

МЕЛІТОПОЛЬСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ  
ДО ЛАБОРАТОРНИХ РОБІТ З КУРСУ  
«РОЗРОБКА ТА СУПРОВІД БАЗ ЗНАНЬ»

Методичні рекомендації

Мелітополь – 2015

**УДК 004.65(072)**  
**ББК 32.973-01я7**  
**М 54**

Методичні рекомендації до лабораторних робіт з курсу «Розробка та супровід баз знань» / Укл.: С.В. Шаров. – Мелітополь: РВЦ МДПУ, 2015. – 44 с.

**Рецензенти:**

**Єремєєв В.С.** – д.т.н., професор кафедри інформатики і кібернетики Мелітопольського державного педагогічного університету імені Богдана Хмельницького.

**Малкіна В.М.** – д.т.н., професор, завідувач кафедри інформаційних технологій Таврійського державного агротехнологічного університету.

Дані методичні рекомендації призначені для студентів Мелітопольського державного педагогічного університету імені Богдана Хмельницького, які навчаються за спеціальністю 8.04030201 Інформатика (галузь знань 0403 Системні науки і кібернетика).

*Рекомендовано до друку рішенням Вченої Ради Мелітопольського державного педагогічного університету імені Богдана Хмельницького.*

*Протокол № 1 від 31.08.2015.*

© Шаров С.В.

© Мелітопольський державний педагогічний університет імені Богдана Хмельницького, 2015

## ЗМІСТ

Вступ .....	4
Лабораторна робота №1 .....	6
Лабораторна робота №2 .....	10
Лабораторна робота №3 .....	13
Лабораторна робота №4 .....	16
Лабораторна робота №5 .....	21
Лабораторна робота №6 .....	25
Лабораторна робота №7 .....	33
Теми для самостійного опрацювання .....	40
Завдання для самостійної роботи студентів .....	41
Теми рефератів .....	42
Література .....	43

## ВСТУП

Активне застосування інформаційних систем з елементами штучного інтелекту істотним чином перетворює сучасну повсякденну реальність та формує особливий тип світовідчуття людини. Зазначені закономірності призвели до появи інформаційних систем нового типу під назвою інтелектуальні інформаційні системи, під якими розуміються автоматизовані інформаційні системи, засновані на знаннях, або комплекс програмних, логіко-математичних і лінгвістичних засобів для здійснення підтримки діяльності людини та пошуку інформації. Основним елементом інтелектуальних інформаційних систем є бази знань, в яких накопичені факти та правила про наочну область.

Програма вивчення нормативної навчальної дисципліни “Розробка та супровід баз знань” складена відповідно до освітньо-професійної програми підготовки магістрів спеціальності 8.04030201 Інформатика. Дисципліна розрахована на один семестр і включає два змістовні модулі: "Етапи проектування баз знань та інтелектуальних систем" та "Робота з онтологіями".

**Предметом** вивчення навчальної дисципліни є систематичний огляд моделей представлення знань, проектування баз знань.

**Міждисциплінарні зв'язки:** змістово-функціональна гілка дисципліни "Розробка та супровід баз знань" тісно пов'язана з такими навчальними дисциплінами, як "Теорія штучного та природнього інтелекту", "Бази даних та інформаційні системи", що знаходить своє відображення у міжпредметних зв'язках цих дисциплін.

### **1. Мета та завдання навчальної дисципліни**

Метою викладання навчальної дисципліни “Розробка та супровід баз знань” є формування у студентів базових знань з баз знань та опанування ними практичними навичками розробки баз знань.

Основними завданнями вивчення дисципліни “Розробка та супровід баз знань” полягають у наступному: створити уявлення про штучний інтелект, знання, представлення знань; ознайомити студентів із способами представлення знань у електронному вигляді; розглянути різні моделі представлення знань; познайомити студентів з різноманітними підходами до створення систем зі штучним інтелектом.

У результаті вивчення навчальної дисципліни студент повинен

#### **знати:**

базові поняття штучного інтелекту; моделі подання знань (мережеву, логічну, фреймову); основи онтологічного представлення знань; етапи проектування інтелектуальних систем та баз знань;

#### **вміти:**

будувати логічну, фреймову модель знань; працювати з конструктором онтологій Protege; застосовувати алгоритми і методи для обробки знань у вигляді онтологій; використовувати теоретичні знання в практичній діяльності.

### **володіти компетентностями:**

- здатність до абстрактного мислення, аналізу і синтезу наукової та професійної інформації;
- здатність належно спілкуватися у письмовій, усній та графічній формі;
- здатність працювати самостійно;
- здатність накопичувати, обробляти і систематизувати професійні знання;
- знання предметної області у галузі штучного інтелекту.

## **2. Інформаційний обсяг навчальної дисципліни**

**Змістовий модуль 1.** Етапи проектування баз знань та інтелектуальних систем

### **Тема 1.** Знання та їх властивості

Знання як спосіб подання інформації. Класифікація знань. Особливості знань. Інженерія знань. Подання знань.

### **Тема 2.** Етапи проектування інтелектуальних систем

Визначення інтелектуальних систем. Етапи проектування інтелектуальних систем. Аналіз предметної області і методи придбання знань. Робота з експертами та проблема вилучення знань. Автоматизація вилучення знань і формування моделі.

### **Тема 3.** Проектування баз знань

Розробка бази знань. Поняття знання в контексті досліджуваної проблемної області. Оцінка простору пошуку рішень. Вибір способу представлення знань. Структура бази знань і взаємодію з іншими компонентами інформаційної системи.

### **Тема 4.** Мультиагентні інтелектуальні системи

Основні поняття теорії агентів. Колективна поведінка агентів. Приклади мультиагентних систем. Технології проектування мультиагентних систем. Мультиагентні системи для пошуку інформації. Перспективи мультиагентних технологій.

## **Змістовий модуль 2.** Робота з онтологіями

### **Тема 5.** Моделі подання знань

Вибір моделі представлення знань як необхідний етап розробки інтелектуальної системи. Класифікація моделей. Логічна модель представлення знань. Технології штучного інтелекту.

### **Тема 6.** Онтології як моделі представлення знань

Загальні поняття Semantic Web. Рівні онтологій. Визначення онтології як моделі представлення знань. Формально модель онтології. Модель онтологічної системи. Загальні рекомендації щодо створення онтологій

### **Тема 7.** Protégé як інструментальний засіб створення онтологій

Огляд редакторів онтологій Ontolingua, DOE, OntoEdit, OilEd, WebOnto. Загальна характеристика редактору онтологій Protege. Етапи створення онтологій в Protege.

## ЛАБОРАТОРНА РОБОТА №1

**Тема:** ВИЗНАЧЕННЯ ШТУЧНОГО ІНТЕЛЕКТУ ТА ЗНАНЬ

**Мета:** Отримати знання щодо представлення знань, виявити різницю між знаннями та даними, правилами та фактами.

**Література:** [2], [9], [16], [19]

**Обладнання:** персональний комп'ютер із доступом до мережі Інтернет.

**Питання до контролю попередніх занять, обговорення, самостійного вивчення та осмислення навчального матеріалу:**

1. Визначення понять “інтелект”, “штучний інтелект”
2. Основні напрями досліджень у області ШІ
3. Приклади найбільш відомих ШІ-систем
4. Знання як спосіб подання інформації

### Теоретичні відомості

Історія штучного інтелекту як нового наукового напрямку починається в середині ХХ століття.

Термін інтелект (intelligence) походить від латинського intellectus – що означає розум, розумові здібності людини.

Інтелект (intelligence) – здатність осмислено здобувати, відтворювати і використовувати знання, розуміти конкретні й абстрактні ідеї, досягти відношення між ідеями й об'єктами.

Штучний інтелект (artificial intelligence) – наука і технологія створення інтелектуальних машин (програм), здатних брати на себе окремі функції інтелекту людини, наприклад, вибирати і приймати оптимальні рішення на основі раніше отриманого досвіду і раціонального аналізу зовнішніх впливів.

Штучний інтелект – один з напрямів інформатики, метою якого є розробка комп'ютерних систем, здатних виконувати функції що традиційно вважаються інтелектуальними, – розуміння мови, логічний висновок, використання накопичених знань, навчання, планування дій, розпізнавання образів, а також навчатися і пояснювати свої рішення.

#### *Основні напрями досліджень у області ШІ*

1. Прийняття рішень в завданнях діагностики, проектування та інших подібних завдань. Наприклад, діагностика звуження коронарних судин, контроль за роботою електростанції, синтез електричних ланцюгів і т.д.

2. Розробка природно-мовних інтерфейсів і машинний переклад. Основна складність – текст можна перекласти тільки на основі розуміння його сенсу і в конкретному контексті попередньої інформації.

3. Розпізнавання образів. Цей напрям включає розробку методів представлення інформації про зорові образи. Основний підхід — опис класів

об'єктів через певні значення головних ознак.

4. Навчання і самонавчання. Накопичення і формування знань на основі аналізу і узагальнення даних. Простий метод – кореляційний аналіз (це сукупність заснованих на математичній теорії кореляції методів виявлення кореляційної залежності між двома випадковими ознаками або чинниками).

5. Інтелектуальні роботи. Цей напрям включає вищеперелічені напрями, а також ряд інших напрямів.

6. Ігри і машинна творчість. Цей напрям включає інтелектуальні ігрові завдання – шашки, шахи і т.п., а також твір комп'ютерної музики, віршів, казок, афоризмів і т.п.

*Типи знань* виділяють такі.

1. Базові елементи знання – інформація про властивості об'єктів реального світу.

2. Твердження і визначення – засновані на базових елементах і заздалегідь розглядаються як достовірні.

3. Концепції – перегрупування або узагальнення базових елементів.

4. Відношення – виражають як елементарні властивості базових елементів, так і відношення між концепціями.

5. Теореми і правила перезапису – окремий випадок продукційних правил виду «якщо..., то..., інакше...» з цілком визначеними властивостями.

6. Алгоритми рішення – необхідні для виконання визначених задач.

7. Стратегії й евристика – уроджені або придбані правила поведінки, що дозволяють у конкретній ситуації прийняти рішення про необхідні дії.

8. Метазнання – є присутнім на багатьох рівнях і подає знання того, що відомо, визначає значення коефіцієнта довіри до цього знання, важливість елементарної операції стосовно всієї множини знань. Сюди ж відносяться питання організації різного типу знань і вказівки, де, коли і як вони можуть бути використані.

*Основні визначення до теми*

Інформація (у широкому розумінні) – це будь-які відомості про певний об'єкт, процес або явище.

Факти – це інформація, що розглядається як надійна.

Ієрархію способів подання інформації (у порядку збільшення рівня ієрархії) виділяють таку.

1. Шум – відсутність видимих ознак інформації, складається з інформаційних елементів, що не становлять інтересу і можуть лише ускладнити сприйняття й подання інформації.

2. Дані – потенційне джерело інформації – елементи інформації, що у принципі можуть становити певний інтерес. Даними називають інформацію фактичного характеру, що є фіксованою певним способом та описує об'єкти, процеси і явища предметної області, а також їхні властивості. Дані являють собою ізольовані факти, відносини між якими з зовнішнім світом у них самих не

зафіксовані.

3. Інформація (у вузькому розумінні) – потенційне джерело знань – оброблені дані, що явно становлять інтерес для користувачів.

4. Знання – це формалізована система суджень із принциповою і єдиною організацією, заснована на об'єктивній закономірності, що спостерігається у визначеній предметній області, встановленій в результаті розумової діяльності людини, спрямованої на узагальнення досвіду, отриманого нею у результаті практичної діяльності. Знання визначають здатність використовувати інформацію і являють собою добре структуровані дані або метадані (дані про дані) – елементи інформації, зв'язані між собою і з зовнішнім світом.

*Види методів витягу знань* виділяють такі:

1. Комунікативні методи – охоплюють методи і процедури контактів інженера зі з безпосереднім джерелом знання – експертом. Комунікативні методи поділяються на активні і пасивні.

Пасивні методи включають такі методи, де ведуча роль у процедурі витягу фактично передається експерту, а інженер зі знань тільки фіксує судження експерта під час роботи з прийняття рішень.

Активні методи витягу знань припускають, що ініціатива знаходиться цілком у руках інженера зі знань, який активно контактує з експертом різними способами. Активні методи підрозділяють на індивідуальні і групові.

*У залежності від часу існування* виділяють:

– *статичні знання* – включають логічні правила рішення задач, правила реалізації процедур опитування експертів і користувачів, правила побудови функціональних та процедур опитування і математичних моделей, факти і дані про предметну область; діагностичних моделей і аналізу результатів роботи, а також безпосередньо програмні модулі

– *динамічні знання* – являють собою сукупність фактів і даних, одержуваних у ході рішення задачі, а також висновків (логічних висновків і строгих аналітичних рішень), вироблених у процесі рішення задачі.

У загальному вигляді знання подаються певною *семіотичною* (знаковою) *системою*. З поняттям «знак» безпосередньо зв'язані поняття денотат і концепт. *Денотат* – це об'єкт, що позначається даним знаком, а *концепт* – властивість денотата.

*Інтенціонал* знака визначає зміст пов'язаного з ним поняття через його властивості, тобто значеннєвий зміст поняття. Інтенціонал відокремлює знання від даних, що завжди задаються екстенціонально.

*Екстенціонал* знака визначає конкретний клас усіх його припустимих денотатів. Екстенціонал поняття – набір конкретних фактів, що відповідають даному поняттю.

*Інтенціональні знання* описують абстрактні об'єкти, події, відношення.

*Екстенціональні знання* являють собою дані, що характеризують конкретні



об'єкти, їхній стан, значення параметрів у визначені моменти часу.

Зазначені відмінності призвели до появи спеціальних формалізмів у вигляді *моделей подання знань*.

