Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №3

«Построение онтологической модели»

Вариант №18

Выполнил студент

группы ИВТИИбд-11

Мехтеркин И.А.

Преподаватель

Хайрулин И.Д.

Ульяновск, 2025

### **1. Структурное описание онтологической модели**

Онтологическая модель описывает систему хронологии исторических событий и формализует ключевые объекты и их взаимосвязи.  
Основные сущности (далее — классы) модели включают:

* **События (Event):** объекты, представляющие значимые исторические происшествия, происходящие в определённое время и в определённой стране.
* **Личности (Person):** участники исторических событий, влияющие на их ход и развитие эпохи.
* **Страны (Country):** государства, на территории которых происходят исторические события и формируются политические и культурные процессы.
* **Эпохи (Epoch):** временные периоды, объединяющие события и страны по общим историческим характеристикам и признакам.

Для реализации модели использованы два инструмента:

1. **Protege** — применяется для построения OWL-онтологии, определения классов, их свойств и взаимосвязей, а также для формулирования логических правил с использованием языка SWRL.
2. **Neo4j** — используется для создания графовой модели, в которой исторические объекты представлены узлами, а связи между ними реализованы в виде отношений, формализованных через язык запросов Cypher.

# 2. Перечень классов, отношений и аксиом

Онтологическая модель включает классы (сущности), свойства данных и объектов, а также логические правила, описывающие взаимосвязи между элементами.

## 2.1 Классы (Classes)

Модель содержит четыре основных класса:  
• Event (Событие) — отражает исторические события с указанием года;  
• Person (Личность) — участники событий с данными о рождении;  
• Country (Страна) — место, где произошло событие;  
• Epoch (Эпоха) — исторический период, к которому принадлежит страна и событие.

## 2.2 Свойства данных (Data Properties)

Свойства данных определяют характеристики объектов:  
• name — имя, название объекта (для всех классов);  
• year — год события (Event);  
• born — год рождения (Person);  
• continent — континент (Country);  
• period — временные границы (Epoch).

## 2.3 Свойства объектов (Object Properties)

Object Properties описывают связи между сущностями модели:  
• participatedIn — Person → Event (участвовал в событии);  
• occurredIn — Event → Country (событие произошло в стране);  
• belongsToEpoch — Country → Epoch (страна относится к эпохе);  
• belongsToEpochEvent — Event → Epoch (событие относится к эпохе);  
• livedInEpoch — Person → Epoch (личность жила в эпохе).

## 2.4 Аксиомы (SWRL-правила) для Protege

SWRL-правила формализуют логические зависимости между объектами:  
1. Event(?e) ^ occurredIn(?e, ?c) ^ belongsToEpoch(?c, ?ep) -> belongsToEpochEvent(?e, ?ep)  
 Если событие произошло в стране, и страна относится к эпохе, то событие принадлежит этой эпохе.  
2. Person(?p) ^ participatedIn(?p, ?e) ^ belongsToEpochEvent(?e, ?ep) -> livedInEpoch(?p, ?ep)  
 Если личность участвовала в событии, принадлежащем эпохе, то она жила в этой эпохе.

## 2.5 Аналог аксиом для Neo4j (Cypher-запросы)

MATCH (e:Event)-[:OCCURRED\_IN]->(c:Country)-[:BELONGS\_TO\_EPOCH]->(ep:Epoch)  
MERGE (e)-[:BELONGS\_TO\_EPOCH]->(ep);  
  
MATCH (p:Person)-[:PARTICIPATED\_IN]->(e:Event)-[:BELONGS\_TO\_EPOCH]->(ep:Epoch)  
MERGE (p)-[:LIVED\_IN\_EPOCH]->(ep);

# 3. Реализация в Protege

**Определение классов.** Вкладка Classes — созданы четыре класса: Event, Person, Country, Epoch.

**Создание свойств данных.** Вкладка Data Properties — созданы свойства:

name (xsd:string) — общая метка для всех сущностей;

year (xsd:integer) — для Event;

born (xsd:integer) — для Person;

continent (xsd:string) — для Country;

period (xsd:string) — для Epoch.  
Для каждого свойства указаны соответствующие domain/range.

**Создание объектных свойств.** Вкладка Object Properties — добавлены связи:

participatedIn (Domain: Person, Range: Event);

occurredIn (Domain: Event, Range: Country);

belongsToEpoch (Domain: Country, Range: Epoch);

belongsToEpochEvent (Domain: Event, Range: Epoch);

livedInEpoch (Domain: Person, Range: Epoch).  
Для важных связей заданы domain и range, что улучшает проверку согласованности.

