

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д.В.Топольский  
«\_\_\_» \_\_\_\_\_ 2023 г.

РАЗРАБОТКА ПРОГРАММЫ ДЛЯ НАУКОМЕТРИЧЕСКОГО  
АНАЛИЗА ПУБЛИКАЦИЙ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУРГУ-090301.2023.150 ПЗ ВКР

Руководитель работы,  
к. пед. н., доцент каф. ЭВМ  
\_\_\_\_\_ М.А.Алтухова  
«\_\_\_» \_\_\_\_\_ 2023 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ И.В.Шестерин  
«\_\_\_» \_\_\_\_\_ 2023 г.

Нормоконтролёр,  
ст. преподаватель каф. ЭВМ  
\_\_\_\_\_ С.В.Сяськов  
«\_\_\_» \_\_\_\_\_ 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Д.В.Топольский

« \_\_\_\_ » \_\_\_\_\_ 2023 г.

### **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Шестерину Игорю Вячеславовичу,  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

**1) Тема работы:** «Разработка программы для наукометрического анализа публикаций» утверждена приказом по университету от «\_\_\_\_» \_\_\_\_\_ 2023 г. №\_\_\_\_\_

**2) Срок сдачи студентом законченной работы:** «\_\_\_\_» \_\_\_\_\_ 2023 г.

#### **3) Исходные данные к работе.**

В качестве входных данных программе должны предоставляться файлы с расширением .txt с информацией о публикациях.

Компьютерная программа должна обеспечивать выполнение следующих функций:

– предварительная обработка данных о публикациях из файла, предварительно собранных другой программой;

– построение гистограмм публикаций за каждый год в выбранном диапазоне;

– построение карты с визуальным отображением связей между ключевыми словами в различных публикациях;

– построение карты с визуальным отображением связей между авторами различных публикаций;

#### **4) Перечень подлежащих разработке вопросов.**

В качестве подлежащих разработке вопросов были выделены следующие:

1) анализ предметной области и выбор оптимальной среды разработки и языка программирования;

2) уточнение системных, функциональных и нефункциональных требований;

3) проектирование, разработка модели;

4) разработка и тестирование программы для наукометрического анализа публикаций.

**5) Дата выдачи задания:** «\_\_\_» \_\_\_\_\_ 2022 г.

Руководитель работы \_\_\_\_\_ / М.А. Алтухова /

Студент \_\_\_\_\_ / И.В. Шестерин /

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Анализ предметной области и рассмотрение оптимальных сред разработки и языков программирования	04.03.2023	
Уточнение функциональных и нефункциональных требований	18.03.2023	
Проектирование, разработка модели	01.04.2023	
Разработка и тестирование программы для наукометрического анализа публикаций	11.05.2023	
Компоновка текста работы и сдача на нормоконтроль	18.05.2023	
Подготовка презентации и доклада	29.05.2023	

Руководитель работы \_\_\_\_\_ / М.А. Алтухова /

Студент \_\_\_\_\_ / И.В. Шестерин /

## АННОТАЦИЯ

Шестерин И. В. Разработка программы для наукометрического анализа публикаций. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 38 с., библиогр. список – 18 наим.

Темой выпускной квалификационной работы является «Разработка программы для наукометрического анализа публикаций».

Данная работа включает в себя аналитический обзор предметной области, в ходе которого были рассмотрены некоторые существующие аналоги. Были выявлены недостатки этих систем, с целью их ликвидации в выпускной квалификационной работе. Последующие главы посвящены уточнению функциональных и нефункциональных требований, проектированию, разработке и тестированию программы.

Результатом работы является программа для наукометрического анализа публикаций, реализующая заявленный автором в работе функционал.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	9
1.1 Обзор аналогов .....	9
1.2 Формат входных данных.....	15
1.3 Выбор средств разработки .....	16
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ .....	20
2.1 Минимальные системные требования.....	20
2.2 Функциональные требования .....	20
2.3 Нефункциональные требования .....	21
3 ПРОЕКТИРОВАНИЕ.....	22
3.1 Архитектура предлагаемого решения .....	22
3.2 Алгоритмы решения задач.....	23
3.3 Описание данных .....	25
4 РАЗРАБОТКА И ТЕСТИРОВАНИЕ.....	26
4.1 Реализация графического интерфейса и проведение тестирования .....	26
4.2 Реализация построения гистограммы.....	31
4.3 Реализация построения карт .....	33
ЗАКЛЮЧЕНИЕ .....	35
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	36
ПРИЛОЖЕНИЕ. Развернутая диаграмма основных классов разрабатываемой программы .....	38

## ВВЕДЕНИЕ

Наукометрия представляет собой особую часть научной отрасли, позволяющую отследить эволюцию конкретного направления науки или области наук, а также оценить ее роль в развитии человечества, вклад конкретной работы или статьи.

Наукометрия захватывает как качественные, так и количественные методы оценки научных трудов. Если оценивать работы комплексно, то можно получить полную экспертизу направления науки или их области. Анализируя же ряд показателей или конкретные материалы работы обособленно, сделать аргументированный вывод о качестве и роли научной работы будет сложно и необъективно [1].

Актуальность. В настоящее время активно развиваются компьютерные системы наукометрического анализа научных публикаций, которые призваны облегчить процесс получения информации, необходимой экспертам для определения перспективных направлений в конкретной предметной области. К таким системам относятся цитатные базы научных публикаций, поисковые и аналитические системы.

Системы наукометрического анализа позволяют обрабатывать огромные массивы данных, выявлять и структурировать публикации различных научно-исследовательских направлений, проводить аналитику на множестве публикаций, вычислять различные библиометрические показатели. К таким показателям относят, например, суммарные показатели цитирования, хронологическое распределение библиографических ссылок и другие.

Однако данные системы не лишены недостатков. Некоторые из них имеют исключительно иностранный язык интерфейса, некоторые, ввиду геополитических событий недоступны для российских пользователей, а некоторые не способны работать с Российским индексом научного цитирования (РИНЦ) Elibrary – библиографической базой данных научных публикаций российских учёных и индексом цитирования научных статей [2].

Представленная в данной работе программа – это средство, которое могло бы облегчить и автоматизировать труд научных работников. В дальнейшем данную программу планируется использовать с открытым доступом для желающих провести наукометрический анализ.

Цель работы. Выпускная квалификационная работа посвящена разработке программы для наукометрического анализа публикаций.

Задачи работы:

1) проведение анализа предметной области с целью выявления преимуществ и недостатков некоторых существующих систем наукометрического анализа, а также выбор оптимальных средств разработки;

2) уточнение системных, функциональных и нефункциональных требований;

3) проектирование, разработка модели;

4) разработка и тестирование программы для наукометрического анализа публикаций.



# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Обзор аналогов

В настоящее время существует множество программ для наукометрического анализа. Но большинство из них работают исключительно с форматами определенных библиографических баз, которые распространяются, в первую очередь, на территории недружественных стран. Также у представленных далее программ отсутствуют некоторые полезные функции. Среди обозреваемых в работе аналогичных программ можно выделить следующие:

- программный модуль сбора и анализа данных о наукометрических показателях сотрудников РГПУ им. А. И. Герцена [3];
- Elsevier SciVal [4];
- VOSviewer [5].

**Программный модуль сбора и анализа данных о наукометрических показателях сотрудников РГПУ им. А. И. Герцена** позволяет в режиме реального времени осуществлять вывод структурированных текущих и сравнительных данных о наукометрических показателях преподавателей и научных сотрудников университета.

Данный программный модуль состоит из двух основных компонентов:

- 1) служебный внутренний модуль сбора (корректировки) данных;
- 2) внешний табличный интерфейс вывода данных.

Служебный внутренний модуль сбора (корректировки) данных имеет следующие функциональные возможности и содержание:

- краткие данные об авторах;
- идентификаторы авторов (ссылки);
- добавление/удаление/скрытие авторов;
- цветовые отметки о проверке сотрудником библиотеки;
- переход к табличному онлайн-отчету;
- данные о показателях, которые выводятся в онлайн-форму;

– возможность корректировки данных.

Внешний табличный модуль вывода данных: функциональные возможности и содержание:

– данные о подразделениях (ссылки) – переход к данным по подразделению;

– возможность выбора периода для сравнения;

– ранжирование всех столбцов;

– выделение строки;

– всплывающие подсказки для каждой ячейки;

– экспорт в Excel.

