

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления  
Кафедра интеллектуальных информационных технологий

*К защите допустить:*

Заведующий кафедрой ИИТ

\_\_\_\_\_ Д. В. Шункевич

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовому проекту  
по дисциплине «Проектирование баз знаний»  
на тему:

**РЕШАТЕЛЬ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНОЙ  
СПРАВОЧНОЙ СИСТЕМЫ ПО ИСКУССТВУ**

БГУИР КП5 1-40 03 01 009 ПЗ

Студент гр. 121702  
Руководитель

С. И. Витковская  
Н. В. Зотов

Минск 2023

# СОДЕРЖАНИЕ

Перечень условных обозначений . . . . .	2
Введение . . . . .	3
1 Анализ подходов к решению поставленной задачи . . . . .	5
1.1 Анализ предметной области искусства . . . . .	5
1.2 Подходы к разработке РЗ ИСС по искусству . . . . .	7
1.3 Сравнительный анализ аналогов . . . . .	7
1.3.1 Pinterest . . . . .	7
1.3.2 Поисковая система Google . . . . .	8
1.3.3 TinEye . . . . .	9
1.3.4 Artlex . . . . .	9
1.3.5 Smithsonian libraries . . . . .	9
1.3.6 Российская государственная библиотека искусств . . . . .	10
1.4 Вывод . . . . .	11
2 Проектирование решателя задач интеллектуальной справочной системы по искусству . . . . .	12
2.1 Проектирование агента определения жанра . . . . .	14
2.2 Проектирование агента определения наличия изображения в БЗ . . . . .	17
2.3 Вывод . . . . .	20
3 Разработка решателя задач . . . . .	22
3.1 Разработка агента определения жанра изображения . . . . .	24
3.2 Тестирование агента определения жанра произведения . . . . .	25
3.3 Разработка агента определения наличия изображения в БЗ . . . . .	26
3.4 Тестирование агента определения наличия изображения в БЗ . . . . .	28
3.5 Вывод . . . . .	29
Заключение . . . . .	30
Список использованных источников . . . . .	31

## **ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

БЗ – база знаний

ИСС – интеллектуальная справочная система

ПрО – предметная область

РЗ – решатель задач

# ВВЕДЕНИЕ

Искусство играет важную роль в обществе: отображает его ценности и идеалы, влияет на эмоциональное состояние людей и их восприятие мира. Все произведения искусства являются отражением той эпохи, в которую были созданы. Авторы мировых шедевров зачастую вдохновляются окружающей действительностью и передают свое видение в работах. Например, наблюдая переход от реализма к модернизму, можно отследить и появление общественных настроений, направленных на разрушение устоявшихся представлений и традиционных концепций.

Изучение предметной области искусства может помочь лучше понять значение искусства и его влияние на общество, обнаружить новые тенденции и перспективы развития. Также изучение искусства идет непосредственно рука об руку с изучением истории, политологии и других фундаментальных наук. Отсюда вытекает необходимость сохранения, обработки и анализа информации о проделанном человечеством многовековом пути.

В нынешнее время, доступность информации и развитие технологий позволяют людям получать новые знания и впечатления, а также проявлять свой творческий потенциал. Сегодня создание решателя задач для ИСС по искусству имеет большую актуальность. Благодаря такой системе пользователи могут получить помощь в определении, анализе и оценке художественных работ, определении стиля и(или) жанра произведения искусства, что может быть полезно для художников, организаторов выставок, коллекционеров и других людей, связанных с искусством.

Ныне существующие информационные системы, специализированные непосредственно на искусстве, состоят в основном из огромного количества книг и статей, и обычному пользователю тяжело разобраться в этом потоке информации и вычленил из него необходимое. Поэтому создание РЗ, который сможет предоставлять необходимый и достаточный объем данных в ответ на запрос пользователя, все еще актуальная задача. Хорошо продуманная ИСС может играть роль информационного ресурса для обучения и исследования и может помочь студентам, преподавателям, деятелям искусства или просто его любителям открывать для себя новые направления, о которых они, возможно, не знали.

Целью настоящего курсового проекта является создание решателя задач интеллектуальной справочной системы по искусству, реализующего функции поиска изображения по заданному описанию и генерация описания, определение жанра и оригинальности заданного изображения.

Основные задачи текущего курсового проекта:

- разработать агента определения жанра задаваемого пользователем изображения и агента определения наличия изображения в базе – составных частей вышеописанного решателя задач, обрабатывающего онтологически

структурированный граф знаний;

– объединение данных агентов, с другими агентами, разрабатываемых в рамках коллектива, в полноценный решатель задач.

# 1 АНАЛИЗ ПОДХОДОВ К РЕШЕНИЮ ПОСТАВЛЕННОЙ ЗАДАЧИ

## 1.1 Анализ предметной области искусства

Искусство развивалось на протяжении всего существования человечества, а с появлением информационных технологий стало приобретать все больше новых форм. Изучение Про «Искусство» способствует духовно-нравственному развитию человека, учит распознавать символы в деталях. Искусство расширяет культурные горизонты: помогает понять и оценить различные культуры, традиции и опыт. Изучение и понимание различных художественных стилей, техник и средств, учит мыслить творчески, так как требует определенного анализа и интерпретирования, которые помогают отточить навыки критического мышления и применять их в других сферах нашей жизни.[1]

Некоторые из ключевых особенностей данной предметной области включают в себя:

Неограниченность идей. В основе искусства лежит творчество, способность приносить в мир новые идеи и перспективы. Человек в мире искусства абсолютно не имеет ограничений на то, что может быть задумано и создано, что может быть интерпретировано по-новому, куда можно привести хаос, а где создать гармонию.

Неограниченность материалов. Произведение искусства может быть создано с использованием широкого спектра средств, от красок и холста до цифрового программного обеспечения, и может принимать форму изобразительного искусства, перформанса, музыки, литературы и многого другого. Искусство выходит за пределы своего физического представления. Оно строится на знаках и символах, на чувствах, которые вкладывает творец в свое произведение и эмоциях, которые его работа вызывает у других людей. Идея может быть продемонстрирована тысячами уникальных способов, вне зависимости от материалов и средств, а проинтерпретирована еще миллионами вариантов.

Важность техники. Технические навыки и мастерство часто являются важными компонентами искусства, особенно в таких областях, как традиционная живопись, скульптура и музыка. Художники, в широком смысле, могут потратить годы на изучение и оттачивание своих навыков, прежде чем они почувствуют себя готовыми создать законченную работу.

Искусство, хоть и ассоциируется в первую очередь с художественной литературой и изобразительным искусством, включает в себя намного больше областей, в том числе:

– Архитектура — это процесс проектирования и строительства сооружений, ключевой вид искусства, который часто упускают из виду. Архи-

текстура может служить средством культурного самовыражения и самобытности, а в проектировании и строительстве зданий и других сооружений может быть много творчества и инноваций.

– Музыка — чрезвычайно важная форма искусства. От классической музыки до поп-музыки и всего, что между ними, музыка может быть невероятно мощным средством передачи эмоций и переживаний в глубоко резонансной форме.

– Скульптура — это процесс создания трехмерного произведения искусства с использованием различных материалов, включая бронзу, мрамор и глину. Скульптура может принимать самые разные формы и использоваться для создания самых разных произведений, от функциональных до чисто эстетических.

– Живопись — это процесс нанесения краски на поверхность, обычно холст или бумагу, для создания изображения или визуальной работы. Он может принимать различные формы, включая масляную живопись, акварельную живопись и акриловую живопись.

– Хореография — это искусство создания и организации движения, как правило, для целей танца, но также и для других видов движений, таких как сценический бой или гимнастика. Хореографы работают над созданием физического языка, который передает смысл и эмоции и требует глубокого понимания человеческого тела, ритма и музыкальности.

– Театральное искусство включает в себя использование актерского мастерства, декораций, костюмов и письменного слова для создания драматического или комедийного представления. Ключевой особенностью театрального искусства является то, что это очень совместная форма творчества, в которой участвует широкий круг художников и исполнителей, работающих вместе, чтобы воплотить пьесу или спектакль в жизнь.

– Художественная литература — это форма искусства, охватывающая широкий спектр жанров и стилей, включая научную фантастику, фэнтези, ужасы и другие жанры. Она часто используется как способ исследования и критики реальных проблем и может быть мощным инструментом для изучения сложных тем и идей от политической сатиры до социальных комментариев и личной драмы.

Все выше перечисленные формы искусства, а также кинематограф, фотография, перформанс являются важными и влиятельными областями в рамках более широкой ПрО «Искусство» и способны создавать мощные эмоциональные и эстетические переживания для своей аудитории. Они играют решающую роль в создании искусства, которое может затронуть человеческую душу и вдохновить нас думать и чувствовать по-новому.

## 1.2 Подходы к разработке РЗ ИСС по искусству

Назначение системы обуславливает акценты, которые будут расставлены при ее разработке. Так, при построении системы в качестве образовательного ресурса, упор будет делаться на достоверность и качество информации, использование данных исключительно из проверенных источников, а так же простота и понятность в изложении. При ориентации же на творческое назначение большее внимание уделяется на разнообразие знаний в системе, вариативность жанров, стилей и техник, возможность дополнять систему.

## 1.3 Сравнительный анализ аналогов

Для проектирования решателя задач ИСС необходимо провести сравнительный анализ уже существующих справочных систем, подметить их преимущества и вычислить недостатки и разработать способы устранения последних.

### 1.3.1 Pinterest

Pinterest — визуальный инструмент для поиска идей и вдохновения.

