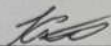


Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

Бакалаврська кваліфікаційна робота на тему:
РОЗРОБКА ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ ІЗ ТЕОРІЇ ШАХІВ

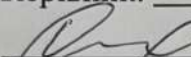
Виконала: студентка 4 курсу, групи ЗКН-216
спеціальності 122 Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)

 Степанюк К.М.

(прізвище та ініціали)

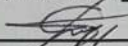
Керівник: к.т.н., доцент каф. КН

 Сілагін О. В.

(прізвище та ініціали)

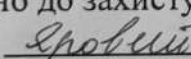
« 04 » червня 2025 р.

Рецензент: к.т.н., доцент каф. АІТ

 Кривогубченко С.Г.

(прізвище та ініціали)

« 05 » червня 2025 р.

Допущено до захисту Зав.
кафедри  А.А.

« 06 » червня 2025 р.

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

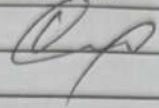
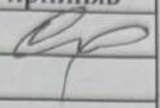
ЗАТВЕРДЖУЮ

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

 « 05 » травня 2025 р.

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Степанюк Карині Миколаївні

1. Тема роботи: Розробка онтологічної бази знань із теорії шахів.
керівник роботи: Сілагін Олексій Віталійович, к.т.н., доц.
затверджені наказом вищого навчального закладу “20” серпня 2025 року № 97
2. Термін подання студентом роботи 30 травня 2025 р.
3. Вихідні дані до роботи :
Кількість основних класів – не менше 5; кількість підкласів – не менше 20;
кількість індивідів – не менше 20; ступінь ієрархії – не менше 2.
4. Зміст текстової частини (перелік питань, які потрібно розробити):
Вступ, аналіз предметної області онтологічного моделювання баз знань,
моделювання онтологічної бази знань з теорії шахів, створення SPARQL запитів
для пошуку інформації в онтологічній базі знань з теорії шахів, висновки, перелік
використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):
Схема здійснення альфа-бета-відсічення слабких ходів у дебютному дереві, граф
онтології предметної області “Теорія шахів”, граф ієрархії класів онтології
предметної області “Теорія шахів”, граф індивідів онтології предметної області
“Теорія шахів”, створений SPARQL запит, результат виконання SPARQL запиту.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Сілагін О.В., к.т.н., доц. каф. КН		

7. Дата видачі завдання 05 травня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчен ня	
1	Обґрунтування актуальності дослідження предметної області з теорії шахів та онтологій	05.05.25	11.05.25	
2	Моделювання та проектування онтологічної бази знань з теорії шахів	11.05.25	20.05.25	
3	Реалізація бази знань та створення SPARQL запитів для пошуку інформації в онтологічній базі знань з теорії шахів	20.05.25	29.05.25	
4	Апробація та/або впровадження результатів дослідження	30.05.25	31.05.25	
5	Розробка інструкції користувача	01.06.25	02.06.25	
6	Оформлення матеріалів до захисту БКР	03.06.25	04.06.25	

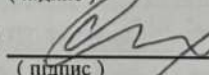
Студент


(підпис)

Степанюк К.М.

(прізвище та ініціали)

Керівник проекту (роботи)


(підпис)

Сілагін О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 90 сторінок формату А4, на яких є 22 рисунки і 3 таблиці. У списку використаних джерел зазначено 16 найменувань.

Метою роботи є розширення функціональних можливостей інформаційної технології онтологічного моделювання бази знань у сфері теорії шахів.

Робота присвячена створенню онтологічної бази знань із теорії шахів. У ній досліджено взаємозв'язок між сучасними інформаційними технологіями та шаховою теорією, а також окреслено історичні передумови й розвиток онтологічного підходу в галузі ІТ. Було змодельовано предметну область шахової гри з використанням засобів онтологічного проектування.

Реалізацію здійснено в середовищі Protégé, де сформовано логічно узгоджену структуру класів і індивідів. Для перевірки функціональності побудованої моделі створено та виконано SPARQL-запити, що засвідчили коректність структури та вмісту онтології. Отримана база знань є логічно цілісною, зручною для використання у навчальних цілях і має потенціал для подальшого розширення й інтеграції в інші інтелектуальні системи.

Ключові слова: онтологія, теорія шахів, база знань, Protégé.

ABSTRACT

The bachelor's qualification work consists of 90 pages of A4 format, on which there are 22 figures and 3 tables. The list of used sources includes 16 items.

The purpose of the work is to expand the functional capabilities of the information technology of ontological modeling of the knowledge base in the field of chess theory.

The work is devoted to the creation of an ontological knowledge base in chess theory. It explores the relationship between modern information technologies and chess theory, and also outlines the historical prerequisites and development of the ontological approach in the field of IT. The subject area of the chess game was modeled using ontological design tools.

The implementation was carried out in the Protégé environment, where a logically consistent structure of classes and individuals was formed. To verify the functionality of the constructed model, SPARQL queries were created and executed, which confirmed the correctness of the structure and content of the ontology. The resulting knowledge base is logically coherent, convenient for use for educational purposes, and has the potential for further expansion and integration into other intellectual systems.

Keywords: ontology, chess theory, knowledge base, Protégé.

ЗМІСТ

ВСТУП	3
1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ СТВОРЕННЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ З ТЕОРІЇ ШАХІВ.....	5
1.1 Інформаційні технології і теорія шахів.....	5
1.2 Історія виникнення онтології у сфері інформаційних технологій.....	8
1.3 Огляд відомих технологій для вивчення теорії шахів.....	12
1.4 Формулювання вимог та постановка задач на створення онтологічної бази знань з теорії шахів.....	15
1.5 Висновок до розділу 1.....	16
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ З ТЕОРІЇ ШАХІВ.....	17
2.1 Обґрунтування вибору середовища для моделювання.....	17
2.2 Методологія 7 кроків і її застосування в проектуванні та моделюванні онтологічної бази знань з теорії шахів.....	21
2.3 Проектування онтологічної бази знань з теорії шахів.....	25
2.4 Висновок до розділу 2.....	38
3 РЕАЛІЗАЦІЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ З ТЕОРІЇ ШАХІВ	39
3.1 Створення індивідів в онтологічній базі знань з теорії шахів.....	39
3.2 Специфіка реалізації онтологічної бази знань з теорії шахів в середовищі Protege.....	46
3.3 Створення SPARQL запитів та тестування онтологічної бази знань із теорії шахів.....	51
3.4 Висновок до розділу 3.....	56
ВИСНОВКИ.....	57
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	61
ДОДАТОК А (обов'язковий) Протокол перевірки кваліфікаційної роботи	62
ДОДАТОК Б (обов'язковий) Лістинг програми.....	63
ДОДАТОК В (обов'язковий) Графічна частина.....	75
ДОДАТОК Г (довідниковий) Інструкція користувача.....	84
ДОДАТОК Д (довідниковий) Довідка про впровадження.....	87

ВСТУП

Актуальність. Шахи — це не лише інтелектуальна гра, яка вимагає логічного мислення, стратегічного планування та прийняття рішень, а й потужна модель для дослідження процесів навчання, ухвалення рішень та штучного інтелекту. В умовах стрімкого розвитку інформаційних технологій виникає потреба в системах, здатних ефективно структурувати, зберігати, інтерпретувати та аналізувати великі обсяги знань. У цьому контексті онтології виступають як потужний інструмент для формалізованого представлення знань у певній предметній галузі. Актуальність створення онтологічної бази знань з теорії шахів зумовлена кількома ключовими факторами. По-перше, шахи мають чітко визначену логіку, структуру та правила, що робить цю сферу ідеальною для моделювання знань у вигляді онтології. По-друге, зростає інтерес до автоматизованого навчання шахам, створення систем підтримки прийняття рішень у грі, цифрових тренерів та аналітичних інструментів. По-третє, наявність формалізованої онтології дозволяє створювати сумісні, розширювані та повторно використовувані знання, що можуть застосовуватися як у навчальних цілях, так і в розробці інтелектуальних систем.

Доцільність проєктування онтологічної моделі теорії шахів також обґрунтовується потребою в уніфікації знань, що існують у вигляді фрагментованих джерел — навчальних посібників, відеоуроків, стратегічних схем тощо. Онтологічна модель дозволяє об'єднати ці знання в єдину систему, з чіткими відношеннями між поняттями, можливістю автоматичного аналізу й побудови логічних запитів. Завдяки цьому можна не лише зберігати структуровану інформацію про шахові фігури, правила, тактичні прийоми й типові комбінації, а й здійснювати інтелектуальний пошук, виявляти закономірності та автоматично будувати рекомендації.

Метою дослідження є розширення функціональних можливостей онтологічного моделювання бази знань у сфері теорії шахів. Це дозволить отримувати нові структуровані та приховані знання про елементи гри, що

можуть бути корисними як для початківців, так і для досвідчених гравців, аналітиків або систем штучного інтелекту.

Об’єкт дослідження – процес онтологічного моделювання бази знань з теорії шахів.

Предмет дослідження – онтології та програмні засоби для реалізації онтологічної бази знань з теорії шахів.

Задачі дослідження:

- 1 проаналізувати предметну область та обґрунтувати доцільність створення онтологічної бази знань з теорії шахів;
- 2 провести аналіз та порівняння існуючих технологій;
- 3 здійснити моделювання та проектування онтологічної бази знань з теорії шахів;
- 4 здійснити програмну реалізацію онтологічної бази знань;
- 5 виконати тестування онтологічної бази знань з теорії шахів за допомогою SPARQL запитів.

Апробація роботи. Результати досліджень були апробовані на Міжнародній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «Молодь у науці: дослідження, проблеми, перспективи (МН-2025)» [1].

Публікації: За результатами бакалаврської кваліфікаційної роботи опубліковано тези доповідей на науково-технічній конференції [1].

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ СТВОРЕННЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ З ТЕОРІЇ ШАХІВ

1.1 Інформаційні технології і теорія шахів

Шахи є однією з найдавніших та найпопулярніших настільних інтелектуальних ігор, котрі зберігають свою актуальність протягом сторіч. Ця стратегічна гра не лише потребує від гравців великої обдуманості та кмітливості, але також є визнаним видом спорту. Теорія шахів охоплює вивчення дебютів, мітельшпілю і ендшпілю, а також різноманітних тактичних та стратегічних прийомів. Завдяки поширенню онлайн-платформ та турнірів, шахи стали доступними для мільйонів гравців у всьому світі, дозволяючи їм спілкуватися, змагатися та підвищувати свій рівень гри. З розвитком інформаційних технологій теоретична база шахів значно розширилася, ставши більш доступною, точною і практичною.

Основні алгоритми сучасних програм. Комп'ютерні шахові програми розглядають шахові ходи як ігрове дерево. Теоретично, вони повинні оцінювати всі позиції, які виникнуть після всіх можливих ходів, потім всі можливі ходи після цих ходів і так далі. Кожний хід одного гравця називається «вузол». Перебирання ходів продовжується, поки програма не досягає максимальної глибини пошуку або визначає, що досягнута кінцева позиція (наприклад мат чи пат). І вже на підставі оцінки позиції обирає оптимальну стратегію. У кожній позиції кількість можливих ходів гравця близько 35. Для повного аналізу чотирьох напівходів (по два ходи кожного гравця) треба дослідити близько півтора мільйона можливостей, для шести — майже два мільярди [2].

Програмісти намагаються по-різному обмежити обшир, який треба перебрати (обрізання дерева пошуку — game tree pruning). Найпопулярнішим є відсічення альфа-бета, в якому не розглядаються позиції, що мають меншу оцінку ніж уже оцінені. Найважче дати оцінку ходам у дебюті. Більшість програм використовують при цьому написані заздалегідь дебютні бібліотеки, в яких є

певна невелика кількість початкових ходів та відповідей до певного числа ходів, яке не є постійним, бо залежить від типу дебюту. Схема здійснення альфа-бета-відсічення слабких ходів зображена на рисунку 1.1

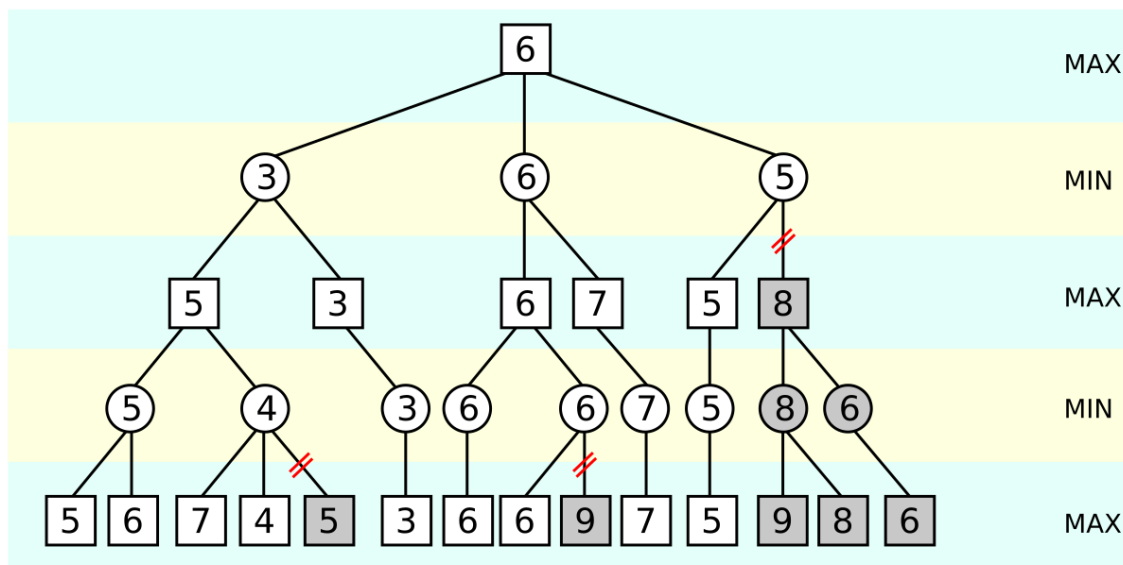


Рисунок 1.1 – Схема здійснення альфа-бета-відсічення слабких ходів [2]

На сьогоднішній день існує великий вибір шахових рушіїв, баз даних та онлайн платформ для покращення власної майстерності у шахах. Наприклад, Stockfish є відкритим рушієм, що регулярно оновлюється спільнотою, а також вже кілька років є одним із найсильніших шахових двигунів у світі. Stockfish використовує алгоритм деревоподібного пошуку, заснований на альфа-бета пошуку з кількома розробленими вручну евристичними методами, а з часів Stockfish 12 (2020) використовує ефективно оновлювану нейронну мережу як функцію оцінки. Вона представляє позиції за допомогою бітових дощок. Ще один шаховий рушій - Leela Chess Zero (LCZero) базується на нейронних мережах і самонавчанні, подібно до AlphaZero від DeepMind [3].

Також шахісти мають доступ до мільйонів професійних партій через бази даних, як-от ChessBase, lichess.org, Chess.com Explorer. Це дозволяє аналізувати партії чемпіонів світу, вивчати дебютну теорію та тенденції. До прикладу, ChessBase містить аналітичні коментарі та пошукові функції для глибокого аналізу. Lichess.org надає безкоштовну базу з фільтрацією за ело, дебютами,

роками. Існують платформи, які поєднують ІТ з теорією шахів - Chessable — тренування дебютів за допомогою повторення (спейсінг-ефект). Chess.com lessons, iChess, The Chess Dojo — онлайн-курси з теорії.

Алгоритми штучного інтелекту, зокрема нейронні мережі, використовуються для персоналізованого навчання, підбору слабких місць шахіста та формування програми тренувань. І значно вплинув на розвиток шахової теорії [4]. AlphaZero - система, яка навчилася грати в шахи без людських даних, демонструючи нові стратегічні підходи та глибоке розуміння гри. Maia модифікація AlphaZero, яка передбачає ходи людей з різними рівнями майстерності, сприяючи кращій взаємодії між ШІ та людьми. DecodeChess - платформа, яка пояснює ходи шахового рушія зрозумілою мовою, сприяючи навчанню гравців. Онлайн-платформи дозволяють відтворювати й аналізувати позиції з 3D-візуалізацією, інтерактивним поясненням теоретичних положень і автоматичним виявленням помилок.

У шаховій теорії велику роль відіграють і бази знань. База знань - це структурована система, яка зберігає не лише фактичну інформацію, але й логічні зв'язки між елементами предметної області, правила виведення знань та евристики. У контексті шахів це означає формалізацію шахових понять: дебюти, мітельшпілі, ендшпілі, тактичні мотиви, стратегічні плани. При онтологічному моделюванні створюються онтології, які описують взаємозв'язки між шаховими концепціями, що дозволяє системам ШІ краще розуміти та аналізувати гру. Також за допомогою онтологій можна розвивати навчання та аналіз, а саме надавати гравцям та тренерам інструменти для глибшого розуміння шахових позицій та стратегій.

Отже, інтеграція інформаційних технологій, баз знань та штучного інтелекту в шахову теорію відкриває нові горизонти для гравців, тренерів та дослідників. Вона сприяє глибшому розумінню гри, покращує навчальні процеси та дозволяє виявляти нові стратегічні підходи. З розвитком цих технологій шахи стають не лише грою, але й потужним інструментом для дослідження інтелектуальних процесів.

1.2 Історія виникнення онтології у сфері інформаційних технологій

Термін «онтологія» має філософське походження — він виник із намагань Аристотеля систематизувати об'єкти навколишнього світу. У філософії цим поняттям позначають систему знань про зовнішній світ (на відміну від знань про внутрішній світ людини). Інакше кажучи, онтологія — це наука про буття, про природу речей і зв'язки між ними.

Онтологія як наукове поняття виникла у філософії, але з другої половини XX століття вона почала активно розвиватися в контексті інформатики. Основний поштовх до цього надала потреба в стандартизованому та формалізованому представленні знань для комп'ютерної обробки [5].

У сфері інформаційних технологій та представлення знань, онтологія розглядається як засіб або метод для опису певної предметної області - її базових понять, властивостей і взаємозв'язків між ними.

Зі зростанням складності інформаційних систем і обсягів даних виникла необхідність у створенні структур, які б дозволяли різним програмам, базам даних і користувачам однаково розуміти значення термінів та зв'язків у певній предметній області. Таким чином, онтологія в інформатиці стала засобом опису об'єктів реального світу, їхніх властивостей та взаємозв'язків у машинно-оброблюваному вигляді. На початку 1990-х років онтології набули широкого визнання завдяки розвитку штучного інтелекту, зокрема в галузях експертних систем та логічного виведення. Визначення Тома Грубера (Gruber, 1993) — “онтологія – це явна специфікація концептуалізації” — стало базовим у цій сфері [6].

У 2000-х роках інтерес до онтологій значно зріс у зв'язку з появою Семантичної павутини (Semantic Web) — концепції, запропонованої Тімом Бернерсом-Лі. Семантична павутина передбачає використання онтологій для опису даних в Інтернеті таким чином, щоб машини могли автоматично їх інтерпретувати та обробляти. Було створено мови опису онтологій, зокрема RDF,

OWL (Web Ontology Language), які стали стандартами для побудови семантично-орієнтованих систем.

Існує багато різних визначень терміну «онтологія», іноді навіть суперечливих. Одне з найвідоміших і найуживаніших: онтологія — це явна специфікація концептуалізації, де концептуалізація означає абстрактне уявлення певної предметної області. Ще одне поширене трактування: онтологія — це спільне розуміння певної області інтересу.

У практиці онтологію часто ототожнюють із інженерією знань у штучному інтелекті, концептуальним моделюванням у базах даних, а також із моделюванням предметної області в об'єктно-орієнтованому проектуванні. На сьогодні онтологія може розглядатися як надійна семантична основа для визначення змісту; логічна теорія, яка містить словник термінів і набір тверджень, сформульованих мовою логіки; база для комунікації між людьми та комп'ютерними агентами.

Онтологія — формалізоване представлення знань про певну предметну область (середовище, світ), придатне для автоматизованої обробки. Онтологію неодмінно супроводжує деяка концепція цієї області інтересів. Найчастіше ця концепція виражається за допомогою визначення базових об'єктів (індивідуумів, атрибутів, процесів) і відношень між ними. Визначення цих об'єктів і відношень між ними зазвичай називають концептуалізацією [3].

Таке визначення онтології є узагальнюючим: онтологія — це загальноприйнята і загальнодоступна концептуалізація певної області знань (світу, середовища), яка містить базис для моделювання цієї області знань і визначає протоколи для взаємодії між агентами, які використовують знання з цієї області, і, нарешті, включає домовленості про представлення теоретичних основ даної області знань.

Онтологія служить спільним словником для фахівців, яким потрібно взаємодіяти та обмінюватися інформацією в межах певної предметної області. Вона містить формалізовані, придатні для машинного оброблення визначення

ключових понять і взаємозв'язків у цій галузі. Основні причини створення онтологій:

- забезпечення єдиного бачення структури інформації для людей і програмних агентів;
- можливість багаторазового використання вже сформульованих знань;
- явне формулювання припущень, які використовуються в моделюванні предметної області;
- розмежування предметних знань і процедур (оперативних знань);
- полегшення аналізу знань у певній галузі.

Однією з головних цілей розробки онтологій є досягнення єдиного розуміння структури знань, спільного як для людей, так і для програм. Наприклад, кілька медичних вебсайтів можуть надавати інформацію про лікування або платні послуги. Якщо вони використовують спільну онтологію медичних термінів, програмні агенти зможуть об'єднувати дані з різних ресурсів, відповідати на запити користувачів або використовувати ці дані як вхідну інформацію для інших систем.

Повторне використання знань стало одним із ключових факторів розвитку онтологій. Наприклад, концепція часу — така як інтервали, моменти, тривалість — потрібна в багатьох галузях. Якщо одна група експертів створила онтологію часу, інші дослідники можуть включити її у власні проєкти. Також можливо об'єднувати кілька менших онтологій у велику, або взяти загальноприйнятну базову (наприклад, UNSPSC) і адаптувати її під конкретну галузь.

