

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра мережних технологій факультету інформатики



**Реалізація бази знань за допомогою системи PROTEGE**

Текстова частина до курсової роботи  
за спеціальністю „Комп’ютерні науки” 122

Керівник курсової роботи  
к.н., доц. Жежерун О.П.

\_\_\_\_\_

(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

Виконав студент КН-3

Ємельянов О.А

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

Київ 2020

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри мультимедійних систем,

Доцент., к. ф.-м. н. О.П. Жежерун

\_\_\_\_\_ (підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

на курсову роботу

студенту Ємельянову Олександрю Анатолійовичу факультету інформатики 3-го курсу

ТЕМА: Розробка бази знань за допомогою системи Protégé

Зміст ТЧ до курсової роботи:

1. Індивідуальне завдання
2. Календарний план
3. Анотація
4. Вступ
5. Поняття семантичної павутини
6. Онтології
7. Властивості в OWL
8. Висновки
9. Список використаної джерел

Дата видачі “ \_\_\_\_ ” \_\_\_\_\_ 2020 р. Керівник \_\_\_\_\_

(підпис)

Завдання отримав \_\_\_\_\_

(підпис)

### Календарний план виконання роботи

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи	15.10.2019	
2.	Огляд мови OWL	30.10.2019	
3.	Огляд редактора Protégé	15.02.2020	
4.	Написання перших двох розділів	20.02.2020	
5.	Створення бази знань	20.04.2020	
6.	Написання останнього розділу	30.04.2020	
7.	Створення презентації та написання доповіді для захисту роботи.	02.05.2020	
8.	Корегування роботи згідно із зауваженнями керівника	08.05.2020	
9.	Захист курсової роботи	29.05.2020	

## Зміст

Календарний план виконання роботи .....	3
Анотація .....	5
Вступ.....	6
1. Поняття семантичної павутини .....	8
1.1.Семантична павутина .....	8
1.2.Рівні семантичної павутини .....	9
1.2.1. Уніфікований ідентифікатор ресурсу (URI) .....	9
1.2.2. Мова розмітки XML .....	9
1.2.3. Середовище опису ресурсів(RDF) .....	10
1.2.4. Мова OWL(Web Ontology Language).....	10
2. Онтології .....	11
2.1.Опис онтології та її складових частин .....	11
2.2.Порівняльна характеристика бази даних та бази знань .....	12
2.3.Опис редактора Protégé .....	13
2.4.Створення онтологій.....	14
2.5.Створення класів(понять).....	15
2.5.1. Непересічні та іменовані класи .....	17
2.5.2. Класи перелічення .....	19
3. Властивості в OWL .....	20
3.1.Види властивостей .....	20
3.2.Характеристика властивостей .....	22
3.2.1. Обмеження на властивості.....	23
3.3.Використання системи логічного виводу.....	24
3.4.Створення екземплярів.....	25
3.5.Запити SPARQL .....	26
Висновок.....	28
Список використаних джерел .....	29

## Анотація

В роботі опрацьовано та досліджено можливості системи Protégé, яка розроблена для побудови та редагування онтологій. Також розкрито плюси та мінуси використання баз знань та баз даних.

У курсовій роботі охарактеризовано мову OWL, бази знань, онтології та редактор онтологій Protégé. Показано як потрібно створювати онтології в ньому.

База знань створювалась за допомогою програмного забезпечення Protégé(5.5.0)

**Ключові слова:** Protégé, онтологія, OWL, база знань, клас, властивість, екземпляр, редактор онтологій, XML.

## Вступ

**Актуальність теми:** З розвитком інформаційних технологій стрімко зростає потреба в структуризації даних, які знаходяться в мережі Інтернет, так як раніше ці дані були розміщені хаотично і могли сприйматись тільки людьми. У 1998 науковець Тім Бернерс-Лі запропонував ідею семантичної павутини, вона полягала у трансформації всіх неструктурованих даних, які розміщені в мережні інтернет в спільну базу знань, яка буде зручною для користування як людям так і електронно-обчислювальним машинам. Принцип ґрунтується на тому, щоб інформація в інтернеті відображалась у вигляді мови, яка буде зрозумілою для комп'ютера. OWL та декотрі інші мови для створення онтологій стали найкращим підходом для вирішення проблеми розміщення даних в мережі інтернет.

В наш час є велика кількість програмного забезпечення, яке призначене для розробки та редагування баз знань. Однією з таких програм є редактор онтологій з відкритим кодом Protégé. Це програмне забезпечення і є предметом дослідження у цій роботі.

Protégé підтримує мову описання онтологій OWL. Вона є стандартом Консорціуму World Wide Web та одним з головних вирішень проблеми розміщення даних в мережні інтернет. Також OWL є одним з найкращих засобів розробки баз знань, який надає великі можливості для опису та редагування онтологій.

**Мета і завдання дослідження:** вивчити та описати головні можливості, які надає редактор онтологій Protégé для створення та редагування баз знань, а також охарактеризувати основні переваги використання баз знань у порівнянні з базами даних.

Успішне виконання роботи передбачає виконання наступних задач:

- огляд основних переваг та можливостей принципу семантичної павутини;

- аналіз основних технологічних рішень для впровадження принципу семантичної павутини;
- характеристика способів задання знань через онтології;
- порівняння використання бази знань з базою даних;
- аналіз головних функцій системи розробки онтологій Protégé(5.5.0);
- опис реалізації онтологій за допомогою системи Protégé та мовою OWL.

**Об’єкт дослідження** – безкоштовне програмне забезпечення з відкритим кодом Protégé для створення баз знань.

**Предметом дослідження** є функції, які надає Protégé у процесі реалізації онтологій.

**Логіка дослідження** зумовила таку структуру роботи: анотація, вступ, 3 розділи, висновки, список використаної літератури та практична частина.

## **1. Поняття семантичної павутини**

Основні проблеми мережі Інтернет – це переповненість і надлишковість інформацією та формат подання цієї інформації, який орієнтований на людей, а не на програмне забезпечення.

Подальший розвиток Інтернету пов'язують із принципом семантичної павутини, яку започаткував Тім Бернерс-Лі, науковець з Великої Британії та голова Консорціуму W3.

Основна ідея семантичної павутини полягає в тому, щоб представляти інформацію в мережі не тільки для візуального сприйняття, а й для того, щоб програми могли легко та ефективно взаємодіяти з цією інформацією. Семантична павутина – це так зване розширення мережі Інтернет, де інформація відображається за чіткими правилами, що дозволяє і людям, і програмами з легкістю її використовувати. Також семантичний веб передбачає об'єднання інформації в одну структуру, де кожній одиниці даних буде відповідати тег.

### **1.1.Семантична павутина**

Семантична павутина складається з двох частин. Перша частина відповідає за представлення даних. Дані відображаються двома головними мовами: RDF(Resource Description Framework) та XML (Extensible Markup Language). Друга частина відповідає за теоретичні відомості про онтології(моделі предметних областей).

У 2004 році Консорціум Всесвітнього павутиння затвердив OWL як технологію специфікації мови мережеских онтологій[1]. Отже, відтоді семантичний веб використовує 3 основні мови:

- RDF – мова, яка призначена для забезпечення сумісності метаданих за допомогою спільного синтаксису та семантики[2]
- XML – мова, яка визначає синтаксис та структуру елемента[3]
- OWL – мова написання онтологій для семантичного вебу[4]