### ***Порядок виконання роботи***

1. Самостійно ознайомитися із лекцією за даною темою
2. Письмово відповісти на питання, які є у лабораторній роботі
3. Відповісти на поставлені питання викладачем питання за темою

### **Контрольні запитання:**

1. Чим відрізняються знання від фактів
2. Дати визначення поняттю «Онтологія»
3. Дати визначення метазнанням
4. Охарактеризувати типи знань
5. Охарактеризувати різновиди знань
6. Порівняти інтенціональні та екстенціональні знання
7. Порівняти поняття «здоровий глузд», «досвід» та «інтуїція»
8. Охарактеризувати властивості знань
9. Призначення та особливості інженерії знань
10. Методи отримання знань від експертів
11. Дати загальну характеристику теорії подання знань

## ЛАБОРАТОРНА РОБОТА №2

**Тема:** ЕТАПИ ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

**Мета:** Ознайомитися з етапами проектування інтелектуальних систем

**Література:** [2], [7], [8], [12]

**Обладнання:** персональний комп'ютер із доступом до мережі Інтернет.

**Питання до контролю попередніх занять, обговорення, самостійного вивчення та осмислення навчального матеріалу:**

1. Дати визначення інтелектуальним системам
2. Навіщо потрібне проектування інформаційних систем?
3. Сучасні способи проектування інтелектуальних систем

### Теоретичні відомості:

Постійне збільшення обсягів професійних знань та високий ступінь розвитку інформаційно-комунікаційних технологій актуалізує проблему розробки високоякісних програмних продуктів, призначених для задоволення інформаційних потреб користувачів.

Актуальність теми роботи визначається значним розширенням сфери застосування різноманітних інформаційних систем, які засновані на знаннях. Їх можна зустріти і в освіті, і у бізнесі, в управлінській діяльності, тощо. Робота будь-якої інтелектуальної системи передбачає обробку бази знань, яка складає її основу та має властивість поступово накопичувати нові знання.

Проектування інтелектуальних систем - це ітеративний і еволюційний процес, в якому беруть участь кілька фахівців: експерт, що володіє знаннями про предметну область і бажає допомогти роботі зі створення системи, а також фахівці в галузі штучного інтелекту - інженери знань, аналітики та програмісти.

Процес побудови системи ШІ розділений на п'ять етапів (рис. 1.):

1. Ідентифікація визначення завдань і ідентифікація їх характеристик. На цьому етапі встановлюються завдання, які передбачається вирішувати, їх характеристики та особливості.

2. Виділення основних концепцій предметної області, яка відображає знання кола експертів. Це дає можливість аналізувати тип знань, якими оперує експерт.

3. Вибір формалізму представлення знань і визначення механізму виведення рішень. Ці компоненти моделювання в значній мірі впливають на успішне вирішення поставленого завдання з проектування системи.

4. Вибір або розробка мови представлення знань. Після того як правила сформульовані і представлені на обраному мові представлення, вони заносяться інженером знань в БЗ.

5. Тестування системи.

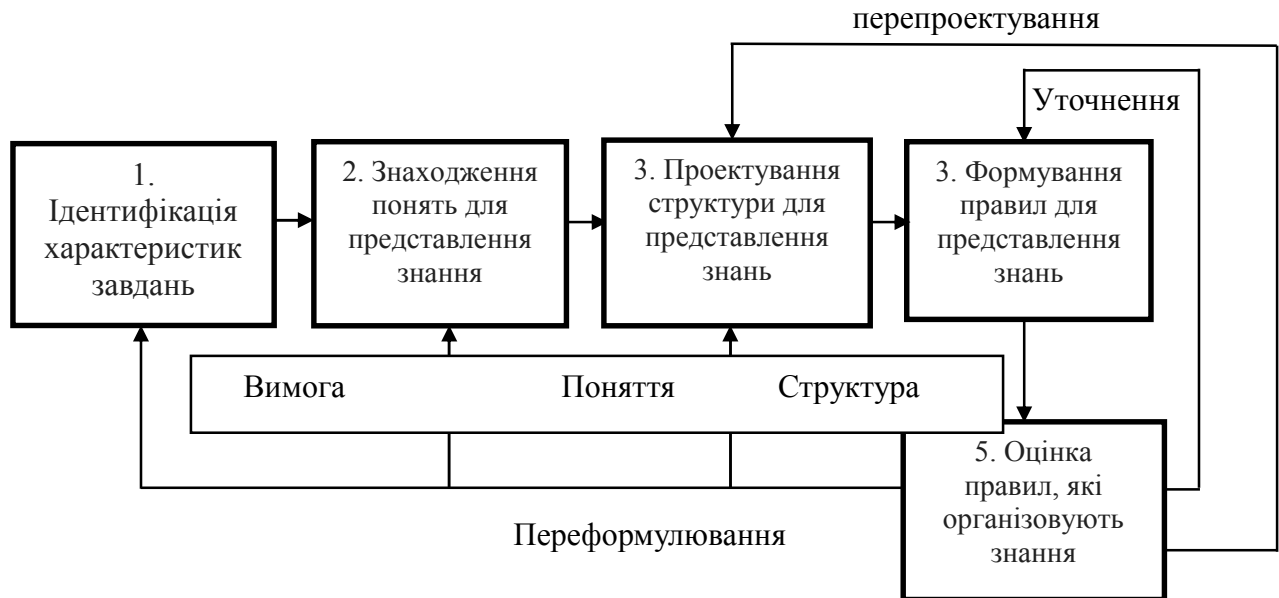


Рис. 1 Етапи проектування ІІІ

Більш докладний перелік робіт при проектуванні інтелектуальних систем:

- витяг знань з експерта і передача їх системі;
- вибір способу представлення знань в системі;
- вибір стратегії виведення (управління);
- вибір підсистеми пояснення;
- вибір підсистеми взаємодії з користувачем;
- вибір адекватних засобів реалізації системи.

Зміст робіт, число етапів проектування інтелектуальних систем, послідовність їх виконання залежать від цілого ряду об'єктивних і суб'єктивних факторів. Однак багато етапів і змістів робіт є загальними і необхідними для інтелектуальних систем практично всіх типів. Нижче наводиться перелік таких етапів та їх складових:

1. Опис проблемної області.
2. Персонал: визначення групи проектувальників та відповідних завдань.
3. Прийняття проекту: проведення організаційного засідання; підготовка до монтажу необхідних технічних засобів та інструментарію; обговорення основного підходу до проблеми; підготовка спеціального-плану розвитку.
4. Прототип системи: розвиток системного прототипу: тестування; отримання інформації про проблемної області.
5. Розвиток повної системи: розширення бази знань прототипу; оцінювання структури інтерфейсу користувача.
6. Верифікація системи: залучення в процес перевірки експертів і потенційних користувачів.
7. Інтеграція системи: виконання розробленої системи; забезпечення сумісності та взаємодії системи з уже діючими.
8. Підтримка системи: забезпечення безперервної підтримки системи;

модернізація БЗ у разі надходження нової інформації.

9. Документація: підготовка повної документації системи; підготовка керівництва для користувача.

### ***Порядок виконання роботи***

1. Самостійно ознайомитися із лекцією за даною темою
2. Письмово відповісти на питання, які є у лабораторній роботі
3. Відповісти на поставлені питання викладачем питання за темою

### **Контрольні запитання:**

1. Охарактеризувати етапи побудови систем штучного інтелекту
2. Способи визначення працездатності системи
3. Охарактеризувати стадії існування інтелектуальних систем
4. Вибір предметної та проблемної області
5. Охарактеризувати навчання систем штучного інтелекту без логічних висновків
6. Охарактеризувати навчання систем штучного інтелекту з використанням інформації, вже представленої у вигляді знань
7. Охарактеризувати навчання за прикладами систем штучного інтелекту
8. Способи виявлення джерел знань
9. Які питання повинні бути вирішені при роботі з експертами
10. Охарактеризувати техніку вилучення знань
11. Загальна характеристика методів і програмних засобів для спрощення процесу придбання знань
12. Автоматизація процесу вилучення знань і формування моделі

## ЛАБОРАТОРНА РОБОТА №3

**Тема:** ПРОЕКТУВАННЯ БАЗИ ЗНАНЬ

**Мета:** вивчення та опанування знаннями про етапи проектування баз знань

**Література:** [2], [11], [15], [19]

**Обладнання:** персональний комп'ютер із доступом до мережі Інтернет.

**Питання до контролю попередніх занять, обговорення, самостійного вивчення та осмислення навчального матеріалу:**

1. Чи може обійтися інтелектуальна система без бази знань
2. Відмінність бази даних від бази знань
3. Чи можна прибрати певні етапи проектування баз знань
4. Інструментальні засоби для створення бази знань

### Теоретичні відомості

Основу - ядро будь-якої інтелектуальної системи - складають база знань і закладений в систему механізм виведення рішень. Якщо говорити узагальнено, ці компоненти визначають дві основні інтелектуальні характеристики системи: здатність зберігати знання про щось і вміння оперувати цими знаннями. Більш розвиненим системам, заснованим на знаннях, притаманна також здатність навчатися, тобто здобувати нові знання, розширювати БЗ, коригувати знання в відповідно до мінливих умов і ситуацією в предметній області.

База знань (knowledge base) в інформатиці та дослідженнях з ШІ – це особливого роду база даних, розроблена для оперування знаннями (метаданими). База знань містить структуровану інформацію, яка покриває деяку область знань, для використання людиною або кібернетичним пристроєм з конкретною метою. Сучасні бази знань працюють спільно з системами пошуку інформації, мають класифікаційну структуру і формат представлення знань.

При проектуванні інтелектуальних систем значні зусилля і час витрачаються на розробку БЗ, тобто на накопичення знань, створення моделі подання знань, їх структурування, заповнення БЗ і подальше підтримання її в актуальному стані. Перш ніж приступити до проектування та реалізації БЗ, розробникам необхідно осмислити і дозволити ряд питань, безпосередньо пов'язаних з процесом створення БЗ та інтелектуальної системи в цілому.

Спробуємо окреслити коло завдань, що вирішуються на початковому етапі розробки інтелектуальної системи (за умови, що питання про доцільність розробки інтелектуальної системи в цій області вирішено позитивно):

- вивчення проблемної області (об'єкта, завдань, цілей), тобто «Що представляти в БЗ» і «для чого представляти»;
- визначення поняття «знання» в контексті досліджуваної проблемної області;
- виявлення джерел знань, активна і копітка робота з ними;

- визначення типів знань для вирішення завдання;
- оцінка на основі дослідження проблемної області та характеру знань простору пошуку рішень з метою вибору способу структуризації знань і методу пошуку рішень (механізму виведення);
- визначення способу структуризації знань, тобто «як представляти знання»;
- вибір способу представлення знань;
- визначення структури БЗ;
- визначення характеру взаємодії структурних частин БЗ, а також взаємодії її з іншими компонентами ІВС в процесі пошуку рішень;
- підготовка до процесу заповнення БЗ.

Необхідно підкреслити, що існують різні варіанти як організації самої БЗ, так і взаємодії БЗ з іншими компонентами ІС. Крім того, на верхньому рівні БЗ наведені метазнання, необхідні для вироблення раціональної стратегії пошуку. Проте висновок рішення або генерація нових правил і знань здійснюється тут за допомогою блоку виведення, який взаємодіє з мета півня БЗ при інтерпретації правил і даних БЗ (рис. 1.)

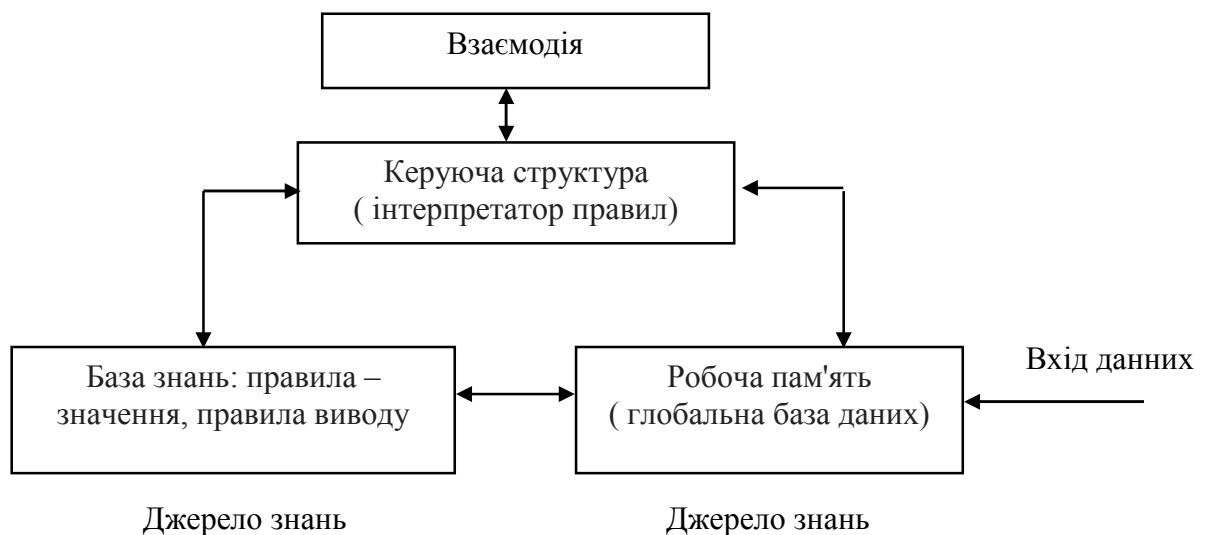


Рис. 1. Структура взаємодії БЗ з основними компонентами ІС для продукційних систем

Глибинні знання - це категорії, абстракції і аналогії, за допомогою яких експерт проходить до розуміння структури та. призначення поточних уявлень. Ці знання використовуються ним переважно при вирішенні неординарних ситуацій.

Поверхневі знання - це «вміння» і навички, відповідні знанню на рівні рефлекторних реакцій, відпрацьованих дій. Сюди ж можна віднести правила та асоціації для стандартних міркувань і ситуацій. У цих асоціаціях взаємозв'язок між вихідними посилками і висновками, зробленими за певними правилами, встановлюється з емпіричного аналізу виниклих асоціацій. Причинність евристично мається на увазі і не закладена в явному вигляді в структурі правила. Цей тип знання можна також назвати експертними знаннями.

