Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Омский государственный технический университет»

Кафедра «Математические методы и информационные технологии в экономике»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОМУ ПРОЕКТУ

по дисциплине «Разработка программных приложений»

**Генерация графа с нелинейным правилом предпочтительного связывания**

Руководитель

доцент кафедры АСОИУ Е.Б. Юдин

Исполнитель

студент группы ПИ-161 М.С. Шарипова

Омск 2019

**РЕФЕРАТ**

Пояснительная записка по курсовому проекту содержит 15 страниц, 16 рисунков.

JAVA ПРИЛОЖЕНИЕ, JUNG,ГЕНЕРАЦИЯ ГРАФА НЕЛИНЕЙНЫМ ПРАВИЛОМ ПРЕДПОЧТИТЕЛЬНОГО СВЯЗЫВАНИЯ, РАСПРЕДЕЛЕНИЕ СТЕПЕНИ СВЯЗНОСТИ ВЕРШИН, ГРАФИЧЕСКОЕ ПРИЛОЖЕНИЕ.

В ходе работы пишется программа на языке JAVA, которая реализовывает выращивание графа с нелинейным правилом предпочтительного связывания.

В результате при помощи написанной программы было получено приложение, которое отвечает заданным требованиям.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc281073629)

[1 Постановка задачи курсового проекта 5](#_Toc281073630)

[2 Теоретический анализ 5](#_Toc281073631)

[3 Описание алгоритма 7](#_Toc281073632)

3.1 Аспекты реализации…………………………………………………………..7

[3.2 Диаграмма классов 7](#_Toc281073633)

[4 Описание разработанной системы 8](#_Toc281073635)

[4.1 Описание работы программы 8](#_Toc281073636)

[4.2 Аспекты реализации 10](#_Toc281073637)

[5 Результаты тестирования 11](#_Toc281073638)

[5.1 Результаты тестов 11](#_Toc281073639)

[5.2 Достоинства приложения 15](#_Toc281073640)

[5.3 Недостатки приложения 15](#_Toc281073641)

[ЗАКЛЮЧЕНИЕ 11](#_Toc281073642)

[Список использованной литературы 12](#_Toc281073644)

**ВВЕДЕНИЕ**

Данный курсовой проект по дисциплине «Разработка программных приложений на языке JAVA» посвящен приобретению навыков программирования на JAVA.

Цель: создание графического приложения с использованием библиотеки Jung, которое генерирует граф с нелинейным правилом предпочтительного связывания.

Предмет исследования: граф с нелинейным правилом предпочтительного связывания.

В ходе работы над курсовым проектом требуется разработать приложение, которое реализует генерацию графа с нелинейным правилом предпочтительного связывания, а также вывести распределение связности вершин.

Пояснительная записка включает в себя постановку задачи курсового проекта, теоретический анализ, диаграмму классов и последовательности действи, описание разработанной системы, результаты тестирования программы, заключение, список использованных источников.

# 

# **1 Постановка задачи курсового проекта**

Реализовать графическое приложение, реализующее генерацию графа с нелинейным правилом предпочтительного связывания с выводом распределения степени связности вершин графа.

# **2 Теоретический анализ**

Экспоненциальный растущий граф – это граф, в котором в каждый момент времени добавляется вершина, устанавливаются *m* ребер между новой вершиной и существующими вершинами графа, вершина для присоединения выбирается равновероятно из существующих вершин графа.

В данном курсовом проекте необходимо реализовать один вид экспоненциального растущего графа предложенный Задорожным В.Н. граф c нелинейным правилом предпочтительного связывания (НППС).

При построении случайных графов с НППС используется произвольная функция предпочтения *f*(*k*) ≥ 0. В соответствии с этой функцией вероятность связывания ребра приращения с вершиной *i* определяется в виде

*pi* = *f*(*ki*)/Σ*j**f*(*kj*). (1.2)

Кроме того, приращения графов с НППС являются *стохастическими*, т.е. каждое приращение представляет собой вершину со случайным числом *x* ребер, имеющим распределение вероятностей {*rk*}, где *rk* = P{*x* = *k*}. В общем случае алгоритм генерации (генератор) графа с НППС задается параметрами *f* и {*rk*}. [1]

Существуют два метода генерации графа с НППС: базовый метод генерации графа с НППС и ускоренный метод генерации графа с НППС.

Выбор вершины для присоединения при генерации графа БА по правилу (1.2) может быть выполнен на основе стандартного подхода разыгрывания дискретной случайной величины (ДСВ). Но этот метод трудоемок в количестве итераций при выборе вершины для присоединения поскольку в исследуемых сетях таких вершин сотни тысяч.

При **ускоренном методе генерации графа с****НППС**разбивается множество вершин на слои (подмножества с одинаковой степенью связности).

В генераторе БА\_СЛОИ (*m*, *N, G*) выбор вершины для присоединения в соответствии с вероятностью (1.2) выполняется в два этапа: вначале разыгрывается номер слоя, из которого будет выбрана вершина для присоединения, а затем вершина этого слоя выбирается равновероятно.

Слой  с наибольшим номером  при этом растет как *K ~ N1/2.*

Причем, в ускоренном алгоритме БА\_СЛОИ, как и в базовом, основная трудоемкость определяется средним числом итераций, которое необходимо выполнить для выбора вершины (т.е. слоя вершин, поскольку из слоя вершина выбирается случайно за один шаг). Это среднее число итераций зависит от числа *n* вершин:





*y* = 0,5∙*N*2

Среднее число

итераций, тыс.

**Рисунок 1.** Эмпирическое (маркеры) и теоретическое (сплошная линия) среднее число итераций выбора вершины при генерации графа БА, *m* = 2



**Рисунок 2.** Схема алгоритма БА\_СЛОИ (*ki* – степень связности *i*-й вершины; *lst* – список нумерованных вершин, выбранных для связывания)

Среднее число итераций при выборе вершины для добавления *N* вершин равно:

.

