Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №1**

**По системам искусственного интеллекта**

Выполнил:

студент 3 курса

Батманов Даниил Евгеньевич

Группа: Р3307

Принял:

Кугаевских Александр Владимирович

Отчёт принят «\_\_»\_\_\_\_\_2024 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2024

# Задание

**Часть 1. Создание базы знаний и выполнение запросов в Prolog**

Требуется создать базу знаний в языке программирования Prolog и реализовать набор запросов, используя эту базу знаний. Задача направлена на развитие навыков работы с фактами, предикатами, и правилами в логическом программировании.

**Задание:**

* **Создание базы знаний:**

Создайте базу знаний. База знаний должна включать в себя **не менее 20 фактов с одним аргументом, 10-15 фактов с двумя аргументам, которые дополняют и показывают связь с другими фактами и 5-7 правил.** Факты могут описывать объекты, их свойства и отношения между ними. Факты 2 и более аргументами могут описывать различные атрибуты объектов, а правила - логические законы и выводы, которые можно сделать на основе фактов и предикатов.

* **Выполнение запросов:**

Напишите несколько запросов для БЗ. Запросы **должны быть разной сложности** и включать в себя:

* + Простые запросы к базе знаний для поиска фактов.
  + Запросы, использующие логические операторы (**и, или, не**) для формулирования сложных условий (или использовать логические операторы в правилах).
  + Запросы, использующие переменные для поиска объектов с определенными характеристиками.
  + Запросы, которые требуют выполнения правил для получения результата.
* **Документация:**

В коде должны быть комментарии описания фактов, предикатов и правил.

**Критерии оценки**

* Корректность базы знаний и выполненных запросов.
* Сложность и разнообразие запросов.
* Качество документации и комментариев к коду.

**Тематики БЗ**

Любая, связанная с играми. Например,

* Видеоигры
* Правила настольных игр
* Профили игроков
* Игровые персонажи
* История игры
* Игровые механики

**Часть 2.  Создание онтологии в Protege**

Целью этой лабораторной работы является знакомство со средой разработки онтологий Protege и перевод базы знаний, созданной в предыдущей лабораторной работе в онтологическую форму в Protege.

**Задание**

Преобразовать факты и отношения из Prolog в концепты и свойства в онтологии. Описать классы и свойства в онтологии, которые соответствуют объектам и отношениям из базы знаний. Например, если у были классы "Человек" и "Машина" и свойство "возраст", создайте аналогичные классы и свойства в онтологии в Protege.

**Критерии оценки**

* Корректное создание онтологии в Protege на основе базы знаний в Prolog.
* Качество перевода фактов, предикатов и отношений из Prolog в онтологию.
* Определение классов, свойств и иерархии классов в Protege.
* Тестирование онтологии и демонстрация ее функциональности (визуализация и проверка запросов).

# Ход работы

# Часть 1. Создание базы знаний и выполнение запросов в Prolog

***Создание базы знаний:***

% Факты с одним аргументом: герои, роли, атрибуты

% Герои

hero(antimage).

hero(axe).

hero(crystal\_maiden).

hero(pudge).

hero(invoker).

hero(sven).

hero(windranger).

hero(lina).

hero(earthshaker).

hero(lion).

hero(spectre).

hero(faceless\_void).

hero(rubick).

hero(naga\_siren).

hero(batrider).

hero(drow\_ranger).

hero(juggernaut).

hero(zeus).

hero(slark).

hero(lich).

hero(tiny).

% Роли

role(carry).

role(support).

role(disabler).

role(nuker).

role(initator).

role(durable).

% Атрибуты (силовые, ловкие, интеллект)

attribute(strength).

attribute(agility).

attribute(intelligence).