Данный программный модуль был реализован для решения прикладных задач РГПУ им. А. И. Герцена, и недоступен для использования сторонними лицами, что является серьезным недостатком.

Программа **Elsevier SciVal** – аналитическая система, предназначенная для анализа публикационной активности в библиографической базе данных Scopus. SciVal позволяет обрабатывать огромные массивы статистических данных и визуализировать результаты. Система SciVal состоит из нескольких рабочих модулей с различным назначением. Объектами анализа в SciVal являются организации и их группы, исследователи и группы исследователей, области исследования, страны, массивы публикаций.

Возможности SciVal:

1) визуализация результатов научно-исследовательской деятельности;

2) сравнение достижений;

3) развитие сотрудничества.

Функция визуализации результатов научно-исследовательской деятельности включает в себя доступ к полному обзору результатов деятельности любой научно-исследовательской организации, определение ее уникальных сильных сторон и многопрофильных областей исследования, а

также создание наглядных стандартных отчетов и доступ к картам компетенций для всех организаций и стран.

Функция сравнения достижений включает в себя сопоставление результатов деятельности любых организаций, стран и predetermined групп, или создание своих собственных областей исследований и отслеживание прогресса с течением времени. Также в ней есть возможность проводить детальные анализы путем выбора необходимых комбинаций областей исследований и метрик для того, чтобы достичь поставленных целей.

Функция развития сотрудничества позволяет выявлять и анализировать существующие и потенциальные возможности совместной деятельности на основе публикаций и цитирования, ознакомиться с текущими и потенциальными сотрудничествами своей организации на Google maps.

Этот инструмент имеет свои недостатки:

- ограничены возможности экспорта: не более 1000 документов;
- используются зарубежные цитатные базы. Поскольку многие российские направления слабо представлены в зарубежных цитатных базах, эти инструменты не в состоянии предоставить объективную картину;
- для работы с программой необходимо состоять в научном сообществе, имеющем подписку на SciVal.

**VOSviewer** – одна из самых популярных на момент написания работы программ для визуализации, кластеризации и анализа разнообразных данных публикаций. Программа доступна каждому желающему. Она может работать с данными из многих наукометрических баз, и также может автоматически подгружать данные о публикациях из некоторых открытых API.

VOSviewer является программой, с помощью которой исследователи могут визуализировать различные данные в виде наукометрических карт. При построении карты, VOSviewer извлекает термины из аннотаций и ключевых слов публикаций и добавляет их в общий текстовый блок. Далее, VOSviewer выделяет кластеры терминов на основе их совместной встречаемости в

используемых текстах. При визуализации VOSViewer окрашивает определенные кластеры в различные цвета. Пример карты, составленной в данной программе представлен на рисунке 1.1.

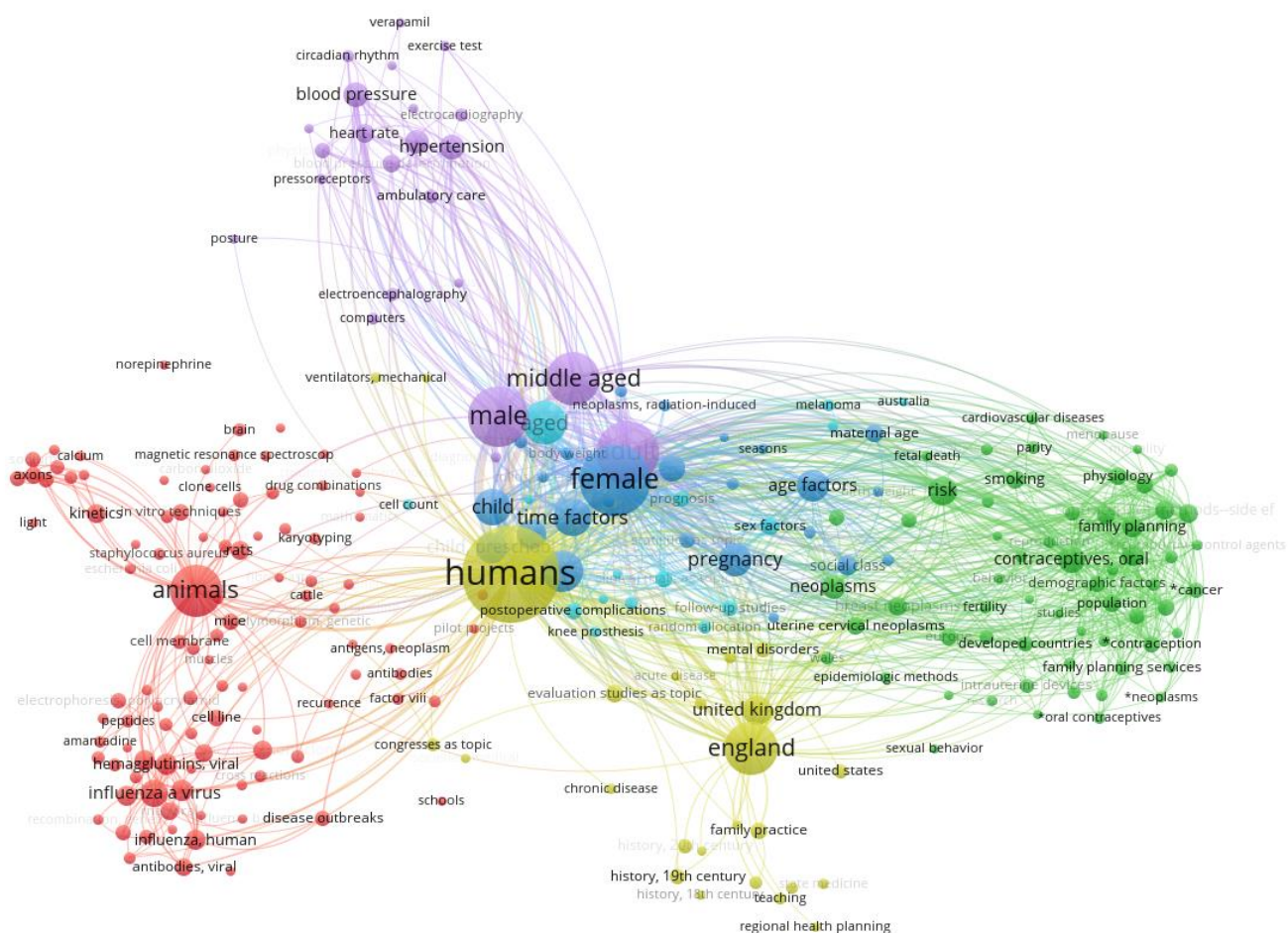


Рисунок 1.1 – Пример карты с анализом ключевых слов в VOSViewer

На рисунке 1.1 видны 5 кластеров, каждый из которых отличается своей тематикой и предметной областью.

С помощью данной программы можно провести:

- анализ ключевых слов и авторов по совместной встречаемости;
- анализ соавторства;
- анализ цитирования.

Эта технология широко используется и может быть полезна в различных областях как средство быстрой визуализации и картирования. Для анализа используются ключевые слова и авторы, которые могут быть взяты непосредственно от автора или извлечены из названий и аннотаций с помощью

внутренних алгоритмов. Термины, полученные из названий и аннотаций статей, представлены на карте, и их размер отражает частоту их встречаемости вместе.

Анализ соавторства. На рисунке 1.2 представлена схема соавторства, где наблюдается большое количество авторов, не имеющих связей с другими, где крупные коллаборации не являются необходимостью. Карта построена на основе фамилий и инициалов авторов, что требует ручного сопоставления различных вариантов написания и ограничивает возможность полноценного анализа представителей стран с малым разнообразием фамилий.

Цитирования. Классический вариант сети основан на цитированиях, именно ссылки в данном случае являются связями между статьями. На рисунке 1.3 показаны только высокоцитируемые (в данном случае 20+) публикации, размер отражает число цитирований. Узнать название и прочие данные можно по клику.

Этот инструмент имеет свои недостатки:

- программа не предоставляет функций для работы с РИНЦ;
- интерфейс программы представлен на английском языке;
- любая визуализация представлена в виде карты с отображением взаимных связей между элементами, что не позволяет провести некоторые виды анализов.

