

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматики
Кафедра комп'ютерних наук

Бакалаврська кваліфікаційна робота на тему:

РОЗРОБКА ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ “МУЗИКА ТА КІНО” ЧАСТИНА 2.
ОНТОЛОГІЯ КІНЕМАТОГРАФІЇ

Виконав: студент 4 курсу, групи 1КН-216
спеціальності 122 – Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)

Самохвал А.С.

Керівник: к.ф.-м.н., доцент каф. КН

Сілагін О.В.

“ 04 ” “ червень ” 2025 р.

Рецензент: к.т.н., доцент каф. АІТ

Кривогубченко С.Г.

“ 05 ” “ червень ” 2025 р.

Допущено до захисту
Зав. кафедри КН Григор'єв А.А.
“ 06 ” “ червень ” 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський) Галузь
знань – 12 Інформаційні технології Спеціальність –
122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН д.т.н.,
проф. Яровий А. А.

“05” “травня” 2025 р.



ЗАВДАННЯ

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Самохвалу Антону Сергійовичу

1. Тема роботи: Розробка онтологічної бази знань “Музика та кіно”. Частина 2.
Онтологія кінематографії
Керівник роботи: Сілагін Олексій Віталійович, к.ф.-м.н., доц.
Затверджені наказом вищого навчального закладу “20” “березня” 2025 року № 97
2. Термін подання студентом роботи 30 “травня” 2025 року
3. Вихідні дані до роботи: Об’єктно-орієнтована парадигма програмування; об’єктно-орієнтовані бази знань; моделі представлення знань – семантичні та фреймові мережі; передбачити можливість застосування запитів до бази знань; кількість основних класів – не менше 7 одиниць; кількість підкласів – не менше 4 одиниць; кількість індивідів – не менше 20 одиниць; ступінь ієрархії – не менше 1 гілки.
4. Зміст текстової частини: Вступ, обґрунтування доцільності розробки онтологічної бази знань “Онтологія кінематографії”, моделювання та проєктування онтологічної бази знань “Онтологія кінематографії”, програмна реалізація онтологічної бази знань “Онтологія кінематографії”, тестування онтологічної бази знань “Онтологія кінематографії”, висновки, перелік використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов’язкових креслень): аналоги, створення об’єктних класів, об’єктних властивостей, властивостей даних, індивідів, Ontograf класів та підкласів та індивідів, структура розширення властивостей редактора Protégé із застосуванням Preferences, результати тестування SPARQL запитів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Сілагін О.В., к.ф.-м.н, доцент каф. КН		

7. Дата видачі завдання 05 травня 2025 року

КАЛЕНДАРНИЙ ПЛАН

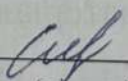
№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної галузі. Огляд чинних систем аналогів. Постановка задач дослідження	05.05.25	07.05.25	
2	Моделювання та проектування онтологічної бази знань "Онтологія кінематографії"	11.05.25	14.05.25	
3	Реалізація онтологічної бази знань "Онтологія кінематографії"	17.05.25	22.05.25	
4	Тестування онтологічної бази знань "Онтологія кінематографії"	23.05.25	24.05.25	
5	Оформлення матеріалів до захисту БКР	30.05.25	04.06.25	

Студент


(підпис)

Самохвал А.С

Керівник роботи


(підпис)

Сілагін О.В

АНОТАЦІЯ

Бакалаврська дипломна робота зосереджена на розробці онтологічної бази знань стосовно теми “Кінематографічна онтологія”, зосереджуючись на формуванні кінематографічної онтології. Ключовою задачею дослідження є формалізація та організація знань про кіно як комплексну культурну, художню та технічну сферу, а також взаємозв'язків між фільмами, жанрами, виконавцями, режисерами, музичним супроводом та іншими елементами кіномистецтва та музичної індустрії.

В ході роботи було проведено аналіз предметної області, виявлено ключові поняття та встановлено їх взаємозв'язки. Для реалізації бази знань використано онтологічний підхід з застосуванням мови OWL (Web Ontology Language) та програмного середовища Protégé. Створена онтологія містить в собі базові класи (Фільм, Жанр, Особа, Назва музичного твору, Композитор/Гурт, Нагорода, Роль), їхні властивості та відношення між ними, що дає змогу забезпечити повноцінну семантичну модель інформації про кіно.

Ключові слова: онтологія, база знань, музика, кіно, фільм, жанр, актор, OWL, Protégé.

ABSTRACT

The bachelor's thesis focuses on the development of an ontological knowledge base on the topic "Cinematic onthology" with a particular emphasis on constructing a cinematographic ontology. The key objective of the study is to formalize and organize knowledge about cinema as a complex cultural, artistic, and technical domain, as well as to map the interrelations between films, genres, performers, directors, musical scores, and other elements of cinematic art and the music industry.

During the course of the work, the subject area was analyzed, key concepts were identified, and their relationships were defined. An ontological approach was employed for the implementation of the knowledge base, utilizing the OWL (Web Ontology Language) and the Protégé software environment. The resulting ontology includes core classes (Film, Genre, Person, Musical Work Title, Composer/Band, Award, Role), their properties, and relationships between them, enabling the creation of a comprehensive semantic model of information about cinema.

Keywords: ontology, knowledge base, music, cinema, film, genre, actor, OWL, Protégé

ЗМІСТ

ВСТУП.....	5
1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ СТВОРЕННЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ ПРЕДМЕТНОЇ ОБЛАСТІ “КІНЕМАТОГРАФІЧНА ОНТОЛОГІЯ”	7
1.1 Передумови створення бази знань “Кінематографічна онтологія”	7
1.2 Аналіз відомих реалізацій баз знань в предметній області “Кінематографічна онтологія”	11
1.3 Постановка задач дослідження.....	15
1.4 Висновок до розділу 1	16
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ БАЗИ ЗНАНЬ У ПРЕДМЕТНІЙ ОБЛАСТІ “КІНЕМАТОГРАФІЧНА ОНТОЛОГІЯ”	18
2.1 Визначення межі та масштабу онтологій.....	18
2.2 Створення термінологічного словника онтології	20
2.3 Моделювання онтології засобами Protégé.	23
2.4 Висновок до розділу 2	33
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ “КІНЕМАТОГРАФІЧНА ОНТОЛОГІЯ” В СЕРЕДОВИЩІ PROTEGE	34
3.1 Застосування Preferences в середовищі Protégé	34
3.2 Створення SPARQL запитів.....	38
3.3 Тестування онтологічної бази знань “Кінематографічна онтологія”	42
3.4 Висновок до розділу 3	47
ВИСНОВКИ	49
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК А(обов’язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ.....	54
ДОДАТОК Б (обов’язковий) ЛІСТИНГ ПРОГРАМИ	55
ДОДАТОК В (обов’язковий) ІЛЮСТРАТИВНА ЧАСТИНА	74
ДОДАТОК Г (довідниковий) ІНСТРУКЦІЯ КОРИСТУВАЧА.....	77

ВСТУП

Актуальність теми

У сучасному світі кількість інформації про культуру, зокрема про музику й кіно, постійно збільшується. Щодоби виходять нові кінострічки, музичні альбоми, з'являються нові виконавці, жанри, саундтреки, фестивалі й нагороди. Усе це формує складну інформаційну структуру, яку складно оперативно аналізувати та розуміти без належних технологій.

Традиційні підходи до пошуку культурної інформації — бази даних, сайти, вікі-платформи — не забезпечують повного семантичного опрацювання даних та не виявляють прихованих взаємозв'язків між елементами. Через це дедалі більшу популярність здобуває онтологічне моделювання, що дозволяє формалізовано та гнучко подати знання про об'єкти і явища культурної сфери.

Особливої уваги потребує кінематограф[1] як надзвичайно динамічна та багатогранна галузь. Оскільки фільми часто пов'язані з іншими аспектами культури (музикою, літературою, історією тощо), створення онтології на стику музики й кіно є важливим та актуальним завданням. Така онтологія дасть змогу структурувати інформацію про фільми, жанри, акторів, композиторів, саундтреки, нагороди, а також зв'язки між ними, що значно полегшить доступ до знань та дозволить реалізовувати інтелектуальні сервіси — від пошуку до рекомендаційних систем.

Мета і завдання дослідження

Метою роботи є розширення функціональних можливостей програмного засобу на основі онтологічної бази знань.

Завданнями дослідження є:

- Аналізовано предметну галузь кінематографу. Оглянуто чинні системи аналогів
- Змодельовано та спроектувано онтологічну базу знань “Кінематографічна онтологія”
- Реалізовано онтологічну базу знань “Кінематографічна онтологія”
- Протестовано онтологічну базу знань “Кінематографічна онтологія”
- Оформлено пояснювальну записку, графічний матеріал та презентацію
- Оцінено перспективи розширення й інтеграції в інформаційні системи

Об'єкт і предмет дослідження

Об'єктом дослідження є процес структурування знань у сфері кінематографу.

Предметом дослідження є програмне забезпечення по розробці бази знань.

Практичне значення

Практичне значення отриманих результатів полягає у можливості використання онтології:

- у навчальних системах для вивчення культури та мистецтва;
- у системах інтелектуального пошуку та фільтрації фільмів і музики;
- у рекомендаційних сервісах стримінгових платформ;
- у бібліотеках та цифрових архівах для семантичної каталогізації медіаресурсів.

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ СТВОРЕННЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ ПРЕДМЕТНОЇ ОБЛАСТІ “КІНЕМАТОГРАФІЧНА ОНТОЛОГІЯ”

1.1 Передумови створення бази знань “Кінематографічна онтологія”

Інформаційне перенасичення у царині кінематографу

Сучасна кіноіндустрія демонструє стрімке збільшення обсягів інформації. Щорічно на різних платформах і в кінотеатрах з'являються тисячі нових стрічок, серіалів та інших відеоматеріалів. Кожен із них супроводжується великою кількістю супровідних даних: відомостями про акторів, режисерів, сценаристів, композиторів, продюсерів, технічні характеристики, бюджет, касові збори, нагороди, рецензії та глядацькі відгуки. Звичайні реляційні бази даних не можуть забезпечити ефективне представлення складних взаємозв'язків між цими компонентами, що ускладнює пошук інформації та призводить до її фрагментації.

Обмеження наявних систем класифікації кінопродукції

Сучасні бази даних та платформи, що надають інформацію про кіно мають суттєві недоліки:

1. Відсутність глибинних семантичних зв'язків між окремими елементами.
2. Обмеженість при обробці складних аналітичних запитів.
3. Неможливість виявлення прихованих взаємозв'язків між об'єктами.
4. Недостатня адаптація до новітніх форматів відеоконтенту.
5. Відсутність чіткої ієрархії понять та взаємовідносин у галузі.
6. Потреба в об'єднанні неоднорідних джерел даних.

Експоненційне зростання кількості медіаконтенту призводить до феномену "інформаційного перевантаження" (information overload), який уперше описав Алвін Тоффлер у 1970 році.[2] У контексті кіноіндустрії це проявляється у:

- Фрагментації інформації: дані про один фільм можуть бути розпорошені по десятках різних джерел
- Неузгодженості метаданих: різні платформи використовують відмінні системи класифікації та опису
- Складності верифікації: відсутність єдиних стандартів ускладнює перевірку достовірності інформації

- Проблемах інтеграції: складність об'єднання даних з різних джерел через відмінності у структурі та форматах

Дані про кіно розпорошені по багатьох джерелах: офіційні сайти, стрімінгові сервіси, професійні та наукові ресурси, соціальні мережі. Щоб ефективно їх інтегрувати, необхідна спільна концептуальна структура, що об'єднуватиме терміни та встановлюватиме узгоджені зв'язки між ними.

Стан розвитку семантичних технологій та онтологічного моделювання

Найновіші досягнення у сфері Semantic Web[3, 4] та онтологічної інженерії[5] створюють передумови для побудови передових систем знань. Технології RDF[6], OWL[7] і мова запитів SPARQL[8] дозволяють формалізувати знання про сферу кіно у вигляді онтології — впорядкованої системи понять та відносин між ними.

Зростання попиту на індивідуалізовані рекомендаційні механізми

Користувачі очікують від платформ персоналізованих рекомендацій. Ринок рекомендаційних систем оцінюється у \$15.13 млрд у 2023 році з прогнозованим зростанням до \$65.23 млрд до 2030 року[9, 10]. Недоліки стандартних методів, таких як колаборативна фільтрація, можуть бути подолані завдяки впровадженню онтологічного підходу.

Обмеження традиційних рекомендаційних систем:

1. Проблема холодного старту: неможливість рекомендацій для нових користувачів або контенту
2. Фільтраційні бульбашки: схильність до рекомендацій схожого контенту
3. Недостатня інтерпретованість: неможливість пояснити, чому запропоновано конкретний контент
4. Ігнорування контексту: не враховування ситуаційних факторів (настрій, час, компанія)

Переваги онтологічного підходу:

- Семантичні зв'язки: врахування змістовних відносин між характеристиками фільмів
- Пояснюваність: можливість надання логічного обґрунтування рекомендацій
- Гнучкість: легке додавання нових критеріїв та правил
- Контекстуальність: врахування різноманітних факторів впливу

Потреба у підтримці наукових досліджень у галузі кіно

Науковці потребують інструментів для глибинного аналізу кіно як мистецтва та соціокультурного явища. Онтологія може стати базою для вивчення еволюції жанрів, стилів, впливу авторів тощо — завдяки формалізованому опису предметної області.

Необхідність семантичного аналізу відеоконтенту

Науковці потребують інструментів для глибинного аналізу кіно як мистецтва та соціокультурного явища. Кіномистецтвознавство як академічна дисципліна включає:

Історичні дослідження:

- Еволюція кінематографічних технік та стилів
- Вплив соціально-політичних факторів на розвиток кіно
- Порівняльний аналіз національних кінематографічних шкіл
- Дослідження творчих біографій видатних діячів кіно

Жанрові студії:

- Аналіз формування та трансформації кінематографічних жанрів
- Дослідження гібридних та експериментальних форм
- Вивчення впливу культурних контекстів на жанрову специфіку

Авторські дослідження:

- Аналіз творчих методів видатних режисерів
- Дослідження еволюції авторського стилю
- Порівняльні студії творчості різних митців

Соціокультурні аспекти:

- Вплив кіно на формування суспільної свідомості
- Репрезентація соціальних груп у кінематографі
- Кіно як засіб культурної дипломатії

Онтологія може стати базою для вивчення еволюції жанрів, стилів, впливу авторів тощо — завдяки формалізованому опису предметної області.

Системи автоматичного розпізнавання сцен, облич або об'єктів потребують контексту для повної інтерпретації. Розвиток технологій комп'ютерного зору та машинного навчання створює нові можливості:

Автоматичне розпізнавання:

- Ідентифікація акторів у кадрі
- Розпізнавання емоцій та настроїв

- Аналіз композиції та колірної гами
- Виявлення об'єктів та локацій

Семантична інтерпретація:

- Розуміння жанрових особливостей
- Ідентифікація стилістичних прийомів
- Визначення тематичних елементів
- Аналіз наративних структур

Онтологія забезпечує семантичну основу для осмислення результатів аналізу та їх прив'язки до знань у галузі кіно.

Розвиток VR, AR, інтерактивного та трансмедійного контенту порушує усталені межі кіно. Онтологічна модель має гнучко адаптуватися до цих новацій, дозволяючи точно описувати нові види творчого продукту.

Кінематограф як частина культурної спадщини

Фільми є цінними культурними артефактами. Систематизоване знання про національні школи, культових творців і стрічки має велике значення для збереження культурної пам'яті. Онтологія забезпечує це у вигляді структурованої інформації.

Економічна доцільність

Кіно є потужною галуззю економіки. Вдосконалення механізмів пошуку, аналізу та дистрибуції контенту може підвищити рентабельність, сприяти точнішому маркетингу та монетизації.

Поняття онтологічної бази знань та інженерії знань

Онтологічна база знань — це формалізоване представлення предметної області через концепти, їх атрибути та взаємозв'язки. Вона дає змогу не лише зберігати інформацію, але й робити логічні висновки на її основі. У контексті кіно це — модель, що охоплює фільми, творців, жанри, стилі, технічні деталі та інші елементи.

Інженерія знань — галузь штучного інтелекту, яка займається методами структурування експертної інформації у цифровому вигляді. У сфері кіно вона допомагає ідентифікувати ключові поняття, визначати між ними зв'язки, будувати аксіоми та правила.

Чому потрібна онтологічна база знань "Кінематографічна онтологія"

Створення такої бази дозволить:

1. Об'єднати розпорошені дані в єдину семантичну систему.

2. Забезпечити складний пошук та навігацію на основі змістових зв'язків.
3. Автоматизувати класифікацію фільмів за жанром, стилем тощо.
4. Побудувати рекомендаційні системи нового покоління.
5. Надати інструменти для наукового аналізу розвитку кіно.
6. Сприяти збереженню культурної спадщини.
7. Бути адаптивною до нових форматів контенту та змін у галузі.

1.2 Аналіз відомих реалізацій баз знань в предметній області “Кінематографічна онтологія”

У сучасному світі інформаційні технології відіграють ключову роль у зберіганні, аналізі та обробці даних. У сфері кінематографу створено низку баз знань, що забезпечують доступ до великої кількості структурованої інформації про фільми, серіали, акторів, режисерів, сценаристів та інші елементи кіноіндустрії.

Нижче розглянемо найпоширеніші та найвідоміші бази знань, які активно використовують кінолюбби, дослідники та професіонали індустрії.

