**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Методические указания
по выполнению практических работ
по дисциплине «Искусственный интеллект в профессиональной сфере»
для студентов направления 40.03.01 Юриспруденция,
профиль Правозащитная деятельность

Ставрополь, 2026 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3
Практическая работа 1. Анализ современных программных средств с применением ИИ ... 4
Практическая работа 2. Формализация знаний. Использование семантических сетей для представления знаний 5
Практическая работа 3. Представление знаний. Продукционная модель 9
Практическая работа 4. Изучение принципов создания продукционных баз знаний 12
Практическая работа 5. Построение онтологической модели в Protégé. Создание классов 18
Практическая работа 6. Построение онтологической модели в Protégé. Определение эквивалентных, инверсных, транзитивных ролей (свойств) и создание цепочек ролей (свойств) 34
Практическая работа 7. Разработка учебной экспертной системы 43

ВВЕДЕНИЕ

Цель освоения дисциплины является овладение студентами основными методами теории интеллектуальных систем, приобретение навыков по использованию интеллектуальных систем, изучение основных методов представления знаний и моделирования рассуждений.

Задачи освоения дисциплины – помочь студентам овладеть навыками и знаниями в области искусственного интеллекта

Наименование практических занятий

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов	Из них практическая подготовка, часов
5 семестр			
1.	Анализ современного состояния области искусственного интеллекта	4	
2.	Формализация знаний Использование семантических сетей для представления знаний	4	
3.	Представление знаний. Продукционная модель	4	
4.	Изучение принципов создания продукционных баз знаний	6	
5.	Построение онтологической модели в Protégé. Создание классов	6	
6.	Построение онтологической модели в Protégé. Определение эквивалентных, инверсных, транзитивных ролей (свойств) и создание цепочек ролей (свойств)	6	
7.	Разработка учебной экспертной системы	6	2
	Итого за 5 семестр	**36**	**2**
	Итого	**36**	**2**

СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ РАБОТ

Практическая работа 1.
Анализ современных программных средств с применением ИИ

Цель занятия – сформировать представление об современных программных средств с применением ИИ.

Задачи занятия:

- изучить литературные источники и материалы из сети Интернет, связанные с нормативно-правовыми основами развития искусственного интеллекта в России и мире;
- проанализировать примеры применения технологий ИИ в образовательной сфере.

Задания для выполнения и методические рекомендации:

Задание 1.

Изучив литературные источники и материалы из сети Интернет, заполните таблицу «Нормативно-правовые основы развития искусственного интеллекта в России»:

№ п/п	Документ	Дата принятия	Ссылка на документ	Сроки реализации (если имеются)	Цели и задачи	Ответственные в реализации	Основные пункты
1.	«Национальная стратегия развития искусственного интеллекта на период до 2030 года»						
2.	…						

Задание 2.

Подготовьте сообщение (текст и презентацию) о любом из перечисленных в таблице Задания 1 документов. Будьте готовы рассказать его группе.

Задание 3.

Изучив литературные источники и материалы из сети Интернет, заполните следующую таблицу «Нормативно-правовые основы развития искусственного интеллекта в мире» (привести не менее 7 стран):

№ п/п	Страна	Документ	Дата принятия	Сроки реализации (если имеются)	Цели и задачи
1.		1.1.			
		1.2.			
		…			
2.	…				

Задание 4.

Изучив литературные источники и материалы из сети Интернет, заполните следующую таблицу «Примеры технологий ИИ в моей профессиональной сфере» (привести 5- 7 технологий):

№	Название	Разработчи	Время	Описание	Примеры

п/п	технологии	к технологии	появления технологии	возможностей технологии	использования технологии в профессиональной сфере
1.					
2.					

Задание 5.

Подготовьте сообщение (текст и презентацию) о любом из перечисленных в таблице Задания 4 технологий. Будьте готовы рассказать его группе.

Сделайте **вывод** о проделанной работе.

Практическая работа 2.
Формализация знаний. Использование семантических сетей для представления знаний

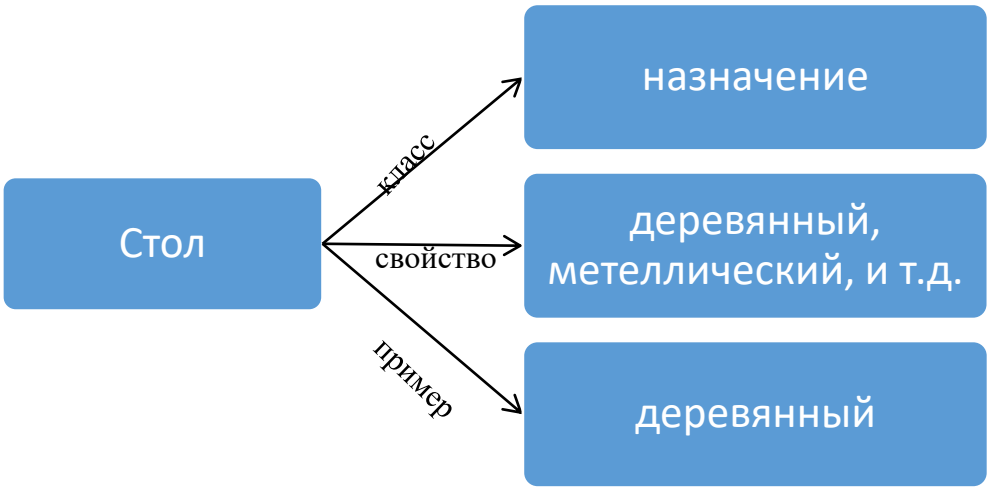
Цель занятия – формирование навыков построения семантических сетей и фреймовых моделей.

Задачи занятия:

- изучить различные способы представления знаний;
- научиться строить семантическую сеть и фреймовую модель представления знаний.

Задания для выполнения и методические рекомендации:

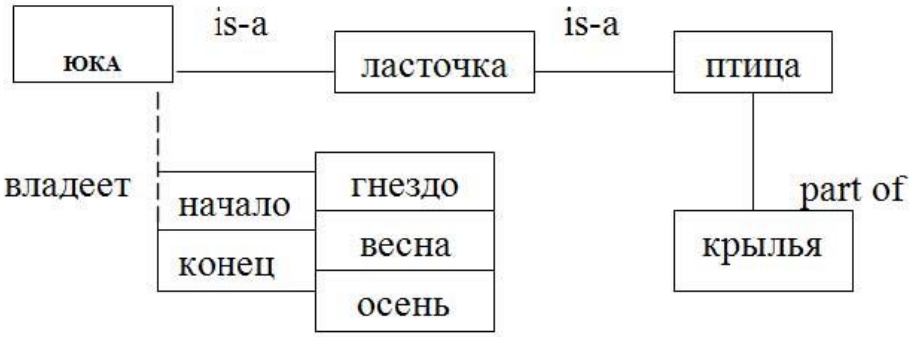
Семантическая сеть – это один из способов представления знаний. Изначально семантическая сеть была задумана как модель представления долговременной памяти в психологии, но впоследствии стала одним из способов представления знаний в ЭС.
Семантика – означает общие отношения между символами и объектами из этих символов.



Простейший образец семантической сети

Вершины – это объекты, дуги – это отношения. Семантическая модель не раскрывает сама по себе каким образом осуществляется представление знаний. Поэтому семантическая сеть рассматривается как метод представления знаний и структурирования знаний. При

расширении семантической сети в ней возникают другие отношения:
IS –A (принадлежит) и PART OF (является частью) отношение:
целое → часть.
Ласточка IS – A птица, «нос» PART OF «тело». Например:



Расширение семантической сети
Могут быть и другие отношения: владеет. Тогда семантическая сеть расширяется иерархически (вершина имеет две ветви). Кроме того, можно расширить сеть и другим отношением:
период → «весна – лето».
Получается иерархическую структуру понятия ЮКА можно разбить на подсхемы. Большой проблемой для семантических сетей является то, что результат вывода не гарантирует достоверности, так как вывод есть просто наследование свойств ветви is-a.
Для отображения иерархических отношений между объектами и введения единой семантики в семантические сети было предложено использовать процедурные сети. Сеть строится на основе класса (понятия); вершины, дуги и процедуры представлены как объекты.

Задание 1
Используя соответствующие дуги построить семантическую сеть, касающуюся:
1. географии какого-либо региона. Дуги: государство, страна, континент, широта.
2. диагностики глазных заболеваний. Дуги: категории болезней, патофизиологическое состояние, наблюдения, симптомы.
3. распознавания химических структур. Дуги: формула вещества, свойства вещества, область применения, меры предосторожности.
4. процедуры поиска полезных ископаемых. Дуги: наименование ископаемого, расположение месторождения, глубина залегания, методы добычи.
5. судебной процедуры. Дуги: юридическое лицо, событие, меры воздействия, способы расследования.
6. распределения продуктов по магазинам. Дуги: источник снабжения, наименование продукта, способ транспортировки, конечный пункт транспортировки.
7. определения принадлежности животного к определенному виду, типу, семейству. Дуги: место обитания, строение, особенности поведения, вид питания.
8. классификации пищевых продуктов. Дуги: наименование продукта, составляющие части, способ приготовления, срок хранения.
9. распознавания типа компьютера. Дуги: страна изготовитель, стандартная конфигурация, область применения, используемое программное обеспечение.
10. иерархической структуры БД. Дуги: система, состояние, назначение, взаимодействие составляющих.

Использование фреймов для представления знаний

Фреймы - один из распространенных формализмов представления знаний в ЭС. Фрейм можно представить себе как структуру, состоящую из набора ячеек - слотов. Каждый слот состоит из имени и ассоциируемых с ним значений. Значения могут представлять собой данные, процедуры, ссылки на другие фреймы или быть пустыми. Такое построение оказывается очень удобным для моделирования аналогий, описания областей с родовидовыми связями понятий и т.п.

Любой фрейм состоит из некоторых составляющих, имена и содержание которых описано ниже:

1. Имя фрейма. Это идентификатор, присваиваемый фрейму, фрейм должен иметь имя уникальное в данной фреймовой системе.
2. Имя слота. Это идентификатор, присваиваемый слоту; слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой смысловой нагрузки и является лишь идентификатором данного слота.
3. Указатели наследования. Эти указатели касаются только фреймовых систем иерархического типа, основанные на отношениях “абстрактное-конкретное”, они показывают, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с такими же именами во фрейме нижнего уровня. Типичные указатели наследования Unique (U: - уникальный), Same (S: такой же), Range (R: установление границ), Override (O: игнорировать) и т.п. U показывает, что фрейм может иметь слоты с разными значениями: S - все слоты должны иметь одинаковые значения, R - значение слотов фрейма нижнего уровня должны находиться в пределах, указанных значениями слотов фрейма верхнего уровня, O - при отсутствии указания значение слота фрейма верхнего уровня становится значением слота фрейма нижнего уровня, но в случае определения нового значения слотов фреймов нижних уровней указываются в качестве значений слотов.
4. Указание типа данных. указывается, что слот имеет численное значение, либо служит указателем другого фрейма. К типам данных относятся:

FRAME (указатель), INTEGER (целый), REAL (действительный), BOOL (булев), LISP (присоединенная процедура), TEXT (текст), LIST (список), TABLE (таблица), EXPRESSION (выражение) и др.

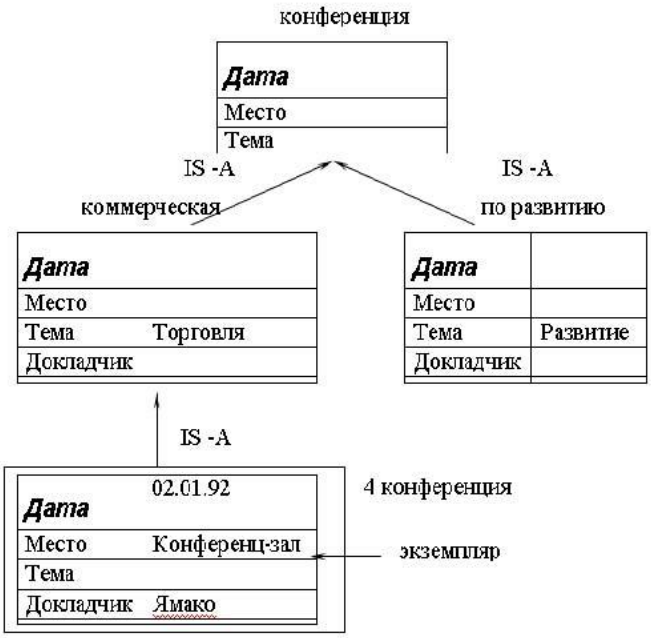
5. Значение слота. Пункт ввода значения слота. Значение слота должно совпадать с указанным типом данных этого слота, кроме того должно выполняться условие наследования.
6. Демон. Здесь дается определение демонов типа IF-NEEDED, IF-ADDED, IF-REMOVED и т.д. Демоном называется процедура, автоматически запускаемая при выполнении некоторого условия. демоны запускаются при

обращении к соответствующему слоту. Кроме того, демон является разновидностью присоединенной процедуры.

7. Присоединенная процедура. В качестве значения слота можно использовать программу процедурного типа. Когда мы говорим, что в моделях представления знаний фреймами объединяются процедурные и декларативные знания, то считаем демоны и присоединенные процедуры процедурными знаниями.

Особенностью иерархической структуры является то, что информация об атрибутах фрейма на верхнем уровне совместно используется всеми фреймами нижних уровней, связанных с ним.

Например: Фреймовое представление конференции.



Пример фреймовой модели

Иерархические фреймовые структуры базируются на отношениях IS – A между фреймами, описывающими некоторую конференцию. Все фреймы должны содержать информацию о ДАТЕ, МЕСТЕ, НАЗВАНИИ ТЕМЫ, ДОКЛАДЧИКЕ. Таким образом, на самом верхнем уровне определен фрейм КОНФЕРЕНЦИЯ.
Конференции разделяются на коммерческие и по развитию. Они составляют дочерние фреймы. В них могут быть добавлены слоты: объем торговли и бюджет.

Задание 2.
Используя фреймовую модель представления знаний реализовать структуру отношений, описывающие следующие ситуации:
1. экзамен по дисциплине за семестр у преподавателя при составляющих: семестр, экзамен, преподаватель, оценка, студент, получать.
2. ведомость при составляющих: дисциплина, студент, экзамен, семестр, преподаватель, оценка.
3. конференция по коммерческим вопросам при составляющих: дата, место проведения, тема, цель выступающие.
4. получение оценки при составляющих: преподаватель, студент, оценка, получать.
5. использования изделия при составляющих: организация, разработка технологического решения, исследование «физического эффекта», методы создания изделия.
6. информационная структура БД в машиностроении при составляющих: физические эффекты, технические решения, изделия, объект поставки изделия, приборы и стенды, нормативы.
7. классификация продукта при составляющих: название, область применения, способ хранения, способ транспортировки.
8. аудитория (описание) при составляющих: вместимость, назначение, составляющие, местонахождение.
9. животный мир при составляющих: вид, тип, среда обитания, особенности поведения.

Практическая работа 3.
Представление знаний. Продукционная модель

Цель занятия – формирование навыков построения продукционную модель представления знаний в предметной области.

Задачи занятия:

- рассмотреть общее представление о продукционной модели;
- научиться реализовывать продукционную модель представления знаний в предметной области.

Задания для выполнения и методические рекомендации:

Продукция – это предложение-образец вида «Если, то», по которому осуществляется поиск в базе знаний.

В продукции выделяют левую часть (начинается с «если» и заканчивается перед «то») и правую (начинается после «то»). Левая часть продукции – антецедент – условие выполнения правой часть продукции. Правая часть – консеквент – действие, выполняемое в случае нахождения элементов, удовлетворяющих левой части. Действие может быть промежуточным и выступать затем в качестве консеквента или целевым, завершающим процедуру вывода.

Антецедент формируется из фактов, входных данных задачи и логических связок (и, или, не). Консеквент может представлять из себя действие по изменению фактов, данных, рекомендацию, решение задачи. Кроме этого, любая продукция имеет имя и приоритет, определяющий последовательность проверки продукций машиной вывода.

Продукции отражают причинно-следственные связи, которые и позволяют человеку принимать решения, базируясь на знаниях и предположениях о том, что есть и что будет, если что-то сделать.

Пример решения задачи

Задача. Построить продукционную модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Описание процесса решения. Для построения продукционной модели представления знаний необходимо выполнить следующие шаги:

1) Определить целевые действия задачи (являющиеся решениями).

2) Определить промежуточные действия или цепочку действий, между начальным состоянием и конечным (между тем, что имеется, и целевым действием).

3) Опередить условия для каждого действия, при котором его целесообразно и возможно выполнить. Определить порядок выполнения действий.

4) Добавить конкретики при необходимости, исходя из поставленной задачи.

5) Преобразовать полученный порядок действий и соответствующие им условия в продукции.

6) Для проверки правильности построения продукций записать цепочки продукций, явно проследив связи между ними.

Этот набор шагов предполагает движение при построении продукционной модели от результата к начальному состоянию, но возможно и движение от начального состояния к результату (шаги 1 и 2).

Решение.

