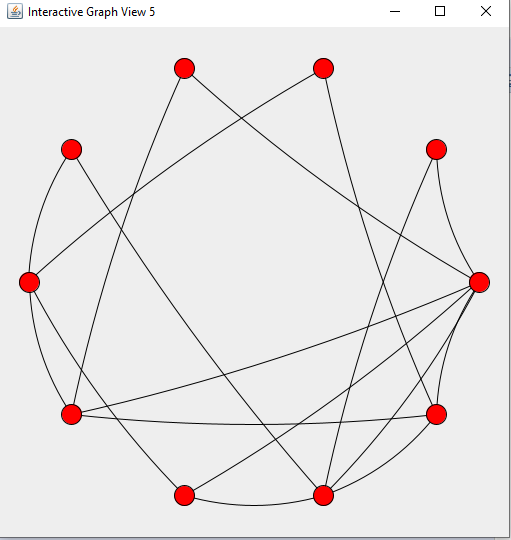


Рисунок 8 – Пример работы программы

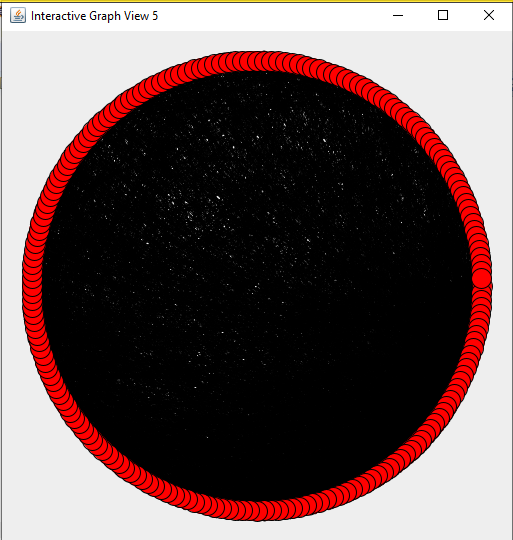


Рисунок 9 – Пример работы программы

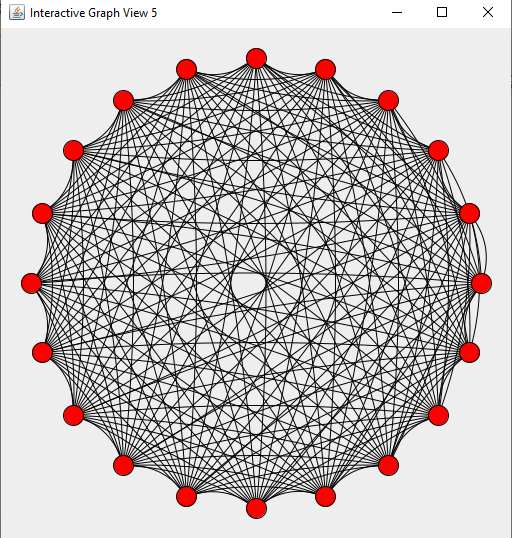


Рисунок 10 – Пример работы программы

## Перечень достоинств

Данное приложение имеет ряд достоинств:

* удобный графический интерфейс;
* простота использования;

## Перечень недостатков

Как и любое другое приложение имеет ряд недостатков:

* Чем массивнее граф, тем больше оперативной памяти необходимо для его отрисовки, при 2 ребрах в граф затравке и 100000 вершин компиляция кода длится порядка 5 секунд последующее увеличение размеров приведет к еще большему времени компиляции кода;

# Заключение

Результатом выполнения курсового проекта является графическое приложение, которое реализует создание графа Барабаши-Альберт.

Приложение может применяться для изучения графов.

# Библиографический список

1. Сайт библиотеки JUNG [Электронный ресурс] – Режим доступа: <http://jung.sourceforge.net/>, свободный. – Загл. с экрана.
2. JUNG 2.0 Tutorial [Электронный ресурс] – Режим доступа: <http://www.grotto-networking.com/JUNG/JUNG2-Tutorial.pdf>, свободный. – Загл. с экрана.