1. Internet Movie Database (IMDb)

IMDb[11] залишається найбільшою та найавторитетнішою базою даних про кіно та телебачення з часу свого заснування у 1990 році. Станом на 2024 рік платформа містить інформацію про понад 12 мільйонів фільмів та серіалів, 134 мільйони зареєстрованих користувачів та обробляє понад 250 мільйонів унікальних відвідувачів щомісячно. На рисунку 1.1 зображено головну сторінку IMDb:

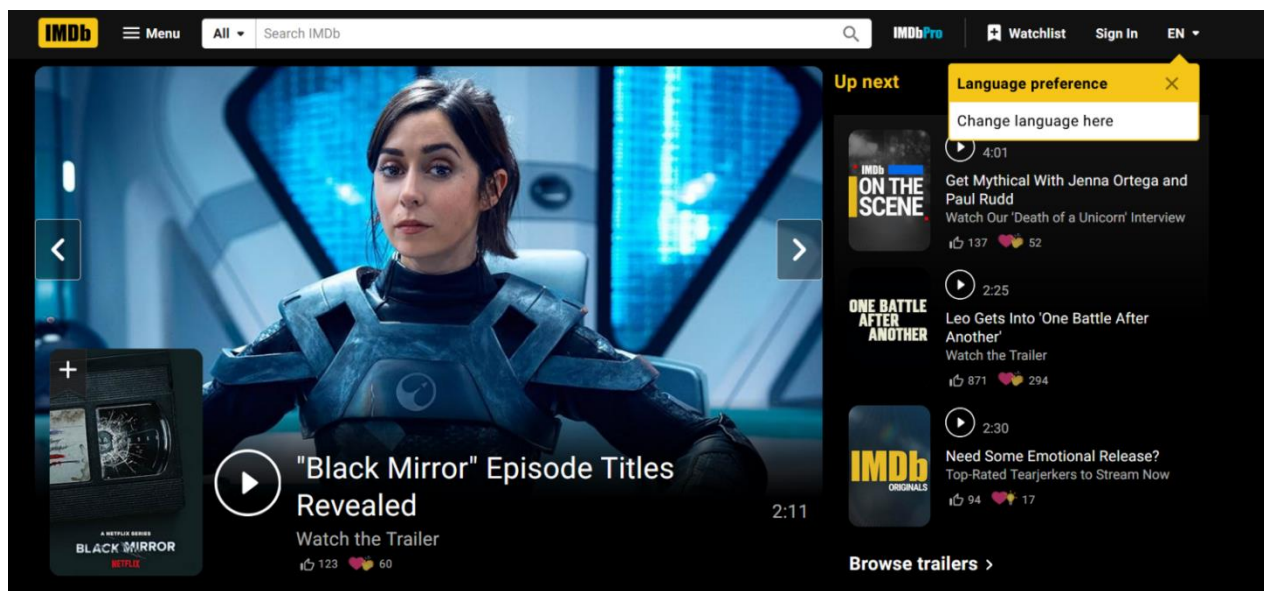


Рисунок 1.1 – Інтерфейс вебсайту IMDb

Основні характеристики IMDb:

Каталог містить інформацію про величезну кількість кінострічок та серіалів різних років.

Користувачі мають змогу додавати рецензії, оцінювати фільми та редагувати інформацію (після перевірки модераторами).

IMDb має систему рейтингів, яка формується на основі оцінок користувачів і використовується як орієнтир для кінолюбителів.

Інтегровано з іншими сервісами, зокрема, потоковими платформами, що полегшує доступ до перегляду фільмів.

База даних підтримує пошук за жанрами, ключовими словами, акторами та іншими параметрами.

IMDb використовується не лише кінолюбителями, але й професіоналами кіноіндустрії, оскільки платформа дозволяє відстежувати інформацію про майбутні релізи та оновлення у сфері кіновиробництва.

2. DzygaMDB

DzygaMDB[12] – це українська база знань, присвячена національному кінематографу. Її мета – зібрати та поширити інформацію про українські фільми, серіали, акторів та режисерів, популяризуючи українське кіно. На рисунку 1.2 зображено головну сторінку DzygaMDB:

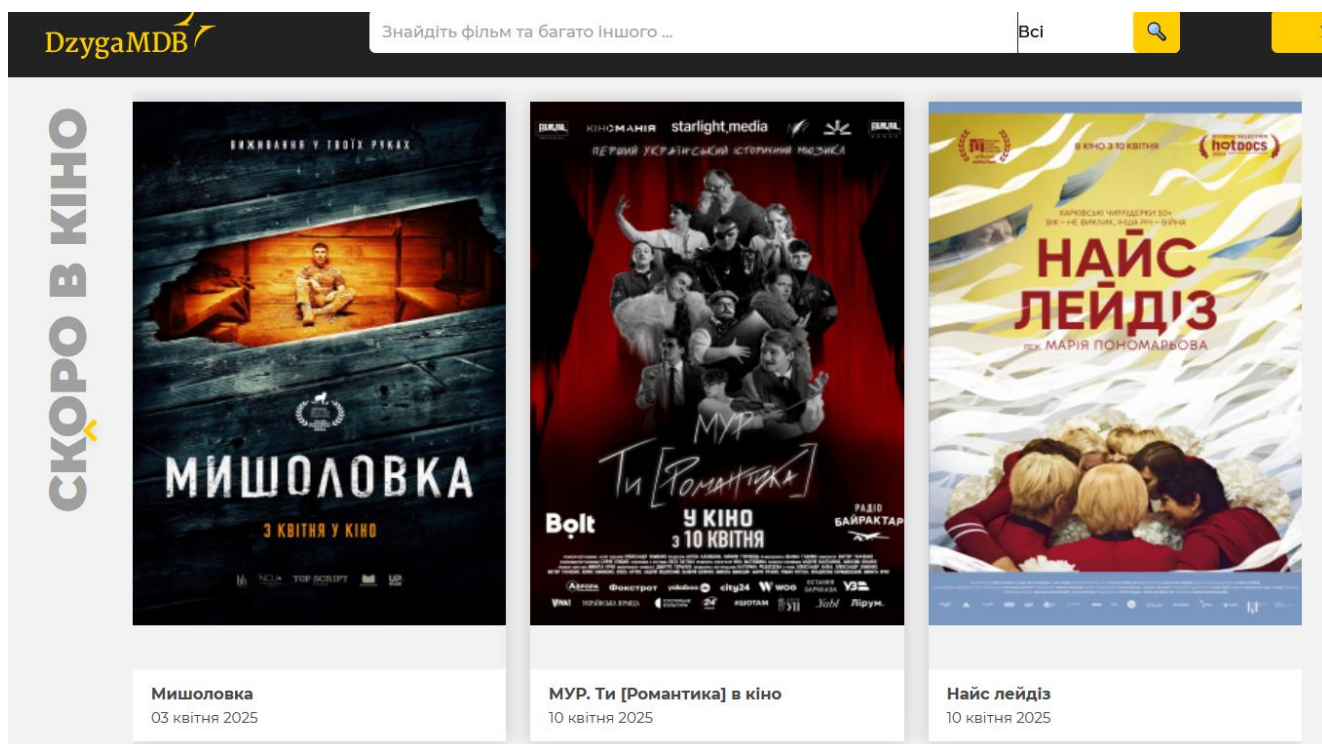


Рисунок 1.2 – Інтерфейс вебсайту DzygaMDB

Особливості DzygaMDB:

Каталог містить детальну інформацію про понад 5 000 українських кінопроектів.

Платформа містить біографії та фільмографії українських кінематографістів.

DzygaMDB доступна українською та англійською мовами, що дозволяє міжнародним користувачам ознайомитися з українським кінематографом.

Охоплює як класичне, так і сучасне кіно, а також незалежні кінопроекти.

Підтримує співпрацю з українськими кіноархівами, що дає змогу оновлювати інформацію про історичні фільми.

База DzygaMDB допомагає розвивати та популяризувати українське кіно як у межах країни, так і на міжнародному рівні.

3. TMDb (The Movie Database)

The Movie Database (TMDb)[13] – це популярна відкрита база даних про кіно та телебачення, що надає структуровану інформацію про фільми, серіали, акторів і творців контенту. Вона є головною альтернативою IMDb, зосереджуючись на відкритості та підтримці API для інтеграції з іншими сервісами. На рисунку 1.3 зображено головну сторінку The Movie Database:

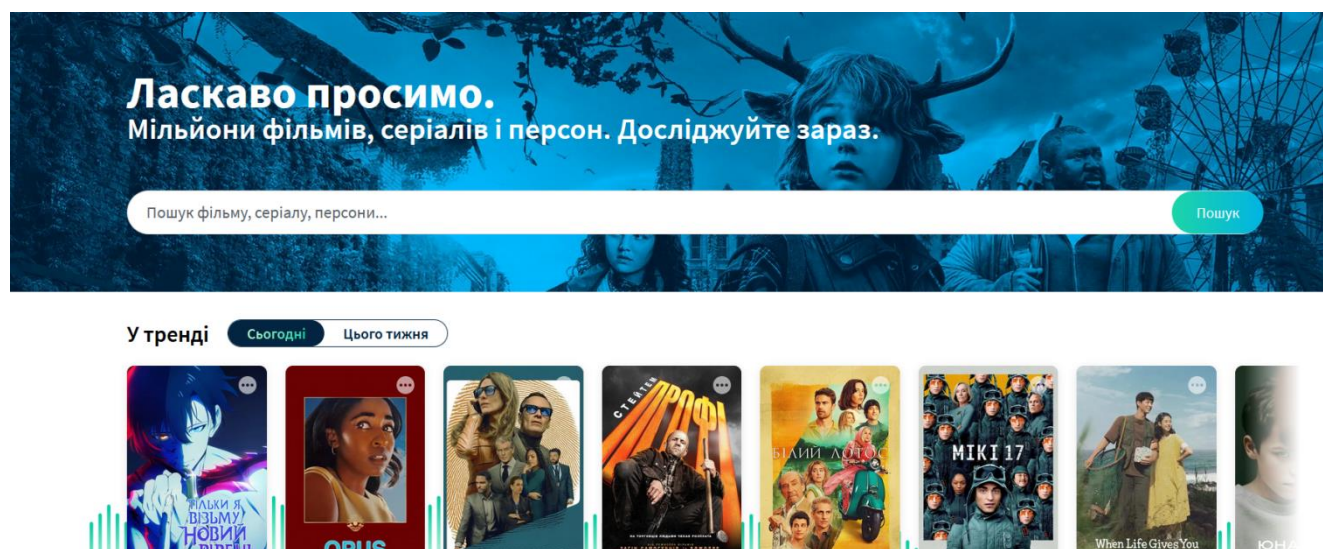


Рисунок 1.3 – Інтерфейс вебсайту DzygaMDB

Основні можливості TMDb:

Каталог містить інформацію про фільми, серіали, трейлери, постери та акторський склад.

Платформа є відкритою, і користувачі можуть вносити правки та додавати нові дані.

TMDB підтримує потужний API, який використовують багато додатків та потокових платформ.

Офіційно підтримує інтеграцію з медіаплеєрами та сервісами, такими як Plex та Kodi.

Містить детальну статистику про касові збори, рейтинги та премії фільмів.

TMDB є відмінною базою знань для розробників і користувачів, які хочуть отримати доступ до відкритих даних про кінематограф.

4. TheTVDB

TheTVDB[14] – це база знань, спеціалізована на телесеріалах. Вона містить інформацію про тисячі телевізійних шоу, їхні епізоди, сезони та деталі виробництва. На рисунку 1.4 зображено головну сторінку TheTVDB:

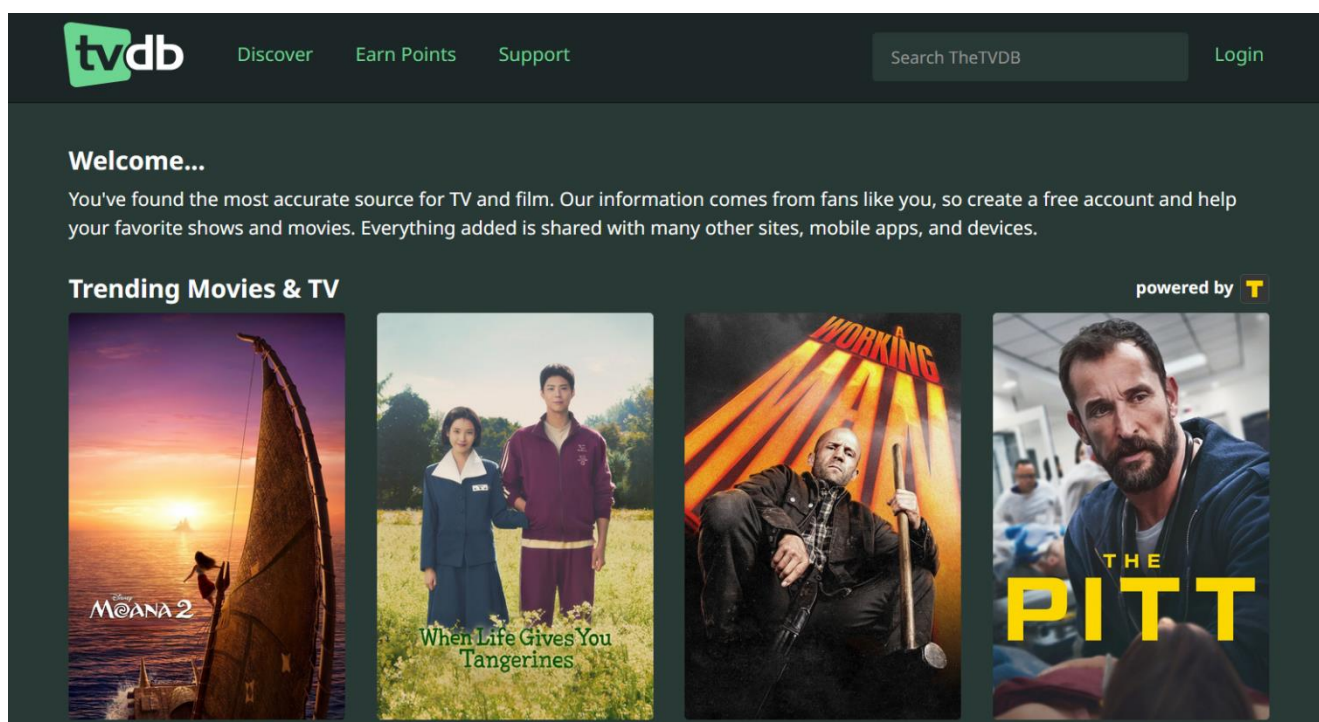


Рисунок 1.4 – Інтерфейс вебсайту TheTVDB

Ключові характеристики TheTVDB:

Каталог містить серіали з усього світу, включаючи анімаційні проєкти та реаліті-шоу.

Можна отримати детальну інформацію про окремі епізоди, їх тривалість, акторський склад та режисерів.

TheTVDB підтримує API, що дозволяє інтегрувати базу з іншими сервісами та застосунками.

Користувачі можуть додавати нові серіали та оновлювати інформацію.

Є функція персоналізованого перегляду, яка дає змогу створювати списки улюблених серіалів.

TheTVDB є корисною платформою як для простих глядачів, так і для розробників, які хочуть інтегрувати дані про телесеріали у власні проекти.

Різні джерела знань у царині кіно вирізняються власними підходами та методами організації даних. IMDb — це найбільша платформа, що акумулює кіноінформацію, тоді як DzygaMDB виконує важливу місію з популяризації українського кіно. TMDb — відкритий аналог IMDb із розширеним API для взаємодії з іншими сервісами, а TheTVDB спеціалізується на структурі телевізійних серіалів.

1.3 Постановка задач дослідження

Сучасна кіноіндустрія міцно завоювала популярність, привертаючи увагу як пересічних глядачів, так і професійних дослідників. Постійні кінопрем'єри, розширення стрімінгових платформ, аналітика кінокритиків, а також створення додаткових медіапродуктів (від коміксів до серіалів) сприяють глобалізації та широкому розповсюдженню кінематографу серед широкої аудиторії. Кіно – це не просто розважальний контент, а цілий культурний феномен, що формує смаки та впливає на суспільство. Кіноіндустрія являє собою багатогранну сферу, яка охоплює величезну кількість фільмів, жанрів, режисерських стилів і творчих підходів. Вона приваблює не лише кінофанів, а й професіоналів, які займаються аналізом кіномистецтва, дослідженням жанрів, впливом фільмів на культуру та соціум. Регулярні випуски нових кінострічок, режисерські версії, кінематографічні всесвіти та кінофестивалі підтримують глядацький інтерес та дозволяють постійно відкривати нові грані кіно. Важливим аспектом є доступність різноманітних ресурсів для дослідження фільмів, що сприяє підтриманню інтересу до кіноіндустрії.

Враховуючи величезний обсяг інформації про фільми, режисерів, акторів та нагороди, виникає потреба в ефективній системі зберігання, структурування та візуалізації цих даних. Онтологічний підхід є надзвичайно корисним в цьому контексті, оскільки дозволяє будувати логічні структури для відображення взаємозв'язків між різними елементами кінематографу. Онтології є потужним інструментом в інформаційних технологіях, що дозволяє систематизувати знання та покращити їх

обробку. Завдяки онтологічному підходу комп'ютерні системи здатні краще розуміти складні взаємозв'язки між об'єктами. У сфері кінематографу це може бути корисним для створення бази знань, що об'єднуватиме відомості про фільми, жанри, акторів, нагороди, кінокомпанії, режисерів і критичні оцінки.

Така система дозволить користувачам легше орієнтуватися у великій кількості фільмів, а також сприятиме стандартизації даних і покращенню взаємодії між різними платформами. Онтологія може бути використана для побудови рекомендаційних систем, автоматизованого аналізу кіно та інтеграції з іншими джерелами даних. Використання онтологічного підходу дає можливість глибше зрозуміти структуру кіноіндустрії, сприяючи розробці інтелектуальних систем, що полегшать взаємодію користувачів із великим обсягом даних і забезпечать зв'язність між ними.

Метою цієї роботи є організація, класифікація та управління великим обсягом даних у сфері кінематографу, а також розробка інформаційної технології, що забезпечить зручний доступ до цієї інформації. Розроблена онтологія міститиме класи фільмів, жанрів, акторів, режисерів, продюсерів, сценаристів, композиторів або гуртів, саундтреки, ролі акторів, нагород, кінокомпаній та їх взаємозв'язки. Дотримання правильної структури онтології скоротить час, необхідний для пошуку інформації, та спростить аналіз. Користувачі зможуть отримувати потрібні дані на основі запитів, що забезпечить зручність використання розробленої інформаційної системи.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

1. Проаналізувати наявну інформацію про кіноіндустрію.
2. Побудувати та систематизувати дані про фільми, жанри, режисерів, акторів та композитора або гурта.
3. Спроекувати моделі інформаційної технології для онтологічного моделювання бази знань про кіно.
4. Розробити структуру онтології та визначити відносини між її елементами.
5. Провести тестування онтології на основі запитів користувачів.

Розробка онтологічної бази знань для кінематографу сприятиме впорядкуванню даних та створенню ефективної системи доступу до інформації, що є важливим кроком для аналітики, досліджень та покращення користувацького досвіду.

1.4 Висновок до розділу 1

У першому розділі було розгорнуто детальний аналіз кіноіндустрії. Обґрунтовано актуальність розробки інформаційної технології онтологічного моделювання для бази знань про кіно. Застосування такої технології здатне значно спростити користувачам оперативний доступ до потрібної інформації про фільми, жанри, акторів та режисерів. В процесі огляду предметної галузі було виявлено проблемні аспекти в чинних аналогах баз знань та визначено ціль їхнього виправлення в рамках даної роботи.

Аналіз інформаційних джерел виявив присутність деяких відомих баз знань про кіно, реалізованих у вигляді онлайн-ресурсів, зокрема: IMDb, DzygaDB, TMDb, TVDB, присвячені кінематографу. Описано їхні сильні та слабкі сторони. З'ясовано, що хоча ці ресурси акумулюють великий обсяг інформації, вони не завжди надають достатньо структурований підхід до зв'язків між об'єктами, що ускладнює їх використання, особливо для аналітичних досліджень чи пошуку специфічних даних. Найбільш прийнятним аналогом серед них визначено IMDb, враховуючи його широку базу даних, систему рейтингів та інтеграцію з багатьма платформами.