Поиск изображений происходит по ключевым словам. Введя в панели поиска «Новый год», можно увидеть изображения напрямую ассоциирующиеся с этим праздником: снежные пейзажи, фото украшенных елок, салют. Тут же можно почерпнуть и идеи для подарков. При этом, если какое-то изображение заинтересовало пользователя сильнее остальных, приводятся изображения в более конкретной тематике.

Обработка изображений в Pinterest происходит с использованием алгоритмов машинного обучения для идентификации объектов, людей и окружающей обстановки на изображениях и создания «карты объектов» контента. Pinterest также использует методы категоризации изображений для организации изображений по темам или интересам, что позволяет пользователям легко находить изображения определенной тематики.[2]

Платформа анализирует сходства между изображениями с учетом таких факторов как композиция, объекты, цвет и текстура, для поиска запрашиваемого изображения. Pinterest также учитывает действия пользователя, чтобы определить релевантность и качество результата поиска. Это позволяет платформе постоянно учиться и со временем улучшать результаты поиска.

Достоинства системы:

- Общедоступность;
- Простой и эффективный поиск;
- Категоризация изображений.



Недостатки системы:

- Поиск изображений в Pinterest ограничен визуальным и теговым поиском, т.е пользователи не могут искать контент по полноценному текстовому описанию;
- Т.к. платформа предназначена для визуального вдохновения, здесь практически полностью отсутствуют описания изображений. Даже изображения знаменитых произведений искусства ограничиваются указанием названия;
- Хотя категоризация изображений и является важной частью работы системы, получаемые в результате поиска изображения представляют собой хаотичную совокупность несвязанных между собой картинок.

### 1.3.2 Поисковая система Google

Google — крупнейшая в мире поисковая система интернета. Реализует как возможность поиска изображений по описанию, так и обратную задачу.

Google использует сложный процесс для обработки запросов на поиск изображений. Первым шагом в этом процессе является индексирование изображений. Для этого Google анализирует имена файлов изображений. После того как изображения были проиндексированы, Google использует алгоритмы машинного обучения, чтобы распознать содержимое изображения: объекты, люди и местоположения. Когда пользователь вводит запрос на поиск изображений в Google, поисковая система использует комбинацию алгоритмов понимания изображений и алгоритмов сопоставления, чтобы определить, какие изображения наиболее релевантны запросу.

Однако система поиска изображений Google не адаптирует свои результаты к индивидуальным предпочтениям или истории запросов каждого пользователя, что может затруднить поиск. Это может привести к отсутствию разнообразия в результатах поиска и нерелевантности для пользователя.

Достоинства системы:

- Общедоступность;
- Простой поиск;
- Разнообразие информации;
- Огромная библиотека данных;.

Недостатки системы:

- Непроверенные источники;
- Отсутствие категоризации;
- Отсутствие персонализации.

### 1.3.3 TinEye

TinEye — это платформа обратного поиска изображений, где пользователи могут искать похожие изображения на основе файла или ссылки цифрового изображения. Это полезно, например, если пользователь хочет найти исходный источник изображения или версию изображения в полном разрешении. Обратный поиск изображений TinEye основан на алгоритмах сопоставления изображений на основе контента, которые позволяют пользователям идентифицировать и извлекать похожие изображения из индексированной базы данных.

Достоинства системы:

- Понятный интерфейс;
- Простота поиска.

Недостатки системы:

- Отсутствие поиска изображений по тексту.

### 1.3.4 Artlex

ArtLex[3] — это онлайн-энциклопедия искусства, охватывающая широкий круг тем в этой области. Сайт предназначен для всестороннего и доступного введения в ключевые понятия и темы в искусстве, предлагая широкий спектр статей и учебных пособий по истории, теории и практике искусства, охватывающих все, от древнего искусства до современного искусства. Сайт предназначен для любителей искусства и исследователей всех уровней квалификации, а также для обеспечения глубокого понимания языка и терминологии мира искусства. Сообщество стремится создать всеобъемлющий и точный ресурс о мире искусства и всегда стремится улучшать и расширять содержание сайта. Достоинства системы:

- Общедоступность;
- Простой поиск по тексту;
- Присутствие ссылок на другие статьи.

Недостатки системы:

- Отсутствие поиска изображений по описанию;
- Отсутствие категоризации;
- Поверхностность.

### 1.3.5 Smithsonian libraries

Smithsonian libraries[4] — это обширный источник знаний, информации и ресурсов, связанных со Смитсоновским институтом, одним из крупнейших и самых разнообразных исследовательских и культурных учреждений в мире. В библиотеках хранится более 2,5 миллионов произведений, включая книги, журналы, фотографии, карты, видео и другие материалы,

охватывающие широкий спектр тем и дисциплин. Коллекции библиотек разнообразны и полны, содержат материалы, связанные с историей, наукой, искусством и культурой, и предоставляют множество ресурсов для исследователей всех уровней квалификации. Содержат образовательные программы, виртуальные выставки, галереи, музеи.

Достоинства системы:

- Общедоступность;
- Простой поиск;
- Возможность сортировки итогов поискового запроса по дате публикации, объекту исследования, типу ресурса, авторам;
- Большое количество перекрестных ссылок между статьями;
- Разнообразие информации;
- Масштабный фонд.

Недостатки системы:

- Нет поиска изображений.

### **1.3.6 Российская государственная библиотека искусств**

Российская государственная библиотека искусств [5] - хранилище ценностей российской культуры и искусства, ведущее научно-информационное учреждение страны. Является главной библиотекой России, собирающей фонды литературы по вопросам искусства. Российская государственная библиотека искусств как научно-методический центр, осуществляет разработку научно-методической документации, проводит семинары, способствующие обмену знаниями в рамках библиотечного и музейного сообщества и оказания методической помощи библиотекам. Российская государственная библиотека искусств является методическим центром для библиотек по искусству, музейных библиотек и библиотек, имеющих отделы литературы по искусству. РГБИ является базовым ресурсом для сохранения, изучения и развития культуры, искусства, гуманитарных дисциплин, информационным центром отрасли. Сегодня фонд библиотеки насчитывает около двух миллионов единиц хранения.

Достоинства системы:

- Понятный интерфейс;
- Масштабный фонд.

Недостатки системы:

- Необходимость регистрации или непосредственного обращения на электронную почту для получения пароля доступа к электронным ресурсам и ожидание ответа отнимает огромное количество времени;
- Разрозненность БЗ, отсутствие ссылок друг на друга как на уровне баз знаний, так и на уровне статей в рамках одной БЗ;
- Сложность составления поискового запроса.

## 1.4 Вывод

Исходя из анализа предметной области, подходов к разработке и анализа схожих интеллектуальных систем, можно сделать следующие выводы:

- Разработка решателя задач интеллектуальной справочной системы должна вестись с ориентацией на образовательный подход.
- Важнейшими свойствами, которыми должен обладать РЗ интеллектуальной справочной системы являются удобство и простота поиска и масштабируемость.
- Необходимо избегать хаотичного представления информации. Обязательно указывать описание произведений искусства.
- Недостатки, обнаруженные при анализе аналогов ИСС по искусству, которые необходимо нивелировать при разработке текущего проекта включают: отсутствие поиска изображений по описанию и обратного поиска по изображению, слабая структурированность, заключающаяся в том, что знания во многих из рассмотренных ресурсах не обозначаются относящимися к определенной предметной области, а представляют собой скорее хаотичный набор информации.

## 2 ПРОЕКТИРОВАНИЕ РЕШАТЕЛЯ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНОЙ СПРАВОЧНОЙ СИСТЕМЫ ПО ИСКУССТВУ

Целью текущего курсового проекта является создание решателя задач интеллектуальной справочной системы по искусству, реализующего функции поиска изображения по заданному описанию и генерация описания, определение жанра и оригинальности заданного изображения.

Одним из обязательных условий курсового проектирования является использование концепции многоагентного подхода. Каждый агент – исполнитель логически атомарных процессов – должен взаимодействовать с общей БЗ.

Решатель задач можно разбить на данный список агентов:

- Агент формирования ответа пользователю;
- Агент поиска в БЗ изображения произведения по автору и названию;
- Агент поиска в БЗ изображения произведения по автору и году создания;
- Агент фильтрации для определения изображения наиболее удовлетворяющего запросу пользователя;
- Агент, создающий теги к изображению произведения, для облегчения дальнейшего поиска по запросу пользователя;
- Агент по сравнению имеющихся в БЗ тегов с запросом пользователя;
- Агент обрабатывающий запрос пользователя, осуществляя поиск по базе шаблонов запроса, и вычленяющий ключевые слова, определяющий тип запроса;
- Агент определения процента оригинальности изображения;
- Агент, сохраняющий изображение, введенное пользователем, в базу;
- Агент проверки наличия схожего изображения в базе;
- Агент, определяющий жанр изображения по выделенным на нем объектам;
- Агент генерации изображения по описанию;
- Inpainting по описанию;
- Агент формирования фрагмента БЗ;
- Агент поиска изображения в заданном стиле.

Потенциальными пользователями интеллектуальной справочной системы по искусству могут быть коллекционеры художественных произведений, сотрудники музеев, галерей, выставочных центров и деятели искусства. Выполняя роль образовательного ресурса ИСС может быть интересна студентам и преподавателям или просто любителям искусства, желающим узнать что-то новое. Проектируемый РЗ ИСС по искусству может использоваться для целей, указанных на рисунке 2.1.