1) Обязательное действие, выполняемое в ресторанах – поглощение пищи и ее оплата. Значит, есть уже два целевых действия «съесть пищу» и «оплатить», которые

взаимосвязаны и следуют друг за другом.
2) Прежде чем что-либо съесть в ресторане, туда нужно придти, дождаться официанта и сделать заказ. Кроме того, нужно выбрать, в какой именно ресторан пойти. Значит, цепочка промежуточных действий: «выбор ресторана и путь туда», «сделать заказ официанту».
3) Прежде чем идти в ресторан, необходимо убедиться, что есть необходимая сумма денег. Выбор ресторана может обуславливаться многими причинами, выберем территориальный признак – к какому ближе в тот и идем. В разных ресторанах работают разные люди, поэтому в зависимости от выбора ресторана, официанты будут разные. Кроме того, разные рестораны специализируются на разных кухнях, поэтому заказанные блюда будут в разных ресторанах отличаться. Значит вначале идут действия, позволяющие выбрать ресторан, затем характеризующие рестораны, а уже после заказ, еда, и оплата заказа.
4) Пусть в задаче будут рассматриваться два ресторана: «Вкусная еда» и «Вкуснятина». Первый –паб и заказы приносят быстрее, чем во втором, второй –пиццерия. В первом работает официант Сергей, а во втором официантка Марина. Петр –это клиент.
5) Выше описанное можно преобразовать в следующие предложения типа «Если, то»:

- *Если субъект хочет есть и у субъекта есть достаточная сумма денег, то субъект может пойти в ресторан.*
- *Если субъект ближе к ресторану «Вкусная еда», чем к ресторану «Вкуснятина» и субъект может пойти в ресторан, то субъект идет в ресторан «Вкусная еда».*
- *Если субъект ближе к ресторану «Вкуснятина», чем к ресторану «Вкусная еда» и субъект может пойти в ресторан, то субъект идет в ресторан «Вкуснятина».*
- *Если субъект идет в ресторан «Вкуснятина» и в ресторане «Вкуснятина» работает официант Марина, то у субъекта принимает заказ Марина.*
- *Если субъект идет в ресторан «Вкусная еда» и в ресторане «Вкусная еда» работает официант Сергей, то у субъекта принимает заказ Сергей.*
- *Если субъект выбрал блюда и у субъекта принимает заказ Марина, то заказ принесут через 20 мин.*
- *Если субъект выбрал блюда и у субъекта принимает заказ Сергей, то заказ принесут через 10 мин.*
- *Если заказ принесут через 20 мин. или заказ принесут через 10 мин., то субъект может есть.*
- *Если субъект может есть, то после еды субъект должен оплатить заказ.*

Введем обозначения для фактов (Ф), действий (Д) и продукций (П), тогда:
Субъект = *Петр*;
Ф1= *субъект хочет есть*;
Ф2= *у субъекта есть достаточная сумма денег*;
Ф3= *субъект ближе к ресторану «Вкусная еда», чем к «Вкуснятина»*;
Ф4=*в ресторане «Вкуснятина» работает официант Марина*;
Ф5=*в ресторане «Вкусная еда» работает официант Сергей*;
Ф6= *субъект выбрал блюда*;
Д1= *субъект может пойти в ресторан*;
Д2=*субъект идет в ресторан «Вкусная еда»;*
Д3=*субъект идет в ресторан «Вкуснятина»;*
Д4= *у субъекта принимает заказ Марина;*
Д5=*у субъекта принимает заказ Сергей;*
Д6=*заказ принесут через 20 мин.*
Д7=*заказ принесут через 10 мин.*
Д8=*после еды субъект должен оплатить заказ.*

Для продукций установим приоритет (в скобках перед запятой, чем выше приоритет, чем раньше проверяется правило).
П1(4 , Ф1 и Ф2)= Д1;
П2(5 , Ф3 и Д1)= Д2;
П3(4 , не Ф3 и Д1)= Д3;
П4(3 , Д3 и Ф4)= Д4;
П5(3 , Д2 и Ф5)= Д5;
П6(2 , Д4)= Д6;
П7(2 , Д5)= Д7;
П8(1 , Д6 или Д7)= Д8;
6) Для отображения взаимосвязи продукций построим граф (рис. 1).

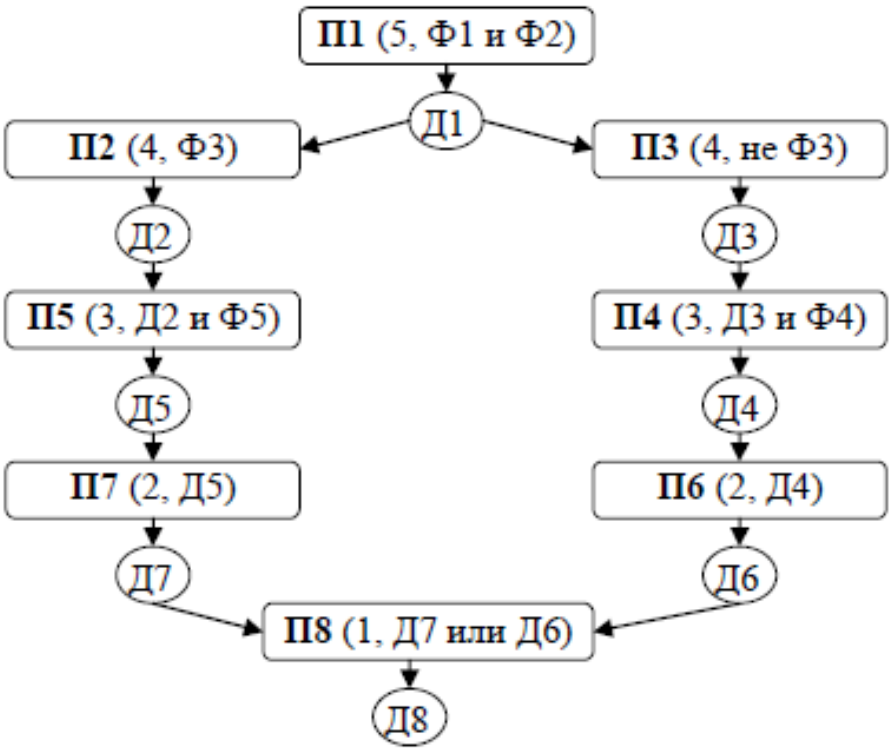


Рис. 1. Схема продукций предметной области «Ресторан».

Задание 1
Построить продукционную модель представления знаний в предметной области:
1. «Аэропорт» (диспетчерская).
2. «Железная дорога» (продажа билетов).
3. «Торговый центр» (организация).
4. «Автозаправка» (обслуживание клиентов).
5. «Университет» (учебный процесс).
6. «Интернет-кафе» (организация и обслуживание).
7. «Туристическое агентство» (работа с клиентами).
8. «Кухня» (приготовление пищи).
9. «Больница» (прием больных).
10. «Кинопрокат» (ассортимент и работа с клиентами).

Практическая работа 4.
Изучение принципов создания продукционных баз знаний

Цель занятия – изучение принципов и способов создания базы знаний продукционных экспертных систем, анализ методы сокращения баз знаний.

Задачи занятия:

- Выбрать предметную область для построения базы знаний.
- Разработать дерево целей базы знаний.
- Создать базу знаний для работы с оболочкой «Экспертная система».

Задания для выполнения и методические рекомендации:

1. Выбрать предметную область. Провести ее анализ и декомпозицию, выделив целевую и исходные вершины. Построить дерево целей. Рассчитать мощность базы знаний. (Можно взять продукционную модель, разработанную в л/р №3)
2. В результате анализа дерева целей реорганизовать его, введя промежуточные вершины, сократив таким образом мощность базы знаний. Рассчитать полученную мощность базы знаний.
3. Создать базу знаний (база знаний может быть дополнительно уменьшена за счет исключения некоторых ветвей дерева решений) использую оболочку «ЭС». База знаний должна удовлетворять требованиям полноты и непротиворечивости.

3.1. Скачать и разархивировать файл с «ЭС Divinus Expert System»

3.2. Запустить оболочку «ЭС».

3.3. Создать новый проект.

3.4. Внести в базу знаний исходные вершины (истоки) с возможными значениями и искомое значение (сток).

3.5. Внесите в базу знаний правила. Заполните истоки правила с указанием их значений и следствие (я).

4. Выполните тестирование базы знаний.

5. Сформируйте дерево решений и дерево правил.

Пример

Построим простую базу знаний, которая отвечает на вопрос: «Поехать на работу на автобусе?» (см. рисунок 1.1). Истоками базы будут являться: наличие денег (деньги), наличие времени (время), количество и размер багажа (багаж), наличие сидячих мест (сидячие места).

Пусть нашим истокам соответствуют значения:

1) Деньги – нет, есть
2) Время – мало, есть
3) Багаж – малый, средний, большой
4) Сидячие места – нет, есть.

Целевой вершиной будет являться: «Поехать на автобусе» со значениями – да, нет, ждать (ждать следующего).

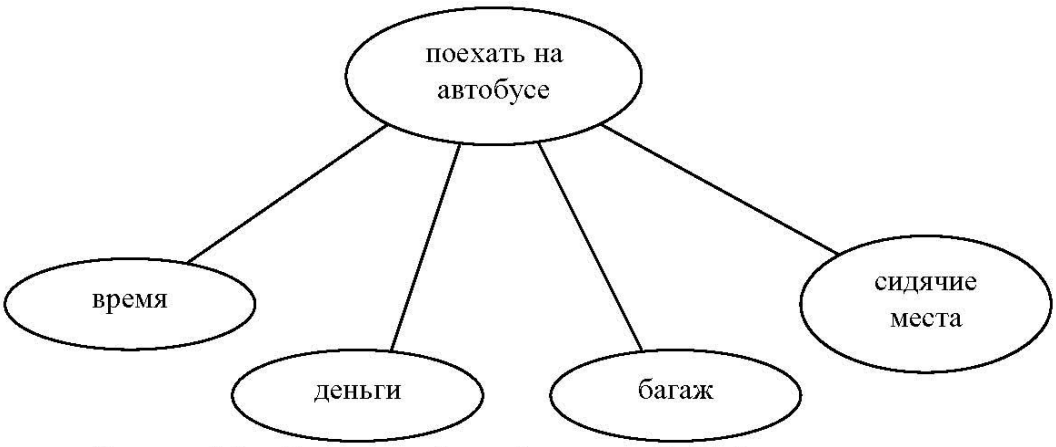


Рисунок 1.1 – Дерево целей для базы знаний «Поехать на автобусе»

Возникает вопрос: а сколько правил потребуется для реализации базы знаний с таким деревом целей?

Количество правил в базе знаний называется мощностью базы, и может быть рассчитана по дереву целей.

Для структуры дерева целей, приведенного на рисунке 1, мощность считается путем перемножения количества всех возможных вариантов по каждому истоку. Для рассматриваемого примера количества вариантов по вершинам равны:

Деньги – 2

Время – 2

Багаж – 3

Сидячие места – 2.

Значит мощность базы знаний равна = 2*2*3*2 = 24 правила.

Если присмотреться к имеющемуся на рисунке 1 дереву целей, можно прийти к выводу, что вершины «багаж» и «сидячие места» логически описывают один из аспектов вопроса «Поехать на работу на автобусе?». Поэтому их можно объединить одной общей вершиной – комфорт.

После добавления 1 новой вершины дерево целей изменится (рисунок 1.2).

Рисунок 1.2 – Дерево целей с промежуточными вершинами

Вершина «комфорт» называют промежуточной, потому что находится между истоками базы знаний и целевой вершиной. Она играют двоякую роль: с одной стороны, для вершин «багаж» и «сидячие места» она является целевой, с другой стороны для вершины «поехать на автобусе» она является истоком.

Нужно помнить, что истоки базы знаний могут встречаться только в левой части правил, целевая вершина – только в правой, а промежуточные в одних правилах встречаются в левой части, а в других – в правой.

Мощность базы знаний, в которую введены промежуточные вершины, рассчитывается с учетом количества значений у промежуточных вершин.

Для рассматриваемого примера одна промежуточная вершина:

Комфорт – не удобно, удобно, средне – 3

Для расчета мощности условно разобьем базу знаний на две части. Первая объединит вершины «багаж», «сидячие места» и «комфорт»; вторая – «комфорт», «деньги», «время» и «поехать на автобусе».

Каждая из частей представляет собой дерево подобное изображенному на рисунке 1 и их мощность будет рассчитываться аналогично.

Мощность первой части=2*3=6

Мощность второй части=2*2*3=12

Общая мощность рассчитывается как сумма мощностей, входящих в нее частей, т.е. 6+12=18.

Дерево целей не всегда имеет правильную форму. Существуют деревья целей, в которых вершина-исток может замыкаться и на промежуточную, и на целевую вершину.

Дерево решений предназначено для отображения всех возможных вариантов ответов базы знаний. Для дерева целей, изображенного на рисунке 1.2, дерево решений примет следующий вид (рисунки 1.3 и 1.4). Если посмотреть на получившиеся дерево решений с точки зрения логики, можно прийти к следующим заключениям:

Поехать на автобусе невозможно, когда нет денег (не купишь билет).

Поехать на автобусе нужно сейчас, если мало времени и есть деньги.

Эти соображения определяются целиком и полностью экспертом, составляющим базу знаний, и носят индивидуальный характер. Анализ выдвинутых положений приводит нас к пониманию избыточности дерева решений. Ненужные нам варианты можно исключить из базы знаний, отсекая ветви в дереве решений.

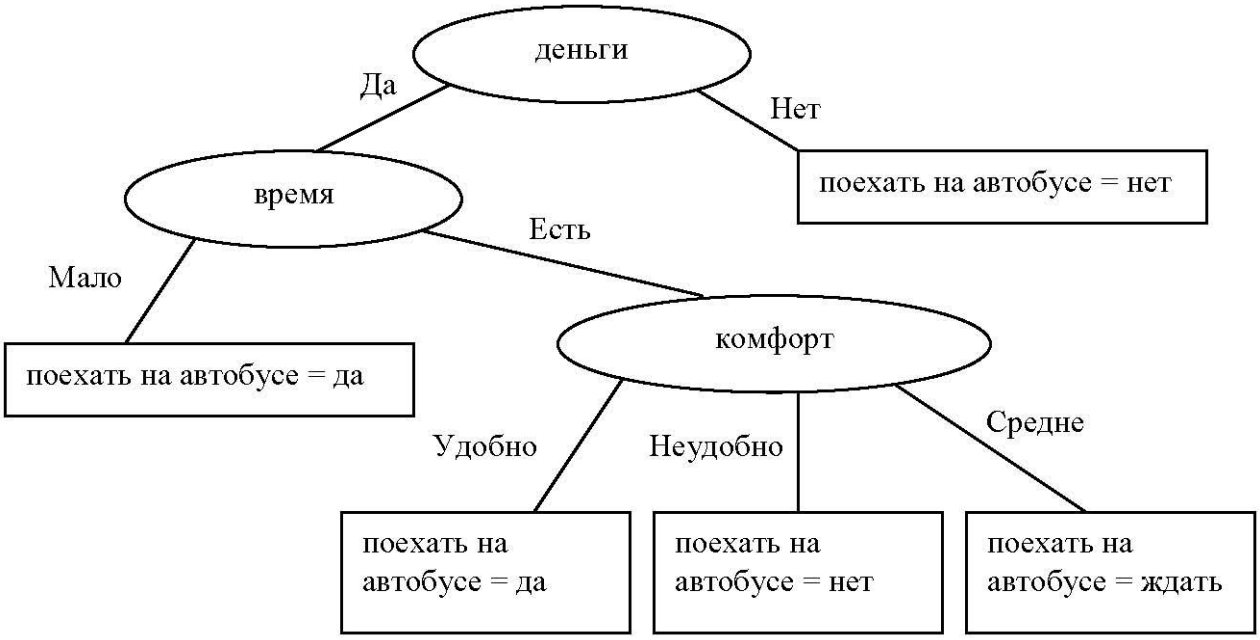


Рисунок 1.3 – Дерево решений для целевой вершины

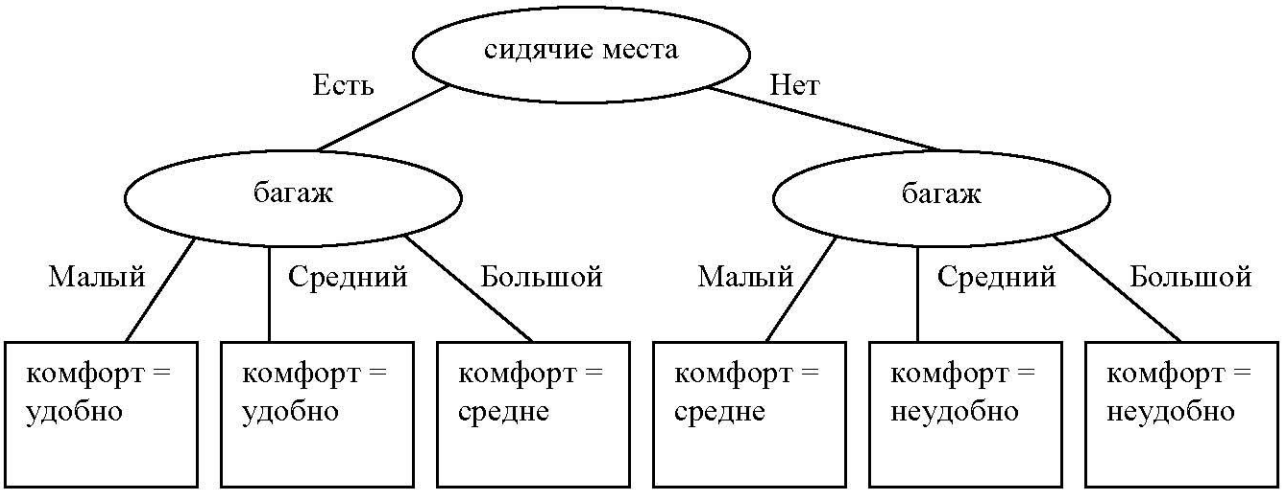


Рисунок 1.4 – Дерево решений для промежуточной вершины «Комфорт»

В результате, количество правил вместо 6+12=18 получится 6+5=11.

Построим дерево правил, которое предназначено для изображения всех правил базы знаний. Если для базы знаний было выполнено сокращение дерева решений (отсечены ветви), то в дереве правил появятся правила (кружки) к которым линиями присоединены не все истоки (квадраты слева).

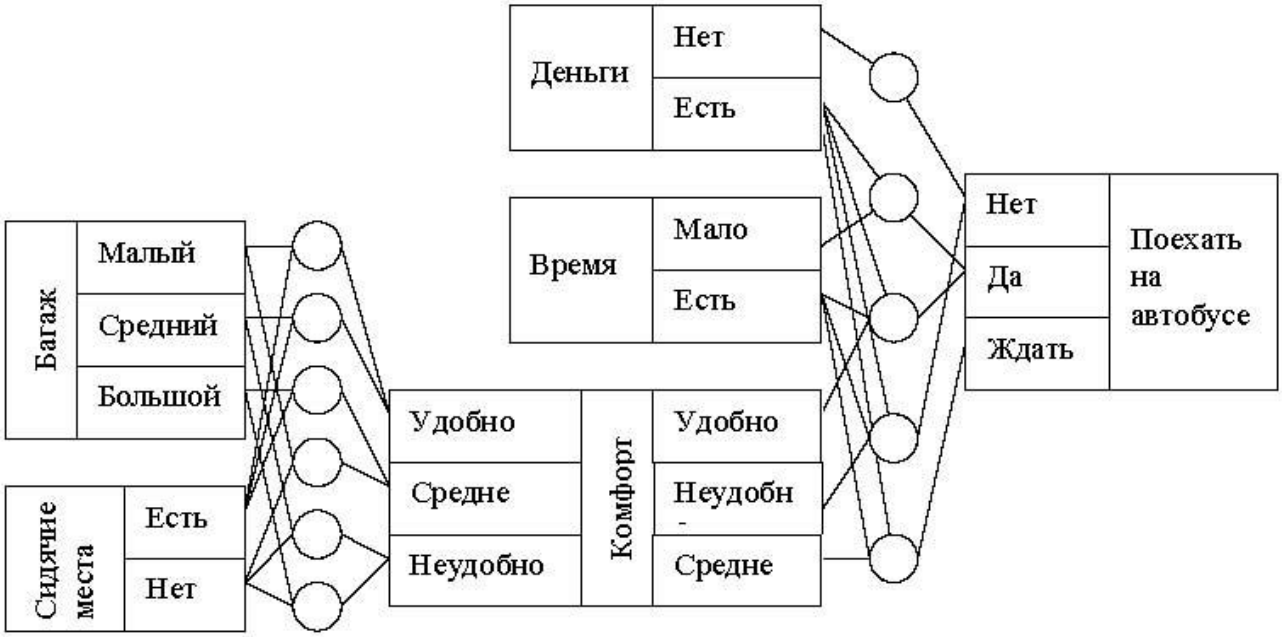


Рисунок 1.5 – Дерево правил

Покажем пример работы с программной оболочкой для создания продукционных экспертных систем. Запускаем ES.exe (рисунок 1.6).