**Добавление экземпляров (Individuals).** Вкладка Individuals — созданы по два индивида для каждого класса (например, French\_Revolution, Christianization\_of\_Rus; Napoleon\_Bonaparte, Prince\_Vladimir; France, Kievan\_Rus; Modern\_Era, Middle\_Ages) и заполнены их data property assertions (name, year, born, continent, period).

**Установка объектных утверждений.** Для индивидов заданы object property assertions:

Napoleon\_Bonaparte participatedIn French\_Revolution;

Prince\_Vladimir participatedIn Christianization\_of\_Rus;

French\_Revolution occurredIn France;

Christianization\_of\_Rus occurredIn Kievan\_Rus;

France belongsToEpoch Modern\_Era;

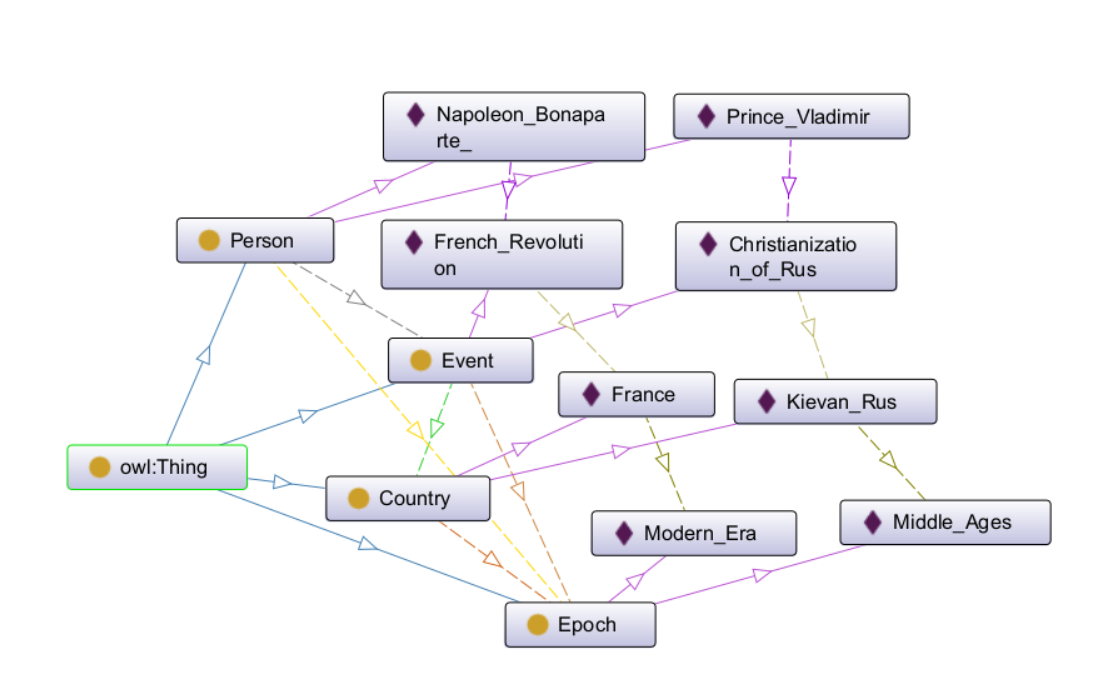
Kievan\_Rus belongsToEpoch Middle\_Ages.

**Формулирование SWRL-правил. Через вкладку SWRLTab добавлены правила в ASCII-синтаксисе:**

Event(?e) ^ occurredIn(?e, ?c) ^ belongsToEpoch(?c, ?ep) -> belongsToEpochEvent(?e, ?ep)

Person(?p) ^ participatedIn(?p, ?e) ^ belongsToEpochEvent(?e, ?ep) -> livedInEpoch(?p, ?ep)  
Правила корректно сопоставляют имена классов и свойств, чувствительных к регистру.