Явне формулювання припущень, що лежать в основі моделі, дозволяє легко вносити зміни при оновленні знань. Якщо ж такі припущення «зашиті» у код, вони стають складними для розуміння й редагування, особливо для тих, хто не має програмістських навичок. Натомість чітко описані знання допомагають новачкам швидше засвоїти терміни й структуру предметної галузі [4].

Відокремлення предметних знань від операційних — ще одна ключова роль онтологій. Наприклад, можна описати задачу збирання продукту з окремих компонентів і реалізувати її у вигляді програми, яка працює незалежно від

конкретного продукту. Згодом, створивши онтологію, наприклад, комп'ютерних компонентів, можна використовувати ту саму програму для складання нестандартних ПК або навіть інших систем, наприклад, ліфтів — за умови наявності відповідної онтології.

Зазвичай розробка онтології — це не самоціль, а підґрунтя для побудови систем, що використовують ці знання. Так само, як створення набору структурованих даних для програм, онтології слугують основою для розв'язання задач, роботи додатків і агентів, які використовують як самі онтології, так і бази знань, побудовані на їх основі [7].

Сьогодні онтології застосовуються в багатьох сферах, таких як – штучний інтелект, системи управління знаннями, медицина, електронна комерція, біоінформатика, освітні системи, тощо. У сфері штучного інтелекту вони використовуються для моделювання знань і логічного виведення. В системах управління знаннями онтології допомагають ефективно організовувати корпоративну інформацію. У медицині набули поширення спеціалізовані онтології, такі як SNOMED, які забезпечують єдину термінологічну базу для медичних даних. В електронній комерції онтології сприяють уніфікації опису товарів і послуг, як-от у випадку з класифікатором UNSPSC. У біоінформатиці вони відіграють важливу роль у таких проєктах, як Gene Ontology, що систематизують знання про гени та їхні функції. Крім того, онтології застосовуються в освітніх системах, де використовуються для адаптивного навчання та структурування навчального контенту.

З розвитком машинного навчання й обробки природної мови онтології відіграють дедалі важливішу роль як основа для інтерпретації семантики даних, інтеграції інформації з різних джерел, автоматизованого аналізу та прийняття рішень. Онтології стали не лише інструментом структурування знань, а й ключовим елементом сучасних інтелектуальних систем.

1.3 Огляд відомих технологій для вивчення теорії шахів

З появою перших шахових програм (ChessBase, Fritz, Shredder, Rybka) розпочалося масове комп'ютерне зберігання партій. Разом із цим почали з'являтися так звані дебютні дерева в автоматизованому вигляді. Шахові дебютні книги в електронному варіанті працюють за принципом дебютного дерева — структури, яка зберігає послідовності ходів, починаючи з початкової позиції шахової дошки. Цей інструмент і до сьогоднішніх днів широко використовується для вивчення і підготовки дебютів як гравцями, так і тренерами. Дебютне дерево — це граф (зазвичай деревоподібної структури), де вузли — це шахові позиції (позиція після певного ходу), гілки — це можливі ходи з цієї позиції, корінь дерева — початкова позиція (старт шахової партії), шляхи — послідовності ходів, які формують варіанти дебютів [8]. Кожна гілка дерева відповідає одному з можливих ходів, а в ній зазвичай зберігається назва дебюту (наприклад, "Сицилійський захист"), статистика (скільки разів застосовувався у партіях), ефективність (виграші білих/чорних/нічий), типові суперники (за базами партій). Дебютна книга для програми ChessBase зображена на рисунку 1.2

Notation + Openings Book				
Notation	Training	Table	Score sheet	Openings Book
		N	%	Av Perf
My book 11.1		34176	57.1	2727 2770
1.e4		26260	58.5	2730 2782
1.d4		2594	49.6	2738 2764
1.h3		2514	56.1	2699 2712
1.a3		1183	58.9	2743 2765
1.Nf3		596	49.4	2674 2698
1.c4		306	47.7	2668 2706
1.Nc3		233	48.7	2735 2765
1.b3		115	39.6	2672 2645
1.f4		71	47.9	2883 2902
1.g3		71	40.1	2683 2688
1.c3		56	50.0	2708 2679
1.a4		34	39.7	2716 2693
1.e3		29	43.1	2717 2719
1.b4		25	30.0	2702 2621
1.d3		19	31.6	2667 2546
1.h4		17	32.4	2707 2642
1.Nh3		16	40.6	2715 2697
1.g4		16	31.3	2710 2628
1.f3		11	22.7	2719 2570
1.Na3		10	35.0	2703 2666

Рисунок 1.2 – Дебютна книга [9]

Незважаючи на те, що дебютна книга користується великою популярністю серед шахістів, вона також має ряд недоліків:

- лінійність та відсутність семантики.

Проблема полягає у тому, що дебютне дерево – це механічна структура без глибокого розуміння змісту. Тобто дерево знає, що після 1.e4 c5 часто грають 2.Nf3, але не "розуміє", що це Сицилійський захист або яка його ідея. Як наслідок немає узагальнень, немає зв'язків між подібними ідеями в різних дебютах;

- залежність від статистики.

Рішення про хід ґрунтується лише на частоті та результативності, а не на логіці чи позиційних ідеях. Як наслідок втрачається можливість адаптувати дебютну підготовку до стилю гравця або його цілей;

- складність масштабування та оновлення.

Кожна нова партія або варіант вимагає ручного оновлення бази. Саме через це дерева швидко старіють, особливо на аматорському рівні.

Створивши онтологічну базу знань з теорії шахових дебютів можна уникнути даних проблем. Онтологія дозволяє представити не лише ходи, а і їх значення, цілі та властивості. Наприклад клас Сицилійський_Захист має підкласи Найдорф, Дракон, Шевенінген, які поділяють спільні стратегічні риси. Як результат, машина "розуміє", що обирає гравець і чому. Можливість логічного висновку - онтологія дозволяє використовувати логічні запити (наприклад, OWL + SPARQL) для пошуку варіантів, які відповідають певним критеріям. За рахунок зв'язків між класами, оновлення однієї частини онтології не вимагає перебудови всієї бази. За рахунок чого онтологія є більш масштабованою та стійкою до змін. Крім того, така структура забезпечує можливість інтеграції з іншими системами, наприклад, з навчальними платформами або шаховими движками, що відкриває перспективи для створення інтелектуальних тренажерів. Порівняння дебютного дерева з онтологічною базою знань показано у таблиці 1.1

Таблиця 1.1 – Порівняння дебютного дерева з онтологічною базою знань

Критерій	Дебютне дерево	Онтологічна база знань
Структура	Ієрархія ходів	Семантична модель з класами та зв'язками
Змістовність	Низька	Висока (значення та контекст ходів)
Адаптація до користувача	Немає	Є (можна враховувати цілі гравця)
Оновлення	Вручну або програмно	Гнучке, семантичне
Висновки	Неможливо	Можливі (за допомогою логіки)

Для професійного шахіста наявність добре структурованої бази знань стає невід'ємною складовою успіху в конкурентному середовищі. На відміну від звичайної бази партій, база знань може включати формалізовані уявлення про стилі гри суперників, шаблони тактичних і стратегічних прийомів, правила виведення та автоматизований аналіз позицій. Зокрема, така база знань є цінним інструментом при підготовці до турнірів, вона дозволяє адаптуватися до конкретних суперників, виявляти типові помилки, обирати найефективніші дебютні схеми. Крім того, онтологічна база знань забезпечує можливість узгодженого та цілісного представлення складних шахових концепцій, що особливо важливо під час аналізу нестандартних ситуацій або рідкісних варіантів. Вона може служити основою для створення інтелектуальних систем підтримки прийняття рішень, що не лише економлять час, а й підвищують якість стратегічного планування гри. Таким чином, інтеграція онтологічного підходу в підготовчий процес відкриває нові горизонти для професійного зростання шахіста та ефективного опрацювання великих обсягів інформації.

1.4 Формулювання вимог та постановка задач на створення онтологічної бази знань з теорії шахів

Розробка онтологічної бази знань з шахових дебютів потребує ретельного аналізу предметної області та постановки чітких вимог до структури майбутньої системи. У цьому підрозділі описано основний функціонал майбутньої онтології, що дозволить ефективно моделювати та візуалізувати знання про шахові дебюти.

Онтологія буде орієнтована на представлення базових варіантів шахових дебютів та елементів шахової гри, таких як фігури, шахова дошка. Основною метою є створення зручної структури для вивчення, аналізу та систематизації дебютних варіантів. Серед ключових функціональних вимог до онтології можна виділити:

- моделювання шахової дошки – представлення дошки як сукупності клітинок з координатами (наприклад, e4, d5) і встановлення зв'язків між ними;
- опис шахових фігур – створення класів для кожного типу фігур (пішак, кінь, слон, тура, ферзь, король), з атрибутами кольору та положення;
- базові дебюти – формалізація класів для загальновідомих дебютів (наприклад, Іспанська партія, Сицилійський захист, Ферзевий гамбіт тощо);
- дебютні варіанти – реалізація гілок і підваріантів дебютів із вказанням послідовностей ходів;
- зв'язки між ходами – побудова графа або ієрархії ходів, що дозволяє прослідкувати розвиток гри у межах дебюту;
- пошук дебютів – реалізація механізмів пошуку за назвою дебюту, початковими ходами або класифікацією;
- сумісність з OWL – онтологія реалізується у форматі OWL, що дозволяє здійснювати запити, розширення та інтеграцію з іншими семантичними системами.

У підсумку буде створено структуровану та гнучку онтологічну базу знань, яка стане основою для вивчення шахових дебютів, їх класифікації та подальшого розширення системи з метою навчання, аналізу або дослідження.

1.5 Висновок до розділу 1

У даному розділі було розглянуто роль сучасних інформаційних технологій у розвитку шахової науки, зокрема – у процесах зберігання, аналізу та вивчення шахових партій. Було встановлено, що цифрові інструменти, такі як шахові движки, бази даних та навчальні ресурси, значно покращили доступ до знань і прискорили підготовку гравців різного рівня. Особливу увагу приділено використанню інформаційних систем для вивчення дебютної теорії. Було проведено огляд найбільш популярні програмні засоби й онлайн-платформи, які використовуються для навчання шахам, зокрема ChessBase, Lichess, Chess.com, Stockfish та інші.

Проаналізовано переваги таких систем, як-от наявність дебютних дерев, статистики і можливість глибокого аналізу. Разом з тим, виявлено певні недоліки існуючих інструментів, зокрема – обмеження у формалізації шахових знань, відсутність єдиного семантичного підходу, а також складність інтеграції знань у більш гнучку логічну структуру.

Визначені цілі створення онтологічної бази знань, яка структуровано описуватиме елементи шахової дошки, фігури та основні дебютні варіанти. Сформульовано перелік функціональних вимог до онтології, включно з підтримкою ієрархії дебютів, відображенням зв'язків між ходами та можливістю розгалуження варіантів. Було також окреслено основні завдання для реалізації проєкту, зокрема аналіз інформаційної структури дебютів, проектування моделей, розробку відносин між об'єктами онтології та тестування бази знань.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ З ТЕОРІЇ ШАХІВ

2.1 Обґрунтування вибору середовища для моделювання

У процесі розробки онтології важливим етапом є вибір відповідного програмного середовища, яке дозволить ефективно моделювати, візуалізувати та тестувати базу знань. Це середовище має відповідати як технічним вимогам, так і потребам кінцевого користувача. Найпопулярнішими системами для роботи з базами знань є CLASSIC, Protege, TopBraid Composer, OntoStudio.

CLASSIC — це одна з description logic-систем (DL-систем), тобто систем, що базуються на логіках опису (description logics), які є підмножиною першого порядку логіки і призначені для формального представлення та обробки ієрархічних знань про концепти (класи) та об'єкти (інстанції). Ось основні характеристики CLASSIC:

- концепти (Concepts) — абстрактні класи об'єктів, наприклад, Person, Employee;
- ролі (Roles) — бінарні відношення між концептами, наприклад, hasChild, worksFor. Інстанції (Individuals) — конкретні об'єкти, що належать до концептів.

CLASSIC використовує deterministic reasoning — вона обмежена в виразності, але завдяки цьому гарантує ефективне виведення. Є попередником і натхненником для сучасних OWL DL (використовується в Protégé). Була розроблена в AT&T Bell Labs у 1980–1990-х роках.

Protégé — це безкоштовне і відкрито доступне середовище для створення, редагування та візуалізації онтологій, з особливою підтримкою OWL (Web Ontology Language) [10]. За допомогою даного середовища можна побудувати ієрархії класів, властивостей, індивідів, редагувати правила SWRL, запитів DL Query, SPARQL. Protégé підтримує OWL DL, OWL2, RDF(S), також є доступною

графічна візуалізація онтології (OntoGraf, VOWL). Є можливість експортування в RDF/XML, OWL, Turtle та інші формати [11]. Особливості:

- плагіни: можна підключити додаткові функції (наприклад, для валідації або побудови графів);
- висока гнучкість у моделюванні: легко будувати складні логічні структури та зв'язки;
- активне співтовариство: велика база знань, приклади, форум. Можна підключити додаткові функції (наприклад, для валідації або побудови графів). Легко будувати складні логічні структури та зв'язки.

Protégé широко використовують для створення онтологій у біомедицині, освіті, IT, а також таких тем як шахи — де важливі ієрархії, правила, позиції та відношення.

TopBraid Composer — це професійне середовище розробки семантичних моделей від компанії TopQuadrant. Воно орієнтоване на підприємства, які використовують семантичні веб-технології у виробництві, IT, фінансах, тощо. TopBraid Composer повністю підтримує OWL, RDF(S), RDFS, SPIN, SHACL, а також є зручним для побудови та валідації семантичних моделей і правил. Присутній вбудований редактор SPARQL, підтримка SPIN-правил (SWRL-подібні правила на базі SPARQL). Підтримуються графи знань, Linked Data, JSON-LD. Є розширення через JavaScript або SPARQLMotion. Особливості:

- високий рівень інтеграції з інженерними процесами;
- комерційна ліцензія, складний інтерфейс, потребує підготовки;
- має Enterprise-версію, що інтегрується з базами даних і великими системами.

Краще всього TopBraid Composer підходить для побудови промислових або корпоративних онтологій і для обробки складних бізнес-процесів з логікою і метаданими.

OntoStudio — це потужне комерційне середовище для інженерії знань, розроблене компанією Semafora Systems. Використовується для створення

онтологій, правил, семантичних баз знань і інтеграції даних. Підтримує OWL, RDF, RDFS, SWRL, RuleML. Дає можливість побудувати логічні правила з використанням графічного редактора. Присутнє візуальне моделювання концептів, відношень, екземплярів. Можлива інтеграція з реляційними БД, SAP, Java та автоматичне генерування запитів, рефакторинг онтології. Особливості:

- орієнтований на інженерів знань та бізнес-аналітиків;
- висока інтерактивність: зміни одразу відображаються у виводі;
- комерційна ліцензія, переважно використовується в промислових проектах.

Використовується для побудови інтегрованих онтологічних систем у бізнесі, охороні здоров'я, техніці. А також там, де потрібне глибоке моделювання правил та реляцій.

Виходячи з описів таких програм як CLASSIC Protege, TopBraid Composer, OntoStudio, доцільно буде створити порівняльну таблицю 2.1 для кращої наочності результатів.

Таблиця 2.1 Порівняння найпопулярніших систем для роботи з базами знань

Критерій / Система	CLASSIC	Protégé	TopBraid Composer	OntoStudio
Тип	Система логіки опису (DL)	Редактор онтологій (підтримує OWL, RDF, SWRL)	Професійне середовище для семантичного моделювання	Потужна комерційна платформа для онтологій
Підтримка OWL	Ні (власна мова DL)	Повна підтримка OWL DL, OWL2	Повна підтримка	Повна підтримка

Продовження таблиці 2.1 Порівняння найпопулярніших систем для роботи з базами знань

Інтерфейс користувача	Консоль/текст	Зручний графічний інтерфейс	Комерційний GUI	Комерційний GUI
Візуалізація	Немає	Граф ієрархій, плагіни (OntoGraf, VOWL)	Вбудовані графи та SPARQL-візуалізація	Візуалізація класів, правил
Плагіни та розширення	Немає	Широкий вибір (SWRL, DL Query, GraphView тощо)	Інтегровані функції + підтримка SPARQL	Власні скрипти та правила
Підтримка правил	Обмежена (через DL-аксіоми)	SWRL, SPIN, DL Query	SPIN, SPARQLMotion	OntoRules, SWRL
Підтримка SPARQL	Ні	Так (через плагін)	Так (глибока інтеграція)	Так
Навчальна крива	Висока (старий синтаксис)	Середня (інтуїтивно зрозумілий)	Висока (комерційний інтерфейс, складні налаштування)	Висока (націлена на експертів)
Ліцензія / Доступність	Вузькообмежена, застаріла	Безкоштовна, відкрита (Stanford)	Комерційна (доступна демо-версія)	Комерційна
Придатність для шахової онтології	Низька (немає графіки, сучасних форматів)	Висока: класи дебютів, фігури, позиції, SWRL-правила, візуалізація	Надлишкова для освітнього або дослідницького проєкту	Надмірна складність для проєкту середнього рівня

Отже, можна зробити висновок, що CLASSIC найкраще підходить для історичних досліджень логіки опису. Protégé – для освітніх, дослідницьких, навчальних проєктів. TopBraid Composer – для великих корпоративних графів знань. OntoStudio – для промислових проєктів з правилами й інтеграцією. Саме через вищенаведені причини та ознаки, для розробки бази знань з теорії шахів було обрано середовище Protégé.

2.2 Методологія 7 кроків і її застосування в проектуванні та моделюванні онтологічної бази знань з теорії шахів

Опис методології 7 кроків є ключовим етапом у систематичному проектуванні онтологічних баз знань. Цей підхід дозволяє створити чітку, логічну та масштабовану структуру знань, що відповідає поставленим цілям та потребам користувачів. У контексті створення онтологічної бази знань з теорії шахів застосування цієї методології забезпечує узгоджене моделювання понять, термінів і зв'язків, характерних для шахової гри. На початкових етапах розглядаються такі критично важливі кроки, як визначення області та масштабу онтології, пошук можливостей повторного використання існуючих онтологій, а також перерахування ключових термінів, які будуть використовуватись у моделі. Ці кроки створюють надійне підґрунтя для подальшої формалізації та реалізації знань у шаховій предметній області [7].

Крок 1. Розпочнемо розробку онтології з визначення її області та масштабу. Тобто відповімо на кілька основних питань:

- Яку область буде охоплювати онтологія?
- Навіщо ми збираємося використовувати онтологію?
- На які типи питань має відповідати інформація в онтології?
- Хто буде використовувати та підтримувати онтологію?

Відповіді на ці запитання можуть змінюватися під час процесу проектування онтології, але в будь-який момент вони допомагають обмежити і структурувати її масштаб [12].

Область онтології — це представлення знань із теорії шахів, зосереджене насамперед на шахових дебютах. Онтологія описуватиме основні класи дебютів, варіації, послідовності ходів, цілі дебютної стратегії, пов'язані шахові поняття. Дана онтологія буде використовуватися у навчальних додатках та системах, а саме допомагатиме новачкам та аматорам у вивченні шахових дебютів. Також використовуватиметься для аналізу партій, визначення типу дебюту, що був застосований, і пошук можливих продовжень. У нашу онтологію будуть включені поняття, що описують:

- типи шахових дебютів (відкриті, закриті, напіввідкриті);
- ключові дебюти (іспанська партія, сицилійський захист, ферзевий гамбіт тощо);
- варіації дебютів.

Натомість, малоймовірно, що онтологія включатиме теми, як-от структура шахових турнірів, біографії шахістів або комерційні аспекти гри, хоча ці теми частково дотичні до шахової теорії.

Залежно від того, хто використовуватиме онтологію, можуть бути включені різні рівні деталізації. Для початківців важливо мати прості пояснення типових дебютів, приклади. Для просунутих гравців — зв'язки між дебютами, типи позицій, сильні і слабкі сторони варіантів. Для розробників навчальних застосунків — структурована система класів, властивостей і інстанцій. Для аналітиків шахових партій — можливість інтеграції онтології в шахові движки або бази даних партій.

Якщо мова користувача онтології (наприклад, гравець-початківець) суттєво відрізняється від мови експерта або тренера, що підтримує онтологію, потрібно буде створити синонімічні відповідності, наприклад:

"Ферзевий гамбіт" ↔ "Queen's Gambit" ↔ "1.d4 d5 2.c4"

Один із способів визначити масштаб онтології — перелік питань, на які має відповідати база знань. Ці питання допоможуть перевірити, чи містить онтологія достатньо знань для практичного використання. Можливі запитання для онтології шахових дебютів:

- Які основні категорії шахових дебютів існують?
- Яка послідовність ходів відповідає сицилійському захисту?
- Які основні варіації існують у ферзевому гамбітові?
- Які дебюти зазвичай ведуть до відкритих позицій?

Судячи з цього списку, онтологія має включати структуровану класифікацію дебютів, послідовності ходів, пов'язані позиційні і стратегічні концепції, можливість логічного виведення (наприклад, визначити дебют за ходами).

Крок 2. Під час розробки онтології для теорії шахів важливо врахувати існуючі ресурси, які можуть бути адаптовані або розширені для нашої предметної області. Хоча спеціалізованих онтологій, присвячених шаховим дебютам, наразі обмежено, існують кілька проектів, що моделюють шахові партії та пов'язані концепції.

Ontology Design Pattern for Chess Games. Цей шаблон онтології описує структуру шахових партій, включаючи гравців, турніри, ходи та просторово-часову інформацію. Він також розглядає можливість інтеграції з пов'язаними наборами даних, такими як DBpedia та GeoNames, для збагачення інформації про шахові події [7]. З даного прикладу можна включити до власної онтології клас “ChessGame”, який буде слугувати основним контейнером для партій — потрібен для структурування теоретичних дебютів. Також включено клас “BoardSquare” (Клітинка на шахівниці) – координати на шахівниці, для точного відображення позицій.