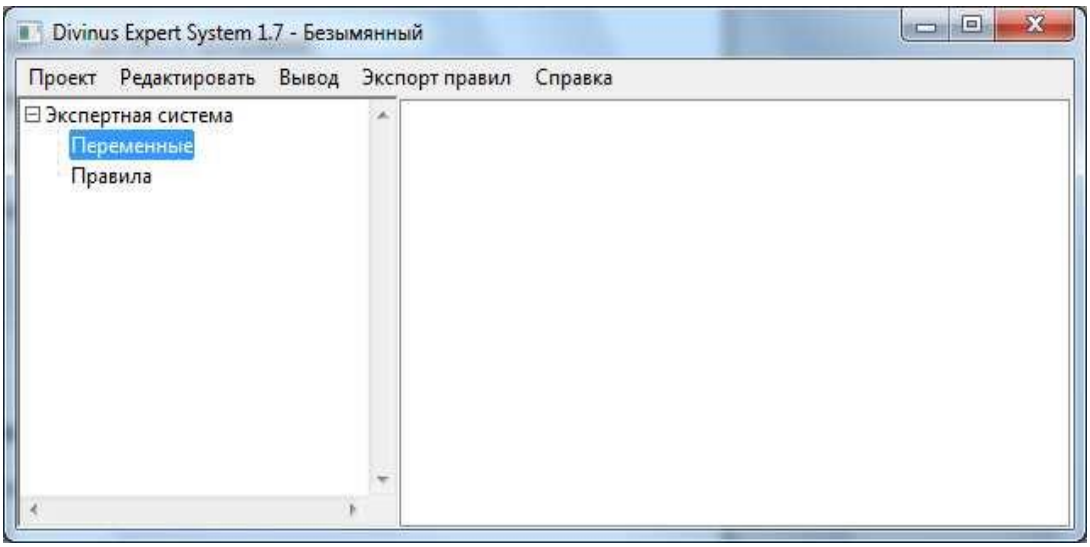


Рисунок 1.6 – Внешний вид оболочки для построения экспертных систем

Добавление новых переменных осуществляется через пункт меню «Редактировать – Добавить переменную» либо контекстное меню (правой кнопкой мыши по дереву слева). В появившейся панели справа необходимо указать наименование переменной, вид (входная, выходная или промежуточная), тип и набор значений. Входная переменная может использоваться в правилах только в левой части (посылке), выходная – только в правой (следствии), промежуточная – и в левой, и правой частях (рисунок 1.7).

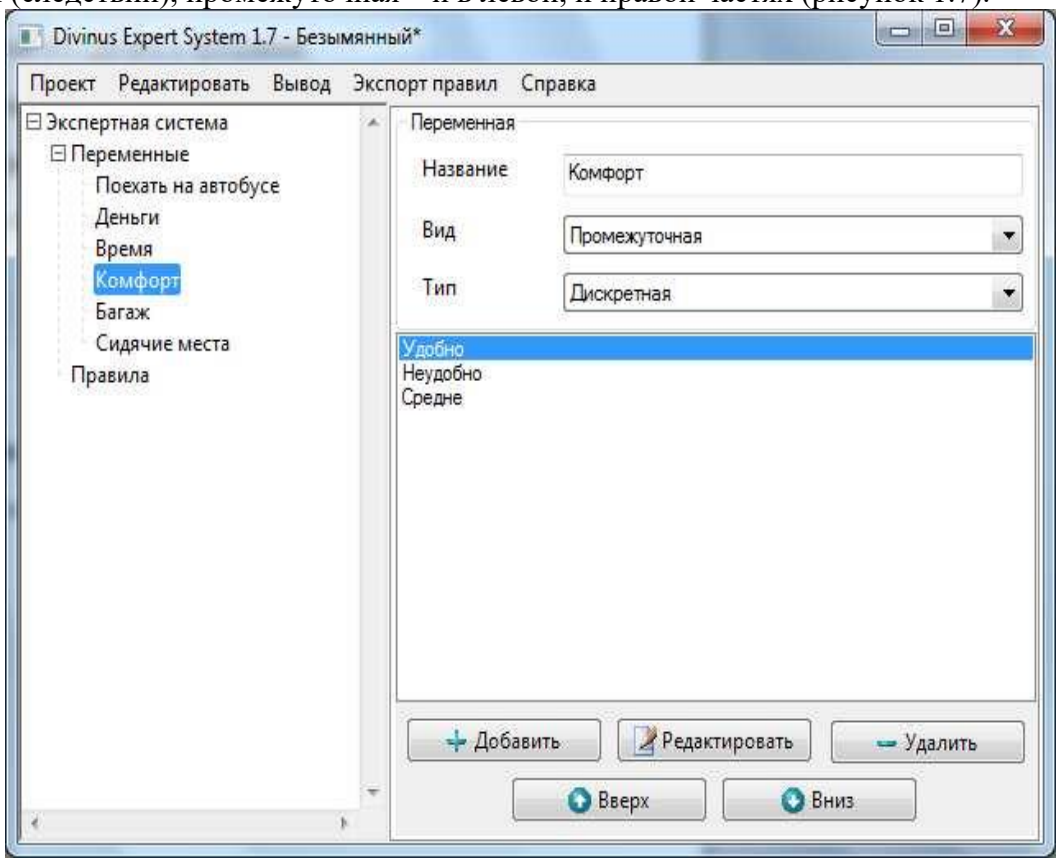


Рисунок 1.7 – Настройка переменных

Далее необходимо создать в оболочке «ES» саму базу правил (рисунок 1.8).

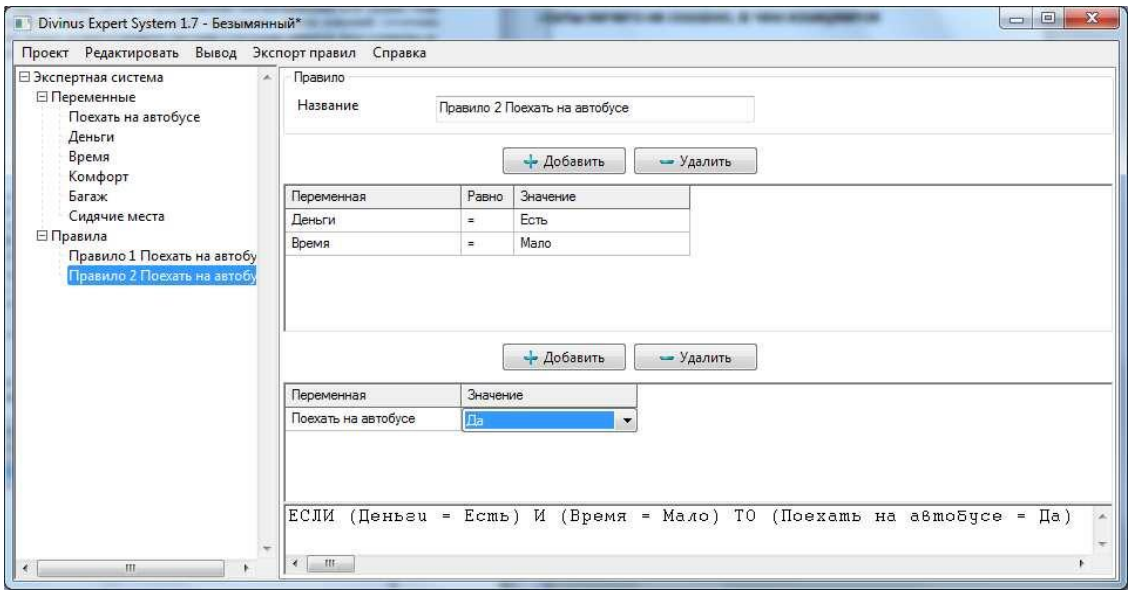


Рисунок 1.8 – Добавление нового правила

Протестировать работу экспертной системы возможно через меню «Вывод» (рисунок 1.9).

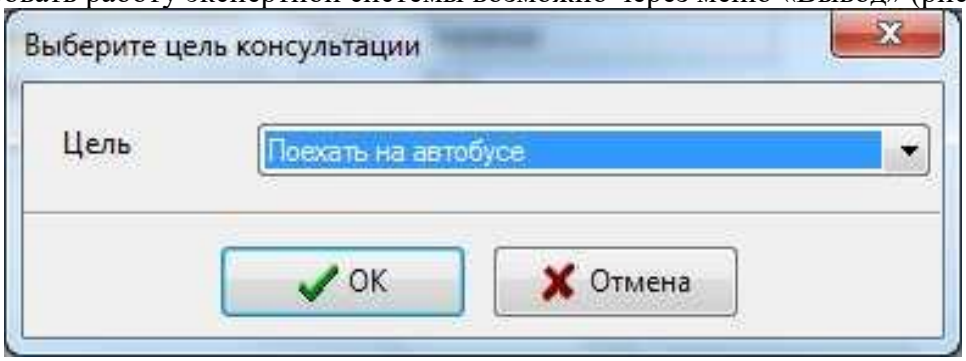


Рисунок 1.9 – Выбор цели консультации пользователя

Далее необходимо ввести значения всех входных переменных (рисунок 1.10).

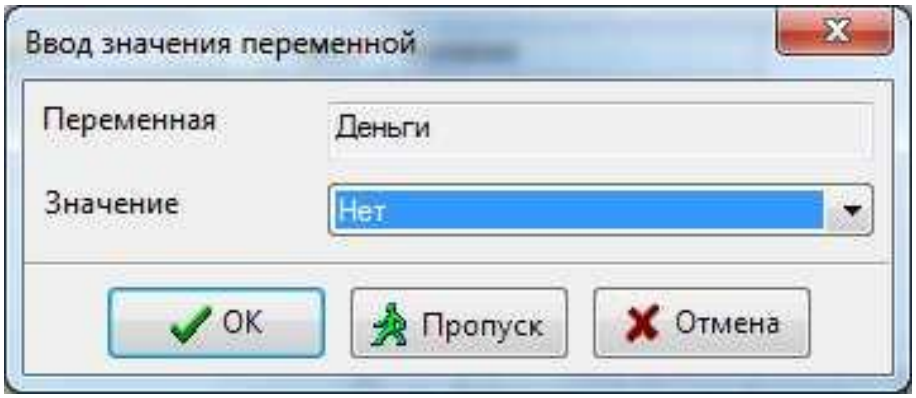


Рисунок 1.10 – Выбор значения входной переменной

Система сама проверит все правила и выдаст результат (рисунок 1.11).

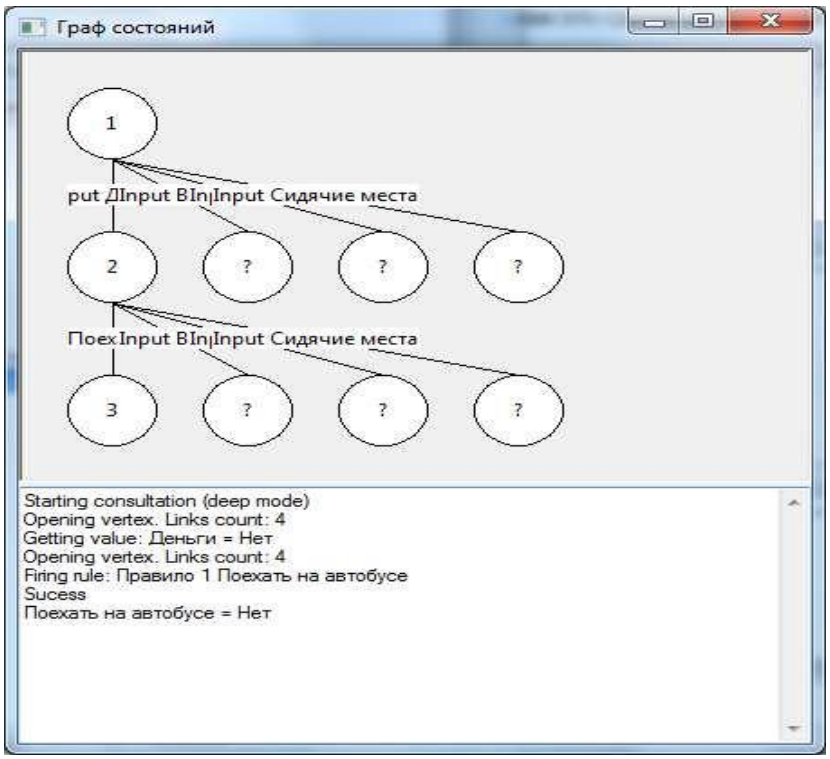


Рисунок 1.11 – Результат тестирования экспертной системы

Практическая работа 5.
Построение онтологической модели в Protégé. Создание классов

Цель занятия – сформировать представление о построении онтологических моделей в среде Protégé.

Задачи занятия:

- изучить общее представление о построении онтологических моделей;
- изучить технологию работы в среде Protégé;
- определить предметную область, создать классы, отношения с использованием системы Protégé, а также на примере созданной БЗ рассмотреть модель RDF, язык OWL и синтаксис Turtle.

Задания для выполнения и методические рекомендации:

Для хранения знаний используются разные структуры:

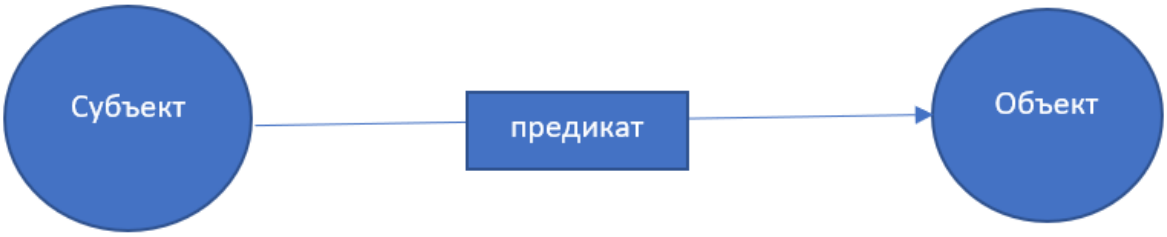
- управляемые словари: обеспечивают способ организации знаний для последующего поиска, используются в схемах предметной индексации, предметных рубриках, тезаурусах, таксономиях и других системах организации знаний,
- тезаурусы: объединяют термины в группы по определенному признаку, например, с учетом похожести (синонимы),
- таксономии: категорированные слова, упорядоченные по иерархическому признаку,
- онтологии: формальное описание знаний из какого-то домена (предметной области) с учетом имеющихся сложных правил и связей между элементами, позволяющим сделать автоматическое извлечение знаний,

- датасеты: наборы машиночитаемых данных.

Онтологии служат для систем организации знаний и применяются в тех областях, где требуется обнаружить новые факты, выявить скрытые взаимосвязи между элементами (например, рекомендательные и экспертные системы). Это альтернатива классическим базам данных, которые используют «гипотезу закрытого мира», когда все, чего нет в базе данных – не существует. В противоположность этому онтологии используют «гипотезу открытого мира», то есть если чего-то нет в базе знаний, то это не обязательно не существует, а просто не описано.

Системы организации знаний на основе онтологий уже очень распространены и используются во многих отраслях. Самый яркий пример это knowledge graph для поиска информации в Интернет, благодаря этой технологии качество поиска стало очень высоким.

В математическом основании онтологии лежит так называемая дескриптивная логика (раздел математики), предполагающая, что любая информация, высказанная на естественном языке, может быть представлена в виде цепочки триплетов:

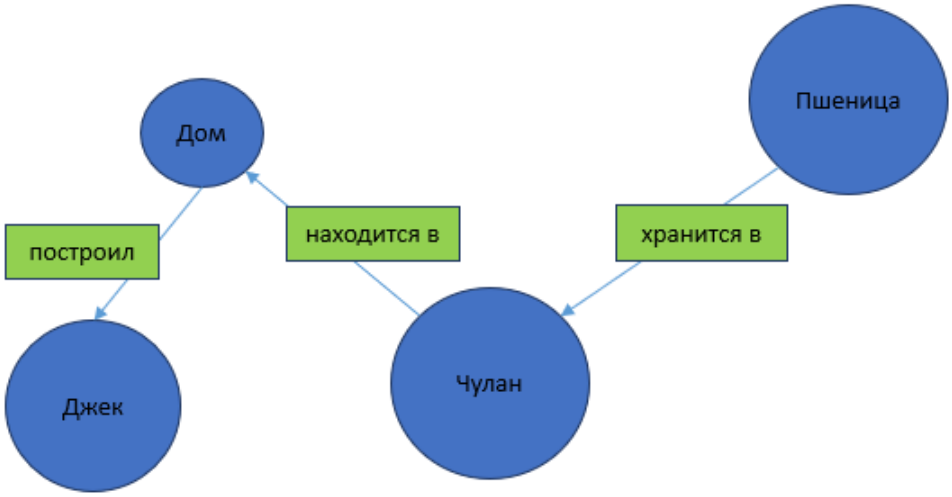


Отрывок из стихотворения «Дом, который построил Джек…» (пер. С.Я. Маршака).

Вот **дом**,
Который построил **Джек**.

А это **пшеница**,
Которая в тёмном **чулане хранится**
В **доме**,
Который построил **Джек**.
...

Описанные в стихах отношения между различными сущностями мы можем представить в виде онтологии.



Онтология представляется в виде графа, вершины которого это сущности, а ребра – отношения между сущностями. Считается, что любое утверждение на естественном языке можно представить в виде простых предложений, из которых можно извлечь сущности и

отношения между ними. Есть два основных инструмента: RDF (Resource Description Framework) или OWL (Ontology Web Language). OWL позволяет дополнительно описывать логические правила над данными. Онтологии (в отличие от обычных баз данных) позволяют находить скрытые данные. Обычные хорошо подходят для поиска конкретной информации, а базы знаний нужны там, где надо выявлять новые знания, например, в системах поддержки принятия решений (экспертных системах).