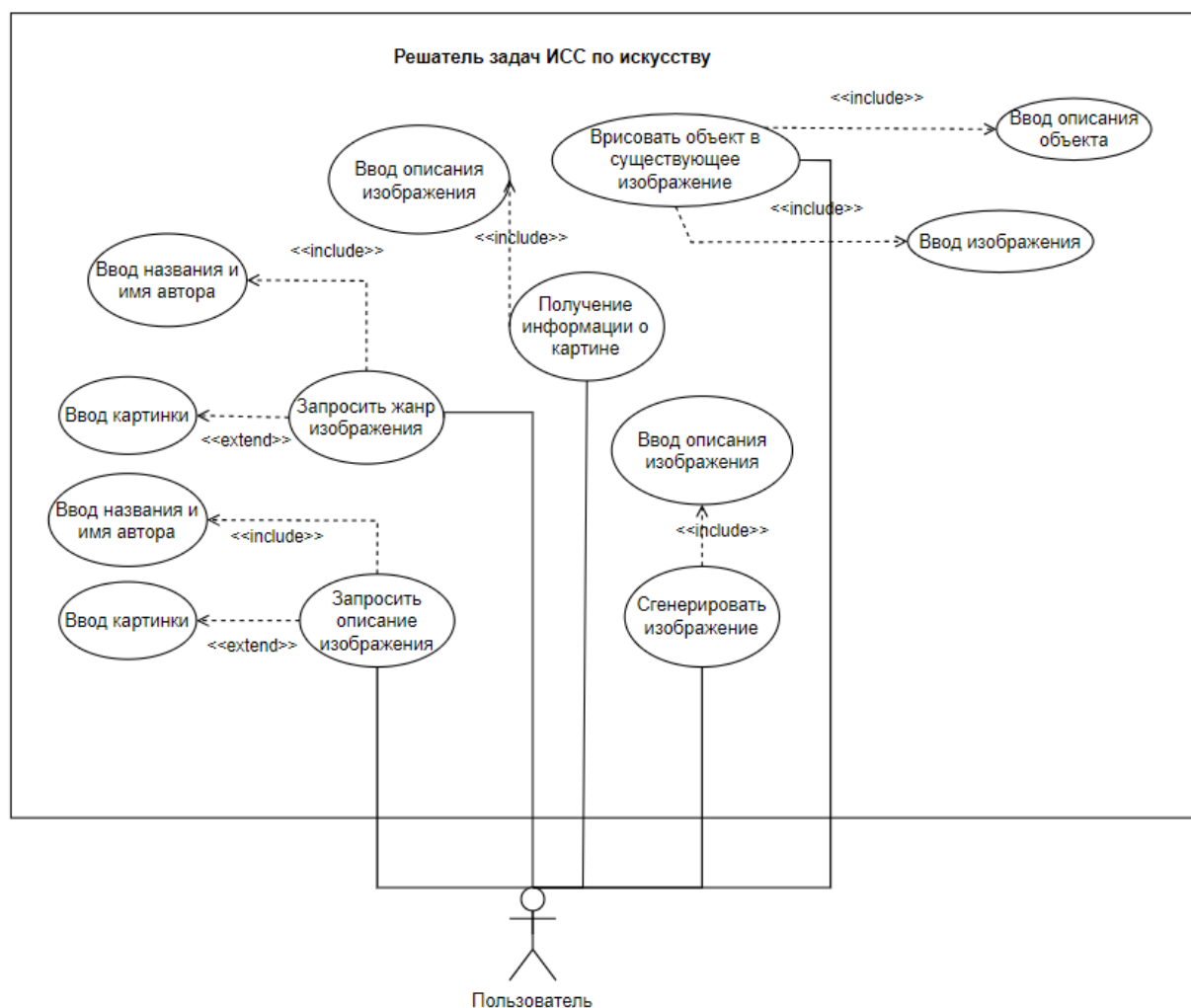


Рисунок 2.1 – Диаграмма вариантов использования решателя задач ИСС по искусству

Второй концепцией, на которой строится тематика текущего курсового проектирования, – использование событийно-ориентированной архитектуры. Событийно-ориентированная архитектура – это парадигма программной архитектуры, способствующая порождению, обнаружению, потреблению событий и реакции на них.[6] В данном проекте, в качестве событий, на которые будут реагировать агенты, принимаются изменения, происходящие в общей базе знаний: создание вершин, их удаление или образование необходимого отношения между вершинами.

Например, при добавлении нового произведения в базу знаний, после анализа текстового запроса пользователя, создаются вершины, хранящие в себе имя автора, название и ссылку на изображение. Вместе с этим происходит триггер агента сохранения, который в свою очередь создает события, на которые реагируют агент поиска по названию и автору и агент проверки на наличие изображения в базе, чтобы убедиться, что подобного объекта в БЗ пока не существует. Агенты поиска проводят отношение между вершиной запроса и найденным произведением, если результат поиска

окажется положительным. Если этого не произойдет, агент сохранения добавит в БЗ новый объект.

Следует учесть, что знания хранимые в базе должны быть представлены в виде онтологически структурированного графа. Формально онтология состоит из понятий, организованных в таксономию, их описаний и правил вывода. Т.е. недостаточно только определить сущности и связи в предметной области, но и категоризировать выделенные понятия и детализировать их.

## 2.1 Проектирование агента определения жанра

Агент определения жанра изображения, передаваемого пользователем, использует заранее определенные теги, ассоциируемые с изображением с помощью соответствующего агента. Эти теги представляют собой выделенные на изображении объекты. Сортируя эти объекты, агент определяет жанр изображения. Подробнее его работу описывает диаграмма на рисунке 2.2.

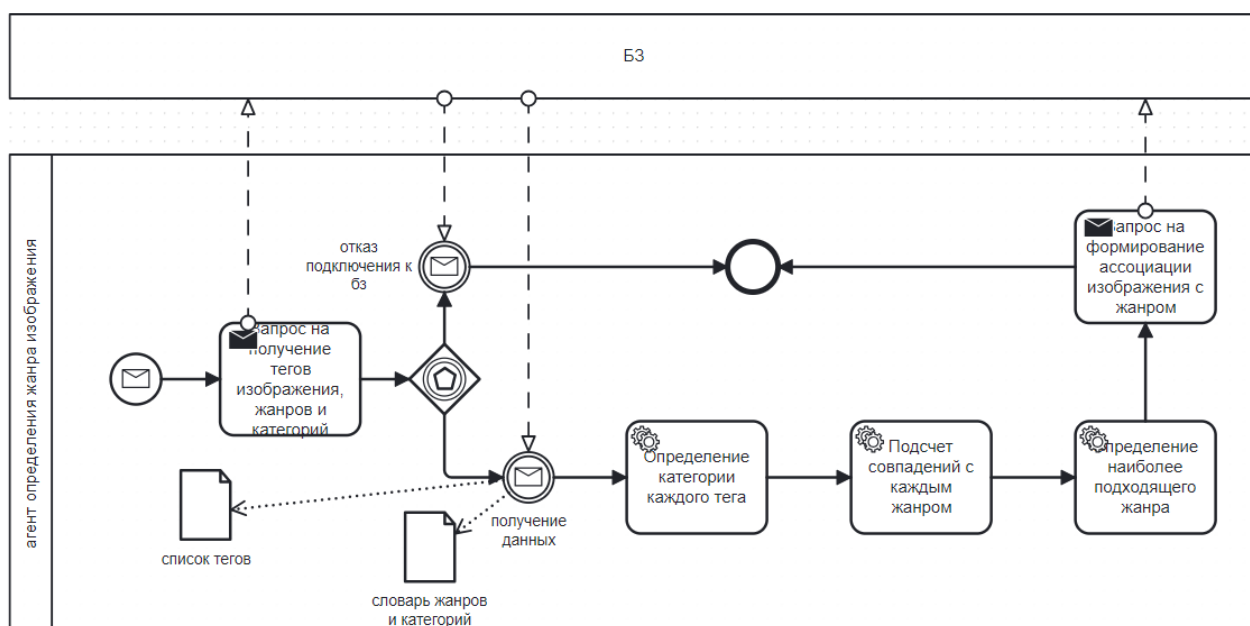


Рисунок 2.2 – BPMN-диаграмма определения жанра изображения

Жанры живописи, имеющиеся в базе, включают в себя: портрет, бытовой жанр, натюрморт, пейзаж, городской пейзаж, анималистика. Каждый жанр можно описать некоторой совокупностью категорий объектов, которые могут быть изображены на соответствующей картине. Так, например, для бытового жанра можно определить категории: люди, одежда, мебель, домашние животные, посуда, хозяйственная утварь и приспособления.

При разбиении таким образом каждого жанра, получается выделить двенадцать категорий объектов. Структура взаимоотношений между жанрами и категориями отображена на рисунке 2.3.