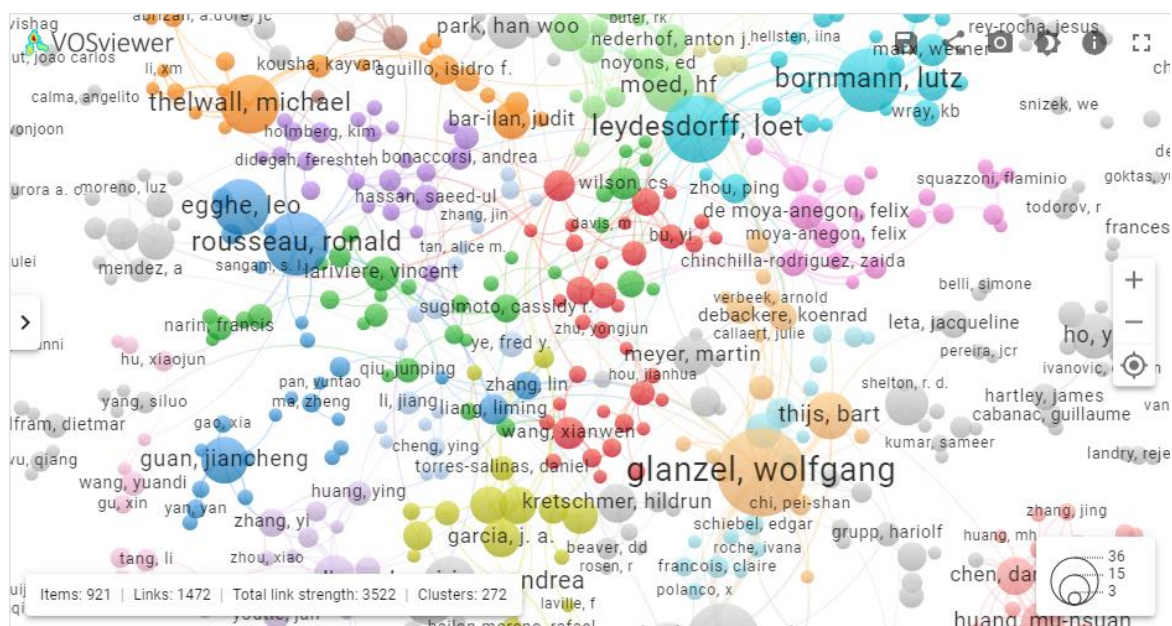


Рисунок 1.2 – Пример карты с анализом соавторства в VOSViewer

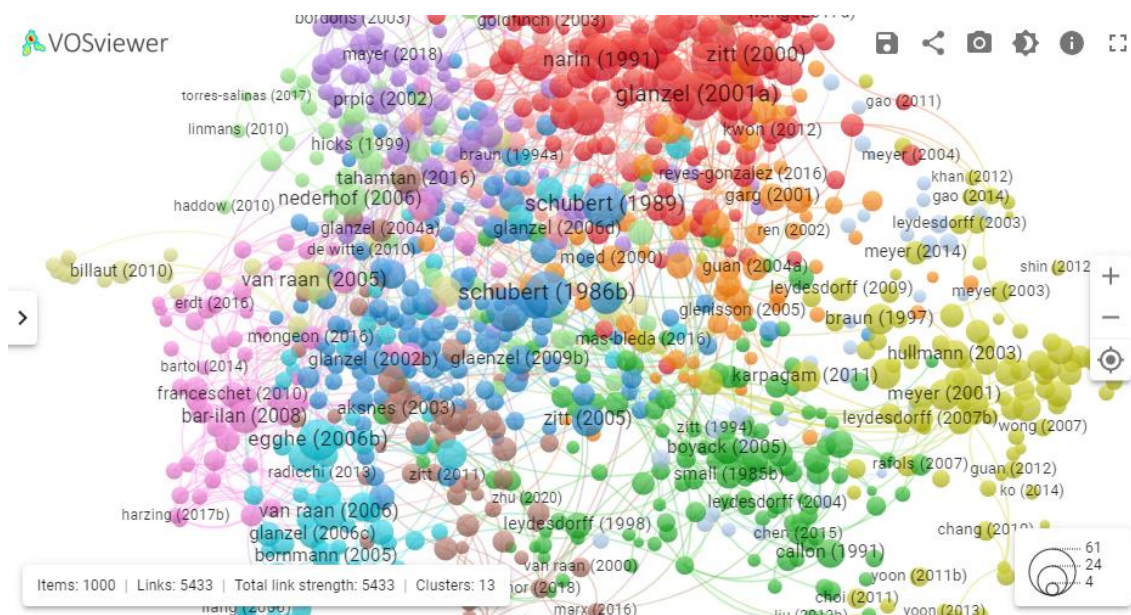


Рисунок 1.3 – Пример карты с анализом цитирования в VOSViewer

Вывод. Из представленных ранее аналогов можно заметить, что данные программы обладают рядом недостатков, главный из которых: большинство из аналогов – иностранные программы, распространяемые на территории недружественных России стран.

Предложенные далее особенности должны не допустить появление представленных ранее недостатков:



– разрабатываемая программа должна иметь интерфейс на русском языке, так как нацелена на российскую аудиторию и призвана работать с РИНЦ;

– разрабатываемая программа должна отображать в графическом пользовательском интерфейсе результаты анализов в разных видах, каждый из которых лучше подойдет для определенных задач;

– разрабатываемая программа должна свободно распространяться для всех желающих воспользоваться ее функционалом.

## **1.2 Формат входных данных**

В качестве входных данных программы для наукометрического анализа используют данные различных форматов. Например, такие базы как Web of Science и Scopus используют свои форматы. В веб-интерфейсе РИНЦ никаких инструментов по выгрузке данных нет (списков публикаций или аналитики нет вовсе), но таблицы из этого интерфейса можно копировать вручную, просто выделяя мышкой, либо с помощью браузерных плагинов, например, table capture.

Однако, так как разрабатываемая программа будет получать на вход данные, предварительно собранные из РИНЦ другой программой, существует риск появления непредвиденных трудностей в разработке. Например, программа по автоматизированному сбору данных из РИНЦ может быть не завершена в срок, или может работать некорректно. И в рамках выбора входных данных было принято решение использования данных другими способами, в универсальном формате данных библиотеки PubMed. Модель данного формата представлена в таблице 1.1 [6].

Данный формат данных достаточно прост для его получения. Он представляет из себя текстовый файл с набор параметров каждой публикации. И в случае непредвиденных проблем разрабатываемая программа сможет загружать данные из базы PubMed [7].

Таблица 1.1 – Модель сбора данных для обработки

Тег	Расшифровка тега
TI	название публикации
PMID	идентификатор публикации
AU	авторы
DP	дата публикации
MH	ключевые слова
LA	язык
TA	издание
SO	тег, который закрывает публикацию

### 1.3 Выбор средств разработки

В качестве языка программирования для разработки стоит выбор между двумя, одними из самых популярных и удобных для данной задачи: C++ и Java. Данный выбор также обусловлен отсутствием знаний и навыков во владении другими языками программирования.

C++ является объектно-ориентированным, многопарадигмальным, функциональным, императивным, процедурным языком программирования. А также этот язык позволяет вам выполнять манипуляции с памятью на низком уровне [8].

Java – современный объектно-ориентированный и типобезопасный язык программирования. Код, написанный на этом языке, запускается в виртуальной машине JVM и без проблем инициализируются в любой системе, где поддерживается соответствующая виртуальная машина [9].

C++ и Java являются языками программирования общего назначения.

C++ находит применение в разработке операционных систем, настольных приложений, веб-браузеров, механизмов рендеринга для веб-браузеров, библиотек машинного обучения, графически интенсивных приложений, систем управления базами данных, встроенных систем, мобильных приложений и других областях. Java используется для создания серверов приложений, веб-приложений, мобильных приложений, настольных



приложений, модульных тестов, корпоративных приложений, игр, облачных приложений, веб-API и других сферах разработки.

Данные языки могут использоваться в разработке настольных приложений. В таблице 1.2 представлено сравнение некоторых особенностей языков [10].