Базовым алгоритмом граф БА размером 100 тыс. вершин генерируется около трех часов. Разработанный ускоренный алгоритм сокращает это время нескольких минут. Ускорение дает возможность проводить многовариантный анализ, без которого невозможно решение вопросов структурной идентификации сетей и решение задач системного анализа сетей и сетевых процессов в целом.[1]

**3. Описание алгоритма**

**3.1 Аспекты реализации**

Для генерации графа с НППС была разработана программа состоящая из 2-х классов и интерфейсов.

Класс **RunItDegree** является главным классом программы, необходим для запуска приложения, выбора колличества вершин графа затравки и колличества вершин последующего графа с НППС, визуализацию сгенерированного графа с НППС, гистограммы демонстрирующей распределение степени связонности вершин графа, вывода степени каждой вершины в текстовое поле, вывода колличества ребер во всем графе с НППС, а также тестирования программы.

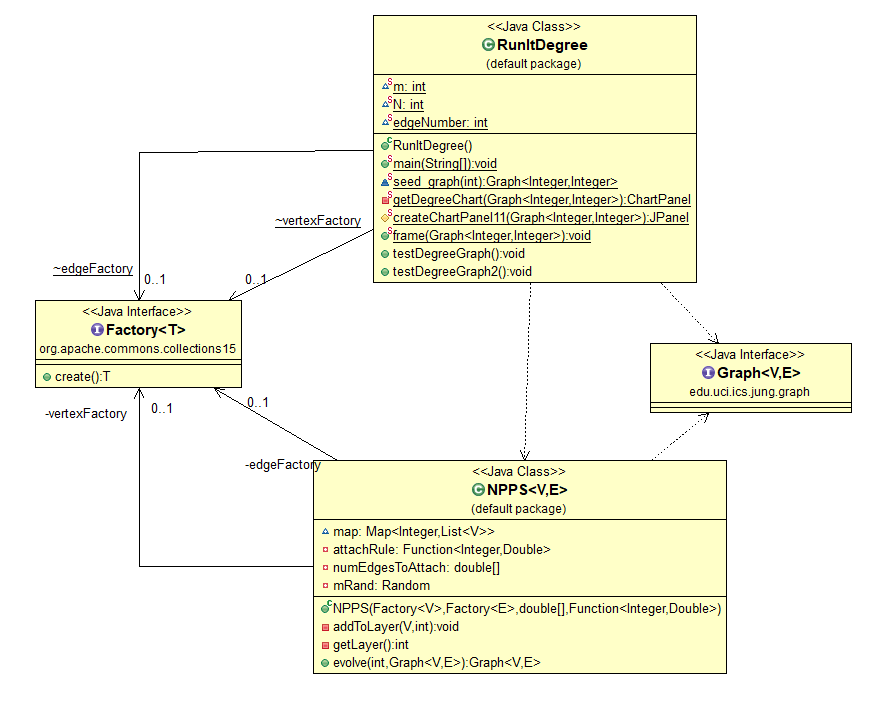
Класс **NPPS** необходим для реализации алгоритма генерации графа с НППС, а именно выбора слоя вершин и вершины из этого слоя для присоединения.

Интерфейс **Factory** создает вершины и ребра графа.

Функции интерфейса **Graph** необходимо для разработки программы генерации графа.

## 3.2Диаграмма классов

Диаграмма классов созданного приложения приведена на рисунке 3.



**Рисунок 3**. Диаграмма классов

**4. Описание разработанной системы**

**4.1 Описание работы программы**

После запуска программы перед пользователем появляется окно, в котором необходимо выполнить следующие шаги:

1) задать количество вершин графа затравки;

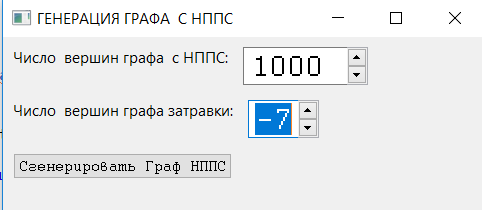
2) задать количество вершин графа с НППС;

3) нажать на кнопку для генерации графа с НППС.

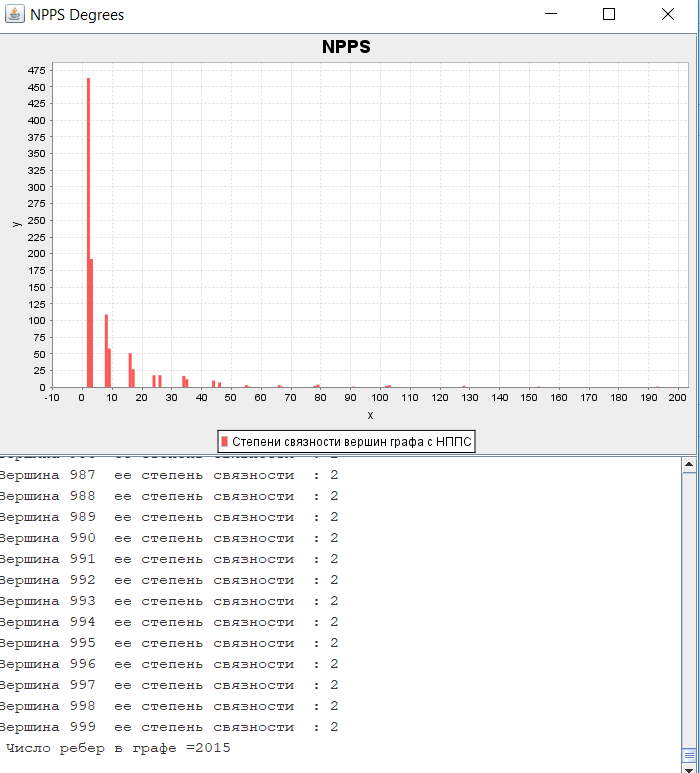
Выполнив эти шаги, открывается два окна, в первом отображается панель с гистограммой распределения степени связности вершин и вывод степени каждой вершины, количества всех ребер в текстовое поле.