Метазнання визначаються як знання інтелектуальної системи про себе,

знання про свою роботу і своєю структурою. Вони дозволяють усвідомлювати, досліджувати й аналізувати лінії міркування самої інтелектуальної системи при вирішенні задачі. На основі цих знань система при поясненні своєї роботи зможе реконструювати раціональний зміст міркувань, виходячи з фундаментальних принципів (глибинних знань експерта), що характеризують предметну область. Цей тип знань при вирішенні задачі ІС використовується ще недостатньо широко і не на належному рівні уявлення, але він має великі потенційні можливості розвитку.

На основі вивчення проблемної області та характеру отримуваних знань можна оцінити простір пошуку рішень. Ця оцінка важлива для вибору способу представлення і структуризації знань, а також так для вибору методу пошуку рішень і побудови процедур виводу. Розміри простору пошуку рішень визначаються багатьма факторами, і насамперед - характером даних і знань предметної області та специфікою вирішуваних завдань. У найбільш простих випадках відбувається деяка ідеалізація проблемної області і приймається, що дані і знання досить детерміновані і надійні. У деяких ІВС цих припущень достатньо для досить успішного їх функціонування. Все залежить від поставлених перед системою завдань і складності предметної області.

### ***Порядок виконання роботи***

1. Самостійно ознайомитися із лекцією за даною темою
2. Письмово відповісти на питання, які є у лабораторній роботі
3. Відповісти на поставлені питання викладачем питання за темою

### **Контрольні запитання**

1. Дати визначення поняттю «база знань», чим бази знань відрізняються від бази даних
2. Які завдання повинні бути вирішені на початковому етапі розробки інтелектуальної системи
3. Дати визначення глибинним та поверхневим знанням
4. Дати визначення мета знанням, для чого вони потрібні
5. Рекомендації щодо визначення простору пошуку рішень
6. Рекомендації щодо вибору способу представлення знань
7. Рекомендації щодо структури бази знань

## ЛАБОРАТОРНА РОБОТА №4

**Тема:** ЗАГАЛЬНІ ПОНЯТТЯ ПРО МУЛЬТИАГЕНТНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

**Мета:** вивчення та опанування знаннями про мультиагентні системи

**Література:** [1], [13], [14]

**Обладнання:** персональний комп'ютер із доступом до мережі Інтернет.

**Питання до контролю попередніх занять, обговорення, самостійного вивчення та осмислення навчального матеріалу:**

1. Дати визначення інтелектуального агента та мультиагентної системи
2. Основні властивості інтелектуальних агентів
3. Сфери застосування мультиагентних систем

### Теоретичні відомості

Інтелектуальні мультиагентні системи - одне з нових перспективних напрямів штучного інтелекту, яке сформувалося на основі результатів досліджень в області розподілених комп'ютерних систем, мережових технологій вирішення проблем і паралельних обчислень. У мультиагентних технологіях закладений принцип автономності окремих частин програми (агентів), спільно функціонуючих в розподіленій системі, де одночасно протікає безліч взаємопов'язаних процесів.

Під *агентом* розуміється автономний штучний об'єкт (комп'ютерна програма), що володіє активною мотивованою поведінкою і здатний до взаємодії з іншими об'єктами в динамічних віртуальних середовищах. Кожен агент може приймати повідомлення, інтерпретувати їх зміст і формувати нові повідомлення, які або передаються на «дошку оголошень», або направляються іншим агентам.

Інтелектуальним агентам притаманні такі основні властивості:

- автономність - здатність функціонувати без втручання з боку свого власника і здійснювати контроль власних дій і внутрішнього стану. Автономність припускає відносну незалежність агента від навколишнього середовища, тобто наявність «свободи волі», обумовлює власну поведінку, яка повинна бути забезпечена необхідними ресурсами;
- активність - здатність до організації та реалізації дій;
- товарицькість (*рос. общительность*) - взаємодія і комунікація з іншими агентами;
- реактивність - адекватне сприйняття стану середовища і реакція на його зміну;
- цілеспрямованість, що припускає наявність власних джерел мотивації;
- наявність базових знань про себе, про інших агентів, навколишнє середовище;
- впевнення - змінна частина базових знань, мінливих у часі;



- побажання - прагнення до певних станів;
- наміри - дії, які плануються агентом для виконання своїх зобов'язань та/або бажань;
- обов'язки - завдання, які виконує один агент на прохання і/або доручення інших агентів.

Іноді до цього списку додаються інші якості, у тому числі:

- правдивість - нездатність до підміни істинної інформації завідомо неправдивою;
- доброзичливість - готовність до співпраці з іншими агентами у процесі вирішення власних завдань, що зазвичай передбачає відсутність конфліктуючих цілей, поставлених перед агентами;
- альтруїзм - пріоритетність загальних цілей у порівнянні з особистими;
- мобільність - здатність агента мігрувати по мережі в пошуках необхідної інформації.

Одна з можливих класифікацій агентів наведена в табл. 1, з якої випливає, що для інтелектуальних агентів характерно доцільну поведінку, яке передбачає наявність у агента цілей функціонування і здібностей використовувати знання про довкілля, партнерах і про свої можливості.

Таблиця 1.

#### Класифікація агентів

Ознака	Тип агента			
	Простий	Розумний	Інтелектуальний	Дійсно інтелектуальний
Автономність	+		+	+
Взаємодія з іншими агентами та/або користувачами	+	+	+	+
Реактивність	+	+	+	+
Здатність використання абстракції		+	+	+
Адаптивна поведінка		+	+	+
Навчання на основі взаємодії з оточенням			+	+
Толерантність до помилок та/або невірних вхідних сигналів			+	

Головна риса МАС, що відрізняє їх від інших інтелектуальних систем – взаємодія між агентами. Взаємодія означає встановлення двосторонніх і багатосторонніх динамічних відносин між суб'єктами. Воно є не тільки наслідком

діяльності агентів, але і необхідною умовою формування віртуальних спільнот. Взаємодія – не просто зв'язок між співіснуючими агентами, а й передумова для взаємних перетворень самих агентів і відносин між ними. Головними характеристиками будь-якої взаємодії є спрямованість, вибірковість, інтенсивність і динамічність.

Загальна проблема аналізу взаємодії між агентами включає наступні завдання:

- ідентифікацію ситуації взаємодії агентів;
- виділення основних ролей і їх розподіл між агентами;
- визначення числа і типів взаємодіючих агентів;
- побудова формальної моделі взаємодії;
- визначення набору можливих стратегій поведінки агентів;
- формування безлічі комунікативних дій.

До базових видів взаємодії між агентами відносяться:

- кооперація (співробітництво);
- конкуренція (конфронтація, конфлікт);
- компроміс (облік інтересів інших агентів);
- конформізм (відмова від своїх інтересів на користь інших);
- ухилення від взаємодії.

У роботі безліч можливих ситуацій вибору поведінки пари агентів класифікується наступним чином.

1. Симетрична кооперація, коли існує непорожнє безліч стратегій (переговорний безліч), при використанні яких обидва агента досягають своїх цілей і отримують більший ефект, ніж в ситуаціях, коли вони діють поодиночі.
2. Симетричний компроміс, коли досягнення мети поодиночі більш вигідно для кожного агента, однак неможливо у присутності іншого агента.
3. Несиметрична кооперація або несиметричний компроміс - один з агентів може самостійно досягти своєї мети у присутності іншого агента, а інший - тільки за рахунок кооперації з першим.
4. Конфлікт - переговорний безліч порожньо, тобто не існує стратегій, що забезпечують досягнення цілей обох агентів.

Взаємодія агентів обумовлено цілим рядом причин, найважливішими серед яких є наступні:

1. Сумісність цілей (загальна мета). Ця причина зазвичай породжує взаємодія по типу кооперації або співробітництва. При цьому слід з'ясувати, чи не веде взаємодія до зниження життєздатності окремих агентів. Несумісність цілей або переконань зазвичай породжує конфлікти, позитивна роль яких полягає в стимулюванні процесів розвитку. Відома модель хижак-жертва являє собою приклад одночасної взаємодії за двома типами кооперація-конфронтація.
2. Ставлення до ресурсів. Ресурсами будемо називати будь-які засоби, використовувані для досягнення агентами своїх цілей. Завдання розподілу часток ринку, витрат і прибутків спільних підприємств можна розглядати як

приклади взаємодії, обумовленого загальними ресурсами. Обмеженість ресурсів, які використовуються багатьма агентами, звичайно породжує конфлікти. Одним їх найбільш простих і ефективних способів вирішення подібних конфліктів є право сильного - сильний агент відбирає ресурси у слабких. Більш тонкі способи вирішення конфліктів забезпечують переговори, спрямовані на досягнення компромісів, в яких враховуються інтереси всіх агентів.

Кожен агент має обмеженим набором знань, необхідних йому для реалізації власних і спільних цілей. У зв'язку з цим йому доводиться взаємодіяти з іншими агентами. При цьому можливі різні ситуації:

- агент здатний виконати завдання самостійно;
- агент може обійтися без сторонньої допомоги, але кооперація дозволить вирішити завдання більш ефективним способом;
- агент не здатний вирішити завдання поодиноці.

Залежно від ситуації агенти вибирають тип взаємодії і можуть проявляти різну ступінь зацікавленості у співпраці.

3. Взаємні зобов'язання. Зобов'язання є одним з інструментів, що дозволяють упорядкувати хаотичні взаємодії агентів. Вони дозволяють передбачати поведінку інших агентів, прогнозувати майбутнє і планувати власні дії. Можна виділити наступні групи зобов'язань: а) зобов'язання перед іншими агентами; б) зобов'язання агента перед групою; в) зобов'язання групи перед агентом; г) зобов'язання агента перед самим собою.

*Моделі колективної поведінки автоматів.* Вони засновані на ідеях рандомізації, самоорганізації і повної розподіленост. Моделі цього типу підходять для побудови протоколів переговорів у завданнях, які характеризуються великою кількістю дуже простих взаємодій з невідомими характеристиками.

*Моделі планування колективної поведінки.* Планування може бути централізованим, частково централізованим або розподіленим (децентралізованим). В останньому випадку агенти самі приймають рішення про вибір своїх дій у процесі координації приватних планів, у зв'язку з чим виникають питання про раціональної децентралізації, про можливість зміни цілей при виникненні конфліктів, а також проблеми обчислювальної складності.

*Моделі на основі BDI-архітектур.* У моделях цього класу застосовуються аксіоматичні методи теорії ігор і логічної парадигми штучного інтелекту. Для опису агентів використовуються логічні засоби, у тому числі темпоральні і модальні логіки. Акцент робиться на описі інтенціональних понять, таких, як переконання (belief), бажання (desire) і наміри (intention). Логічний висновок здійснюється безпосередньо в процесі функціонування агентів, що призводить до високої складності моделей, обчислювальним труднощам і до проблем, пов'язаних з аксіоматичним описом нетривіальних ситуацій, наприклад, коли перед агентом виникає вибір між рішенням власної завдання і виконанням зобов'язань по відношенню до партнерів.

*Моделі на основі конкуренції.* У моделях даного класу використовується поняття аукціон в якості механізму координації поведінки агентів. Використання механізму аукціону ґрунтується на припущенні про можливість явної передачі «корисності» від одного агента до іншого або до агента-аукціонера, причому ця корисність зазвичай має сенс грошей.

*Аукціони* прийнято розділяти на відкриті та закриті. У першому випадку пропонувані ціни оголошуються всім учасникам. У закритому аукціоні пропонувані цінах знає тільки аукціонер. Відкриті аукціони розрізняються за способом проведення. У так званих англійських аукціонах зазвичай задається стартова ціна, яка може збільшуватися учасниками в ході торгів. Перемагає той, хто дасть максимальну ціну. Закриті аукціони поділяють на аукціони першої та другої ціни. В аукціонах першого ціни перемагає той, хто запропонував найвищу ціну, відому тільки аукціонерам. В аукціонах другої ціни переможець визначається таким же способом, але платить за товар не свою ціну, а другу за величиною.

### ***Порядок виконання роботи***

1. Самостійно ознайомитися із лекцією за даною темою
2. Письмово відповісти на питання, які є у лабораторній роботі
3. Відповісти на поставлені питання викладачем питання за темою

### **Контрольні запитання**

4. Класифікація агентів за ступенем розвитку внутрішнього уявлення про зовнішній світ
5. Класифікація агентів за способом поведінки
6. Перечислити завдання, що повинні бути вирішені при аналізі взаємодії між агентами
7. Перечислити базові видів взаємодії між агентами
8. Перечислити основні причини взаємодії інтелектуальних агентів
9. Перечислити основні ознаки природних систем, які необхідно враховувати при моделюванні віртуальних середовищ
10. Перечислити основні ідеї, що використовуються в моделях координації поведінки агентів
11. Типи можливих ситуацій вибору поведінки пари агентів
12. Охарактеризувати теоретико-ігрові моделі координації поведінки агентів
13. Охарактеризувати моделі колективної поведінки автоматів та моделі планування колективної поведінки
14. Охарактеризувати моделі на основі BDI-архітектур та моделі на основі конкуренції
15. Охарактеризувати МАС «Віртуальне підприємство»
16. Охарактеризувати МАС «Електронний магазин»

## ЛАБОРАТОРНА РОБОТА №5

**Тема:** СПОСОБИ ПРЕДСТАВЛЕННЯ ЗНАНЬ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

**Мета:** вивчення та опанування знаннями про способи подання знань в інтелектуальних системах

**Література:** [4], [7], [Вибір 17]

**Обладнання:** персональний комп'ютер із доступом до мережі Інтернет.

**Питання до контролю попередніх занять, обговорення, самостійного вивчення та осмислення навчального матеріалу:**

1. Навищо взагалі представляти знання в інтелектуальних системах
2. Які існують на даний момент сучасні способи подання знань
3. В чому полягає проблема представлення та подання знань при розробці інтелектуальних систем
4. Які вимоги висуваються до моделей подання знань, які застосовуються у навчальних інтелектуальних системах

### Теоретичні відомості

Основа будь-якої інтелектуальної системи складають база знань і закладений в систему механізм виведення рішень. Ці компоненти визначають дві основні інтелектуальні характеристики системи: здатність зберігати знання про щось і вміння оперувати цими знаннями. Тому на початковому етапі розробки інтелектуальної системи потрібно було визначитися із моделлю представлення знань.