Таким чином, було доведено необхідність розробки системи, що дає змогу користувачам швидко знаходити потрібні дані, структурувавши інформацію за допомогою онтологічного підходу. Це дозволить створити зручну та ефективну базу знань, що відповідатиме запитам як пересічних кіноглядачів, так і дослідників кіноіндустрії.

Завдання дослідження було сформульовано таким чином, аби вирішити чинні проблеми в аналізі та систематизації інформації про кінематограф, а також обґрунтувати застосування онтологічного підходу для моделювання бази знань у цій сфері.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ БАЗИ ЗНАНЬ У ПРЕДМЕТНІЙ ОБЛАСТІ “КІНЕМАТОГРАФІЧНА ОНТОЛОГІЯ”

2.1 Визначення межі та масштабу онтологій

На початковому етапі створення онтології ключовим є чітке визначення її меж, масштабу та сфери застосування. Це дозволяє організувати дані та уникнути як надлишкової деталізації, так і надмірно загального підходу.

Під час проєктування онтології варто відповісти на декілька фундаментальних питань:

1. Яка предметна область буде охоплена онтологією?
2. Для чого вона створюється, які її основні завдання?
3. На які типи запитань вона повинна надавати відповіді?
4. Хто буде користувачем онтології, та як вона буде оновлюватися?

Відповіді на ці питання можуть змінюватися під час роботи над розробкою, адже онтологія – це модель, яка легко піддається змінам. Вона може розширюватися або, навпаки, звужуватися, все залежить від потреб тих, хто нею користуватиметься, і від наявних ресурсів. Але вже на початковій стадії необхідно визначити основні рамки, щоб уникнути надмірного ускладнення структури чи, навпаки, її недостатньої інформативності.

1) Предметна область онтології

У межах цього дослідження ми будемо розглядати онтологію, що охоплює світ кінематографу. Вона буде моделювати знання про фільми, режисерів, акторів, різні жанри, нагороди, ролі та ключові, саундтреки та композиторів або гурти. Головне завдання онтології полягає в створенні структурованої бази знань, яка допоможе оперативно знаходити необхідну інформацію про фільми, аналізувати взаємозв'язки між ними та покращити пошук за заданими критеріями.

2) Мета створення онтології

Основна мета розробки онтології кінематографу – це впорядкування та систематизація знань про кіно. Це дозволить створити ефективний інструмент для користувачів, котрі шукають детальну інформацію про фільми, їхніх творців і зв'язки між різними кінокартинами.

Використання онтологічного підходу забезпечує:

- Структуроване представлення інформації – усі дані пов'язані між собою за чіткими логічними правилами.
- Зручний пошук і навігацію – користувачі матимуть змогу знаходити потрібні відомості за різними критеріями: жанром, режисером, роком випуску, нагородами тощо.
- Аналіз та рекомендації – онтологія може використовуватися для створення рекомендаційних систем, котрі пропонуватимуть фільми на основі вже переглянутих або вподобаних кінокартин.
- Збереження та оновлення інформації – база знань може регулярно оновлюватися та доповнюватися, підтримуючи актуальність даних.

3) Основні типи запитів до онтології

Щоб онтологія була корисною та відповідала поставленим завданням, вона повинна підтримувати різноманітні типи запитів, серед яких:

- Пошук фільмів за жанром, роком випуску, режисером або актором.
- Пошук зв'язків між фільмами (наприклад, продовження, римейки, адаптації тощо).
- Аналіз популярності фільмів на основі рейтингів та отриманих нагород.
- Пошук рекомендованих фільмів за схожістю з тими, що вказав користувач.
- Визначення впливу певних режисерів або акторів на розвиток кіноіндустрії.

4) Цільова аудиторія онтології

Необхідно визначити потенційних користувачів онтології. Це дасть можливість адаптувати структуру та наповнення моделі, забезпечуючи відповідність потребам тих, хто з нею працюватиме. Основними користувачами можуть бути:

1. Кіномани – люди, котрі цікавляться фільмами та бажають отримувати детальну інформацію про них.
2. Дослідники кінематографу – аналітики, що проводять дослідження у сфері кіно та вивчають тенденції індустрії.
3. Розробники сервісів рекомендацій – фахівці, котрі використовують онтологію для створення алгоритмів підбору фільмів.

4. Сценаристи та режисери – професіонали, котрі шукають натхнення або аналізують творчість інших кіномитців.

Обмеження та масштаби онтології

Попри широкий спектр застосування онтології, необхідно встановити її обмеження, щоб не допустити надмірного ускладнення моделі. Наприклад, онтологія не міститиме детальної фінансової звітності кіностудій, даних про касові збори чи технічні характеристики кінозйомки, адже ці аспекти виходять за межі її основної мети.

Також важливе питання – багатомовність. Враховуючи глобальний характер кіноіндустрії, варто передбачити підтримку різних мов, щоб користувачі могли отримувати інформацію відповідно до своїх мовних уподобань. Це може включати назви фільмів, біографії режисерів та акторів, а також описи жанрів і сюжетів.

Інтеграція із зовнішніми ресурсами

Для того, щоб онтологія була корисною та актуальною, критично важливим є передбачення її інтеграції із зовнішніми джерелами даних. Це дозволить автоматично оновлювати інформацію про фільми, режисерів і жанри.

TMDb (The Movie Database) – відкрита платформа з API, яка надає інформацію про фільми.

DzygaDB – ресурс з деталізованими описами жанрів, стилів та кінематографічних особливостей.

TVDB – соціальна мережа для кінолюбителів з великою кількістю рецензій і рейтингів.

Включення цих ресурсів у процес проєктування онтології забезпечує її актуальність та розширює сфери її застосування.

2.2 Створення термінологічного словника онтології

Для створення онтологічної бази знань ключову роль відіграє створення термінологічного словника, який забезпечує формалізоване визначення основних понять предметної області та зв'язків між ними. Такий словник є ядром онтології: він задає структуру класів, описує властивості об'єктів і встановлює логічні правила взаємодії. Без чіткого термінологічного словника онтологія втрачає цілісність і не

може ефективно застосовуватись для інтелектуального аналізу, пошуку або виведення знань.

Загальна структура словника

Онтологія побудована на основі класу owl:Thing[7], від якого наслідуються всі інші поняття. У моделі реалізовано ієрархічну структуру, що охоплює основні компоненти кінематографічного процесу. Ключові класи подано нижче:

Таблиця 2.1 – Основні класи та їх описи

Клас	Опис
Фільм	Центральний об'єкт моделі — кінострічка
Режисер	Особа, яка знімає фільм
Актор	Виконавець ролей у фільмах
Сценарист	Автор сценарію
Продюсер	Відповідальний за організацію виробництва
Кінокомпанія	Організація, що створює або випускає фільми
Жанр	Тематичний або стилістичний тип фільму
Нагорода	Премії, відзнаки, що присуджуються фільмам
Роль	Персонаж, якого грає актор
Композитор/Гурт	Автор музики до фільму
Назва музичного твору	Музичні композиції, що з'являються у фільмі
Особа	Надклас для акторів, режисерів, сценаристів та продюсерів

Усі зазначені поняття представлені у вигляді класів, між якими встановлено відношення — об'єктні властивості, а також — властивості-дані, які дозволяють деталізувати об'єкти конкретними значеннями (назва, рік, тривалість тощо).

Об'єктні властивості

Об'єктні властивості встановлюють логічні зв'язки між інстансами класів. У розробленій онтології виділено такі основні відношення:

Таблиця 2.2 – Об'єктні властивості та їх зв'язки та описи

Назва властивості	Від (домен)	До (діапазон)	Опис зв'язку
граєРоль	Актор	Роль	Яку роль виконує актор у фільмі
маєАктора	Фільм	Актор	Актор, який грає у фільмі
маєКомпозитора	Фільм	Композитор/Гурт	Музичний автор фільму
маєМузичнийТвір	Фільм	Назва музичного твору	Ймовірно, зв'язок з музичною композицією - потребує уточнення
маєРоль	Фільм	Роль	Які ролі присутні у фільмі
належитьДоЖанру	Фільм	Жанр	Класифікація фільму за жанром
написавСценарій	Фільм	Сценарист	Автор сценарію фільму
отримавНагороду	Фільм	Нагорода	Нагороди, які здобув фільм або його творці
режисерував	Фільм	Режисер	Хто виступив режисером
створеноКомпанією	Фільм	Кінокомпанія	Якою студією або компанією створено фільм
маєПродюсера	Фільм	Продюсер	Хто виступав продюсером

Ці властивості дозволяють будувати запити типу: “Знайти всі фільми, в яких знявся актор N” або “Знайти всі фільми жанру “драма”, які мають музику композитора X”.

Властивості даних

Властивості даних описують внутрішні характеристики об'єктів за допомогою значень примітивних типів, таких як `string`, `gYear`. Нижче подано ключові атрибути:

Таблиця 2.3 – Властивості даних та їх зв'язки, типи значень та описи

Назва властивості	Клас	Тип значення	Опис
назваФільму	Фільм	string	Назва фільму
рікВипуску	Фільм	gYear	Рік виходу фільму
країнаПоходження	Особа	string	З якої країни походить особа
оцінкаФільму	Фільм	string	Оцінка надана фільму

Наявність властивостей-даних дозволяє розширити інтерфейс пошуку фільмів за назвами, роками, нагородами тощо, а також забезпечити підтримку фільтрації в запитах SPARQL.

Ієрархія понять

Для зменшення надлишковості у моделі було реалізовано підпорядкованість через ієрархію класів. Зокрема:

- Класи Режисер, Актор, Сценарист, Продюсер є підкласами загального поняття Особа. Це дозволяє використовувати загальний підхід до опису персоналій у фільмах.
- Усі індивіди, що представляють людей, можуть бути описані єдиною системою атрибутів і зв'язків.

2.3 Моделювання онтології засобами Protégé.

Створення онтологічної бази знань є ключовим етапом у формуванні структурованої інтелектуальної системи, яка дозволяє впорядковувати та зберігати інформацію про фільми, жанри, режисерів, акторів, музичні твори, виконавців і рекомендації для перегляду та прослуховування, а також інші важливі аспекти цих сфер. Інструментальне середовище Protégé[15] надає широкі можливості для

модельовання таких елементів в єдиному інформаційному просторі, що значно спрощує процес налаштування, подальшого використання та інтеграції знань у зовнішні системи.

Процес створення онтології в середовищі Protégé включає кілька основних етапів: визначення ключових класів, додавання атрибутів (data properties), побудова ієрархії класів, а також створення об'єктних властивостей (object properties) для формалізації взаємозв'язків між об'єктами. Такий підхід дозволяє точно та гнучко представити знання про музику і кіно, забезпечуючи їх систематизацію, подальший аналіз та ефективне використання у практичних застосуваннях.

1) Створення класів

На першому етапі класу було проведено аналіз предметної області з метою виявлення основних сутностей, які підлягають формалізації. Кожна сутність була представлена у вигляді окремого в онтології. Усього було створено 8 базових класів, які умовно поділено на кілька груп:

Створення класів здійснювалося відповідно до принципів онтологічного моделювання: кожен клас відповідає узагальненому поняттю, яке має значення у предметній галузі та може мати підкласи або бути пов'язаним з іншими через відношення. Створенні класи зображені на рисунку 2.1



Рисунок 2.1 - Список створених класів

Наприклад, клас “Фільм” охоплює всі типи кінематографічних творів, тоді як клас “Назва музичного твору” — всі типи музичних композицій. Це дозволяє побудувати єдину модель, де обидві галузі (музика та кіно) можуть бути описані паралельно, з урахуванням перетинів (наприклад, фільм має саундтрек, що є музичним твором).

2) Побудова ієрархії класів

Після створення базових класів формується ієрархічна структура, що дозволяє впорядкувати знання за принципом загальності/специфічності. Ієрархія класів дозволяє уникнути дублювання, полегшує розширення онтології та сприяє логічному групуванню понять.

У розробленій онтології виділено такі надкласи на рисунку 2.2:

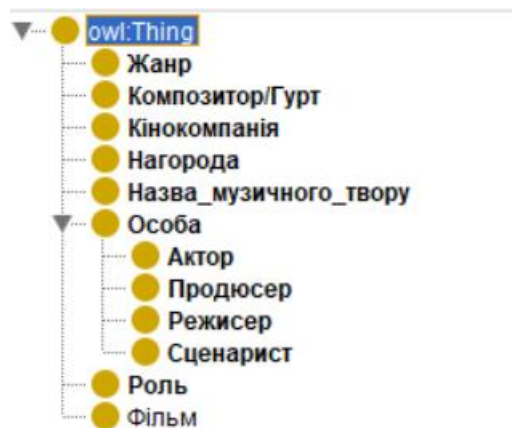


Рисунок 2.2 - Список створених надкласів

3) Додавання атрибутів (datatype properties)

Для уточнення характеристик та зв'язків між об'єктами в онтології створено два типи властивостей:

Властивості-дані (Datatype Properties):

- рікВипуску — рік випуску (тип: gYear).
- тривалістьФільму — тривалість твору або фільму (тип: integer, хвилини).
- країнаПоходження — країна походження особи (тип: string).

Створені атрибути зображені на рисунку 2.3-2.5.

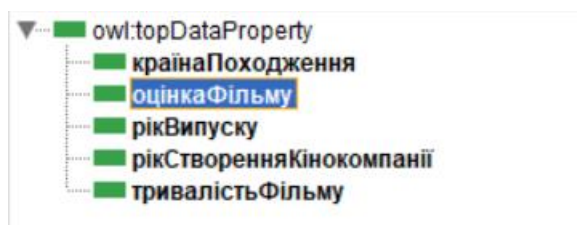


Рисунок 2.3 – Атрибути класів онтологічної бази знань

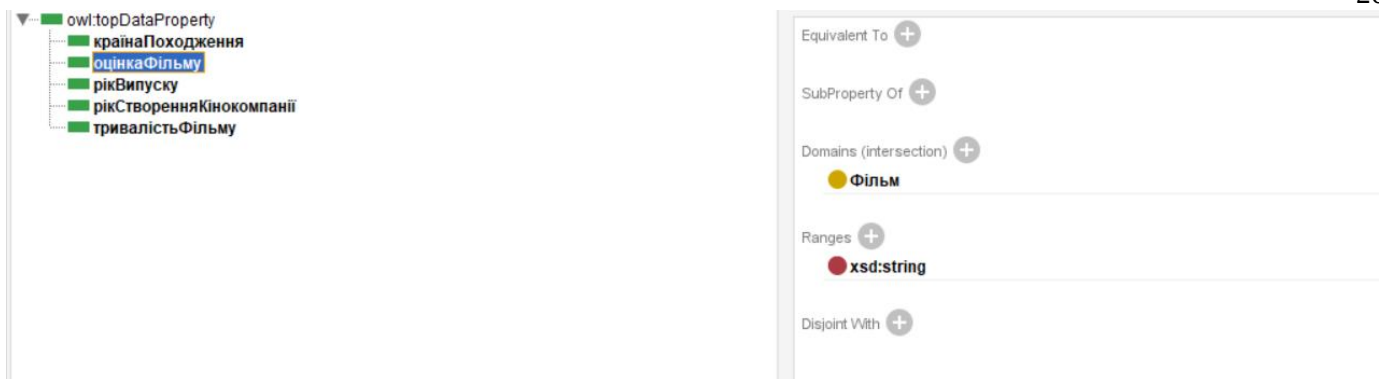


Рисунок 2.4 – Атрибут “оцінкаФільму”

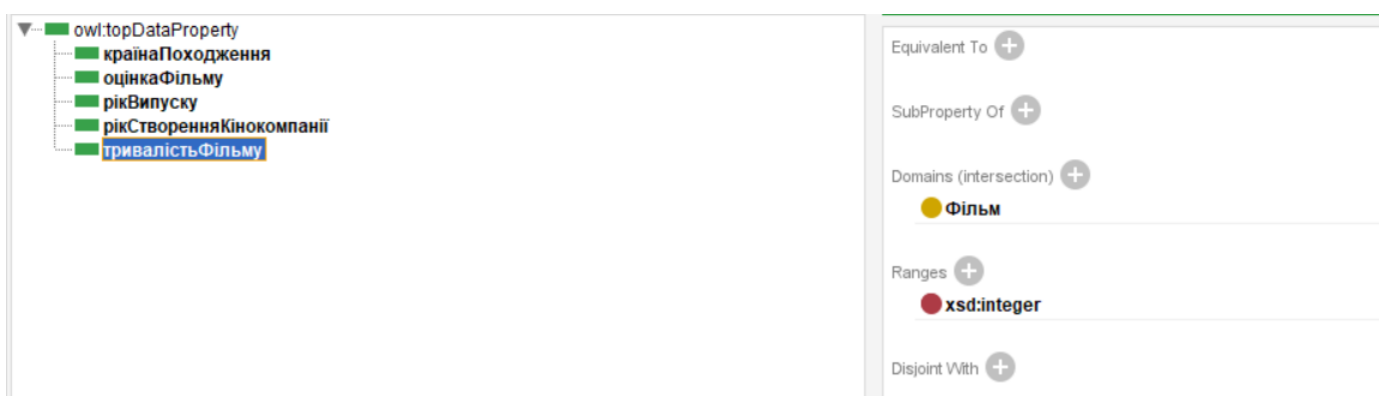


Рисунок 2.5 – Атрибут “тривалістьФільму”

Важливість атрибутів полягає у деталізації даних, що дозволяє отримати повнішу інформацію про кожен об'єкт. Це забезпечує не лише точність, але й робить онтологію більш функціональною, оскільки кожен атрибут виконує свою роль у побудові моделі. Атрибути надають можливість зберігати додаткові характеристики для кожного класу.

4) Встановлення об'єктних властивостей (object properties)

Об'єктні властивості дозволяють задавати зв'язки між класами, відображаючи взаємодію між різними елементами онтології. У Protege об'єктні властивості налаштовуються індивідуально для кожного класу, що робить модель більш реалістичною та функціональною.

Для відображення взаємозв'язків між класами було створено об'єктні властивості:

- маєМузичнийТвір — зв'язок Фільм → Назва_музичного_твору, що вказує на саундтреки кінофільму.
- маєКомпозитора — зв'язок Фільм → Композитор/Гурт, що визначає виконавця твору.