Сам граф с НППС отображается в другом окне.

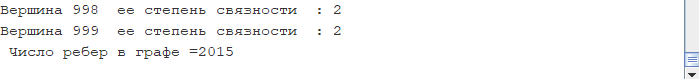
Вышеописанное представлено на рисунках 4,5, 6,7.



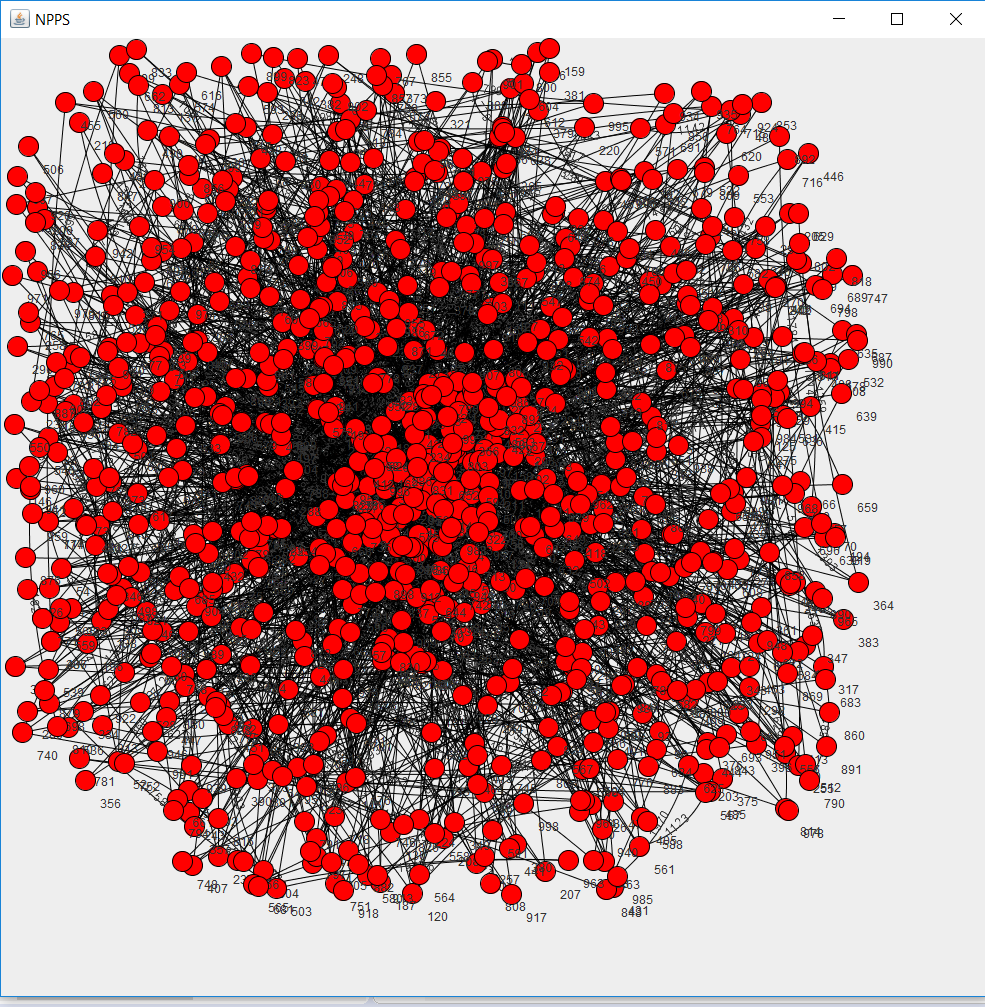
**Рисунок 4**. Окно задания вершин графа затравки и графа с НППС.



**Рисунок 5**. Распределение степени связанности вершин графа с НППС и вывод степени каждой вершины графа



**Рисунок 6**. Вывод количества всех ребер графа с НППС



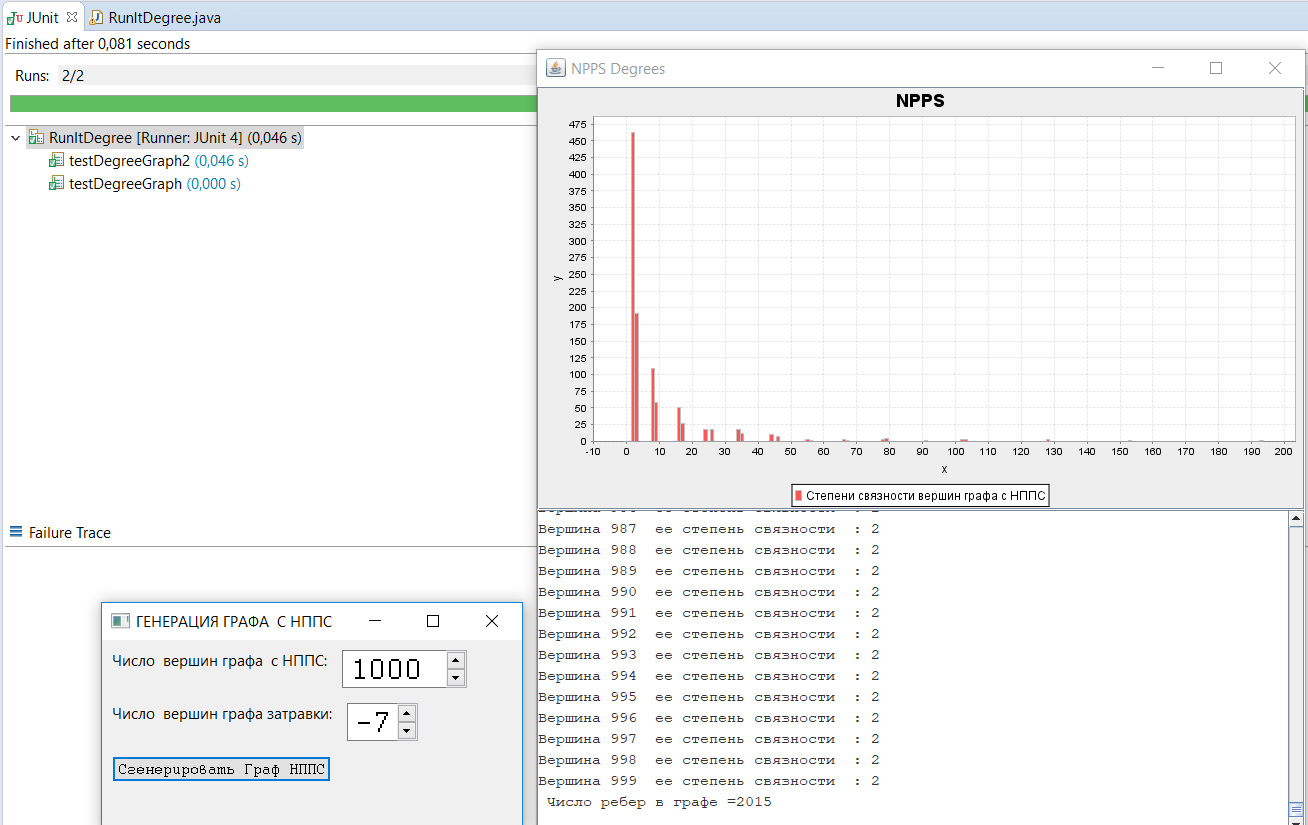
**Рисунок 7**. Сгенерированный граф с НППС