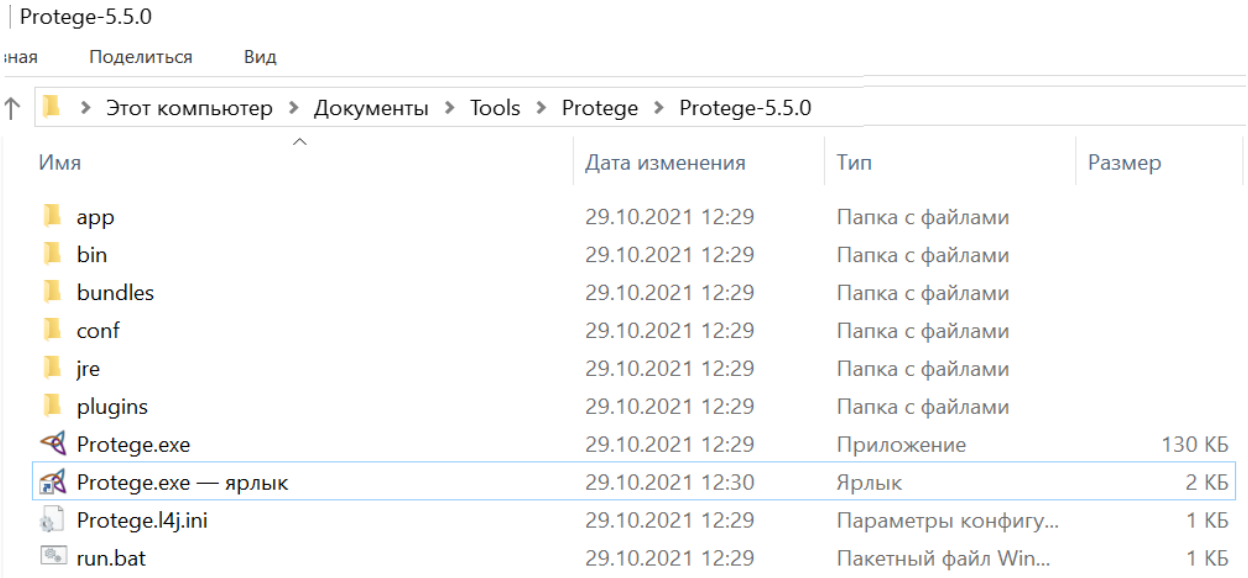
Сила онтологии проявляется в том случае, если подробно и качественно описаны взаимосвязи между ее элементами, с использованием математического аппарата дескриптивной логики. Например, для отношений можно задать их свойства (функциональное, транзитивное, рефлексивное). И тогда можно автоматически из онтологии извлекать факты, этот процесс называется ризонинг (reasoning), есть типовые алгоритмы ризонинга, основанные на графах. Возможные применения: уточнение характеристик объекта и выделение из набора похожих объектов уникального, поиск похожих объектов, «понимание текста» и отнесение текста к определенному классу, помощь в NLP задачах (NER, Relation Extraction), анализ корневых причин, выявление паттернов в данных. Наиболее популярный редактор онтологий, поддерживающий ризонинг, это Protégé. Существуют и другие инструменты, например: IBM Watson, Wolfram Alpha.

Создание процесса онтологий обычно выполняется вручную, силами экспертов. Однако есть и примеры автоматизированного создания онтологий на основе имеющихся баз знаний. Открытые графы знаний (по состоянию на конец 2021 года, по данным The Linked Open Data Cloud):

- DBpedia;
- Yago + wordnet.princeton.edu;
- WikiData;
- Открытые базы знаний, в том числе предметные и отраслевые базы знаний, например, медицина: BioPortal, Bio2RDF

Описанные выше инструменты содержат различные методы работы с онтологиями. Однако наиболее популярным инструментом является Protégé.

Программа Protégé для создания, редактирования и использования онтологий разработана в Стэнфордском университете, распространяется бесплатно, ее можно скачать себе на компьютер по ссылке, либо воспользоваться web-версией. На официальной странице можно скачать архив для использования на локальном компьютере. Важно обратить внимание на операционную систему и разрядность процессора (64 или 32). Актуальная версия Protégé - 5.5, при необходимости использования 32-разрядной версии надо выбрать более низкую версию Protégé, например 4.3. Файл архива выглядит следующим образом.



Можно разархивировать в любую папку и запустить файл Protégé.exe. Теперь можно начинать создавать онтологии.

Рекомендации по выполнению лабораторной работы.

Создание и разработка онтологий с помощью языков RDF/RDFS и OWL, а также выполнение SPARQL-запросов возможны в редакторе Protégé .

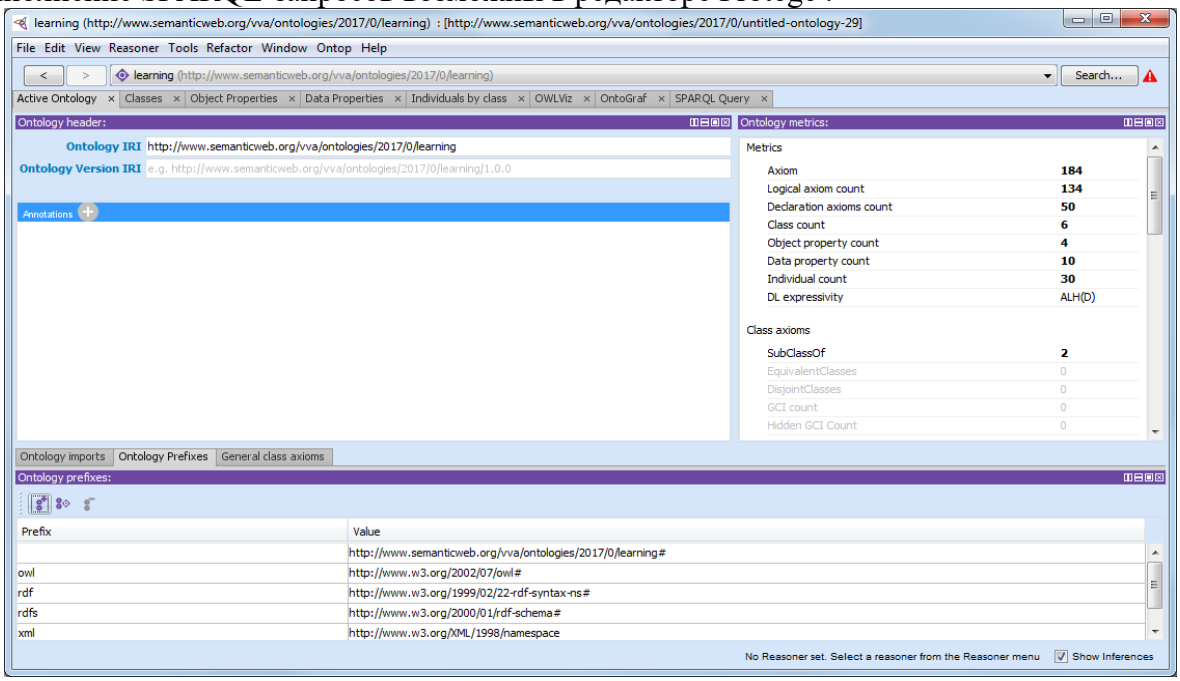


Рис.1. Редактор онтологий Protégé

Перед созданием онтологии необходимо настроить отображение в окне программы необходимых вкладок.

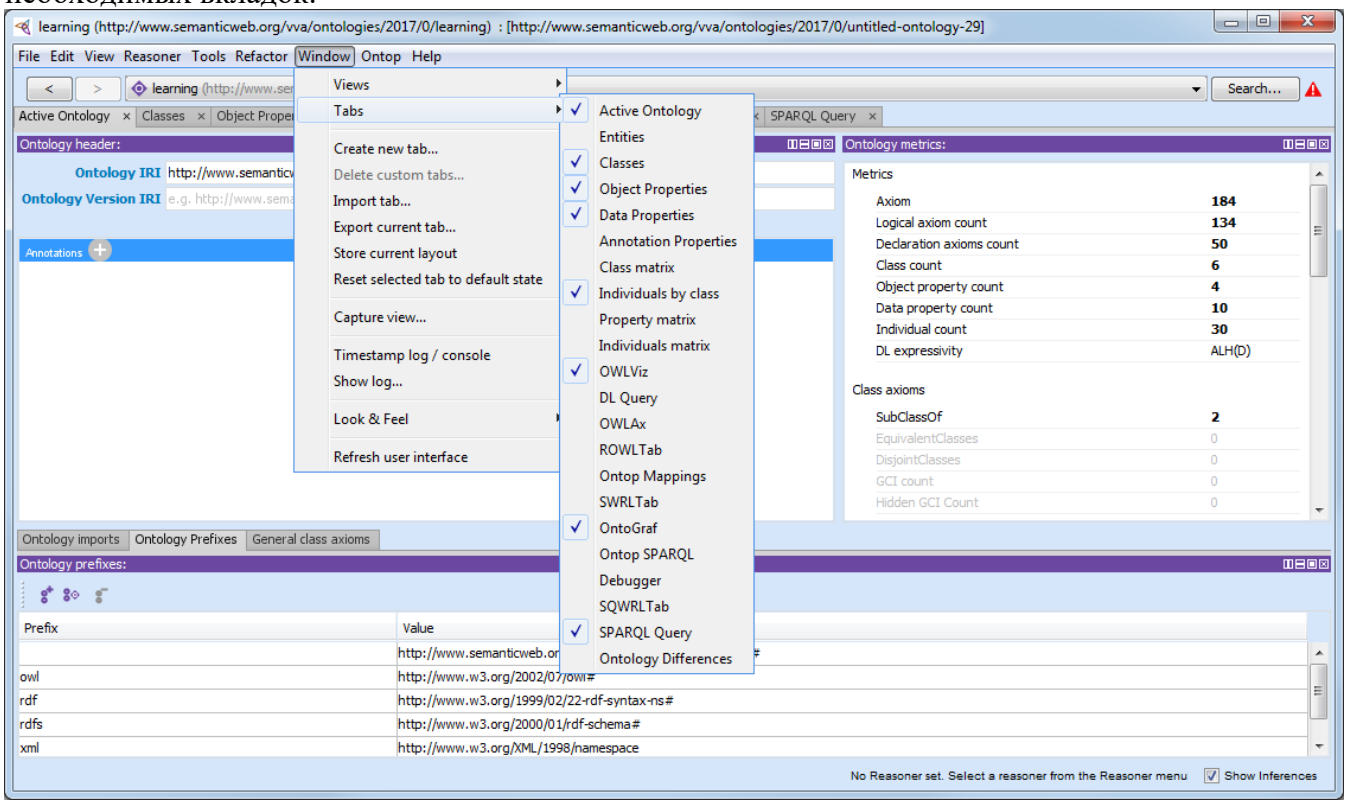


Рис.2. Выбор отображаемых вкладок

Назначение вкладок:

- Active Ontology – отображение IRI, общих характеристик (количества аксиом, классов, свойств, объектов и т.п.) и аннотации онтологии;
- Classes – просмотр и редактирование классов;
- Object Properties – просмотр и редактирование свойств-отношений между индивидами;
- Data Properties – просмотр и редактирование свойств-данных индивидов;
- Individuals by class – просмотр и редактирование экземпляров классов (индивидов);
- OWLVis и OntoGraf – визуализация онтологии в виде графа;
- SPARQL Query – выполнение запросов.

Классы создаются на вкладке «Classes». Добавить новый (удалить имеющийся) класс или подкласс можно с помощью кнопок панели инструментов или выбора пункта контекстного меню.

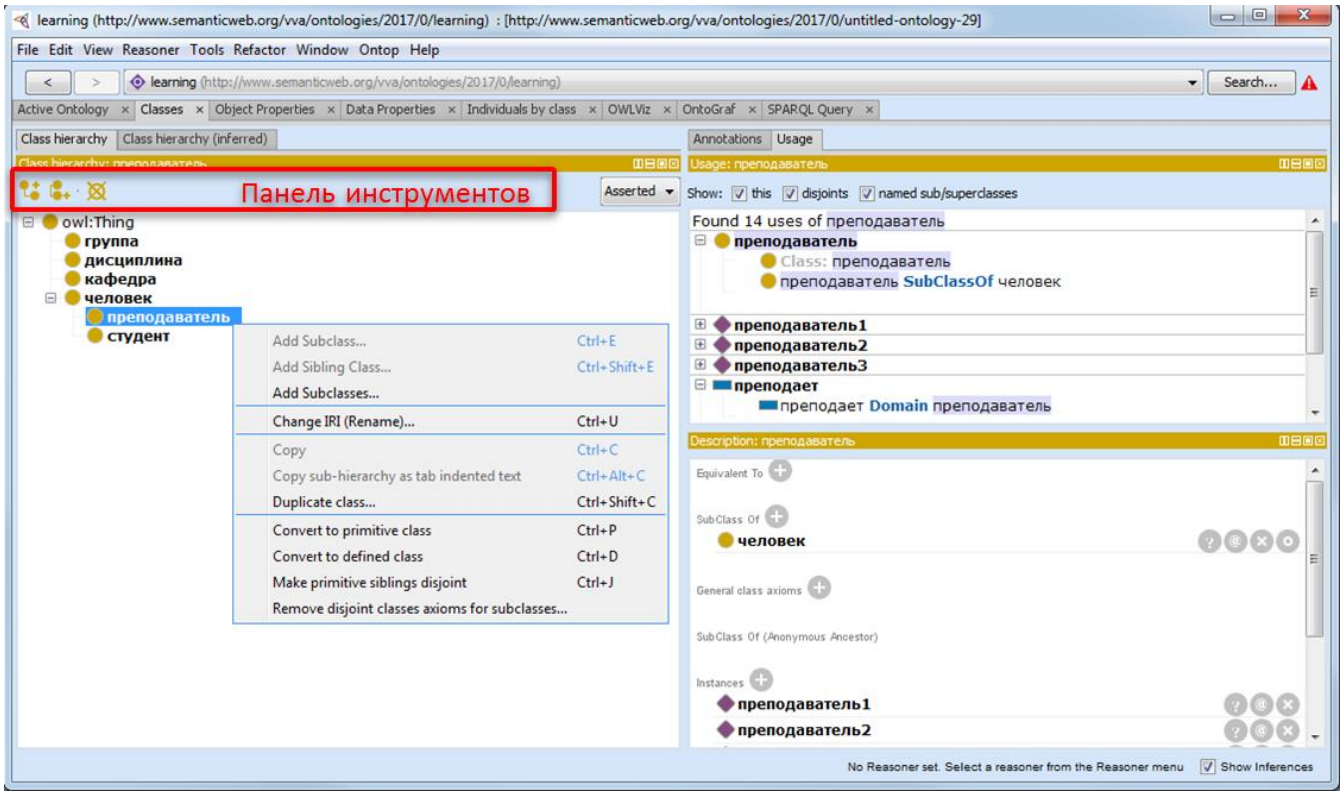


Рис.3. Вкладка «Classes»

В OWL базовым классом, на основе которого создаются классы онтологии, является класс «owl:Class». Остальные классы по отношению к нему являются дочерними подклассами (англ. Subclass). Классы одного уровня иерархии в Protégé называются родственными (англ. Sibling Class).

На вкладке «Применение» (англ. Usage) для класса, выделенного в иерархии, отображается его связь с родительскими и дочерними классами, его свойства, экземпляры и т.п.

В правой нижней панели «Описание» (англ. Description) можно указать дополнительные характеристики класса. Например, его эквивалентность другим классам (англ. Equivalent To) или невозможность принадлежности экземпляров класса другим классам (англ. Disjoint With) – запрет множественного наследования.

Свойства классов и их экземпляров (предикаты RDF-троек) делятся на два вида:

- **свойства-отношения** задаются на вкладке «Object Properties» и определяют некоторые отношения между двумя индивидами (экземплярами классов), т.е. субъектом и объектом RDF-тройки будут индивиды;
- **свойства-данные** задаются на вкладке «Data Properties» и определяют некоторые фактические характеристики индивидов (экземпляров классов), т.е. субъектом

RDF-тройки будет индивид, а объектом значение характеристики в виде строки, числа, даты и т.п.

Создание и редактирование свойств-отношений выполняется на вкладке «Object Properties».

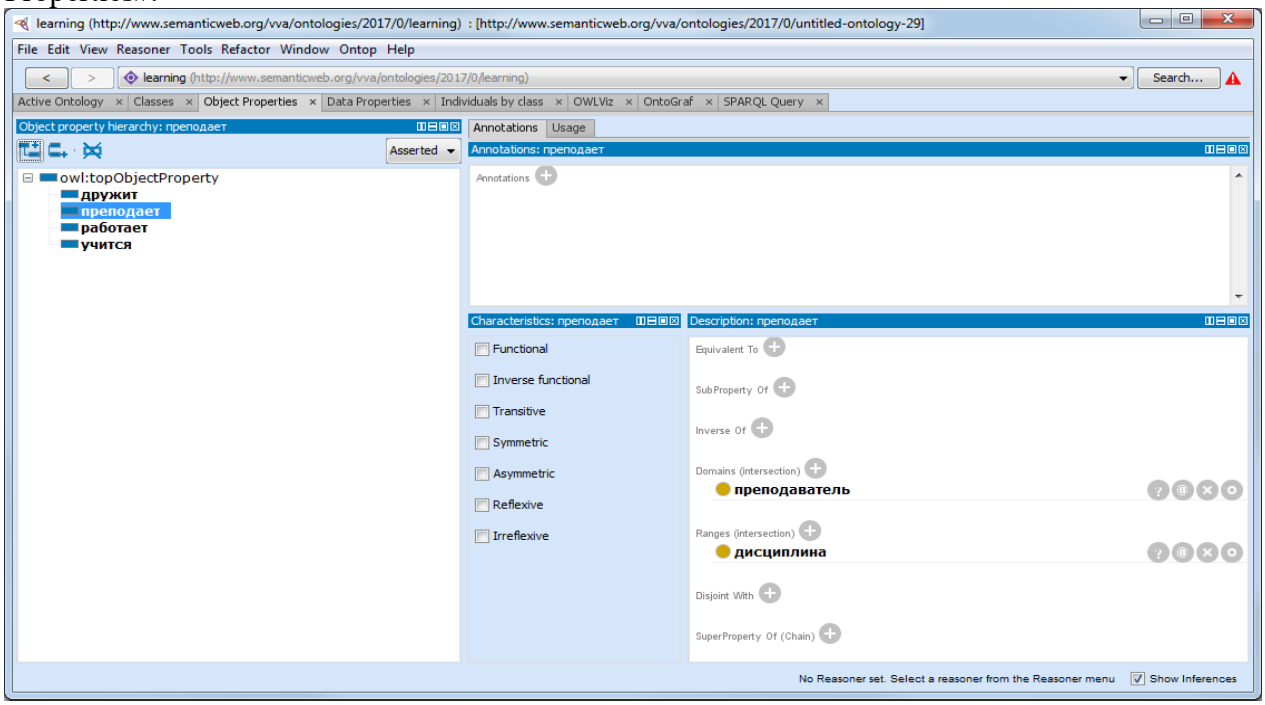


Рис.4. Вкладка «Object Properties»

Домен (англ. Domain) указывает, экземпляры каких классов в RDF-тройке при использовании данного свойства будут выступать в качестве субъектов, а диапазон (англ. Range) – в качестве объектов.