## 1.2.Рівні семантичної павутини

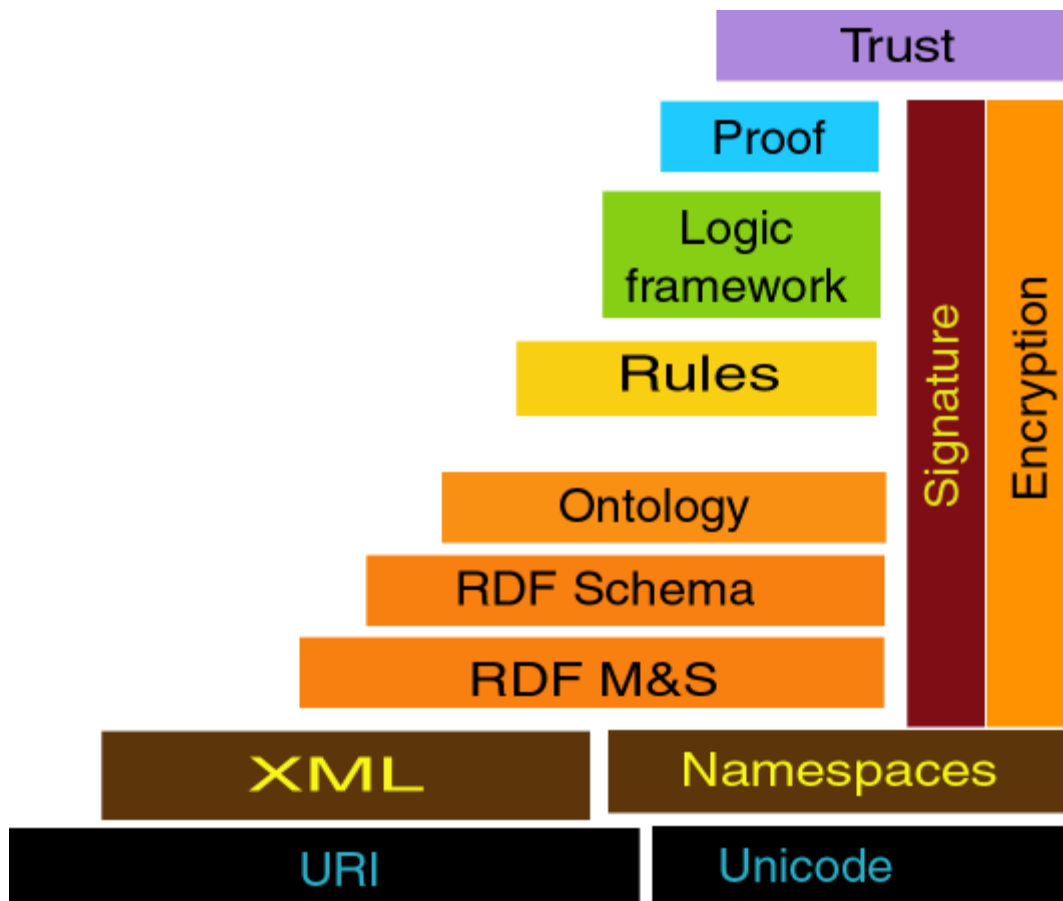


Рис. 1

### 1.2.1. Уніфікований ідентифікатор ресурсу (URI)

URI – рядок символів, котрий однозначно ідентифікує будь-який об’єкт в мережі Інтернет. Може посилатись або на віртуальний, або на реальний ресурс. Наприклад, картинка, документ, діаграма, адреса електронної пошти. Уніфікований ідентифікатор ресурсів являє собою спосіб ідентифікації ресурсів у Всесвітній павутині. Один з найвідоміших URI є URL.

### 1.2.2. Мова розмітки XML

Як було вказано вище, XML – це мова, яка визначає синтаксис елемента. XML та засоби, які описують XML-документи та поведінку XML-документів – це DTD, XML Schema. XML Schema визначає структуру XML-документів та дозволяє використовувати конкретні типи даних. RDF Schema – дозволяє передавати властивості RDF-ресурсів, а також семантику відношень між ними.

XML був розроблений з метою спрощення розробки та використання документів машинами та розуміння людиною. XML є розширюваною мовою через те, що не обмежує розробника, він може створювати розмітку відповідно до необхідної предметної області і керуватись тільки обмеженням семантикою мови.

### **1.2.3. Середовище опису ресурсів(RDF)**

RDF – дозволяє описувати абстрактні моделі даних деяких об'єктів і відношень між ними також описує структуру метаданих. Використовує семантику, яка базується на XML синтаксисі. RDF Schema(RDFS) дає можливість описувати властивості та класи RDF-ресурсів та семантику відношень між ними. Середовище опису ресурсів було розроблено для опису або моделювання інформації, яка представляється як ресурс в мережні Інтернет.

### **1.2.4. Мова OWL(Web Ontology Language)**

Мова OWL розширює можливості вищеперерахованих технологій, дозволяє описувати відношення між класами. Мова базується на представленні відношення «об'єкт – властивість». За допомогою OWL можна описувати не тільки сторінки в мережі Інтернет, а й будь-які інші об'єкти. Кожному елементу в OWL ставиться у відповідність URI. Мови OWL описуються формальною семантикою. Виділяють три різновиди мови OWL:

- OWL Full – надає користувачам максимальну виразність та синтаксичну свободу, але не гарантує, що запит буде виконаний. Не відповідає жодній логіці, так як не є дозволеним
- OWL Lite – підмножина повної специфікації, котра надає мінімально-достатні інструменти для створення онтологій. Цей різновид мови гарантує, що складність алгоритму належить класу P. OWL Lite заснований на логіці SHLF(D)
- OWL DL – це так зване поєднання OWL Full та OWL Lite, адже з одного боку надає максимальну виразність та гарантує повноту обчислень, проте

з іншого боку має строгі обмеження(на зв'язках класів, часу виконання запитів, тощо). OWL DL заснований на логіці SHOLN(D).

## 2. Онтології

### 2.1.Опис онтології та її складових частин

Онтологія – структура, яка описує значення елементів, деякої предметної області. Вона дає можливість описати будь-яку предметну область в зрозумілих для комп'ютера термінах. Складається з термінів та правил, які описують зв'язки та відношення між ними.

Онтології складаються із екземплярів, класів(поняття), атрибутів та відношень.

Екземпляр – одиниця найнижчого рівня. Головна мета онтології – класифікація екземплярів. Приклади екземплярів: марки автомобілів, назви країн, міста, тощо.

Класи(поняття) – абстрактні колекції або набори об'єктів. Класи можуть включати в себе екземпляри або інші класи. Приклади класів: поняття «собака» вкладене в поняття «тварини». Клас «Ауді» вкладене в клас «автомобіль»

Атрибути. Об'єкти в онтології мають атрибути. В них є ім'я та значення, використовуються атрибути для зберігання даних. Приклад атрибуту: об'єкт *KIA Carnival* має атрибути:

- Марка: KIA
- Модель: Carnival
- Кількість-місць: 7
- Коробка-перемикання-передач: {механічна, 5-ступна}
- Двигун: 2900 см<sup>3</sup>
- Рік випуску: 1999
- Колір: чорний

Атрибут також може складатись з декількох типів даних. В прикладі атрибут з назвою «Коробка-перемикач-передач» складається з двох простих типів даних: «механічна» та «5-ступна».

## **2.2.Порівняльна характеристика бази даних та бази знань**

Спочатку дамо визначення, що ж таке база даних та база знань. Отже, база даних - це множина взаємопов'язаних даних, об'єднаних спільним середовищем зберігання, спільним застосуванням, однаковою формою представлення, єдиними методами і засобами керування[6].

База знань – частина експертної системи, що містить подання знань, яке стосується визначеної предметної області. У статичній частині бази знань зберігаються довгострокові знання, що описують розглянуту предметну область у вигляді загальних фактів(фраз без умов, що містять твердження, які завжди є абсолютно вірними) і правил(тверджень, істинність яких залежить від деяких умов, що утворюють тіло правила), які описують доцільні перетворення фактів цієї області з метою породження нових фактів або гіпотез[7].

База знань має перевагу над базами даних, адже бази даних перш за все розроблені для зберігання даних, їх структурування та ефективного пошуку. Тоді як база знань дозволяє не тільки зберігати та обробляти дані, а й додавати нові дані на основі вже існуючих, використовуючи систему логічного виводу.

### 2.3.Опис редактора Protégé

Protégé – це безкоштовне програмне забезпечення для створення та редагування онтологій та побудови інтелектуальних систем з відкритим кодом[8]. Розроблений Стенфордським центром досліджень біомедичної інформатики та підтримується великою спільнотою розробників, корпоративних та державних користувачів, котрі використовують Protégé для розв’язання проблем в різних галузях. Редактор надає користувачу графічний інтерфейс для розробки та редагування онтологій. Розроблений на мові програмування Java. Архітектура Protégé може адаптуватись для розробки різних програм, які будуть базуватись на онтологіях. Остання версія Protégé – 5.5.0.

Система Protégé має два способи створення онтологій. Один з них – Protégé-OWL, а інший – Protégé-Frames. Онтології, які були побудовані за допомогою Protégé можуть експортуватись в різні формати: OWL, XML Schema та RDF Schema. Редактор Protégé-OWL – розширення, яке підтримує OWL, допомагає створювати та редагувати онтології для семантичної павутини. Редактор Protégé-Frames – надає користувачу інтерфейс та базу знань для створення та збереження онтологій, які основані на фреймах.