Взагалі, поняття знань розглядається як сукупність даних, що мають розвинену і складну структуру. Знання в базі знань подаються у вигляді певних інформаційних одиниць знань і зв'язків між ними. Слід зазначити, що форма подання знань істотно впливає на властивості та характеристики інформаційних систем, тому це є однією з важливіших проблем, характерних для систем, орієнтованих на знання. Оскільки логічні висновки та дії над знаннями в інформаційних інтелектуальних системах проводяться програмним чином, то знання не можуть бути подані безпосередньо в звичайному вигляді, тобто в тому вигляді, в якому вони використовуються користувачами системи.

Тому одним із головних завдань, при створенні інтелектуальних інформаційних систем, полягає у структуруванні та занесенні знань так, щоб програмні агенти або модулі, які входять до складу інтелектуальної системи, змогли осмислено обробляти їх та задовольняти інформаційні потреби користувача.

Проблема представлення та подання знань полягає в тому, що в більшості випадків знання є продуктом діяльності людини, точніше, експерта в деякій

предметній області. Ці знання часто мають якісний, але суперечливий характер. Для подальшої обробки та використання цих знань, з урахуванням специфіки функціонування комп'ютера, вони повинні бути приведені до однозначного формалізованого виду.

До моделей подання знань, які застосовуються у навчальних інтелектуальних системах, повинні висуватися такі вимоги:

- прогнозована ефективність – здатність системи оцінити якість згенерованого рішення;
- здатність до пояснення рішення;
- продуктивність – прогнозований час реакції на зовнішній вплив і наявність механізмів оптимізації процедур виведення;
- здатність до навчання – придатність системи в автоматизованому режимі накопичувати знання і враховувати нові вхідні дані;
- масштабованість – можливість нарощування бази знань або об'єднання декількох баз знань різних рівнів в єдину ієрархічну систему;
- можливість імпорту-експорту знань – придатність бази знань до модернізації в процесі еволюції системи;
- наочність – придатність до сприйняття і аналізу цілісного образу знань.

Центральним питанням побудови систем, заснованих на знаннях, є вибір форми представлення знань. Представлення знань - це спосіб формального вираження знань про предметну область у комп'ютерно-інтерпритованій формі. Відповідні формалізми, що забезпечують зазначене представлення, називають моделями представлення знань.

Модель є інформаційним еквівалентом частини реального світу (предметної області) і лежить в основі процесу пізнання. В інженерній практиці саме моделі асоціюються зі знанням. Це знання, на основі, яких можна ефективно використовувати об'єкти реального світу для досягнення визначених цілей.

Модель представлення знань повинна відбивати істотні характеристики задачі, що розв'язується і забезпечувати відповідною інформацією процедури, що виконують пошук рішень.

Конкретні моделі, застосовувані на практиці, являють собою комбінацію декларативних і процедурних представлень. Найбільш розповсюдженими є наступні моделі представлення знань: логічні моделі; продукційні моделі; мережні моделі; фреймові моделі.

*Логічні моделі* реалізуються засобами логіки предикатів. У цьому випадку знання про предметну область представляються у вигляді сукупності логічних формул.

У продукційних моделях знання представляються набором правил виду "Якщо А, то В", де умова правила А є твердженням про вміст бази фактів, а наслідок В говорить про те, що треба робити, коли дане продукційне правило активізоване.

Семантичні мережі є винятковим випадком мережних моделей

представлення знань. Формально мережні моделі задаються у вигляді.

$$H = \langle I, C_1, C_2, \dots, C_n, Q \rangle,$$

де  $I$ -множину інформаційних елементів, що зберігаються у вузлах мережі;  $C_1, C_2, \dots, C_n$  - типи зв'язків між інформаційними елементами;  $Q$  - відображення, що встановлює відповідність між множиною типів зв'язків і множиною інформаційних елементів мережі.

Фреймові моделі представлення знань використовують теорію організації пам'яті, розуміння і навчання, запропоновану М.Мінським. Фрейм (від англ. frame - рамка, каркас, кістяк) - структура даних, призначена для представлення стереотипних ситуацій. Фрейм складається зі слотів (slot - гніздо, щілінка, паз).

Описи предметних областей, виконані в логічних мовах, називаються логічними (формальними) моделями. В основі моделей такого типу лежить формальна система, що задається четвіркою виду:  $M = \langle T, P, A, B \rangle$ .

Множина  $T$  є множина базових елементів різної природи, наприклад слів з деякого обмеженого словника, деталей дитячого конструктора, що входять до складу деякого набору тощо.

Множина  $P$  є множина синтаксичних правил. З їхньою допомогою з елементів  $T$  утворюються синтаксично правильні сукупності

У множині синтаксично правильних сукупностей виділяється деяка підмножина  $A$ . Елементи  $A$  називаються аксіомами (або постулатами). Аксіомами є істинні логічні вирази. Множина  $B$  є множина правил виведення. Застосовуючи їх до елементів  $A$ , можна одержувати нові синтаксично правильні сукупності, до яких знову можна застосовувати правила з  $B$ .

*Продукційні моделі* можна вважати найбільш поширеними моделями представлення знань. Продукції — це сукупність правил вигляду “умова — дія”:

$$\text{ЯКЩО (умова), ТО (дія).}$$

Під продукцією розуміється вираз:

$$(i); Q; P; A \rightarrow B; N; R,$$

де  $i$  — ім'я продукції;

$Q$  — сфера застосування продукції;

$P$  — умова застосовності продукції;

$A \rightarrow B$  — ядро продукції, звичайно у вигляді "ЯКЩО-ТО-ІНАКШЕ",

$N$  — післямова продукції, що актуалізується при позитивній реалізації продукції;

$R$  — неформальне пояснення (обґрунтування) продукції, коментар, час введення в БЗ і т.д.

Такі правила звичайно є емпіричними. Крім того, будучи незалежними один від одного, правила вимагають контролю на несуперечність і утрудняють відображення ПрГ як цілісної системи.

Механізм, реалізований як засіб виводу в продукційних системах, називається машиною логічного висновку (МЛВ) і виконує функції пошуку в базі правил, послідовного виконання операцій над знаннями і отримання висновків.

Існує два способи проведення таких висновків — прямі висновки і зворотні висновки.

Прямим висновкам (прямому ланцюжку міркувань) відповідає шлях від посилок до слідств.

Зворотнім виводам (зворотному ланцюжку міркувань) відповідає шлях від мети (факту, який потрібно встановити) до передумов.

Приклад правил продукції:

ЯКЩО: противник розгортається у батальйонні колони  
і пророблюються проходи в мінних полях і загородженнях,  
і артилерія противника веде вогонь по передньому краю,  
і авіація противника завдає ударів по об'єктах,

ТО: ситуація класифікується як “Противник на підступах до переднього краю оборони полку”.

Кожна з розглянутих моделей представлення знань має свої переваги та недоліки. Використання тієї або іншої з них у реальних системах обумовлене специфікою предметної галузі, а також наявністю відповідних технічних і програмних засобів.

Зазначимо, що для більшості розроблених ІС характерне одночасне використання декількох моделей представлення знань. Досить повний опис різних методів представлення, обробки знань і їх теоретичних основ можна знайти в літературі.

### ***Порядок виконання роботи***

1. Самостійно ознайомитися із лекцією за даною темою
2. Письмово відповісти на питання, які є у лабораторній роботі
3. Відповісти на поставлені питання викладачем питання за темою

### **Контрольні запитання**

5. Дати визначення поняттю «модель» та «модель представлення знань»
6. Загальна характеристика декларативних і процедурних моделей
7. Класифікація моделей представлення знань
8. Загальна характеристика логічних моделей
9. Загальна характеристика фреймових моделей
10. Загальна характеристика мережних моделей
11. Загальна характеристика продукційних моделей
12. Дати характеристику формальній системі, яка використовується у логічній моделі представлення знань
13. Висновок у формальній логічній системі



## ЛАБОРАТОРНА РОБОТА №6

**Тема:** Створення онтології засобами редактору Protege

**Мета:** опанувати вміннями створювати простіші онтології у редакторі онтологій Protege

**Література:** [3], [5], [6], [10]

**Обладнання:** персональний комп'ютер із доступом до мережі Інтернет.

**Питання до контролю попередніх занять, обговорення, самостійного вивчення та осмислення навчального матеріалу:**

1. Що дозволяє зробити онтологія
2. Де використовуються онтології
3. Порівняти різні редактори онтологій

### Теоретичні відомості

У загальних рисах під онтологією розуміється система понять деякої предметної області, яка представляється як набір сутностей, з'єднаних різними відносинами. Онтології використовуються для формальної специфікації понять і відносин, які характеризують певну галузь знань. Перевагою онтологій як способу представлення знань є їх формальна структура, яка спрощує їх комп'ютерну обробку.

Оскільки класифікаційна структура (таксономія) є невід'ємною частиною будь-якої онтології, можна говорити про присутність елементів онтологій в спеціальних класифікаціях і системах індексації (наприклад, в бібліотечних класифікаційних кодах). У явному вигляді онтології використовуються як джерела даних для багатьох комп'ютерних додатків, дозволяючи більш ефективно обробляти складну і різноманітну інформацію. Цей спосіб представлення знань дозволяє додаткам розпізнавати ті семантичні відмінності, які є само собою зрозумілими для людей, але не відомі комп'ютеру. Саме поняття онтології відомо давно, але, будучи переосмисленим, воно стало застосовуватися в комп'ютерних технологіях лише недавно.

Повноцінна розробка онтологій у новому змісті цього терміна почалося лише наприкінці 90-х. Це досить нова і мало розроблена галузь прикладної лінгвістики. Вже на даному етапі створено ряд обширних онтологій, що включають кілька тисяч понять: OMEGA, SUMO, DOLCE та інші.

Онтології широко використовуються у всіх областях, що займаються обробкою даних на природній мові. У зв'язку з використанням онтологій в різних додатках виникла необхідність створення стандартизованих способів їх подання. Почалося розвиток різноманітних мов, які могли б застосовуватися повсюдно у всіх системах, найвідомішими є RDF і OWL. Виникло також велика кількість редакторів для створення, поповнення та зміни онтологій. Кожне з цих коштів

зазвичай спрямоване на роботу з певним форматом даних і володіє своїми особливостями.

Повна (або комп'ютерна) онтологія є (формальним) виразом концептуальних знань про предметну область і за своєю значимістю порівнянна з базою знань знання-орієнтованої інформаційної системи (Зоіс), а її побудова є специфічною формою людської творчості. Творчий процес можна представити сукупністю операцій-процедур з судженнями, твердженнями, поняттями і відносинами між ними. А останні є фундаментом, основою для побудови складової частини наукової теорії - онтологічної бази знань у заданій предметній області. При цьому такі знання описуються в декларативною формі.

Формально *модель онтології* определяется как  $O = \langle X, R, F \rangle$ , где  
**X** - конечное множество понятий (концептов) предметной области,  
**R** - конечное множество отношений между понятиями,  
**F** - конечное множество функций интерпретации, заданных на концептах и/или отношениях.

При  $R=0$  и  $F=0$  онтология трансформируется в простой словарь. Пример – индексы поисковых машин

*Модель онтологической системы:  $Z = \langle O, P, M \rangle$ , где*

**O** – онтология верхнего уровня (метаонтология) (содержит общие понятия и отношения, не зависящие от предметной области - «объект», «свойство», «значение» и т.п.);

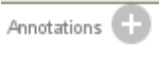
**P** – множество предметных онтологий и онтологий задач предметной области (с учетом предпочтений пользователя);

**M** – модель машины вывода данной онтологической системы (например, для изменения критериев релевантности поиска или критериев формирования репозитория).

У центрі більшості онтологій знаходяться класи. Класи описують поняття предметної області. Наприклад, клас вин представляє всі вина. Конкретні вина - екземпляри цього класу. Вино Bordeaux в келиху перед вами, коли ви читаете цей документ, - це екземпляр класу вин Bordeaux. Клас може мати підкласи, які представляють більш конкретні поняття, ніж надклас.

### ***Порядок виконання роботи***

1. Завантажити Protege 4. Автоматично буде створений новий проект.
2. Кожна онтологія створюється з використанням унікального імені - ідентифікатора ресурсу (URI). Замініть значення за замовчуванням на <http://www.pizza.com/ontologies/pizza.owl>.
3. Виконайте команду File – SaveAs та збережіть створену онтологію під назвою lab\_pizza.owl

4. На вкладинці «Active Ontology» натиснути на кнопку  та ввести коментар до онтології, як показано на рис. 1.