На панели «Характеристики» (англ. Characteristics) выбираются дополнительные характеристики свойства: транзитивность (англ.Transitive), симметричность (англ. Symmetric), рефлексивность (англ. Reflexive) и т.п.

Создание и редактирование свойств-данных выполняется на вкладке «Data Properties».

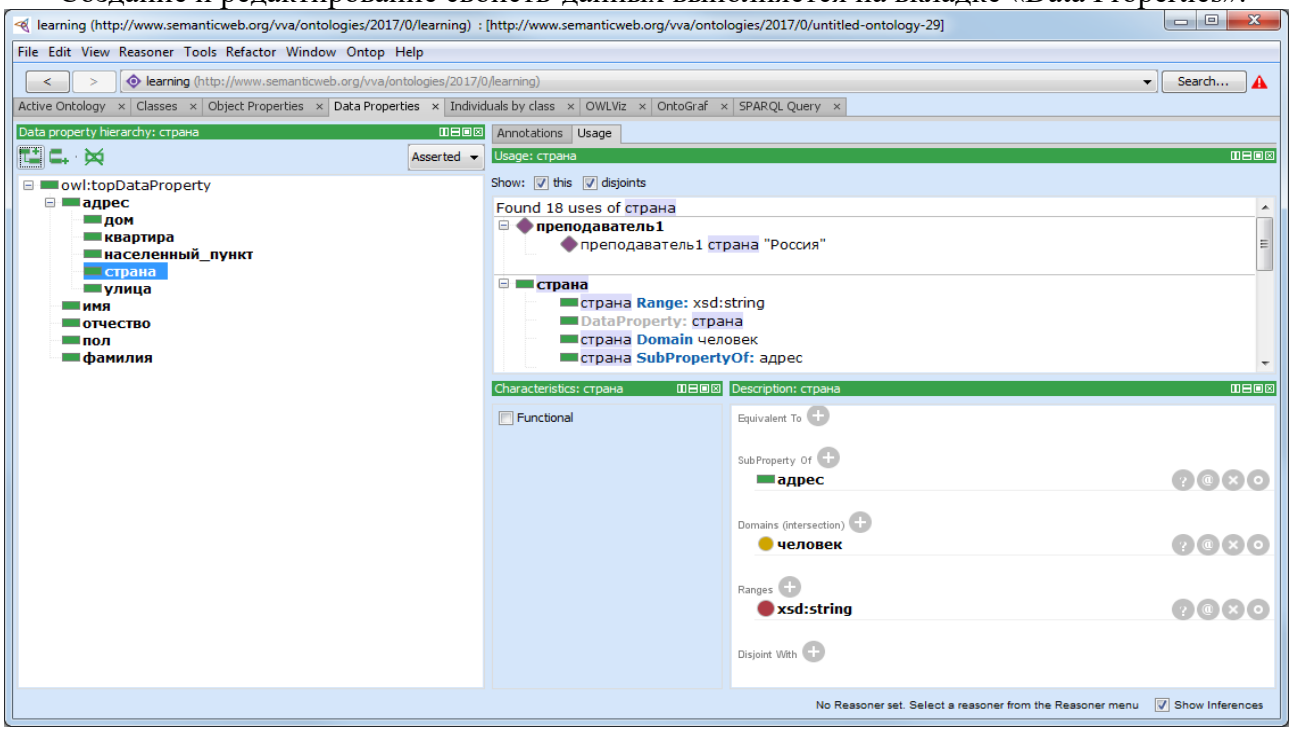


Рис.5. Вкладка «Data Properties»

Домен указывает, для экземпляров каких классов данное свойство может быть использовано. Диапазон задает область допустимых значений (тип данных и ограничения), которые можно указывать для данного свойства экземпляру класса. Тип данных выбирается из стандартных типов XML.

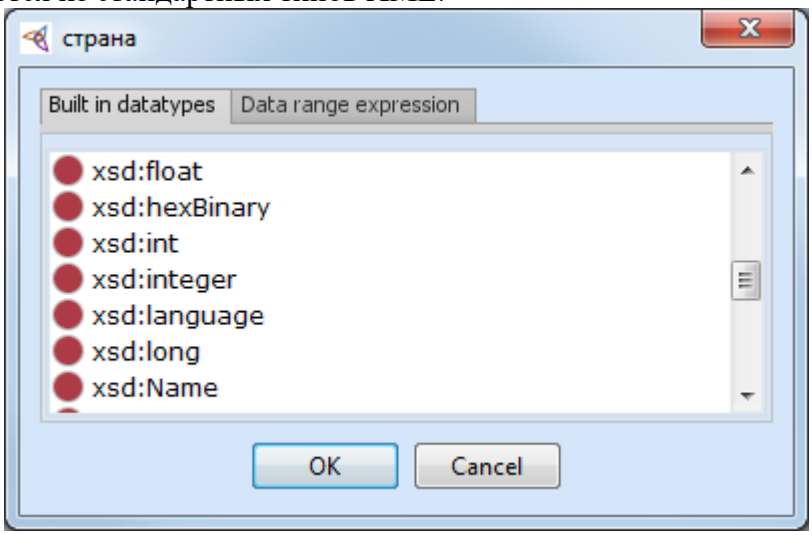
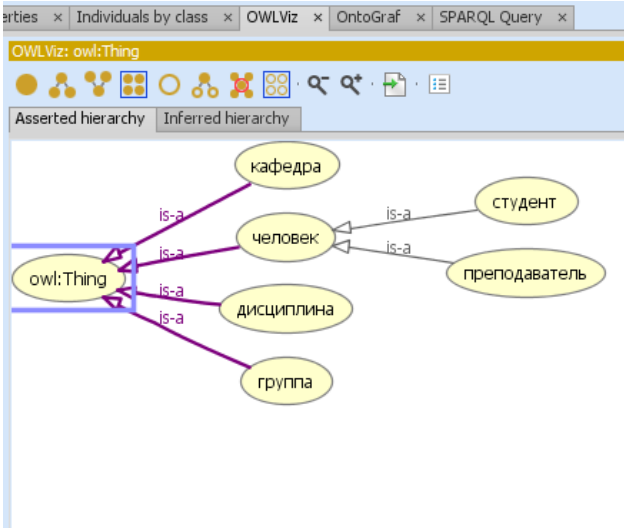
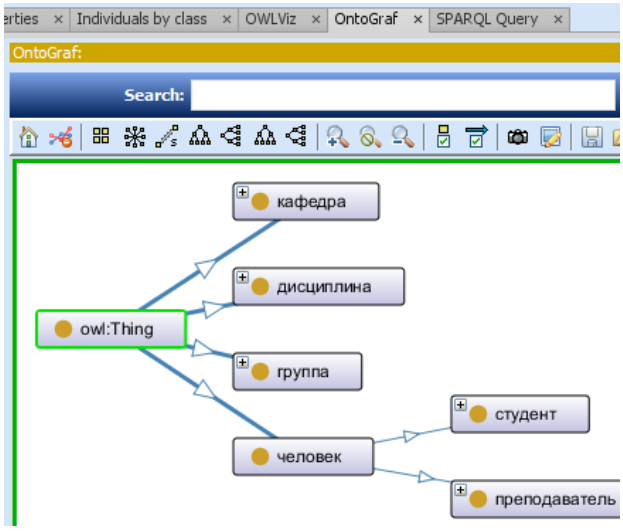


Рис.6. Окно выбора типа данных

Автоматическое построение и отображение графа онтологии и взаимосвязей между классами выполняется при выборе вкладок «OWLViz» и «OntoGraf».



а) OWLViz



б) OntoGraf

Рис.7. Варианты графического отображения онтологии

Перед записью на диск (пункт меню «File / Save as …») можно выбрать формат (нотацию) хранения онтологии.

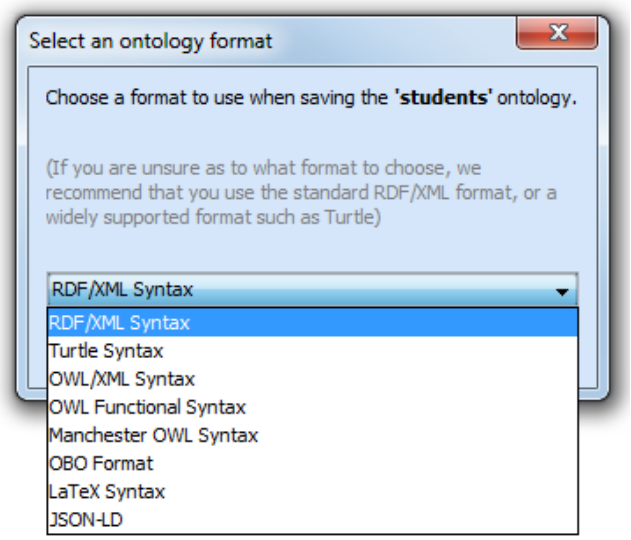


Рис.8. Выбор формата хранения онтологии

Задания для выполнения и методические рекомендации:

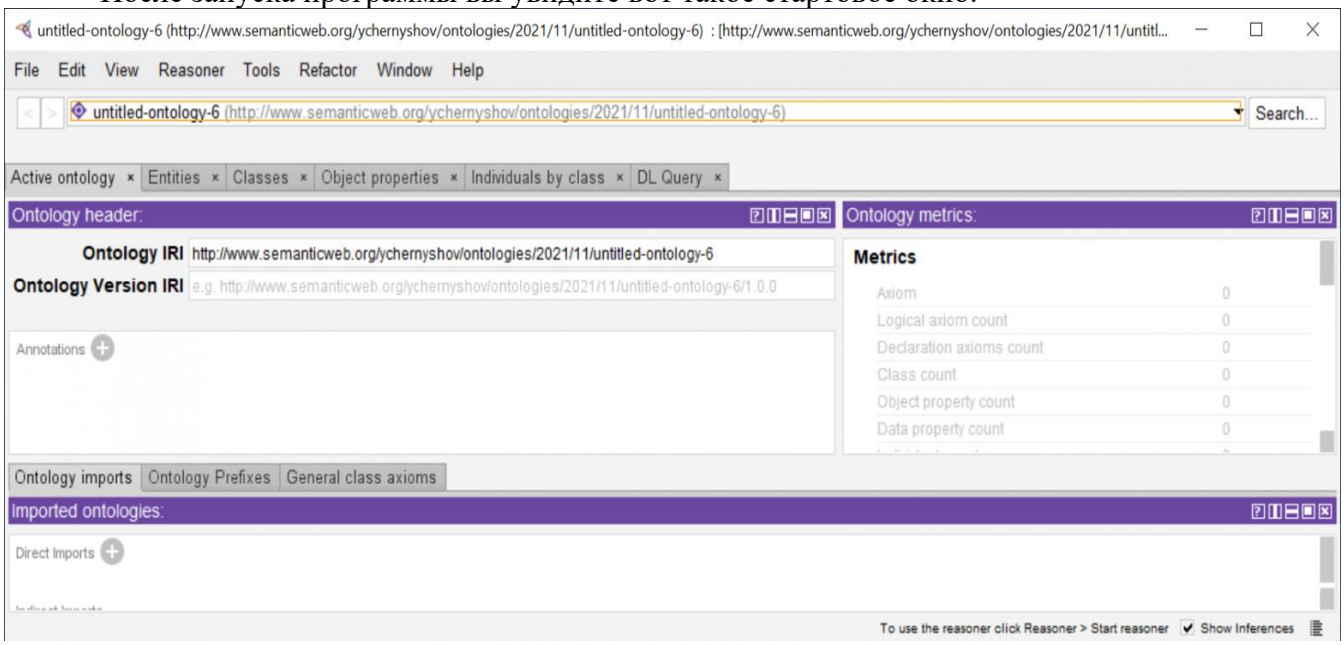
Создание проекта

В качестве примера рассмотрим создание онтологии родственных отношений, то есть онтологии, которая позволяет описывать родственные отношения между людьми

Для решения данной задачи выполним следующее.

1. Запустите редактор Protégé (нужно запустить файл Protege.exe или run.bat)

После запуска программы вы увидите вот такое стартовое окно.

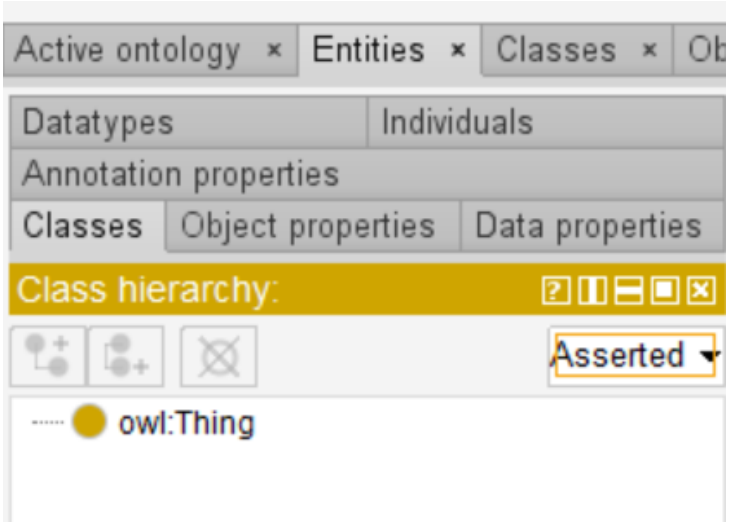


Тут все написано на ЯП Java и на английском, но для понимания происходящего достаточно английского.

Для каждого проекта есть уникальный идентификатор IRI (Internationalized Resource Identifier).

Protégé позволяет записывать триплеты, то есть тройки вида «субъект-предикат-объект».

Раздел «Сущности» (Entities) позволяет описывать субъекты и объекты.



Важные вкладки в этом разделе:

- экземпляры классов (Individuals) это объекты классов, как в объектно-ориентированном программировании. Например, класс «Сервер», объект «prod-serv-002».
- свойства (object properties или data properties), похоже на свойства классов в ООП, однако в онтологии свойства имеют независимую природу, свойство не является неотделимым от класса (как в ООП).

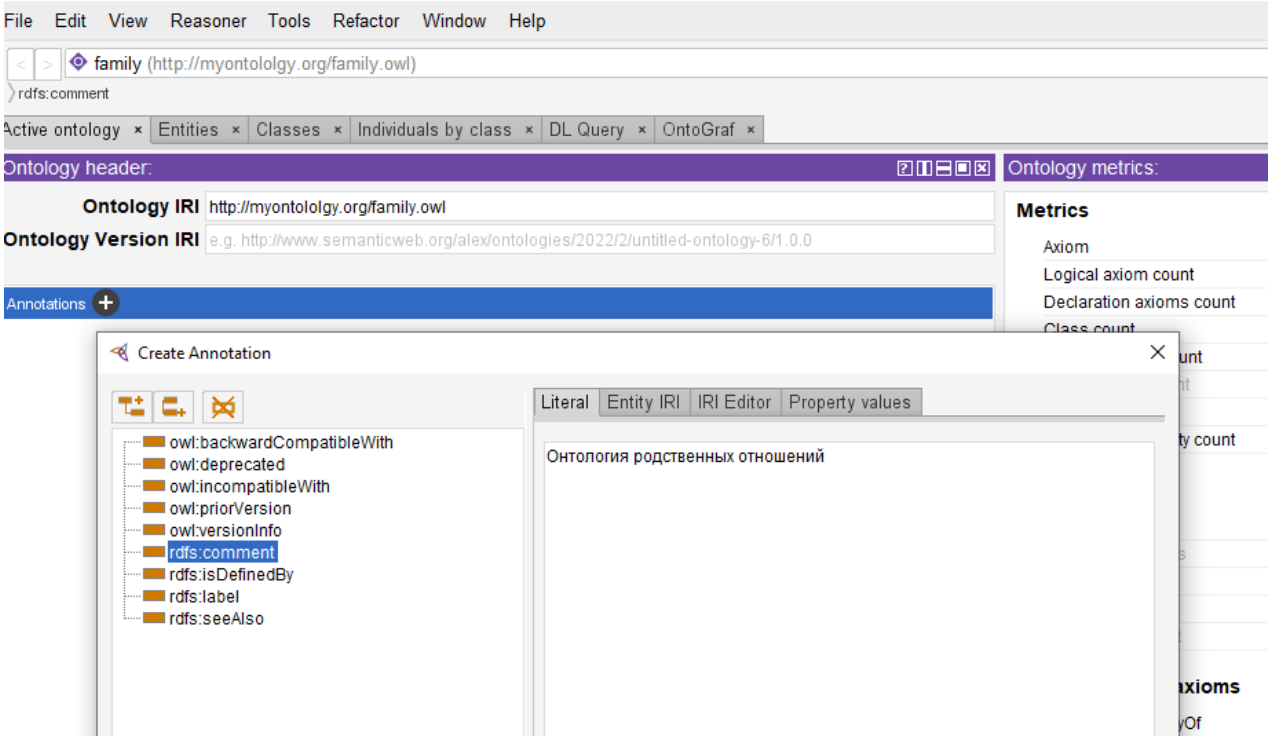
Для предикатов мы можем задавать разные свойства, например:

- функциональный;
- обратный;
- транзитивный.

По описанной онтологии можно применять ризонер, который дает предложения по найденным фактам (можно принять или отказаться).