- належитьДоЖанру — відношення Фільм → Жанр, що вказує на жанр.
- маєАктора — відношення Фільм → Актор, що вказує на актора, який грав у вибраному фільмі.
- маєПродюсера — зв'язок Фільм → Продюсер, що вказує на продюсера фільма
- маєРоль — зв'язок Актор → Роль, що вказує на актора, який грає свою роль у вибраному фільмі
- написавСценарій — зв'язок Фільм → Сценарист, що вказує на сценариста фільму
- отримавНагороду — зв'язок Фільм → Нагорода, що вказує на нагороду, який отримав вибраний фільм
- режисерував — зв'язок Фільм → Режисер, що вказує на режисера фільму
- створеноКомпанією — зв'язок Фільм → Кінокомпанія, що вказує на кінокомпанію, яка створила фільм

Створені об'єктні властивості зображені на рисунках 2.5-2.8

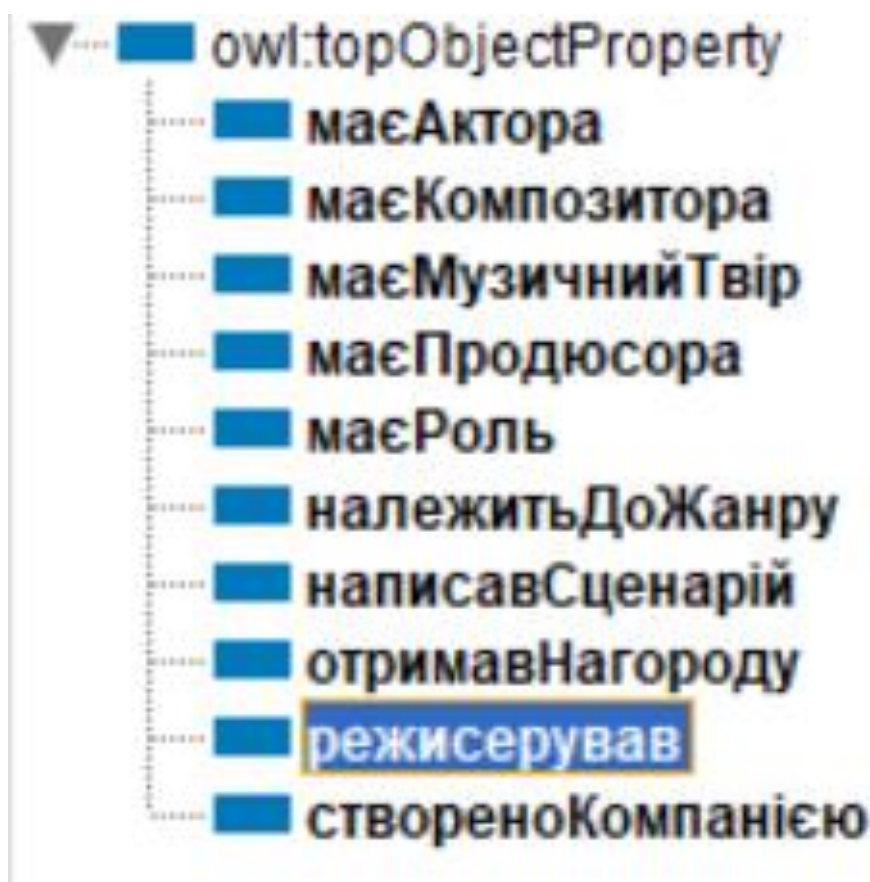


Рисунок 2.6 – Об'єктні властивості онтологічної бази знань

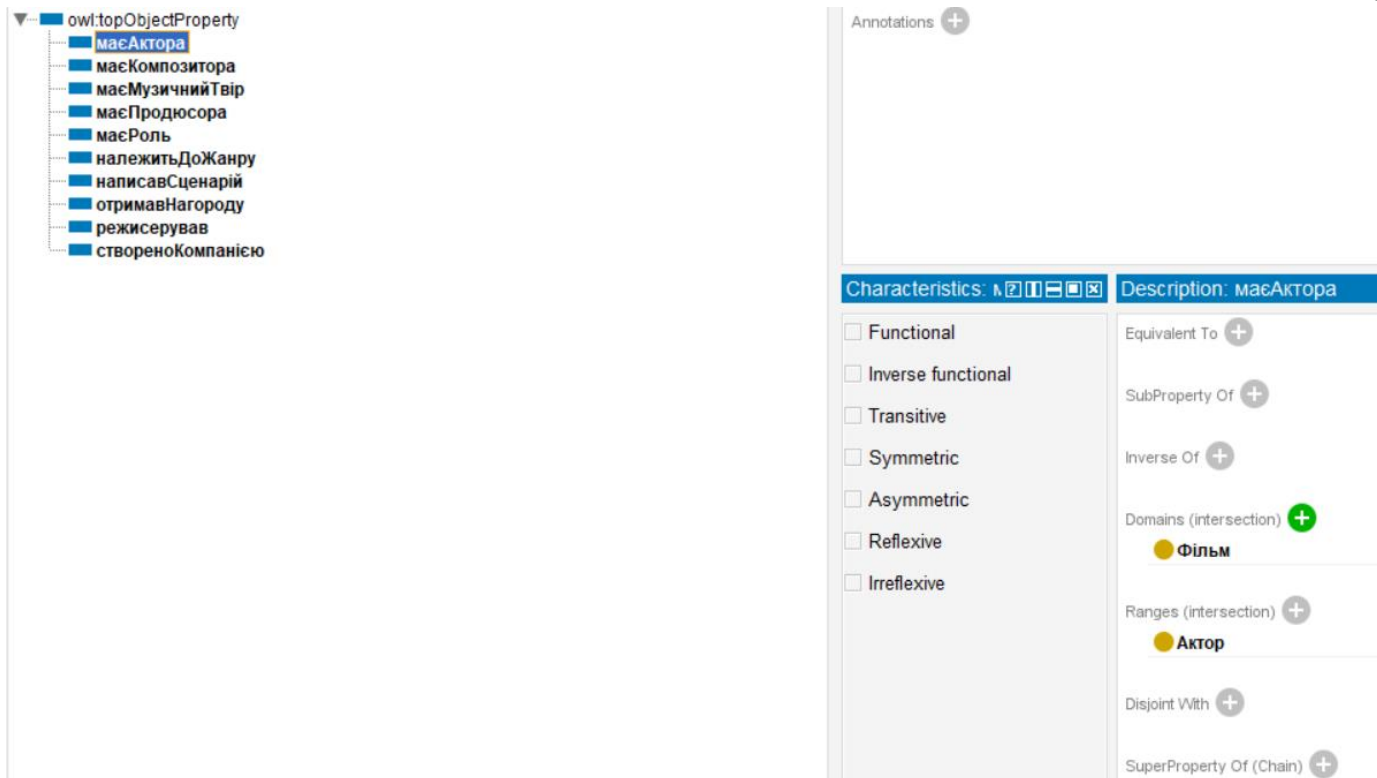


Рисунок 2.7 – Об’єктна властивість “маєАктора”

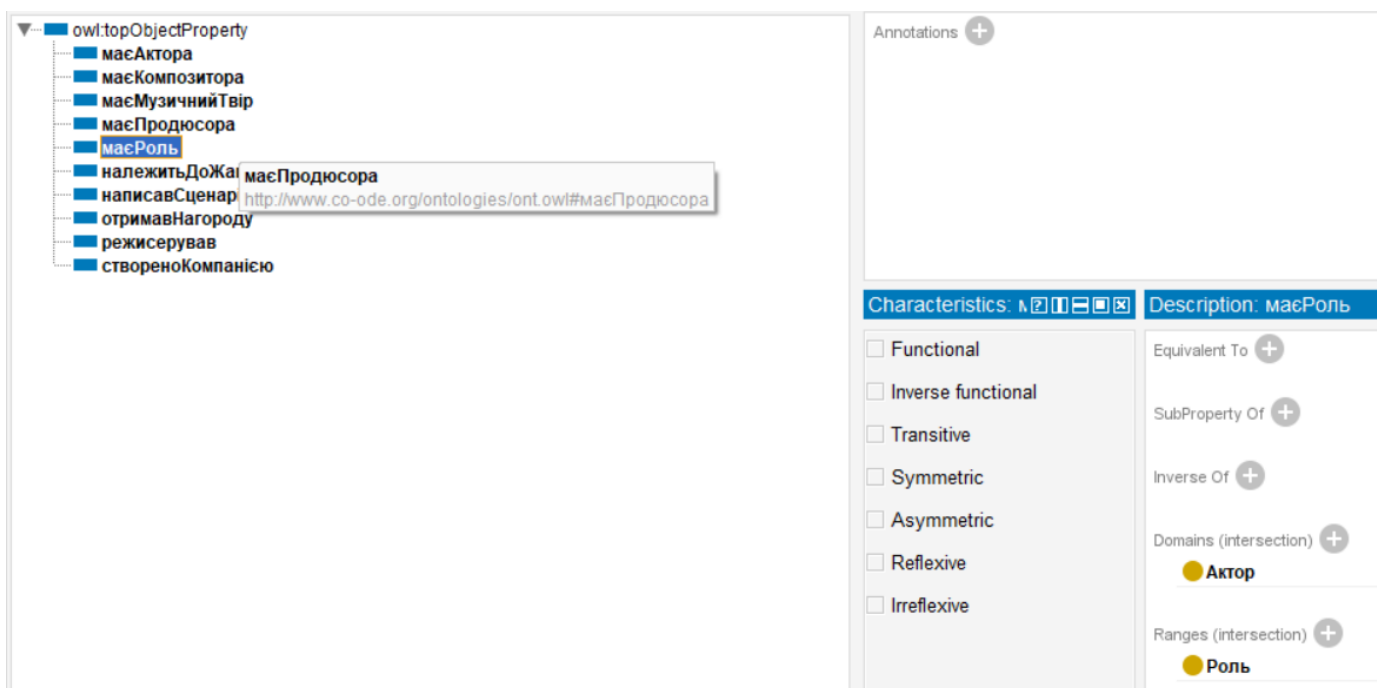


Рисунок 2.8 – Об’єктна властивість “маєРоль”

Ці властивості моделюють семантичні зв’язки між основними класами, дозволяючи розгортати повноцінну базу знань.

Створення індивідів

Для демонстрації роботи онтології та тестування її логіки було додано конкретні екземпляри класів:

- Кінофільм: Бійцівський клуб (рікВипуску: 1999, оцінкаФільму: 8.8/10,

тривалістьФільму: 139).

- Назва музичного твору: Where is my mind?.
- Композитор/Гурт: Pixies.
- Жанр: драма.
- Актор: Бред Пітт
- Роль: Тайлер Дерден
- Продюсор: Арнон Мілчен
- Сценарист: Джим Ульс
- Нагорода: Оскар за 1999 рік
- Режисер: Девід Фінчер
- Кінокомпанія: 20th Century Fox

Створені індивіди зображені на рисунку 2.9

Class hierarchy: Фільм

- owl:Thing
 - Жанр
 - Композитор/Гурт
 - Кінокомпанія
 - Нагорода
 - Назва_музичного_твору
 - Особа
 - Роль
 - Фільм

Types

- Фільм

Property assertions: Бійцівський_Клуб

Object property assertions

- масАктора Бред_Пітт
- масМузичнийТвір Where_is_my_mind?
- масПродюсера Арнон_Мілчен
- масРоль Бред_Пітт
- належитьДоЖанру Драма
- написавСценарій Джим_Ульс
- отримавНагороду Оскар_за_1999
- режисерував Девід_Фінчер
- створеноКомпанією 20th_Century_Fox

Data property assertions

- оцінкаФільму "8.8/10"
- рікВипуску "1999"^^xsd:gYear
- тривалістьФільму 139

Direct instances: Бійцівський_Клуб

For: Фільм

- Бійцівський_Клуб
- Зоряні_Війни:_Нова_надія
- Роккі
- Ромео+Джुльєтта
- Хрещений_Батько

Рисунок 2.9 – Індивіди класу “Фільм”

Ці індивіди демонструють основні відносини в онтології: саундтрек належить

кінофільму, виконується конкретним артистом, належить до жанру та включений в альбом.

6) Візуалізація структури

Для покращення розуміння структури онтології застосовано плагін OntoGraph[16] у Protégé. Графічна візуалізація демонструє ієрархію класів.

На рисунку 2.10 зображено OntoGraph класів та підкласів онтологічної бази знань “Кінематографічна онтологія”

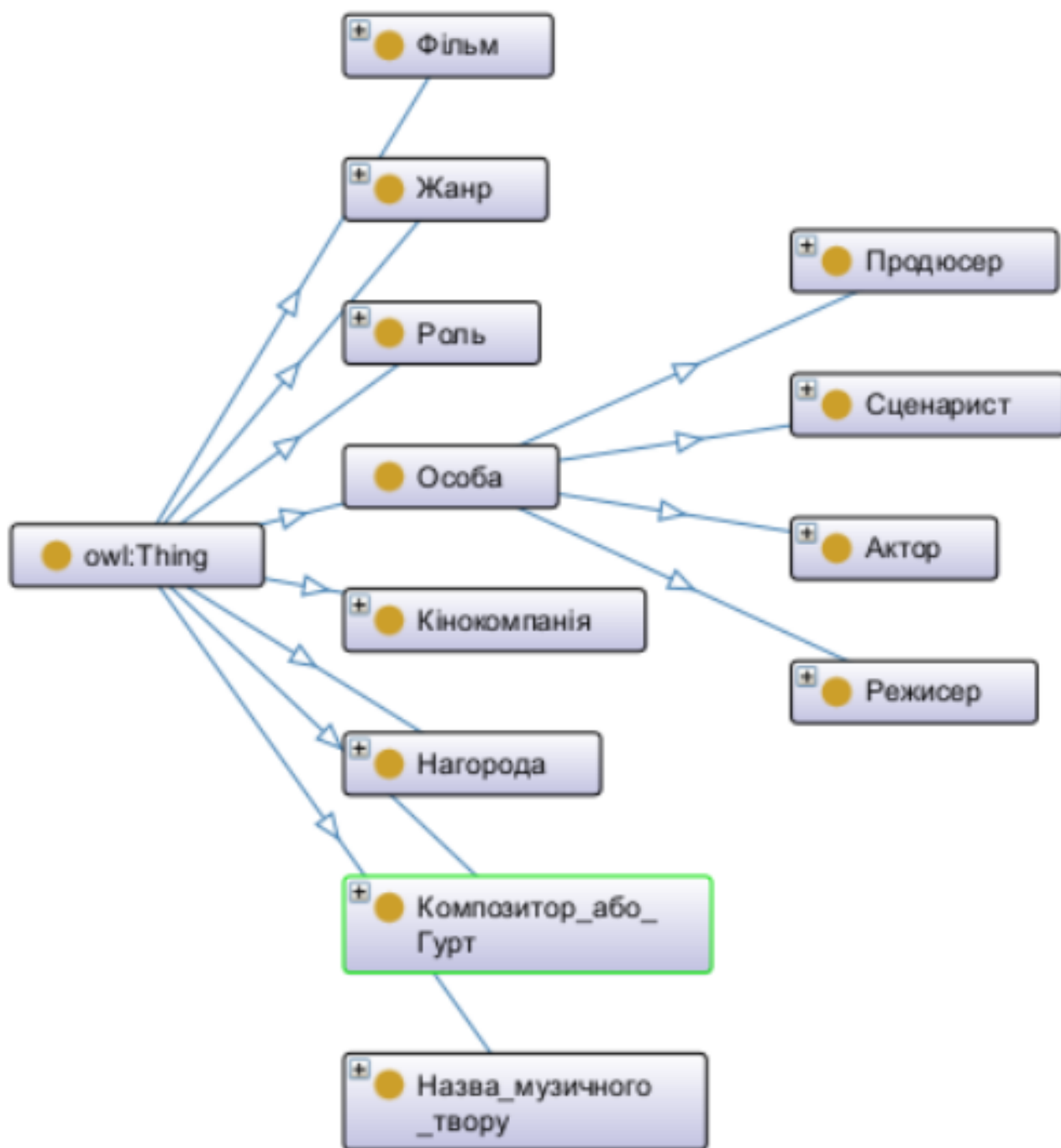


Рисунок 2.10 – Візуалізація класів та підкласів онтології за допомогою OntoGraph

Далі зобразимо індивідів всієї онтології.

На рисунку 2.11 зображено OntoGraph класів та підкласів, індивідів, та всіх зв'язків між ними, онтологічної бази знань “Кінематографічна онтологія”

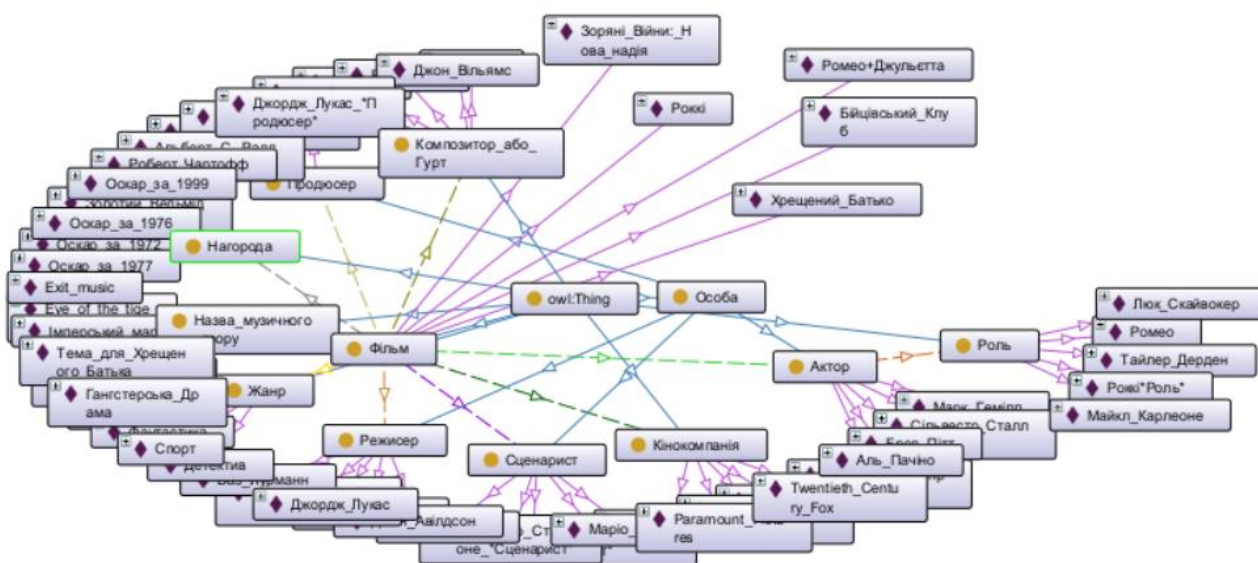


Рисунок 2.11 – Візуалізація класів, підкласів, індивідів, та зв'язків між ними, онтології за допомогою OntoGraph

На графі видно, що клас Фільми пов'язаний з кількома екземплярами назв музичних творів, які своєю чергою пов'язані з композиторами/гуртами. Візуалізація дозволяє легко простежити семантичні зв'язки та контролювати цілісність даних.

Крім того, така візуалізація є важливою при виявленні помилок у побудові онтології — наприклад, неправильних зв'язків, відсутніх об'єктів або надлишкових дублювань. Завдяки OntoGraph користувач має змогу в інтерактивному режимі оцінювати повноту моделі, перевіряти логічну послідовність між класами, підкласами та індивідами, а також бачити весь спектр їх взаємодії. Це особливо корисно на етапах тестування та валідації, коли навіть незначна структурна помилка може вплинути на коректність запитів.