## 2.4. Створення онтологій

Після відкриття редактора вказуємо назву онтології:

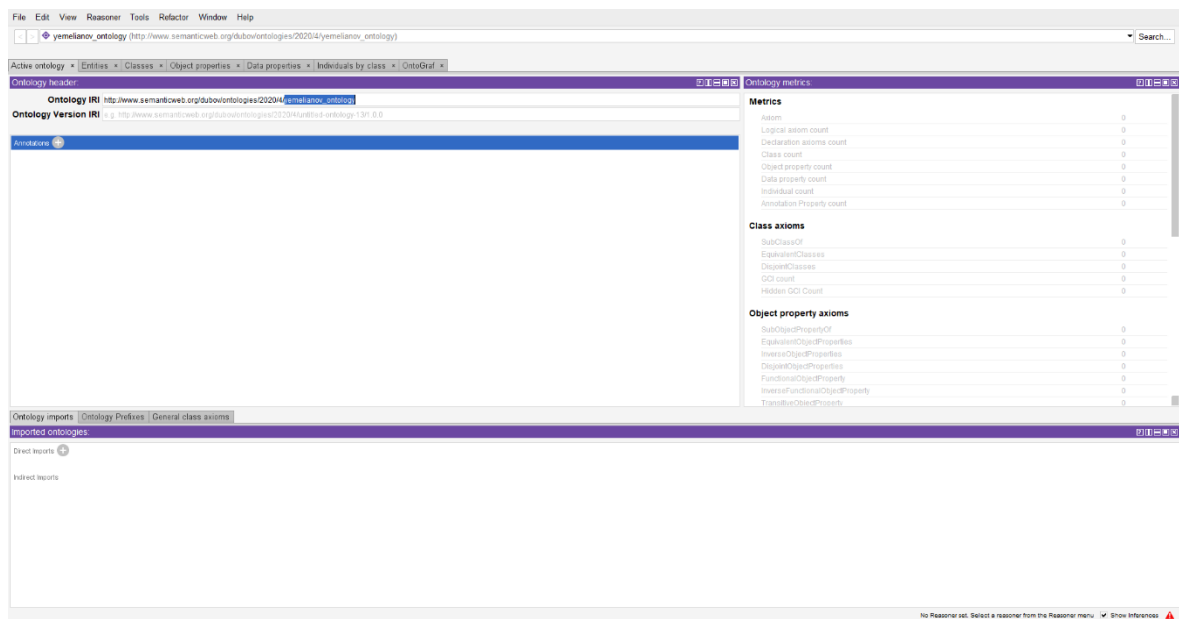


Рис. 2

За бажанням можемо додати опис до онтології, для кращого розуміння для чого вона створена. Для цього потрібно натиснути на значок «плюс» біля рядка «Annotation».

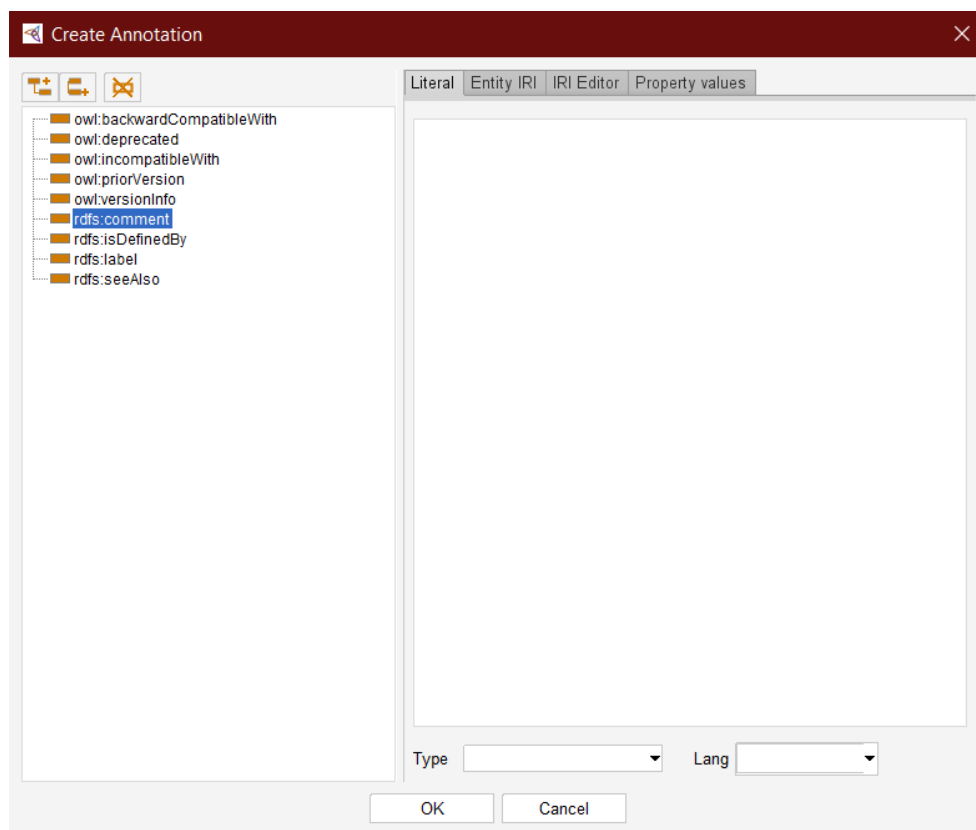


Рис. 3

У вікно, яке відкрилось вводимо текст, опис онтології.

## 2.5. Створення класів(понять)

Класи в Protégé інтерпретуються як множини, а елементи цих класів – екземпляри. Їх побудова базується на використанні формальних конструкцій, які описують вимоги для екземплярів класу. Protégé надає можливість створення ієрархії класів, яка має вигляд «Суперклас – Підклас».

Для створення класів в Protégé потрібно перейти на вкладку «Classes». За замовчуванням пуста онтологія має клас Thing – це клас, який є суперкласом і всі об’єкти, які будуть створені входять в нього.

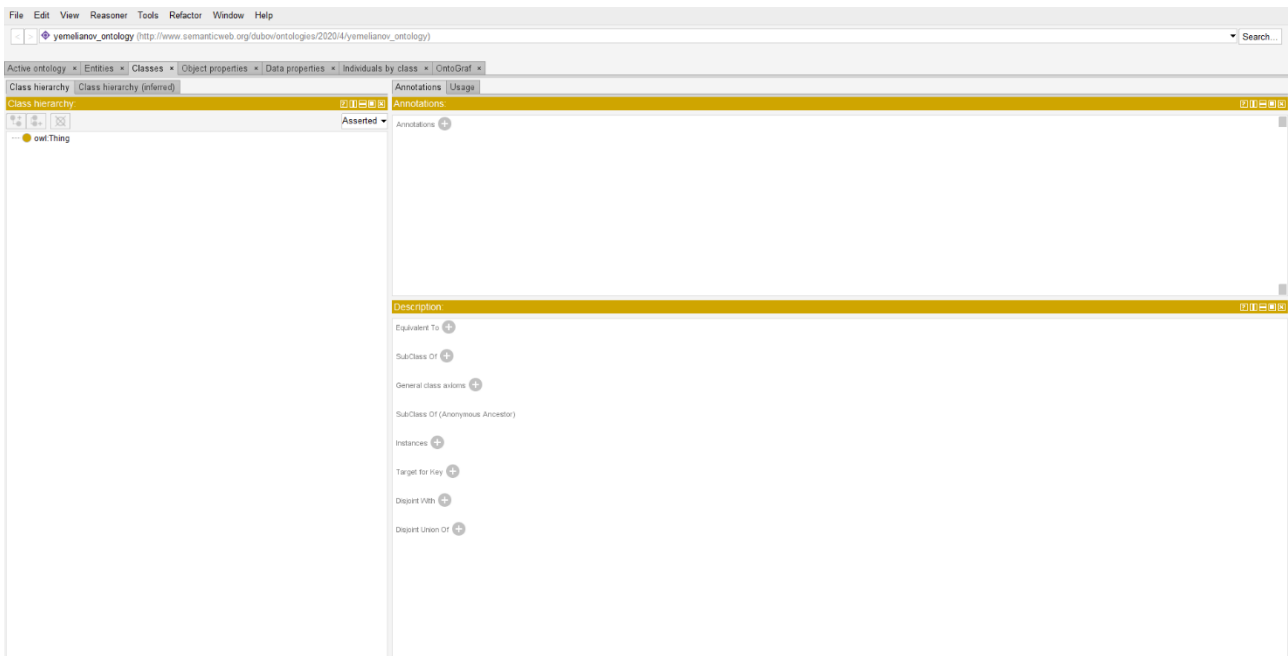


Рис. 4

Натискаємо «Add subclass» та вводимо назву цього класу. Створюється новий клас, який є підкласом Thing

