Основы теории систем

**Отчет по лабораторной №3**

«Формирование онтологической модели с помощью инструментов  
онтологического инжиниринга Protege и Neo4g»

Вариант 22

Выполнила:

Студентка 1 курса   
Группы ИВТИИбд-12  
Силаева Софья

Ульяновск,  
2025

Задание на лабораторную работу:

* Сформировать онтологическую модель структуры автомобиля в инструментах онтологического инжиринга Protege и Neo4j
* Провести сравнительный анализ данных инструментов
* Сделать заключение какой же инструмент является более удобным для разработки онтологических моделей

1. Разработка онтологической модели

Нужно разработать онтологическую модель структуры машины.Требования:

* Минимум 3 класса
* у каждого классса хотя бы 1 свойство
* минимум 2 типа связи
* минимум 2 экземпляра каждого класса
* минимум 2 правила логического вывода

Классы:

* Закон
* Статья
* Гражданин

Отношения:

* содержитСтатью
* регулируется

Аксиомы:

* Если Гражданин регулируется Статьей, а Статья содержится в Законе → Гражданин является субъектом Закона
* Если Гражданин регулируется Статьей, а Статья содержится в Законе → Гражданин является субъектом Закона

**1.1 Protege**

Создание классов

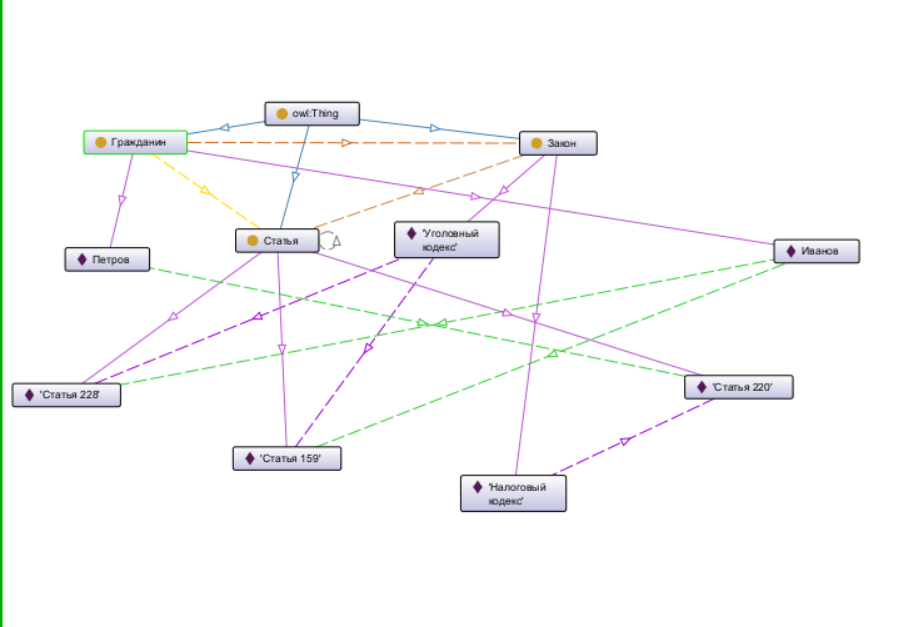
В данной онтологической модели существует 3 класса: «Закон», «Статья», «Гражданин». 

Рисунок 1 – скриншот модели

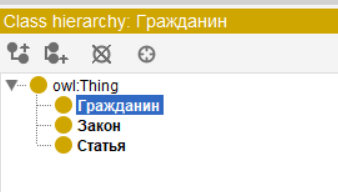


Рисунок 2 – скриншот классов

#### ****Экземпляры классов****

У каждого класса есть свои экземпляры:

**Класс "Закон":**

* Уголовный кодекс РФ
* Налоговый кодекс РФ

**Класс "Статья":**

* Статья 228
* Статья 159
* Статья 220

**Класс "Гражданин":**

* Иванов И.И.
* Петров П.П.

#### ****Создание отношений****

В Protege связи между элементами онтологии определяются с помощью отношений:

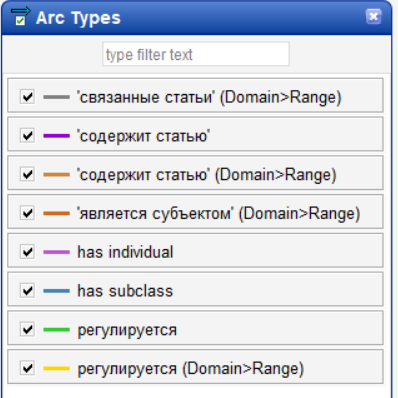


Рисунок 3 – скриншот связей

**Свойства экземпляров**

**Экземпляр "Уголовный кодекс":**

* название: "Уголовный кодекс Российской Федерации"
* содержитСтатью: Статья 228, Статья 159

**Экземпляр "Статья 228":**

* номерСтатьи: "Статья 228"
* текстСтатьи: "Незаконные приобретение, хранение, перевозка наркотических средств"

**Экземпляр "Иванов И.И.":**

* ФИО: "Иванов Иван Иванович"
* регулируется: Статья 228, Статья 159

Эта система позволяет строить логические выводы благодаря встроенному Reasoner'у. Он способен самостоятельно генерировать новые факты на основе заданных правил и ограничений. После активации Reasoner'а автоматически добавляется связь "являетсяСубъектом", упрощая построение модели. Среди преимуществ - интуитивно понятный интерфейс, возможность автоматического вывода данных, наличие плагинов для визуализации и удобство проектирования онтологических моделей. К недостаткам относятся ограничения по объему обрабатываемых данных и сложности интеграции с другими приложениями.