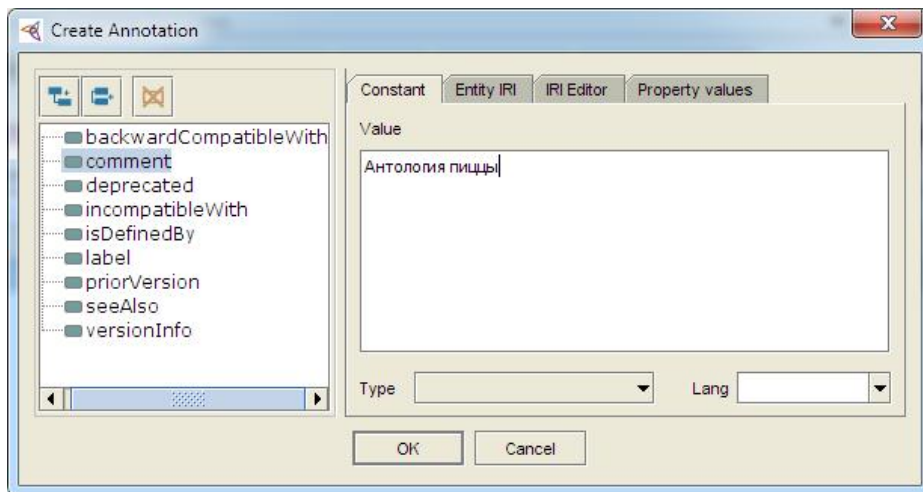


Рис. 1. Введення коментарю до онтології

5. Як вказувалося у лекції, основними будівельними блоками онтології OWL є класи. У Protege 4 створення класів здійснюється в закладці «Classes». Порожня онтологія містить один клас з ім'ям THING. В OWL класи інтерпретуються як безлічі індивідів (або об'єктів). Клас THING - це клас, що представляє набір, що містить всі об'єкти предметної області. З цієї причини всі класи є підкласами THING.

6. Давайте додамо деякі класи в онтології для того щоб визначити, якими бувають піци. Для цього слід переконатися, що обрана вкладка «Classes» і виділений батьківський клас THING. Натиснути кнопку Add Subclass «Додати підклас». Ця кнопка створює новий клас як підклас обраного класу (в даному випадку ми хочемо створити підклас класу THING). У діалоговому вікні для введення назви класу слід ввести Pizza і натиснути Enter.

7. Аналогічно додати класи PizzaTopping і PizzaBase, враховуючи, що прапорець встановлений на THING до натискання кнопки «Додати підклас». Таким чином класи будуть створюватися як підкласи THING. Ієрархія класів повинна бути подібною до ієрархії класів на рис. 2.

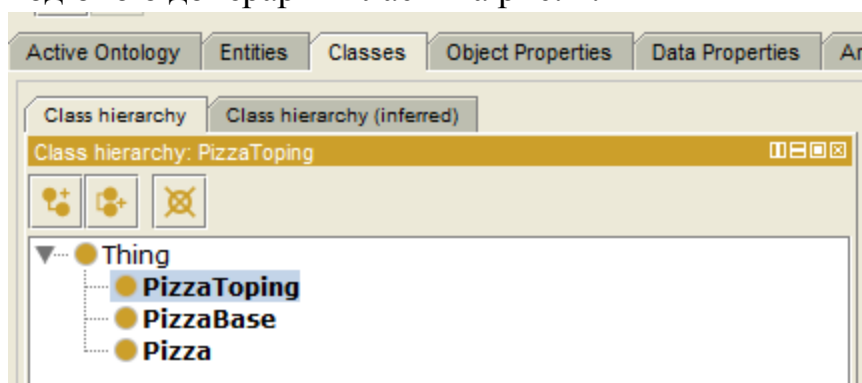


Рис. 2 Ієрархія класів

8. Додавши класи Pizza, Pizza Topping і PizzaBase до онтології, можна вказати, що ці класи є непересічними, так що будь-який об'єкт не може бути екземпляром більш ніж одного з цих трьох класів. Для вказівки, що обраний клас не перетинається з іншими класами, слід виділити клас Pizza, натиснути на кнопку «Disjoint with», яка розташована в нижній частині панелі «Description». Після цього з'явиться діалогове вікно, де слід вибрати класи PizzaBase і

PizzaTopping (родинні класи Pizza), що зробить їх непересічними з класом Pizza (рис. 3).

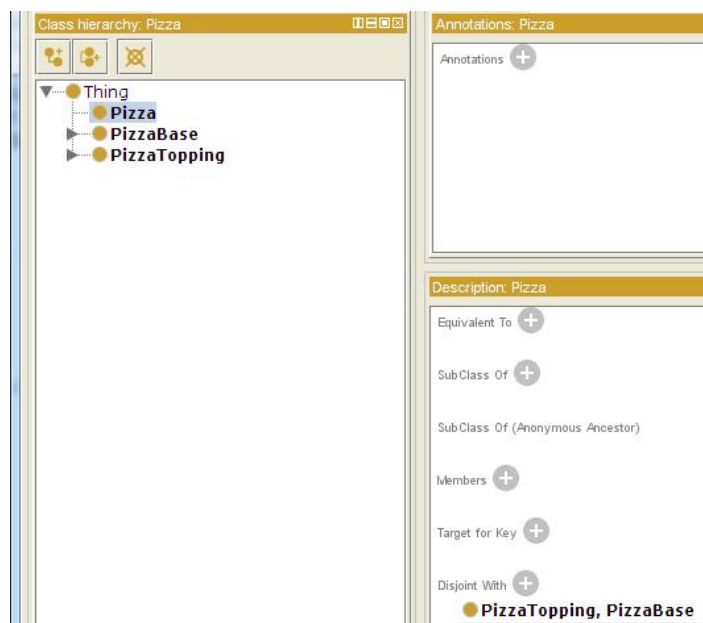


Рис. 3. Disjoint with

9. Створити класи ThinAndCrispy и DeepPan як підкласи PizzaBase. Для цього слід в ієрархії класів вибрати клас PizzaBase. У головному меню вибрати Tools – Create Class hierarchy.

10. З'явиться панель інструментів, де вибраний клас PizzaBase. Слід натиснути кнопку «Далі» (Continue). На наступному кроці потрібно вказати у вікні підкласи PizzaBase. У текстовому полі слід ввести ім'я класу ThinAndCrispyBase (для тонкої і хрусткої піци) і натиснути Enter. На новому рядку ввести ім'я класу DeepPanBase, щоб набране виглядало так, як показано на малюнку 5.

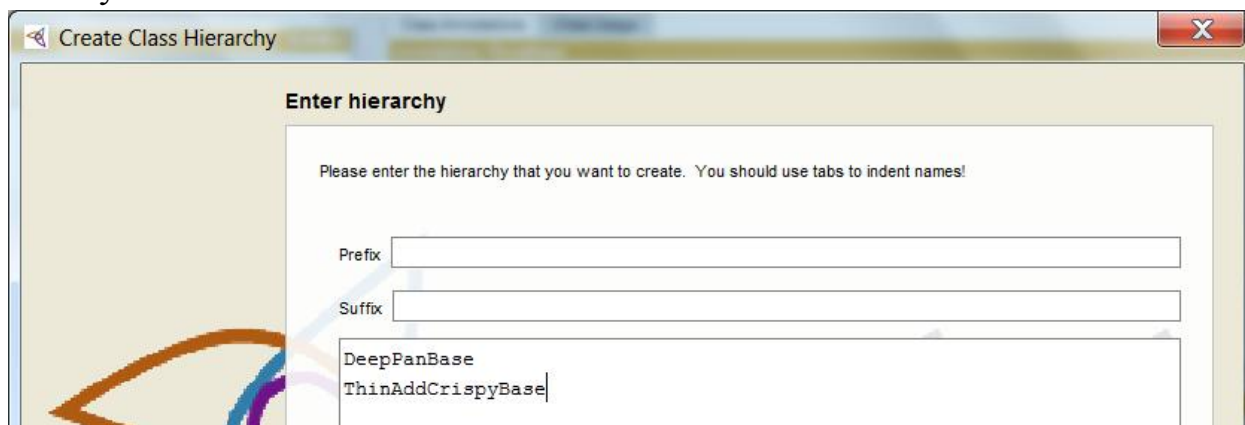


Рис. 5. Підкласи класу PizzaBase

11. Після натискання кнопки «Далі» майстер створює класи, робить їх непов'язаними (опція «Make sibling classes disjoint»). На останньому етапі слід натиснути кнопку «Готово» (Finish) для закриття інструменту. Онтологія тепер повинна мати ThinAndCrispyBase і DeepPanBase, як підкласи PizzaBase. Ці нові класи повинні бути непов'язаними між собою. Таким чином PizzaBase не може бути тонкою і хрусткий і одночасно пишною.

12. Аналогічно слід створити підкласи для класу PizzaTopping. Слід звернути

увагу, що у даному завданні слід ввести значення суфіксу «Topping». Для створення ієрархії класів слід використовувати Enter и Tab.

13. Після завдання структурованого списку класів, натисніть кнопку «Далі» і потім переконаєтеся що опція «Make sibling classes disjoint»). відзначена таким чином, щоб нові родинні класи робилися непов'язаними один з одним.

14. Після виконання зазначених вище дій повинна бути створена онтологія з класами та підкласами, показаними на рис. 6.

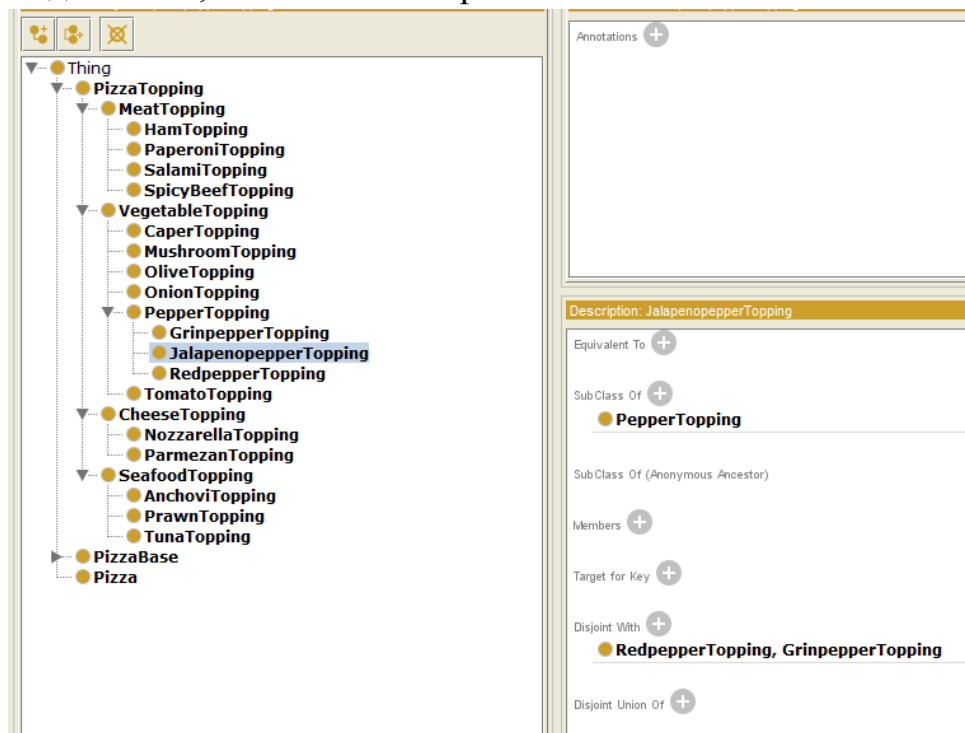


Рис. 7. Ієрархія класу Thing

15. Властивості можуть бути створені за допомогою вкладки «Властивості об'єкта». Нам потрібно створити властивість об'єкта під назвою hasIngredient. Для цього слід перейти на вкладку «Object Properties» та використати кнопку «Add Sub Property» (див. рис. 7) для створення нової властивості об'єкта, ввести ім'я властивості hasIngredient в діалоговому вікні «Ім'я властивості».

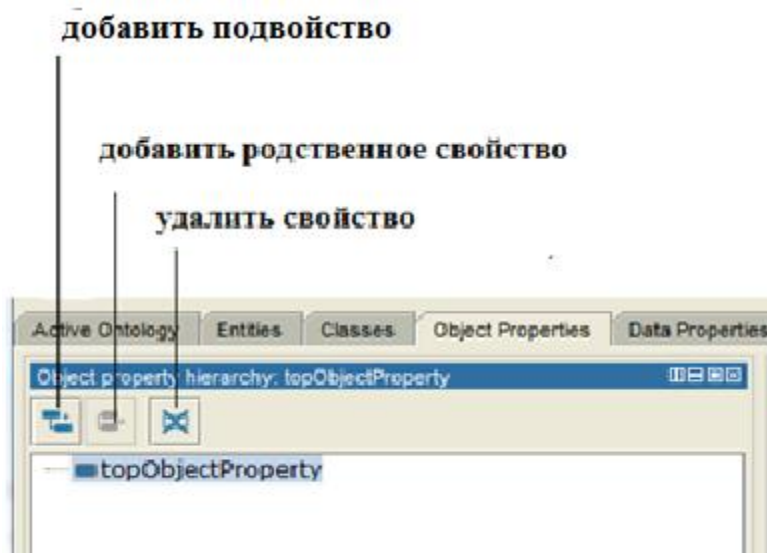


Рис. 7. Вкладка «Object Properties»

**Примітка!** Хоча немає ніяких обмежень на імена для властивостей, ми рекомендуємо імена властивостей починати з літери нижнього регістра, не мати прогалин і використовувати накоплюваючі (раніше задіяні) імена. Ми також рекомендуємо створювати властивості з префіксом слова *has* – «має» або слова *is* – «є», наприклад *hasPart*, *isPartOf*, *hasManufacturer*, *isProducerOf*.

**Примітка!** У OWL властивості можуть мати підвластивості таким чином, щоб це відображало форму ієрархії властивостей. Підвластивості спеціалізуються їх супер-властивістю (так само, як підкласи спеціалізуються їх суперкласом).

16. Після додавання властивості *hasIngredient* («має Інгредієнт») ми додамо два додаткові властивості - *hasTopping* і *hasBase*. У нашій онтології *Pizza* властивості *hasTopping* і *hasBase* повинні створюватися як підвластивості *hasIngredient*. Якщо властивість *hasTopping* (або властивість *hasBase*) пов'язує двох індивідів це означає, що два індивіди пов'язані у властивості *hasIngredient*.