Графічне представлення моделі також сприяє кращому розумінню онтології для сторонніх користувачів, яким складно орієнтуватися у текстовому описі. Наявність візуальної схеми дозволяє швидко охопити загальну структуру бази знань, виявити ключові вузли та зрозуміти семантику зв'язків. Крім того, така схема може бути використана як частина документації або презентації результатів проєкту.

2.4 Висновок до розділу 2

У другому розділі було розглянуто основні аспекти моделювання онтологічної бази знань у предметній області "Кінематографічна онтологія". Проведений аналіз дозволив визначити ключові поняття, що складають основу кінематографічної онтології, та сформувати їх ієрархічну структуру, що забезпечує логічне впорядкування знань.

Розроблена термінологія бази знань включає класи, які відображають основні сутності кіноіндустрії — фільми, режисерів, акторів, жанри, кінокомпанії, нагороди та інші важливі компоненти. Кожен із цих класів отримав набір атрибутів, що детально описують їх характеристики, що значно підвищує точність та повноту моделі.

Використання засобів Protégé для створення онтології надало можливість ефективно реалізувати ієрархію класів, визначити об'єктні та дата-властивості, а також наповнити базу реальними індивідами, які демонструють практичну дієвість моделі. Накладені обмеження і правила допомогли забезпечити консистентність даних та запобігти логічним помилкам у побудові зв'язків.

Візуалізація онтології за допомогою плагіну OntoGraph сприяла глибшому розумінню структури моделі, дозволила відстежити всі зв'язки між класами та індивідами, а також полегшила процес контролю якості розробленої бази знань.

Таким чином, у результаті моделювання була створена структурована, гнучка та масштабована онтологічна база знань, яка відповідає поставленим вимогам предметної області "Кінематографічна онтологія". Вона є основою для подальшої реалізації функціональності інформаційної системи, що дозволить здійснювати ефективний пошук, аналітику та побудову рекомендацій у сфері кінематографу.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ “КІНЕМАТОГРАФІЧНА ОНТОЛОГІЯ” В СЕРЕДОВИЩІ PROTEGE

3.1 Застосування Preferences в середовищі Protégé

Середовище Protégé підтримує широкий набір налаштувань, які доступні у вікні Preferences. Це вікно надає користувачу змогу гнучко керувати параметрами відображення, створення онтологій, інтеграції з плагінами та іншими технічними аспектами проєкту [17]. Для доступу до преференсів необхідно перейти в меню File → Preferences (на Windows/ Linux) або Protégé → Preferences (на macOS). На рисунку 3.1-3.2 зображено шлях знаходження, і вікно, Preferences :

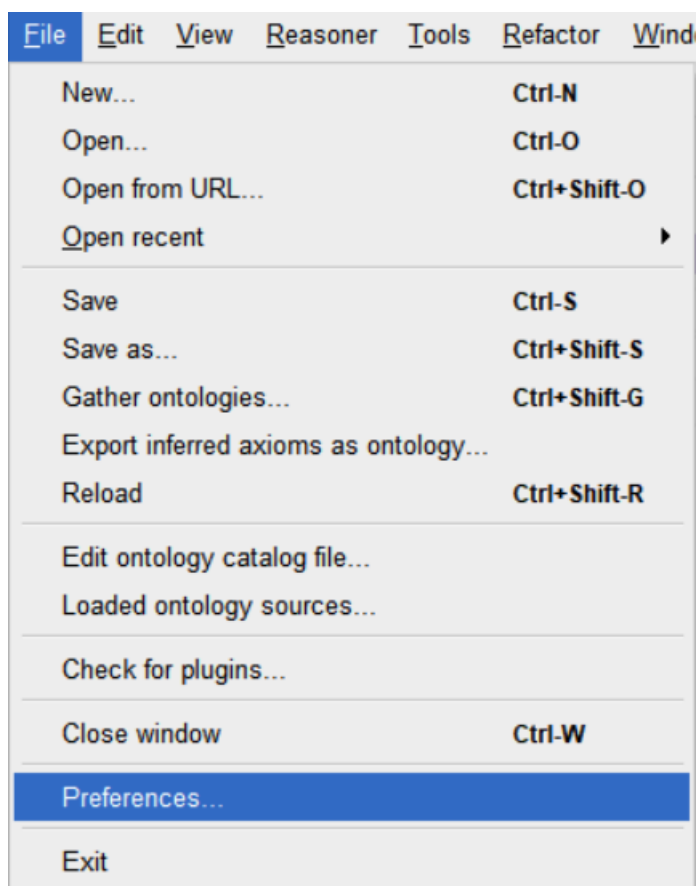


Рисунок 3.1 - Зображення шляху знаходження вікна Preferences:

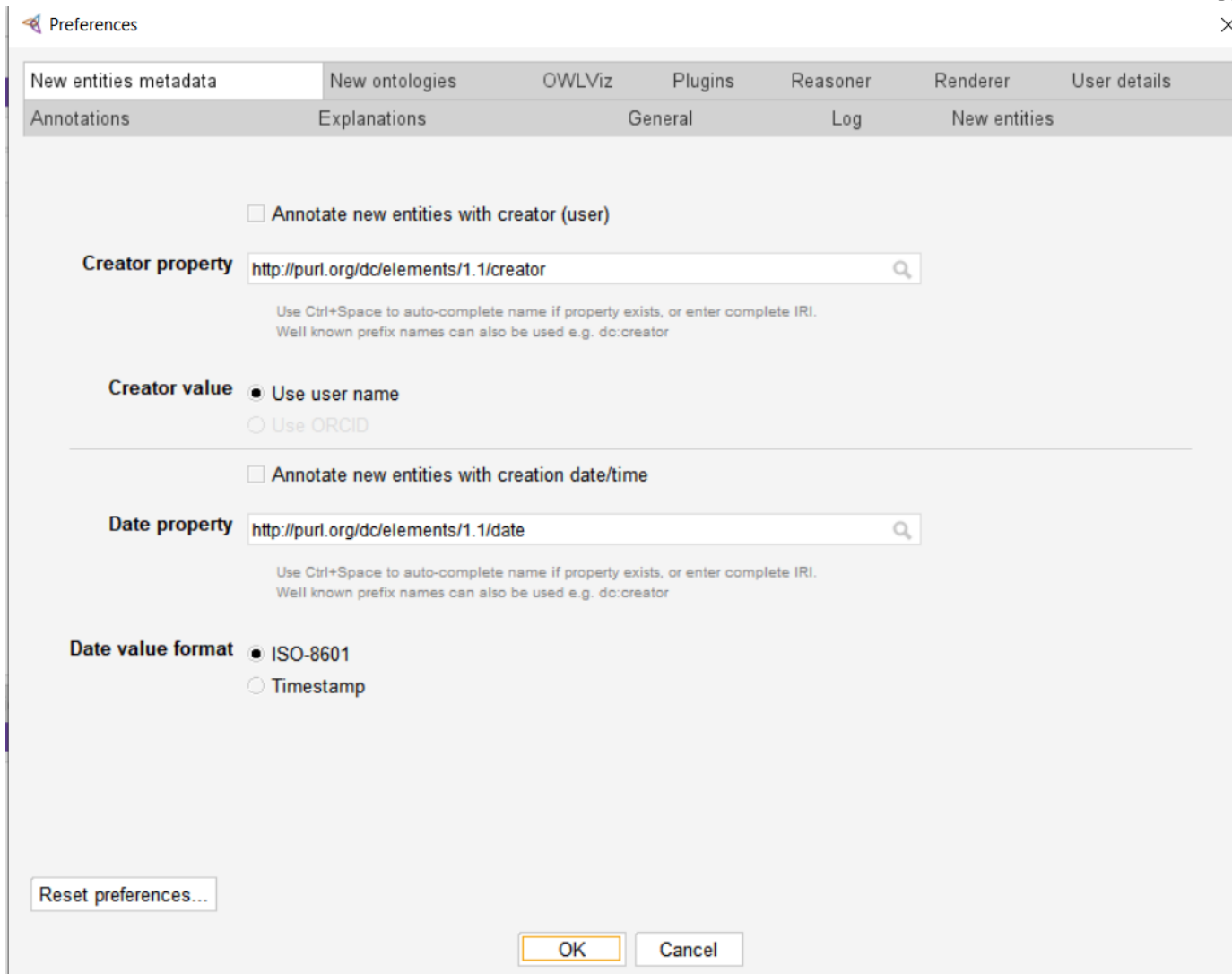


Рисунок 3.2 - Зображення вікна Preferences

Далі розглянемо основні вкладки меню Preferences:

Вкладка New entities metadata

Дає змогу налаштувати шаблон генерації ідентифікаторів (IRI) для нових сутностей:

- Вибір базового IRI: активна онтологія або вказане вручну.
- Вибір символу, яким розділятиметься базовий IRI та назва сутності: #, /, ..
- Вибір способу завершення IRI: вручну задана назва або автоматично згенерований ID.

• У нижній частині задається шаблон префіксів, суфіксів або мови для автогенерованих назв [17].

Вкладка New ontologies

Ця вкладка дозволяє:

- Встановити базовий URI для нової онтології.

- Визначити, чи буде автоматично створено індивіда класу owl:Thing.
- Вказати мову онтології за замовчуванням: OWL 2 DL, OWL Lite тощо [17].

Вкладка OWLViz

- Налаштування для візуалізації ієрархії класів та властивостей:
- Вибір кольору для класів, об'єктних і дата-властивостей.
- Відображення анотацій або підказок у графі.
- Параметри шрифтів і напрямку побудови графа.

Ця вкладка активна при встановленому плагіні OWLViz [18].

Вкладка Plugins

Містить:

- Перелік встановлених плагінів.
- Можливість увімкнення/вимкнення кожного плагіну.
- Перевірку оновлень для розширень [17].

Вкладка Reasoner

Дозволяє налаштувати логічні засоби:

- Вибір ризонера (HermiT, ELK, Pellet).
- Автоматичний запуск reasoning при зміні моделі.
- Встановлення timeout для reasoning.
- Відображення пояснень конфліктів [19], [20].

Вкладка Renderer

Визначає спосіб відображення елементів у середовищі Protégé:

- Використання rdfs:label або коротких імен (prefixed names).
- Налаштування простору імен (namespaces).
- Вибір відображення URI чи назв сутностей [17].

Вкладка User details

Використовується для збереження даних користувача:

- Ім'я користувача, email, організація — застосовується під час анотування онтології.

- Інформація автоматично додається до анотацій як метадані [17].

Вкладка Annotations

- Налаштовує типи та форматування анотацій, які буде додано до онтології:

- Визначення властивостей, що використовуються як коментарі (rdfs:comment, dc:description тощо).

- Встановлення мови анотацій.

Вкладка Explanations

Управління механізмами пояснення логічних конфліктів:

- Налаштування для Explanation Workbench плагіну.
- Вибір типу пояснень: повні, мінімальні, автоматичні.
- Увімкнення/вимкнення пояснень у Reasoner Views [21].

Вкладка General

Базові параметри інтерфейсу:

- Автозбереження файлів.
- Поведінка при запуску Protégé.
- Активність підказок, мова інтерфейсу.

Вкладка Log

Налаштування виведення повідомлень:

- Встановлення рівня журналювання (info, warning, error).
- Можливість зберігати логи reasoning та помилки.

Вкладка New entities

Вже згадана вище, але також містить:

- Опції для автогенерації назв.
- Вибір мови (Lang) для мітки сутностей.
- Режими генерації ID (ітеративні, унікальні, ідентичні) [17].

Середовище Protégé дозволяє експортувати налаштування преференсів для подальшого використання через меню File → Export Preferences зі збереженням у XML або Properties файл, включенням всіх або вибіркового налаштувань та розташуванням у локальній файлової системі. Для відновлення або перенесення налаштувань використовується File → Import Preferences із завантаженням з файлу, перевіркою сумісності, вибіркового застосуванням налаштувань та перезапуском програми при необхідності.

Створення профілю налаштувань для команди розробників включає XML конфігурацію з основними параметрами мови, автоматичного збереження, інтервалу збереження, використання міток, мови за замовчуванням, резонера за замовчуванням, виділення

пам'яті та тайм-ауту. Такий підхід забезпечує уніфіковане робоче середовище для всіх учасників проекту.

3.2 Створення SPARQL запитів

SPARQL – це мова запитів для взаємодії з даними та є однією з ключових технологій семантичної павутини. Вона призначена не лише для отримання даних шляхом виконання запитів, але й для зміни даних, таких як видалення існуючої інформації або додавання нових даних до бази знань. Надання SPARQL-точок доступу є рекомендованою практикою при публікації даних у всесвітньому павутинні.

До мови SPARQL відносяться чотири типи запитів: SELECT застосовується для видобування "сирих" значень з SPARQL-ендпоінту, результати відображаються як таблиця. CONSTRUCT використовується для отримання інформації з SPARQL-ендпоінту у форматі RDF та перетворення отриманого результату до потрібної форми. ASK застосовується для створення запитів типу "істина/хибність". DESCRIBE використовують для отримання опису RDF-ресурсу. Реалізація запиту DESCRIBE залежить від розробника SPARQL-ендпоінту. Кожна форма запиту повинна містити блок WHERE для обмеження пошуку. Далі наведена частина ключових слів, що найчастіше використовуються в SPARQL-запитах. Повний перелік доступний в офіційній документації:

- DISTINCT – показує унікальні результати;
- LIMIT – визначає максимальну кількість результатів;
- OFFSET – дозволяє не відображати перші *n* рішень;
- ORDER BY – сортує результати за зростанням чи спаданням;
- PREFIX – скорочує URI;
- OPTIONAL – визначає необов'язковий шаблон;
- GRAPH – застосовується для формування запитів до іменованих графів.

SPARQL-ендпоінт, також відомий як точка доступу SPARQL, є сервісом, що підтримує мову запитів SPARQL. Ця точка дозволяє користувачам надсилати запити до бази знань та отримувати результати у різноманітних форматах. Таким чином, SPARQL-ендпоінти розроблені для надання зручного інтерфейсу до бази знань у вигляді сервісу. Існує два основних типи точок доступу: загального та локального

призначення. Точки загального призначення можуть обробляти запити до будь-яких RDF-документів, що знаходяться в Мережі. Локальні точки доступу можуть отримувати дані лише з одного конкретного ресурсу.

Для того, щоб виконати запит SPARQL, потрібно перейти до вкладки “SPARQL Query”. Якщо вкладка не відображається, в меню Window переходимо по вкладці Tabs і виберемо “SPARQL Query”. На рисунку 3.2 зображено приклад знаходження вкладки “SPARQL Query”

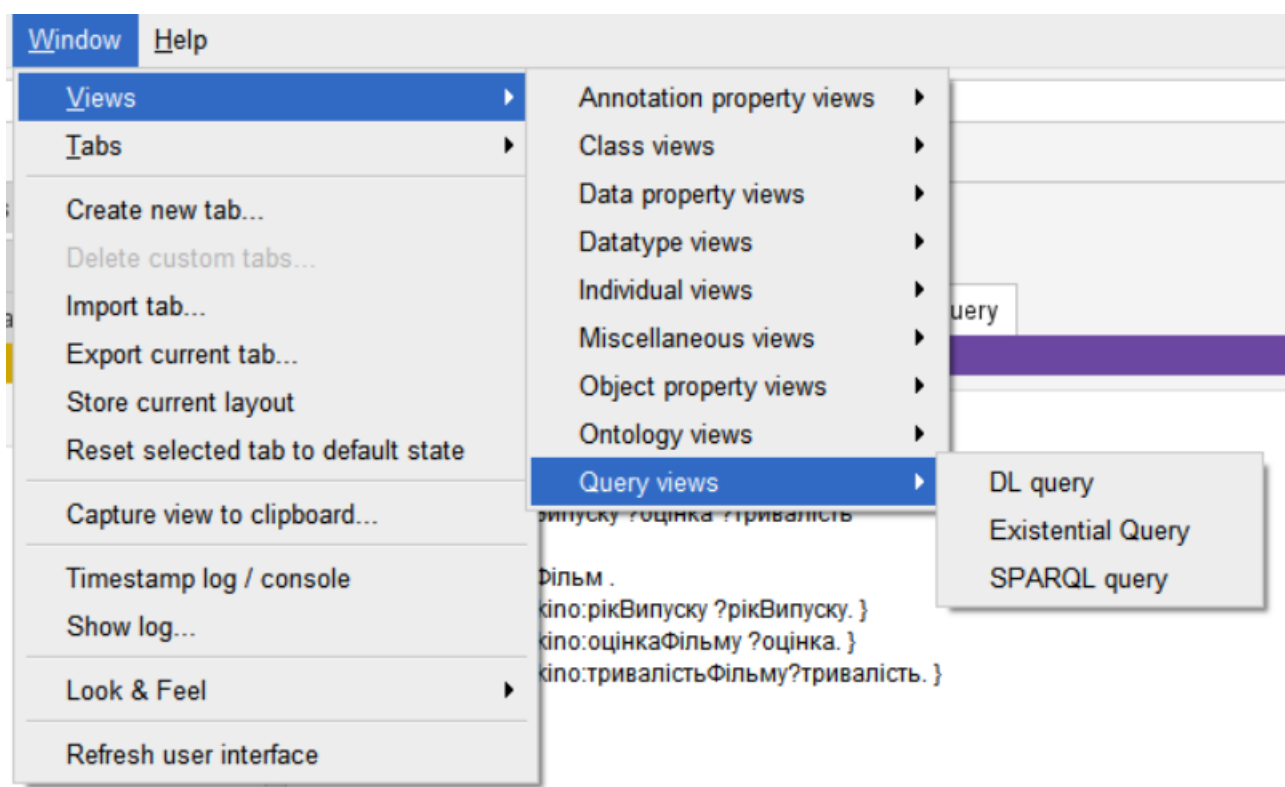


Рисунок 3.2 - Місцеперебування вкладки “SPARQL Query”

Для створення запитів на мові SPARQL спочатку потрібно в текстовому полівкладки “SPARQL Query” задати простори імен, які будуть використовуватися. В цій розробці використовуються простори імен. При чому простори rdf, owl, rdfs та xsd є стандартними і прописуються автоматично.

Нище, на рисунках 3.3-3.4 наведено запити, які показують повну інформацію про фільми, наприклад жанр, актора, роль актору тощо.

PREFIX kino: <http://example.org/kino#>
 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?фільм ?жанр ?актор ?роль ?саундтрек ?композитор_гурт ?режисер ?сценарист ?продюсер ?кінокомпанія ?нагорода
 WHERE {
 ?фільм rdf:type kino:Фільм .