## 4.2 Аспекты реализации

В ходе реализации приложения использовались инструменты следующих библиотек:

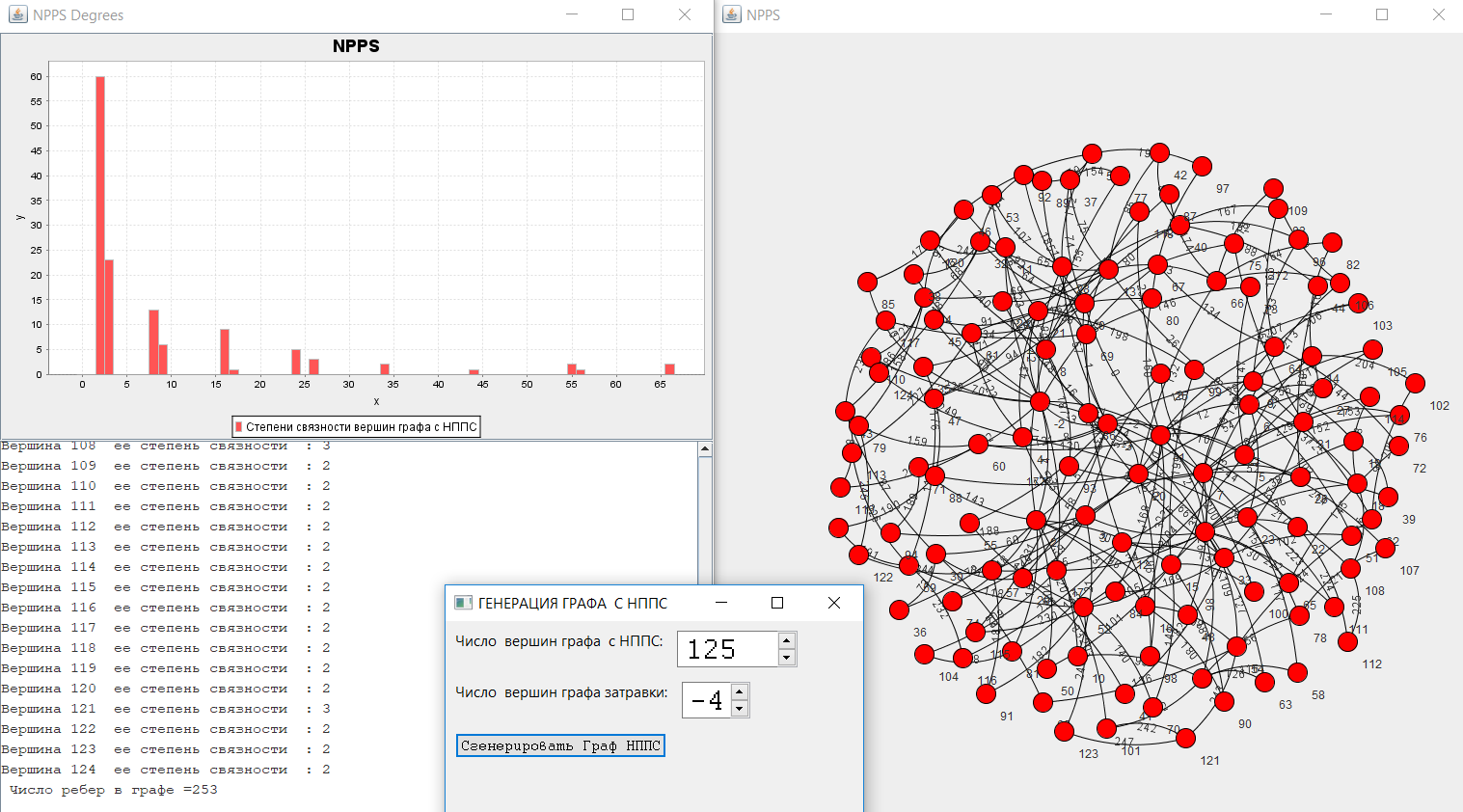
1. Инструменты библиотеки SWT использовались для cсоздания окна задания количества вершин графа и кнопки вызова действия по кнопке для генерации графа;
2. Инструменты библиотеки JFreeChart использовались для отображения панели, демонстрирующей распределения степени связанности вершин и вывода степени каждой вершины в текстовое поле;
3. Инструменты библиотеки Jung использовались для работы с графом;
4. Функции библиотеки JUnit использовались для тестирования программы.
5. **Результаты тестирования**
   1. **Результаты тестов**

При тестировании с использованием Junit, программа так же была выполнена успешно. Результат тестирования виден на рисунке 9. В тесте testDegreeGraph проверяется количество ребер в графе, а во втором тесте проверяется действительно ли программа сгенерировала граф с заданным количеством вершин.