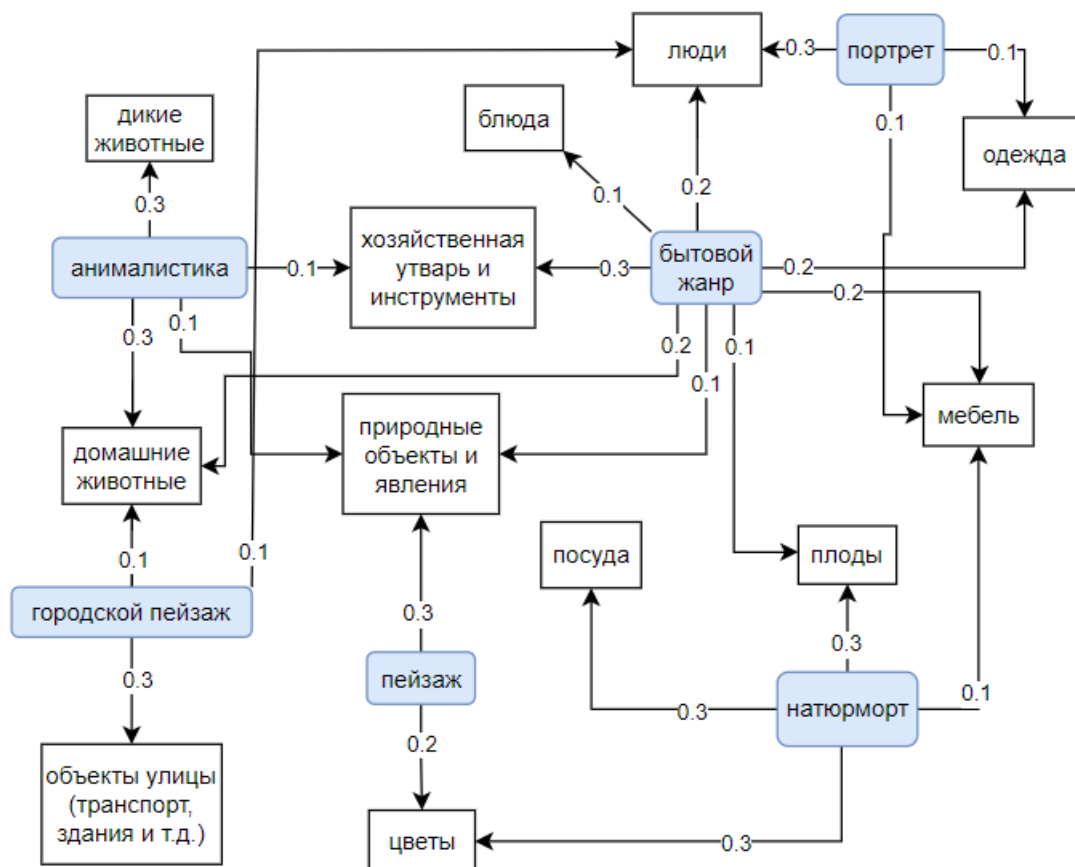


Рисунок 2.3 – Отношения между жанрами изображений и категориями объектов

Числа на диаграмме отношений отображают весовые коэффициенты, которые каждая категория имеет в рамках того или иного жанра.

- 0.3 – объекты данной категории всегда присутствуют на изображениях рассматриваемого жанра;
- 0.2 – объекты данной категории присутствуют на изображениях рассматриваемого жанра достаточно часто, но являются лишь дополнением;
- 0.1 – объекты данной категории изредка присутствуют на изображениях рассматриваемого жанра или могут отсутствовать вообще;

В базе знаний жанры хранятся в качестве классов, принадлежащих классу «concept\_genre», а вершины категорий – классу «concept\_category». При этом все вершины, представляющие собой класс, принадлежат общей вершине «node\_class». На изображении 2.4 приведен пример хранения в



БЗ информации о жанре городского пейзажа (вершина «concept\_town\_scape» посередине изображения), характеризуемого такими категориями как объекты улицы, люди и домашние животные.

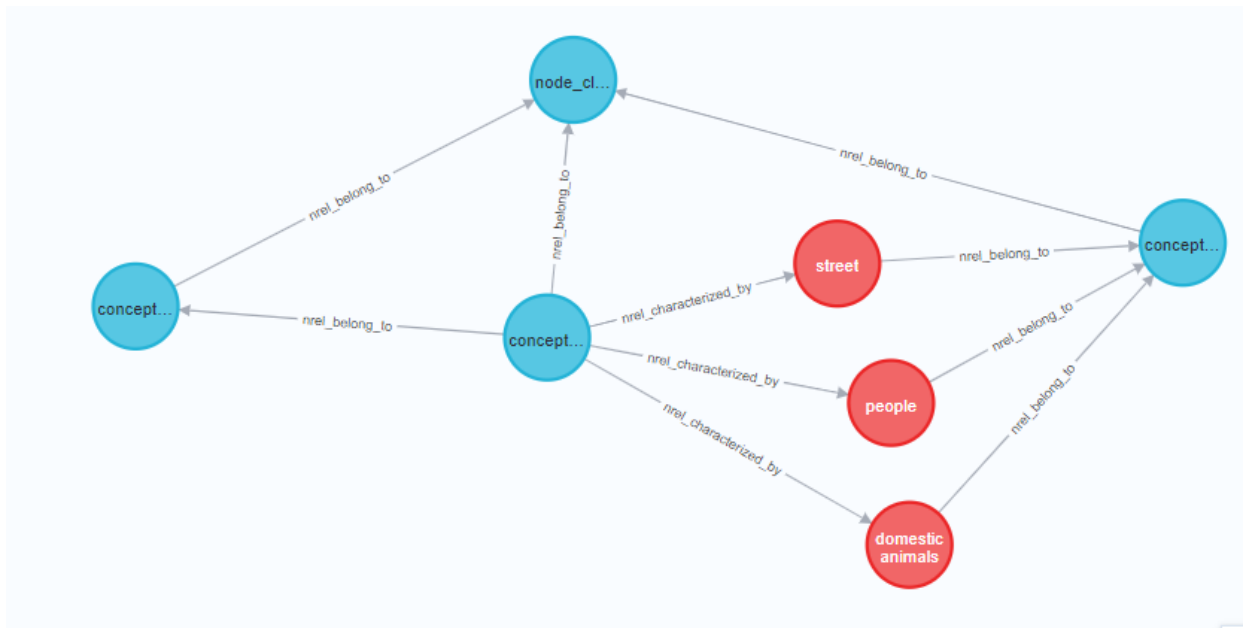


Рисунок 2.4 – Пример представления категорий и жанров в БЗ

Весовые коэффициенты хранятся в БЗ в качестве свойства отношения, связывающего соответствующие жанр и категорию. Пример представлен на рисунке 2.5

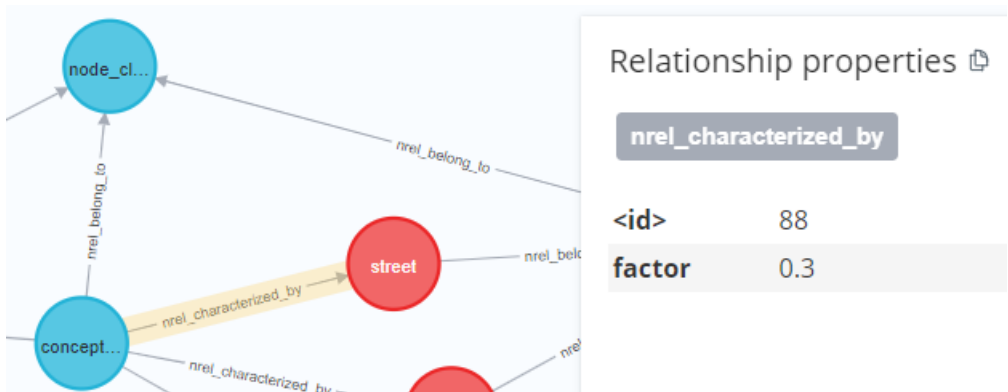


Рисунок 2.5 – Пример хранения весового коэффициента

Диаграмма последовательности выполнения алгоритма работы агента определения жанра отражена на рисунке 2.6.

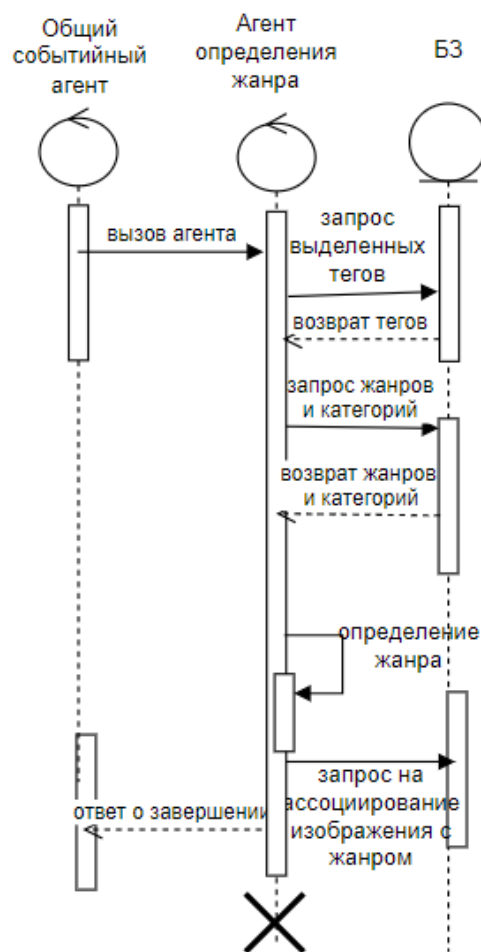


Рисунок 2.6 – Диаграмма последовательности для агента определения жанра

## 2.2 Проектирование агента определения наличия изображения в БЗ

Агент определения наличия изображения в БЗ инициируется событием, создаваемым агентом сохранения и реализует сравнение изображения, передаваемого пользователем и всеми изображениями, хранимыми в базе знаний. Он необходим для того, чтобы не сохранять новые изображения в БЗ, если они являются копиями уже существующих в базе.

Также, использование этого агента гораздо упрощает работу решателя, т.к. если пользователь хочет увидеть описание передаваемого им изображения, при нахождении схожей иллюстрации, хранимой в БЗ, можно использовать ее описание. Работа агента описана на рисунке 2.7.