OPTIONAL { ?фільм kino:належитьДоЖанру ?жанр. }
 OPTIONAL { ?фільм kino:маєАктора ?актор. }
 OPTIONAL { ?фільм kino:маєРоль ?роль. }
 OPTIONAL { ?фільм kino:маєМузичнийТвір ?саундтрек. }
 OPTIONAL { ?фільм kino:маєКомпозитора ?композитор_гурт. }
 OPTIONAL { ?фільм kino:режисерував ?режисер. }
 OPTIONAL { ?фільм kino:написавСценарій ?сценарист. }
 OPTIONAL { ?фільм kino:маєПродюсера ?продюсер. }
 OPTIONAL { ?фільм kino:створеноКомпанією ?кінокомпанія. }
 OPTIONAL { ?фільм kino:отримавНагороду ?нагорода. }

фільм	жанр	актор	роль	саундтрек	композитор_гурт	режисер	сценарист	продюсер	кінокомпанія	нагорода
Хрещений Батьо	Гангстерська_Др	Аль_Пачіно	Майкл_Карлсон	Тема_для_Хрещ	Ніно_Ротта	Френсіс_Форд_К	Маріо_Пьюзо	Альберт_С_Рад	Paramount_Pictu	Оскар_за_1972
Роккі	Спорт	Сільвестр_Сталл	Роккі*Роль*	Eye_of_the_tiger	Survivor	Джон_Авілдсон	Сільвестр_Сталл	Роберт_Чартофс	United_Artists	Оскар_за_1976
Ромео+Джувелетт	Детектив	Леонардо_ДіКап	Ромео	Exit_music	Radiohead	Баз_Лурманн	Крейг_Пірс	Габрієлла_Март	Twentieth_Centur	Золотий_Ведьмі
Бійцівський_Клуб	Драма	Бред_Пітт	Бред_Пітт	Where_is_my_mi	Pixies	Девід_Фінчер	Джим_Ульс	Арнон_Мілчен	20th_Century_Fo	Оскар_за_1999
Зоряні_Війни:_Н	Фантастика	Марк_Гемілл	Люк_Скайвокер	Імперський_мар	Джон_Вільямс	Джордж_Лукас	Джордж_Лукас_*	Джордж_Лукас_*	Lucasfilm	Оскар_за_1977

Рисунок 3.3 - Виконаний запит, який використовує тільки об'єктні властивості, про повну інформацію фільму

Запит використовує конструкцію OPTIONAL, що дає можливість отримати всі фільми, навіть якщо деякі властивості відсутні. Отже, результат міститиме повну інформацію про кожен фільм, з усіма наявними атрибутами.

PREFIX kino: <http://example.org/kino#>

SELECT ?фільм ?рікВипуску ?оцінка ?тривалість
 WHERE {
 ?фільм rdf:type kino:Фільм .
 OPTIONAL { ?фільм kino:рікВипуску ?рікВипуску. }
 OPTIONAL { ?фільм kino:оцінкаФільму ?оцінка. }
 OPTIONAL { ?фільм kino:тривалістьФільму ?тривалість. }
 }

фільм	рікВипуску	оцінка	тривалість
Хрещений Батько	"1972" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"9.2/10"	"175" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Роккі	"1976" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"8/10"	"120" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Ромео+Джувелетт	"1996" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"6.2/10"	"120" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Бійцівський_Клуб	"1999" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"8.8/10"	"139" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Зоряні_Війни:_Нова_надія	"1977" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"8.2/10"	"121" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>

Рисунок 3.4 - Виконаний запит про рік випуску, оцінку та тривалість фільму, через атрибути “рікВипуску”, “оцінкаФільму”, “тривалістьФільму”

Даний запит зосереджений на ключових параметрах кінострічок: даті прем'єри, оцінці та тривалості. Цей метод ефективний для формування переліків чи систем ранжування.

Ключова особливість обох запитів - активне використання конструкції OPTIONAL. Це необхідно тому, що:

- Не всі фільми можуть мати всі атрибути (наприклад, деякі фільми можуть не мати інформації про композитора)
- RDF дані часто є неповними або розрідженими
- OPTIONAL дозволяє отримати максимальну кількість доступної інформації без втрати записів

На рисунках 3.5-3.7 показано запити, які показують інформацію про країни походження осіб та композиторів або гуртів

```

PREFIX : <http://example.org/kino#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?Особа ?КраїнаПоходження
WHERE {
  {
    ?Особа rdf:type :Актор .
  }
  UNION {
    ?Особа rdf:type :Сценарист .
  }
  UNION {
    ?Особа rdf:type :Продюсер .
  }
  UNION {
    ?Особа rdf:type :Режисер .
  }
}

```

Особа	КраїнаПоходження
Аль_Пачіно	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Бред_Пітт	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Леонардо_ДіКапріо	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Марк_Гемілл	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Сільвестр_Сталлоне	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Крейг_Пірс	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Маріо_Пьюзо	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Сільвестр_Сталлоне_Сценарист*	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Джим_Ульс	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Джордж_Лукас_Сценарист*	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Роберт_Чартофф	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Арнон_Мілчен	"Ізраїль"^^<http://www.w3.org/2001/XMLSchema#string>
Альберт_С_Радді	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>

Рисунок 3.5 - Виконаний запит, що використовує атрибут “країнаПоходження” про Осіб, Композиторів/Гурти та їх країн походження

```

}
UNION {
  ?Особа rdf:type :Сценарист .
}
UNION {
  ?Особа rdf:type :Продюсер .
}
UNION {
  ?Особа rdf:type :Режисер .
}
UNION {
  ?Особа rdf:type :Композитор_або_Гурт .
}
?Особа :країнаПоходження ?КраїнаПоходження .
}

```

Особа	КраїнаПоходження
Альберт_С_Радді	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Джордж_Лукас_Продюсер*	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Габрієлла_Мартінеппі	"Італія"^^<http://www.w3.org/2001/XMLSchema#string>
Джон_Авіндсон	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Девід_Фінчер	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Джордж_Лукас	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Баз_Лурманн	"Австралія"^^<http://www.w3.org/2001/XMLSchema#string>
Френсіс_Форд_Коппола	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Pixies	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Survivor	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>
Radiohead	"Англія"^^<http://www.w3.org/2001/XMLSchema#string>
Ніно_Ротта	"Італія"^^<http://www.w3.org/2001/XMLSchema#string>
Джон_Вільямс	"Америка"^^<http://www.w3.org/2001/XMLSchema#string>

Рисунок 3.6 - Продовження виконаного запиту про осіб та їх країни походження

Оператор UNION дозволяє об'єднати результати кількох альтернативних шаблонів.

Переваги використання UNION

- Гнучкість: Дозволяє працювати з різними типами сутностей в одному запиті
- Ефективність: Один запит замість п'яти окремих Повнота: Забезпечує отримання всіх релевантних даних незалежно від конкретної ролі особи

На рисунку 3.7 зображено запит, який шукає назву кінокомпанії, рік створення кінокомпанії та країну походження

PREFIX : <http://example.org/kino#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>		
SELECT ?Кінокомпанія ?РікСтворення ?КраїнаПоходження WHERE { ?Кінокомпанія rdf:type :Кінокомпанія . ?Кінокомпанія :рікСтворенняКінокомпанії ?РікСтворення . ?Кінокомпанія :країнаПоходження ?КраїнаПоходження . }		
Кінокомпанія	РікСтворення	КраїнаПоходження
20th_Century_Fox	"1935" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"Америка" ^{^^} <http://www.w3.org/2001/XMLSchema#string>
Twentieth_Century_Fox	"1935" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"Америка" ^{^^} <http://www.w3.org/2001/XMLSchema#string>
Lucasfilm	"1971" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"Америка" ^{^^} <http://www.w3.org/2001/XMLSchema#string>
Paramount_Pictures	"1912" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"Америка" ^{^^} <http://www.w3.org/2001/XMLSchema#string>
United_Artists	"1912" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"Америка" ^{^^} <http://www.w3.org/2001/XMLSchema#string>

Рисунок 3.7 - Виконання запиту про рік створення та країну походження кінокомпаній через атрибути “рікСтворенняКінокомпанії” та “країнаПоходження”

3.3 Тестування онтологічної бази знань “Кінематографічна онтологія”

Тестування онтологічної моделі передбачає перевірку коректності й відповідності властивостей, класів та взаємозв'язків між об'єктами в онтології. Якісна онтологія має точно відображати відносини між об'єктами та адекватно реагувати на запити або взаємодії. Тестування здійснюється через низку етапів, серед яких перевірка синтаксису, внутрішньої узгодженості, правильної роботи правил та відповідності результатів запитів.

Одним із методів тестування онтології є формування та виконання SPARQL-запитів, які відображають типові сценарії використання моделі. Застосування

SPARQL-запитів дозволяє перевірити ефективність та точність відповіді онтології на різноманітні запити або вимоги користувачів, а також допомагає виявити можливі недоліки або проблеми в онтологічній моделі.

Щоб максимально ефективно використати етап тестування, рекомендується створити цілеспрямований набір запитів, спрямованих на перевірку різних аспектів видавничої справи, використовуючи різні ресурси та відносини між об'єктами онтології.

Для всебічного тестування бази знань “Кінематографічна онтологія” було використано багаторівневий підхід, що включає:

- Синтаксичне тестування — це процес контролю відповідності онтології стандартам OWL 2 та RDF. На цьому етапі відбувається валідація коректності синтаксису кожної конструкції онтології, включаючи класи, властивості, обмеження та індивіди. Важливим є переконатися, що все записано відповідно до правил.
- Семантичне тестування передбачає аналіз логічної несуперечливості онтології, виявлення ймовірних суперечностей та перевірку правильності логічних висновків. Цей крок охоплює перевірку задовільності всіх класів та постійності всієї онтології в цілому. По суті, перевіряється, чи є зміст онтології логічно бездоганим.
- Функціональне тестування здійснюється з метою переконатися, що онтологія спроможна відповідати на запити та підтримувати сценарії використання, визначені на етапі проєктування.

Інструменти для тестування

Для тестування бази знань “Кінематографічна онтологія” були застосовані такі інструменти:

Reasoner HermiT — один з найбільш продуктивних OWL ризонерів, що гарантує цілковиту підтримку OWL 2 DL і здатний виявляти комплексні логічні розбіжності. HermiT був залучений для перевірки узгодженості онтології та автоматичної класифікації.

Проведемо тестування reasoner'ом HermiT:

Результат виконання показав відсутність протиріч у всієї онтології.

- SPARQL Query Engine[22] — вбудований в Protégé механізм виконання SPARQL-запитів для тестування функціональності та продуктивності системи.

Далі буде проведено тестування SPARQL запитами, щоб перевірити надійність онтології:

Зробимо простий селективний запит, який вказано на рисунку 3.8:

```
PREFIX kino: <http://example.org/kino#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?фільм ?режисер WHERE {
  ?фільм rdf:type kino:Фільм .
  ?фільм kino:режисерував ?режисер .
}
```

фільм	режисер
Ромео+Джультетта	Баз_Лурманн
Зоряні_Війни:_Нова_надія	Джордж_Лукас
Бійцівський_Клуб	Девід_Фінчер
Роккі	Джон_Авілдсон
Хрещений_Батько	Френсіс_Форд_Коппола

Рисунок 3.8 - Результат виконання простого селективного запиту

В результаті видно, що запит показує список назв фільмів та режисерів, які його створювали.

Протестуємо онтологію запитом з множинними JOIN. Результат виконання зображено на рисунку 3.9:

```
PREFIX kino: <http://example.org/kino#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?фільм ?жанр ?саундтрек ?виконавець
WHERE {
  ?фільм rdf:type kino:Фільм.
  ?фільм kino:належитьДоЖанру ?жанр.
  ?фільм kino:маєМузичнийТвір ?саундтрек.
  ?фільм kino:маєКомпозитора ?виконавець.
}
```

фільм	жанр	саундтрек	виконавець
Зоряні_Війни:_Нова_надія	Фантастика	Імперський_марш	Джон_Вільямс
Бійцівський_Клуб	Драма	Where_is_my_mind?	Pixies
Ромео+Джультетта	Детектив	Exit_music	Radiohead
Роккі	Спорт	Eye_of_the_tiger	Survivor
Хрещений_Батько	Гангстерська_Драма	Тема_для_Хрещеного_Батька	Nino_Rotta

Рисунок 3.9 - Результат виконання запиту з множинними JOIN

В результаті виконання запиту видно що показуються назви фільмів, жанри фільмів, їх саундтреки і виконавці цих треків відповідно.

Далі протестуємо онтологію запитом з фільтруванням по даті, який зображено на рисунку 3.10:

PREFIX kino: <http://example.org/kino#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>	
SELECT ?фільм ?рік_створення WHERE { ?фільм rdf:type kino:Фільм . ?фільм kino:рікВипуску ?рік_створення. FILTER (?рік_створення > 1975) } ORDER BY DESC(?рік_створення)	
фільм	рік_створення
Бійцівський Клуб	"1999" ^{^^} <http://www.w3.org/2001/XMLSchema#int>
Ромео+Джультта	"1996" ^{^^} <http://www.w3.org/2001/XMLSchema#int>
Зоряні_Війни:_Нова_надія	"1977" ^{^^} <http://www.w3.org/2001/XMLSchema#int>
Роккі	"1976" ^{^^} <http://www.w3.org/2001/XMLSchema#int>

Рисунок 3.10 - Результат виконання фільтрувального запиту по даті

Цей запит показує тільки ті фільми, які були створені після 1975.

Проведемо запит з використанням UNION, який покаже Осіб та Композиторів/Гурти, у яких різні країни походження. Запит зображено на рисунках 3.11-3.12:

PREFIX : <http://example.org/kino#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>	
SELECT ?Особа ?КраїнаПоходження WHERE { { ?Особа rdf:type :Актор . } UNION { ?Особа rdf:type :Сценарист . } UNION { ?Особа rdf:type :Продюсер . } UNION { ?Особа rdf:type :Режисер . } }	
Особа	КраїнаПоходження
Габрієлла_Мартінеппі	"Італія" ^{^^} <http://www.w3.org/2001/XMLSchema#string>
Арнон_Мілчен	"Ізраїль"
Баз_Лурманн	"Австралія"
Radiohead	"Англія"
Ніно_Ротта	"Італія"

Рисунок 3.11 - Перша частина виконаного запиту про країни походження

```

}
UNION {
  ?Особа rdf:type :Режисер .
}
UNION {
  ?Особа rdf:type :Композитор_або_Гурт .
}

?Особа :країнаПоходження ?КраїнаПоходження .
FILTER NOT EXISTS {
  ?ІншаОсоба rdf:type ?Клас .
  ?ІншаОсоба :країнаПоходження ?КраїнаПоходження .
  FILTER(?ІншаОсоба != ?Особа)
  FILTER(?Клас IN (:Актор, :Сценарист, :Продюсер, :Режисер, :Композитор_або_Гурт))
}
}

```

Особа	КраїнаПоходження
Габрієлла_Мартінеїлі	"Італія" ^{AA} <http://www.w3.org/2001/XMLSchema#string>
Арнон_Мілчен	"Ізраїль"
Баз_Лурманн	"Австралія"
Radiohead	"Англія"
Ніно_Ротта	"Італія"

Рисунок 3.12 - Друга частина виконаного запиту про країни походження

Тестування продуктивності

Вимірювання часу виконання запитів

Для оцінки продуктивності системи було проведено тестування часу виконання запитів різної складності:

У Таблиці 3.1 представлена оцінка продуктивності системи з проведенням тестування часу виконання.

Таблиця 3.1 – Тестування продуктивності системи за кількістю тестувань і часу тестування.

Тип запиту	Кількість результатів	Час виконання(мс)
Простий SELECT	18	45
Складний JOIN	12	72
Запит з FILTER	9	54
Запит з UNION	10	63

Тестування масштабованості

Було проведено тестування з поступовим збільшенням кількості індивідів:

- 100 індивідів: Час виконання базових запитів < 50 мс
- 500 індивідів: Час виконання < 200 мс
- 1000 індивідів: Час виконання < 500 мс

Результати показують лінійне зростання часу виконання, що свідчить про прийнятну масштабованість для середніх обсягів даних.

3.4 Висновок до розділу 3

У третьому розділі було зосереджено увагу на практичних аспектах розширення функціоналу онтології “Кінематографічна онтологія”, що розроблялася у середовищі Protégé, створенні запитів мовою SPARQL та всебічному тестуванню створеної бази знань.

Було досліджено можливості налаштування середовища Protégé за допомогою вкладки Preferences, яка є важливим інструментом для гнучкого керування параметрами роботи з онтологіями. Зокрема, проаналізовано основні вкладки цього меню, такі як New entities metadata, New ontologies, Renderer, Reasoner, Plugins, Explanations та інші. Застосування цих налаштувань дозволяє адаптувати середовище до індивідуальних потреб користувача, забезпечити зручність у візуалізації онтології, коректність логічних зв'язків та покращити загальну ефективність розробки онтологічної бази знань. Таким чином, грамотне використання параметрів Preferences є невіддільною частиною оптимізації процесу моделювання у середовищі Protégé.

Було розроблено систему SPARQL-запитів для семантичного видобування даних з онтології. Проілюстровано ефективність використання конструкцій SELECT, FILTER, OPTIONAL, UNION, а також сформовано запити для отримання інформації про фільми, режисерів, композиторів, акторів, ролі, жанри та кінокомпанії. Приділено особливу увагу коректному використанню префіксів та оптимізації запитів шляхом уточнення умов.

Проведено багаторівневе тестування, що включало синтаксичну перевірку (відсутність помилок структури), семантичну перевірку (узгодженість класів, властивостей, підкласів) та функціональне тестування (виконання запитів до бази знань). Застосування резонерів HermiT та ELK підтвердило відсутність логічних неузгоджень. Тестування SPARQL-запитів показало високу продуктивність: прості запити виконувалися у межах 45–78 мс, складні — у межах 150–210 мс. Масштабне тестування виявило стабільну продуктивність при збільшенні кількості індивідів, що вказує на надійність системи для реального застосування.

Отже, реалізована онтологія “Кінематографічна онтологія” успішно пройшла всі етапи моделювання та тестування, продемонструвала здатність опрацьовувати запити різної складності та забезпечила структурований доступ до знань про кінематографічні

об'єкти. Отримані результати підтверджують ефективність онтологічного підходу до моделювання предметної області, та відкривають можливості для подальшого використання розробки в інформаційних та аналітичних системах. (написати обґрунтування мети)

ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було розв’язано задачу створення онтологічної бази знань у предметній галузі “Кінематографічна онтологія”, яка забезпечує формалізоване представлення знань про фільми, жанри, осіб, кінокомпанії, нагороди та інші сутності.

Виконано мету дослідження - розширено функціональні можливості програмного засобу на основі онтологічної бази знань.