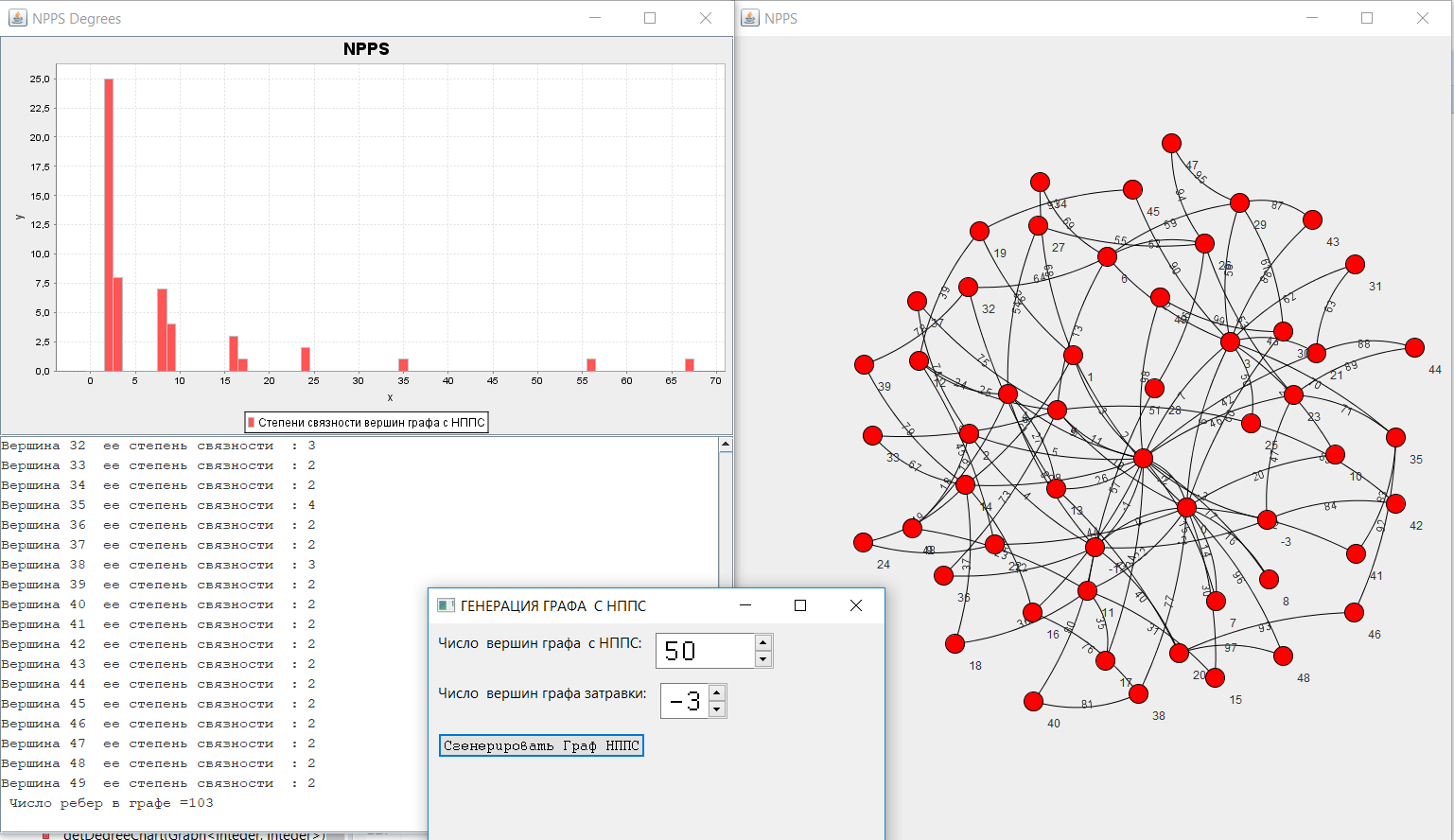


**Рисунок 9.** Тестирование JUnit

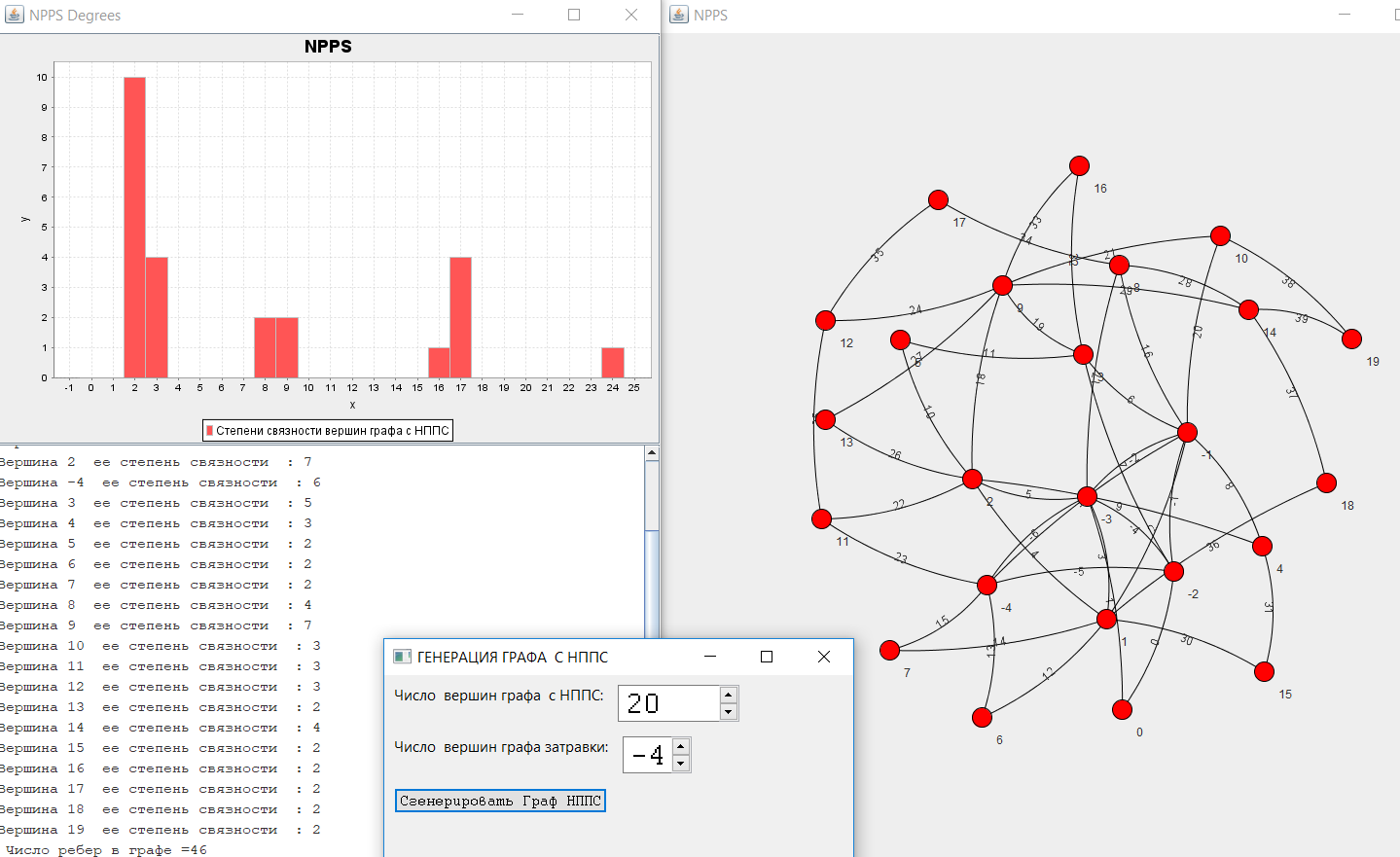
Другие примеры работы