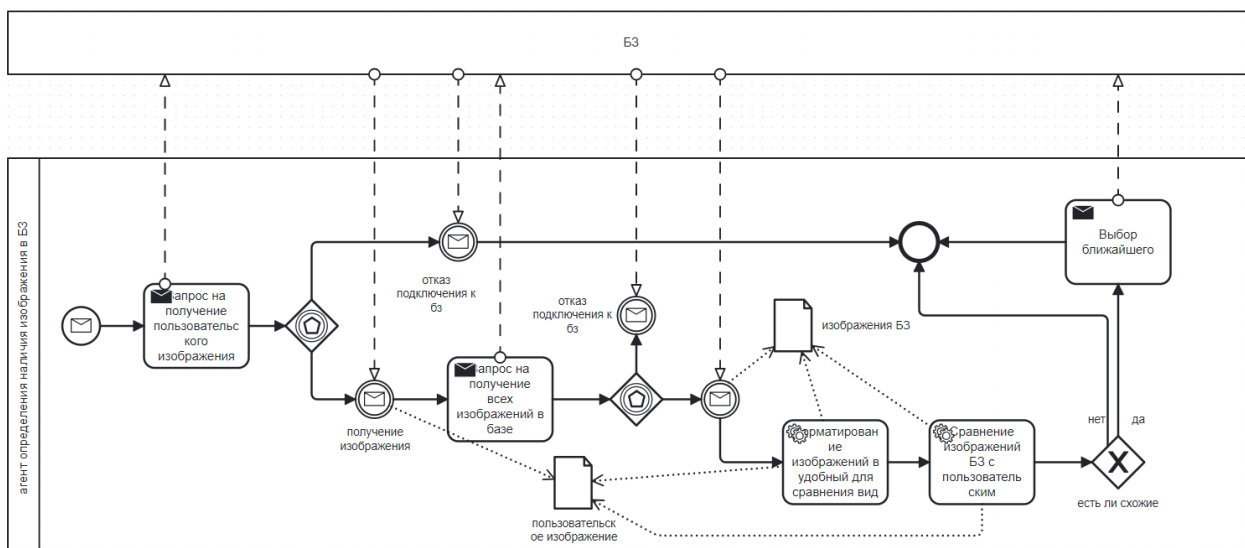


Рисунок 2.7 – BPMN-диаграмма определения наличия изображения в БЗ

Алгоритм сравнения изображений основан на концепции перцептивного хеша. Перцепционное хеширование используется для создания последовательности фиксированной длины, состоящей либо из двоичных, либо из действительных чисел, для представления изображения на основе его содержимого. [7]

Целью перцептивного хеширования является получение одинаковых или очень похожих хэш-кодов для идентичных изображений. Визуально разные изображения должны быть представлены разными хэш-кодами. Общим свойством перцептивных алгоритмов хеширования является то, что преобразования изображений, такие как масштабирование, соотношение сторон или изменение цвета, приводят только к незначительным изменениям хеш-функции. [8]

Алгоритмы перцептивного хеширования имеют свои характеристики и могут выполняться с разной скоростью и эффективностью, в зависимости от типа изображения и требований к алгоритму. Алгоритм вычисления перцептивного хеша, используемый для сравнения изображений в рамках работы агента определения наличия изображения в базе знаний, представлен на блок-схеме на рисунке 2.8.

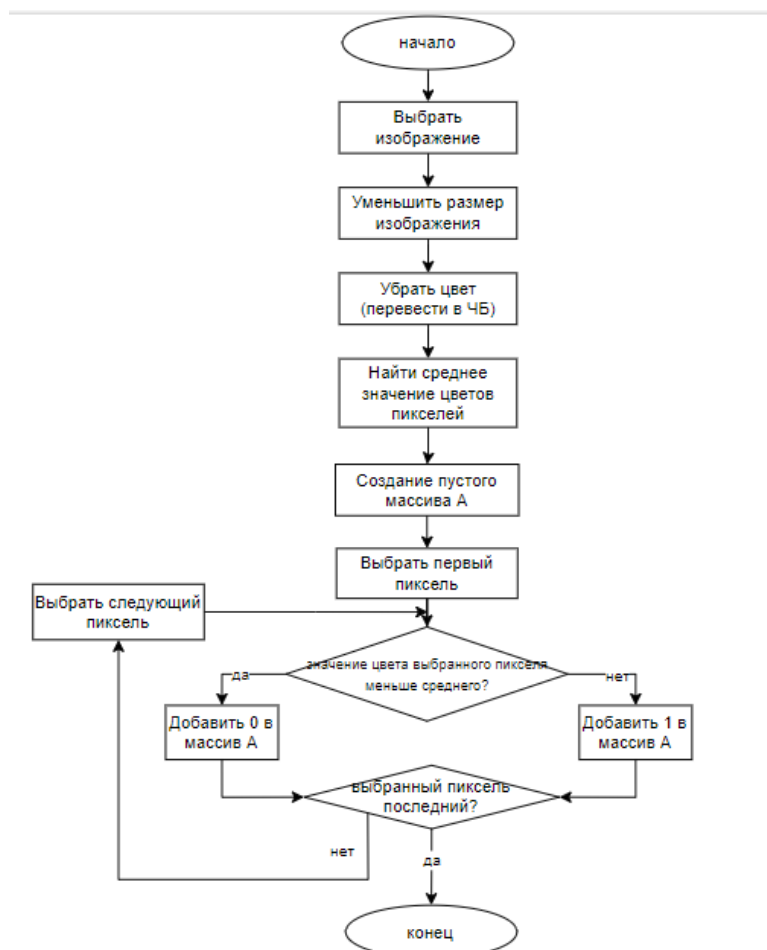


Рисунок 2.8 – Блок-схема алгоритма вычисления перцептивного хэша изображения

Путь, который при этом проходит изображение можно увидеть на рисунке 2.9.

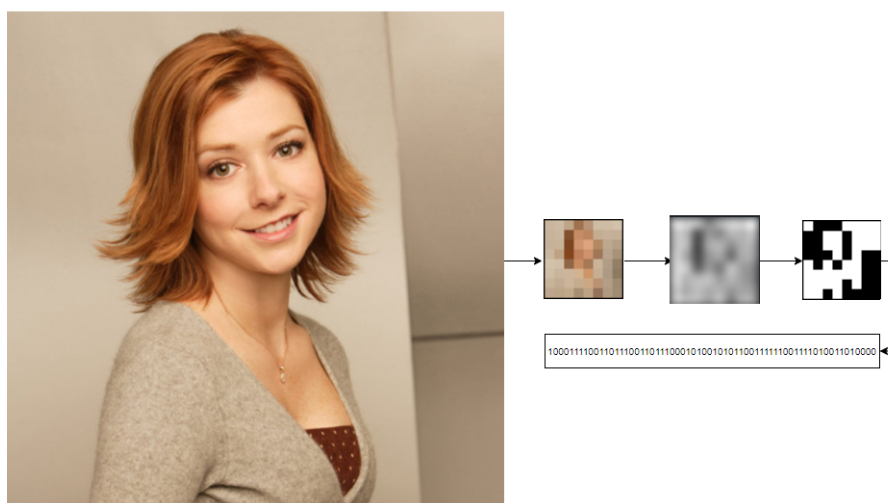


Рисунок 2.9 – Преобразования изображения при вычислении перцептивного хэша

Диаграмма последовательности выполнения алгоритма работы агента представлена рисунке 2.10.

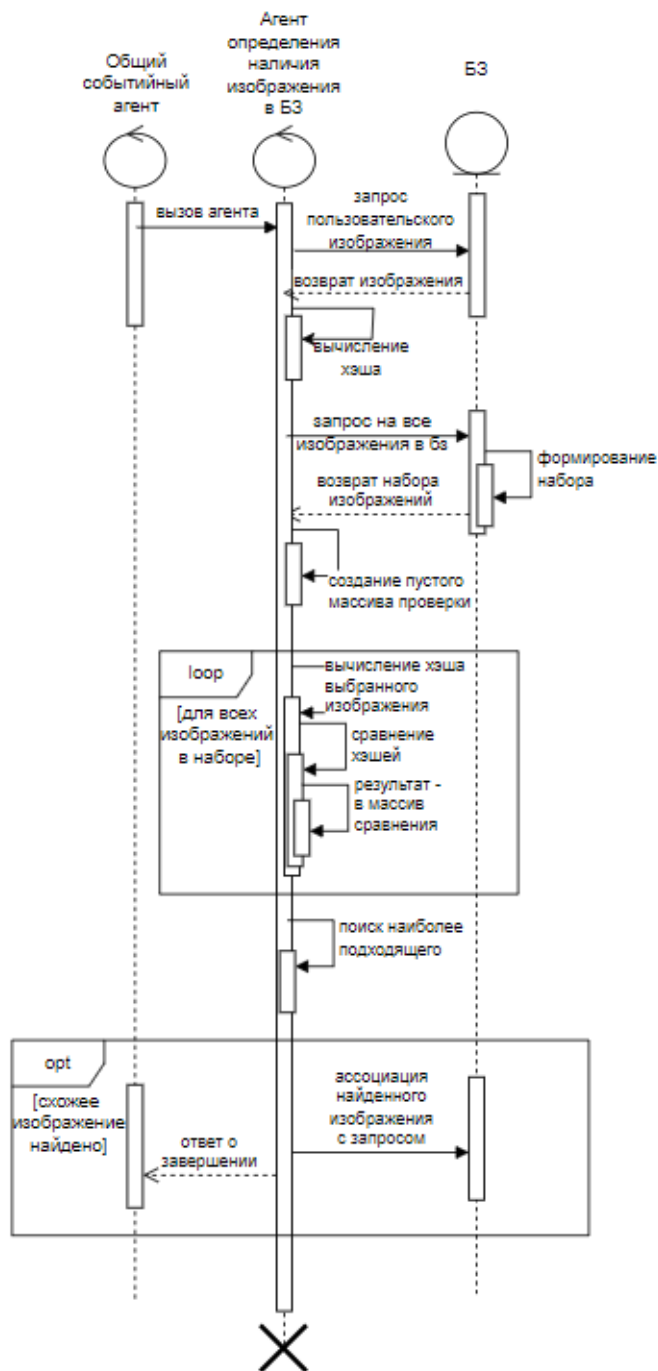


Рисунок 2.10 – Диаграмма последовательности для агента по определению наличия изображения в бз

## 2.3 Вывод

На этапе проектирования были выделены всевозможные агенты РЗ ИСС по искусству, подлежащие реализации, определены задачи агентов

определения наличия изображения в БЗ и определения жанра изображения и описаны алгоритмы их работы, построены диаграммы соответствующих процессов. Определены алгоритмы, необходимые для дальнейшей реализации.