17. Щоб створити властивість *hasTopping* як підвластивість властивості *hasIngredient*, слід вибрати властивість *hasIngredient* в ієрархії властивостей на вкладці «Object Properties» та натиснути на кнопку «Add sibling class». Буде створено нову властивість об'єкта як підвластивість властивості *hasIngredient*. Їй потрібно задати значення імені «*hasTopping*».

18. Виконати аналогічні дії для створення підвластивості «*hasBase*».

19. Створимо зворотні властивості властивостей *hasIngredient*, *hasBase*, *hasTopping*. Для цього слід виділити властивість *hasIngredient* та натиснути кнопку «Add sibling class» для створення нової властивості об'єкта з ім'ям *isIngredientOf* – «інгредієнтом в (начинці)» (це буде задано зворотне властивість для властивості *hasIngredient* «має інгредієнтом»).

20. Натисніть значок (+) поруч із кнопкою «inverse Of» на панелі «Опис властивості» (рис. 8). У діалоговому вікні вибрати властивість *hasIngredient* та натиснути "ОК". Властивості *hasIngredient* тепер повинні бути відображені в поданні «Inverse Of».

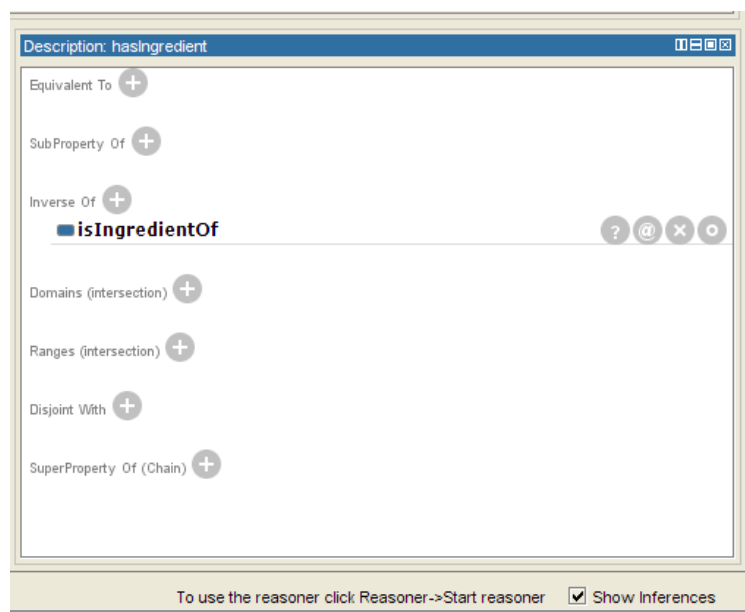


Рис. 8. Вікно «Опис властивості *hasIngredient*»



21. Аналогічним чином слід створити зворотні властивості для властивостей `hasBase`, `hasTopping`.

**Примітка!** Якщо властивість Транзитивна і вона пов'язує індивіда А та індивіда В, а також індивіда В пов'язує з індивідом С, то ми можемо вивести, що індивід А пов'язаний з індивідом С через цю властивість. Прикладом може стати властивість `hasAncestor`. Якщо індивід Метью став родоначальником Пітера, і Пітер має предка Вільяма, то ми можна зробити висновок що Метью став родоначальником Вільяма.

22. Нам потрібно зробити властивість `hasIngredient` транзитивною. Наприклад, якщо `PizzaTopping` має інгредієнт, то сама `Pizza` також повинна мати цей інгредієнт. Щоб задати зазначену характеристику властивостей, слід вибрати властивість `hasIngredient` в ієрархії властивостей на вкладці «Властивості об'єкта» та вибрати опцію «Transitive» на панелі характеристик.

23. Якщо властивість є транзитивною, то її зворотня властивість також має бути транзитивним. Тому потрібно вибрати властивість `isIngredientOf`, яке є зворотною `hasIngredient`, та встановити транзитивну позначку.

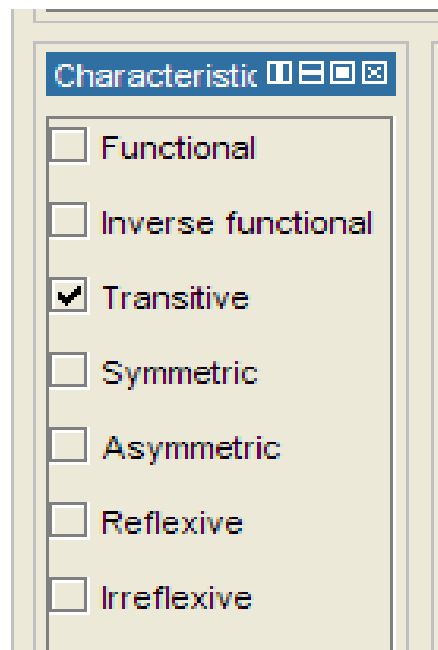
**Примітка!** Якщо властивість транзитивна, то вона не може бути функціональною.

24. Тепер треба зробити так, щоб клас `Pizza` мав тільки одну базу. Для цього слід зробити властивість `hasBase` функціональною. В цьому випадку вона може задавати тільки одне значення для обраного індивіда. Для цього слід вибрати властивість `hasBase` та встановити галочку «Functional» на панелі на панелі характеристик.

25. Для того, щоб вказати, що властивість `hasTopping` має діапазон `PizzaTopping`, слід використовувати панель діапазону (Description).

26. Для того, щоб вказати діапазон `hasTopping`, слід переконатися, що вибрана властивість `hasTopping` в ієрархії властивостей на вкладці «Властивості об'єкта», після чого натиснути кнопку «Додати» (+) поруч із «Ranges (Intersection)» на панелі «Опис властивості». З'явиться діалогове вікно, яке дозволяє вибирати класи з ієрархії класів онтології. Слід вибрати `PizzaTopping` і натисніть кнопку «ОК». Клас `PizzaTopping` тепер повинен бути відображений в списку діапазону.

27. Для того, щоб вказати клас `Pizza` як домен властивості `hasTopping`, слід виділити властивість `hasTopping` в ієрархії властивостей на вкладці «Властивості об'єкта», натиснути значок «Додати» (+) поруч із «Domains (Intersections)» на панелі «Опис властивостей». З'явиться діалогове вікно, яке дозволяє вибирати класи з ієрархії класів онтології. Слід вибрати `Pizza` і натисніть на кнопку ОК. Клас `Pizza` тепер повинен відображатися в списку доменів.



28. Після виконання пунктів 27, 28 властивість hasTopping повинна мати наступний домен і діапазон (рис. 9).

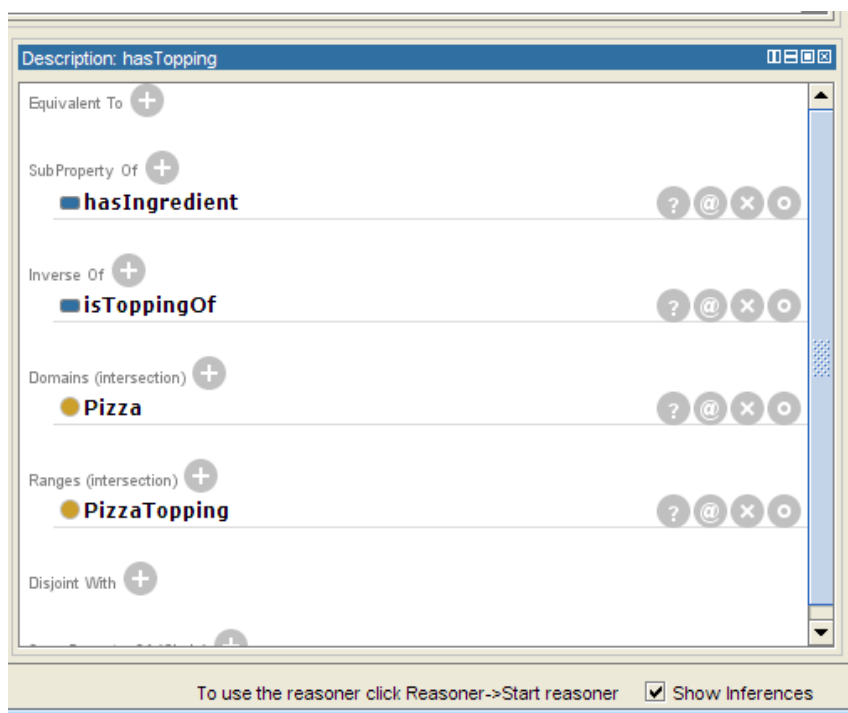


Рис. 9. Вікно опису властивості hasTopping

29. Аналогічні дії провести над зворотною властивістю isToppingOf за допомогою Tools – Reasoner – FaCT++ – Start reasoner. Всі дії виконуються автоматично.

30. Аналогічно п. 26, 27, для властивості hasBase вказати клас Pizza як домен, а клас Pizza Base як діапазон.

31. провести дії, аналогічну п. 29, над зворотною властивістю isBaseOf.

### Контрольні запитання

1. Які рішення покладені в основу Semantic Web?
2. Назвати причини появи Semantic Web?
3. Які технології складають основу Semantic Web?
4. Охарактеризувати рівні онтології з точки зору Semantic Web?
5. Дати визначення онтології?
6. Формальна модель онтології
7. Модель онтологічної системи
8. Загальні етапи розробки онтології
9. Деталізувати етапи розробки онтології по крокам
10. Підходи до розробки ієрархії класів



## ЛАБОРАТОРНА РОБОТА №7

**Тема:** СТВОРЕННЯ ОНТОЛОГІЇ ЗАСОБАМИ РЕДАКТОРУ PROTEGE

**Мета:** опанувати вміннями створювати властивості datatype у редакторі онтологій Protege

**Література:** [3], [5], [6], [10]

**Обладнання:** персональний комп'ютер із доступом до мережі Інтернет.

**Питання до контролю попередніх занять, обговорення, самостійного вивчення та осмислення навчального матеріалу:**

1. Що дозволяє зробити онтологія
2. Де використовуються онтології
3. Порівняти різні редактори онтологій

### Теоретичні відомості

Проект нового покоління глобальної знаковою середовища після Інтернету був названий його авторами Semantic Web. За думки авторів проекту, цей проект повинен усунути багато накопичилися (у тому числі перераховані вище) недоліки Інтернету. У його основу повинні бути покладені наступні рішення:

1. Нова знакова система - XML, що має в якості своєї основної можливості вільну систему базових знаків (тегів), що дозволяють користувачам більш гнучко і точно відобразити семантику своїх текстів, тим самим різко оптимізувавши прагматичні можливості програм їх обробки.

2. Побудова на цій новій знаковою основі XML-подібних мов знакових систем, виражають елементарні когнітивні структури - онтології.

3. Використання власних програмних систем на базі мобільних програм-агентів, що мають пошукові або будь-які інші приписи про їх бажаних чи можливих діях, виражені в онтологіях, для більш ефективного доступу до текстів в Мережі, а також комунікації між користувачами, тим самим різко підвищивши прагматичні можливості користувачів по відношенню до знакової глобальній системі нового покоління - Semantic Web.

У Semantic Web інформації надають чітке значення, що покращує допуск комп'ютерів і людей до інформації та забезпечує їх спільну роботу в тісній співпраці. Для того, щоб Web досягла свого повного потенціалу, вона повинна розвинутися в Semantic Web, забезпечуючи універсально доступну платформу, що дозволяє при цьому розподілених даних бути обробленими програмами так само як і людьми. Три технології складають основу такої платформи:

- програмні агенти - для того, щоб представляти реальні об'єкти і автоматизовану роздільну здатність завдання від імені їх власника;
- Онтології - для семантичного розширення інформації, якою обмінюються і обробляють Web- додатки;

- Web-сервіси як обчислювальні засоби, доступні через Інтернет.

Крім того, в стек технологій Semantic Web входять засоби логічного висновку, перевірки істинності, мови запитів до онтологій і т.д. Використання технологій Semantic Web при розробці програмних систем дозволяє істотно спростити проблему сумісності систем з суміжних областей і є першим кроком до побудови високоінтелектуальних компонентів і агентів. Центральним компонентом концепції є застосування онтологій, які розробляються і можуть бути використані при вирішенні різних завдань, у тому числі для спільного застосування людьми або програмними агентами, для можливості накопичення та повторного використання знань в предметній області, для створення моделей і програм, що оперують онтологіями, а НЕ жорстко заданими структурами даних, для аналізу знань в предметній області.

Онтології широко використовуються у всіх областях, що займаються обробкою даних на природній мові. У зв'язку з використанням онтологій в різних додатках виникла необхідність створення стандартизованих способів їх подання. Почалося розвиток різноманітних мов, які могли б застосовуватися повсюдно у всіх системах, найвідомішими є RDF і OWL. Виникло також велика кількість редакторів для створення, поповнення та зміни онтологій. Кожне з цих коштів зазвичай спрямоване на роботу з певним форматом даних і володіє своїми особливостями.

Повна (або комп'ютерна) онтологія є (формальним) виразом концептуальних знань про предметну область і за своєю значимістю порівнянна з базою знань знання-орієнтованої інформаційної системи (Зоіс), а її побудова є специфічною формою людської творчості. Творчий процес можна представити сукупністю операцій-процедур з судженнями, твердженнями, поняттями і відносинами між ними. А останні є фундаментом, основою для побудови складової частини наукової теорії - онтологічної бази знань у заданій предметній області. При цьому такі знання описуються в декларативною формі.

### ***Порядок виконання роботи***

1. Завантажити Protege 4, відкрити онтологію попередньої лабораторної роботи.

2. Для вирішення поставлених завдань необхідно використовувати властивості datatype для опису змісту калорій в піці. Потім слід використовувати деякі числові діапазони для широкої класифікації конкретних піц, як висококалорійних або низькокалорійних. Для це необхідно створити властивість типу даних hasCalorificContentValue, яке буде використовуватися для опису калорійності піци (рис. 1.). Інструменти для роботи з властивостями DataType аналогічні тим, яке ми використовували при роботі з властивостями об'єктів.