Моделювання онтологій з використанням SHACL. У цьому прикладі демонструється використання SHACL для моделювання онтології шахових партій. Хоча приклад є базовим, він ілюструє застосування типових шаблонів моделювання, які можуть бути адаптовані для більш складних онтологій. З даного прикладу можна включити до власної онтології клас “Move” – один конкретний хід у шаховій партії. Має властивості – номер ходу, фігура, клітинка "з" і "на". Важливий для побудови послідовності ходів у дебюті. Також включено клас Piece (фігура).

CHREST (Chunk Hierarchy and REtrieval Structures). Хоча CHREST не є онтологією в традиційному сенсі, ця когнітивна модель використовується для моделювання процесу навчання та розпізнавання шаблонів у шахах. Вона імітує, як гравці різного рівня майстерності сприймають та запам'ятовують шахові позиції, що може бути корисним при розробці онтології, орієнтованої на навчання.

Незважаючи на те, що повноцінної онтології, присвяченої виключно шаховій теорії, наразі не існує, наявні ресурси можуть служити основою для розробки спеціалізованої онтології. Інтеграція існуючих шаблонів та моделей дозволить ефективно структурувати знання у сфері шахової теорії та забезпечити можливість їх повторного використання в різних додатках.

Крок 3. На даному етапі створення онтології необхідно скласти список усіх термінів, про які ми хотіли б сказати щось або які хотіли б пояснити користувачеві. Складання списку термінів є важливим етапом у проектуванні онтології, оскільки це допомагає зрозуміти, про що саме буде онтологія, які поняття в ній мають бути представлені. Він визначає межі предметної області та допомагає уникнути надмірного розширення або звуження теми. Список термінів служить основою для створення класів і властивостей. Кожен термін із початкового списку згодом може стати класом, властивістю, або інстанцією. Зі списку термінів легше визначити, які поняття пов'язані, які мають бути об'єднані в ієрархії або яким чином вони взаємодіють [13]. Для моделювання онтології з шахової теорії було визначено такі основні терміни:

дебют – це початкова стадія шахової партії, зазвичай перші 10–15 ходів, що мають теоретичне значення. Дебют включає стандартні послідовності ходів, які використовуються з метою розвитку фігур, контролю центру, забезпечення безпеки короля тощо;

варіація дебюту – Це конкретна гілка в рамках певного дебюту, що описує послідовність ходів у межах цього дебюту. Наприклад – «Іспанська партія: Варіація Берда»;

хід – один окремий рух фігури на шахівниці (наприклад, e4, Kf3);

- номер ходу – визначає порядковий номер ходу в партії або варіації;
 - колір фігур – може бути властивістю для фігури або гравця. Значення – білий або чорний;
 - шахова фігура – репрезентує будь-яку фігуру: король, ферзь, тура, слон, кінь, пішак;
 - шахівниця – ігрове поле розміром 8x8 клітин, кожна має унікальну назву (наприклад, a1, h8);
 - клітинка – окрема позиція на шахівниці, позначена координатами (буква + цифра);
 - початкова позиція – стандартне початкове розташування всіх фігур на шахівниці;
 - назва дебюту (наприклад, Іспанська партія, Захист Каро-Канн) – може використовуватись як унікальний ідентифікатор дебюту.
- Обрані терміни відображають ключові поняття, необхідні для моделювання знань про шахові дебюти та хід партії. Вони охоплюють як структурні елементи шахової гри (шахівниця, клітинка, фігура), так і динамічні аспекти (хід, номер ходу, дебют, варіація дебюту). Наявність таких термінів, як «назва дебюту» та «варіація дебюту», дозволяє не лише класифікувати шахові відкриття, а й зв'язати їх з конкретними послідовностями ходів. Цей набір понять створює міцну основу для побудови онтології, яка може бути корисною як для навчання новачків, так і для аналізу шахових партій у навчальних або аналітичних системах.

2.3 Проектування онтологічної бази знань з теорії шахів

Крок 4. Визначення класів та ієрархій класів. Визначення класів та ієрархій класів є однією з ключових складових створення онтологічної бази знань, зокрема й у сфері теорії шахів. Ієрархія класів — це скелет онтології. Без неї система не зможе масштабуватись, робити логічні висновки, або ефективно

відповідати на запити. У контексті теорії шахів вона дозволяє чітко моделювати як саму гру, так і її стратегічні й тактичні елементи [14].

Клас «Дебюти» представляє початкову фазу шахової партії, коли обидві сторони — білі та чорні — намагаються швидко розвинути фігури, захопити контроль над центром дошки та забезпечити безпеку короля (зазвичай шляхом рокіровки). Дебют є фундаментом усієї подальшої гри, від якого залежать стратегічні плани у мительшпілі. У цьому класі зібрано основні типи початку партії. Підкласи:

- захист_двух_коней — варіант дебюту, в якому чорні швидко виводять обох коней (на поля f6 і c6), створюючи тиск на центральні позиції та потенційно готуючи контратаку;
- захист_Каро-Канн — спокійний і позиційний дебют, в якому чорні грають 1...c6 з подальшим 2...d5, прагнучи до стабільної пішакової структури та мінімізації ризиків;
- сицилійський_захист — агресивна відповідь чорних на 1.e4: 1...c5. Призначений для динамічної гри та контратаки на флангах;
- французький_захист — захист, в якому чорні відповідають 1...e6, готуючи підрив центру ходом d5. Особливістю є закрита структура та стратегічна боротьба;
- італійська_партія — класичний відкритий дебют, в якому білі грають 1.e4 e5 2.Nf3 Nc6 3.Bc4. Основна ідея — швидкий розвиток легких фігур і тиск на слабе поле f7.

Кожен дебют має варіанти та підваріанти, що можуть бути розглянуті в окремих підкласах або властивостях дебюту (наприклад, атакувальний варіант, захисний варіант, основна ідея, тип позиції тощо).

Клас «Дошка» представляє фізичну шахову дошку — площину 8x8 клітин, чергуючи світлі та темні кольори. Вона є простором для гри, що визначає дозволені пересування фігур. Підклас «Клітинка» — окрема комірка дошки, яка має координати (наприклад, e4) і може бути зайнята фігурою або вільна; під ударом (атакована фігурою супротивника).

Клас «Сторона» описує команду, до якої належать гравець або фігура. Білі — завжди роблять перший хід у партії. Чорні — відповідають на хід білих. Цей клас може включати властивості, як-от: час на партію, приналежність фігури, стратегічні плани сторони.

Клас «Фігури» описує всі типи фігур, що використовуються в грі. Кожна фігура має унікальний спосіб руху, роль у партії та вартість (в умовних одиницях для оцінки позиції). Підкласи:

- король — найважливіша фігура. Рухається на одну клітинку в будь-якому напрямку. Його втрата означає поразку (мат);
- ферзь — найсильніша фігура, що поєднує можливості тури й слона: рухається вертикально, горизонтально або по діагоналі на будь-яку відстань;
- тура — сильна фігура, що рухається по горизонталях і вертикалях. Особливо ефективна у відкритих лініях;
- слон — рухається лише по діагоналях, може атакувати з великої відстані, але завжди залишається на полях одного кольору;
- кінь — єдина фігура, яка може "перестрибувати" інші. Рухається «буквою Г»: на дві клітинки в одному напрямку і потім на одну в перпендикулярному;
- пішак — найчисленніша фігура. Рухається вперед на одну клітинку (з початкової позиції — на дві), атакує по діагоналі. Може перетворитися на іншу фігуру, досягнувши останньої горизонталі (промоція).

Кожна фігура може мати такі властивості, як – позиція (поточна клітинка), активність (чи бере участь у грі), можливі ходи, під атакою або атакує.

Клас «Хід» описує окрему дію, що здійснюється фігурою під час партії. Властивостями класу Хід можуть бути: фігура, яка здійснює хід; клітинка, з якої здійснено хід; клітинка, на яку здійснено хід; номер ходу в партії; тип ходу: звичайний, взяття, шах, мат, рокіровка, перетворення пішака (промоція), взяття

на проході. Ходи можуть бути частиною стратегічних планів, тактик або комбінацій (наприклад, жертва фігури для атаки).

Формування класів розпочинається з визначення основних категорій, які необхідно включити до онтології. У середовищі Protege кожному класу надається унікальна назва, що спрощує його ідентифікацію та дозволяє легко посилатися на нього в межах моделі. Візуальне представлення створених класів наведено на рисунку 2.1.

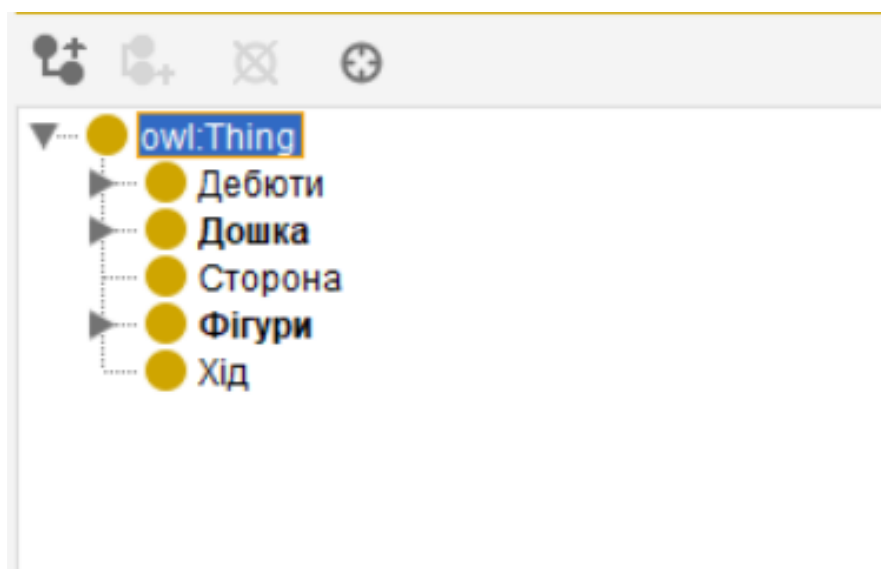


Рисунок 2.1 – Класи онтологічної бази знань

Після створення основних класів формується ієрархічна структура, яка відображає зв'язки між загальними поняттями та їх підкласами. Така ієрархія дозволяє детально описати об'єкти предметної області та впорядкувати знання у вигляді дерева. Наприклад, клас «Фігури» поділяється на підкласи «Король», «Ферзь», «Тура», «Слон», «Кінь» і «Пішак», що дозволяє точніше структурувати інформацію про різновиди шахових фігур. Подібним чином, клас «Дебюти» містить підкласи «Італійська_партія», «Захист_Каро-Канн», «Сицилійський_захист» та інші, що забезпечує детальніше представлення основних стратегій початку гри. Такий підхід робить онтологію більш гнучкою та зручною для розширення. Ієрархія класів наведена на рисунку 2.2.

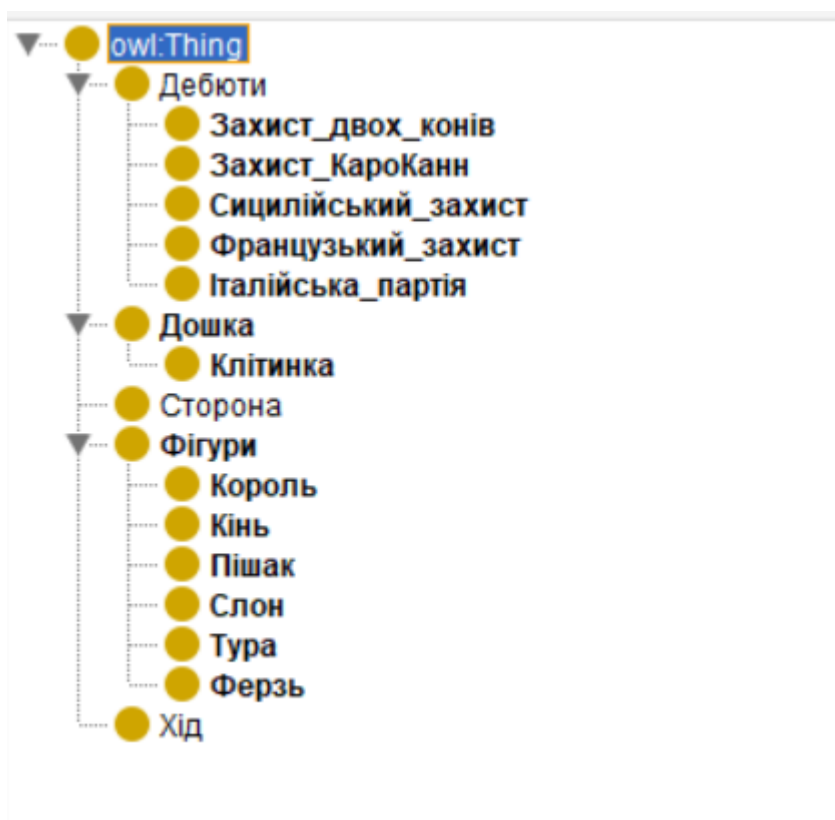


Рисунок 2.2 – Ієрархія класів онтологічної бази знань

Дана ієрархія дозволяє структурувати шахові знання у вигляді чітких концептуальних об'єктів, представити різні аспекти гри, від типів фігур і дебютів до механіки ходів. Також саме за допомогою ієрархії можна забезпечити масштабованість — легко додати нові дебюти, фігури, типи ходів та покращити аналіз та пошук інформації (наприклад, фільтрація дебютів, аналіз стратегій).

Крок 5. Визначення властивостей класів-слотів.

Після визначення класів та побудови їхньої ієрархічної структури наступним важливим кроком є додавання атрибутів (data properties), які надають більш детальний опис характеристик об'єктів, що належать до кожного класу. Атрибути в онтології служать для збереження додаткової інформації про властивості конкретних елементів, таких як фігури, клітинки, ходи та інші сутності, що моделюються. У середовищі Protégé атрибути прив'язуються безпосередньо до відповідних класів, і кожен атрибут має свій тип даних — наприклад, текстовий (string), числовий (integer), логічний (boolean) або інші, що забезпечує коректне і структуроване зберігання інформації. Це дозволяє значно

розширити семантичну насиченість онтології, зробити її більш гнучкою для подальшого аналізу, пошуку та автоматизованої обробки даних [15]. У створеній онтології було визначено наступні атрибути:

- колір — визначає колір фігури або сторони (наприклад, білий або чорний);
- має_назву — використовується для вказання назви об'єкта (наприклад, назва дебюту чи фігури);
- номер_вертикалі — числовий атрибут, що вказує позицію клітинки по вертикалі (від 1 до 8);
- номер_горизонталі — числовий атрибут, що вказує позицію клітинки по горизонталі (від 'a' до 'h' або їх числове представлення);
- номер_ходу — визначає порядковий номер конкретного ходу в партії;
- позначення_ходу — текстовий атрибут, що описує хід у шаховій нотації (наприклад, e4, Nf3, Bb5).

Процес додавання атрибутів включає визначення назви, типу даних та класу, до якого цей атрибут буде застосовуватись. Це забезпечує точне та структуроване представлення знань у предметній області шахів. Атрибути, додані до моделі, зображені на рисунку 2.3.

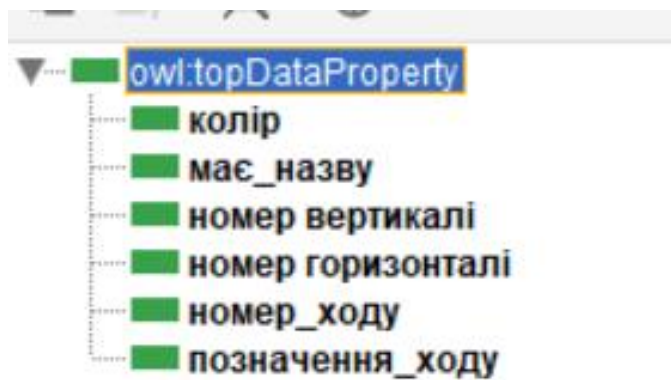


Рисунок 2.3 – Атрибути класів онтологічної бази знань

У процесі моделювання онтології з теорії шахів у середовищі Protégé було створено дата-властивість "колір", яка є важливою характеристикою як для шахових фігур, так і для клітинок на шахівниці. Для цієї властивості було вказано тип значення — `xsd:string`, що означає, що значенням властивості буде текстовий рядок (наприклад, "білий", "чорний"). У полі `domain` було вказано, що ця властивість може бути застосована до наступних класів: фігура (англ. Piece), клітинка (англ. Cell або Square). Це означає, що лише екземпляри цих класів можуть мати атрибут "колір". Створення властивості "колір" зображено на рисунку 2.4.

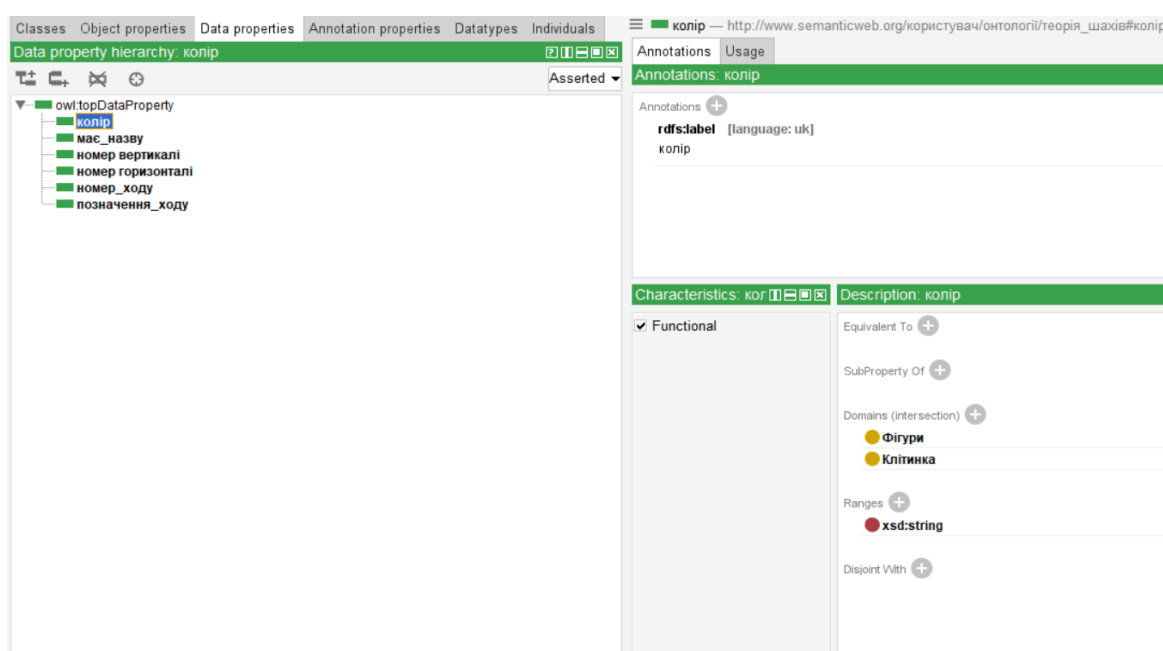


Рисунок 2.4 – Атрибут “Колір”

У рамках моделювання шахової онтології у середовищі Protégé було також створено функціональну дата-властивість "номер горизонталі", яка дозволяє задавати номер ряду (від 1 до 8) для кожної клітинки шахової дошки. Оскільки кожна клітинка може належати лише до однієї горизонталі, ця властивість була визначена як функціональна (Functional Data Property). Це означає, що один індивід може мати тільки одне значення цієї властивості. Було встановлено, що значення властивості мають бути цілими числами, тобто обрано тип: `xsd:integer`.

(XML Schema Definition — ціле число). Створення властивості "номер_горизонталі" зображено на рисунку 2.5.

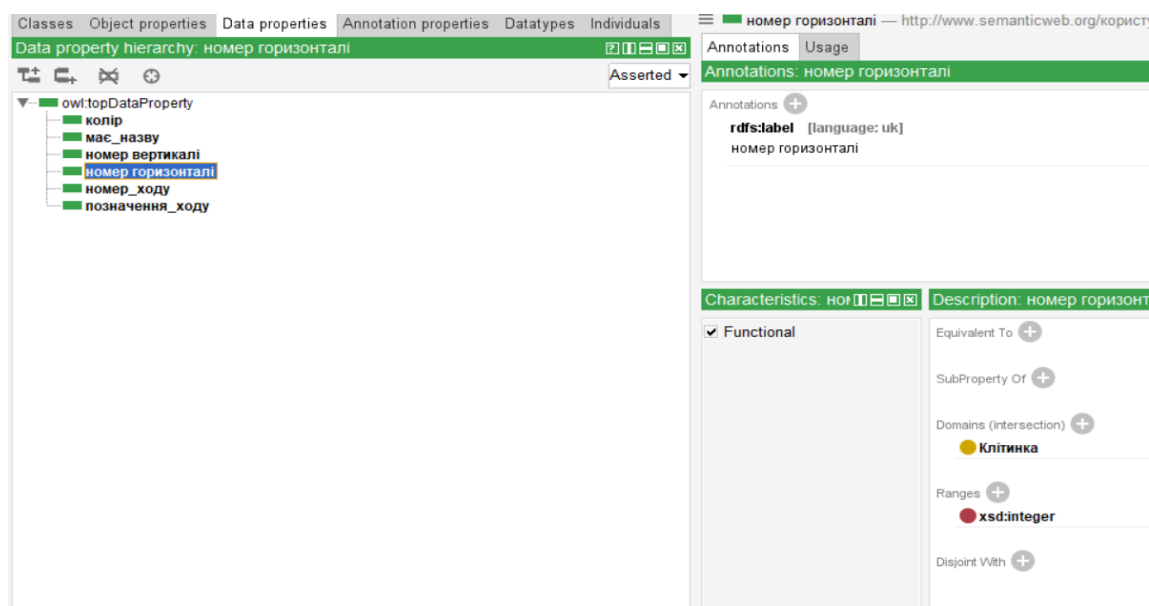


Рисунок 2.5 – Атрибут “Номер горизонталі”

Атрибути відіграють ключову роль у деталізації даних, що дозволяє отримати максимально повну та точну інформацію про кожен об’єкт в онтології. Вони не лише підвищують якість і точність представлених знань, а й значно розширюють функціональні можливості самої моделі. Кожен атрибут має свою конкретну роль і відповідає за збереження певного аспекту або характеристики об’єкта, що допомагає більш глибоко описати його властивості. Крім того, наявність детальних атрибутів спрощує процес пошуку, фільтрації та аналізу даних, що важливо для ефективного використання онтології в різних прикладних задачах. Таким чином, атрибути не просто доповнюють опис об’єктів, а створюють основу для більш гнучкого і адаптивного представлення знань, що відповідає потребам користувачів і специфіці предметної області [15].