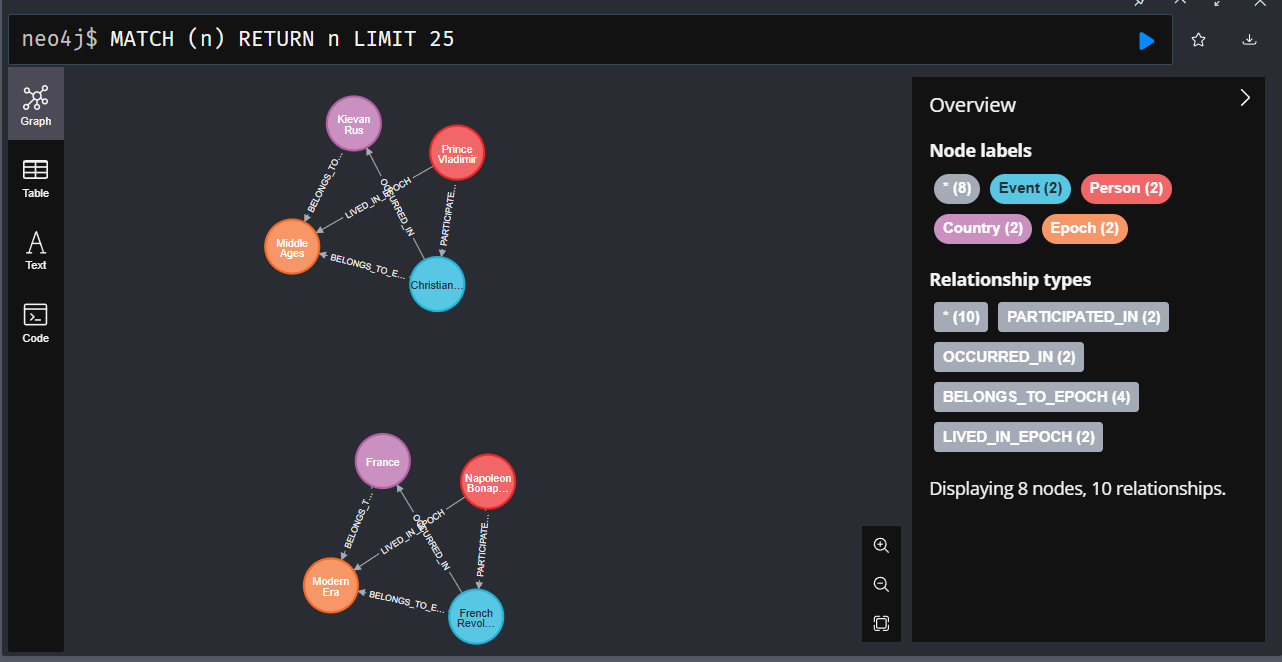
Итоговый граф:



# 4. Реализация в Neo4j

В Neo4j модель реализована как граф, где узлы представляют объекты (события, личности, страны, эпохи), а связи — отношения между ними. Пример Cypher-кода для создания модели:

MATCH (n) DETACH DELETE n;  
  
CREATE  
(e1:Event {name:'French Revolution', year:1789}),  
(e2:Event {name:'Christianization of Rus', year:988}),  
(p1:Person {name:'Napoleon Bonaparte', born:1769}),  
(p2:Person {name:'Prince Vladimir', born:958}),  
(c1:Country {name:'France', continent:'Europe'}),  
(c2:Country {name:'Kievan Rus', continent:'Eurasia'}),  
(ep1:Epoch {name:'Middle Ages', period:'5th–15th centuries'}),  
(ep2:Epoch {name:'Modern Era', period:'15th–19th centuries'});  
  
MATCH (e1:Event {name:'French Revolution'}), (e2:Event {name:'Christianization of Rus'}),  
(p1:Person {name:'Napoleon Bonaparte'}), (p2:Person {name:'Prince Vladimir'}),  
(c1:Country {name:'France'}), (c2:Country {name:'Kievan Rus'}),  
(ep1:Epoch {name:'Middle Ages'}), (ep2:Epoch {name:'Modern Era'})  
CREATE  
(p1)-[:PARTICIPATED\_IN]->(e1),  
(p2)-[:PARTICIPATED\_IN]->(e2),  
(e1)-[:OCCURRED\_IN]->(c1),  
(e2)-[:OCCURRED\_IN]->(c2),  
(c1)-[:BELONGS\_TO\_EPOCH]->(ep2),  
(c2)-[:BELONGS\_TO\_EPOCH]→(ep1);

****

# 5. Вывод

В ходе выполнения лабораторной работы была построена онтологическая модель исторических событий, включающая четыре класса: Event, Person, Country и Epoch. SWRL-правила и Cypher-запросы обеспечили автоматический вывод новых связей между объектами, что подтвердило корректность и логическую согласованность модели. Модель успешно реализована как в Protege, так и в Neo4j, демонстрируя возможности онтологического моделирования и графовых баз данных для представления знаний о хронологии исторических процессов.