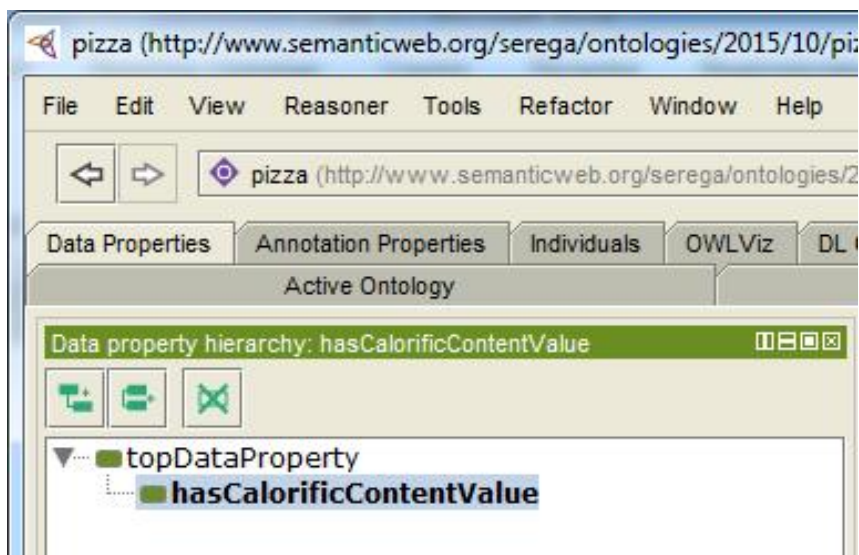


Рис. 1. Створення властивості hasCalorificContentValue

**Примітка!** Властивість datatype може використовуватися для індивіда, щодо конкретних значень даних, які можуть бути типізованими і нетипізованими.

3. Створити деяку безліч індивідів, які будуть володіти властивістю hasCalorificContentValue. Для цього слід вибрати закладку «Individuals», у вікні ієрархії класів створити підкласи класу «Pizza» (рис. 2), у вікні «Individuals» натиснути на кнопку «Додати індивіда» («Add Individual»), створити декілька екземплярів з назвою у будь-якому класі/підкласі Pizza (рис. 2).

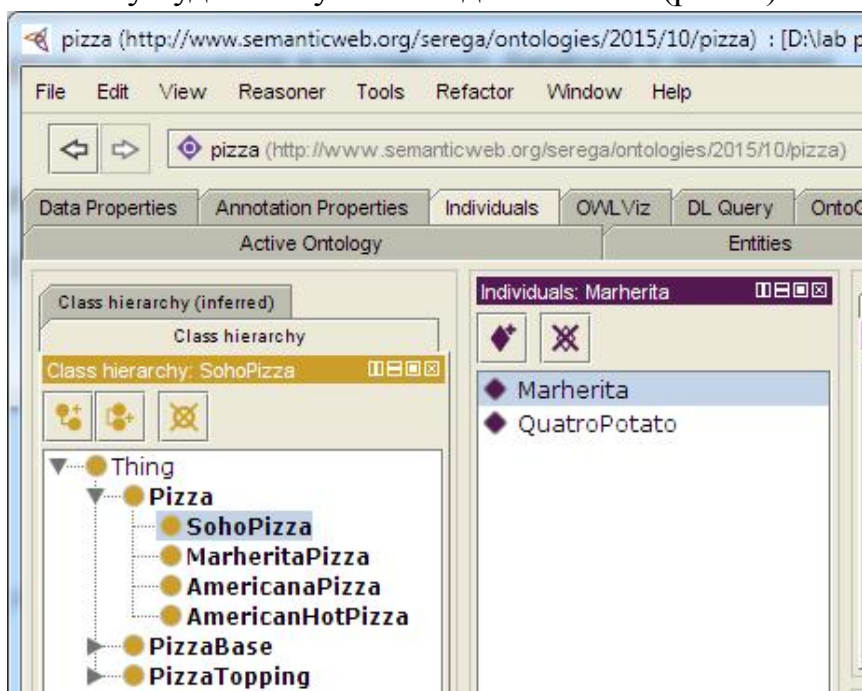
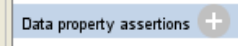


Рис. 2. Створення екземплярів

**Примітка!** В Protégé 4 індивіди мають спочатку при створенні тільки ім'я, а потім при описі у них з'являються властивості з тих властивостей, які розробник задає також окремо. При цьому набір властивостей може бути абсолютно індивідуальним.

4. Виділити екземпляр «Marherita», натиснути на кнопку , у діалоговому вікні, зображеному на рисунку 3, зв'язати індивіда Margherita властивістю hasCalorificContentValue зі значенням типу integer 263.

5. Додати аналогічно кілька індивідів з різними значеннями, включаючи екземпляр QuattroFormaggio з 723 калоріями.

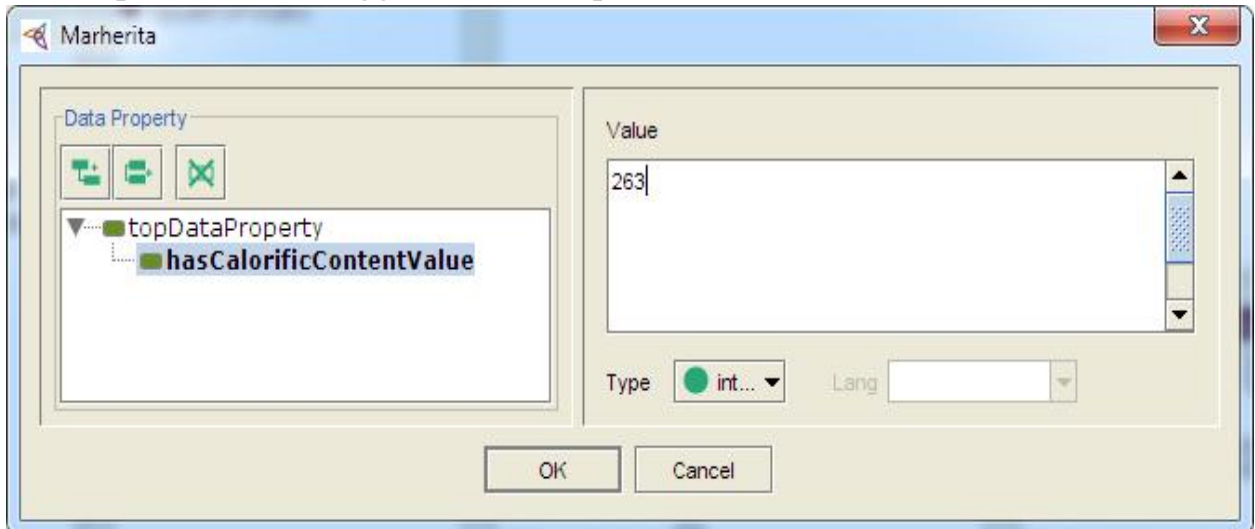


Рис. 3. Зв'язування індивіда Margherita властивістю hasCalorificContentValue

6. Створити обмеження DataType, яке оголосить, що всі піци мають енергетичну цінність (калорійність). Для цього слід вибрати закладку «Classes» або «Entities», вибрати Pizza і на панелі опису класу вибрати «+» у рядку «SubClass Of». Це викличе діалогове вікно редактора обмежень. Далі вибрати закладку «data restriction creator» і ввести «hasCalorificContentValue some integer», як показано на рисунку 4, потім натиснути «Ok».

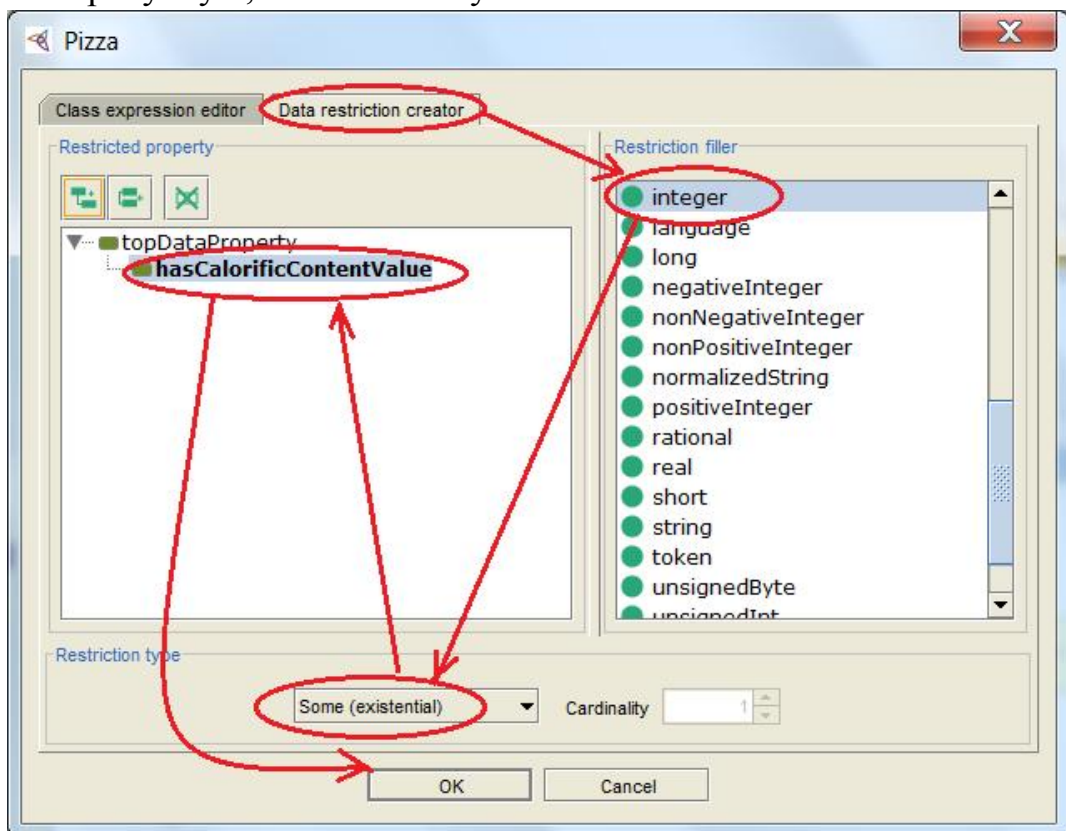


Рис. 4. Створення обмежень

7. Самостійно доробити обмеження у відповідності до рис. 5

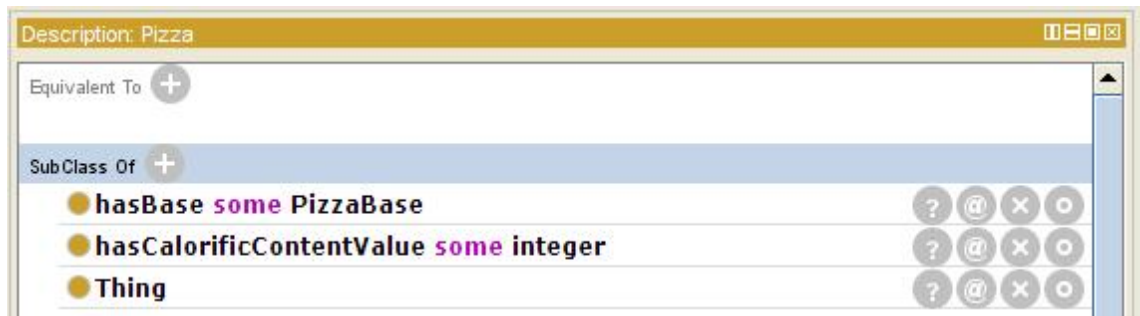


Рис. 5. Обмеження класу Pizza

**Примітка!** Крім використання визначеного набору типів даних можна також спеціалізувати використання `DataType` шляхом зазначення обмежень для можливих значень. Наприклад, це може бути застосовано для вказівки діапазону чисельних значень. За допомогою створеної властивості `datatype` можна створювати певні класи, які задають діапазон потрібних значень калорійності (`minInclusive` і `maxExclusive` фасети, які можуть застосовуватися для числових типів даних).

8. Створити клас `HighCaloriePizza` висококалорійних піц, що мають енергетичну цінність не менше ніж 400. Для цього слід вибрати закладку «Classes» або «Entities», вибрати клас `Pizza` в ієрархії класів, створити підклас з ім'ям `HighCaloriePizza`.

9. Вибрати `HighCaloriePizza` і на панелі опису класу вибрати «+» у рядку «SubClass Of», у діалоговому вікні вибрати закладку «data restriction creator» і ввести «`hasCalorificContentValue some integer`», потім натиснути "Ok". У вікні опису з'явиться обмеження, аналогічне тому, що ми задали для `Pizza`

6. Вибрати кнопку «Edit» у рядку обмеження, відкриється вікно, як на малюнку 5.4. Додайте фасет [`> = 400`] в рядок обмеження (рис. 6.)

7. Конвертіруйте `primitive` клас в `defined` (`Ctrl + d`). Опис класу буде виглядати, як на рисунку 7.

8. Створіть `LowCaloriePizza` так само, але визначте його як еквівалентний піці і має обмеження калорійності `hasCalorificContentValue` менш 400 (будь піца має енергетичну цінність [`<400`]). Зверніть увагу на те, що визначення не перекривається з `HighCaloriePizza`.