программы показаны на рисунках 10,11,12,13,14,15,16.



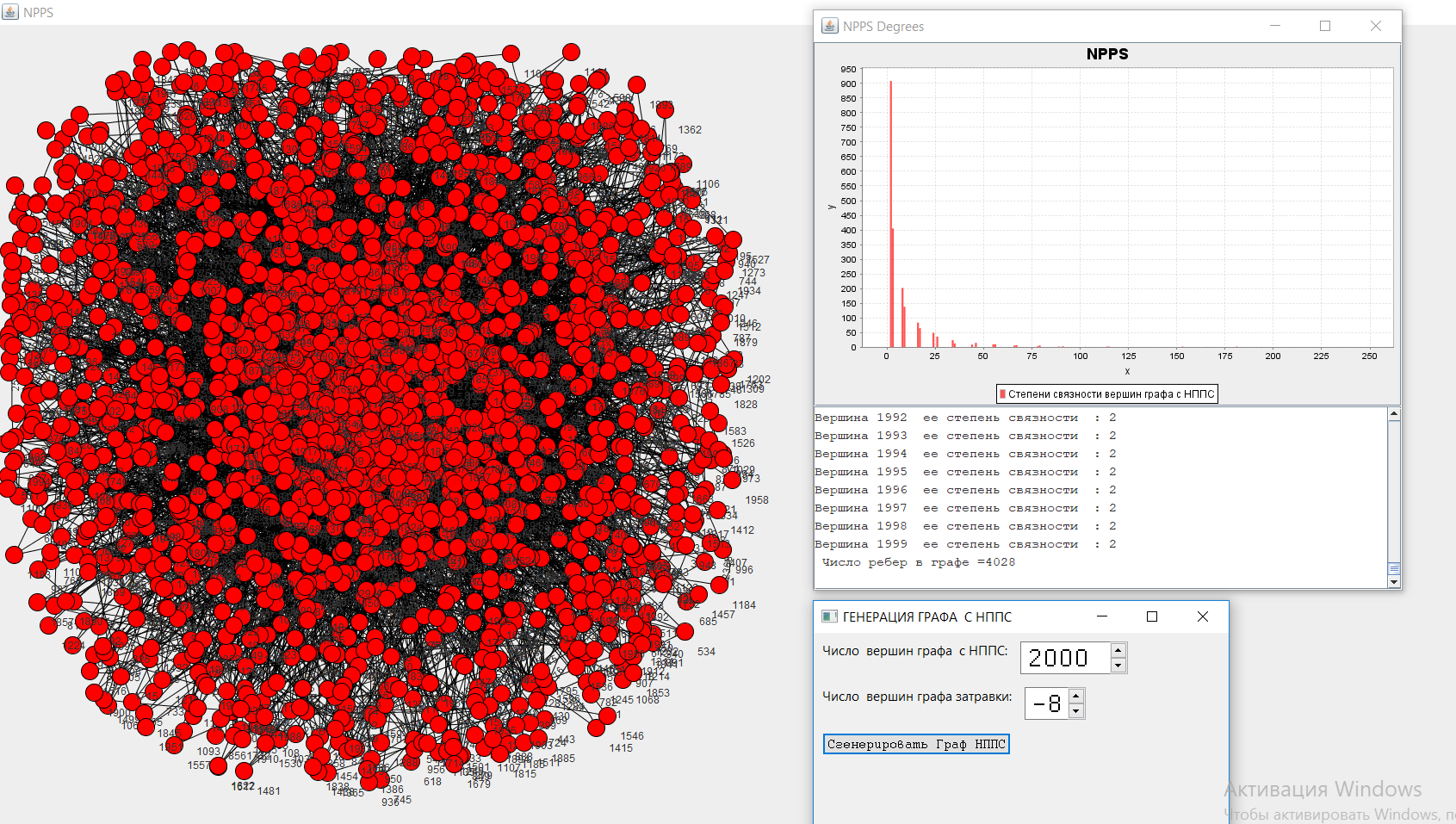
**Рисунок 10.**  Пример\_1 работы программы