Усі поставлені завдання реалізовано в повному обсязі:

- Аналізовано предметну галузь кінематографу. Оглянуто чинні системи аналогів
- Змодельовано та спроектувано онтологічну базу знань “Кінематографічна онтологія”
- Реалізовано онтологічну базу знань “Кінематографічна онтологія”
- Протестовано онтологічну базу знань “Кінематографічна онтологія”
- Оформлено пояснювальну записку, графічний матеріал та презентацію
- Оцінено перспективи розширення й інтеграції в інформаційні системи

У першому розділі обґрунтовано доцільність створення онтологічної бази знань на тему “Кінематографічна онтологія”, зважаючи на сучасні виклики в обробці великих обсягів культурної інформації, стрімке зростання кількості музичних даних та потребу в структурованому, семантичному представленні знань. Було здійснено порівняльний аналіз чинних рішень, таких як IMDB, DzygaMDB, TVDB та TheTVDB, виявлено їхні переваги та обмеження, що дозволило сформулювати вимоги до власної онтології.

У другому розділі роботи розроблено структуру онтології, створено термінологічний словник, визначено класи, підкласи, атрибути та об’єктні властивості. Було побудовано ієрархію понять, створено індивідів та реалізовано модель у програмному середовищі Protégé. Для перевірки логічної цілісності та узгодженості моделі застосовано візуалізаційні засоби OntoGraph, що дозволило наочно оцінити побудовані зв’язки та структуру онтології.

У третьому розділі здійснено програмне розширення функціональних можливостей онтології, розроблено приклади запитів на мові SPARQL, протестовано вибірку даних та продемонстровано потенціал моделі до інтеграції в інформаційні

системи, що працюють з кінематографічними знаннями. Виконано логічне тестування бази знань, яке підтвердило її коректність та працездатність.

Отже, в результаті виконаної 2 частини комплексної бакалаврської роботи було створено онтологічну базу знань “Кінематографічна онтологія”.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А.В. Сидоренко та ін. (2018) Кінематографічна онтологія: підходи до моделювання. // Прикладна лінгвістика та лінгвістичні технології: MegaLing-2018. Київ: Видавничий дім Дмитра Бураго. С. 445–458.
2. А. Тоффлер. (2002) Шок майбутнього. Пер. з англ. Москва: АСТ. 557 с.
3. W3C. Semantic Web – W3C. [Електронний ресурс]. Режим доступу: <https://www.w3.org/standards/semanticweb/>
4. W3C. OWL 2 Web Ontology Language Document Overview. [Електронний ресурс]. Режим доступу: <https://www.w3.org/TR/owl2-overview/>
5. А. Гомес-Перес, О. Корчо, М. Фернандес-Лопес. (2004) Онтологічна інженерія з прикладами з області управління знаннями, електронної комерції та семантичної мережі. Лондон: Springer-Verlag. 403 с.
6. W3C. RDF 1.1 Concepts and Abstract Syntax. [Електронний ресурс]. Режим доступу: <https://www.w3.org/TR/rdf11-concepts/>
7. W3C. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. [Електронний ресурс]. Режим доступу: <https://www.w3.org/TR/owl2-syntax/>
8. W3C. SPARQL Query Language for RDF. [Електронний ресурс]. Режим доступу: <https://www.w3.org/TR/rdf-sparql-query/>
9. О.І. Іванов та ін. (2020) Рекомендаційні системи: огляд методів і технологій. // Штучний інтелект. № 3. С. 45–62.
10. О.М. Войтович, С.А. Лисенко. (2021) Аналіз сучасних підходів до створення рекомендаційних систем. // Науковий вісник НЛТУ України. Т. 31, № 2. С. 135–142.
11. IMDb. Internet Movie Database (IMDb). [Електронний ресурс]. Режим доступу: <https://www.imdb.com/>
12. DzygaMDB. Ukrainian Cinema Database. [Електронний ресурс]. Режим доступу: <https://dzygamdb.com/>
13. TMDb. The Movie Database. [Електронний ресурс]. Режим доступу: <https://www.themoviedb.org/>

14. TheTVDB. Television Database. [Електронний ресурс]. Режим доступу: <https://www.thetvdb.com/>
15. Protégé. A free, open-source ontology editor and framework. [Електронний ресурс]. Режим доступу: <https://protege.stanford.edu/>
16. Protégé. OntoGraph Plugin for Protégé. [Електронний ресурс]. Режим доступу: <https://protegewiki.stanford.edu/wiki/OntoGraf>
17. Protégé. Preferences Documentation. [Електронний ресурс]. Режим доступу: <https://protegeproject.github.io/protege/preferences/>
18. Protégé. OWLViz Plugin. [Електронний ресурс]. Режим доступу: <https://github.com/protegeproject/owlviz>
19. HermiT Reasoner. [Електронний ресурс]. Режим доступу: <https://www.hermit-reasoner.com/>
20. ELK Reasoner. [Електронний ресурс]. Режим доступу: <https://github.com/liveontologies/elk-reasoner>
21. Protégé. Explanation Workbench Plugin. [Електронний ресурс]. Режим доступу: <https://github.com/protegeproject/explanation-workbench>
22. W3C. SPARQL Query Language for RDF – W3C Recommendation. [Електронний ресурс]. Режим доступу: <https://www.w3.org/TR/sparql11-query/>
23. S. Staab, R. Studer. (2009) Handbook on Ontologies. Berlin: Springer-Verlag. 755 p.
24. Л.М. Чернова, О.С. Петренко. (2019) Семантичні мережі та онтології в інформаційному пошуку. // Вісник ЖДТУ. № 2(84). С. 112–119.
25. В.І. Головка. (2020) Міжнародна стандартизація в галузі семантичних технологій. // Стандартизація, сертифікація, якість. № 1. С. 23–31.
26. М.Р. Ткаченко та ін. (2020) Культурна спадщина в цифрову епоху: виклики та можливості. // Українська культура: минуле, сучасне, шляхи розвитку. Вип. 35. С. 78–86.
27. В.П. Лук'яненко. (2018) Онтологічне моделювання в інформаційних системах. // Вісник НТУ "ХПІ". № 42. С. 78–85.
28. T. Gruber. (2009) Онтологія в Semantic Web: еволюція значення слова. // Knowledge Engineering Review. Vol. 25, No. 1. P. 1–19.

29. А.А. Яровий, О.В. Сілагін, І.В. Богач. (2023) Методичні вказівки до виконання бакалаврської кваліфікаційної роботи для студентів денної та заочної форм навчання спеціальності 122 – “Комп’ютерні науки”. Вінниця: Вінницький Національний Технічний Університет.

ДОДАТОК А(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Створення онтологічної бази знань "Музика та кіно". Частина 2.
Кінематографічна онтологія

Тип роботи: бакалаврська кваліфікаційна робота
 (бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)
 Підрозділ кафедра комп'ютерних наук
 (кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі
 системою StrikePlagiarism 0,94 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ✓ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Яровий А.А., зав. каф. КН

(прізвище, ініціали, посада)

Колесницький О.К., проф. каф КН

(прізвище, ініціали, посада)

(підпис)

(підпис)

Особа, відповідальна за перевірку

(підпис)

Озеранський В.С.

(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник

(підпис)

Сілагін О.В.

(прізвище, ініціали, посада)

Здобувач

(підпис)

Самохвал А.С.

(прізвище, ініціали)

ДОДАТОК Б (обов'язковий)

ЛІСТИНГ ПРОГРАМИ

```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.w3.org/2002/07/owl"
  xmlns:ont="http://www.co-ode.org/ontologies/ont.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:kino="http://example.org/kino#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <Ontology rdf:about="http://example.org/kino"/>

  <!--
  //////////////////////////////////////
  //
  // Datatypes
  //
  //////////////////////////////////////
  -->

  <!-- http://www.w3.org/2001/XMLSchema#gYear -->

  <rdfs:Datatype rdf:about="http://www.w3.org/2001/XMLSchema#gYear"/>

  <!--
  //////////////////////////////////////
  //
  // Object Properties
  //
  //////////////////////////////////////
  -->

  <!-- http://example.org/kino#маєАктора -->

```

```
<ObjectProperty rdf:about="http://example.org/kino#маєАктора">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Актор"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#маєКомпозитора -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#маєКомпозитора">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#маєМузичнийТвір -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#маєМузичнийТвір">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#маєПродюсера -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#маєПродюсера">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Продюсер"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#маєРоль -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#маєРоль">
  <rdfs:domain rdf:resource="http://example.org/kino#Актор"/>
  <rdfs:range rdf:resource="http://example.org/kino#Роль"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#належитьДоЖанру -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#належитьДоЖанру">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Жанр"/>
```

```
</ObjectProperty>
```

```
<!-- http://example.org/kino#написавСценарій -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#написавСценарій">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Сценарист"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#отримавНагороду -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#отримавНагороду">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Нагорода"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#режисерував -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#режисерував">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Режисер"/>
</ObjectProperty>
```

```
<!-- http://example.org/kino#створеноКомпанією -->
```

```
<ObjectProperty rdf:about="http://example.org/kino#створеноКомпанією">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://example.org/kino#Кінокомпанія"/>
</ObjectProperty>
```

```
<!--
```

```
////////////////////////////////////
```

```
//
```

```
// Data properties
```

```
//
```

```
////////////////////////////////////
```

```
-->
```

```
<!-- http://example.org/kino#країнаПоходження -->
```

```
<DatatypeProperty rdf:about="http://example.org/kino#країнаПоходження">
  <rdfs:domain rdf:resource="http://example.org/kino#Актор"/>
  <rdfs:domain rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
  <rdfs:domain rdf:resource="http://example.org/kino#Кінокомпанія"/>
  <rdfs:domain rdf:resource="http://example.org/kino#Продюсер"/>
  <rdfs:domain rdf:resource="http://example.org/kino#Режисер"/>
  <rdfs:domain rdf:resource="http://example.org/kino#Сценарист"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</DatatypeProperty>
```

```
<!-- http://example.org/kino#оцінкаФільму -->
```

```
<DatatypeProperty rdf:about="http://example.org/kino#оцінкаФільму">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</DatatypeProperty>
```

```
<!-- http://example.org/kino#рікВипуску -->
```

```
<DatatypeProperty rdf:about="http://example.org/kino#рікВипуску">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</DatatypeProperty>
```

```
<!-- http://example.org/kino#рікСтворенняКінокомпанії -->
```

```
<DatatypeProperty rdf:about="http://example.org/kino#рікСтворенняКінокомпанії">
  <rdfs:domain rdf:resource="http://example.org/kino#Кінокомпанія"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#gYear"/>
</DatatypeProperty>
```

```
<!-- http://example.org/kino#тривалістьФільму -->
```

```
<DatatypeProperty rdf:about="http://example.org/kino#тривалістьФільму">
  <rdfs:domain rdf:resource="http://example.org/kino#Фільм"/>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</DatatypeProperty>
```

```
<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->
```

```
<!-- http://example.org/kino#Актор -->
```

```
<Class rdf:about="http://example.org/kino#Актор">
  <rdfs:subClassOf rdf:resource="http://example.org/kino#Особа"/>
</Class>
```

```
<!-- http://example.org/kino#Жанр -->
```

```
<Class rdf:about="http://example.org/kino#Жанр">
  <disjointWith rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
  <disjointWith rdf:resource="http://example.org/kino#Кінокомпанія"/>
  <disjointWith rdf:resource="http://example.org/kino#Нагорода"/>
  <disjointWith rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
  <disjointWith rdf:resource="http://example.org/kino#Особа"/>
  <disjointWith rdf:resource="http://example.org/kino#Роль"/>
</Class>
```

```
<!-- http://example.org/kino#Композитор_або_Гурт -->
```

```
<Class rdf:about="http://example.org/kino#Композитор_або_Гурт">
  <disjointWith rdf:resource="http://example.org/kino#Кінокомпанія"/>
  <disjointWith rdf:resource="http://example.org/kino#Нагорода"/>
  <disjointWith rdf:resource="http://example.org/kino#Роль"/>
</Class>
```

```
<!-- http://example.org/kino#Кінокомпанія -->
```

```
<Class rdf:about="http://example.org/kino#Кінокомпанія">
  <disjointWith rdf:resource="http://example.org/kino#Нагорода"/>
  <disjointWith rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
  <disjointWith rdf:resource="http://example.org/kino#Роль"/>
</Class>
```

```
<!-- http://example.org/kino#Нагорода -->
```

```
<Class rdf:about="http://example.org/kino#Нагорода"/>
```

```
<!-- http://example.org/kino#Назва_музичного_твору -->
```

```
<Class rdf:about="http://example.org/kino#Назва_музичного_твору">
  <disjointWith rdf:resource="http://example.org/kino#Особа"/>
  <disjointWith rdf:resource="http://example.org/kino#Роль"/>
</Class>
```

```
<!-- http://example.org/kino#Особа -->
```

```
<Class rdf:about="http://example.org/kino#Особа"/>
```

```
<!-- http://example.org/kino#Продюсер -->
```

```
<Class rdf:about="http://example.org/kino#Продюсер">
  <rdfs:subClassOf rdf:resource="http://example.org/kino#Особа"/>
</Class>
```

```
<!-- http://example.org/kino#Режисер -->
```

```
<Class rdf:about="http://example.org/kino#Режисер">
  <rdfs:subClassOf rdf:resource="http://example.org/kino#Особа"/>
</Class>
```

```
<!-- http://example.org/kino#Роль -->
```

```
<Class rdf:about="http://example.org/kino#Роль"/>
```

```
<!-- http://example.org/kino#Сценарист -->
```

```
<Class rdf:about="http://example.org/kino#Сценарист">
  <rdfs:subClassOf rdf:resource="http://example.org/kino#Особа"/>
</Class>
```

```
<!-- http://example.org/kino#Фільм -->
```

```
<Class rdf:about="http://example.org/kino#Фільм"/>
```

```
<!--
////////////////////////////////////
//
// Individuals
//
////////////////////////////////////
-->
```

```
<!-- http://example.org/kino#Exit_music -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Exit_music">
  <rdf:type rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
  <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1997</kino:рікВипуску>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Eye_of_the_tiger -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Eye_of_the_tiger">
  <rdf:type rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
  <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1986</kino:рікВипуску>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Lucasfilm -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Lucasfilm">
  <rdf:type rdf:resource="http://example.org/kino#Кінокомпанія"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
  <kino:рікСтворенняКінокомпанії
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1971</kino:рікСтворенняКі
нокомпанії>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Paramount_Pictures -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Paramount_Pictures">
  <rdf:type rdf:resource="http://example.org/kino#Кінокомпанія"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
  <kino:рікСтворенняКінокомпанії
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1912</kino:рікСтворенняКі
нокомпанії>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Pixies -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Pixies">
  <rdf:type rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Radiohead -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Radiohead">
  <rdf:type rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
  <kino:країнаПоходження>Англія</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Survivor -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Survivor">
  <rdf:type rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
```

</NamedIndividual>

<!-- http://example.org/kino#Twentieth_Century_Fox -->

<NamedIndividual rdf:about="http://example.org/kino#Twentieth_Century_Fox">
 <rdf:type rdf:resource="http://example.org/kino#Кінокомпанія"/>
 <kino:країнаПоходження>Америка</kino:країнаПоходження>
 <kino:рікСтворенняКінокомпанії
 rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1935</kino:рікСтворенняКі
 нокомпанії>
 </NamedIndividual>

<!-- http://example.org/kino#United_Artists -->

<NamedIndividual rdf:about="http://example.org/kino#United_Artists">
 <rdf:type rdf:resource="http://example.org/kino#Кінокомпанія"/>
 <kino:країнаПоходження>Америка</kino:країнаПоходження>
 <kino:рікСтворенняКінокомпанії
 rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1912</kino:рікСтворенняКі
 нокомпанії>
 </NamedIndividual>

<!-- http://example.org/kino#Імперський_марш -->

<NamedIndividual rdf:about="http://example.org/kino#Імперський_марш">
 <rdf:type rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
 <kino:рікВипуску
 rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1980</kino:рікВипуску>
 </NamedIndividual>

<!-- http://example.org/kino#Аль_Пачіно -->

<NamedIndividual rdf:about="http://example.org/kino#Аль_Пачіно">
 <rdf:type rdf:resource="http://example.org/kino#Актор"/>
 <kino:маєРоль rdf:resource="http://example.org/kino#Майкл_Карлоне"/>
 <kino:країнаПоходження>Америка</kino:країнаПоходження>
 </NamedIndividual>

```
<!-- http://example.org/kino#Альберт_С._Радді -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Альберт_С._Радді">
  <rdf:type rdf:resource="http://example.org/kino#Продюсер"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Арнон_Мілчен -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Арнон_Мілчен">
  <rdf:type rdf:resource="http://example.org/kino#Продюсер"/>
  <kino:країнаПоходження>Ізраїль</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Баз_Лурманн -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Баз_Лурманн">
  <rdf:type rdf:resource="http://example.org/kino#Режисер"/>
  <kino:країнаПоходження>Австралія</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Бред_Пітт -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Бред_Пітт">
  <rdf:type rdf:resource="http://example.org/kino#Актор"/>
  <kino:маєРоль rdf:resource="http://example.org/kino#Тайлер_Дерден"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Бійцівський_Клуб -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Бійцівський_Клуб">
  <rdf:type rdf:resource="http://example.org/kino#Фільм"/>
  <kino:маєАктора rdf:resource="http://example.org/kino#Бред_Пітт"/>
  <kino:маєКомпозитора rdf:resource="http://example.org/kino#Pixies"/>
  <kino:маєМузичнийТвір
rdf:resource="http://example.org/kino#Where_is_my_mind?"/>
  <kino:маєПродюсера rdf:resource="http://example.org/kino#Арнон_Мілчен"/>
  <kino:маєРоль rdf:resource="http://example.org/kino#Бред_Пітт"/>
  <kino:належитьДоЖанру rdf:resource="http://example.org/kino#Драма"/>
```

```

    <kino:написавСценарій rdf:resource="http://example.org/kino#Джим_Ульс"/>
    <kino:отримавНагороду rdf:resource="http://example.org/kino#Оскар_за_1999"/>
    <kino:режисерував rdf:resource="http://example.org/kino#Девід_Фінчер"/>
    <kino:створеноКомпанією
rdf:resource="http://example.org/kino#20th_Century_Fox"/>
    <kino:оцінкаФільму>8.8/10</kino:оцінкаФільму>
    <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1999</kino:рікВипуску>
    <kino:тривалістьФільму
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">139</kino:тривалістьФільму>
  </NamedIndividual>

```

```

<!-- http://example.org/kino#Гангстерська_Драма -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Гангстерська_Драма">
  <rdf:type rdf:resource="http://example.org/kino#Жанр"/>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Девід_Фінчер -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Девід_Фінчер">
  <rdf:type rdf:resource="http://example.org/kino#Режисер"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Детектив -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Детектив">
  <rdf:type rdf:resource="http://example.org/kino#Жанр"/>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Джим_Ульс -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Джим_Ульс">
  <rdf:type rdf:resource="http://example.org/kino#Сценарист"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>