Рис. 6. Вікно введення фасети

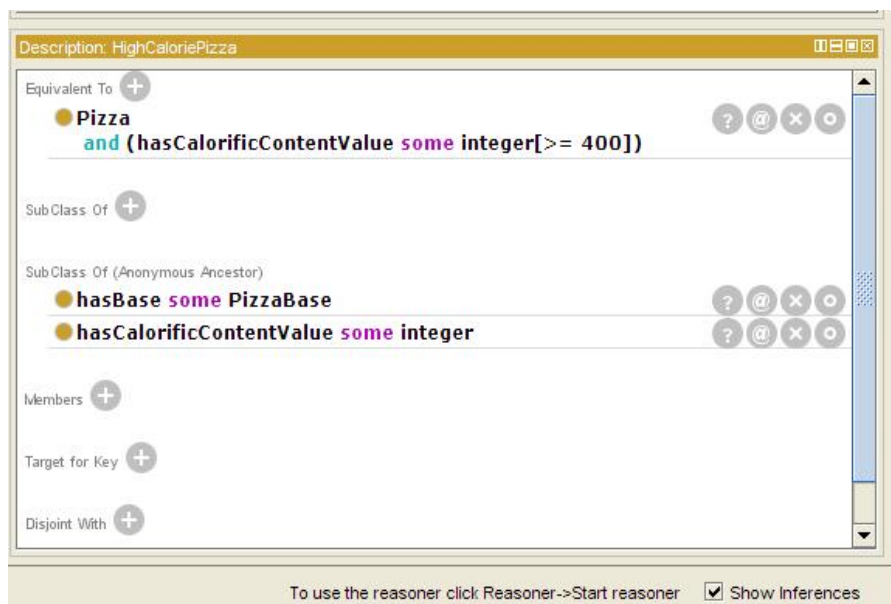


Рис 7. умова на значення підкласу HighCaloriePizza

**Примітка!** Тепер в онтології є дві категорії піци, які повинні охоплювати будь-якого індивіда, який має енергетичну цінність. Тепер потрібно перевірити, як здійсниться класифікація створених індивідів. Для цього слід використовувати реазонер (Reasoner) для виконання класифікації.

9. Потрібно класифікувати індивідів класу Pizza на основі їх обмежень по hasCalorificContentValue. Для цього слід вибрати команду «start reasoned» в меню Reasoner або натисніть «Synchronize reasoner», якщо він активований. Реазонер проведе класифікацію і покаже ієрархію виведених класів. Для перевірки слід вибрати HighCaloriePizza. Можна побачити результати міркувань, показані на панелі опису класу на жовтому фоні пунктирною кордоном.

10. Розділ «members» повинен включати екземпляр «QuattroFormaggio» і можливо інших індивідів, якщо їх ввести та описати, що вони мають значення 400



або більше калорій. Виберіть LowCaloriePizza. Розділ «members» має включати екземпляр «ExampleMargherita» і можливо інші індивіди, зазначені як мають значення енергетичної цінності менше 400 калорій (рис. 8).

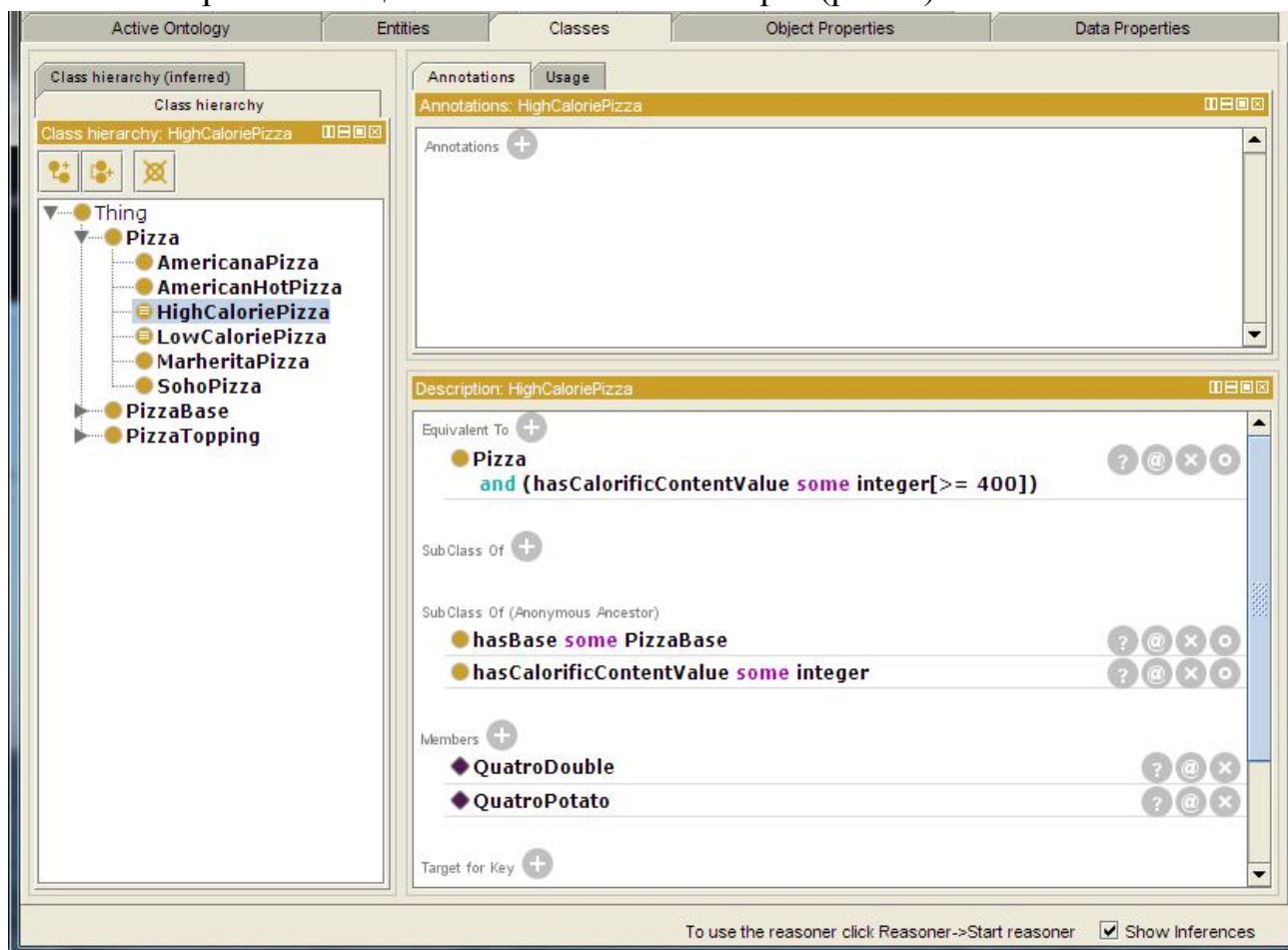


Рис. 8.Вкладка Classes

11. Якщо зробити властивість hasCalorificContentValue функціональною, то будь-яка піца можна мати тільки одне значення енергетичної цінності. Для цього слід перейти на закладку «Data Properties», вибрати пункт hasCalorificContentValue, у групі «Характеристики» відзначити галочкою характеристику «Functional».

### Контрольні запитання

1. Які рішення покладені в основу Semantic Web?
2. Назвати причини появи Semantic Web?
3. Які технології складають основу Semantic Web?
4. Охарактеризувати рівні онтології з точки зору Semantic Web?
5. Дати визначення онтології?
6. Формальна модель онтології
7. Модель онтологічної системи
8. Загальні етапи розробки онтології
9. Деталізувати етапи розробки онтології по крокам
10. Підходи до розробки ієрархії класів

## ТЕМИ ДЛЯ САМОСТІЙНОГО ОПРАЦЮВАННЯ

№ з/п	Назва теми
1	Основні поняття штучного інтелекту
2	Сфери застосування інтелектуальних систем
3	Класифікація інтелектуальних інформаційних систем
4	Вступ до експертних систем
5	Мультиагентні інтелектуальні системи
6	Вступ в проблему розпізнавання образів
7	Проектування баз знань
8	Логічні моделі
9	Продукційні моделі
10	Семантичні мережі
11	Фрейми
12	Нейронні мережі
13	Моделі та методи обробки нечітких знань
14	Онтології як моделі представлення знань
15	Protégé як інструментальний засіб створення онтологій



## ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

№ з/п	Назва теми
1	Визначення штучного інтелекту та знань Зробити огляд основних напрямків використання інтелектуальних систем Написати наукову статтю за темою штучного інтелекту
2	Етапи проектування інтелектуальних систем Визначити етапи проектування мультиагентних систем Визначити етапи проектування експертних систем
3	Проектування баз знань Визначити етапи створення баз знань за допомогою Prolog Визначити етапи створення баз знань за допомогою Python
4	Загальні поняття про мультиагентні інтелектуальні системи Зробити огляд напрямків застосування мультиагентних систем Зробити огляд поширених мультиагентних систем в освіті
5	Способи представлення знань в інтелектуальних системах Створити логічну модель представлення знань за темою Створити фреймову модель представлення знань за темою Створити продукційну модель представлення знань за темою
6	Онтології як моделі представлення знань Зробити огляд застосування онтологій Зробити огляд найбільш поширених онтологій
7	Protégé як інструментальний засіб створення онтологій Створити онтологію засобами Protégé «Навчальний процес» Створити онтологію засобами Protégé «Школа» Створити онтологію засобами Protégé «Підприємство»

## ТЕМИ РЕФЕРАТІВ

1. Застосування технологій штучного інтелекту в освіті.
2. Застосування технологій штучного інтелекту в економіці.
3. Застосування технологій штучного інтелекту в науці.
4. Застосування технологій штучного інтелекту на підприємствах.
5. Застосування експертних систем в освіті та науці.
6. Застосування експертних систем в економіці.
7. Застосування експертних систем на підприємствах.
8. Порівняльна характеристика найбільш поширених моделей представлення знань
9. Використання агентів в мультиагентних системах.
10. Сфери застосування мультиагентних систем.
11. Інструментальні засоби для розробки баз знань.
12. Можливості Ontolingua для розробки онтологій
13. Можливості DOE для розробки онтологій
14. Можливості OntoEdit для розробки онтологій
15. Використання Semantic Web як сучасний обмін знаннями

## Література

1. Балута В.С. Проблема онтологічної підготовки майбутніх спеціалістів з інформаційних технологій / В.С. Балута, В.В. Осадчий, С.В. Шаров, А.Я. Чураков // Проблеми інженерно-педагогічної освіти. – 2015. – № 47. – С. 16–22.
2. Козлов А.Н. Интеллектуальные информационные системы: учебник /А.Н. Козлов. – Пермь: Изд-во ФГБОУ ВПО Пермская ГСХА, 2013.– 278 с.
3. Константинова Н.С. Онтологии как системы хранения знаний: [Электронный ресурс] / Н.С. Константинова, О.А. Митрофанова. – Режим доступа: [http://www.uran.donetsk.ua/~masters/2012/iii/orlova/library/building\\_ontologies.pdf](http://www.uran.donetsk.ua/~masters/2012/iii/orlova/library/building_ontologies.pdf).
4. Морозов М.Н. Курс лекций по дисциплине “Системы искусственного интеллекта” : [электронный ресурс] / М.Н Морозов. – режим доступа : <http://khpi-iip.mipk.kharkiv.edu/library/ai/conspai/index.html>.
5. Невмержицький О.В. Аналіз сучасних моделей, орієнтованих на знання, та методів прийняття рішень / О.В. Невмержицький // Інформаційні технології проектування. – №13. – 2013. – С. 119–125.
6. Петренко Н. Компьютерные онтологии и онтолого-управляемая архитектура информационных систем / Николай Петренко // Information Models of Knowledge. – Kiev, Ukraine – Sofia, Bulgaria, 2010. – С. 86–92.
7. Ревко П.С. Искусственные интеллектуальные системы в повседневной жизни человека / П.С. Ревко // Известия Южного федерального университета. Технические науки. – Вып. № 9-2. – 2006. – С. 109–110.
8. Рогушина Ю.В. Технологии Semantic Web и их использование при разработке интеллектуальных приложений: [Электронный ресурс] / Ю.В. Рогушина, А.Я.Гладун. – Режим доступа: <http://dspace.nbu.gov.ua/bitstream/handle/123456789/1507/49%20-%20Rogushina.pdf?sequence=1>.
9. Сахнюк П.А. Интеллектуальные системы и технологии: Учебное пособие / П.А. Сахнюк. – Ставрополь: Агрус 2012. – 228 с.
10. Смагин А.А. Интеллектуальные информационные системы: учеб. пособ. / А.А. Смагин, С.В. Липатова, А.С. Мельниченко. – Ульяновск : УлГУ, 2010. – 136 с.
11. Смирнов, А.В. Многоагентные системы поддержки принятия решений для предприятий малого и среднего бизнеса / А.В. Смирнов, М.П. Пашкин, И.О. Рахманова // Информационные технологии и вычислительные системы. – 1988. – № 1.– С. 23–25.
12. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник / С.О. Субботін. – Запоріжжя: ЗНТУ, 2008. – 341 с.
13. Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям:

- философия, психология, информатика / В.Б. Тарасов. – М.: Эдиториал УРСС, 2002.
14. Хабаров С.П. Экспертные системы: [электронный ресурс] / С.П. Хабаров. – режим доступа : [http://firm.trade.spb.ru/serp/main\\_es.htm](http://firm.trade.spb.ru/serp/main_es.htm).
  15. Шаров С.В. Сучасний стан розвитку інтелектуальних інформаційних систем / С.В. Шаров // Вісник Чернігівського національного педагогічного університету імені Т.Г.Шевченка. – Вип. 130. – Серія: Педагогічні науки. – Чернігів: ЧНПУ, 2015. – С. 111 – 114.
  16. Шаров С.В. Вибір моделі представлення знань у системі ІСІКС / С.В. Шаров, Д.В. Лубко, В.В.Осадчий // Системи обробки інформації. – 2015. – №. 11. – С. 108-111.
  17. Noy Natalya F. Ontology Development 101: A Guide to Creating Your First Ontology : [Електронний ресурс] // Natalya F. Noy and Deborah L. McGuinness. – Режим доступу: [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf).

Шаров Сергій Володимирович

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ  
ДО ЛАБОРАТОРНИХ РОБІТ З КУРСУ  
«РОЗРОБКА ТА СУПРОВІД БАЗ ЗНАНЬ»

Методичні рекомендації

Видавництво: РВЦ МДПУ  
вул. Леніна, 20, м. Мелітополь, Запорізької обл., 72312  
(0619)440363