Таблица 1.2 – Сравнение некоторых особенностей языков C++ и Java

<b>C++</b>	<b>Java</b>
Отсутствие контроля границ массивов	Контроль границ массивов
Программа сама управляет выделением и освобождением памяти	Автоматическое освобождение памяти
Поддержка множественного наследования	Множественное наследование реализовано с помощью интерфейсов
Нестрогий контроль типов	Строгий контроль типов
Явное управление памятью, поддержка указателей	Не допускается прямое управление памятью

Главным преимуществом языка C++ является его способность прямого управления памятью устройства. Однако в разрабатываемой программе управление памятью не требуется. Поэтому данный подход может значительно усложнить разработку приложения. Также следует отметить, что разработка графического интерфейса на языке C++ значительно тяжелее разработки на языке Java, ввиду низкоуровневости графических библиотек языка C++. В качестве языка программирования для разработки был выбран Java.

Для разработки графического приложения можно использовать следующие библиотеки и интерфейсы программирования приложений: Swing, JavaFX. Проанализируем недостатки перечисленных решений.

Swing является частью базового класса Java, который полностью независим от платформы. Его используют для создания оконных приложений, и он включает в себя такие компоненты, как кнопка, полоса прокрутки, текстовое поле и т. д [11].

JavaFX – это программная платформа, включающая пакеты с графическими и медиа-инструментами с для разработки и развертывания клиентских приложений и программ с насыщенным графическим пользовательским интерфейсом. Клиентские приложения хранят данные и выполняют большинство своих функций на локальной машине. Программы, использующие JavaFX работают в компьютерной, интернет, мобильной и встроенной среде. Они также могут ссылаться на API (Программный Интерфейс Приложения) других библиотек языка Java и за счет этого использовать большую часть особенностей языка и подключаться к программам с серверной частью.

JavaFX создан с целью замены устаревшего Swing, в котором отсутствуют некоторые современные функции. Поэтому у него гораздо больше возможностей для компьютерной разработки, чем у Swing, что подтверждается приведенным в таблице 1.3 перечнем преимуществ [12].

Таблица 1.3 – Преимущества JavaFX над Swing

<b>JavaFX</b>	<b>Swing</b>
Подходит для разработки программ с современным пользовательским интерфейсом	Библиотека для разработки простых программ с графическим интерфейсом
Возможность писать достаточно понятный код	Много устаревших особенностей
Возможность использования технологии MVC	Отсутствие возможности использования технологии MVC
Поддержка CSS стилей	Создание стилей с помощью кода
Возможность реализации многопоточности	Нет возможности многопоточности
Отсоединен от Java Development Kit, начиная с Java 11	Входит в состав Java Development Kit

Для написания графического интерфейса программы был выбран JavaFX, так как он значительно превосходит своего конкурента, Swing.

При выборе среды разработки на языке Java можно упомянуть две самые популярные на данный момент: IntelliJ IDEA [13] и Eclipse [14].

Среда Eclipse требует значительно меньших ресурсных затрат чем IntelliJ IDEA, имеет более 1250 плагинов против около 750 плагинов в IntelliJ

IDEA. IntelliJ IDEA имеет функцию автозаполнения текста, что может значительно ускорить разработку; имеет более понятный интерфейс; пользуется большей популярностью сообщества Java, в связи с чем упрощается поиск ответа на вопросы, связанные с разработкой программных продуктов.

В качестве среды разработки была выбрана JetBrains IntelliJ IDEA 2022, так как обладает большим количеством достоинств по сравнению с Eclipse.

## **2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ**

### **2.1 Минимальные системные требования**

К общим требованиям можно отнести следующие минимальные системные требования, обусловленные выбранными средствами реализации:

- операционная система: Microsoft Windows 10;
- тактовая частота процессора: 2 ГГц;
- количество ядер процессора: 2;
- оперативная память: 4 Гб;
- java: версии 8;
- место на диске 100 Мб.

### **2.2 Функциональные требования**

Программа должна демонстрировать в графическом интерфейсе следующие результаты наукометрического анализа:

- гистограмма визуального отображения количества публикаций за каждый год в указанном диапазоне;

- гистограмма визуального отображения количества публикаций за каждый год в указанном диапазоне с ключевыми словами, выбранными пользователем;

- построение карты с визуальным отображением наличия связей между ключевыми словами;

- построение карты с визуальным отображением наличия связей между авторами публикаций.

В окне с картой с визуальным отображением наличия связей между ключевыми словами и авторами должно быть текстовое поле с вводом ключевого слова или автора для выделения на карте этого ключевого слова или автора, а также всех ключевых слов или авторов имеющих с ним связь.

Также при построении этих карт должен быть реализован функционал, не позволяющий визуально отобразить недостаточно частые связи между публикациями.

### **2.3 Нефункциональные требования**

К нефункциональным требованиям относятся:

- интерфейс разработанной программы должен быть на русском языке;
- количество публикаций в файле с исходными данными не должно превышать 1000;
- разрабатываемая программа должна быть написана на языке программирования Java в среде разработки IntelliJ IDEA с использованием программной платформы JavaFX.

## 3 ПРОЕКТИРОВАНИЕ

### 3.1 Архитектура предлагаемого решения

При описании архитектуры разрабатываемой программы была спроектирована диаграмма основных классов UML для отображения взаимодействия логических частей приложения, представленная на рисунке 3.1. Развернутая версия рисунка 3.1 представлена в приложении.

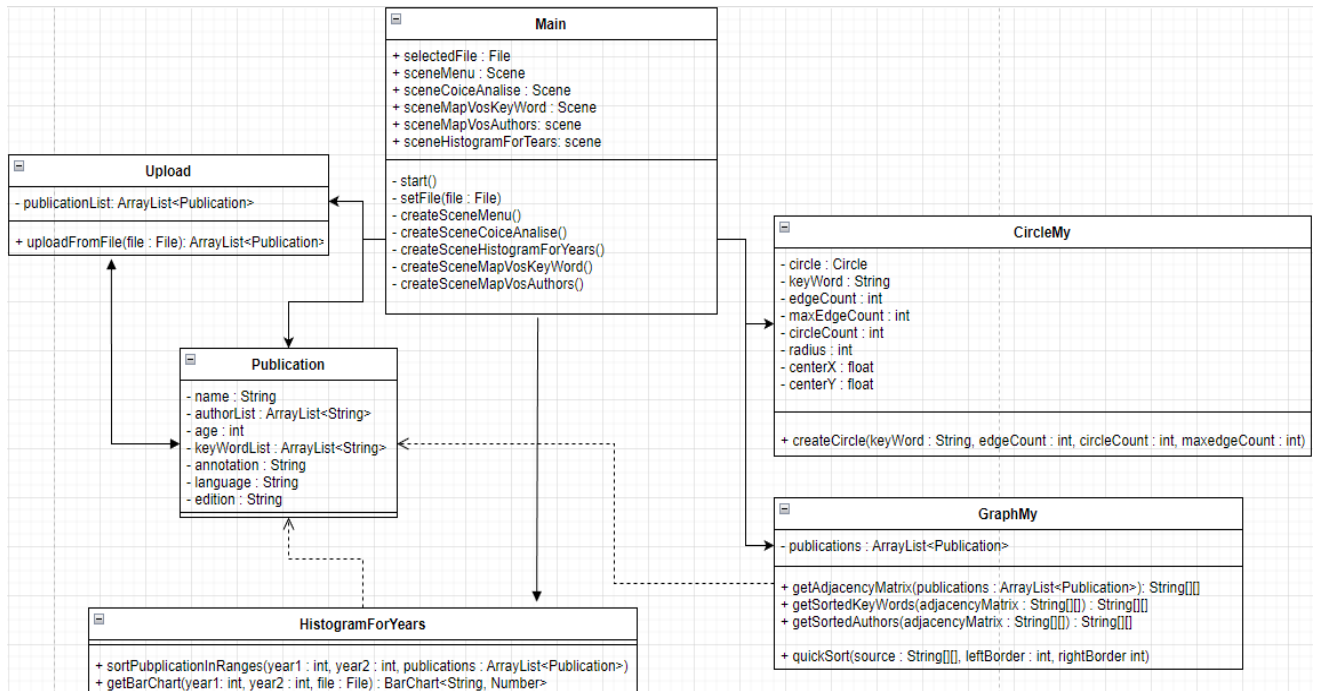


Рисунок 3.1 – Диаграмма основных классов разрабатываемой программы, реализованная с помощью Visual Paradigm Online Express Edition

На представленной на рисунке 3.1 диаграмме классов видны следующие классы:

- Main – главный класс, через который идет управление всей программой, а также создание и переход между сценами анализа;
- Upload – класс занимающийся выгрузкой данных из исходного текстового файла в список экземпляров класса Publication;
- Publication – класс, экземпляры которого являются представлением в оперативной памяти каждой публикации из исходного текстового файла;

– HistogramForYears – класс занимающийся построением гистограммы визуального отображения количества публикаций за каждый год в указанном диапазоне с определенными или любыми ключевыми словами;

– CircleMy – класс хранящий в себе информацию о всех вершинах графа, каждая из которых соответствует тому или иному ключевому слову или автору;

– GraphMy – класс строящий матрицу смежности ключевых слов или публикаций для графа, в виде которого будет представлена одна из карт.