Об’єктні властивості є однією з ключових складових онтологічної моделі, оскільки саме вони дозволяють формалізувати відносини між об’єктами предметної області — у нашому випадку, між такими сутностями як фігури, клітинки, ходи, сторони тощо. У середовищі Protégé ці властивості

налаштовуються індивідуально для кожного класу, що робить модель більш гнучкою, виразною та придатною до семантичного аналізу гри [16].

У побудованій онтології шахової гри було визначено наступні об'єктні властивості, які відображають логіку та динаміку взаємодії між компонентами гри:

`відбувається_на` – зв'язує клас Хід із класом Шахова дошка, що дозволяє вказати, на якій дошці саме відбувається певна ігрова дія. Це важливо у випадках моделювання кількох паралельних партій чи турнірів;

`здійснений_фігурою` – встановлює зв'язок між Ходом і тією Фігурою, яка його виконала. Це дає змогу ідентифікувати активного учасника кожної дії, а також аналізувати динаміку участі фігур у грі (наприклад, які фігури виконують найбільше ходів);

`знаходиться_на` – пов'язує Фігуру з Клітинкою, на якій вона перебуває у певний момент часу. Ця властивість є основною для фіксації позицій на шаховій дошці, що особливо важливо при моделюванні поточних розташувань або перевірці правил ходів;

`зроблений_стороною` – вказує, яка саме Сторона (біла чи чорна) здійснила конкретний Хід. Це дозволяє контролювати черговість ходів, чергування гравців та дотримання правил шахової гри;

`містить_хід` – зв'язує Шахову партію з усіма їїходами. Така властивість дозволяє зберігати повну хронологію подій у межах однієї гри, формуючи послідовність дій і забезпечуючи можливість аналізу ігрової стратегії;

`складається_з` – використовується для опису структурної цілісності складових, наприклад, що Шахова дошка складається з 64 Клітинок, або що Партія складається з послідовності ходів. Це дає змогу моделювати ієрархію елементів;

`сусідня_з` – встановлює просторові відношення між Клітинками шахової дошки, що особливо актуально для реалізації перевірки легальності руху фігур (наприклад, для пішаків, коней або при перевірці загрози).

Використання об'єктних властивостей дозволяє не лише описати статичні зв'язки між елементами, а й реалізувати динамічні сценарії шахової гри. Наприклад, за допомогою властивостей можна:

- виявити, чи атакує певна фігура клітинку, яку займає фігура суперника;
- відстежити, які ходи вже були зроблені і ким;
- перевірити правильність черговості сторін;
- визначити, чи знаходиться король під шахом;
- змоделювати правила рокіровки, взяття на проході чи перетворення пішака.

Крім того, ці властивості є основою для побудови запитів SPARQL, які дозволяють витягати з бази знань потрібну інформацію, наприклад:

Які фігури зараз знаходяться на другій горизонталі?

Які ходи були здійснені білими фігурами у поточній партії?

Чи був вже зроблений хід ферзем у цій грі?

Таким чином, об'єктні властивості виконують подвійну функцію: з одного боку, вони служать для формалізації знань про шахову гру, а з іншого — для побудови інтелектуальних механізмів аналізу, навчання, підказок та перевірки правильності ходів. Їх грамотне використання у середовищі Protégé дозволяє створити повноцінну онтологічну модель, придатну як для навчальних, так і для прикладних задач у сфері штучного інтелекту, гейм-дизайну або шахової аналітики. Створені об'єктні властивості зображені на рисунках 2.6 - 2.8.

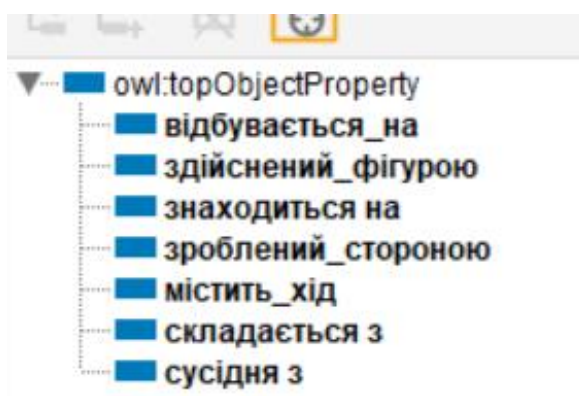


Рисунок 2.6 – Об'єктні властивості онтологічної бази знань

Властивість `сусідня_з` використовується для встановлення просторового зв'язку між клітинками шахівниці, які безпосередньо сусідять одна з одною — по горизонталі, вертикалі або діагоналі. Характеристики властивості: `domain`: Клітинка, `range`: Клітинка, тип: Симетрична (тобто, якщо A1 сусідня з B1, то й B1 сусідня з A1).

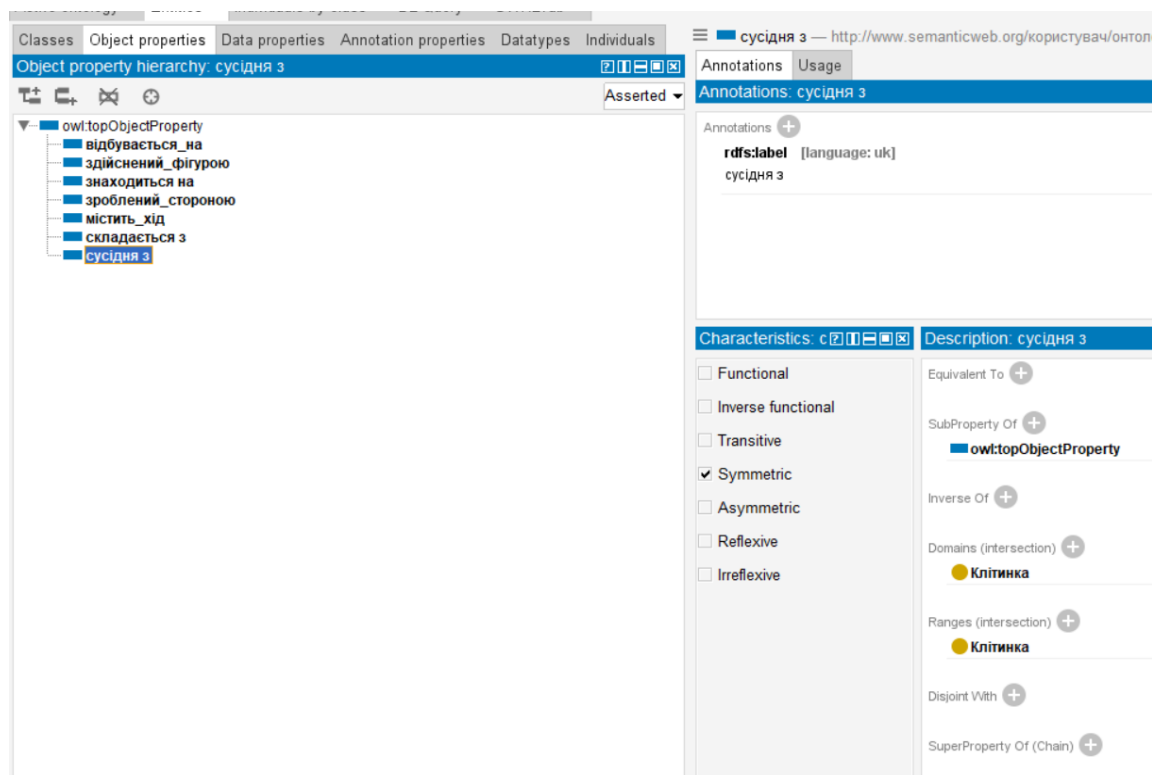


Рисунок 2.7 – Об’єктна властивість “Сусідня з”

Властивість `знаходиться_на` описує розташування шахової фігури на конкретній клітинці шахівниці. Вона є ключовою для відображення актуальної позиції фігури під час партії та служить основою для моделювання шахового стану на будь-якому етапі гри. Характеристики властивості: `domain`: Фігура, `range`: Клітинка, тип: Функціональна (одна фігура може перебувати тільки на одній клітинці одночасно). Крім цього, `знаходиться_на` може бути корисною при реалізації логіки для визначення допустимих ходів, виявлення конфліктів (наприклад, коли дві фігури не можуть одночасно займати одну клітинку), перевірки шаху та мату, або навіть аналізу контрольованих зон на дошці.

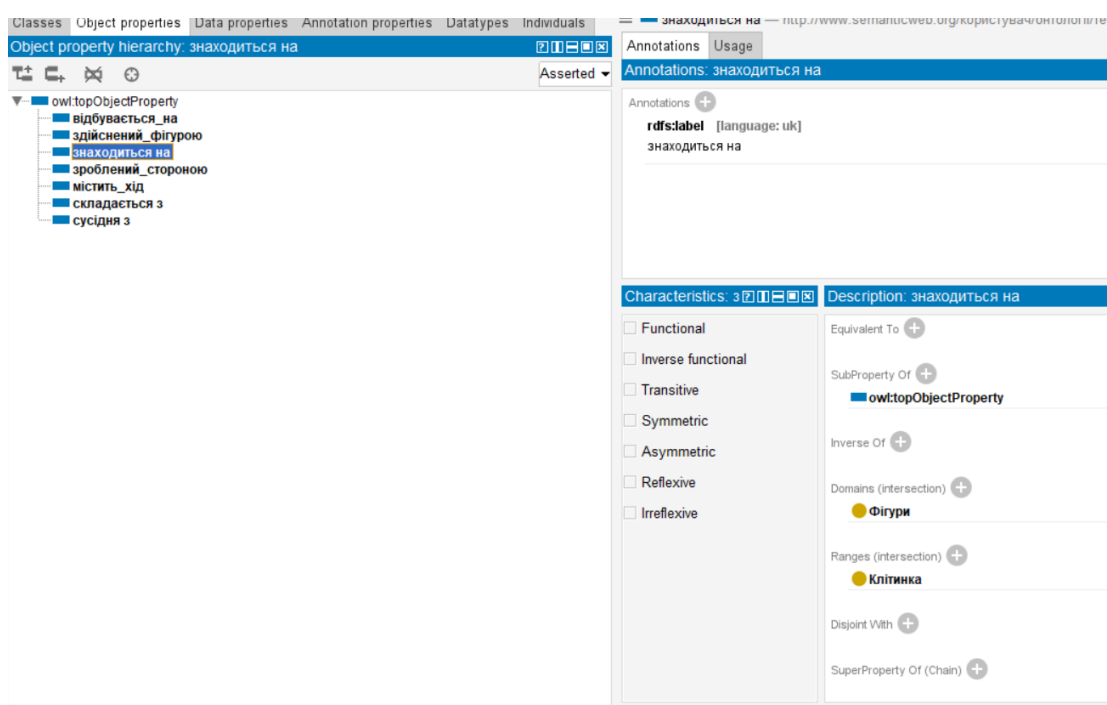


Рисунок 2.8 – Об’єктна властивість “Знаходиться на”

Крок 6. Визначення фацетів слотів. Після визначення атрибутів (слотів) для кожного класу онтології необхідно встановити чіткі обмеження на значення цих слотів. Це гарантує коректність та консистентність даних, уникнення помилок при введенні інформації та точне відображення знань про шахову гру, дебюти, ходи, фігури і позиції на дошці. Обмеження допомагають підтримувати якість онтології, що є критичним для системи аналізу, навчання чи підтримки прийняття рішень [7].

Основні класи онтології шахів та приклади обмежень:

Клас «Фігури». Підкласи: «Король», «Ферзь», «Тура», «Слон», «Кінь», «Пішак». Слоти: колір — обов’язковий, приймає значення із фіксованого набору (білий, чорний), має_назву — текстове поле, обов’язкове, містить офіційну назву фігури (наприклад, «Король»). Значення колір має бути строго «білий» або «чорний». Довжина має_назву — не більше 20 символів.

Клас «Дебюти». Підкласи: «Італійська_партія», «Захист_Каро-Канн», «Сицилійський_захист». Захист двох конів, Французький захист. Слоти: має_назву — обов’язкове, текстове, до 50 символів (наприклад, «Сицилійський захист»), номер_ходу — числовий, позначає номер ходу, з якого починається

дебют, від 1 і більше, опис (додатковий слот) — текстове поле з описом стратегії дебюту, довжина до 500 символів. Обмеження: номер_ходу — ціле число, починаючи з 1, має_назву не може бути порожнім, опис є необов'язковим, але бажаним для повноти інформації.

Клас «Хід». Слоти: номер_ходу — обов'язковий числовий атрибут (1, 2, 3...), визначає порядок ходів у партії, позначення_ходу — текстовий, шахова нотація (наприклад, «e4», «Nf3»), обов'язковий, сторона — посилання на об'єкт класу «Сторона» (підкласи «Білі» або «Чорні»), обов'язкове, початкова_клітинка та кінцева_клітинка — посилання на об'єкти класу «Клітинка» (підклас «Дошка»), для фіксації точного переміщення фігури. Обмеження: номер_ходу — унікальний у межах однієї партії, позитивне ціле число. Позначення_ходу має відповідати офіційним правилам шахової нотації. Сторона обов'язково повинна бути або «Білі», або «Чорні». Початкова_клітинка і кінцева_клітинка — мають бути валідними координатами дошки (номер_вертикалі від 1 до 8, номер_горизонталі від 'a' до 'h'). Забороняються ходи, які виходять за межі дошки.

Клас «Сторона». Підкласи: «Білі», «Чорні». Слоти: колір — обов'язковий, зі значенням «білий» для класу «Білі» і «чорний» для класу «Чорні». Обмеження: для кожного підкласу обов'язковий фіксований колір. Можливе використання для фільтрації ходів і фігур за кольором.

Клас «Дошка» містить підклас «Клітинка», який описує одну з 64 позицій на шахівниці. Кожна клітинка має обов'язкові атрибути – номер_вертикалі (1–8), номер_горизонталі ('a'–'h' або 1–8), має_назву (наприклад, «e4»), що автоматично формується шляхом поєднання горизонталі та вертикалі. Обмеження гарантують відповідність стандартній дошці 8×8 та унікальність кожної клітинки. Валідація значень (наприклад, виключення недопустимих координат чи кольорів фігур) забезпечує коректність моделі. Таке структурування дозволяє легко формувати SPARQL-запити для аналізу позицій, пошуку фігур чи ходів.

2.4 Висновок до розділу 2

У другому розділі було досягнуто низку важливих результатів, що свідчать про доцільність застосування онтологічного підходу для формалізації й структуризації знань у цій предметній галузі.

По-перше, було обґрунтовано вибір середовища для моделювання — Protégé, яке є одним із найпотужніших та найпоширеніших інструментів для створення, редагування та візуалізації онтологій. Його підтримка стандарту OWL, наявність графічного інтерфейсу, інструментів для перевірки консистентності, а також можливість експорту даних у різних форматах забезпечують зручність і гнучкість при роботі з базами знань будь-якої складності. З огляду на це, Protégé було обрано як основне середовище для реалізації онтології в сфері шахів.

По-друге, було застосовано методологію 7 кроків створення онтології, яка передбачає систематичний підхід до проектування бази знань, від визначення цілей до заповнення прикладами індивідів. У межах цього підходу: визначено цілі створення онтології, зокрема: формалізація шахових понять, підтримка навчального процесу, представлення структурованої інформації для автоматизованих систем. Ідентифіковано ключові поняття предметної області, такі як «Шахова фігура», «Клітинка», «Хід», «Позиція», «Гравець», «Сторона», «Тип фігури», «Дошка». Створено структуру класів та підкласів, що дозволяє точно передавати семантику шахових понять і відносин між ними. Визначено основні властивості (слоти) класів, такі як маєКоординати, знаходитьсяНаКлітинці, здійснюєХід, граєФігурою, сторонаХоду, маєНомерХоду. Описано фацети слотів, тобто типи значень, обмеження кардинальності та інші специфікації, які забезпечують логічну цілісність онтології.

Усі етапи розробки були ретельно продумані й реалізовані, що дало змогу сформувати повноцінну та масштабовану структуру знань про шахи, придатну для подальшого розширення й інтеграції з іншими інтелектуальними системами.

3 РЕАЛІЗАЦІЯ ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ З ТЕОРІЇ ШАХІВ

3.1 Створення індивідів в онтологічній базі знань з теорії шахів

Крок 7. Створення екземплярів. Наступним етапом у розробці онтології є створення індивідів (екземплярів) для кожного класу. Індивіди — це конкретні об'єкти або сутності, які належать до певних класів та формують наповнення онтологічної моделі фактичними даними. Завдяки індивідам абстрактна структура онтології набуває прикладного характеру: вони демонструють, як класи та властивості функціонують на практиці, дозволяючи виконувати логічні висновки, здійснювати запити та аналізувати реальні сценарії.

В онтології «Теорія шахів» індивіди створені для таких ключових класів, як:

«Дебют» — початок партії (перші 10-15 ходів);

«Колір гравця» — індивіди «білий» та «чорний» як значення для атрибутів гравців або фігур;

«Фігура» — індивіди «біла тура», «чорний кінь», «біла ферзь» тощо, які можуть мати унікальні властивості, такі як положення на дошці або участь у ході;

«Клітинка шахової дошки» — наприклад, індивіди «E4», «D5», що дозволяє точно описувати розташування фігур і траєкторію ходів;

«Хід» — кожен хід може бути змодельований як індивід зі вказаним початковим і кінцевим положенням фігури, її типом, кольором та порядковим номером у грі;

«Шахова дошка» — індивід може представляти конкретну дошку в певному стані, що особливо корисно для фіксації позицій у різні моменти партії.

Приклади індивідів для кожного класу:

«Дебют»

Індивід 1: Італійська_партія

Назва дебюту: Італійська партія

Початкова розіграш (містить ходи): 1.e4 e5 2.Nf3 Nc6 3.Bc4 Bc5

Індивід 2: Сицилійський_захист

Назва дебюту: Сицилійський захист

Початкова розіграш (містить ходи): 1.е4 с5

Індивід 3: Французький_захист

Назва дебюту: Французький захист

Початкова розіграш (містить ходи): 1.е4 е6

Індивід 4: Захист Каро-Канн

Назва дебюту: Захист Каро-Канн

Початкова розіграш (містить ходи): 1.е4 с6

Індивід 5: Захист двох конів

Назва дебюту: Захист двох конів

Початкова розіграш (містить ходи): 1.е4 е5 2.Nf3 Nc6 3.Bc4 Kf6

«Колір гравця»

Індивід 6: Білі – гравець, що робить перший хід

Індивід 7: Чорні – гравець, що відповідає на перший хід

«Фігура»

Індивід 8: Король_білий

Колір: Білий

Початкова позиція: e1

Індивід 9: Кінь_білий1

Колір: Білий

Початкова позиція: b1

Індивід 10: Кінь_чорний2

Колір: Чорний

Початкова позиція: g8

Індивід 11: Тура_чорна2

Колір: Чорний

Початкова позиція: h8

І таким чином описано усі 32 фігури на шаховій дошці, як індивіди, вказано колір та розташування фігури.

«Клітинка шахової дошки»

Індивід 12: Клітинка_a1

Координати: A1

Колір клітинки: чорна

Сусідня з: a2, b1, b2

Індивід 13: Клітинка_a2

Координати: a2

Колір клітинки: біла

Сусідня з: a1, b1, b2, b3, a3

Індивід 14: Клітинка_d8

Координати: d8

Колір клітинки: чорна

Сусідня з: d7, c7, e7, c8, e8

За таким принципом описано кожен клітинку шахової дошки, тобто кожна клітинка є індивідом (всього 64 клітинки).

«Хід»

Хід 5б Французька (хід5б_французька)

Цей індивід 15 є шаховим ходом, який здійснюється білими фігурами.

Номер ходу: 5

Позначення: Kf3

Фігура: білий кінь №2

Сторона: Білі

Клітинка, на яку здійснено хід: f3

Контекст: Хід є частиною французького захисту.

Індивід 16: Хід 5ч 2коні (хід5ч_2коні)

Номер ходу: 5

Позначення: Ka5

Фігура: чорний кінь №1

Сторона: Чорні

Клітинка, на яку здійснено хід: a5

Контекст: Хід є частиною варіанту із захистом двох коней (2коні).

Індивід 17: Хід 5ч Італійська (хід5ч_Італійська)

Цей індивід представляє п'ятий хід чорних у варіанті італійської партії.

Номер ходу: 5

Позначення: Kf6

Фігура: чорний кінь №2

Сторона: Чорні

Клітинка, на яку кінь був переміщений: f6

Індивід 18: Хід 5ч Каро-Канн (хід5ч_КароКанн)

Цей хід належить до захисту Каро-Канн.

Номер ходу: 5

Позначення: Kf6

Фігура: чорний кінь №2

Сторона: Чорні

Клітинка: f6

Примітка: За характеристиками цей хід ідентичний попередньому, але належить до іншої дебютної системи.