Обмеження на назви класів:

- не можна починати назву класу з цифри, наприклад: «1class», замість цього Protégé відобразить «class». Краще написати class1 або class\_1;
- не можна використовувати апостроф, Protégé замість цілого слова, наприклад «м'яч», залишить тільки частину слова після апострофа – яч;

Також можна створити підкласи 2, 3, n-рівнів від початкового класу, який є класом першого рівня. Щоб створити підклас потрібно скористатись функцією створення ієрархії класів(інструмент «Create Class hierarchy»). У вікні, яке відкрилось потрібно вибрати основний клас. В першому полі пишемо рівні для класів, а в полі «Prefix» дає назву класам:

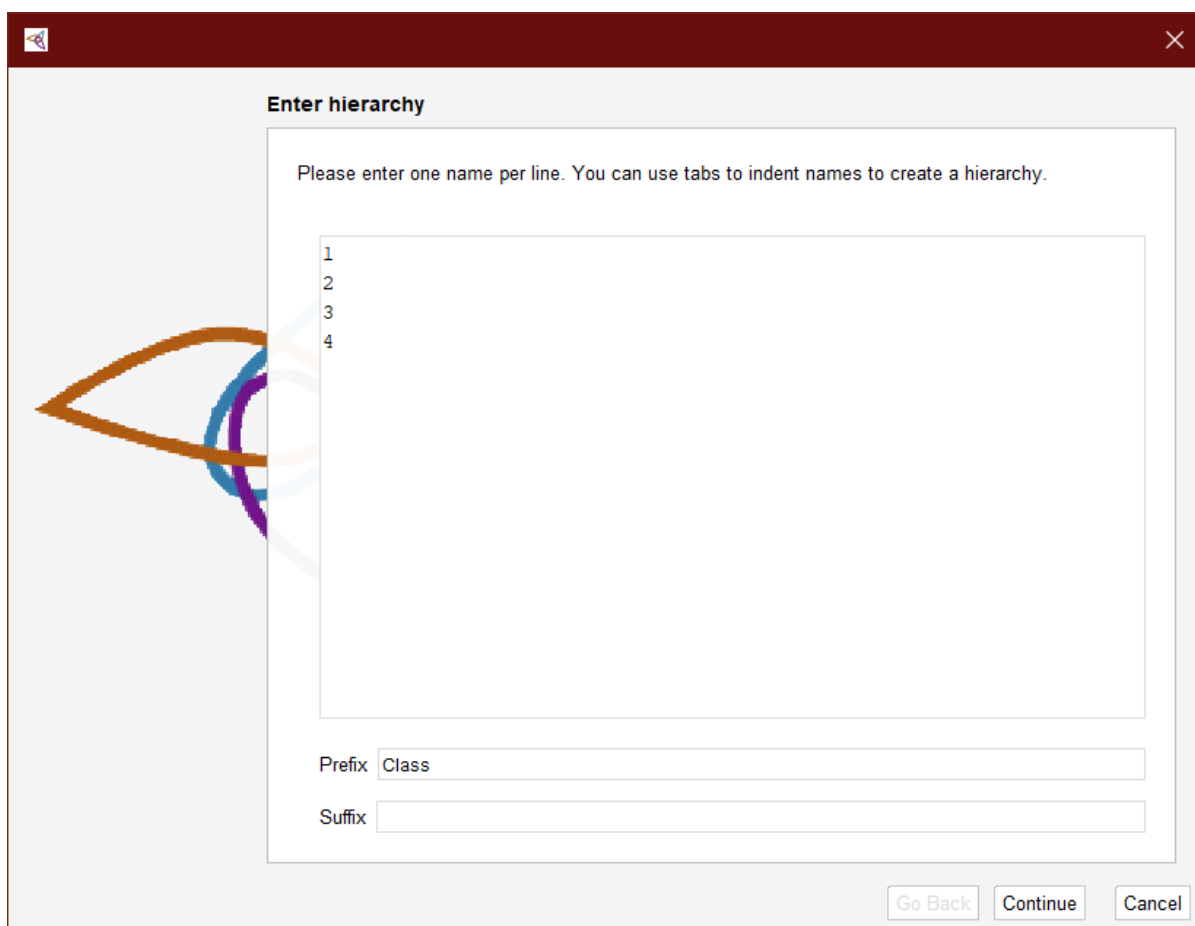


Рис. 5

В результаті ми отримуємо:

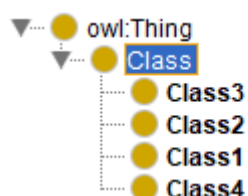


Рис. 6



Після створення всіх класів є можливість зобразити їх у вигляді графу. Для цього потрібно перейти на вкладку OntoGraf та вибрати зручний для Вас спосіб розміщення класів. В результаті отримуємо такий граф(Рис. 7):

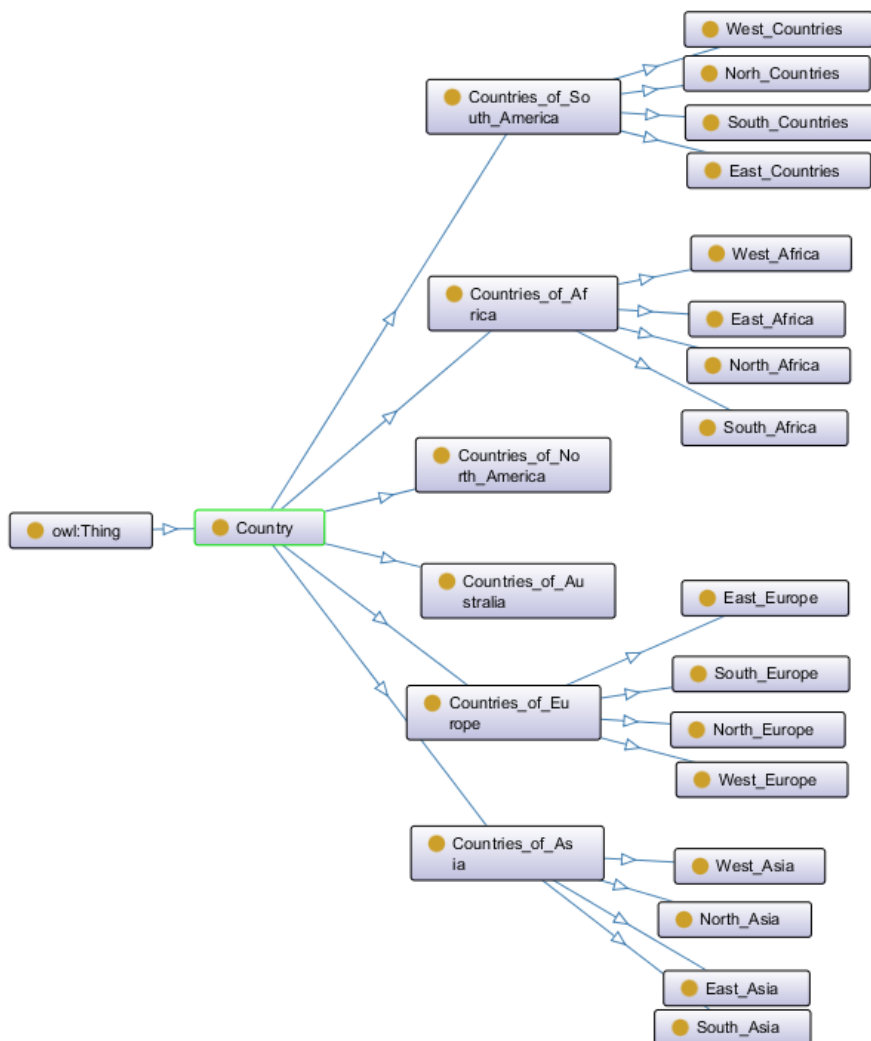


Рис. 7

### 2.5.1. Непересічні та іменовані класи

Основними структурами в розробці онтологій є класи. Іменовані класи – це звичайні класи, які створюються в Protégé. Після додавання класів до онтології можна вказати, що ці класи є непересічні, тобто будь-який об'єкт не може бути екземпляром більше ніж одного класу. Щоб зробити класи непересічними потрібно вибрати клас та натиснути «+» біля клавіші «Disjoint With»(Рис. 8)