### **3.2 Алгоритмы решения задач**

При запуске программы пользователем главный класс Main создает предварительные окна для отображения всех сцен с целью сократить нагрузку на систему пользователя в процессе создания сцен с выбранными данными исходного файла.

На каждой сцене имеются текстовые поля и кнопки для перехода между сценами. При нажатии на определенную кнопку главный класс Main создает сцену с использованием параметров, указанных в исходном файле и полей текстового ввода.

После выбора пользователем файла программа считывает всю информацию и записывает в оперативную память в виде списка экземпляров класса Publication, с которым далее работают все методы, создающие сцены.

При переходе на сцену гистограммы публикаций по годам пользователь может ввести определенные ключевые слова через пробел. В случае, когда пользователь вводит ключевые слова, программа моделирует гистограмму из количеств публикаций, имеющих данные ключевые слова. В ином случае программа отображает количества публикаций в выбранном диапазоне со всеми возможными ключевыми словами.

При переходе на сцену карты связей ключевых слов программа генерирует матрицу смежности, которая содержит в себе информацию о наличии связей между всеми ключевыми словами в исходных данных. Далее

по матрице смежности создается и отображается на сцене граф, каждая вершина которого – ключевое слово, размер вершины – количество связей ключевого слова с другими, а наличие ребра между вершинами данного графа показывает наличие связи между двумя ключевыми словами, соответствующими соединенным ребрами вершинам. Графическая схема алгоритма построения матрицы смежности представлен на рисунке 3.2

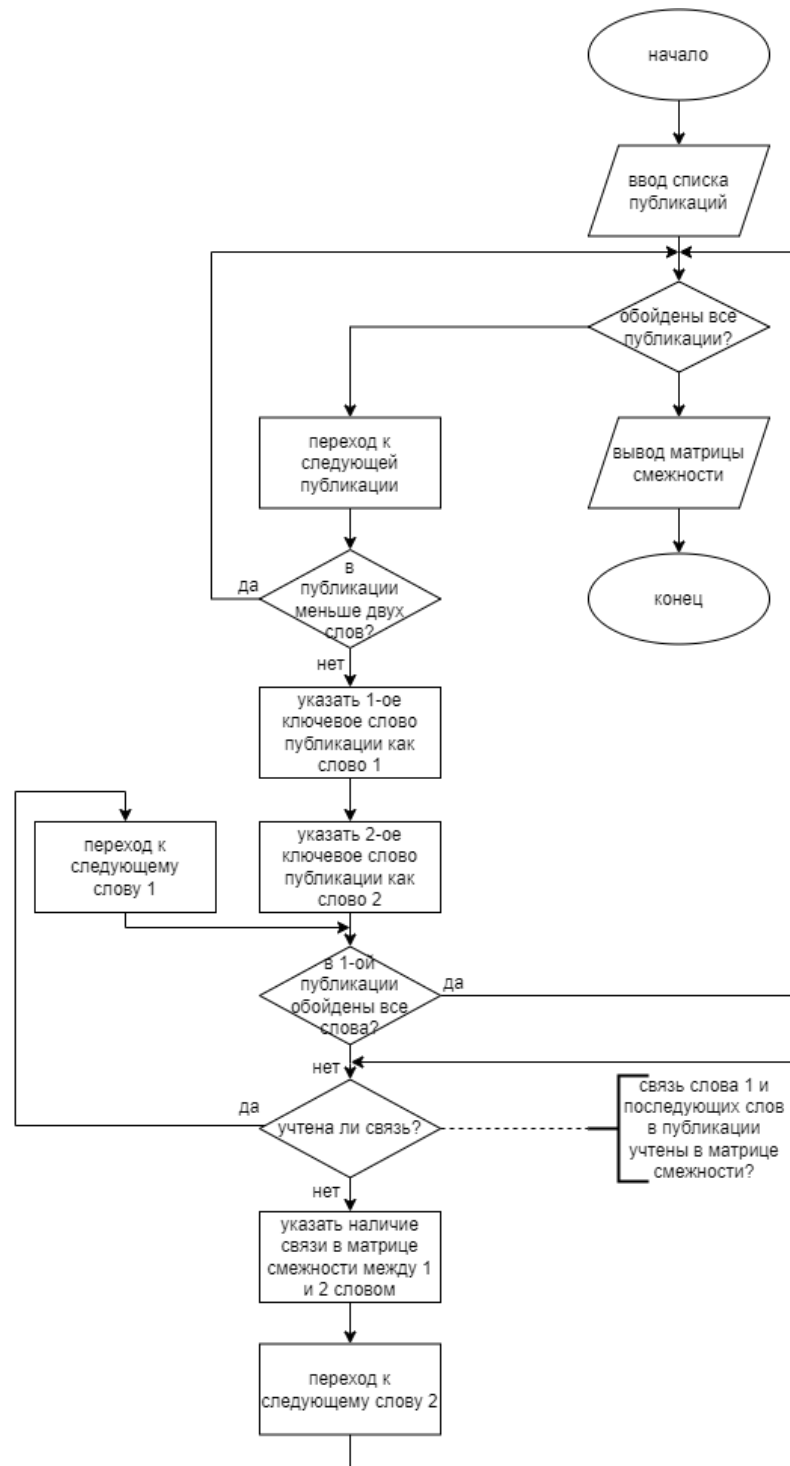


Рисунок 3.2 – Блок-схема алгоритма построения матрицы смежности



Сцена карты связей между авторами публикаций генерируется аналогичным карте связей между ключевыми словами образом.

### 3.3 Описание данных

Входные данные поступают на вход программы в виде файла, содержащего информацию о публикациях. Формат данных о публикациях представлен в таблице 3.1.

Таблица 3.1 – Формат данных о публикациях в исходном файле

<b>Тег</b>	<b>Расшифровка тега</b>
TI	название публикации
PMID	идентификатор публикации
AU	авторы
DP	дата публикации
MH	ключевые слова
LA	язык
TA	издание
SO	тег, который закрывает публикацию

Промежуточными данными являются экземпляры классов, описанных ранее и продемонстрированных с помощью UML диаграммы основных классов, представленной на рисунке 3.1.

## 4 РАЗРАБОТКА И ТЕСТИРОВАНИЕ

### 4.1 Реализация графического интерфейса и проведение тестирования

При разработке программы использовались:

- статьи о разнообразных возможностях и ограничениях языка Java [15];
- каталог готовых ответов на появившиеся в ходе разработке вопросы [16, 17];
- документация к языку стилей CSS, использованном при разработке программы [18].

Интерфейс разрабатываемой программы представлен в виде набора сцен, между которыми пользователь может переходить при нажатии определенных кнопок. Сцена главного меню представлена на рисунке 4.1. Пример сцены с картой с визуальным отображением наличия связей между ключевыми словами представлен на рисунке 4.4. Также в рамках данной главы было проведено функциональное тестирование реализованной программы. Которое включает в себя проверку работоспособности программы с исходными данными, полученными из библиографической базы PubMed.