### ****1.2 Neo4j****

Neo4j -- графовая система управления базами данных (СУБД). Использует собственный язык запросов -- Cypher.

#### ****Создание узлов****

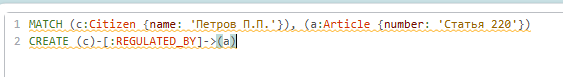


Рисунок 4 - Скриншот запроса создания отношений между узлами



Рисунок 5 - Скриншот запроса создания нового узла

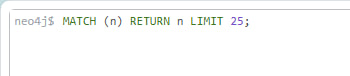


Рисунок 6 - Скриншот запроса вывода всех узлов и связей

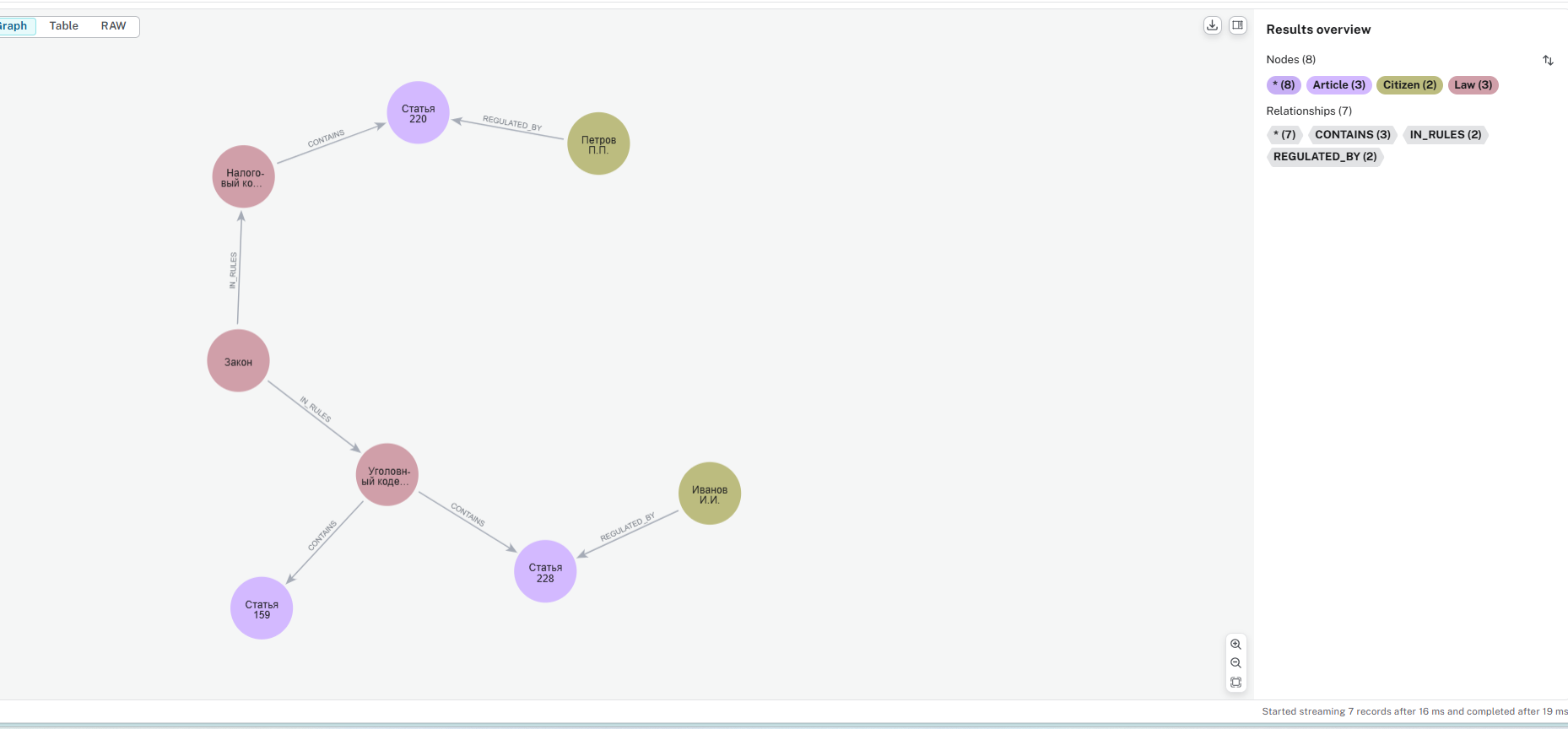


Рисунок 7 – Скриншот графа заданной онтологической модели

В отличие от систем, основанных на классах, Neo4j использует метки (Labels) для группировки узлов. Это позволяет узлам, относящимся к одной группе, обладать различными свойствами, и обеспечивает высокую гибкость, позволяя динамически добавлять узлы и связи. Neo4j оптимизирован для работы с большими объемами данных, отлично интегрируется с приложениями, обеспечивает превосходную визуализацию связей и демонстрирует высокую производительность. Однако, для использования необходимы знания языка Cypher, отсутствует встроенный Reasoner для автоматического логического вывода, и структура данных не такая строгая, как в других системах.

Выбор между Protege и Neo4j зависит от задачи. Если приоритет - строгая структура онтологии, автоматический логический вывод и удобный интерфейс без программирования, а также академические исследования, то Protege - лучший выбор. Если же требуется работа с большим объемом графовых данных, динамическое хранение, интеграция с приложениями и высокая производительность, то Neo4j предпочтительнее. В данной лабораторной работе Protege оказался удобнее благодаря визуальному интерфейсу и возможности реализации логического вывода без написания кода.