### 3 РАЗРАБОТКА РЕШАТЕЛЯ ЗАДАЧ

Для разработки решателя задач была выбрана графовая система управления базами данных Neo4j, разработанная американской компанией Neo Technology. Она является ведущей графовой СУБД в мире. [9]

В Neo4j всего четыре типа компонентов, которыми можно описать схему данных:

- Nodes — сущности;
- Labels — метки к сущностям;
- Relationships — однонаправленные отношения между сущностями;
- Properties — свойства сущностей или отношений.

Neo4j использует язык Cypher для запросов к базе данных, который предоставляет удобный синтаксис и позволяют создавать графовые структуры с разными типами узлов, рёбрам и свойствами, редактировать и удалять их. Cypher поддерживает регулярные выражения, реализует функции фильтрации и сортировки данных.

Основными запросами Cypher являются MATCH для получения информации из графа и CREATE для создания вершин, отношений или сразу их связок. Основа запроса формируется через структуру (nodes)-[:CONNECTED]->(nodes). [10]

```
CREATE (a:Person {name: 'Tom Hanks', born: 1956})-[r:ACTED_IN {roles: ['Forrest']}]->(m:Movie {title: 'Forrest Gump', released: 1994})
CREATE (d:Person {name: 'Robert Zemeckis', born: 1951})-[:DIRECTED]->(m)
RETURN a, d, r, m
```

Рисунок 3.1 – Пример запроса CREATE на языке Cypher

Например запрос на рисунке 3.1 производит создание вершины типа Person с атрибутами имени и года рождения, отношения ACTED\_IN с указанием конкретной роли актера, также передаваемой в качестве атрибута и вершину Movie. При этом созданные объекты сохраняются по соответствующим ссылкам а, г и м, которые могут использоваться в следующем запросе, как , в данном случае, используется м.

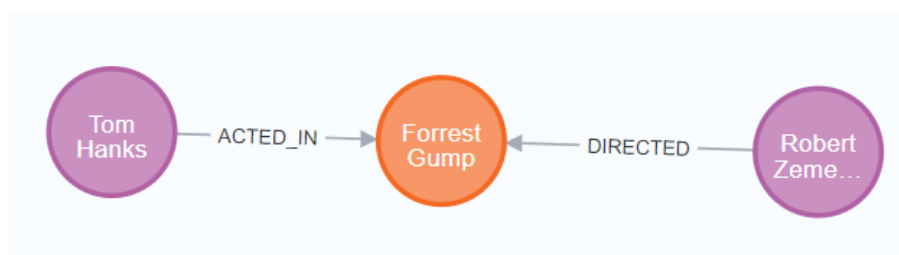


Рисунок 3.2 – Результат выполнения запроса

Из преимуществ Neo4j можно перечислить:

1. Простота и удобство использования. Благодаря языку запросов Cypher, являющемуся интуитивно понятным, облегчается работа с данными и выполнение сложных операций;
2. Наличие обширной документации и активное сообщество пользователей делает процесс изучения и использования Neo4j более доступным;
3. Neo4j поддерживает расширенные запросы, такие как сопоставление с образцом, что в нашем случае может упростить работу с базой знаний, в которой большое количество данных, представляющих собой изображения;
4. Производительность. Neo4j обеспечивает время ответа менее секунды даже на сложные запросы, затрагивающие миллионы узлов.

Однако в некоторых случаях, особенно при работе с большими объемами данных, можно столкнуться с ограничениями масштабирования. Это может быть вызвано ограничениями аппаратного обеспечения или сложностью обработки сложных запросов на больших графах.[11] Но т.к. работа над данным проектом не требует создания и использования больших массивов данных, этим недостатком можно пренебречь.

Интерфейс программирования приложений для Neo4j реализован для многих языков программирования, включая Python на котором будет вестись разработка решателя задач данного курсового проекта. [12] Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным.

Событийно-ориентированное программирование в Python реализуется с помощью модуля `asyncio` начиная с версии 3.4 и доступен во всех последующих версиях.[13] `Asyncio` предоставляет инструменты для написания асинхронных программ, использующих событийное программирование. `Asyncio` позволяет создавать программы, которые могут выполнять множество задач одновременно, используя потоковые выполнения и не блокируя выполнение программы на время, необходимое для выполнения задачи.[14]

В модуле `asyncio` применяются различные концепции событийно-ориентированного программирования:

- Цикл событий, используемый для обработки всех событий, происходящих в программе. Он действует как центральное управляющее звено, обрабатывая события в процессе выполнения программы и отслеживая как завершённые, так и новые входящие события.
- Запланированное вычисление в программе, которое ожидает выполнения.
- Корутины, которые могут приостанавливать свое выполнение и передавать управление другим корутинам, а затем продолжать с того места,



где остановились.

– Классы Transports, используемые для реализации различных типов взаимодействия в программе.

### 3.1 Разработка агента определения жанра изображения

Для реализации агента определения жанра изображения создан класс Genre\_Selection(), функционал которого представлен на рисунке 3.3.

```
30 #поиск тегов изображения
31 async def take_tags(self):
32     response = await self.db_request("MATCH(a:Class{name:'concept_request'})\
33     |-[r:context]->(i:Node)-[:nrel_tag]->(tag:Node) RETURN tag.name")
34     for record in response:
35         self.tags.append(record.value())
36
37 #чтение из файла набора
38 async def take_file_rows(self, filename):
39     with open(filename) as file:
40         return [row.strip() for row in file]
41
42 #определение в файле
43 async def take_tag_cat(self, filename, category, tag):
44     category_tags = await self.take_file_rows(filename)
45     for element in category_tags:
46         if tag in element or element in tag:
47             self.cat[category] +=1
48             break
49
50 #распределение тегов по категориям
51 async def category(self):
52     for tag in self.tags:
53         for category in self.cat:
54             await self.take_tag_cat(f"{category}.txt", category, tag)
55
56 async def count_max(self):
57     max = 0
58     for genre in self.genres:
59         check_max = 0
60         for category in self.genres[genre]:
61             check_max += self.cat[category]
62             if check_max > max:
63                 max=check_max
64                 self.max = genre
65
66 async def get_genre(self):
67     await self.db_request([f"MATCH (p:Class{{name: 'concept_request'}})\
68     |-[r:context]->(a:Node)), (g:Class{{name: '{self.max}'}}) \n CREATE(a)-[:nrel_genre]->(g)"]
69
70
71 async def checking_base():
72     i = Genre_Selection(uri="bolt://localhost:7687", user = "neo4j",password="Qw8pPruC430167")
73     await i.take_tags()
74     await i.category()
75     await i.count_max()
76
```

Рисунок 3.3 – Функционал класса Genre\_Selection()

Метод take\_tags, обращаясь к БЗ с помощью метода db\_request получает теги – найденные на изображении объекты.

Метод category определяет каждый найденный тег в одну из заранее определенных категорий, циклично вызывая метод take\_tag\_cat, который проверяет входит ли тег в список объектов, подходящих для той или

иной категории. Если тег подходит, то значение коэффициента категории увеличивается на единицу.