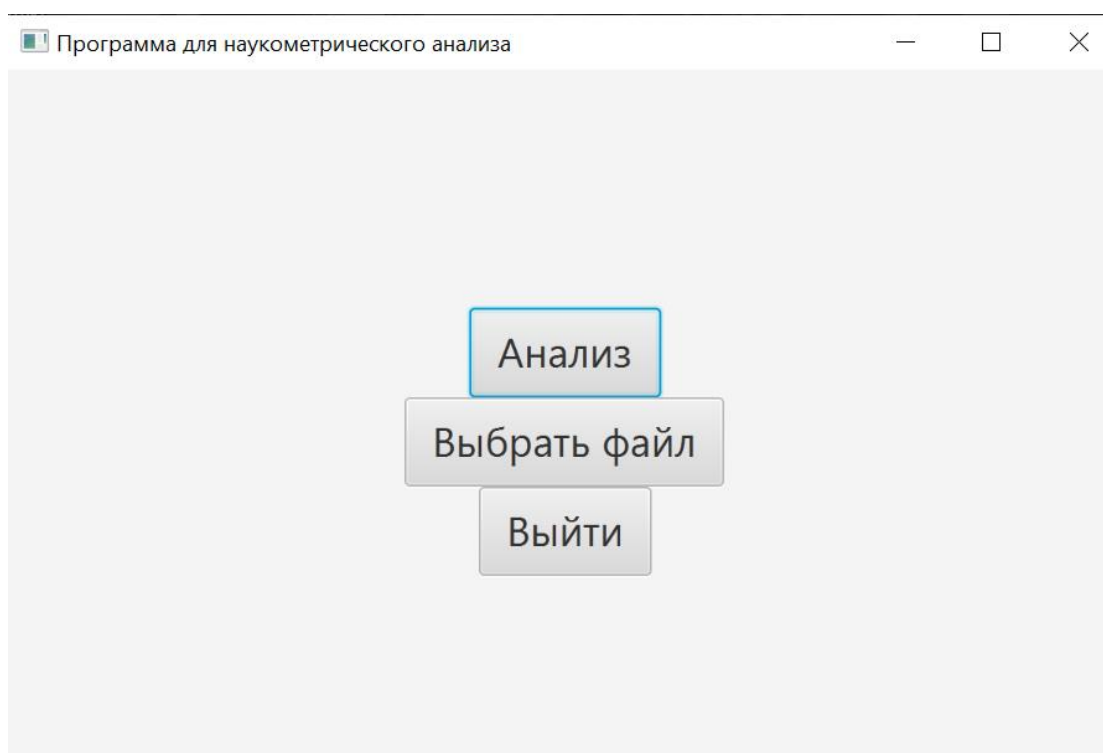


Рисунок 4.1 – Сцена главного меню

Сцена выбора режима анализа представлена на рисунке 4.2. На данной сцене каждая строка относится к определенному виду анализа:

- строка 1 – ввод дополнительных данных для построения сцены гистограммы с визуальным отображением количества публикаций за указанный диапазон и с возможностью ввода ключевых слов;
- строка 2 – построение карты связности ключевых слов;
- строка 3 – построение карты связности авторов.

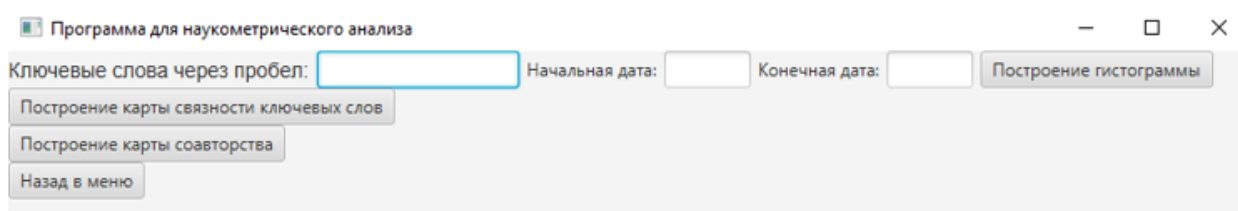


Рисунок 4.2 – Сцена выбора режима анализа

Пример сцены гистограммы с визуальным отображением количества публикаций за указанный диапазон представлен на рисунке 4.3. Пример сцены с картой с визуальным отображением наличия связей между ключевыми словами представлен на рисунке 4.4. Исходный файл с тестовыми данными взят из библиографической базы PubMed за период с 1975 по 1986 год, с публикациями ученых Оксфордского университета.



Рисунок 4.3 – Сцена гистограммы с визуальным отображением количества публикаций за указанный диапазон

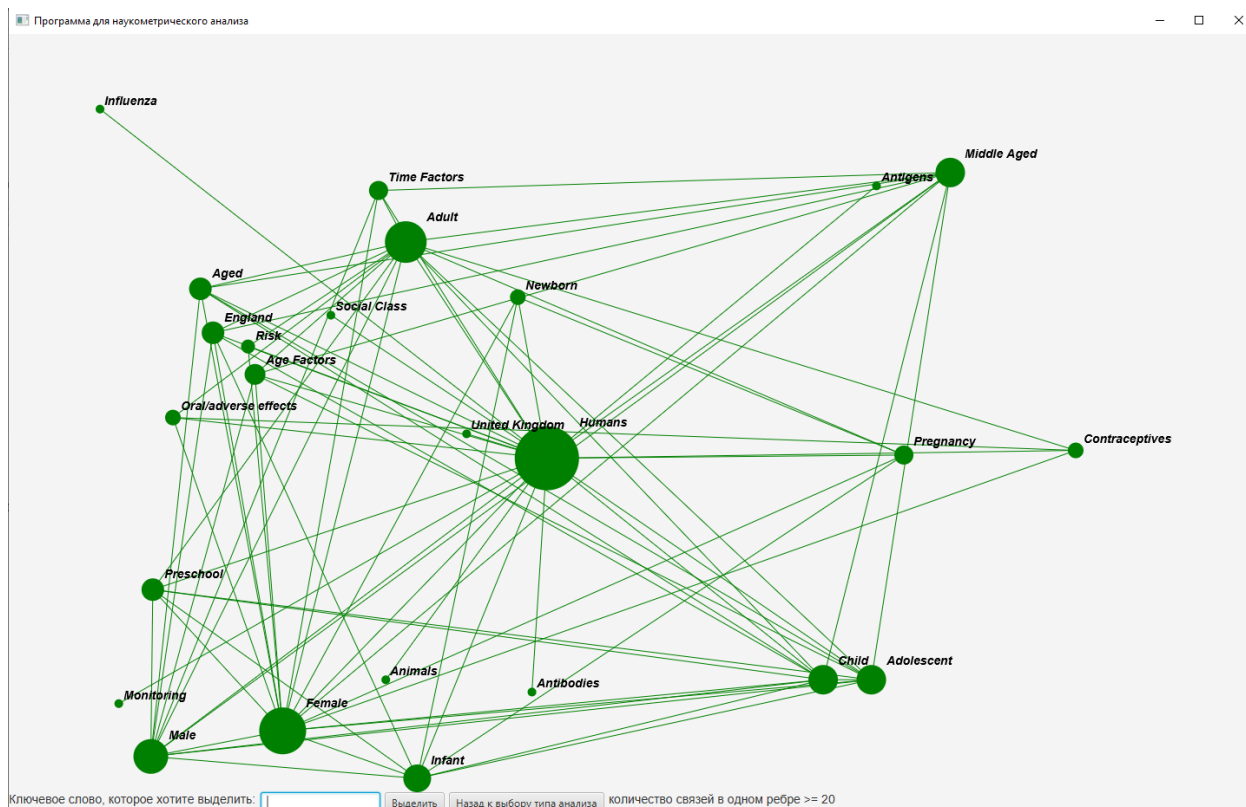


Рисунок 4.4 – Сцена с картой с визуальным отображением наличия связей между ключевыми словами

Осматривая данную карту можно столкнуться со сложностями в поиске связей между ключевыми словами по причине множественных пересечений ребер графа. Поэтому был реализован дополнительный функционал. Он заключается в наличии в нижней части сцены текстового поля и кнопки. При вводе определенного ключевого слова соответствующая вершина и вершины, имеющие связь с выбранной, меняют свой цвет на темно-красный, шрифт текста ключевого слова и слов, связанных с введенным, соответствующим данной вершине, становится больше. Также становятся шире и окрашиваются в темно-красный цвет все ребра между окрашенными вершинами графа. Пример выделения ключевого слова Child представлен на рисунке 4.5. При вводе несуществующего в представленном графе ключевого слова ничего не происходит.

Также на рисунке 4.5 в нижней части сцены видно, что каждая вершина на представленном графе содержит в себе по меньшей мере 20 связей. Это подразумевает то, что если между двумя ключевыми словами на данном графе

присутствует связь, то эти ключевые слова встречаются вместе по меньшей мере в 20-ти публикациях в исходном текстовом файле.