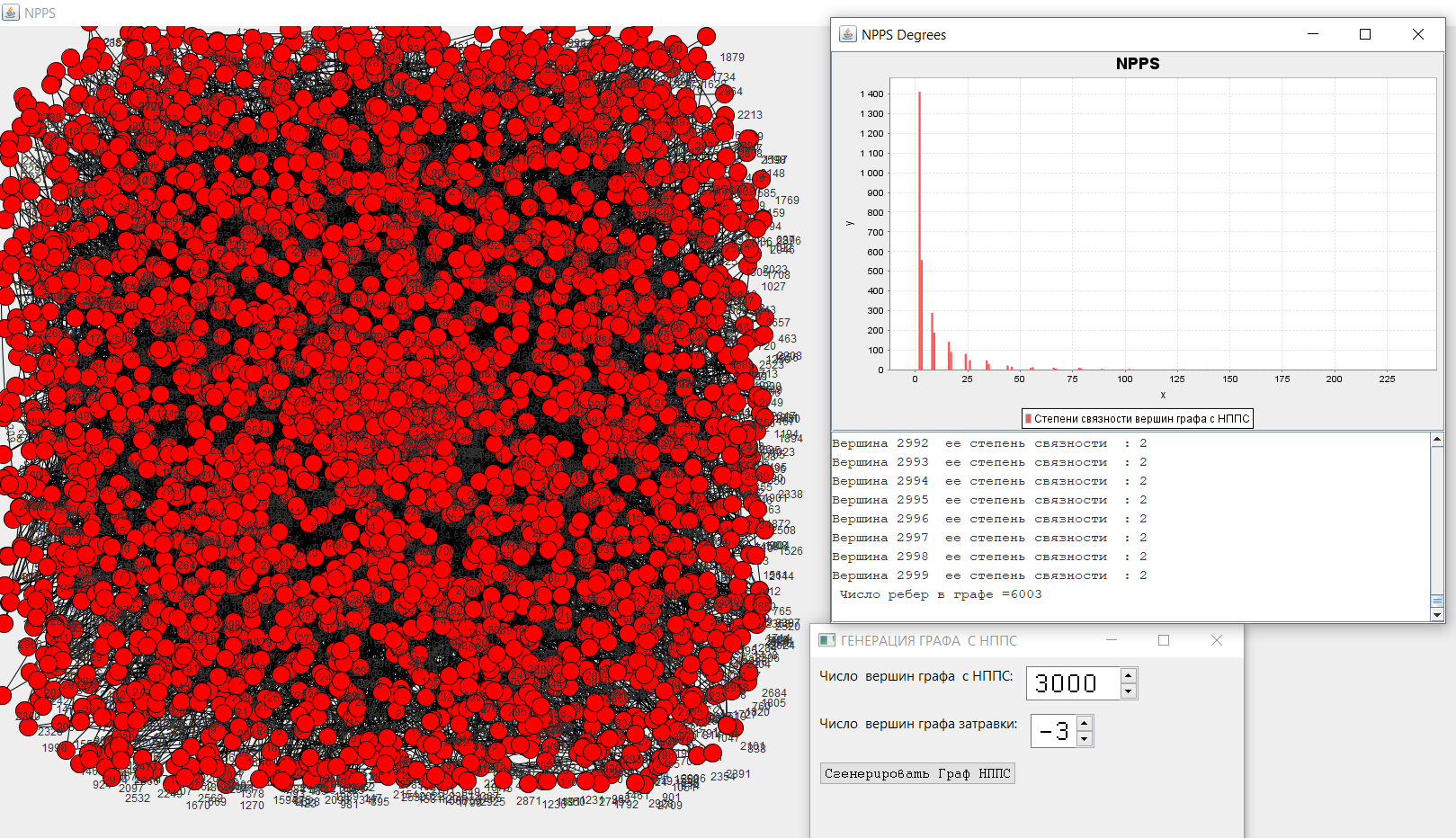
**Рисунок 11.**  Пример\_2 работы программы