Індивід 19: Хід 5ч Сицилійка (хід5ч_сицилійка)

Індивід описує п'ятий хід чорних у сицилійському захисті.

Номер ходу: 5

Позначення: Kc6

Фігура: чорний кінь №1

Сторона: Чорні

Клітинка: c6

Індивід 20: Хід 5ч Французька (хід5ч_французька)

Цей індивід є віддзеркаленням французького захисту з боку чорних.

Номер ходу: 5

Позначення: Kf6

Фігура: чорний кінь №2

Сторона: Чорні

Клітинка: f6

Таким самим чином описано по 5 ходів за білих і за чорних до кожного дебюту.

«Шахова дошка»

Індивід: Шахова_Дошка

Розмір: 8x8 клітинок

Містить фігури: Король_білий, Кінь_чорний1, тощо

Початкове положення: стандартне розташування

Складається з: Клітинки

Наприклад, індивід хід5б_французька описує п'ятий хід білих у варіанті французького захисту, здійснений білим конем на клітинку f3. Інший індивід, хід5ч_сицилійка, представляє п'ятий хід чорних у сицилійському захисті, зроблений чорним конем на клітинку c6. Завдяки таким атрибутам онтологія дозволяє моделювати різні варіанти шахових партій із точністю до окремого ходу, фігури та сторони, яка його здійснила.

Ці індивіди дозволяють представити шахову партію в онтологічному вигляді, де кожен хід пов'язується з клітинкою, фігурою, стороною, дебютом або варіантом, а також має текстове позначення (у стандартній шаховій нотації) та номер. Такий підхід зручний для побудови логічних висновків, аналізу партій і створення навчальних систем для шахів.

У процесі створення індивідів у середовищі Protégé кожному індивіду можуть бути присвоєні атрибути, такі як "номер ходу", "клітинка, на яку відбувається хід", "фігура, яка здійснила хід", "сторона, що зробила хід", а також "позначення ходу". Такі характеристики надають конкретну інформацію щодо кожного шахового ходу в онтології та забезпечують можливість їх подальшого аналізу, наприклад, при вивченні дебютів, стилів гри або побудові шахових стратегій. Створення індивідів у Protégé показано на рисунках 2.9 - 2.12. Індивід хід1б_сицилійка є окремим випадком класу Хід, який описує конкретну шахову дію — перший хід білих у сицилійському захисті, тобто e4.

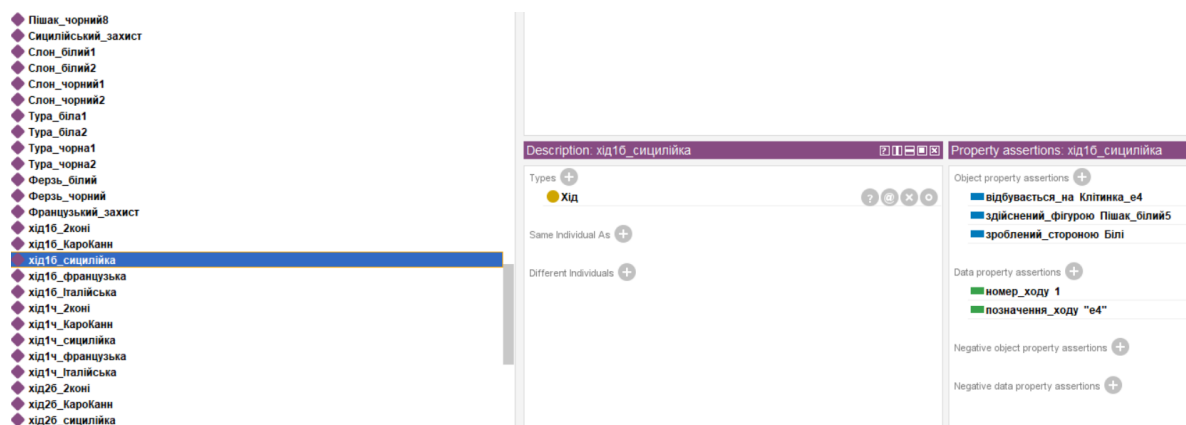


Рисунок 3.1 – Створення індивіда "хід1б_сицилійка" у Protégé

Індивід Клітинка_h4 представляє конкретну клітинку на шахівниці, розташовану на перетині вертикалі "h" (восьма горизонталь) та горизонталі "4". Цей індивід відіграє ключову роль у просторовому представленні шахової дошки в онтології. Він описує позицію клітинки, її колір (білий), сусідні клітинки, що важливо для моделювання можливих ходів фігур.

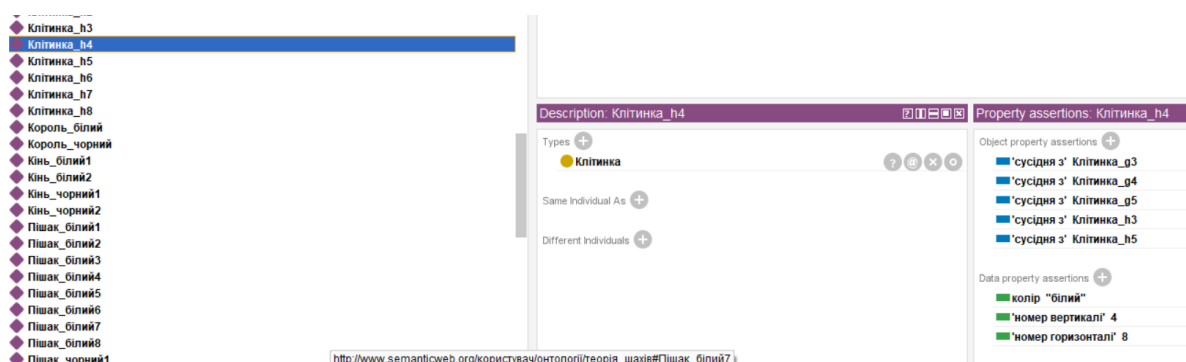


Рисунок 3.2 – Створення індивіда "Клітинка h4"

Сицилійський_захист – це конкретний шаховий дебют, широко вживаний у практиці гри чорними після 1.е4. У межах онтології цей індивід представляє систему початкових ходів, яка має назву "Сицилійський захист". Він включає перелік конкретних хід-індивідів (наприклад, хід1б_сицилійка, хід1ч_сицилійка і т.д.), які описують послідовність дій білих та чорних у перших 5 ходах.

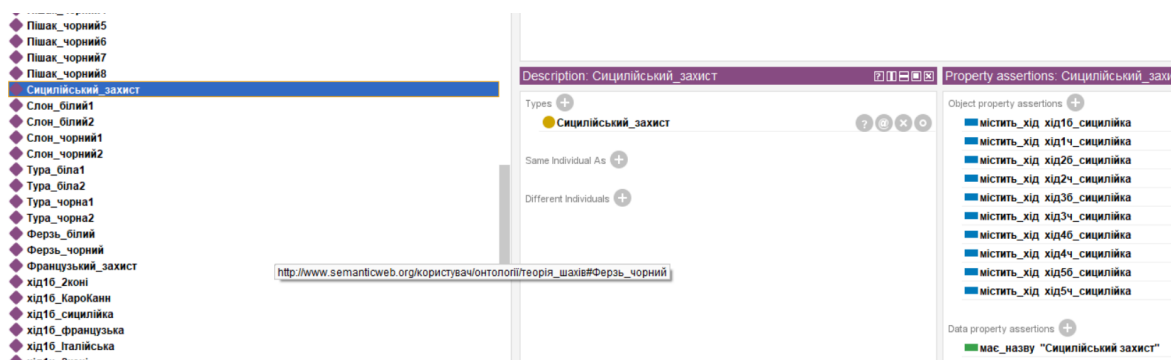


Рисунок 3.3 – Створення індивіда "Сицилійський захист"

Ферзь_білий — це індивід класу Ферзь, який описує конкретну фігуру білого кольору в шаховій грі. Вказується, що вона знаходиться на клітинці d1 (початкова позиція білого ферзя) та має білий колір.

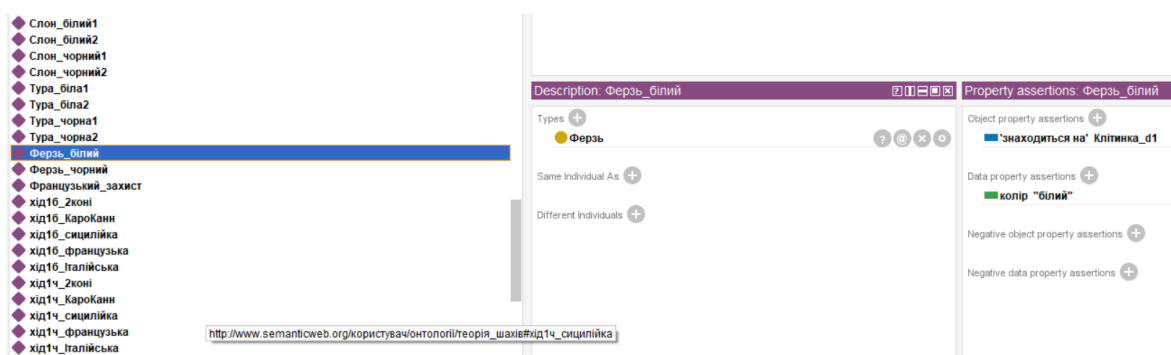


Рисунок 3.4 – Створення індивіда "Ферзь білий"

Створення індивідів дозволяє моделювати типові та унікальні сценарії шахової гри — наприклад, ходи у відомих дебютах (іспанський, сицилійський, ферзевий гамбіт), взаємодію фігур під час атаки або захисту, ситуації шаху, мату чи пату. Це не лише забезпечує наочну демонстрацію роботи онтології, а й робить її зручною для подальших досліджень, автоматичного аналізу шахових партій, створення інтелектуальних систем (наприклад, шахових тренерів, підказувачів або коментаторів) і побудови експертних систем у сфері шахів.

Завдяки цьому індивіди виконують ключову роль у формалізації знань про шахову гру — від базових понять (позицій, фігур, ходів) до складних концепцій (стратегічні схеми, відкриття, типові помилки та прийоми). Вони дозволяють

використовувати онтологію як ефективний інструмент для навчання, ретельного розбору партій, автоматизації аналізу і навіть як базу для створення гібридних систем зі штучним інтелектом.

3.2 Специфіка реалізації онтологічної бази знань з теорії шахів в середовищі Protege

Розробка онтології "Теорія шахів" здійснюється за допомогою мови OWL (Web Ontology Language) у програмному середовищі Protégé. Це забезпечує створення логічно структурованої моделі знань, яка легко розширюється, підтримує повторне використання та може бути інтегрована з іншими інтелектуальними системами [16]. Ключовим етапом розробки є наповнення онтології індивідами – конкретними об'єктами, що репрезентують шахові фігури, клітинки дошки, типові дебюти та їх ходи.

Процес створення онтологічної бази знань охоплює визначення основних класів і підкласів, налаштування їхніх властивостей (відношень), створення індивідів (екземплярів) і побудову візуального представлення моделі для наочного аналізу її структури та взаємозв'язків [17].

На початковому етапі формується ієрархія класів, яка відображає фундаментальні поняття шахової гри. Серед них — класи "Фігури", "Дошка", "Хід", "Дебют", "Сторона". Для кожного класу створені спеціалізовані підкласи, наприклад, "Ферзь", "Тура", "Кінь", "Сицилійський захист", "Іспанська партія", які уточнюють тип об'єкта. Завдяки такій структурі забезпечується гнучке та масштабоване представлення знань з шахової теорії. В онтологічному представленні класи з відповідною ієрархією підкласів виглядають наступним чином:

```
<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Італійська_парті
я -->
<owl:Classrdf:about="http://www.semanticweb.org/користувач/онтології/теорія_ш
```

```

ахів#Італійська_партія">
<rdfs:subClassOf rdfs:resource="http://www.semanticweb.org/користувач/онтології/
теорія_шахів#Дебюти"/> </owl:Class>
<!-- http://www.semanticweb.org/користувач/онтології/теорія_шахів#Дебюти -->
<owl:Class rdfs:about="http://www.semanticweb.org/користувач/онтології/теорія_ш
ахів#Дебюти"/>
<!-- http://www.semanticweb.org/користувач/онтології/теорія_шахів#Дошка -->
<owl:Class rdfs:about="http://www.semanticweb.org/користувач/онтології/теорія_ш
ахів#Дошка">
<rdfs:label xml:lang="uk">Дошка</rdfs:label>
</owl:Class>
<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Захист_КароКан
н →
<owl:Class rdfs:about="http://www.semanticweb.org/користувач/онтології/теорія_ш
ахів#Захист_КароКанн">
<rdfs:subClassOf rdfs:resource="http://www.semanticweb.org/користувач/онтології/
теорія_шахів#Дебюти"/></owl:Class>
<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Захист_двох_ко
нів -->
<owl:Class rdfs:about="http://www.semanticweb.org/користувач/онтології/теорія_ш
ахів#Захист_двох_конів">
<rdfs:subClassOf
rdfs:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Де
бюти"/></owl:Class>
<!-- http://www.semanticweb.org/користувач/онтології/теорія_шахів#Клітинка -->
<owl:Class rdfs:about="http://www.semanticweb.org/користувач/онтології/теорія_ш
ахів#Клітинка">
<rdfs:subClassOf rdfs:resource="http://www.semanticweb.org/користувач/онтології/

```

```
теорія_шахів#Дошка"/>
<rdfs:subClassOf>
<owl:Restriction>
```

Для створення графічного представлення онтології було використано плагін OntoGraph, доступний у середовищі Protégé. Завдяки цьому інструменту структуру онтології візуалізовано у формі орієнтованого графа, де вузли відповідають класам, підкласам або індивідам, а стрілки демонструють зв'язки між ними. Для покращення наочності використано автоматичне впорядкування елементів, що дало змогу чітко відобразити ієрархію класів і взаємозв'язки між ними [18]. На рисунку 3.5 представлено графічну модель онтології з теорії шахів.

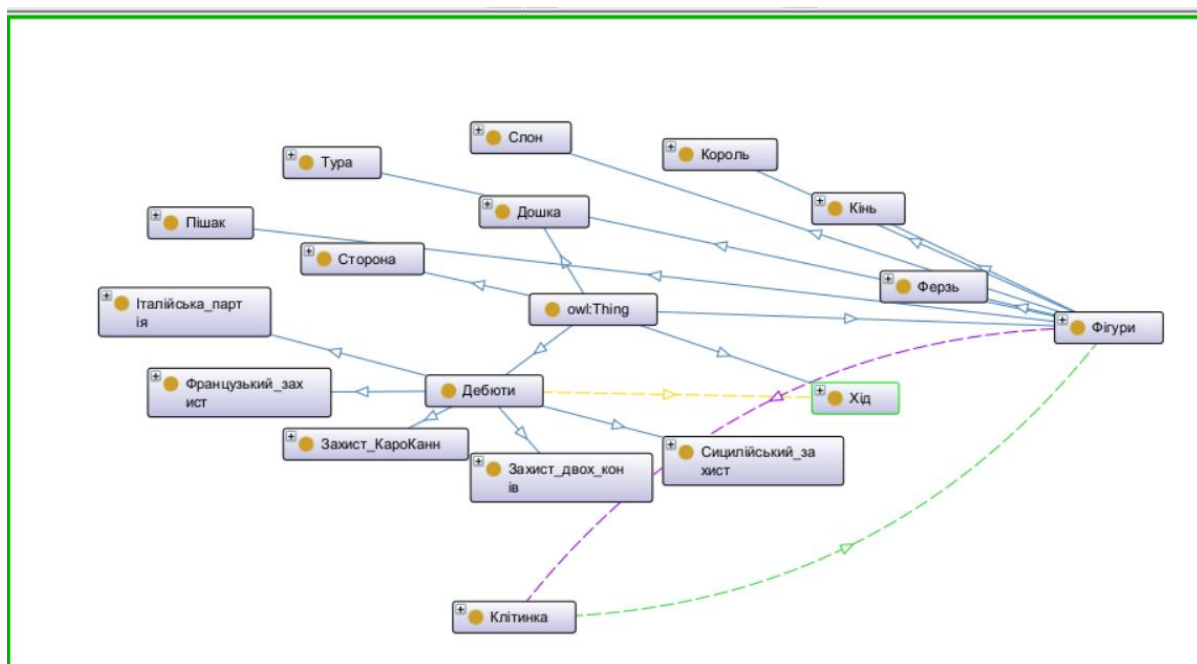


Рисунок 3.5 – Граф ієрархії класів онтології з теорії шахів

Для наочного відображення індивідів було використано плагін OntoGraph. На рисунку 3.6 представлено частину графічної структури онтології, що демонструє створені індивіди.

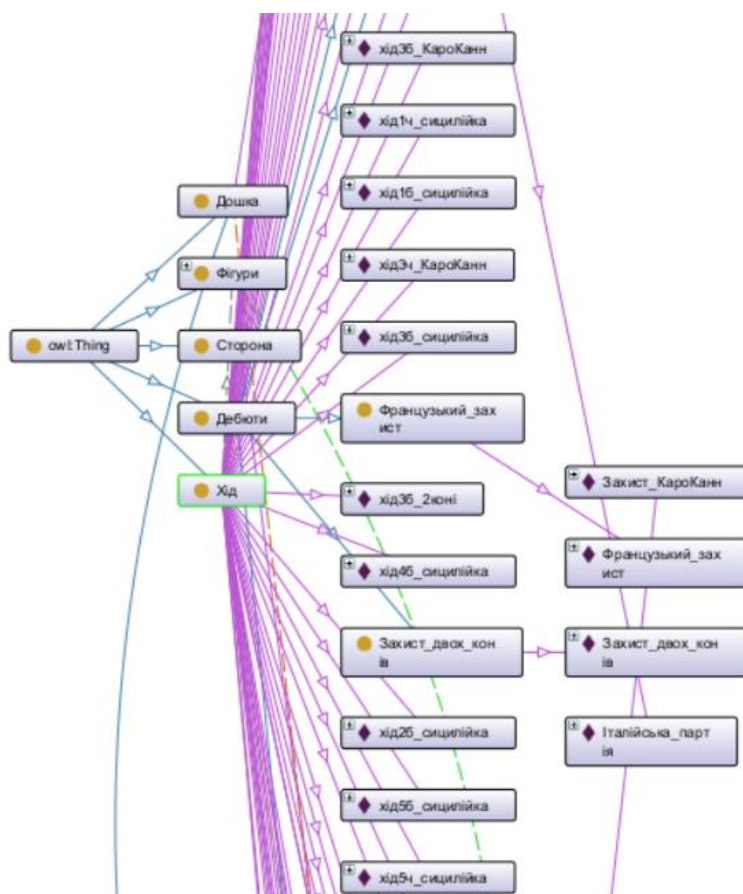


Рисунок 3.6 – Фрагмент графу індивідів онтології з теорії шахів

Після додавання індивідів у середовищі Protégé автоматично формується OWL-код, який описує ці індивіди, їхні властивості та взаємозв'язки у формалізованій формі. Такий код забезпечує чітке структурування інформації, що може бути інтегрована в інші системи або застосована для подальшого аналізу [17]. Приклад OWL-коду:

```
<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Італійська_партія
-->
<owl:NamedIndividualrdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Італійська_партія">
<rdf:typerdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Італійська_партія"/>
<містить_хідrdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід16_Італійська"/>
```

```

<містить_хідrdf:resource="http://www.semanticweb.org/користувач/онтології/тео
рія_шахів#хід1ч_Італійська"/>

<має_назву>Італійська партія</має_назву></owl:NamedIndividual>

<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Король_білий --
>

<owl:NamedIndividualrdf:about="http://www.semanticweb.org/користувач/онтолог
ії/теорія_шахів#Король_білий">

<rdf:typeperdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_
шахів#Король"/>

<знаходиться_наrdf:resource="http://www.semanticweb.org/користувач/онтології/
теорія_шахів#Клітинка_e1"/>

<колір>білий</колір>

</owl:NamedIndividual>

<!-- http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід1б_2коні
-->

<owl:NamedIndividualrdf:about="http://www.semanticweb.org/користувач/онтолог
ії/теорія_шахів#хід1б_2коні">

<rdf:typeperdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_
шахів#Хід"/>

<відбувається_наrdf:resource="http://www.semanticweb.org/користувач/онтології
/теорія_шахів#Клітинка_e4"/>

<здійснений_фігуроюrdf:resource="http://www.semanticweb.org/користувач/онто
логії/теорія_шахів#Пішак_білий5"/>

<зроблений_сторonoюrdf:resource="http://www.semanticweb.org/користувач/онт
ології/теорія_шахів#Білі"/>

<номер_ходуrdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</ном
ер_ходу>

<позначення_ходу>e4</позначення_ходу>

```


</owl:NamedIndividual>

У межах реалізації було створено набір класів, що відображають основні абстракції шахової теорії, а також конкретні індивіди — елементи ігрового процесу, такі як конкретні клітинки дошки, фігури певного кольору, дебютні побудови тощо. Особливу увагу приділено правильній побудові ієрархій, типізації об'єктів та визначенню семантичних відношень, що дозволяють виконувати логічні висновки на основі сформованої онтології.

3.3 Створення SPARQL запитів та тестування онтологічної бази знань із теорії шахів

SPARQL-запити в Protégé — це спеціальні запити, які використовуються для пошуку та витягу інформації з онтологій, створених у Protégé. SPARQL (SPARQL Protocol and RDF Query Language) — це мова запитів до баз знань, побудованих на основі RDF або OWL. SPARQL — це мова для запитів до семантичних баз даних, які зберігають знання у вигляді трикутників (subject–predicate–object) [19]. Вона дозволяє знаходити зв'язки між об'єктами в онтології, фільтрувати інформацію за певними умовами, також виводити дані у зрозумілому вигляді. У Protégé SPARQL-запити використовуються для пошуку індивідів (instances) певного класу, знаходження відносин (object properties) між об'єктами, аналізу структури онтології, отримання значень атрибутів (data properties), пошуку по ієрархії класів (наприклад, всіх підкласів або надкласів).