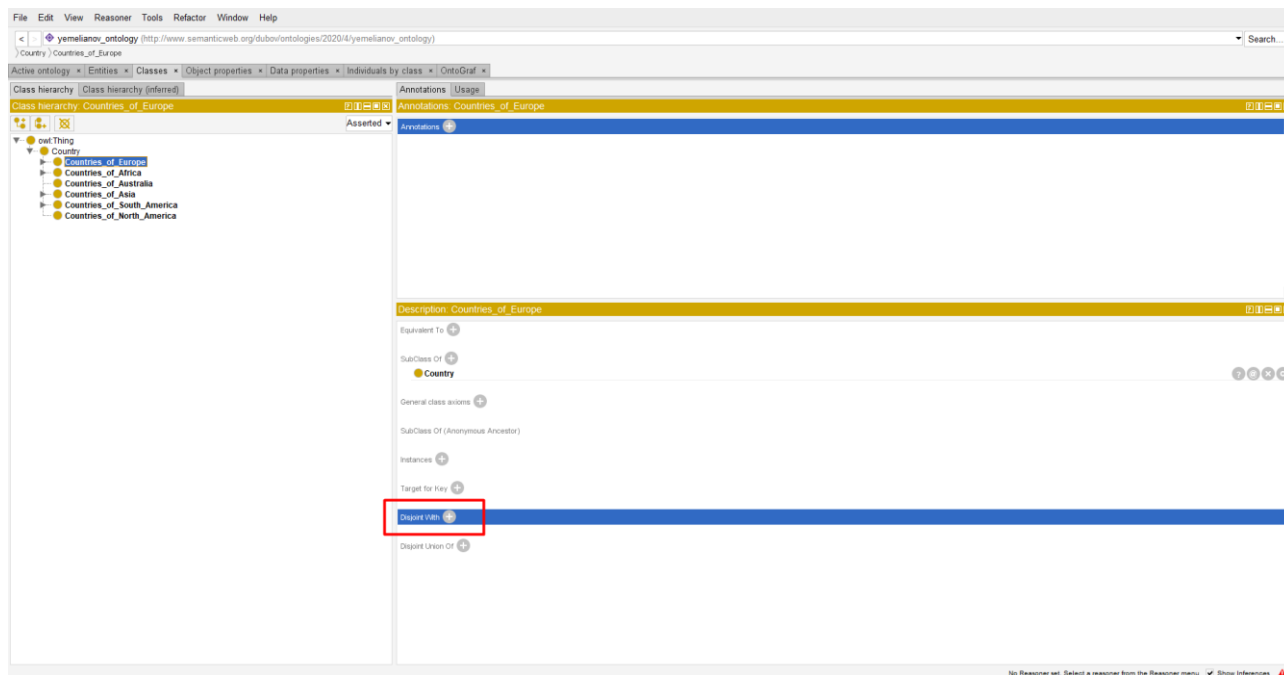


Рис. 8

У вікні, яке відкрилось, вибираємо потрібний клас та натискаємо «ОК» (Рис. 9)

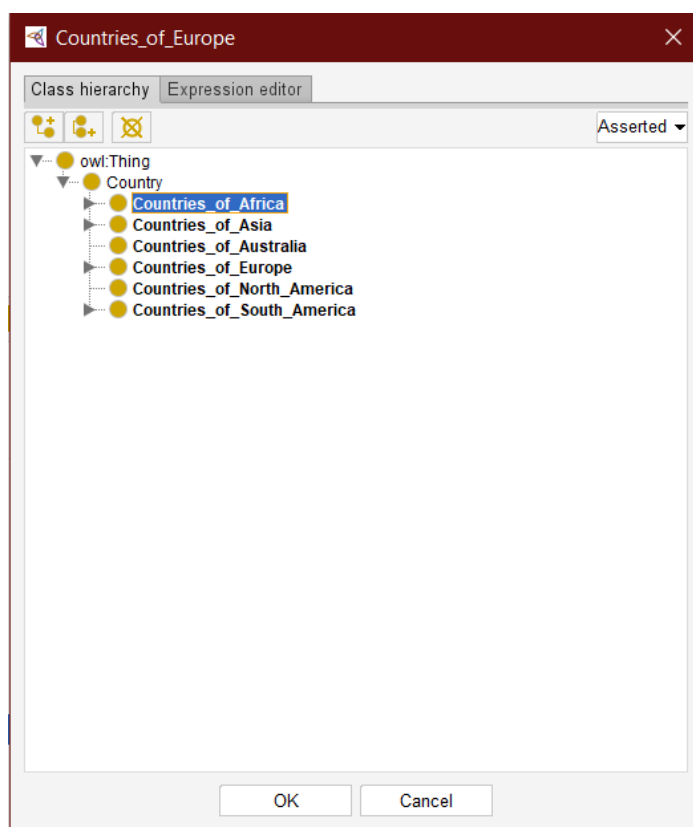


Рис. 9

В результаті отримуємо непересічні класи, які відображаються в розділі «Disjoint With» (Рис. 10)

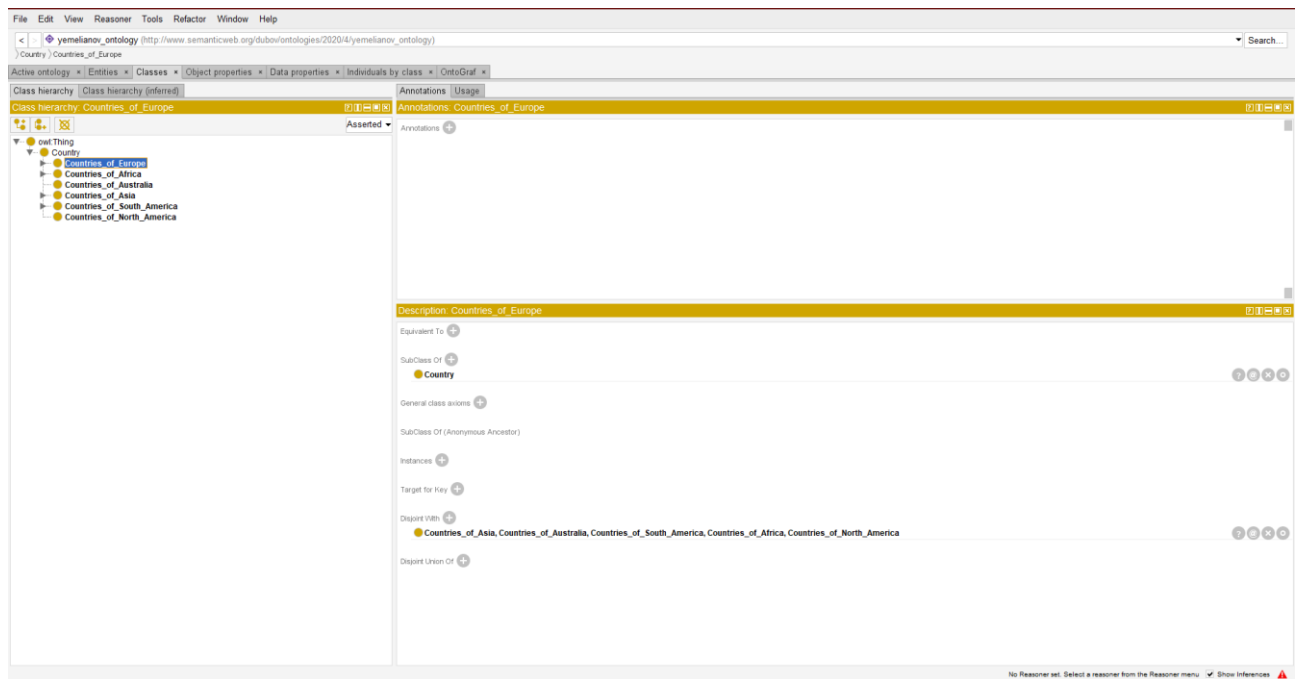


Рис. 10

### 2.5.2. Класи перелічення

Окрім іменованих класів в OWL також існують класи перелічення. Вони допомагають задати, наприклад дні тижня {Понеділок, Вівторок, Середа, Четвер, П'ятниця, Субота, Неділя}, місці, тощо. Класи перелічення задаються за допомогою редактора «class expression editor» - екземплярів, які входять в клас перелічення. Клас задається списком екземплярів через кому та пробіл, взяті в фігурні дужки.

### 3. Властивості в OWL

При створенні властивості вказують до якого класу вона належить. Також потрібно пам'ятати, що властивість може входити до кількох класів одночасно.