### 3.2 Тестирование агента определения жанра произведения

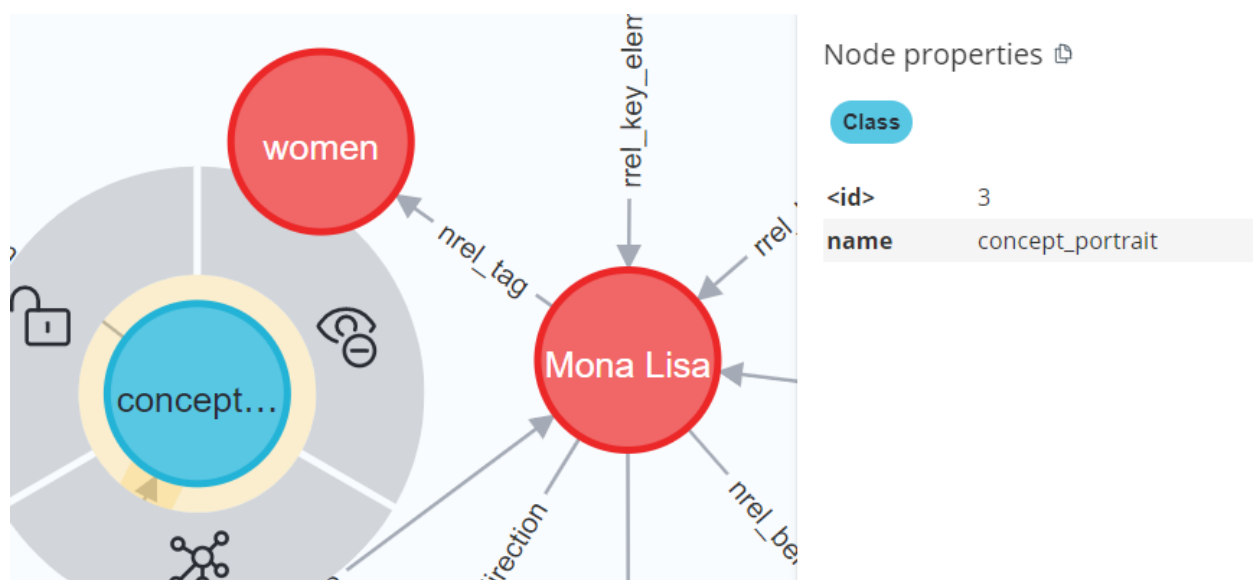


Рисунок 3.4 – Изначальные знания в графе

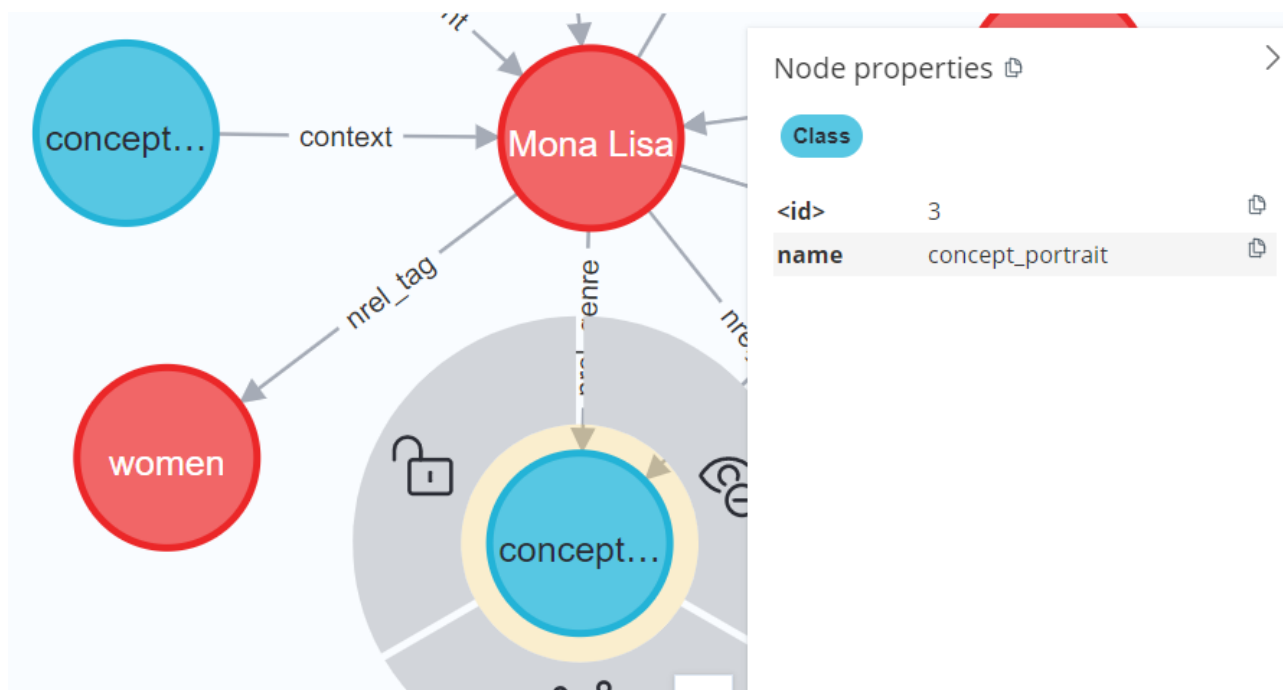


Рисунок 3.5 – Результат работы агента определения жанра

### 3.3 Разработка агента определения наличия изображения в БЗ

Для разработки агента проверки наличия изображения в БЗ использовалась широко распространенная библиотека компьютерного зрения OpenCV.[15] Она содержит множество функций и методов для обработки изображений, таких как загрузка, хранение, преобразование и фильтрация изображений, а также алгоритмы анализа и распознавания объектов.

OpenCV разработан в 2005 году и поддерживает множество языков программирования, включая Python, C++ и Java, что делает его популярным инструментом для создания разных проектов, например биометрических и навигационных систем, систем обработки изображений с камер видеонаблюдения и т.д.

OpenCV имеет открытый исходный код, что позволяет разработчикам адаптировать ее к требованиям проекта с помощью своих собственных расширений. Библиотека также содержит большое количество документации и примеров, которые помогают пользователям работать с ней.

Для реализации агента определения наличия изображения в БЗ создан класс CheckingImgDB(), функционал которого представлен на рисунке 3.6.

```

class CheckingImgDB():
    def __init__(self, uri, user, password):
        self.driver = AsyncGraphDatabase.driver(uri, auth=(user, password))
        self.user_path = ''
        self.user_hash = ''
        self.min = 20
        self.list_of_paths = []

    #вычисление хэша через среднее значение пикселей
    async def find_hash(self, file):
        image = cv2.imread(file)
        resized_img = cv2.resize(image, (8,8), interpolation = cv2.INTER_AREA)
        gray_img = cv2.cvtColor(resized_img, cv2.COLOR_BGR2GRAY)
        avg = gray_img.mean()
        ret, bin_img = cv2.threshold(gray_img, avg, 255, 0)
        hash=""
        for x in range(len(bin_img)):
            for y in range(len(bin_img[x])):
                val=bin_img[x,y]
                hash+="1" if val==255 else "0"
        return hash

    #сравнение по хэшу
    async def compare(self, path2):
        hash2 = await self.find_hash(path2)
        counter = 0
        for i in range(len(self.user_hash)):
            counter += 1 if self.user_hash[i]==hash2[i] else 0
        return counter

    #запрос в neo4j
    async def db_request(self, string): ...

    #поиск пользовательского изображения
    async def take_user_path(self):
        response = await self.db_request("MATCH (a:Text)-[r:user_path]-(il:Class{name:'concept_request'}) RETURN a.content")
        self.user_path = response[0].value()
        self.user_hash = await self.find_hash(self.user_path)

    #формирования списка-результата сравнения
    async def creating_comparison_lst(self):
        for pic in await self.db_request("MATCH (a:Node)-[r:nnel_belong_to]-(il:Class{name:'illustration'}) RETURN a.name"):
            for path in await self.db_request(f"MATCH (a:Text)-[r:nnel_example]->(pic:Node{{name:'{pic.value()}'}}) RETURN a.content"):
                check_min = await self.compare(path.value())
                if check_min < self.min: self.min = check_min
                self.list_of_paths.append((check_min, pic.value()))
        return self.list_of_paths

    #поиск ближайшего по схожести
    async def find_closest(self):
        closest = ''
        for path in self.list_of_paths:
            if path[0] == self.min:
                close = await self.db_request(f"MATCH (a:Node{{name:'{path[1]}'}})-[r:nnel_key_element]->(il:Node) RETURN il.name")
                closest = close[0].value()
                break
        if closest:
            await self.db_request(f"MATCH (p:Class{{name:'concept_request'}}, (a:Node{{name:'{closest}'}}) \n CREATE(p)-[:context]->(a)")
        else: print("не нашел")

    async def checking_base():
        i = CheckingImgDB(uri="bolt://localhost:7687", user = "neo4j",password="Qw8pPruC430167")
        await i.take_user_path()
        await i.creating_comparison_lst()
        await i.find_closest()

```

Рисунок 3.6 – Функционал класса CheckingImgDB()

Метод `take_user_path` обращается к базе для получения изображения, переданного пользователем.

Метод `creating_comparison_lst` циклично получает все изображения, хранимые в БЗ и формирует список кортежей, в которых первый элемент – результат сравнения, получаемый с помощью метода `compare`, второй – изображение из БЗ, с которым проводилось сравнение пользовательского. В этом же методе происходит нахождение изображения с минимальными отличиями от переданного пользователем.

Метод `find_hash`, вызываемый методом `compare` находит хеш изображения: производит сжатие изображения, переводит его в черно-белый

формат и находит среднее значение пикселей. После этого каждый пиксель сравнивается со средним значением и формируется бинарная строка, где единицы - пиксели, значения которых больше среднего, а нули, соответственно, наоборот.

Метод compare сравнивает бинарные строки хешей поразрядно и возвращает количество отличающихся пикселей.

### 3.4 Тестирование агента определения наличия изображения в БЗ

На рисунке 3.7 отображен фрагмент базы знаний: путь к файлу изображения хранится в запросе. В БЗ имеются еще два изображения: портрет Леонардо да Винчи и «Мона Лиза».