% Факты с двумя аргументами: роли и атрибуты героев, способности героев

% Связь героя с его основной ролью

hero\_role(antimage, carry).

hero\_role(axe, durable).

hero\_role(crystal\_maiden, support).

hero\_role(pudge, durable).

hero\_role(invoker, nuker).

hero\_role(sven, carry).

hero\_role(windranger, carry).

hero\_role(lina, nuker).

hero\_role(earthshaker, initator).

hero\_role(lion, disabler).

hero\_role(spectre, carry).

hero\_role(faceless\_void, carry).

hero\_role(rubick, support).

hero\_role(naga\_siren, carry).

hero\_role(batrider, initiator).

hero\_role(drow\_ranger, carry).

hero\_role(juggernaut, carry).

hero\_role(zeus, nuker).

hero\_role(slark, carry).

hero\_role(lich, support).

hero\_role(tiny, initiator).

% Связь героя с атрибутом

hero\_attribute(antimage, agility).

hero\_attribute(axe, strength).

hero\_attribute(crystal\_maiden, intelligence).

hero\_attribute(pudge, strength).

hero\_attribute(invoker, intelligence).

hero\_attribute(sven, strength).

hero\_attribute(windranger, agility).

hero\_attribute(lina, intelligence).

hero\_attribute(earthshaker, strength).

hero\_attribute(lion, intelligence).

hero\_attribute(spectre, agility).

hero\_attribute(faceless\_void, agility).

hero\_attribute(rubick, intelligence).

hero\_attribute(naga\_siren, agility).

hero\_attribute(batrider, intelligence).

hero\_attribute(drow\_ranger, agility).

hero\_attribute(juggernaut, agility).

hero\_attribute(zeus, intelligence).

hero\_attribute(slark, agility).

hero\_attribute(lich, intelligence).

hero\_attribute(tiny, strength).

% Способности героев

hero\_ability(antimage, mana\_void).

hero\_ability(axe, berserkers\_call).

hero\_ability(crystal\_maiden, freezing\_field).

hero\_ability(pudge, rot).

hero\_ability(invoker, sun\_strike).

hero\_ability(sven, god\_strength).

hero\_ability(windranger, focus\_fire).

hero\_ability(lina, laguna\_blade).

hero\_ability(earthshaker, echo\_slam).

hero\_ability(lion, finger\_of\_death).

hero\_ability(spectre, haunt).

hero\_ability(faceless\_void, chronosphere).

hero\_ability(rubick, spell\_steal).

hero\_ability(naga\_siren, song\_of\_the\_siren).

hero\_ability(batrider, flaming\_lasso).

hero\_ability(drow\_ranger, marksmanship).

hero\_ability(juggernaut, omnislash).

hero\_ability(zeus, thunder\_god\_wrath).

hero\_ability(slark, shadow\_dance).

hero\_ability(lich, chain\_frost).

hero\_ability(tiny, avalanche).

% Правила

% Правило: герой считается ловким, если его атрибут - agility

agile\_hero(Hero) :- hero(Hero), hero\_attribute(Hero, agility).

% Правило: герой является керри, если его основная роль - carry

is\_carry(Hero) :- hero(Hero), hero\_role(Hero, carry).

% Правило: герой является магом, если его основной атрибут - intelligence

is\_mage(Hero) :- hero(Hero), hero\_attribute(Hero, intelligence).

% Правило: герой является танком, если его атрибут - strength и роль - durable

is\_tank(Hero) :- hero(Hero), hero\_attribute(Hero, strength), hero\_role(Hero, durable).

% Правило: герой имеет ультимативную способность, если у него есть конкретное умение

has\_ultimate(Hero, Ability) :- hero\_ability(Hero, Ability).

% Правило: герой считается сильным в командных боях, если он инициатор

team\_fighter(Hero) :- hero\_role(Hero, initiator).

***Выполнение запросов:***

% Простой запрос: проверить, является ли Axe героем.

?- hero(axe).

% Запрос с переменной: найти всех героев.

?- hero(Hero).

% Запрос с двумя аргументами: узнать роль Pudge.

?- hero\_role(pudge, Role).

% Запрос с переменными в обоих аргументах: найти всех героев, выполняющих роль carry.

?- hero\_role(Hero, carry).

% Запрос с логическим И (запятая): найти роль и атрибут для Axe.

?- hero\_role(axe, Role), hero\_attribute(axe, strength).

% Запрос с логическим ИЛИ (точка с запятой): найти всех героев, которые являются либо carry, либо support.