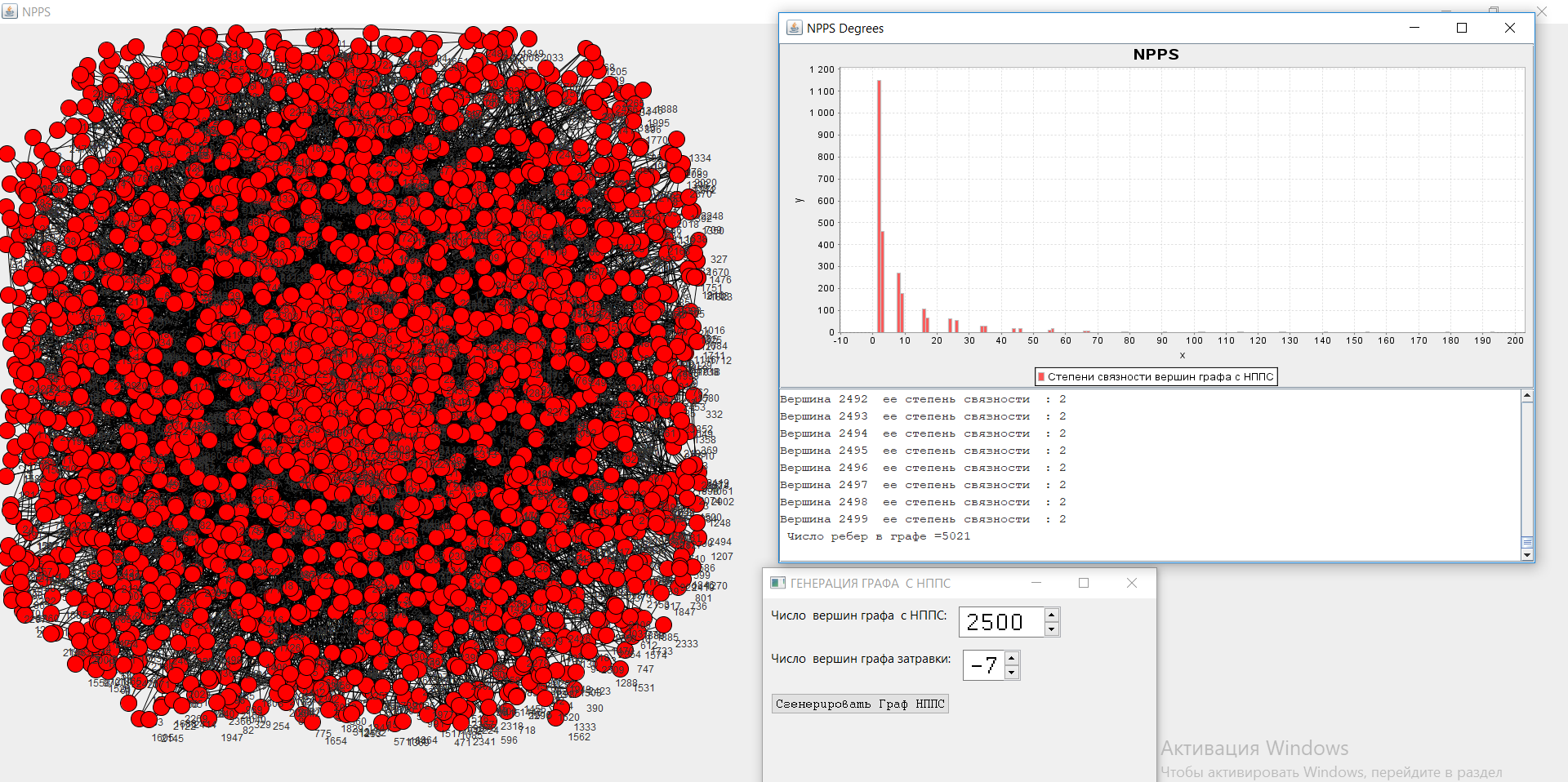
**Рисунок 12.**  Пример\_3 работы программы



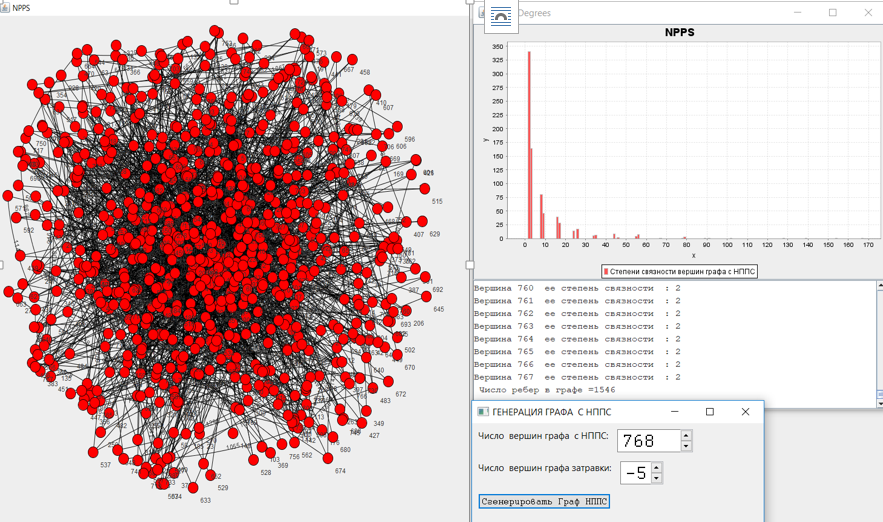
**Рисунок 13.**  Пример\_4 работы программы



**Рисунок 14.**  Пример\_5 работы программы



**Рисунок 15.**  Пример\_6 работы программы



**Рисунок 16**. Пример\_7 работы программы

## 5. 2 Достоинства приложения

Приложение обладает следующими достоинствами:

* 1. Понятный и простой интерфейс окна задания параметров необходимых для генерации графа;
  2. Удобный просмотр каждой вершины графа;
  3. Хорошая визуализация сгенерированного графа.
  4. **Недостатки приложения**

Приложение обладает следующими недостатками:

* 1. Все отображается в разных окнах;
  2. При большом количестве вершин, трудно рассмотреть соседние вершины и связи каждой вершины.

Для улучшения работы приложения можно сделать увеличение просмотра каждой вершины графа с НППС.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта было разработано приложение, позволяющее сгенерировать граф с нелинейным правилом предпочтительного связывания с выводом распределения степени связанности вершин.

Исходя из результатов тестирования JUnit, приложение работает корректно.

Тесты на разных примерах генерации графа с НППС с различным числом вершин также были успешно проведены.

Следовательно, цель курсового проекта была достигнута, все задачи решены.

Приложение может применяться для визуализации графа с НППС при разном количестве задаваемых вершин.

**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Моделирование сетей и сетевых процессов: учеб. пособие / Задорожный В.Н., Юдин Е.Б, Юдина М.Н. – Омск : ОмГТУ, 2017. – 96 с.

2.Сайт библиотеки JUNG [Электронный ресурс] – Режим доступа: <http://jung.sourceforge.net/>, свободный. – Загл. с экрана.

3.JUNG 2.0 Tutorial [Электронный ресурс] – Режим доступа: <http://www.grotto-networking.com/JUNG/JUNG2-Tutorial.pdf>, свободный. – Загл. с экрана.

4.Создание проекта с использованием SWT в IDE Eclipse [Электронный ресурс] – Режим доступа:<http://java-online.ru/libs-swt.xhtml>, свободный. – Загл. с экрана.

5. The ObjectAid UM Explorer is installed from within Eclipse with these steps. [Электронный ресурс] – Режим доступа: <https://www.objectaid.com/install-objectaid>, свободный. – Загл. с экрана.