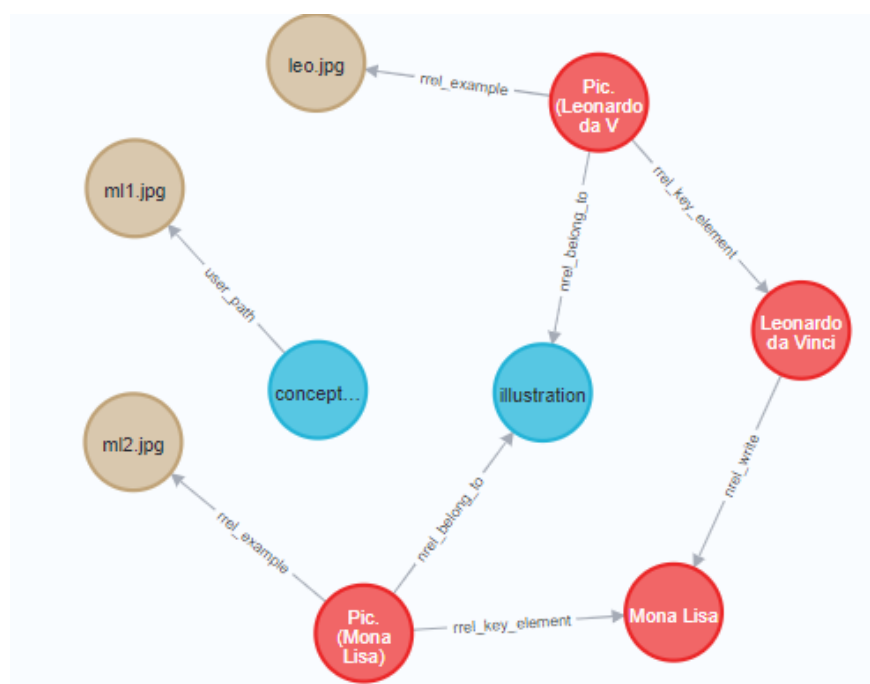


Рисунок 3.7 – Изначальные знания в графе

На рисунке 3.8 слева направо представлены изображения хранимые в файлах «leo.jpg», «ml2.jpg» и «ml1.jpg». Файл «ml1.jpg» является картиной, передаваемой в систему пользователем. Файлы «leo.jpg», «ml2.jpg» хранятся в базе изначально.

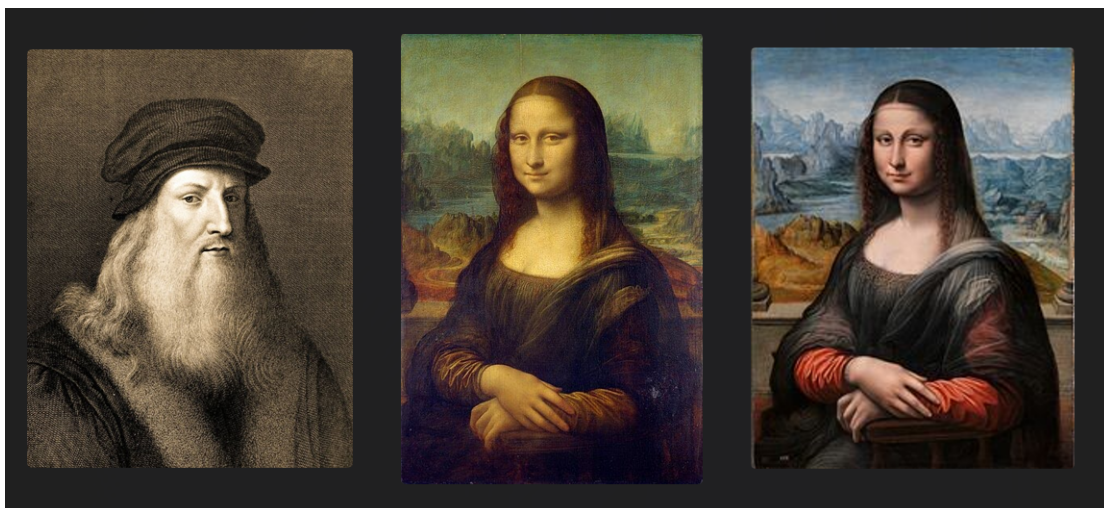


Рисунок 3.8 – Изображения в БЗ

В результате на рисунке 3.9 с вершиной запроса была проассоциирована вершина картины «Мона Лиза». При визуальной оценке видно, что файлы «ml2.jpg» и «ml1.jpg» изображают одну картину, но в разной цветовой обработке.

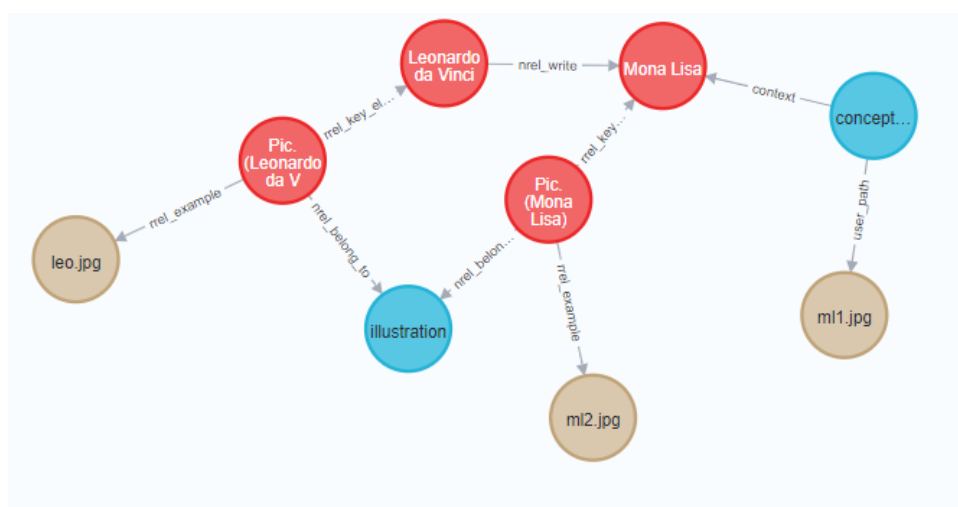


Рисунок 3.9 – Результат работы агента определения наличия изображения в БЗ

### 3.5 Вывод

На этапе разработки был реализован функционал агентов определения наличия изображения в БЗ и определения жанра изображения – составляющих частей РЗ ИСС по искусству, проведено тестирование обоих агентов.

## ЗАКЛЮЧЕНИЕ

В рамках данного курсового проекта был разработан решатель задач интеллектуальной справочной системы по искусству. Личным вкладом в курсовой проект являются разработанные агенты определения жанра изображения и определения наличия изображения в БЗ.

Для разработки РЗ были проанализированы предметная область «Искусство» и была показана актуальность создания РЗ ИСС по искусству. Также были проанализированы уже существующие аналоги справочных систем в тематике обработки изображений и, на основании выделенных в результате этого анализа преимуществ и недостатков, составлены требования для разрабатываемой системы.

Для разработки агента определения наличия изображения в БЗ был изучен и использован алгоритм вычисления перцептивного хэширования.

В результате были решены все сформулированные для достижения цели задачи, а именно:

- были разработаны агенты определения жанра изображения и определения наличия изображения в БЗ как составные части решателя задач интеллектуальной справочной системы по искусству;
- все агенты были объединены в целостный РЗ, выполняющий заявленные функции.

Созданный РЗ позволяет использовать поиск изображения по заданному описанию и генерацию описания, определение жанра и оригинальности заданного изображения и генерация самого изображения по описанию.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Пегов, А.А. Использование современных информационных и коммуникационных технологий в учебном процессе / А.А. Пегов. — Томск.: ТГПУ, 2007. — 71 с.
- [2] Справочный центр Pinterest [Электронный ресурс]. — Режим доступа: <https://help.pinterest.com/ru/guide/all-about-pinterest>. — Дата доступа: 11.09.2023.
- [3] Artlex – Art Dictionary and Encyclopedia [Electronic resource]. — Mode of access: <https://www.artlex.com/>. — Date of access: 25.09.2023.
- [4] Smithsonian libraries [Electronic resource]. — Mode of access: <https://library.si.edu/>. — Date of access: 12.09.2023.
- [5] Российская государственная библиотека искусств [Электронный ресурс]. — Режим доступа: <http://liart.ru/ru/>. — Дата доступа: 05.03.2023.
- [6] Паттерны архитектуры программного обеспечения [Электронный ресурс]. — Режим доступа: <https://systems.education/event-driven-architecture>. — Дата доступа: 16.10.2023.
- [7] Петифорова, Д.Е. Анализ использования перцептивного хеширования в процессе идентификации изображений / Д.Е. Петифорова. — Москва.: РУДН. — 5 с.
- [8] Работа перцептивного хэша [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/articles/120562/>. — Дата доступа: 12.10.2023.
- [9] NEO4J DOCS [Электронный ресурс]. — Режим доступа: <https://neo4j.com/docs/cypher-manual/current/introduction/>. — Дата доступа: 12.10.2023.
- [10] Cypher Manual [Electronic resource]. — Mode of access: <https://neo4j.com/docs/cypher-manual/current/clauses/create/>. — Date of access: 08.10.2023.
- [11] База данных Neo4j Graph: возможности для вашего бизнеса [Электронный ресурс]. — Режим доступа: <https://appmaster.io/ru/blog/baza-dannykh-neo4j>. — Дата доступа: 10.10.2023.
- [12] Neo4j [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Neo4j#:~:text=%D..> — Дата доступа: 11.10.2023.
- [13] Событийно-ориентированная архитектура [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/companies/piter/articles/530514/>. — Дата доступа: 14.10.2023.



[14] Полное руководство по модулю asyncio в Python [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/companies/wunderfund/articles/700474/>. — Дата доступа: 16.10.2023.

[15] OPEN CV в Python [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/articles/519454/>. — Дата доступа: 12.10.2023.