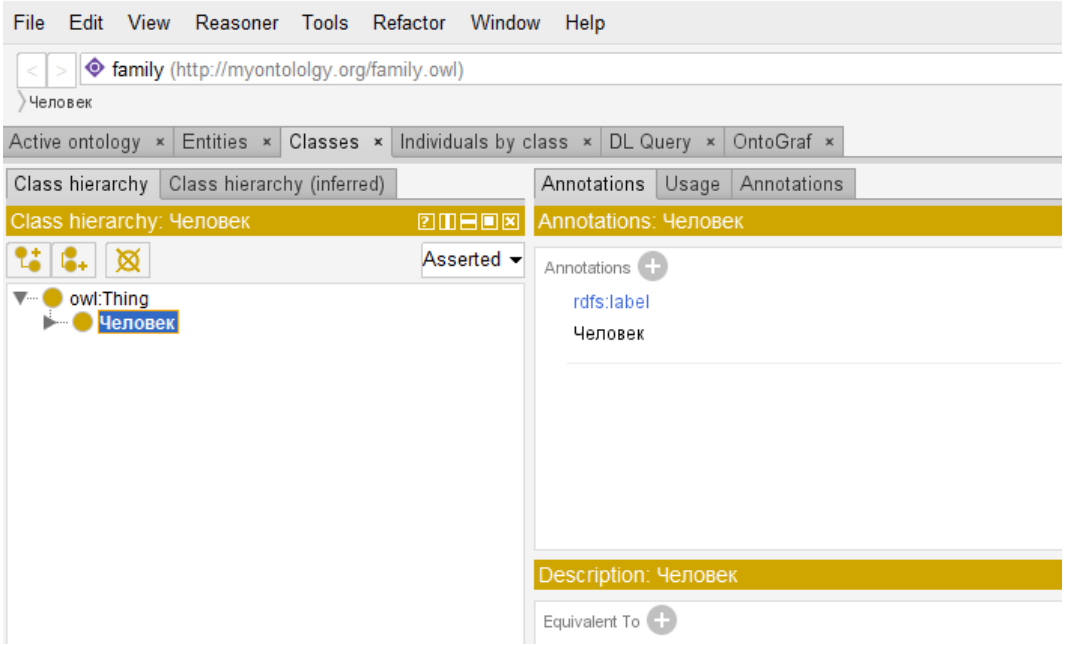
Онтологию можно сохранить во внешний файл, в формате owl, который можно открыть в текстовом редакторе и увидеть, что это xml.

2. Для того, чтобы начать использовать онтологию, нам нужно вести уникальный идентификатор IRI (например, http://myontololgy.org/family.owl). Также необходимо добавить аннотацию, например, «Онтология родственных отношений»

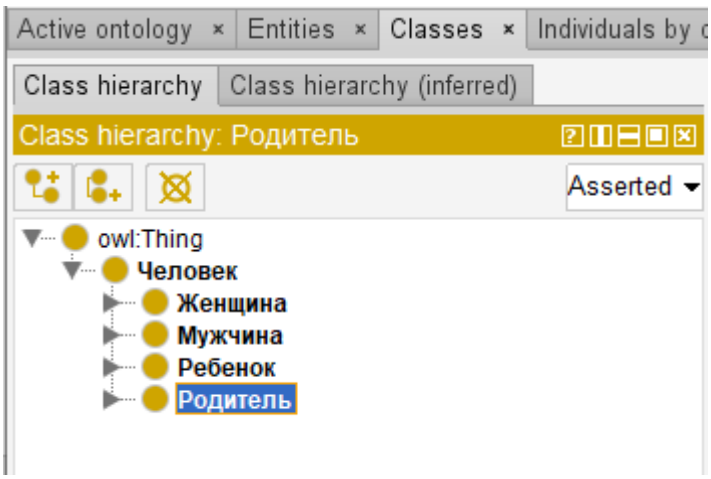


3. Теперь файл можно сохранить, выбрав пункт меню «File / Save»

4. Далее начнем разрабатывать онтологию. Прежде всего в онтологии необходимо определить так называемые концепты или в терминологии редактора Protégé - это классы. Для этого переходим на вкладку классы (Classes) и видим слева панель иерархии классов (Class hierarchy). Сначала создадим класс «Человек». Для этого необходимо выделить класс, подклассами которого будут верхние классы нашей иерархии (в нашем случае owl:Thing). После этого вызвать контекстное меню (правой кнопкой мыши – ПКМ) и выбрать пункт «Add subclasses». Классы также могут добавляться по одному.

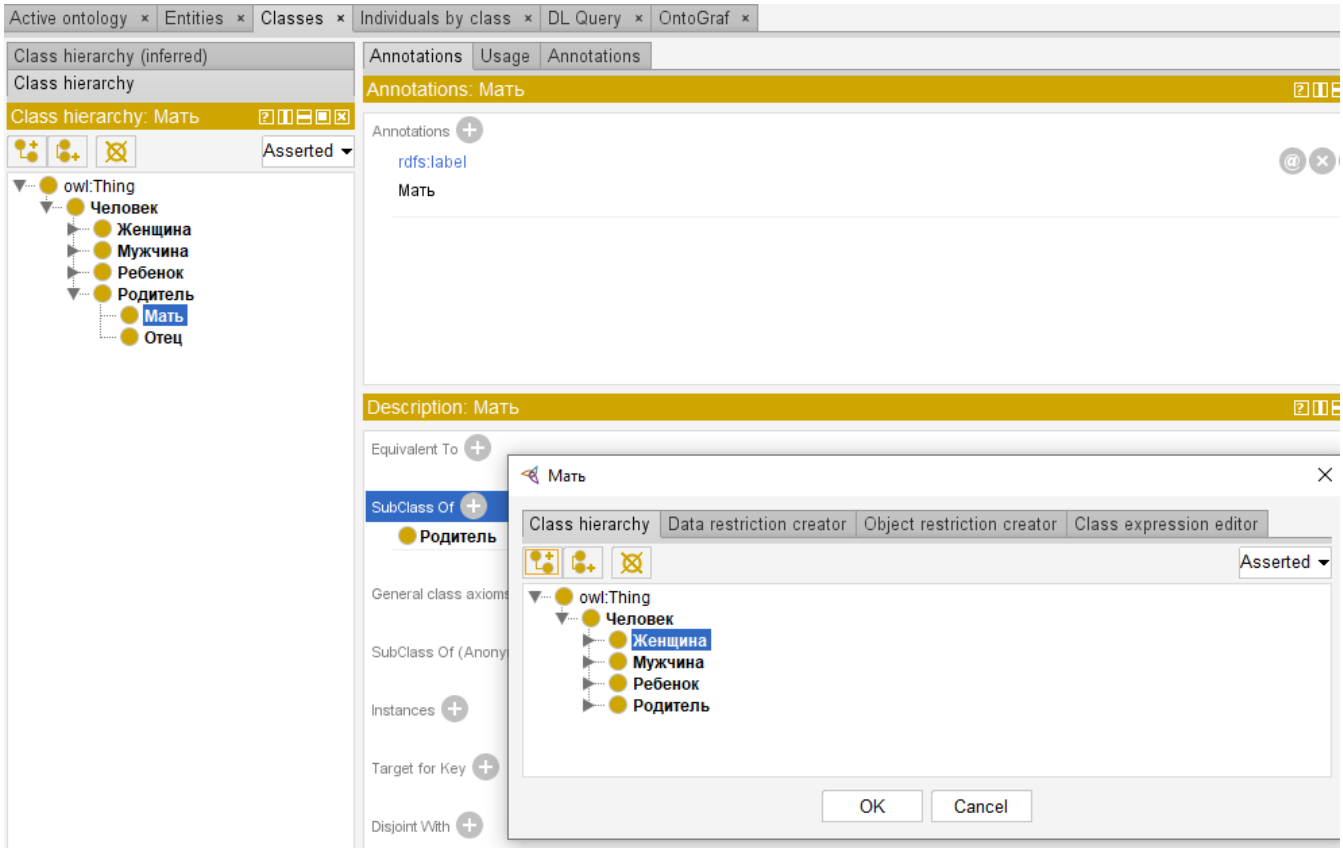


5. Для класса «Человек» создадим 4-е подкласса «Мужчина», «Женщина», «Ребенок» и «Родитель».

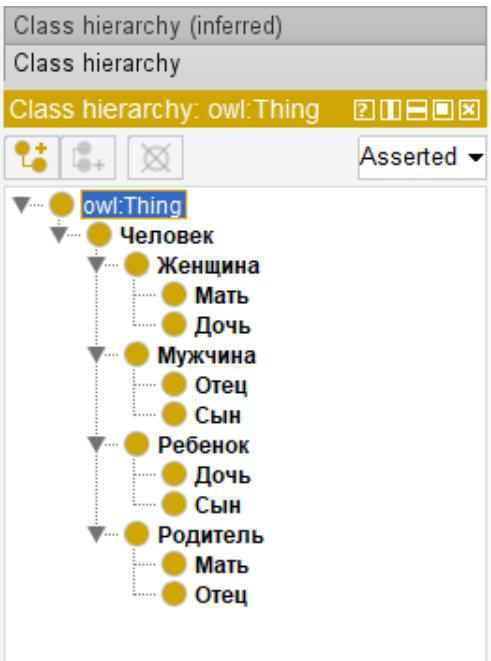


6. Мы знаем, что родители могут быть либо папой, либо мамой, и поэтому в классе родитель создаем еще два дочерних класса «Отец» и «Мать».

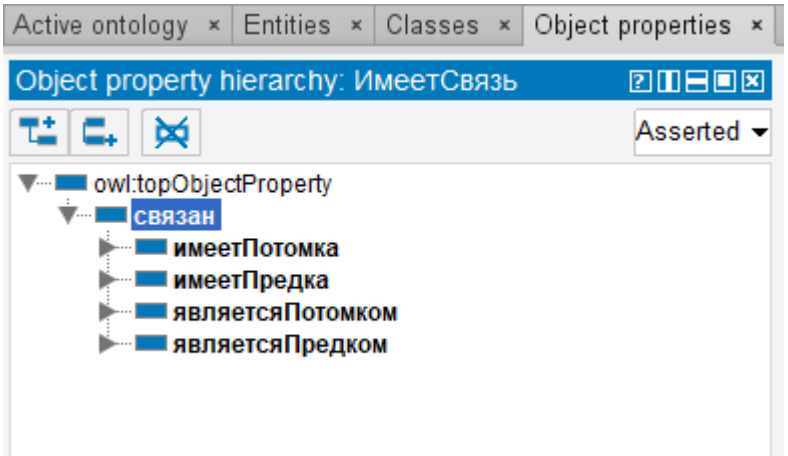
7. Кроме того, данные классы - это не только родители, но еще и одновременно являются женщиной и мужчиной. Соответственно, для того, чтобы сделать эти классы подклассами «Мужчины» и «Женщина», нужно нажать на знак «+» рядом с опцией SubClass Of в панели Description и выбрать в иерархии соответствующий класс и на нажать OK.



8. Также создадим два подкласса «Дочь» и «Сын» для класса «Ребенок». В результате у вас должна получиться следующая иерархия классов:

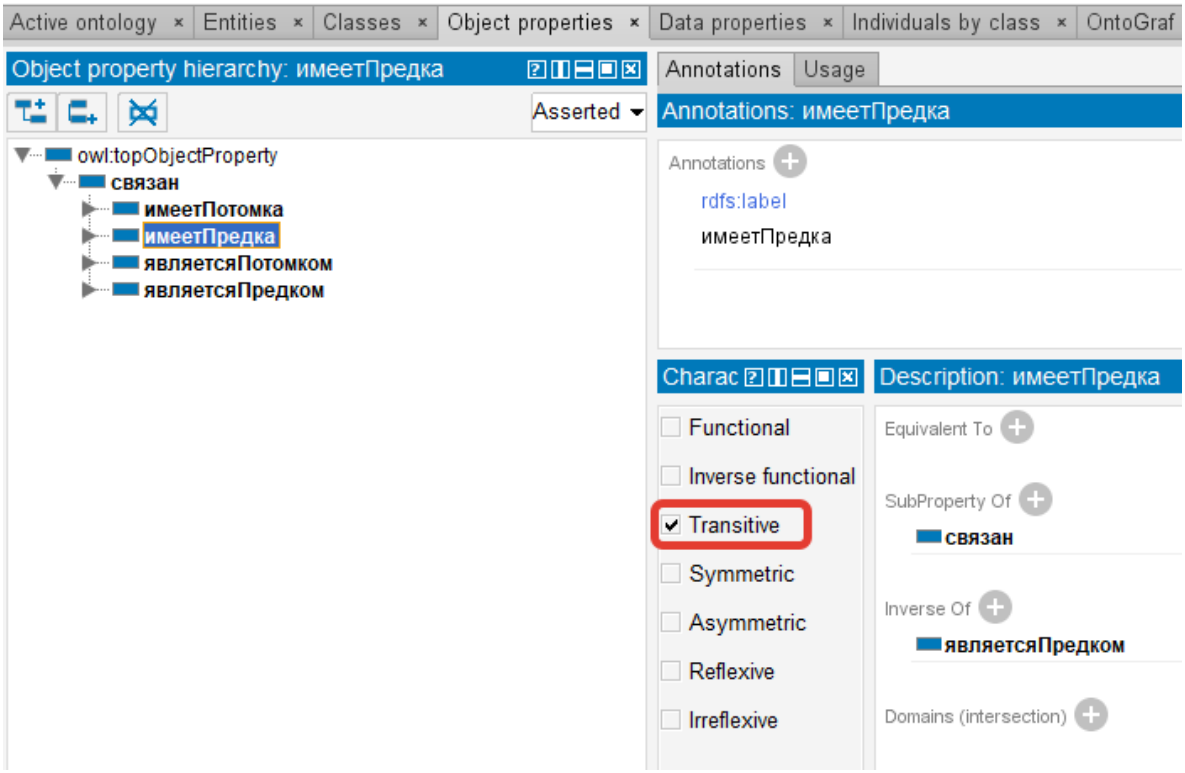


9. Поскольку онтология состоит у нас из трех основных элементов - это концепты, это роли и это индивиды. Концепты или классы мы уже создали. Теперь нужно создать роли или связей между этими концептами, ну или в терминологии данного редактора - это так называемая Object Properties. Для этого нужно перейти в соответствующую вкладку «Object Properties». Роли – это двух местных отношения, то есть они связывают разные концепты. Аналогично с классами роли тоже объединены в иерархии и здесь нужно очень хорошо продумать иерархию ролей, т.к. это менее очевидная вещь, чем иерархия классов. Поэтому давайте создадим корневое свойство, которое будет называться «связан», то есть она будет обозначать любой характер связей между людьми, а дочерние свойства «имеетПотомка», «имеетПредка», «являетсяПотомком», «являетсяПредком»

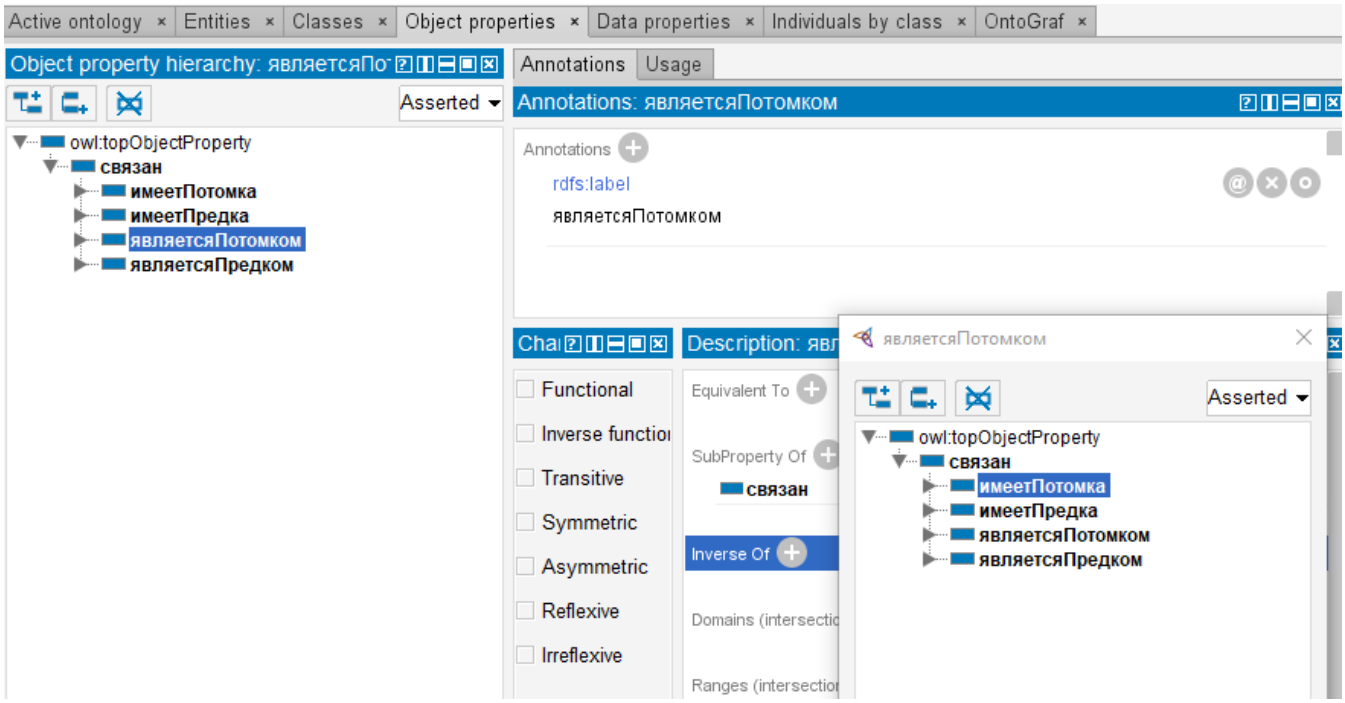


Здесь принято использовать для названий классов имена, начинающиеся с больших букв, а для названий свойств - с маленьких

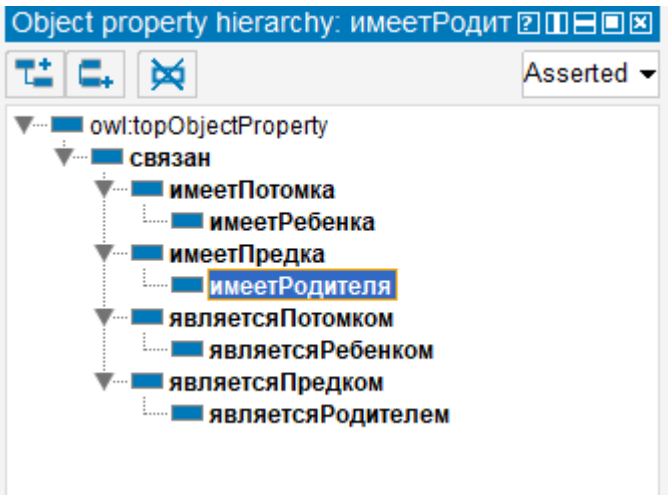
Так как свойства «имеетПотомка», «имеетПредка», «являетсяПотомком», «являетсяПредком» являются транзитивными, т.е. потомок моего потомка - мой потомок, а предок мое предка - мой предок, то для обоих этих свойств нужно установить свойство транзитивности, поставив соответствующую галочку напротив Transitive.