?- hero\_role(Hero, carry); hero\_role(Hero, support).

% Запрос с логическим НЕ (оператор \+): найти всех героев, которые не являются carry.

?- hero(Hero), \+ hero\_role(Hero, carry).

% Запрос на использование правила is\_mage: найти всех героев с атрибутом intelligence (магами).

?- is\_mage(Hero).

% Запрос на использование правила is\_tank: найти всех героев, которые являются танками (атрибут strength и роль durable).

?- is\_tank(Hero).

# Часть 2. Создание онтологии в Protege

<?xml version="1.0"?>

<rdf:RDF xmlns="http://example.org/dota.owl#"

xml:base="http://example.org/dota.owl"

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

xmlns:owl="http://www.w3.org/2002/07/owl#"

xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

xmlns:protege="http://protege.stanford.edu/">

<!-- Определение онтологии -->

<owl:Ontology rdf:about="http://example.org/dota.owl">

<rdfs:label>Dota 2 Heroes Ontology</rdfs:label>

</owl:Ontology>

<!-- Классы -->

<owl:Class rdf:about="#Hero">

<rdfs:label>Hero</rdfs:label>

<rdfs:comment>A class representing all Dota 2 heroes.</rdfs:comment>

</owl:Class>

<owl:Class rdf:about="#Role">

<rdfs:label>Role</rdfs:label>

<rdfs:comment>A class representing the roles of heroes.</rdfs:comment>

</owl:Class>

<owl:Class rdf:about="#Attribute">

<rdfs:label>Attribute</rdfs:label>

<rdfs:comment>A class representing the primary attributes of heroes.</rdfs:comment>

</owl:Class>

<owl:Class rdf:about="#Ability">

<rdfs:label>Ability</rdfs:label>

<rdfs:comment>A class representing hero abilities.</rdfs:comment>

</owl:Class>

<!-- Свойства -->

<owl:ObjectProperty rdf:about="#hasRole">

<rdfs:domain rdf:resource="#Hero"/>

<rdfs:range rdf:resource="#Role"/>

<rdfs:label>hasRole</rdfs:label>

<rdfs:comment>Relates a hero to their primary role.</rdfs:comment>

</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasAttribute">

<rdfs:domain rdf:resource="#Hero"/>

<rdfs:range rdf:resource="#Attribute"/>

<rdfs:label>hasAttribute</rdfs:label>

<rdfs:comment>Relates a hero to their primary attribute.</rdfs:comment>

</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasAbility">

<rdfs:domain rdf:resource="#Hero"/>

<rdfs:range rdf:resource="#Ability"/>

<rdfs:label>hasAbility</rdfs:label>

<rdfs:comment>Relates a hero to their abilities.</rdfs:comment>

</owl:ObjectProperty>

<!-- Экземпляры ролей -->

<owl:NamedIndividual rdf:about="#carry">

<rdf:type rdf:resource="#Role"/>

<rdfs:label>Carry</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#support">

<rdf:type rdf:resource="#Role"/>

<rdfs:label>Support</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#disabler">

<rdf:type rdf:resource="#Role"/>

<rdfs:label>Disabler</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#nuker">

<rdf:type rdf:resource="#Role"/>

<rdfs:label>Nuker</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#initiator">

<rdf:type rdf:resource="#Role"/>

<rdfs:label>Initiator</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#durable">

<rdf:type rdf:resource="#Role"/>

<rdfs:label>Durable</rdfs:label>

</owl:NamedIndividual>

<!-- Экземпляры атрибутов -->

<owl:NamedIndividual rdf:about="#strength">

<rdf:type rdf:resource="#Attribute"/>

<rdfs:label>Strength</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#agility">

<rdf:type rdf:resource="#Attribute"/>

<rdfs:label>Agility</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#intelligence">