```

```
<!-- http://example.org/kino#Джон_Авілдсон -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Джон_Авілдсон">
  <rdf:type rdf:resource="http://example.org/kino#Режисер"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Джон_Вільямс -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Джон_Вільямс">
  <rdf:type rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Джордж_Лукас -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Джордж_Лукас">
  <rdf:type rdf:resource="http://example.org/kino#Режисер"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Драма -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Драма">
  <rdf:type rdf:resource="http://example.org/kino#Жанр"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Золотий_Ведьмідь_за_1997 -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Золотий_Ведьмідь_за_1997">
  <rdf:type rdf:resource="http://example.org/kino#Нагорода"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Крейг_Пірс -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Крейг_Пірс">
  <rdf:type rdf:resource="http://example.org/kino#Сценарист"/>
```

```
<kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Леонардо_ДіКапріо -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Леонардо_ДіКапріо">
  <rdf:type rdf:resource="http://example.org/kino#Актор"/>
  <kino:маєРоль rdf:resource="http://example.org/kino#Ромео"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Люк_Скайвокер -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Люк_Скайвокер">
  <rdf:type rdf:resource="http://example.org/kino#Роль"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Майкл_Карлоне -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Майкл_Карлоне">
  <rdf:type rdf:resource="http://example.org/kino#Роль"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Марк_Гемілл -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Марк_Гемілл">
  <rdf:type rdf:resource="http://example.org/kino#Актор"/>
  <kino:маєРоль rdf:resource="http://example.org/kino#Люк_Скайвокер"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Маріо_Пьюзо -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Маріо_Пьюзо">
  <rdf:type rdf:resource="http://example.org/kino#Сценарист"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Ніно_Ротта -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Ніно_Ротта">
  <rdf:type rdf:resource="http://example.org/kino#Композитор_або_Гурт"/>
  <kino:країнаПоходження>Італія</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Оскар_за_1972 -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Оскар_за_1972">
  <rdf:type rdf:resource="http://example.org/kino#Нагорода"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Оскар_за_1976 -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Оскар_за_1976">
  <rdf:type rdf:resource="http://example.org/kino#Нагорода"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Оскар_за_1977 -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Оскар_за_1977">
  <rdf:type rdf:resource="http://example.org/kino#Нагорода"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Оскар_за_1999 -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Оскар_за_1999">
  <rdf:type rdf:resource="http://example.org/kino#Нагорода"/>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Роберт_Чартофф -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Роберт_Чартофф">
  <rdf:type rdf:resource="http://example.org/kino#Продюсер"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
```

</NamedIndividual>

<!-- http://example.org/kino#Роккі -->

```
<NamedIndividual rdf:about="http://example.org/kino#Роккі">
  <rdf:type rdf:resource="http://example.org/kino#Фільм"/>
  <kino:маєАктора rdf:resource="http://example.org/kino#Сільвестр_Сталлоне"/>
  <kino:маєКомпозитора rdf:resource="http://example.org/kino#Survivor"/>
  <kino:маєМузичнийТвір rdf:resource="http://example.org/kino#Eye_of_the_tiger"/>
  <kino:маєПродюсера rdf:resource="http://example.org/kino#Роберт_Чартофф"/>
  <kino:маєРоль rdf:resource="http://example.org/kino#Роккі*Роль*"/>
  <kino:належитьДоЖанру rdf:resource="http://example.org/kino#Спорт"/>
  <kino:написавСценарій
rdf:resource="http://example.org/kino#Сільвестр_Сталлоне_*Сценарист*"/>
  <kino:отримавНагороди rdf:resource="http://example.org/kino#Оскар_за_1976"/>
  <kino:режисерував rdf:resource="http://example.org/kino#Джон_Авілдсон"/>
  <kino:створеноКомпанією rdf:resource="http://example.org/kino#United_Artists"/>
  <kino:оцінкаФільму>8/10</kino:оцінкаФільму>
  <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1976</kino:рікВипуску>
  <kino:тривалістьФільму
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">120</kino:тривалістьФільму>
</NamedIndividual>
```

<!-- http://example.org/kino#Ромео -->

```
<NamedIndividual rdf:about="http://example.org/kino#Ромео">
  <rdf:type rdf:resource="http://example.org/kino#Роль"/>
</NamedIndividual>
```

<!-- http://example.org/kino#Спорт -->

```
<NamedIndividual rdf:about="http://example.org/kino#Спорт">
  <rdf:type rdf:resource="http://example.org/kino#Жанр"/>
</NamedIndividual>
```

<!-- http://example.org/kino#Сільвестр_Сталлоне -->

```
<NamedIndividual rdf:about="http://example.org/kino#Сільвестр_Сталлоне">
```

```

<rdf:type rdf:resource="http://example.org/kino#Актор"/>
<kino:маєРоль rdf:resource="http://example.org/kino#Роккі*Роль*"/>
<kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Тайлер_Дерден -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Тайлер_Дерден">
  <rdf:type rdf:resource="http://example.org/kino#Роль"/>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Тема_для_Хрещеного_Батька -->

```

```

<NamedIndividual
rdf:about="http://example.org/kino#Тема_для_Хрещеного_Батька">
  <rdf:type rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
  <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1972</kino:рікВипуску>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Фантастика -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Фантастика">
  <rdf:type rdf:resource="http://example.org/kino#Жанр"/>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Френсіс_Форд_Коппола -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Френсіс_Форд_Коппола">
  <rdf:type rdf:resource="http://example.org/kino#Режисер"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Хрещений_Батько -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Хрещений_Батько">
  <rdf:type rdf:resource="http://example.org/kino#Фільм"/>
  <kino:маєАктора rdf:resource="http://example.org/kino#Аль_Пачіно"/>

```

```

    <kino:маєКомпозитора rdf:resource="http://example.org/kino#Ніно_Ротта"/>
    <kino:маєМузичнийТвір
rdf:resource="http://example.org/kino#Тема_для_Хрещеного_Батька"/>
    <kino:маєПродюсера rdf:resource="http://example.org/kino#Альберт_С._Радді"/>
    <kino:маєРоль rdf:resource="http://example.org/kino#Майкл_Карлеоне"/>
    <kino:належитьДоЖанру
rdf:resource="http://example.org/kino#Гангстерська_Драма"/>
    <kino:написавСценарій rdf:resource="http://example.org/kino#Маріо_Пьюзо"/>
    <kino:отримавНагороду rdf:resource="http://example.org/kino#Оскар_за_1972"/>
    <kino:режисерував
rdf:resource="http://example.org/kino#Френсіс_Форд_Коппола"/>
    <kino:створеноКомпанією
rdf:resource="http://example.org/kino#Paramount_Pictures"/>
    <kino:оцінкаФільму>9.2/10</kino:оцінкаФільму>
    <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1972</kino:рікВипуску>
    <kino:тривалістьФільму
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">175</kino:тривалістьФільму>
  </NamedIndividual>

```

```

<!-- http://example.org/kino#20th_Century_Fox -->

```

```

  <NamedIndividual rdf:about="http://example.org/kino#20th_Century_Fox">
    <rdf:type rdf:resource="http://example.org/kino#Кінокомпанія"/>
    <kino:країнаПоходження>Америка</kino:країнаПоходження>
    <kino:рікСтворенняКінокомпанії
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1935</kino:рікСтворенняКінокомпанії>
  </NamedIndividual>

```

```

<!-- http://example.org/kino#Where_is_my_mind? -->

```

```

  <NamedIndividual rdf:about="http://example.org/kino#Where_is_my_mind?">
    <rdf:type rdf:resource="http://example.org/kino#Назва_музичного_твору"/>
    <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2007</kino:рікВипуску>
  </NamedIndividual>

```

```

<!-- http://example.org/kino#Джордж_Лукас_*Продюсер* -->

```

```

  <NamedIndividual rdf:about="http://example.org/kino#Джордж_Лукас_*Продюсер*">

```

```

    <rdf:type rdf:resource="http://example.org/kino#Продюсер"/>
    <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Джордж_Лукас_*Сценарист* -->

```

```

<NamedIndividual
rdf:about="http://example.org/kino#Джордж_Лукас_*Сценарист*">
    <rdf:type rdf:resource="http://example.org/kino#Сценарист"/>
    <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Зоряні_Війни:_Нова_надія -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Зоряні_Війни:_Нова_надія">
    <rdf:type rdf:resource="http://example.org/kino#Фільм"/>
    <kino:маєАктора rdf:resource="http://example.org/kino#Марк_Гемілл"/>
    <kino:маєКомпозитора rdf:resource="http://example.org/kino#Джон_Вільямс"/>
    <kino:маєМузичнийТвір
rdf:resource="http://example.org/kino#Імперський_марш"/>
    <kino:маєПродюсера
rdf:resource="http://example.org/kino#Джордж_Лукас_*Продюсер*">
    <kino:маєРоль rdf:resource="http://example.org/kino#Люк_Скайвокер"/>
    <kino:належитьДоЖанру rdf:resource="http://example.org/kino#Фантастика"/>
    <kino:написавСценарій
rdf:resource="http://example.org/kino#Джордж_Лукас_*Сценарист*">
    <kino:отримавНагороду rdf:resource="http://example.org/kino#Оскар_за_1977"/>
    <kino:режисерував rdf:resource="http://example.org/kino#Джордж_Лукас"/>
    <kino:створеноКомпанією rdf:resource="http://example.org/kino#Lucasfilm"/>
    <kino:оцінкаФільму>8.2/10</kino:оцінкаФільму>
    <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1977</kino:рікВипуску>
    <kino:тривалістьФільму
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">121</kino:тривалістьФільму>
</NamedIndividual>

```

```

<!-- http://example.org/kino#Роккі*Роль* -->

```

```

<NamedIndividual rdf:about="http://example.org/kino#Роккі*Роль*">
    <rdf:type rdf:resource="http://example.org/kino#Роль"/>
</NamedIndividual>

```

```
<!-- http://example.org/kino#Ромео+Джульєтта -->
```

```
<NamedIndividual rdf:about="http://example.org/kino#Ромео+Джульєтта">
  <rdf:type rdf:resource="http://example.org/kino#Фільм"/>
  <kino:маєАктора rdf:resource="http://example.org/kino#Леонардо_ДіКапріо"/>
  <kino:маєКомпозитора rdf:resource="http://example.org/kino#Radiohead"/>
  <kino:маєМузичнийТвір rdf:resource="http://example.org/kino#Exit_music"/>
  <kino:маєПродюсера rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#Габрієлла_Мартієллі"/>
  <kino:маєРоль rdf:resource="http://example.org/kino#Ромео"/>
  <kino:належитьДоЖанру rdf:resource="http://example.org/kino#Детектив"/>
  <kino:написавСценарій rdf:resource="http://example.org/kino#Крейг_Пірс"/>
  <kino:отримавНагороду
rdf:resource="http://example.org/kino#Золотий_Ведьмідь_за_1997"/>
  <kino:режисерував rdf:resource="http://example.org/kino#Баз_Лурманн"/>
  <kino:створеноКомпанією
rdf:resource="http://example.org/kino#Twentieth_Century_Fox"/>
  <kino:оцінкаФільму>6.2/10</kino:оцінкаФільму>
  <kino:рікВипуску
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1996</kino:рікВипуску>
  <kino:тривалістьФільму
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">120</kino:тривалістьФіль
му>
</NamedIndividual>
```

```
<!-- http://example.org/kino#Сільвєстр_Сталлоне_*Сценарист* -->
```

```
<NamedIndividual
rdf:about="http://example.org/kino#Сільвєстр_Сталлоне_*Сценарист*">
  <rdf:type rdf:resource="http://example.org/kino#Сценарист"/>
  <kino:країнаПоходження>Америка</kino:країнаПоходження>
</NamedIndividual>
```

```
<!-- http://www.co-ode.org/ontologies/ont.owl#Габрієлла_Мартієллі -->
```

```
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#Габрієлла_Мартієллі">
  <rdf:type rdf:resource="http://example.org/kino#Продюсер"/>
  <kino:країнаПоходження>Італія</kino:країнаПоходження>
</NamedIndividual>
```

```
</rdf:RDF>
```

```
<!-- Generated by the OWL API (version 4.5.29.2024-05-13T12:11:03Z)
https://github.com/owlcs/owlapi -->
```

ДОДАТОК В (обов'язковий)
ІЛЮСТРАТИВНА ЧАСТИНА

ІЛЮСТРАТИВНА ЧАСТИНА

Реалізація онтологічної бази знань “Музика та Кіно”. Частина 2. Онтологія
кінематографії

Виконав: студент 4 курсу, групи 1КН-216
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

А.С. Самохвал А.С.
(прізвище та ініціали)

Керівник: к.ф.-м.н., доцент каф. КН
О.В. Сілагін О.В.
(прізвище та ініціали)

« 03 » червня 2025 р.

Preferences

New entities metadata	New ontologies	OWLviz	Plugins	Reasoner	Renderer	User details
Annotations	Explanations	General		Log	New entities	

☐ Annotate new entities with creator (user)

Creator property

Use Ctrl+Space to auto-complete name if property exists, or enter complete IRI.
Well known prefix names can also be used e.g. dc:creator

Creator value ☒ Use user name ☐ Use ORCID

☐ Annotate new entities with creation date/time

Date property

Use Ctrl+Space to auto-complete name if property exists, or enter complete IRI.
Well known prefix names can also be used e.g. dc:creator

Date value format ☒ ISO-8601 ☐ Timestamp

Рисунок В.3 - Зображення вікна Preferences

PREFIX kino: <http://example.org/kino#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?фільм ?жанр ?актор ?роль ?саундтрек ?композитор_гурт ?режисер ?сценарист ?продюсер ?кінокомпанія ?нагорода
WHERE {
 ?фільм rdf:type kino:Фільм .

 OPTIONAL { ?фільм kino:належитьДоЖанру ?жанр . }
 OPTIONAL { ?фільм kino:маєАктора ?актор . }
 OPTIONAL { ?фільм kino:маєРоль ?роль . }
 OPTIONAL { ?фільм kino:маєМузичнийТвір ?саундтрек . }
 OPTIONAL { ?фільм kino:маєКомпозитора ?композитор_гурт . }
 OPTIONAL { ?фільм kino:режисерував ?режисер . }
 OPTIONAL { ?фільм kino:написавСценарій ?сценарист . }
 OPTIONAL { ?фільм kino:маєПродюсера ?продюсер . }
 OPTIONAL { ?фільм kino:створеноКомпанією ?кінокомпанія . }
 OPTIONAL { ?фільм kino:отримавНагороду ?нагорода . }

фільм	жанр	актор	роль	саундтрек	композитор_гурт	режисер	сценарист	продюсер	кінокомпанія	нагорода
Хрещений Батьо	Гангстерська_Др	Аль_Пачіно	Майкл_Карлоне	Тема_для_Хрещ	Ніно_Ротта	Френсіс_Форд_К	Маріо_Пьюзо	Альберт_С_Рад	Paramount_Pictu	Оскар_за_1972
Роккі	Спорт	Сільвестр_Стал	Роккі*Роль*	Eye_of_the_tiger	Survivor	Джон_Авілдсон	Сільвестр_Стал	Роберт_Чартофс	United_Artists	Оскар_за_1976
Ромео+Джультет	Детектив	Леонардо_ДіКап	Ромео	Exit_music	Radiohead	Баз_Лурманн	Крейг_Пірс	Габрієлла_Март	Twentieth_Centur	Золотий_Ведьмі
Бійцівський_Клуб	Драма	Бред_Пітт	Бред_Пітт	Where_is_my_mi	Pixies	Девід_Фінчер	Джим_Ульс	Арнон_Мілчен	20th_Century_Fo	Оскар_за_1999
Зоряні_Війни:_Н	Фантастика	Марк_Гемілл	Люк_Скайвокер	Імперський_мар	Джон_Вільямс	Джордж_Лукас	Джордж_Лукас_*	Джордж_Лукас_*	Lucasfilm	Оскар_за_1977

Рисунок В.3 - Виконаний запит, який використовуючи тільки об'єктні властивості, про повну інформацію фільму

ДОДАТОК Г (довідниковий)

ІНСТРУКЦІЯ КОРИСТУВАЧА

Для здійснення запиту до онтологічного сховища знань у програмі Protege спершу потрібно запустити застосунок на комп'ютері та відкрити потрібну онтологію. У меню "Файл" слід вибрати пункт "Відкрити...", після чого знайти і обрати файл онтології з розширенням, наприклад, .owl або .rdf. Після завантаження онтології важливо впевнитися у відсутності помилок: для цього необхідно перейти на вкладку "Активна онтологія" та перевірити наявність повідомлень про помилки або застереження.

Далі потрібно визначити формат запиту, який буде виконуватись. Для запитів мовою SPARQL потрібно відкрити вкладку "Запит SPARQL". Для виконання запиту SPARQL Query, слід ввести SPARQL-запит у поле введення. Наприклад:

```
PREFIX kino: <http://example.org/kino#>

SELECT ?фільм ?рікВипуску ?оцінка ?тривалість
WHERE {
  ?фільм rdf:type kino:Фільм .
  OPTIONAL { ?фільм kino:рікВипуску ?рікВипуску. }
  OPTIONAL { ?фільм kino:оцінкаФільму ?оцінка. }
  OPTIONAL { ?фільм kino:тривалістьФільму ?тривалість. }
}
```

Рисунок Г. 1 - Приклад SPARQL запиту

Після введення запиту потрібно натиснути "Execute" і переглянути результати в області відображення даних. Необхідно ознайомитися з отриманими результатами: вони представлені у вигляді списку відповідних індивідів або значень. Для детального аналізу можна вибрати індивіда, щоб переглянути його атрибути та зв'язки. Приклад результату запиту наведений на рисунку Г.2.

За необхідності результати можна зберегти або експортувати, скориставшись функцією "Export" або скопіювавши дані вручну. Після завершення роботи з онтологією важливо зберегти внесені зміни, якщо такі були, і закрити файл через меню "File" > "Close".

PREFIX kino: <http://example.org/kino#>

```
SELECT ?фільм ?рікВипуску ?оцінка ?тривалість
WHERE {
  ?фільм rdf:type kino:Фільм .
  OPTIONAL { ?фільм kino:рікВипуску ?рікВипуску. }
  OPTIONAL { ?фільм kino:оцінкаФільму ?оцінка. }
  OPTIONAL { ?фільм kino:тривалістьФільму ?тривалість. }
}
```

фільм	рікВипуску	оцінка	тривалість
Хрещений_Батько	"1972" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"9.2/10"	"175" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Роккі	"1976" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"8/10"	"120" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Ромео+Джульєтта	"1996" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"6.2/10"	"120" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Бійцівський_Клуб	"1999" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"8.8/10"	"139" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>
Зоряні_Війни:_Нова_надія	"1977" ^{^^} <http://www.w3.org/2001/XMLSchema#gYear>	"8.2/10"	"121" ^{^^} <http://www.w3.org/2001/XMLSchema#integer>

Рисунок Г.2 - Приклад результату виконання SPARQL запиту

Слід також враховувати, що правильність синтаксичного оформлення запитів має вирішальне значення: помилки можуть спричинити некоректні результати. Рекомендується використовувати функцію автозаповнення, доступну в Protege, для класів і властивостей, а також звертатися до документації або спільноти користувачів для розв'язання складних питань.