А свойства «имеетПотомка» и «являетсяПотомком» взаимно обратные, то есть это одно и то же свойство, только действующее в противоположную сторону. Для этих целей у свойства можно установить характеристику, которая называется инверсность. Тоже самое сделаем и для «имеетПредка» и «являетсяПредком».

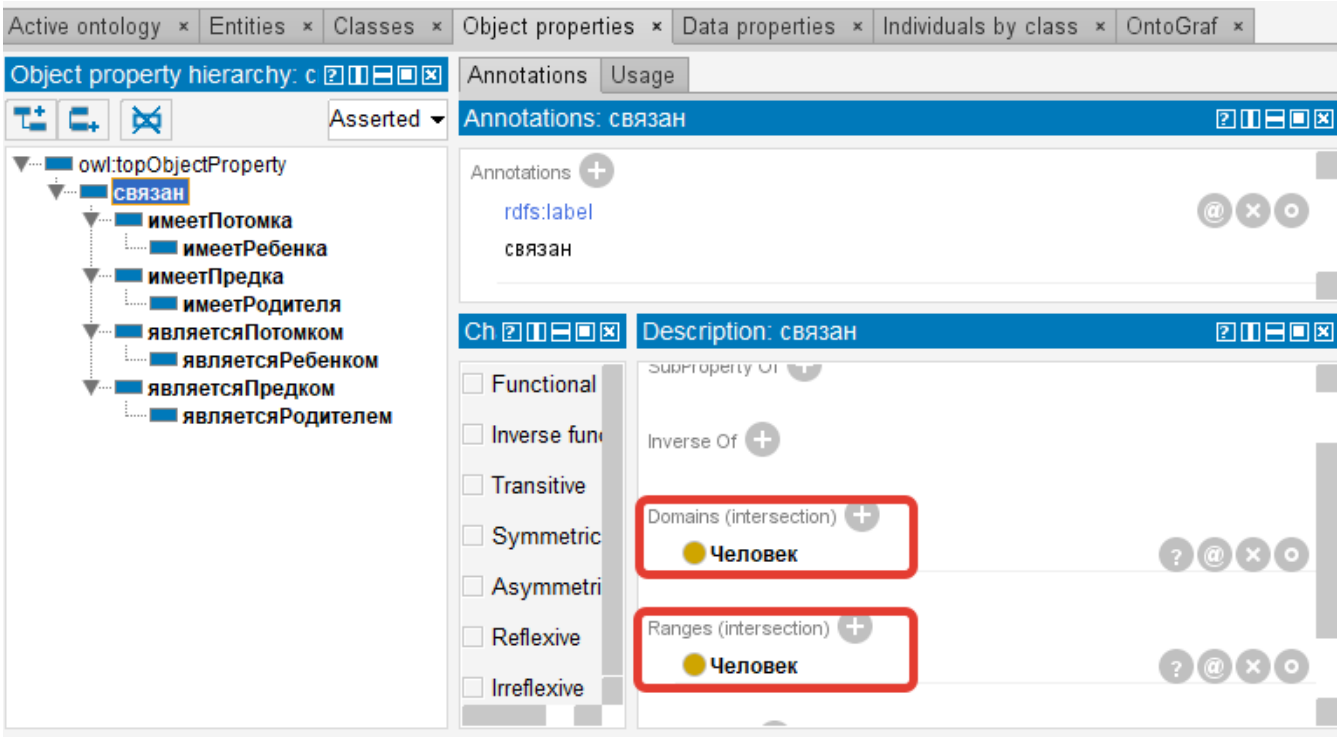


Теперь каждому из свойств нужно добавить соответствующие дочерние свойства.

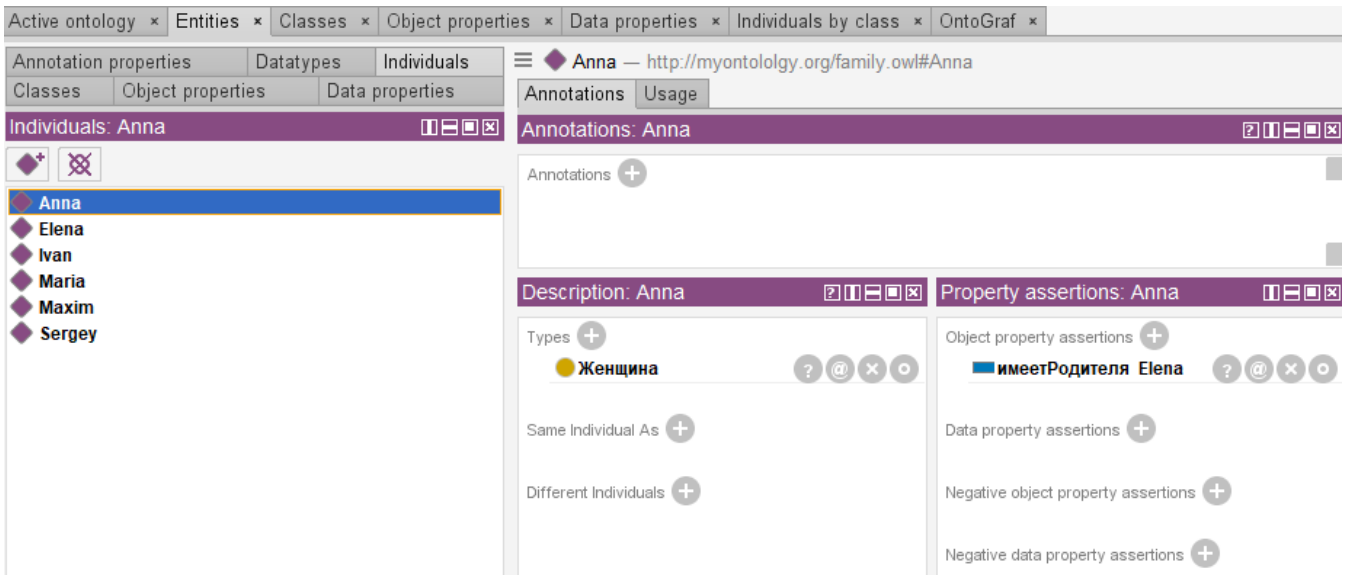


Здесь, также свойства «являетсяРебенком» и «имеетРебенка», а также «являетсяРодителем» и «иметьРодителя» - это инверсные свойства.

10. Для каждого свойства нам необходимо определить области применения этого свойства «Domains» и «Ranges». Свойство «связан» может связывать только двух людей, для этого нужно определить здесь «Domains» и «Ranges» как «Человек»

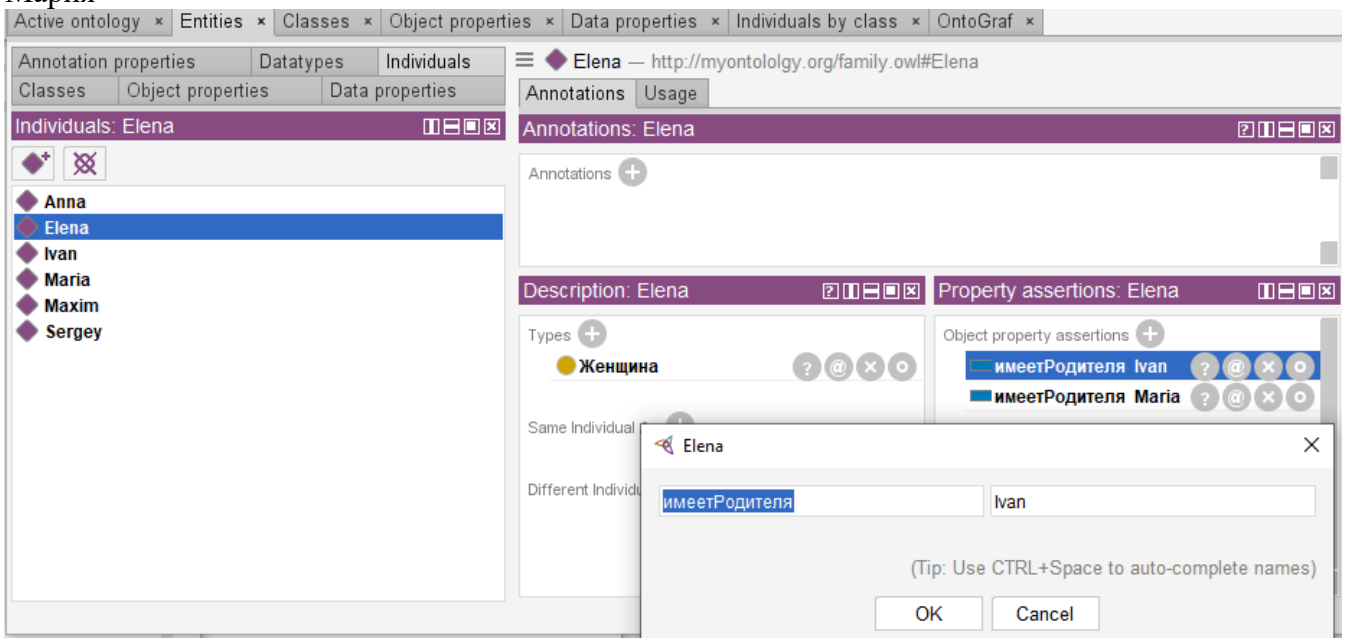


11. Теперь создадим экземпляры этих классов. Чтобы создать экземпляр класса нужно выбрать вкладку Individuals и нажать кнопку Add Individuals. После чего нужно ввести имя человека. Допустим пусть будут имена Иван, Мария, Елена, Максим, Сергей и Анна.



Для каждого имени нужно задать советующий тип, т.е. для мужских имен – это тип «Мужчина», а для женских – «Женщина»

12. Теперь мы должны для созданных индивидов при помощи определенных нами ролей установить какие-то связи. Предположим у Елены у нас будут родители Иван и Мария

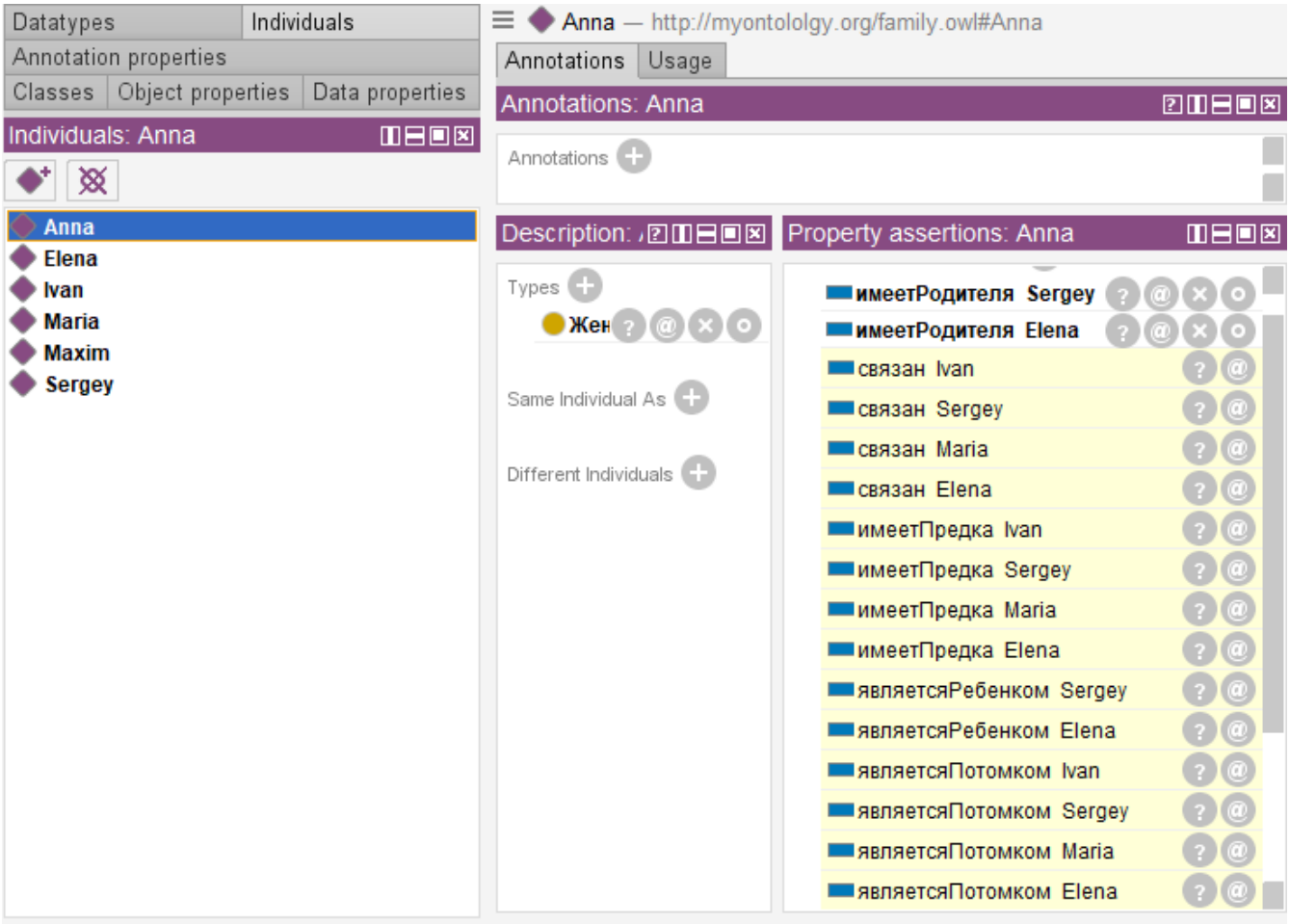


Соответственно Максим и Аня, например, будут иметь в родителях Елену и Сергея

13. Сохраните все изменение, выбрав пункт меню «File / Save».

14. Поскольку онтология создаем не просто для того, чтобы что-то описать и потом на это посмотреть. Самым главным и самым интересным механизмом в онтологических системах является машины логического вывода, то есть это алгоритм, который умеет выполнять операции дискреционной логики и при помощи которого можно выполнять разные логические операции. Машина логического вывода называется ризонер. В редакторе Protégé существуют несколько реализации ризонеров, чтобы определить алгоритм в меню нужно выбрать вкладку Reasoner. Но для того, чтобы он заработал нужно выполнить

команду Start Reasoner, то есть после выполнения этой команды ризонер проанализирует нашу онтологию на предмет целостности и каких-то противоречий, если вдруг в процессе создания индивидов или свойств мы нарушили какие-то правила, то есть в нашей онтологии есть некоторые противоречия, то ризонер даст нам об этом знать. Если онтология не является противоречивой, то он просто запустит ее. Выполните команду Start Reasoner.
В результате у вас должно получиться следующее



Свойства Object properties, которые выделены жёлтым цветом, – это свойства, которые мы не задавали для этого индивида и которые были вычислены ризонером.

Задание 1

А) Построить онтологическую модель в редакторе Protégé, включающую не менее 10 классов и подклассов. Для каждого класса и подкласса определить 3-4 свойства-отношения и 5-6 свойств-данных.

Б) Предметная область онтологии выбирается по одному из перечисленных вариантов (в соответствии с номером ФИО студента в списке группы):

1. автомобили;
2. самолеты;
3. железная дорога;
4. флора;
5. фауна;
6. искусственный интеллект;
7. информационные системы;
8. медицина;
9. строительство;

10. астрономия;
11. физика;
12. туризм;
13. спорт
14. география;
15. вооруженные силы;
16. недвижимость;
17. реклама;
18. игровая индустрия.

Практическая работа 6.
Построение онтологической модели в Protégé. Определение эквивалентных, инверсных, транзитивных ролей (свойств) и создание цепочек ролей (свойств)

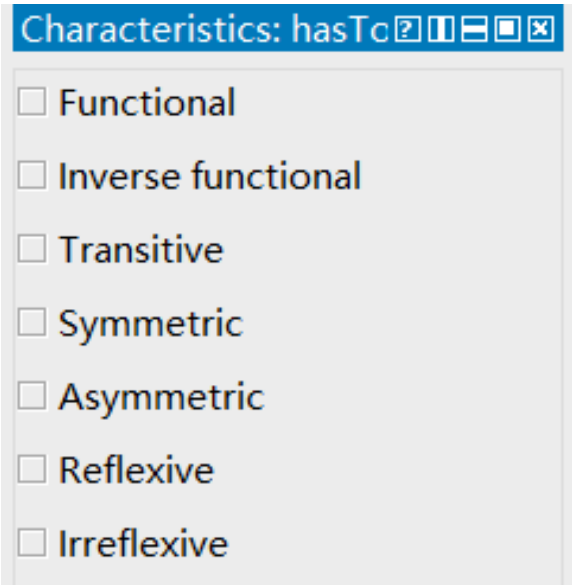
Цель занятия – сформировать представление об использование различных характеристик Object Property в связях онтологической модели в системе Protégé.

Задачи занятия:

- изучить особенности определения эквивалентных, инверсных, транзитивных ролей (свойств) и создание цепочек ролей (свойств) в системе Protégé;
- научиться реализовывать различные роли и создавать цепочки свойств в системе Protégé.

Задания для выполнения и методические рекомендации:

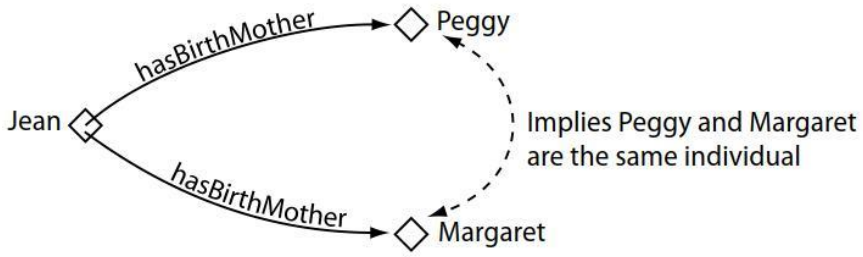
Объекты и субъекты в любая онтологии взаимодействую с помощью связи. Данные связи выстраиваются с помощью логики. В редакторе Protégé логика состоит из характеристик свойств



Характеристики Object Property отображают заявленные характеристики для выбранного свойства объекта. Характеристики отображаются в виде списка флажков. Если флажок установлен для данной характеристики, это означает, что некоторая онтология задействует эту характеристику.

Functional

Здесь Функциональное можно рассматривать как отношение «одного отображения» в функции. Если свойство атрибута объекта является функциональным (однозначным), то для экземпляра существует только один экземпляр, связанный с этим атрибутом объекта.