#### 3.1. Види властивостей

Властивості об'єктів – це відношення між двома екземплярами. В редакторі Protégé існує 2 основні типи властивостей: властивості об'єктів та властивості типів даних (Object Properties та Data Properties відповідно). За замовчуванням автоматично створюється властивість об'єктів верхнього рівня так як і вищезгаданий клас Thing. Щоб створити властивість для певного об'єкту потрібно перейти на вкладку Object Properties:

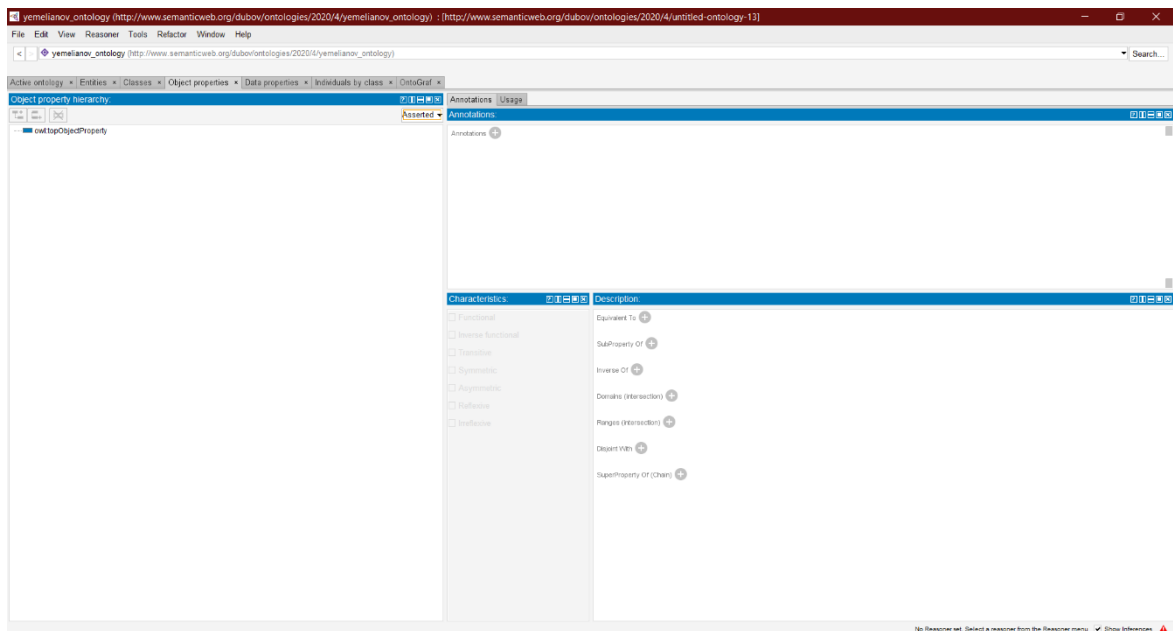
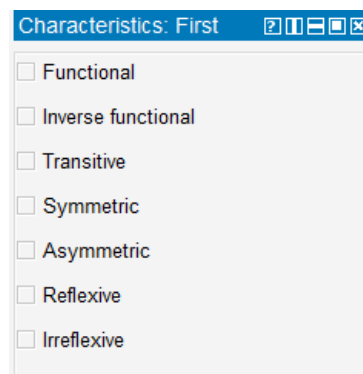


Рис. 11

Будь-яка властивість має набір характеристик:



**Функціональність.** Властивість є функціональною, якщо для її екземпляра існує не більше одного іншого екземпляра, котрий відноситься до першого екземпляра через цю ж властивість. Функціональну властивість також можна назвати однозначною властивістю.

**Зворотна функціональність.** Протилежна до функціональної є зворотна функціональна властивість, тобто для певного екземпляра може існувати багато інших екземплярів, які відносяться до першого через цю властивість.

**Транзитивність.** Властивість транзитивна коли вона зв'язує екземпляр  $x$  та екземпляр  $y$  і також  $y$  та  $z$  і з неї можна вивести зв'язок між екземпляром  $x$  та екземпляром  $z$ . Потрібно пам'ятати, що якщо властивість транзитивна, то вона не може бути функціональною.

**Симетричність.** Якщо властивість  $x$  симетрична і він зв'язує екземпляри  $a$  та  $b$ , то екземпляр  $b$  також зв'язаний з екземпляром  $a$  через властивість  $x$ . Наприклад, якщо екземпляр «Олег» зв'язаний з екземпляром «Михайло» через властивість «friend», то можна вважати, що і «Михайло» зв'язаний з «Олег» через цю властивість. В результаті можемо сказати, що властивість «friend» є симетричною.

**Асиметричність.** Якщо властивість  $x$  асиметрична і вона зв'язує два екземпляри  $a$  та  $b$ , то екземпляр  $b$  не може бути зв'язаний з екземпляром  $a$  через властивість  $x$ . Наприклад, якщо екземпляр «Микола» зв'язаний з екземпляром «Андрій» властивістю «father», то «Андрій» не може бути зв'язаним з «Микола» цією ж властивістю, проте може бути зв'язаним з кимось іншим властивістю «father».

**Рефлексивність.** Властивість може називатись рефлексивною, якщо її екземпляр можна зв'язати сам з собою через цю властивість. Наприклад, екземпляр «Микола» може бути зв'язаний сам з собою властивістю «забезпечує».

Іррефлексивність. Властивість називається іррефлексивною, якщо її екземпляр не можна зв'язати сам з собою через цю властивість. Наприклад, екземпляр «Анна» не може бути з'єднана сама з собою властивістю «Parent», тому ця властивість – іррефлексивна.

### 3.2. Характеристика властивостей

Властивості в OWL можуть мати так звані підвластивості, тобто створювати ієрархії властивостей, так як класи, про які було написано вище. Підвластивості характеризуються своїм супер-властивостями. Наприклад, властивість «hasSister»(мати сестру) характеризується більш загальною властивістю «hasSibling»(мати родича).

Кожна властивість в OWL може мати свою зворотну властивість. Наприклад, властивість «hasMother»(мати сестру) може мати зворотну властивість «hasDaughter»(мати дочку) – якщо Марія hasMother Олена, то зворотна властивість Олена hasDaughter Марія. Для задання зворотної властивості в редакторі Protégé потрібно натиснути на «+» біля клавіші «Inverse Of»(Рис. 12)

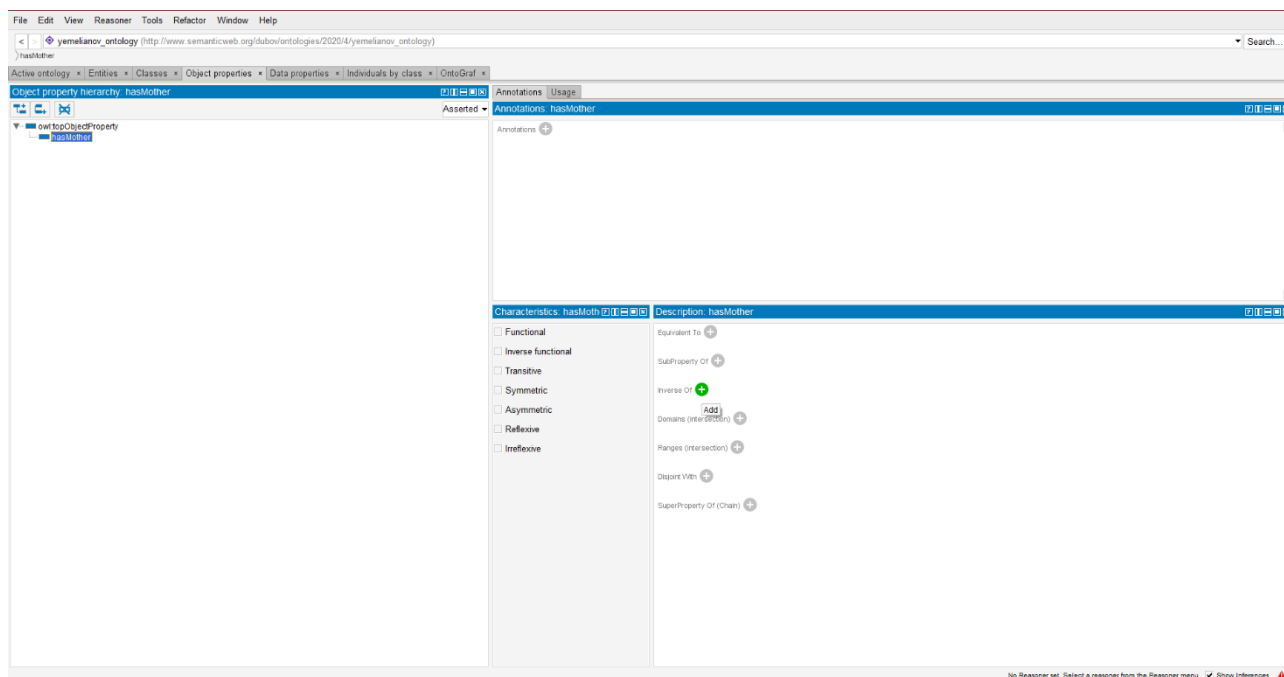


Рис. 12

Далі в діалоговому вікні вказати назву зворотної властивості та натиснути «ОК»(Рис. 13)