Загальна структура SPARQL-запиту. Запит складається з префіксів, оператору SELECT, оператору WHERE. Спочатку ми вказуємо, які простори імен (URI) будемо використовувати у скороченому вигляді. Це як "підключення бібліотек" — щоб не писати довгі адреси, а замінити їх скороченням (наприклад, : або rdf:). Оператор SELECT. Тут ми описуємо, яку інформацію хочемо отримати — наприклад, імена людей, їхній вік, компанії тощо. Всі елементи, які хочемо бачити у відповіді, позначаються змінними (починаються з ?, наприклад, ?name, ?age). Оператор WHERE. Далі йде основна частина — умови пошуку. Тут

ми описуємо структуру триплетів, які повинні виконуватись. Це формулювання типу: "Такий-то об'єкт належить до такого-то класу", "Має таку властивість", "Пов'язаний з іншим об'єктом". Інакше кажучи: кого шукаємо і що про них маємо знати. Фільтри (необов'язково) [20]. Можна уточнити запит, використовуючи умови. Наприклад, тільки ті, в кого вік більше за 18, або ті, хто працює саме в певній організації. Необов'язкові частини (OPTIONAL). Якщо хочемо отримати додаткову інформацію, яка може бути, а може й ні, то додаємо це як необов'язкове — щоб не втратити основні результати. Короткий опис компонентів SPARQL-запиту наведено у таблиці 3.1.

Таблиця 3.1 – Компоненти SPARQL-запиту

Елемент	Опис
PREFIX	Визначає префікс (коротку назву) для URI вашої онтології.
SELECT	Визначає змінні (починаються з ?), які ви хочете вивести.
WHERE	Визначає шаблон (патерн), за яким здійснюється пошук.
FILTER	Фільтрує результати (наприклад, FILTER (?age > 20)).
OPTIONAL	Додає необов'язкову частину до запиту (як LEFT JOIN).

Тепер перейдемо до безпосереднього створення SPARQL-запитів для онтології із теорії шахів. Першим буде запит, що виводить всю ієрархію класів в онтології, показуючи, який клас є "дитиною" іншого класу. Такий запит дозволяє проаналізувати загальну структуру онтології та перевірити правильність побудованої ієрархії понять. Це особливо корисно на етапі тестування, коли потрібно переконатися, що всі класи були правильно організовані відповідно до предметної області. Його показано на рисунку 3.1.



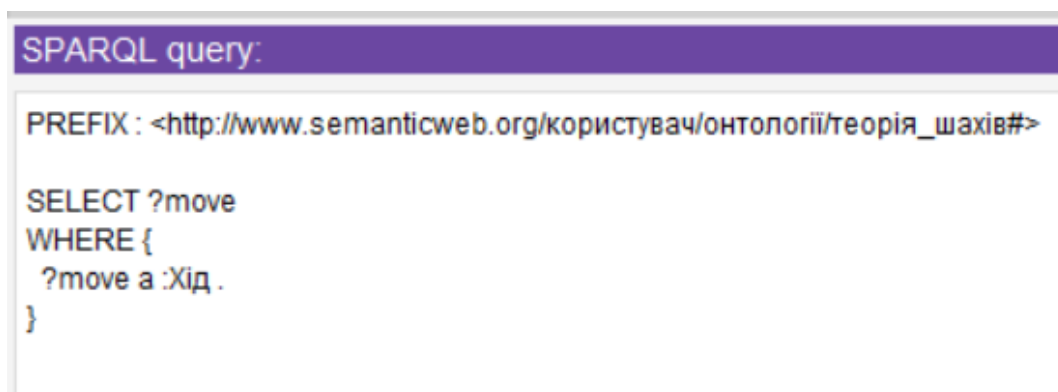
Рисунок 3.1 – Створений SPARQL запит для виведення ієрархії

PREFIX, це стандартні оголошення префіксів для різних онтологічних мов та стандартів (RDF, OWL, RDFS, XSD). Вони дозволяють використовувати короткі псевдоніми замість повних URI. SELECT ?subject ?object: Цей рядок вказує, які змінні ми хочемо бачити у результаті запиту. У цьому випадку, це будуть дві колонки: ?subject і ?object. WHERE { ?subject rdfs:subClassOf ?object }: Це основна частина запиту, яка визначає шаблон, що ми шукаємо в онтології. Результат запиту зображено на рисунку 3.2.

subject	
Захист_двох_конів	Дебюти
Король	Фігури
Французький_захист	Дебюти
Клітинка	Дошка
Пішак	Фігури
Захист_КароКанн	Дебюти
Слон	Фігури
Сицилійський_захист	Дебюти
Кінь	Фігури
Ферзь	Фігури
Тура	Фігури
Італійська_партія	Дебюти
Клітинка	● 'знаходиться на' max 1 Фігури

Рисунок 3.2 – Результат виконання SPARQL запиту для виведення ієрархії

Далі створимо запит, щоб побачити усіх індивідів класу хід. Його зображено на рисунку 3.3.



```
SPARQL query:

PREFIX : <http://www.semanticweb.org/користувач/онтології/теорія_шахів#>

SELECT ?move
WHERE {
  ?move a :Хід .
}
```

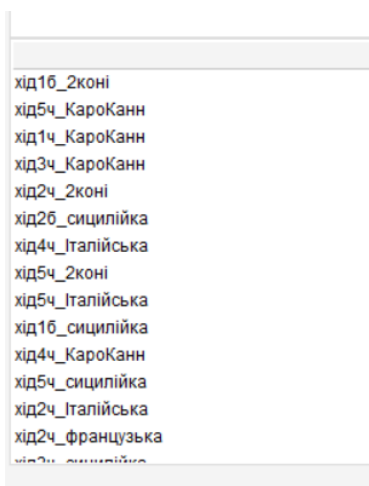
Рисунок 3.3 – Створений SPARQL запит для виведення усіх індивідів класу хід

Даний запит призначений для отримання всіх індивідів (екземплярів), які належать до класу : “Хід” в онтології теорії шахів. Компоненти запиту:

PREFIX : <...> — задає простір імен (namespace), який скорочує повний URI.

SELECT ?move — виводить усі значення змінної ?move, які відповідають умові з WHERE.

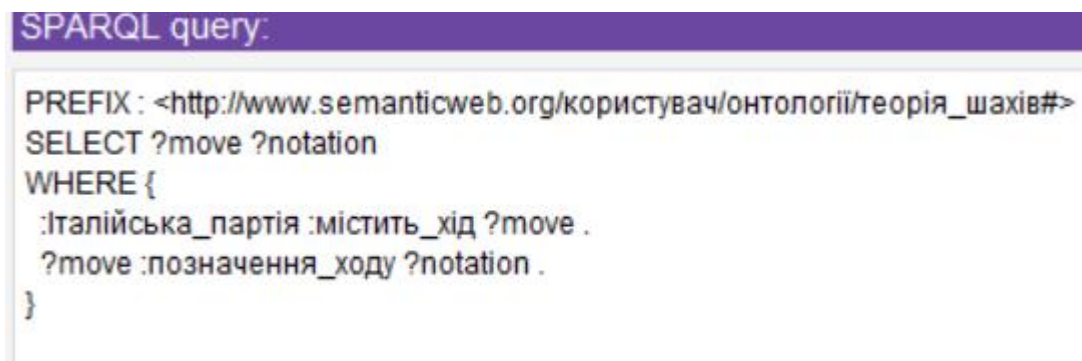
WHERE { ?move a :Хід . } — умова, яка шукає всі ресурси ?move, які є екземплярами (а — це скорочення для rdf:type) класу :Хід. У відповідь на даний запит отримали список всіх індивідів, які представляють шахові ходи, що зображено на рисунку 3.4.



```
хід16_2коні
хід54_КароКанн
хід14_КароКанн
хід34_КароКанн
хід24_2коні
хід26_сицилійка
хід44_Італійська
хід54_2коні
хід54_Італійська
хід16_сицилійка
хід44_КароКанн
хід54_сицилійка
хід24_Італійська
хід24_Французька
хід34_сицилійка
```

Рисунок 3.4 – Результат виконання SPARQL запиту для виведення усіх індивідів класу хід

Також, для більш наглядного тестування створимо запит, який витягує інформацію про ходи, які входять до "Італійської партії" та позначення кожного ходу. Даний запит зображено на рисунку 3.5.



```
SPARQL query:
PREFIX : <http://www.semanticweb.org/користувач/онтології/теорія_шахів#>
SELECT ?move ?notation
WHERE {
  :Італійська_партія :містить_хід ?move .
  ?move :позначення_ходу ?notation .
}
```

Рисунок 3.5 – Створений SPARQL запит для виведення усіх ходів, що належать до італійської партії

Компоненти запиту:

PREFIX : <...> — оголошення простору імен (namespace) онтології.

:Італійська_партія — індивід, що представляє певну шахову партію (у даному випадку — "Італійську партію").

:має_хід — об'єктна властивість, яка пов'язує партію з індивідами типу :Хід.

?move — змінна, що означає індивіда ходу, який належить до вказаної партії.

:позначення_ходу — дата-властивість (data property), яка повертає текстову нотацію шахового ходу (наприклад, e4, Nf3, тощо). У результаті отримали усі ходи, що входять до Італійської партії. У результаті виконання запиту буде отримано перелік усіх ходів, що входять до складу Італійської партії, із вказаними нотаціями. Це дає змогу аналізувати структуру дебюту, здійснювати порівняння з іншими партіями, а також використовувати дані для навчання або побудови стратегій. Результат виконання запиту зображено на рисунку 3.6.

move	
хід1б_Італійська	"e4"
хід1ч_Італійська	"e5"
хід3б_Італійська	"Cc4"
хід5ч_Італійська	"Kf6"
хід4б_Італійська	"d3"
хід4ч_Італійська	"d6"
хід3ч_Італійська	"Cc5"
хід5б_Італійська	"Kc3"
хід2ч_Італійська	"Kc6"
хід2б_Італійська	"Kf3"

Рисунок 3.6 – Результат виконання SPARQL запиту для виведення усіх ходів, що належать до італійської партії

Створення та тестування SPARQL-запитів підтвердило цілісність і функціональність онтології з теорії шахів. Запити дозволили отримувати структуровані дані про ходи, партії та інші елементи, демонструючи ефективність зберігання й доступу до знань. Це свідчить про практичну придатність моделі для навчання, аналізу та програмних застосунків.

Висновок до розділу 3

У даному розділі було розглянуто процес реалізації онтологічної бази знань із теорії шахів. Насамперед було згенеровано програмний код онтології у форматі OWL за допомогою програмного середовища Protégé. У процесі моделювання до онтології були додані індивіди — конкретні приклади шахових фігур, ходів, дебютів тощо, що наповнили модель фактичним змістом.

Також здійснено візуалізацію структури онтології за допомогою плагіна OntoGraph, який дав змогу побачити повну ієрархію класів, їхні підкласи та зв'язки у вигляді графа.

Фінальним етапом реалізації стала побудова та тестування SPARQL-запитів, які дозволили здійснювати пошук, фільтрацію та витяг структурованих даних з онтологічної бази. Завдяки цим запитам була перевірена правильність визначених зв'язків та здатність моделі відповідати на логічні питання, зокрема отримувати всі ходи, що входять до певної партії, або виводити їх нотації.

ВИСНОВКИ

Всі задачі, поставлені для реалізації онтологічної бази знань із теорії шахів були виконані в повному обсязі, а саме – проаналізовано предметну область та обґрунтовано доцільність створення онтологічної бази знань з теорії шахів; проведено аналіз та порівняння існуючих технологій; проведено моделювання та проектування бази знань з теорії шахів; здійснено програмну реалізацію онтологічної бази знань; виконано тестування онтологічної бази знань з теорії шахів за допомогою SPARQL запитів.

На етапі теоретичного обґрунтування проаналізовано взаємозв'язок між сучасними інформаційними технологіями та теорією шахів, окреслено історію виникнення онтологічного підходу у сфері ІТ, а також проведено огляд актуальних технологічних рішень, що використовуються для вивчення шахової теорії.

У процесі виконання роботи було здійснено моделювання предметної області шахової теорії із використанням засобів онтологічного проектування. Визначено основні класи, їхні властивості та семантичні зв'язки, що відображають ключові елементи шахової гри, зокрема фігури, ходи, дебюти, партії. Онтологію реалізовано в середовищі Protégé, де сформовано логічні зв'язки між класами та індивідами.

На фінальному етапі згенеровано OWL-код та створено SPARQL-запити для перевірки онтології. Тестування підтвердило її логічну цілісність і придатність для навчання, розширення та інтеграції в інтелектуальні системи.

Мета роботи була досягнута шляхом розширення функціональних можливостей онтологічної бази знань у сфері теорії шахів за рахунок інтеграції семантичних зв'язків між поняттями, а також можливості логічного виведення нових знань, що суттєво переважає традиційні бази даних.

Результати розробки підтверджують доцільність створення онтологічної бази знань із теорії шахів і демонструють її потенціал для подальшого вдосконалення та розширення функціональності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. O. V. Silagin, K. M. Stepaniuk. Ontology for chess theory. Молодь в науці: дослідження, проблеми, перспективи (МН-2025). Вінницький національний технічний університет. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/25355/20970>
2. Комп'ютерні шахи [Електронний ресурс]. – URL: https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%96_%D1%88%D0%B0%D1%85%D0%B8
3. Стокфіш (шахи) [Електронний ресурс]. – URL: [https://en.wikipedia.org/wiki/Stockfish_\(chess\)](https://en.wikipedia.org/wiki/Stockfish_(chess))
4. Поняття онтології [Електронний ресурс]. – URL: [https://uk.wikipedia.org/wiki/%D0%9E%D0%BD%D1%82%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F_\(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0\)](https://uk.wikipedia.org/wiki/%D0%9E%D0%BD%D1%82%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0))
5. Інформаційно-теоретичний аналіз рішень у комп'ютерних шахах [Електронний ресурс]. – URL: <https://arxiv.org/abs/1112.2144>
6. Gruber, TR. (1993). A Translation Approach to Portable Ontology Specification. Knowledge Acquisition 5: 199-220.
7. Сім кроків для розробки онтології [Електронний ресурс]. – URL: https://www.researchgate.net/figure/Seven-Steps-to-Develop-Ontology-20_fig1_303704422
8. Проектування онтології для шахових ігор [Електронний ресурс]. – URL: https://ceur-ws.org/Vol-1461/WOP2015_pattern_abstract_2.pdf
9. Дебютне дерево [Електронний ресурс]. – URL: <https://en.chessbase.com/support-kb/content/details/725/Creating%20your%20own%20opening%20books>

10. Офіційний сайт Protégé [Електронний ресурс]. – URL: <https://protege.stanford.edu/>
11. Документація Protégé [Електронний ресурс]. – URL: <https://protegeproject.github.io/protege/>
12. Rothenfluh, T.R., Gennari, J.H., Eriksson, H., Puerta, A.R., Tu, S.W. і Musen, M.A. (1996). Використовувані онтології, знання-пристосування інструментів, і розробка систем: PROTEGE-II solutions to Sisyphus-2.
13. Hendler, J. and McGuinness, D.L. (2000). The DARPA AgentMarkup Language. *IEEE Intelligent Systems* 16(6): 67-73.
14. M. Blazquez, M. Fernandez, J. M. Garcia-Pinar, and A. Gomez-Perez. Building ontologies at the knowledge level using the ontology design environment. In *Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98, Banff, Canada, 1998*.
15. Natalya F. Noy, Deborah L. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.
16. Інструкція по створенню онтологій в Protege [Електронний ресурс]. – URL: <https://medium.com/@vindulajayawardana/ontology-generation-and-visualization-with-prot%C3%A9g%C3%A9-6df0af9955e0>
17. Web Ontology Language (OWL) [Електронний ресурс]. – URL: <https://www.w3.org/OWL/>
18. Реалізація OntoGraph в Protege [Електронний ресурс]. – URL: <https://protegewiki.stanford.edu/wiki/OntoGraf>
19. Інструкція по створенню SPARQL запитів [Електронний ресурс]. – URL: <https://sachipcranasinghe.medium.com/intro-to-sparql-part1-103957d8e854>

20. SPARQL Query Beginner's Guide [Електронний ресурс]. – URL: https://docs.data.world/tutorials/sparql/primer_intro.html
21. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. – Вінниця : ВНТУ, 2023. – (PDF, 54 с.).

ДОДАТКИ

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка онтологічної бази знань із теорії шахівТип 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 5,12 %

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)

Степанюк К.М.

(прізвище, ініціали)

ДОДАТОК Б (обов'язковий)

Лістинг програми

```

<?xmlversion="1.0"?><rdf:RDF
xmlns="http://www.semanticweb.org/користувач/онтології/теорія_шахів#"
xml:base="http://www.semanticweb.org/користувач/онтології/теорія_шахів"
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:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:теорія_шахів="http://www.semanticweb.org/користувач/онтології/теорія_шахів"><owl:Ontology
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів"/>
<!-- Object Properties -->
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#відбувається_на"><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хід"/><rdfs:range
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Клітинка"/></owl:ObjectProperty>
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#здійснений_фігурою"><rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хід"/><rdfs:range
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Фігури"/></owl:ObjectProperty>
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#знаходиться_на"><rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Фігури"/><rdfs:range
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Клітинка"/><rdfs:label
xml:lang="uk">знаходиться
на</rdfs:label></owl:ObjectProperty>

```

```

<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#зробл
ений_стороною"><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><rdfs:range
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Ст
орона"/></owl:ObjectProperty>
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#місти
ть_хід"><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Де
бюти"/><rdfs:range
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/></owl:ObjectProperty>
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#склад
ається_з"><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#До
шка"/><rdfs:range
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка"/><rdfs:label
xml:lang="uk">складається
з</rdfs:label></owl:ObjectProperty>
<owl:ObjectProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#сусід
ня_з"><rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/><rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка"/><rdfs:range
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка"/><rdfs:label xml:lang="uk">сусідня з</rdfs:label></owl:ObjectProperty>
<!-- Data properties -->
<owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#колір
"><rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка"/><rdfs:domain
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Фі

```

```

    гури"/><rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string"/><rdfs:label
    xml:lang="uk">коліп</rdfs:label></owl:DatatypeProperty>
<owl:DatatypeProperty
    rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#має_
    назву"><rdfs:domain
    rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Де
    бюти"/><rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string"/></owl:DatatypeProp
    erty>
<owl:DatatypeProperty
    rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#номе
    р_вертикалі"><rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/><rdfs:domain
    rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
    ітинка"/><rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/><rdfs:label
    xml:lang="uk">номер вертикалі</rdfs:label></owl:DatatypeProperty>
<owl:DatatypeProperty
    rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#номе
    р_горизонталі"><rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/><rdfs:domain
    rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
    ітинка"/><rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/><rdfs:label
    xml:lang="uk">номер горизонталі</rdfs:label></owl:DatatypeProperty>
<owl:DatatypeProperty
    rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#номе
    р_ходу"><rdfs:domain
    rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
    д"/><rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/></owl:DatatypeProp
    erty>
<owl:DatatypeProperty
    rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#позна
    чення_ходу"><rdfs:domain
    rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
    д"/><rdfs:range

```

```

rdf:resource="http://www.w3.org/2001/XMLSchema#string"/></owl:DatatypeProperty>
<!-- Classes -->
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Італійська_партія"><rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Дебюти"/></owl:Class>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Атак_Фішера"><rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Дебюти"/></owl:Class>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Відкриття_партії"><rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Дебюти"/></owl:Class>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Дебюти"/>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Клітинка"/>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Фігури"/>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Сторона"/>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Дошка"/>
<owl:Class
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хід"/>
</rdf:RDF>

```



```

<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Сицилійський_з
ахист -->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Сици
лійський_захист">
<rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Си
цилійський_захист"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
1б_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
1ч_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
2б_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
2ч_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
3б_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
3ч_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
4б_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
4ч_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
5б_сицилійка"/>
<містить_хід
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід
5ч_сицилійка"/>

```

```
<має_назву>Сицилійський захист</має_назву>
</owl:NamedIndividual>
```

```
<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон_білий1 -->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон
_білий1">
<rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Сл
он"/>
<знаходиться_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_с1"/>
<колір>білий</колір>
</owl:NamedIndividual>
```

```
<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон_білий2 -->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон
_білий2">
<rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Сл
он"/>
<знаходиться_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_f1"/>
<колір>білий</колір>
</owl:NamedIndividual>
```

```
<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон_чорний1 -
->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон
_чорний1">
```

```

<rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Сл
он"/>
<знаходиться_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_с8"/>
<колір>чорний</колір>
</owl:NamedIndividual>

```

```

<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон_чорний2 -
->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Слон
_чорний2">
<rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Сл
он"/>
<знаходиться_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_f8"/>
<колір>чорний</колір>
</owl:NamedIndividual>

```

```

<!-- http://www.semanticweb.org/користувач/онтології/теорія_шахів#Тура_біла1 -
->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Тура
_біла1">
<rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Ту
ра"/>
<знаходиться_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_a1"/>
<колір>білий</колір>
</owl:NamedIndividual>

```

```

<!-- http://www.semanticweb.org/користувач/онтології/теорія_шахів#Тура_біла2 -
->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Тура
_біла2">
<rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Ту
ра"/>
<знаходиться_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_h1"/>
<колір>білий</колір>
</owl:NamedIndividual>
<!-- http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід1б_2коні
--><owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід1б
_2коні"><rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><відбувається_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_e4"/><здійснений_фігурою
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Пі
шак_білий5"/><зроблений_сторonoю
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Бі
лі"/><номер_ходу
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</номер_ходу><по
значення_ходу>e4</позначення_ходу></owl:NamedIndividual>

<!--
http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід1б_Італійськ
а
--><owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід1б
_Італійська"><rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><відбувається_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_e4"/><здійснений_фігурою
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Пі
шак_білий5"/><зроблений_сторonoю