Аналогичный функционал присутствует на карте связей авторов.

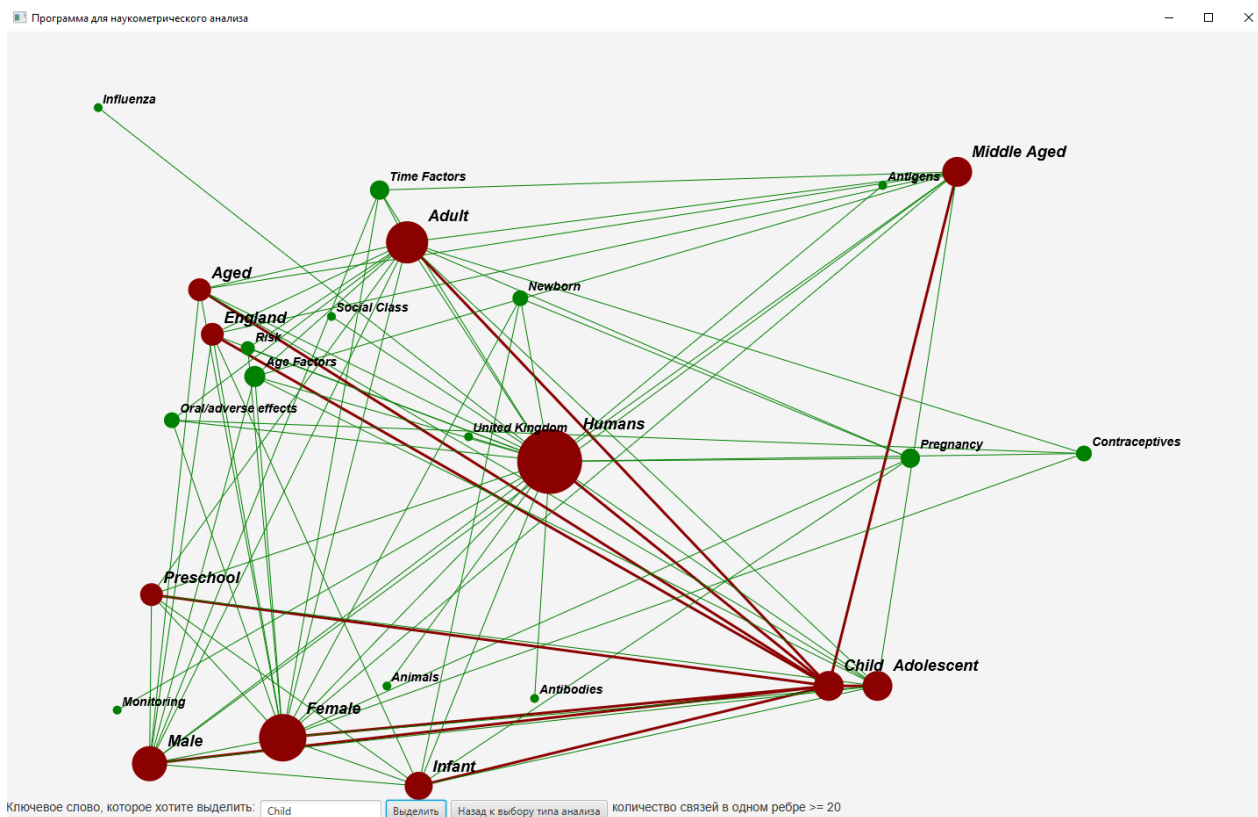


Рисунок 4.5 – Сцена с картой с визуальным отображением наличия связей между ключевыми словами и выделенным пользователем слова

Пример сцены с картой с визуальным отображением наличия связей между авторами публикаций представлен на рисунке 4.6.

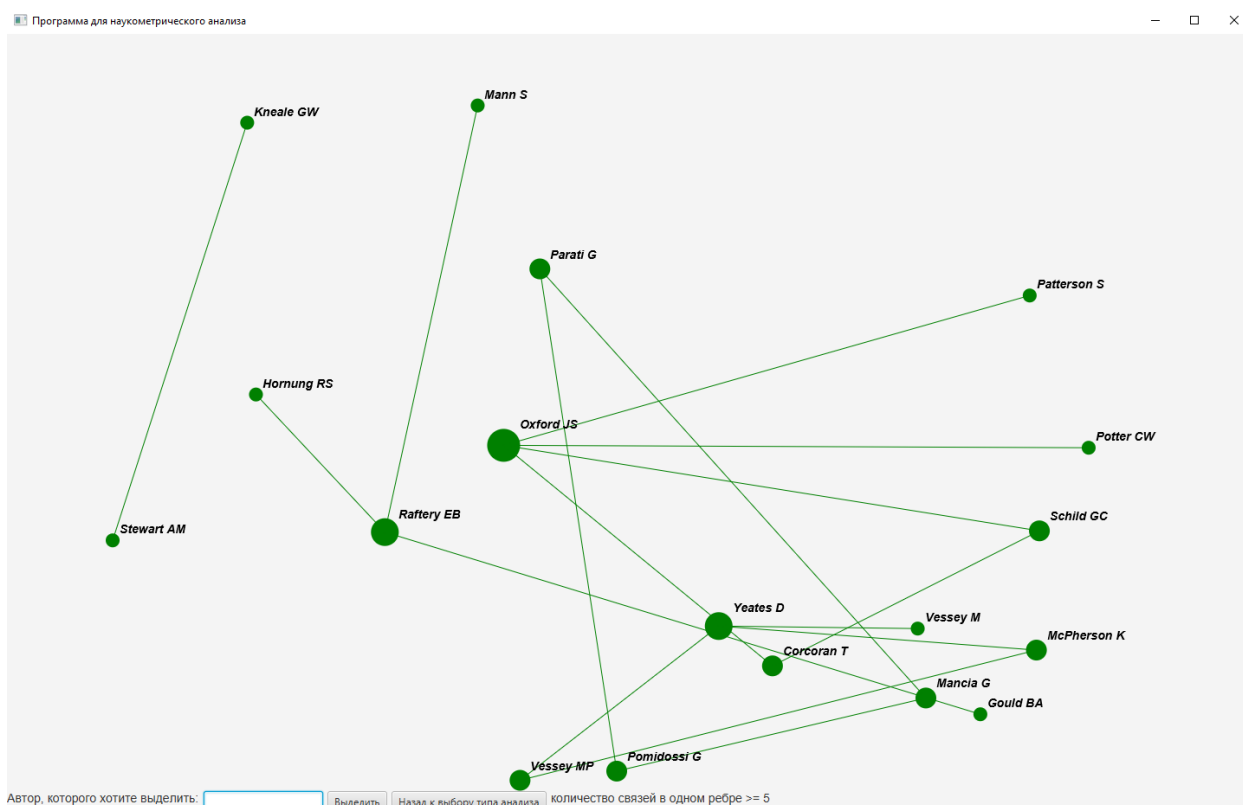


Рисунок 4.6 – Сцена с картой с визуальным отображением наличия связей между авторами публикаций

## 4.2 Реализация построения гистограммы

В целях реализации построения гистограммы с количеством публикаций за каждый год в указанном диапазоне с возможностью выбора публикаций с определенными ключевыми словами был написан класс `HistogramForYears`.

Главный метод данного класса – `getBarChart()`. Он принимает на вход начальную дату, конечную дату, файл с исходными данными, и массив ключевых строк. Выдает на выход данный метод готовую гистограмму, отображением которой на сцене занимается главный класс программы – `Main`. В этом методе вызываются вспомогательные методы класса:

- `sortPublicationInRangesWithKeyWord();`
- `sortPublicationInRanges().`

Их отличает наличие введенных пользователем ключевых слов в окне выбора анализа и отсутствие этих ключевых слов. Два данных метода возвращают массив чисел, который отражает сколько публикаций находится в каждом из восьми поддиапазонов, равномерно распределенных между

начальной и конечной датой, предварительно введенных пользователем.

Реализация метода `getBarChart` представлена на листинге 4.1.