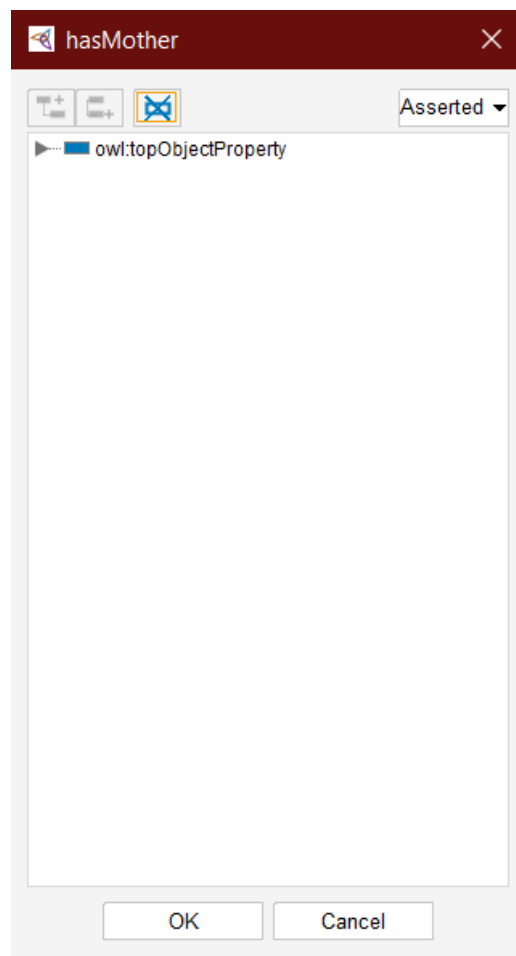


Рис. 13

### 3.2.1. Обмеження на властивості

Властивості в OWL описують бінарні відношення. Властивості типів даних описують зв'язки між екземплярами та значеннями даних. В мові OWL існують такі обмеження:

- обмеження кардинальності;

Дане обмеження властивості дозволяє вказувати кількість відношень в якому може бути задіяний певний клас

- обмеження кванторів;
- обмеження змінних.

### 3.3. Використання системи логічного виводу

Одна з головних особливостей OWL-онтологій полягає в тому, що вони можуть використовувати системи логічного виводу. Система логічного виводу допомагає тестувати ієрархії класів. Виходячи з опису, класифікатор (так ще називають систему логічного виводу) може перевірити чи має клас будь-які екземпляри. Якщо клас не має жодного екземпляру він є несумісним.

Запуску системи логічного виводу відбувається командою «start», яка знаходиться на вкладці Reasoner, вмикається командою none.

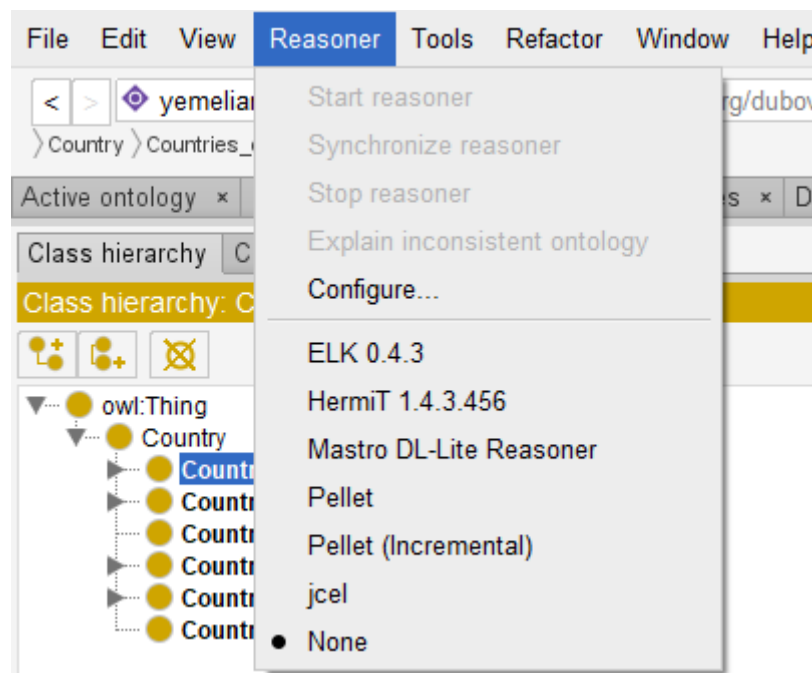


Рис. 14

Класифікатор може працювати у фоновому режимі. Результат його роботи відображається в ієрархії класів. Під час роботи системи логічного виводу для оновлення введених змін потрібно натиснути поєднання клавіш Ctrl+R (команда synchronize reasoner).



### 3.4. Створення екземплярів

Мова OWL дозволяє створювати екземпляри та задавати їх властивості. Екземпляри класів також можуть використовуватись в характеристиках класів, а саме в обмеженнях `hasValue`.

Для створення екземплярів класу потрібно перейти на вкладку «Individuals by class», вибрати клас, для якого потрібно створити екземпляр та натиснути на іконку створення екземпляру (Рис. 15)

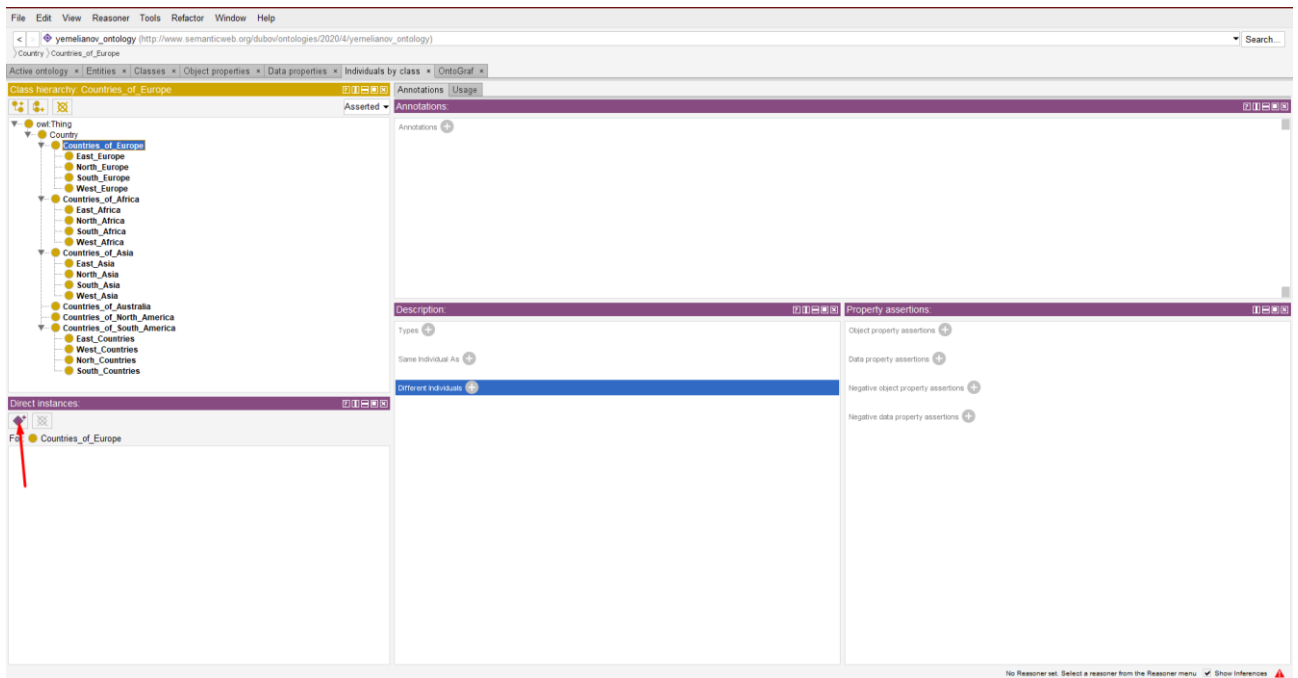


Рис. 15

Далі вказати назву нового екземпляру та натиснути «ОК» (Рис. 16)