```

```

rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Бі
лі"/><номер_ходу
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</номер_ходу><по
значення_ходу>e4</позначення_ходу></owl:NamedIndividual>

```

```
<!--
```

```

http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід16_КароКанн
--><owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід16
_КароКанн"><rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><відбувається_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_e4"/><здійснений_фігурою
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Пі
шак_білий5"/><зроблений_стороною
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Бі
лі"/><номер_ходу
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</номер_ходу><по
значення_ходу>e4</позначення_ходу></owl:NamedIndividual>

```

```
<!--
```

```

http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід16_сицилійка
--><owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід16
_сицилійка"><rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><відбувається_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_e4"/><здійснений_фігурою
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Пі
шак_білий5"/><зроблений_стороною
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Бі
лі"/><номер_ходу
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</номер_ходу><по
значення_ходу>e4</позначення_ходу></owl:NamedIndividual>

```

```
<!--
```

```

http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід16_французь

```

ка --><owl:NamedIndividual
 rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід16_французька"><rdf:type
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хід"/><відбувається_на
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Клітинка_e4"/><здійснений_фігурою
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Пішак_білий5"/><зроблений_сторonoю
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Білі"/><номер_ходу
 rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</номер_ходу><по_значення_ходу>e4</позначення_ходу></owl:NamedIndividual>
 <owl:NamedIndividual
 rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід5ч_2коні"><rdf:type
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хід"/><відбувається_на
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Клітинка_a5"/><здійснений_фігурою
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кінь_чорний1"/><зроблений_сторonoю
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Чорні"/><номер_ходу
 rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">5</номер_ходу><по_значення_ходу>Ka5</позначення_ходу></owl:NamedIndividual>
 <owl:NamedIndividual
 rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід5ч_Італійська"><rdf:type
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хід"/><відбувається_на
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Клітинка_f6"/><здійснений_фігурою
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кінь_чорний2"/><зроблений_сторonoю
 rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Чорні"/><номер_ходу

```
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">5</номер_ходу><по
значення_ходу>Kf6</позначення_ходу></owl:NamedIndividual>
```

```
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід5ч
_КапоКанн"><rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><відбувається_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_f6"/><здійснений_фігурою
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кі
нь_чорний2"/><зроблений_сторonoю
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Чо
рні"/><номер_ходу
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">5</номер_ходу><по
значення_ходу>Kf6</позначення_ходу></owl:NamedIndividual>
```

```
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід5ч
_сицилійка"><rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><відбувається_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_c6"/><здійснений_фігурою
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кі
нь_чорний1"/><зроблений_сторonoю
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Чо
рні"/><номер_ходу
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">5</номер_ходу><по
значення_ходу>Kc6</позначення_ходу></owl:NamedIndividual>
```

```
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/користувач/онтології/теорія_шахів#хід5ч
_французька"><rdf:type
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Хі
д"/><відбувається_на
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кл
ітинка_f6"/><здійснений_фігурою
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Кі
```

```

нь_чорний2"/><зроблений_стороною
rdf:resource="http://www.semanticweb.org/користувач/онтології/теорія_шахів#Чо
рні"/><номер_ходу
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">5</номер_ходу><по
значення_ходу>Kf6</позначення_ходу></owl: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****РОЗРОБКА ОНТОЛОГІЧНОЇ БАЗИ ЗНАНЬ ІЗ ТЕОРІЇ ШАХІВ**

Виконала: студентка 4 курсу, групи ЗКН-216
спеціальності 122 Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)

Степанюк К.М.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Сілагін О. В.

(прізвище та ініціали)

« 03 » серпня 2025 р.

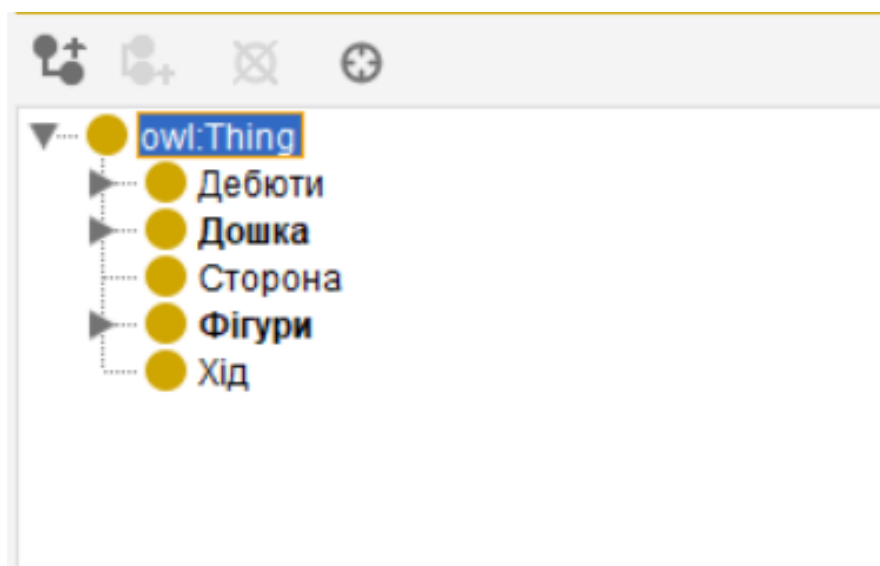


Рисунок В.1 – Класи онтологічної бази знань

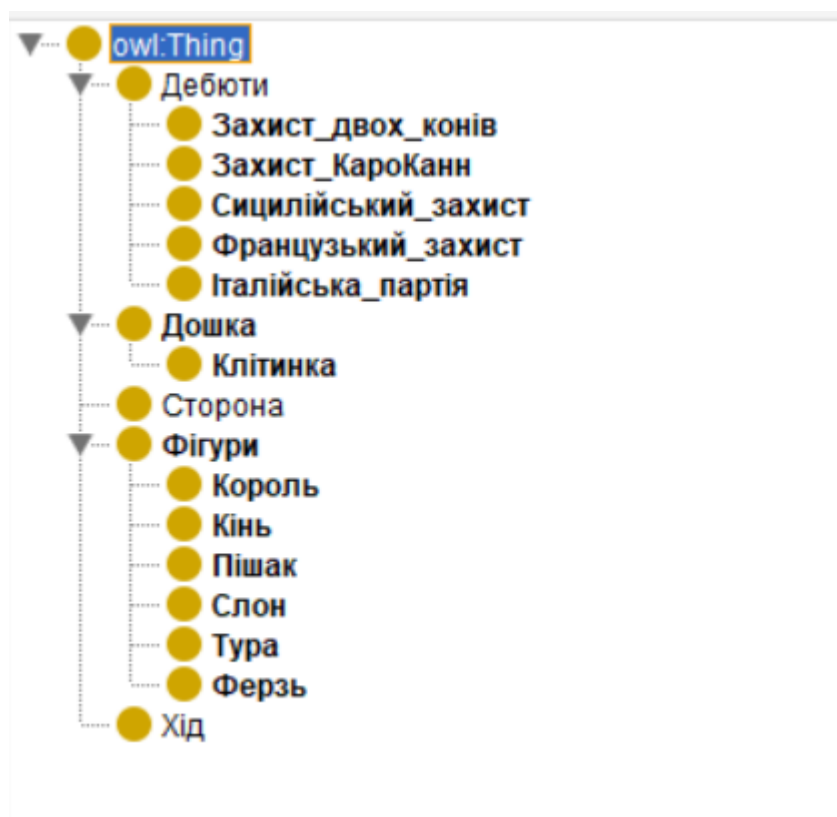


Рисунок В.2 – Ієрархія класів онтологічної бази знань

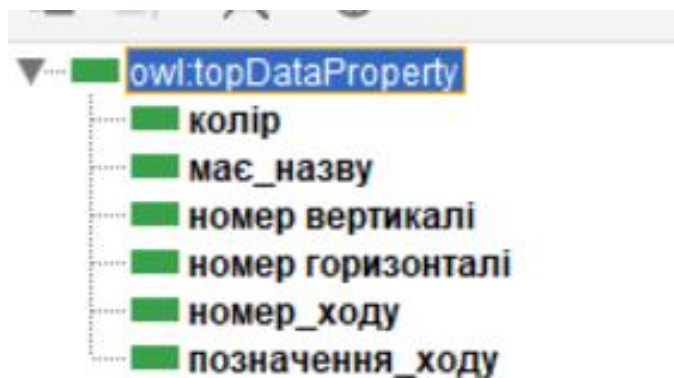


Рисунок В.3 – Атрибути класів онтологічної бази знань

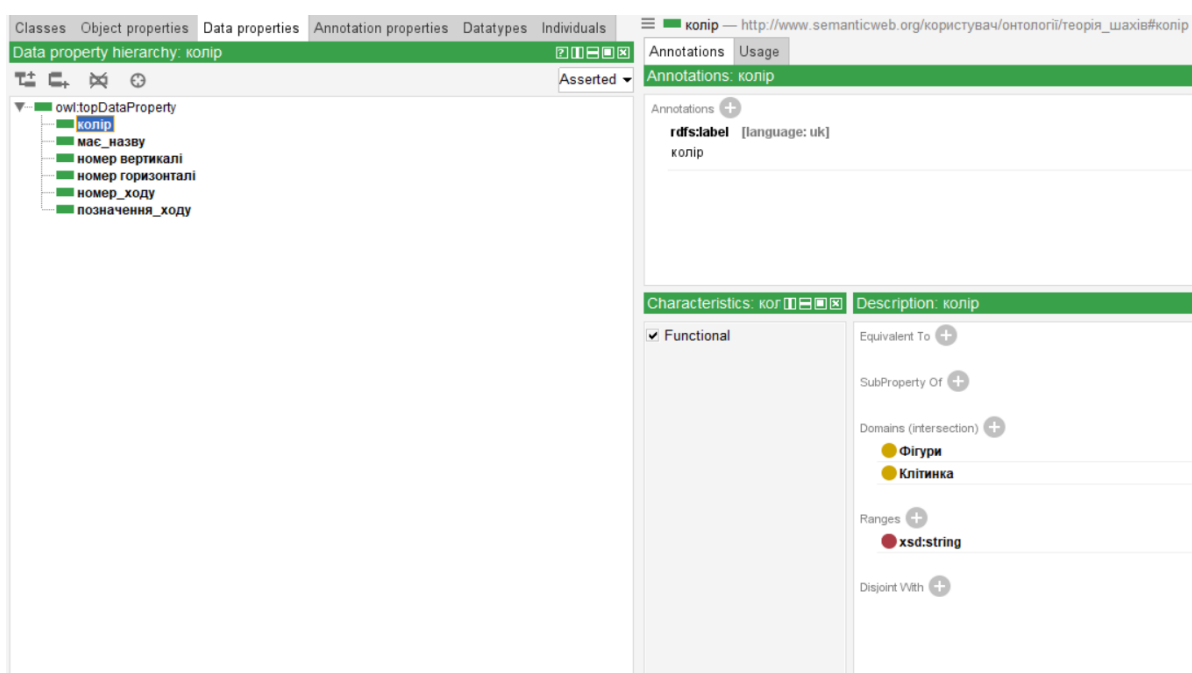


Рисунок В.4 – Атрибут “Колір”

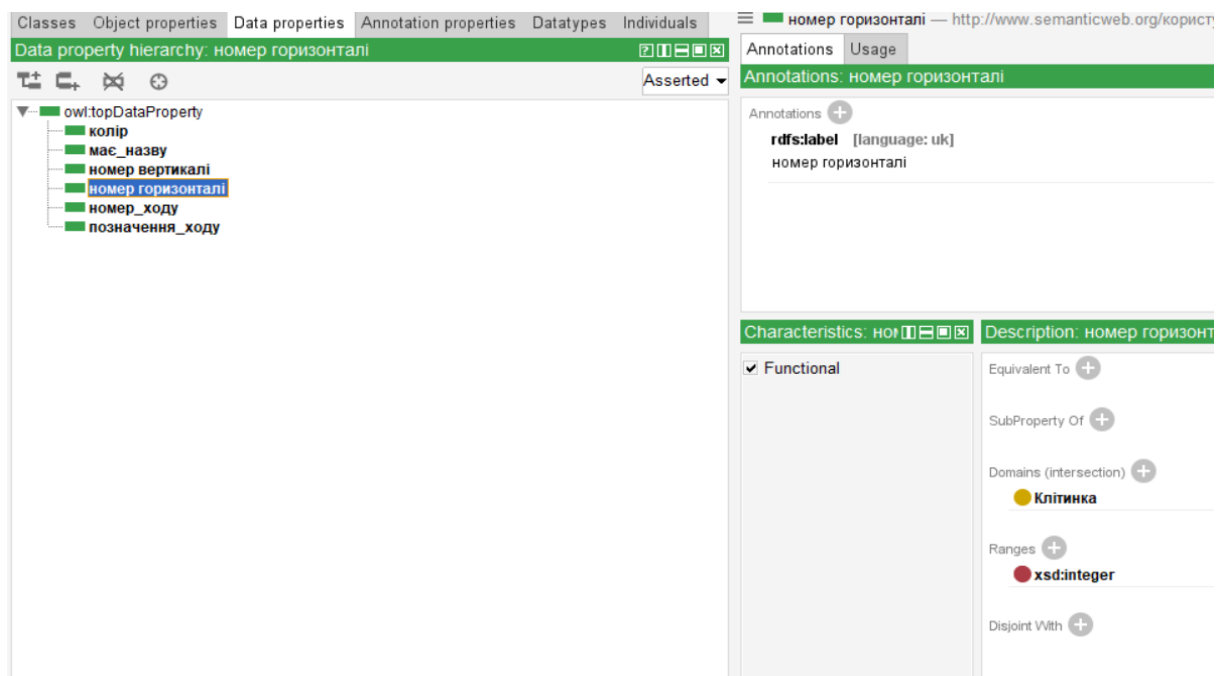


Рисунок В.5 – Атрибут “Номер горизонталі”

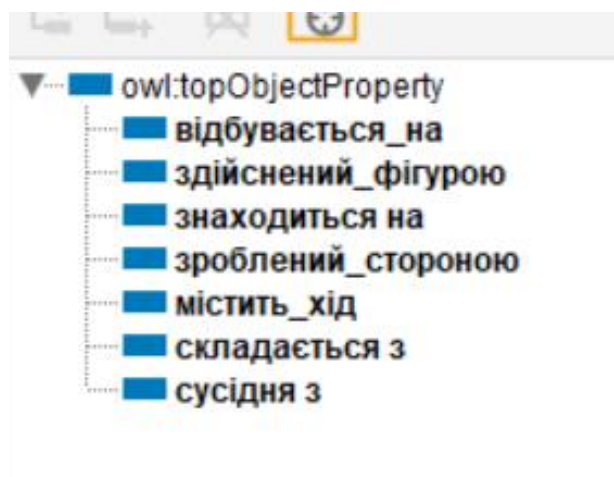


Рисунок В.6 – Об’єктні властивості онтологічної бази знань

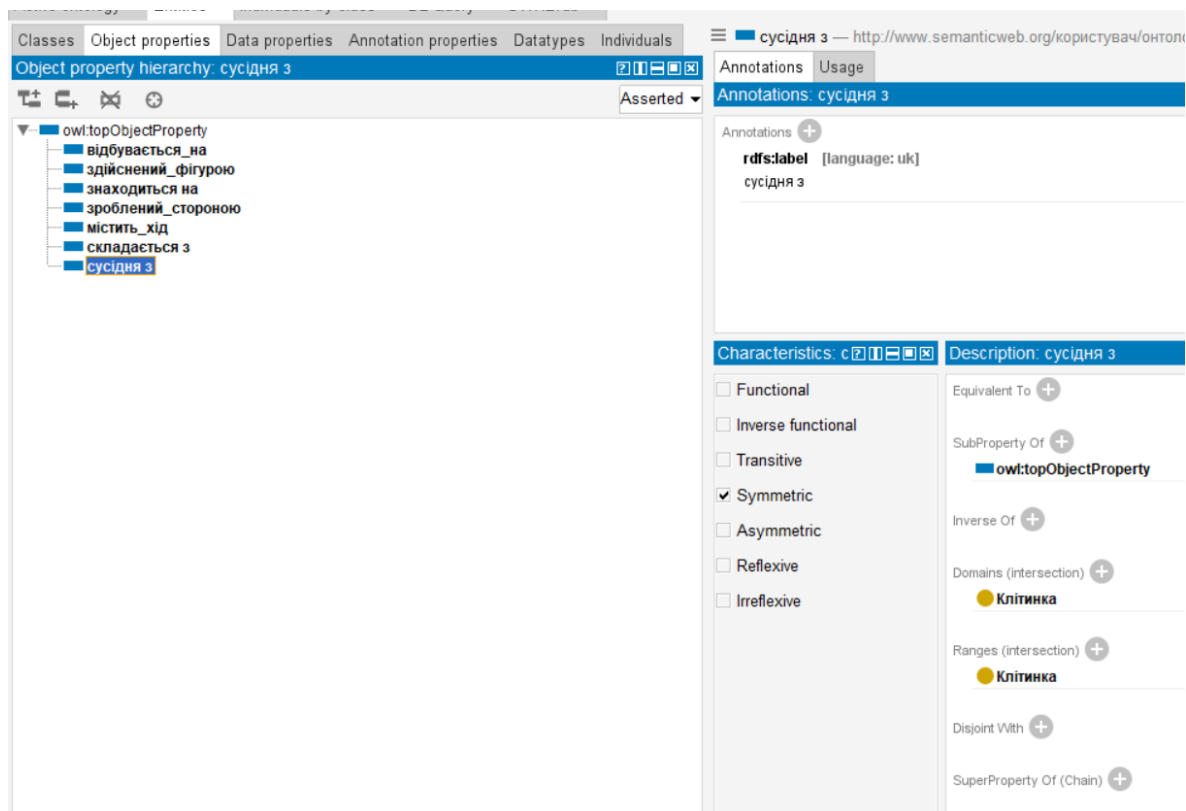


Рисунок В.7 – Об’єктна властивість “Сусідня з”

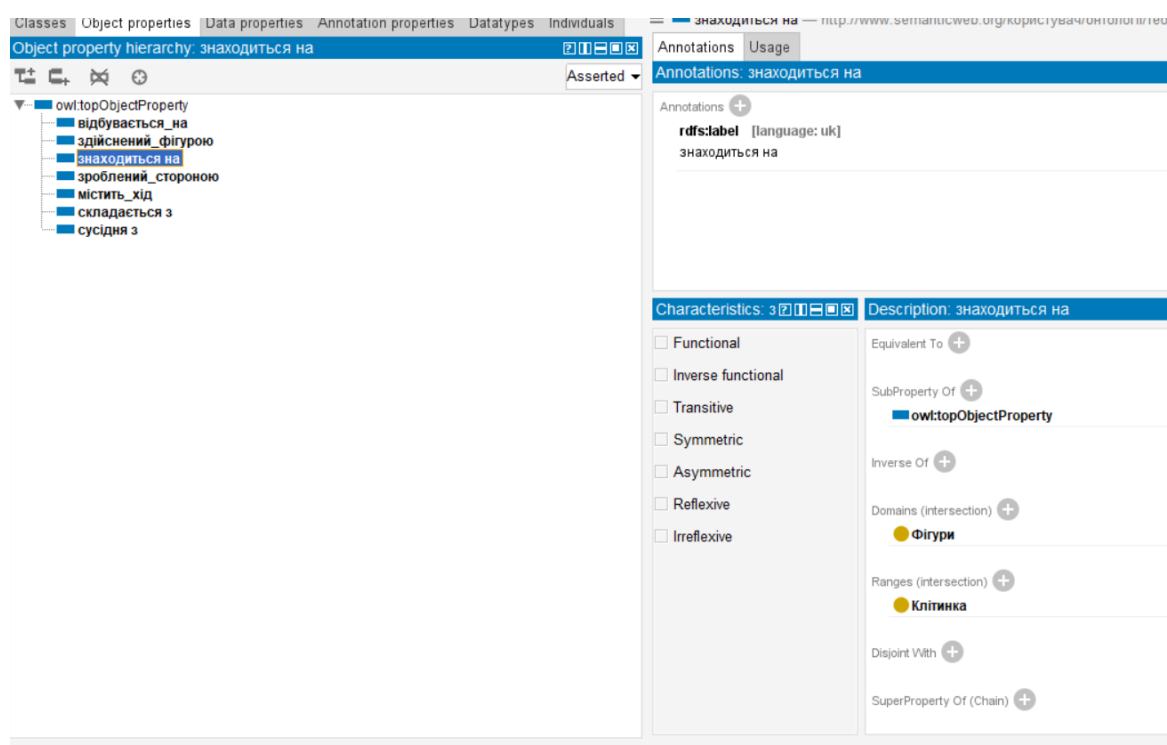


Рисунок В.8 – Об’єктна властивість “Знаходиться на”

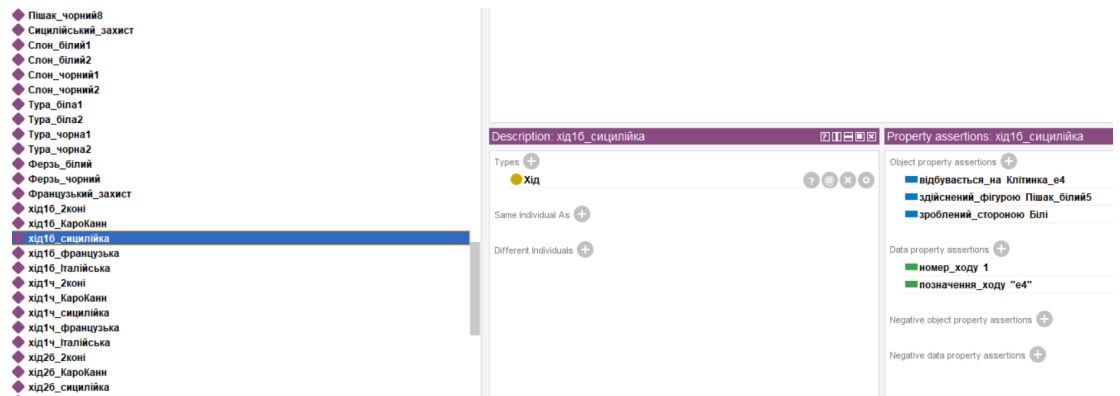


Рисунок В.9 – Створення індивіда "хід16_сицилійка" у Protégé

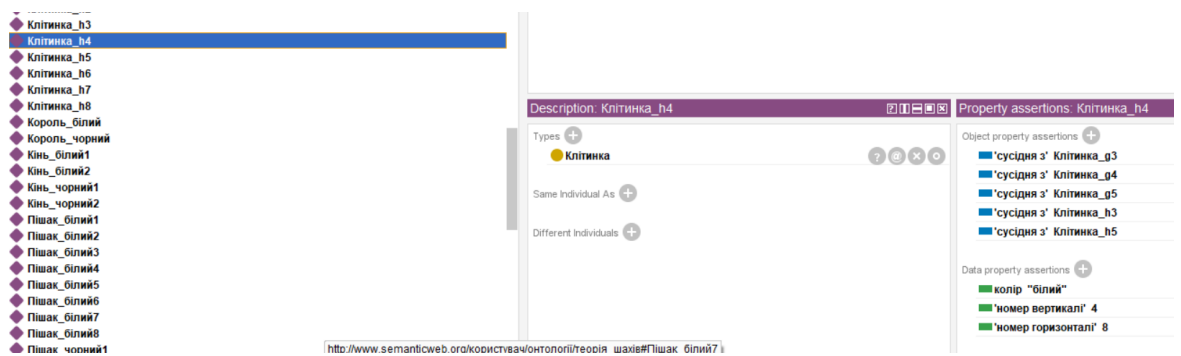


Рисунок В.10 – Створення індивіда "Клітинка h4"

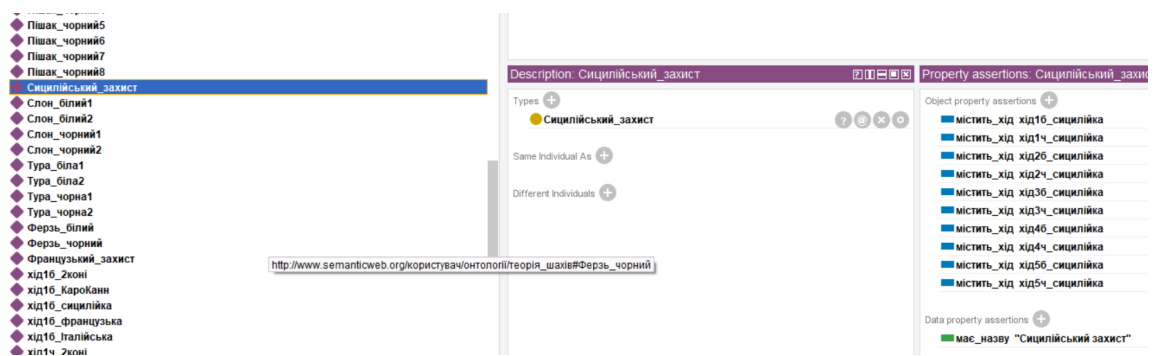


Рисунок В.11 – Створення індивіда "Сицилійський захист"



Рисунок В.12 – Створення індивіда "Ферзь білий"

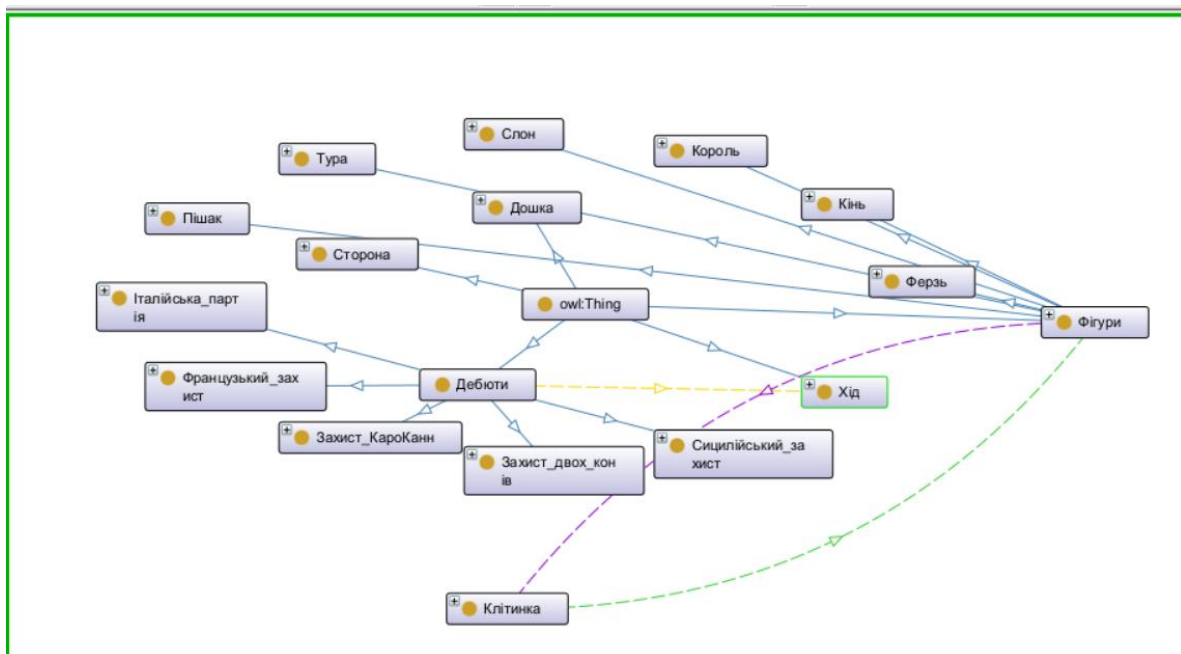


Рисунок В.13 – Граф ієрархії класів онтології з теорії шахів

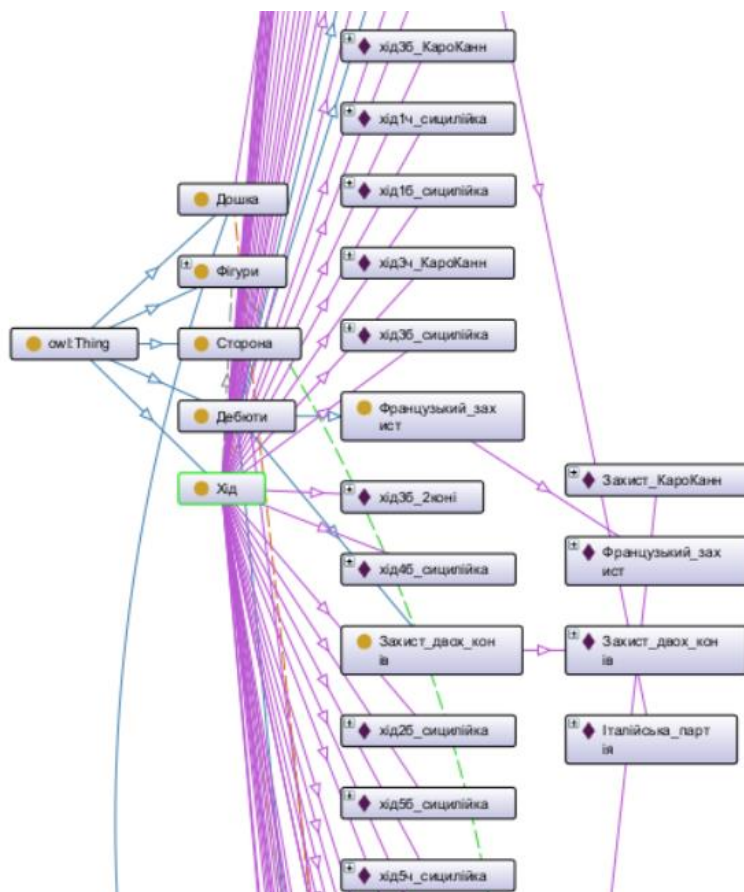


Рисунок В.14 – Фрагмент графу індивідів онтології з теорії шахів