#### Листинг 4.1 – Реализация метода `getBarChart`

```
public BarChart<String,Number> getBarChart(int yearStart, int
yearFinish, File file, boolean keyWordsIsExist, String[]
keyWord) { // компоновка и возврат гистограммы

    ArrayList<Publication> list = null;
    int[] arrayPublicationsInDateRanges;

    try {
        list = Upload.uploadFromFile(file);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }

    HistogramForYears histogramForYears = new
HistogramForYears();

    if (keyWordsIsExist) {
        arrayPublicationsInDateRanges =
histogramForYears.sortPublicationInRangesWithKeyWord(yearStart,
yearFinish, list, keyWord);
    }
    else {
        arrayPublicationsInDateRanges =
histogramForYears.sortPublicationInRanges(yearStart,
yearFinish, list);
    }

    final CategoryAxis xAxis = new CategoryAxis();
    final NumberAxis yAxis = new NumberAxis();
    final BarChart<String,Number> bc = new
BarChart<>(xAxis,yAxis);

    bc.setTitle("Количество публикаций в заданном диапазоне");
    xAxis.setLabel("Поддиапазоны");
    yAxis.setLabel("Количество публикаций");

    XYChart.Series series1 = new XYChart.Series();
    series1.setName("Публикации");

    series1.getData().add(new XYChart.Data("" +
histogramForYears.year1 + " - " + histogramForYears.year2,
arrayPublicationsInDateRanges[0]));
    series1.getData().add(new XYChart.Data("" +
histogramForYears.year2 + " - " + histogramForYears.year3,
arrayPublicationsInDateRanges[1]));
```



#### Продолжение листинга 4.1

```
        series1.getData().add(new XYChart.Data("" +
        histogramForYears.year3 + " - " + histogramForYears.year4,
        arrayPublicationsInDateRanges[2]));
        series1.getData().add(new XYChart.Data("" +
        histogramForYears.year4 + " - " + histogramForYears.year5,
        arrayPublicationsInDateRanges[3]));
        series1.getData().add(new XYChart.Data("" +
        histogramForYears.year5 + " - " + histogramForYears.year6,
        arrayPublicationsInDateRanges[4]));
        series1.getData().add(new XYChart.Data("" +
        histogramForYears.year6 + " - " + histogramForYears.year7,
        arrayPublicationsInDateRanges[5]));
        series1.getData().add(new XYChart.Data("" +
        histogramForYears.year7 + " - " + histogramForYears.year8,
        arrayPublicationsInDateRanges[6]));
        series1.getData().add(new XYChart.Data("" +
        histogramForYears.year8 + " - " + histogramForYears.year9,
        arrayPublicationsInDateRanges[7]));

        bc.getData().addAll(series1);

        return bc;
    }
```

#### 4.3 Реализация построения карт

Основная часть реализации построения карт написана в классе `Main`. Однако с целью улучшения визуального вида кода и упрощения в его понимании было реализовано три вспомогательных класса:

- `CircleMy`;
- `GraphMy`;
- `FieldVosMap`.

Класс `CircleMy` содержит в себе поля с центром круга, ключевым словом, соответствующем данному кругу, радиусом круга и прочими. Экземпляры данного класса используются для упрощения графического отображения вершин графа, соответствующих каждому ключевому слову или автору. Реализация свойств и конструктора класса `CircleMy` представлена на листинге 4.2.

## Листинг 4.2 – Реализация свойств и конструктора класса CircleMy

```
private Circle circle;
private String word;
private int edgeCount; // количество связей
private int maxEdgeCount; // количество связей у самого
распространенного слова
private int circleCount; // количество вершин на странице в
целом
private int radius;
private float centerX;
private float centerY;

public CircleMy(String word, int edgeCount, int circleCount, int
maxEdgeCount) {
    this.word = word;
    this.edgeCount = edgeCount;
    this.maxEdgeCount = maxEdgeCount;
    this.circleCount = circleCount;

    this.radius = 900/circleCount*edgeCount/25 + 3;
    circle = new Circle(0, 0, radius);
    circle.setFill(Color.GREEN);
}
```

Класс GraphMy с помощью написанных в нем методов строит матрицу связей будущего графа, где в первом столбце и первой строке находятся ключевые слова / авторы, а в остальных ячейках матрицы располагаются цифры от нуля. Значение данной цифры определяет сколько раз ключевое слово соответствующей строки и ключевое слово соответствующего столбца встречаются вместе в публикациях исходного текстового файла.

Класс FieldVosMap с помощью написанного в нем метода генерирует для каждого экземпляра класса CircleMy центр круга таким образом, чтобы круги не могли наслаиваться друг на друга при графическом отображении и располагались на минимальном расстоянии в 20 пикселей.

## ЗАКЛЮЧЕНИЕ

Целью работы являлась разработка программы для наукометрического анализа публикаций.

Для достижения данной цели были решены следующие задачи:

- был проведен анализ предметной области, рассмотрены оптимальные средства разработки и выбраны оптимальные из них;
- уточнены системные, функциональные и нефункциональные требования;
- проведено проектирование и разработана модель программы;
- разработана программа для наукометрического анализа публикаций, реализован заявленный функционал, проведено функциональное тестирование на реальных данных.

Поставленные задачи выполнены успешно, цель достигнута. Удобный интерфейс программы позволяет с лёгкостью проводить разные виды наукометрического анализа.

В качестве перспектив в развитии разработанной программы можно выделить следующие:

- построение карты с визуальным отображением наличия связей между ключевыми словами, выделенными из аннотации;
- построение карты с визуальным отображением наличия цитирований между авторами публикаций;
- использование методов искусственного интеллекта в методах построения карт.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Огурцов, А.П. Науковедение / А.П. Огурцов // Нанонаука – Николай Кавасила. – 2013. 143 – 146 с.

2 Наукометрия и библиометрия: зачем измерять результативность научного труда и как это делают в 2020 году. – URL: <https://lala.lanbook.com/naukometriya-i-bibliometriya-zachem-izmeryat-rezultativnost-nauchnogo-truda>. (дата обращения 18.02.2023).

3 Программный модуль сбора и анализа данных о наукометрических показателях сотрудников: интеграционные возможности и перспективы развития. – URL: <https://www.neiconjournal.com/jour/article/view/70>. (дата обращения 21.02.2023).

4 Выявление приоритетных научных направлений: междисциплинарный подход. – URL: [https://www.imemo.ru/files/File/ru/publ/2016/2016\\_001.pdf](https://www.imemo.ru/files/File/ru/publ/2016/2016_001.pdf). (дата обращения 24.02.2023).

5 Citation-Based Clustering of Publications using CitNetExplorer and VOSviewer / Van Eck N.J., Waltman L. / Scientometrics. 2017. Vol. 111(2). 1051 – 1070 с.

6 PubMed User Guide. – URL: <https://pubmed.ncbi.nlm.nih.gov/help/>. (дата обращения 27.02.2023).

7 PubMed Overview. – URL: <https://pubmed.ncbi.nlm.nih.gov/about/>. (дата обращения 26.02.2023).

8 Microsoft C++ language documentation. – URL: <https://learn.microsoft.com/en-us/cpp/cpp/?view=msvc-170>. (дата обращения 29.02.2023).

9 Oracle Java. – URL: <https://www.oracle.com/java/>. (дата обращения 29.02.2023).

10 Лекция 1. Введение в Java. – URL: <https://studfile.net/preview/2090789/>. (дата обращения 29.02.2023).

11 Java-online. – URL: <https://java-online.ru/libs-swing.xhtml>. (дата обращения 29.02.2023).

12 JavaFX Documentation Project. – URL: <https://fxdocs.github.io/docs/html5/>. (дата обращения 29.02.2023).

13 IntelliJ IDEA. – URL: <https://www.jetbrains.com/idea/features/>. (дата обращения 29.02.2023).

14 Eclipse documentation - Current Release. – URL: <https://help.eclipse.org/latest/index.jsp>. (дата обращения 29.02.2023).

15 Javarush. – URL: <https://javarush.com/>. (дата обращения 10.05.2023).

16 Stack Overflow. – URL: <https://ru.stackoverflow.com/>. (дата обращения 10.05.2023).

17 Cyberforum. – URL: <https://www.cyberforum.ru/>. (дата обращения 10.05.2023).

18 Devdocs CSS. – URL: <https://devdocs.io/css/>. (дата обращения 24.04.2023).

# ПРИЛОЖЕНИЕ

Развернутая диаграмма основных классов разрабатываемой программы,  
реализованная с помощью Visual Paradigm Online Express Edition