<rdf:type rdf:resource="#Attribute"/>

<rdfs:label>Intelligence</rdfs:label>

</owl:NamedIndividual>

<!-- Экземпляры способностей -->

<owl:NamedIndividual rdf:about="#mana\_void">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Mana Void</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#berserkers\_call">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Berserker's Call</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#freezing\_field">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Freezing Field</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#rot">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Rot</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#sun\_strike">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Sun Strike</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#god\_strength">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>God Strength</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#focus\_fire">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Focus Fire</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#laguna\_blade">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Laguna Blade</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#echo\_slam">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Echo Slam</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#finger\_of\_death">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Finger of Death</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#haunt">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Haunt</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#chronosphere">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Chronosphere</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#spell\_steal">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Spell Steal</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#song\_of\_the\_siren">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Song of the Siren</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#flaming\_lasso">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Flaming Lasso</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#marksmanship">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Marksmanship</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#omnislash">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Omnislash</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#thunder\_gods\_wrath">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Thunder God's Wrath</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#shadow\_dance">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Shadow Dance</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#chain\_frost">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Chain Frost</rdfs:label>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#avalanche">

<rdf:type rdf:resource="#Ability"/>

<rdfs:label>Avalanche</rdfs:label>

</owl:NamedIndividual>

<!-- Экземпляры героев -->

<owl:NamedIndividual rdf:about="#antimage">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Anti-Mage</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#mana\_void"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#axe">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Axe</rdfs:label>

<hasRole rdf:resource="#durable"/>

<hasAttribute rdf:resource="#strength"/>

<hasAbility rdf:resource="#berserkers\_call"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#crystal\_maiden">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Crystal Maiden</rdfs:label>

<hasRole rdf:resource="#support"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#freezing\_field"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#pudge">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Pudge</rdfs:label>

<hasRole rdf:resource="#durable"/>

<hasAttribute rdf:resource="#strength"/>

<hasAbility rdf:resource="#rot"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#invoker">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Invoker</rdfs:label>

<hasRole rdf:resource="#nuker"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#sun\_strike"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#sven">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Sven</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#strength"/>

<hasAbility rdf:resource="#god\_strength"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#windranger">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Windranger</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#focus\_fire"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#lina">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Lina</rdfs:label>

<hasRole rdf:resource="#nuker"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#laguna\_blade"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#earthshaker">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Earthshaker</rdfs:label>

<hasRole rdf:resource="#initiator"/>

<hasAttribute rdf:resource="#strength"/>

<hasAbility rdf:resource="#echo\_slam"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#lion">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Lion</rdfs:label>

<hasRole rdf:resource="#disabler"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#finger\_of\_death"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#spectre">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Spectre</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#haunt"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#faceless\_void">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Faceless Void</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#chronosphere"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#rubick">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Rubick</rdfs:label>

<hasRole rdf:resource="#support"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#spell\_steal"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#naga\_siren">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Naga Siren</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#song\_of\_the\_siren"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#batrider">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Batrider</rdfs:label>

<hasRole rdf:resource="#initiator"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#flaming\_lasso"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#drow\_ranger">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Drow Ranger</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#marksmanship"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#juggernaut">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Juggernaut</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#omnislash"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#zeus">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Zeus</rdfs:label>

<hasRole rdf:resource="#nuker"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#thunder\_gods\_wrath"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#slark">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Slark</rdfs:label>

<hasRole rdf:resource="#carry"/>

<hasAttribute rdf:resource="#agility"/>

<hasAbility rdf:resource="#shadow\_dance"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#lich">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Lich</rdfs:label>

<hasRole rdf:resource="#support"/>

<hasAttribute rdf:resource="#intelligence"/>

<hasAbility rdf:resource="#chain\_frost"/>

</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#tiny">

<rdf:type rdf:resource="#Hero"/>

<rdfs:label>Tiny</rdfs:label>

<hasRole rdf:resource="#initiator"/>

<hasAttribute rdf:resource="#strength"/>

<hasAbility rdf:resource="#avalanche"/>

</owl:NamedIndividual>

</rdf:RDF>  
  
Заключение

Мне удалось создать базу знаний в языке программирования Prolog и реализовать набор запросов, используя эту базу знаний. Преобразовать факты и отношения из Prolog в концепты и свойства в онтологии. Описать классы и свойства в онтологии, которые соответствуют объектам и отношениям из базы знаний.