```

SPARQL query
SPARQL query
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }

```

Рисунок В.15 – Створений SPARQL запит для виведення ієрархії

subject	
Захист_двох_конів	Дебюти
Король	Фігури
Французький_захист	Дебюти
Клітинка	Дошка
Пішак	Фігури
Захист_КароКанн	Дебюти
Слон	Фігури
Сицилійський_захист	Дебюти
Кінь	Фігури
Ферзь	Фігури
Тура	Фігури
Італійська_партія	Дебюти
Клітинка	● 'знаходиться на' max 1 Фігури

Рисунок В.16 – Результат виконання SPARQL запиту для виведення ієрархії

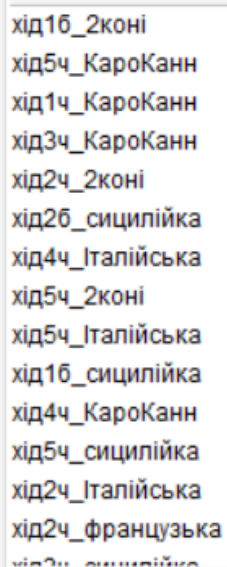
```

SPARQL query:
PREFIX : <http://www.semanticweb.org/користувач/онтології/теорія_шахів#>

SELECT ?move
WHERE {
  ?move a :Хід .
}

```

Рисунок В.17 – Створений SPARQL запит для виведення усіх індивідів класу хід

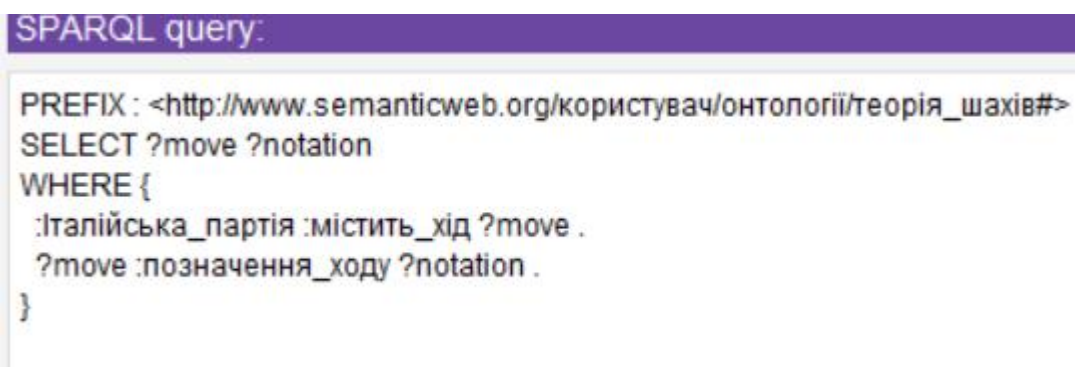


```

хід16_2коні
хід5ч_КароКанн
хід1ч_КароКанн
хід3ч_КароКанн
хід2ч_2коні
хід2б_сицилійка
хід4ч_Італійська
хід5ч_2коні
хід5ч_Італійська
хід1б_сицилійка
хід4ч_КароКанн
хід5ч_сицилійка
хід2ч_Італійська
хід2ч_французька
хід3ч_сицилійка

```

Рисунок В.18 – Результат виконання SPARQL запиту для виведення усіх індивідів класу хід



```

SPARQL query:
PREFIX : <http://www.semanticweb.org/користувач/онтології/теорія_шахів#>
SELECT ?move ?notation
WHERE {
  :Італійська_партія :містить_хід ?move .
  ?move :позначення_ходу ?notation .
}

```

Рисунок В.19 – Створений SPARQL запит для виведення усіх ходів, що належать до італійської партії

move	
хід1б_Італійська	"e4"
хід1ч_Італійська	"e5"
хід3б_Італійська	"Cc4"
хід5ч_Італійська	"Kf6"
хід4б_Італійська	"d3"
хід4ч_Італійська	"d6"
хід3ч_Італійська	"Cc5"
хід5б_Італійська	"Kc3"
хід2ч_Італійська	"Kc6"
хід2б_Італійська	"Kf3"

Рисунок В.20 – Результат виконання SPARQL запиту для виведення усіх ходів, що належать до італійської партії

ДОДАТОК Г (довідниковий)

Інструкція користувача

Щоб відкрити шахову базу знань в Protégé, спочатку потрібно запустити сам застосунок — клацніть на значок програми на робочому столі або знайдіть її через меню «Пуск». Коли відкриється вікно вітання, оберіть варіант «Open existing ontology», щоб розпочати роботу з уже створеним файлом. Далі відкриється вікно провідника, де потрібно знайти ваш файл онтології з розширенням .owl — наприклад, якщо файл збережений під назвою teoria_shahiv.owl, просто знайдіть його, виділіть і натисніть «Open». Через кілька секунд завантаження ви побачите інтерфейс Protégé — зліва буде ієрархія класів, а праворуч — вкладки для властивостей, індивідів, анотацій та іншого. Тут можна не лише переглянути всю структуру онтології, а й вносити зміни, додавати нові сутності або редагувати існуючі. Якщо ж ви редагували онтологію вручну — наприклад, у текстовому редакторі — переконайтеся, що синтаксис RDF/XML правильний, бо інакше Protégé може не зчитати її.

Коли онтологія вже відкрита у Protégé, перейдемо до створення запитів SPARQL. Щоб це зробити, потрібно перейти у верхньому меню до вкладки "Tabs", оберіть пункт "Query" — саме він відкриває вікно для побудови запитів. Якщо ця вкладка ще не активна у вашому інтерфейсі, додайте її через "Window" > "Tabs" > "SPARQL Query" — поставте галочку, і вона з'явиться серед інших.

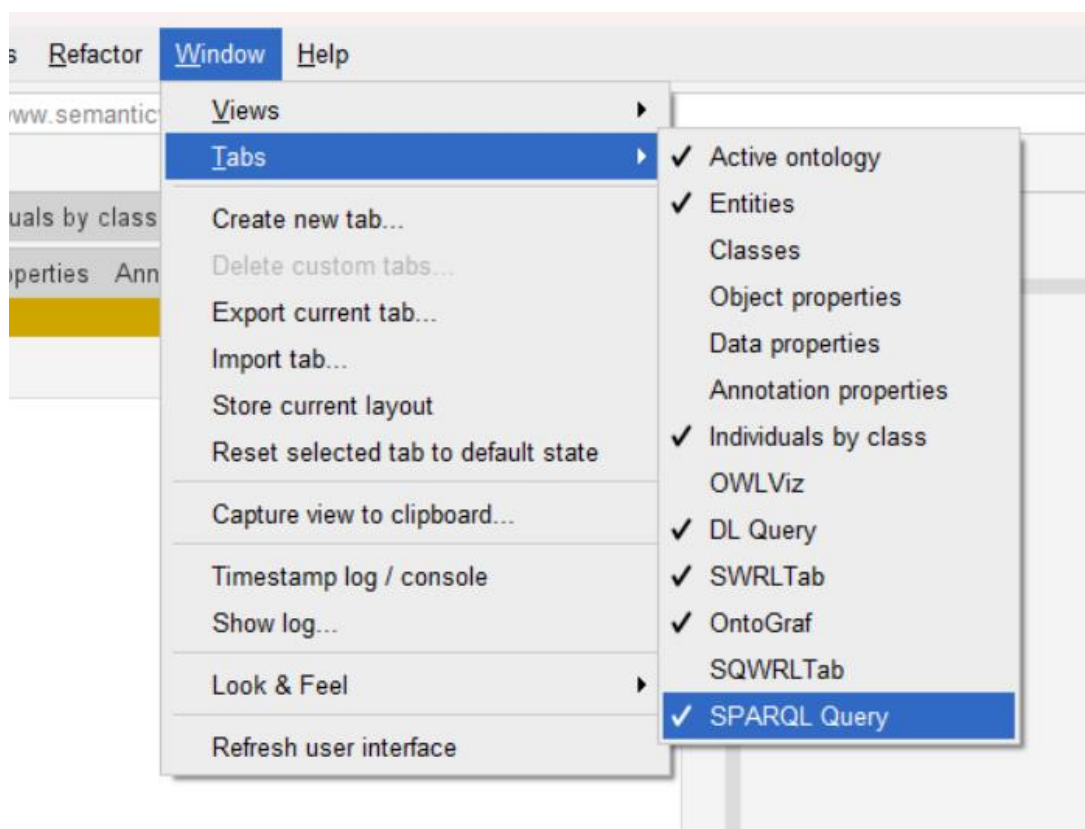


Рисунок Г.1 – Активація вкладки SPARQL-запиту

Коли вкладку відкрито, перед вами відкриється біле поле, схоже на аркуш, де можна писати запити, а під ним — кнопка запуску. Приклад запиту: `SELECT ?x WHERE { ?x a :Хід }`, де `:Хід` — ваш клас, і натискаєте кнопку "Run".

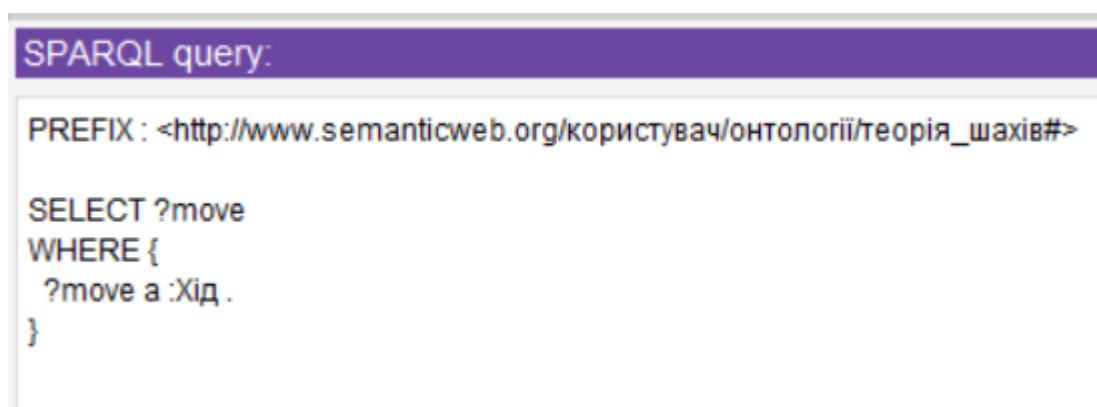


Рисунок Г.2 – Створений SPARQL запит для виведення усіх індивідів класу хід

Результатом буде виведений список всіх індивідів класу Хід, що були

описані у шаховій онтології. Можна шукати клітинки, фігури, навіть хід із певним номером, змінюючи запит — усе залежить від вашої фантазії та структури бази знань.

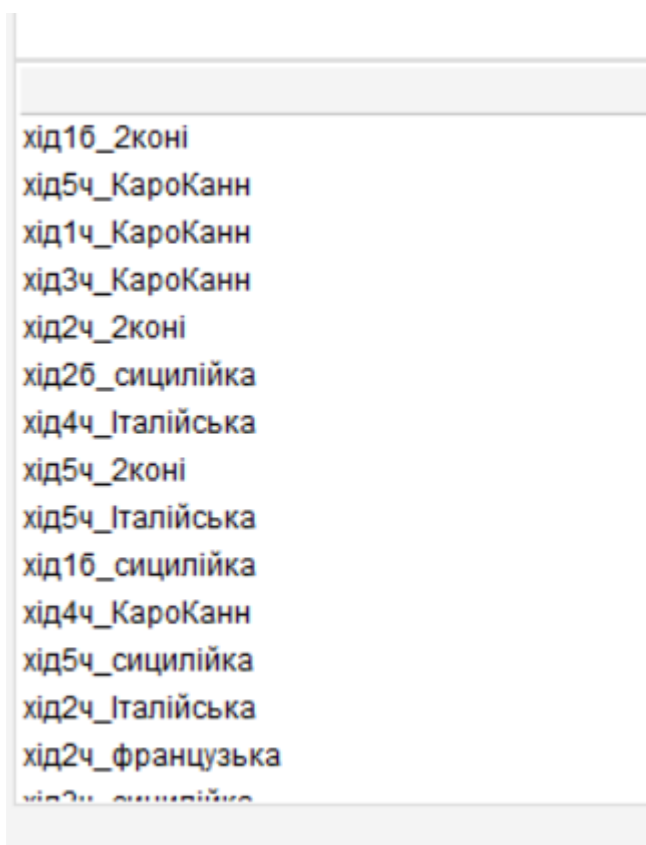


Рисунок Г.3 – Результат виконання SPARQL запиту для виведення усіх індивідів класу хід

Передусім Protégé має знати, які префікси ви використовуєте — тому за потреби додайте на початку запиту PREFIX: `<http://www.semanticweb.org/користувач/онтології/теорія_шахів#>`, і тоді всі імена типу :Хід, :Кінь_чорний1 та інші будуть правильно розпізнані.

DODATOK D (довідниковий) Довідка про впровадження



МАЛЕ НАУКОВО-ВИРОБНИЧЕ ПІДПРИЄМСТВО "ТОВ "ІТІ"
Україна, 21021, м. Вінниця, вул. Келецька, 56

№ 38 від "6" 06 2025р.

ДОВІДКА

Дана студентці групи ЗКН-216 Степанюк Карині Миколаївні в тому що результати, одержані нею в процесі виконання бакалаврської дипломної роботи «Розробка онтологічної бази знань із теорії шахів», а саме: онтологічна база знань із теорії шахів, планується використати в розробках ТОВ «ІТІ».

Зам. Директора ТОВ «ІТІ» Бодяк В.Н.



ТОВ "ІТІ"