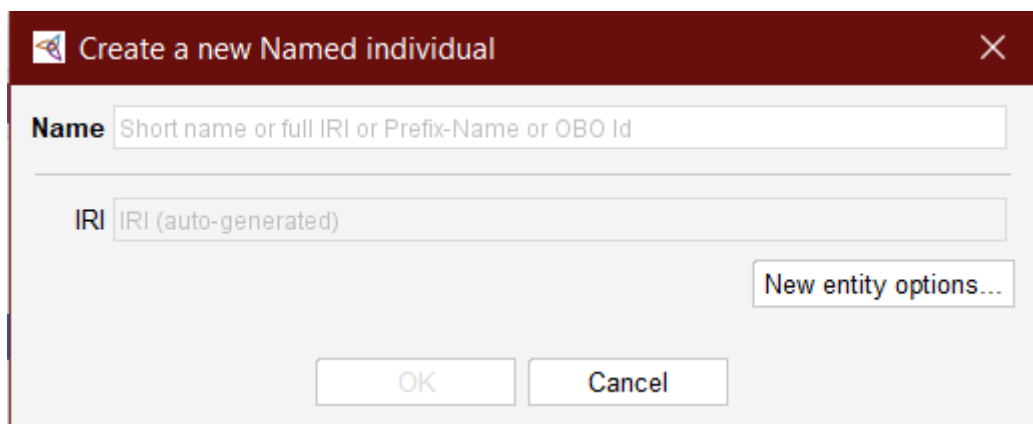


Рис. 16

### 3.5. Запити SPARQL

SPARQL – це мова запитів до даних і є однією з технологій семантичної павутини. Призначена не тільки для отримання даних, шляхом виконання запитів, а й для зміни даних. Наприклад, для видалення даних з бази знань, або для додання нових даних.

Для виконання запитів SPARQL потрібно перейти на вкладку «SPARQL Query». Якщо вкладка не відображається перейдіть в меню Window -> Tabs та виберіть «SPARQL Query» (Рис. 17)

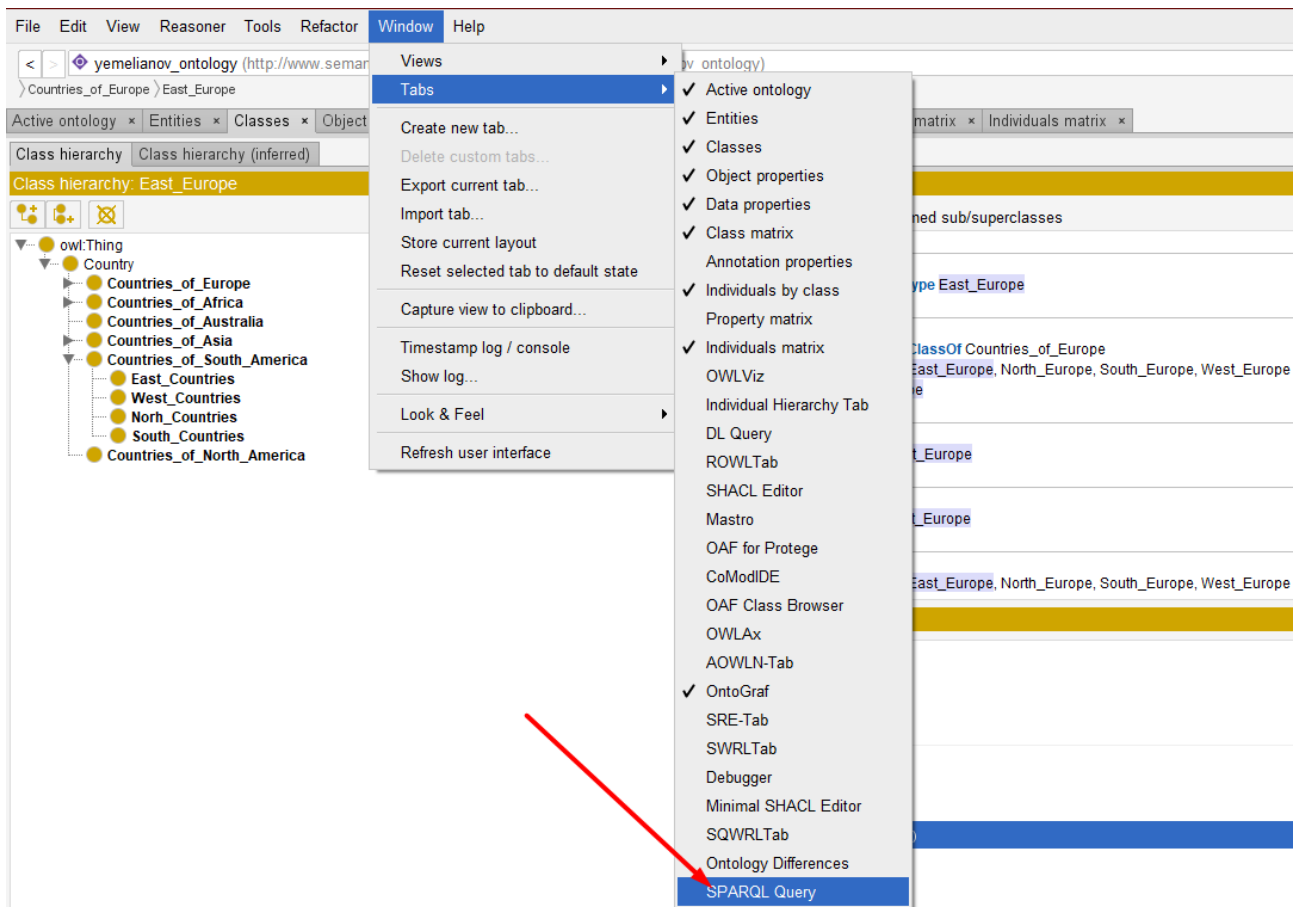


Рис. 17

За замовчуванням на цій вкладці буде відображатись:

```
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 }
```

Для виконання цього запиту потрібно натиснути «Execute» і під кодом відобразиться результат у вигляді таблиці «subject» та «object»(Рис.18).

subject	object
Countries_of_North_America	Country
Countries_of_South_America	Country
East_Europe	Countries_of_Europe
West_Africa	Countries_of_Africa
East_Countries	Countries_of_South_America
Countries_of_Africa	Country
East_Asia	Countries_of_Asia
North_Countries	Countries_of_South_America
North_Asia	Countries_of_Asia
North_Europe	Countries_of_Europe
South_Europe	Countries_of_Europe
West_Countries	Countries_of_South_America
Countries_of_Asia	Country
North_Africa	Countries_of_Africa
Countries_of_Australia	Country
South_Asia	Countries_of_Asia
South_Africa	Countries_of_Africa
South_Countries	Countries_of_South_America
West_Asia	Countries_of_Asia
Countries_of_Europe	Country

Execute

Рис. 18

## Висновок

Отже, вивчивши можливості Protégé в ході розробки бази знань, можна зі впевненістю сказати, що це дуже потужний та зручний редактор онтологій. Одна з його переваг — це можливість використання системи логічного виводу. Створення класів ієрархії також хороша перевага над іншим редакторами.

Створена база знань може бути доповнена іншими даними про країни для використання в різних цілях. Наприклад, можна додати дані про внутрішній валовий продукт країн, та слідкувати за розвитком кожної з них в цьому плані.

## Список використаних джерел

- 1 OWL Web Ontology Language. – [Електронний ресурс]. – Доступно з: <https://www.w3.org/TR/owl-guide/>
- 2 Resource Description Framework (RDF). – [Електронний ресурс]. – Доступно з: <https://www.w3.org/RDF/>
- 3 Extensible Markup Language (XML). – [Електронний ресурс]. – Доступно з: <https://www.w3.org/XML/>
- 4 Web Ontology Language (OWL). – [Електронний ресурс]. – Доступно з: <https://www.w3.org/OWL/>
- 5 8. OWL Full, OWL DL and OWL Lite. – [Електронний ресурс]. – Доступно з: <https://www.w3.org/TR/owl-ref/#OWLFull>
- 6 Організація баз даних: практичний курс: Навч. посіб. для студ. / А. Ю. Берко, О. М. Верес; Нац. ун-т «Львів. політехніка». — Л., 2003. — 9 с. — Бібліогр.: 8 назв.
- 7 Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. — Запоріжжя: ЗНТУ, 2008. — 22 с.
- 8 Офіційний сайт Protege — [Електронний ресурс]. — Доступно з: <https://protege.stanford.edu>
- 9 Глибовець М. М. Інтелектуальні мережі / М. М. Глибовець, А. М. Глибовець, М. В. Поляков. — Дніпропетровськ: ПП Видавництво «Нова ідеологія», 2014.