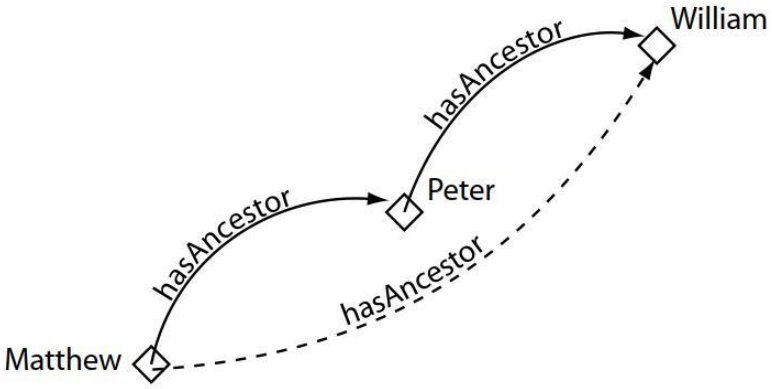
Это пример в «Руководстве», атрибут объекта «hasBirthMother» определяет «функциональные» характеристики, тогда «Jean hasBirthMother Peggy» и «Jean hasBirthMother Margaret» могут содержать два факта. Рассуждение приводит к тому, что «Пегги и Маргарет - одно и то же лицо». Если есть другие факты, указывающие на то, что Пегги и Маргарет не одно и то же лицо, это приведет к противоречиям.

Inverse Functional

Означает, что свойство, обратное выбранному свойству (независимо от того, объявлено оно явно или нет), является функциональным. Другими словами, для этого индивидуума может существовать не более одной входящей связи в свойстве. Обратите внимание, что если в качестве входящих значений для свойства указано несколько лиц, то эти значения будут рассматриваться как обозначающие один и тот же объект.

Transitive

Транзитив определяет транзитивность свойств объекта. На следующем рисунке приведен пример, в котором предполагается, что атрибут объекта «hasAncestor» имеет свойство «Transitive».



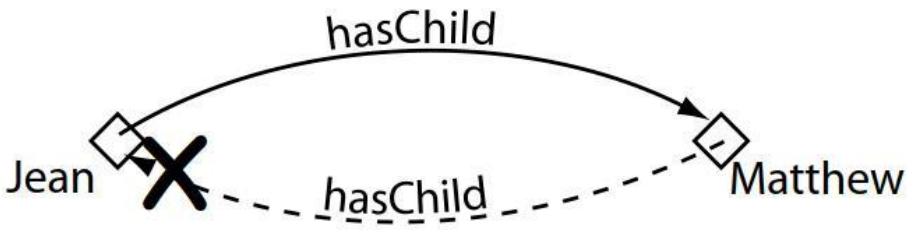
Symmetric

Симметричный определяет симметрию свойств объекта. Среди членов семьи, если у Мэтью есть брат/сестра (hasSibling) по имени Gemma, у Джеммы, естественно, также есть брат/сестра (hasSibling) по имени Matthew. Мы можем определить, что атрибут объекта «hasSibling» является симметричным, тогда нам нужно объявить только один факт, а другой факт возникает естественным образом из-за «симметрии» атрибута объекта.

Asymmetric

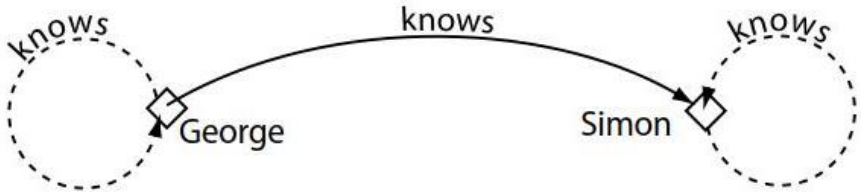
Утверждает, что выбранное свойство является асимметричным. Интуитивно это означает, что если индивид x связан с индивидом y, то индивид y не связан с индивидом x по тому же свойству.

Асимметрию также легче понять, например, атрибут hasChild нельзя определить как симметричный.



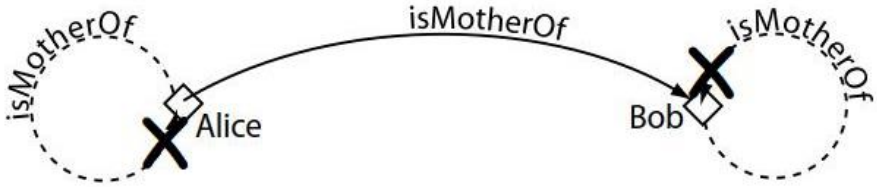
Reflexive

Утверждает, что выбранное свойство является рефлексивным. Утверждение о том, что свойство является рефлексивным, приводит к тому, что каждый отдельный индивид связан с самим собой через это свойство.



Irreflexive

Утверждает, что выбранное свойство является нерефлексивным. Утверждение о том, что свойство является нерефлексивным, означает, что индивид не может быть связан с самим собой через это свойство.



Irreflexive определяет неотражаемость свойств объекта, это означает, что свойство объекта не может описать отношения между самим экземпляром и самим собой. Например, isMotherOf.

Задания для выполнения и методические рекомендации:

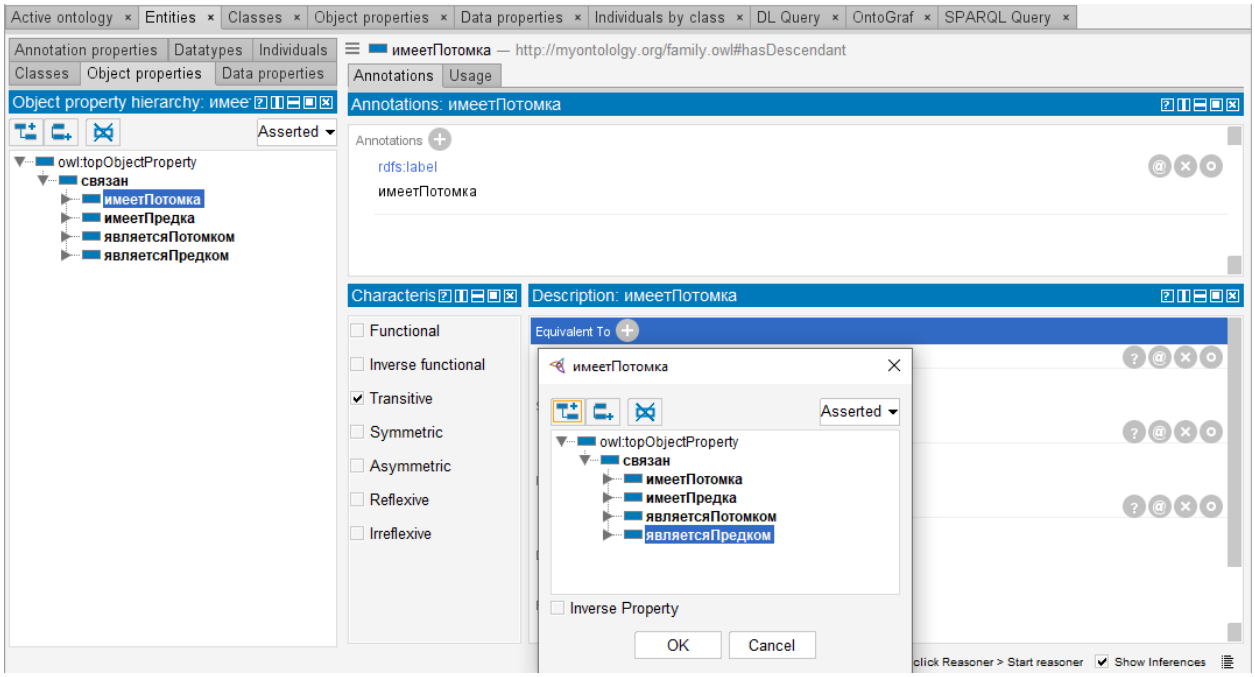
В лабораторной работе 4 была создана онтология родственных отношений, состоящая из связей «предок-потомок». Поэтому вначале давайте проанализируем эти отношениях. Первоначально в антологии определены были четыре вида отношений или ролей, или Object Properties, как они называются в терминах онтологического редактора Protégé. Это:

являетсяПредком
являетсяПотомком
имеетПредка и
имеетПотомка.

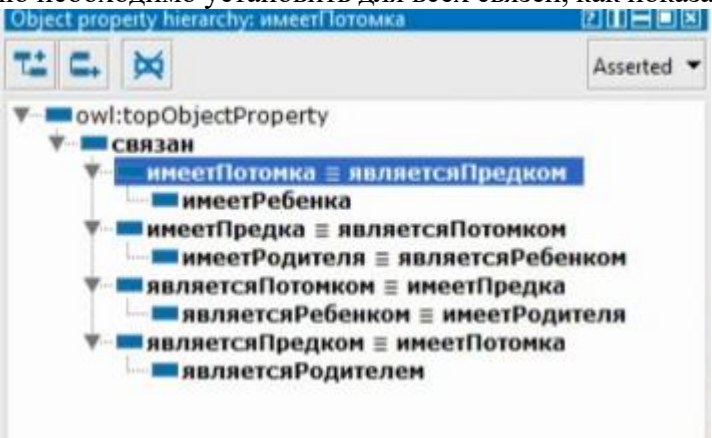
Если проанализировать семантику этих отношений, то мы с вами поймем, что являтьсяПредкам и иметьПотомка - это одно и то же свойство или одна и та же роль

Действительно, если Елена имеет Ивана в качестве предка, то одновременно она является потомком этого же индивида. Поэтому целесообразно установить эквивалентность свойств между предками и потомками.

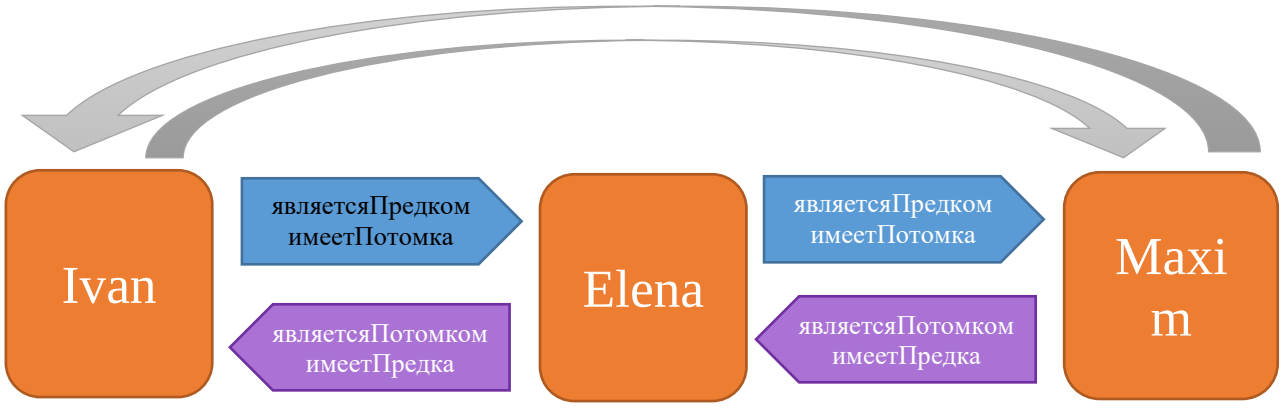
1. Для того, чтобы установить эквивалентность нужно выбрать соответствующее свойство, затем в разделе Description выбрать знак + напротив пункта Equivalent To



Эквиваленцию необходимо установить для всех связей, как показано рисунке.

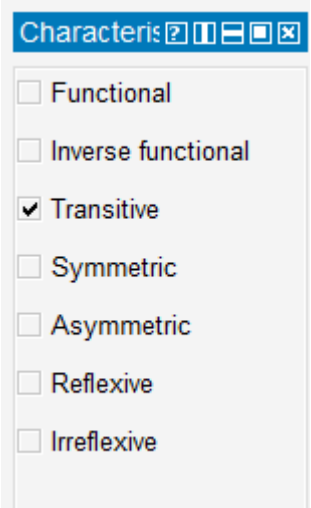


2. Если посмотреть на схему, то можно заметить закономерность: что если Иван Является предком Елены, а Елена является предком Максима, то совершенно очевидно, что Иван в то же самое время является предком Максима. А, если бы у Ивана был еще какой-то предок, то он тоже бы являлся предкам Максима и так далее.



То есть понятие предка является транзитивным то есть предок моего предка является моим предком. Также аналогично с потомками. И таким образом это должно действовать на неограниченное количество уровней вложенности.

Чтобы задать свойство транзитивности, необходимо поставить соответствующую галочку в панели Charachteristics

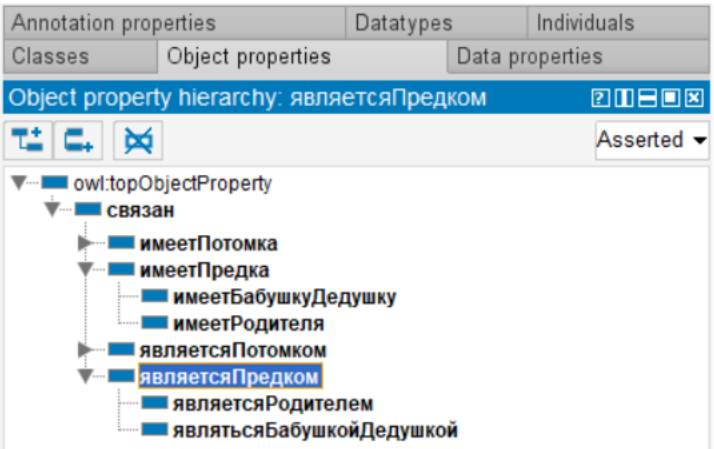


Свойства имеетПотомка и имеетПредка являются транзитными, также как являетсяПредком и являетсяПотомком. Обратите внимание, что для родителей и для детей свойства транзитивности не указываются, потому что ребенок моего ребенка очевидно не мой ребенок, также, как и родитель моего родителя - не мой родитель.

3. С другой стороны, как вы, наверное, уже догадались, есть понятие внуков, бабушек и дедушек, которые тоже неплохо было бы отразить в онтологии и для этого будем использовать так называемые цепочки свойств. Возьмём любые три индивида, если между ними установлены отношения родитель-ребенок, то цепочка из двух таких отношений является эквивалентом отношения бабушки/дедушки и отношения внука.

Давайте посмотрим, как это отразить в онтологическом редакторе. Для этих целей давайте сначала создадим свойства имеетБабушкуДедушку и являетсяБабушкойДедушкой.

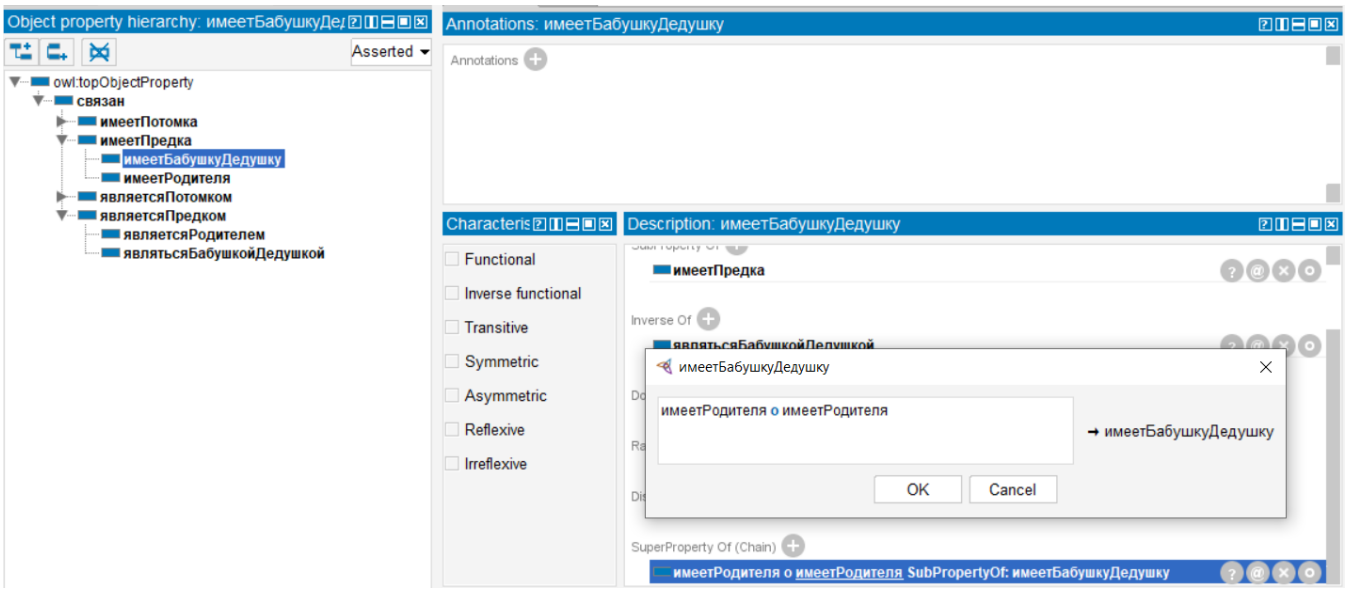
Поскольку бабушка/дедушка тоже являются предками, то логично создавать данное свойство как вложенное свойства имеетПредка и являетсяПредком, поэтому здесь их и создадим.



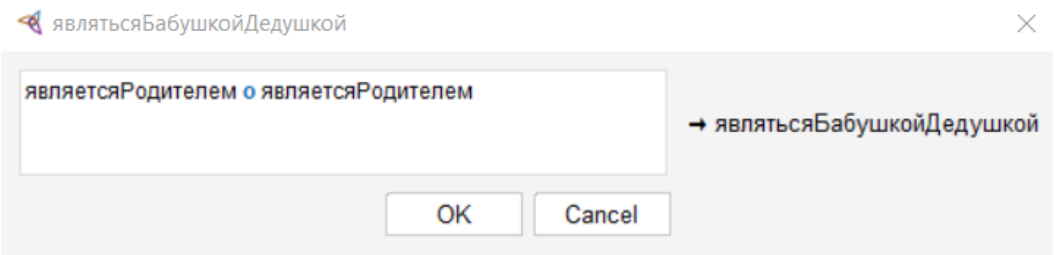
Не будем различать бабушку и дедушку по полу, так как это будет сделано в конечном итоге по тому, к какому классу мужчине или женщине относится тот или иной индивид. То есть для упрощения онтологии, не будем создавать отдельную роль для бабушки и отдельно для дедушки, создадим одну для них вместе.

4. Далее нужно определить цепочку свойств, через которые будет выражена данное свойство. Так как бабушка и дедушка соответственно являются родителями родителей, поэтому иметьБабушкуДедушку - это значит иметь родителя своих родителей.

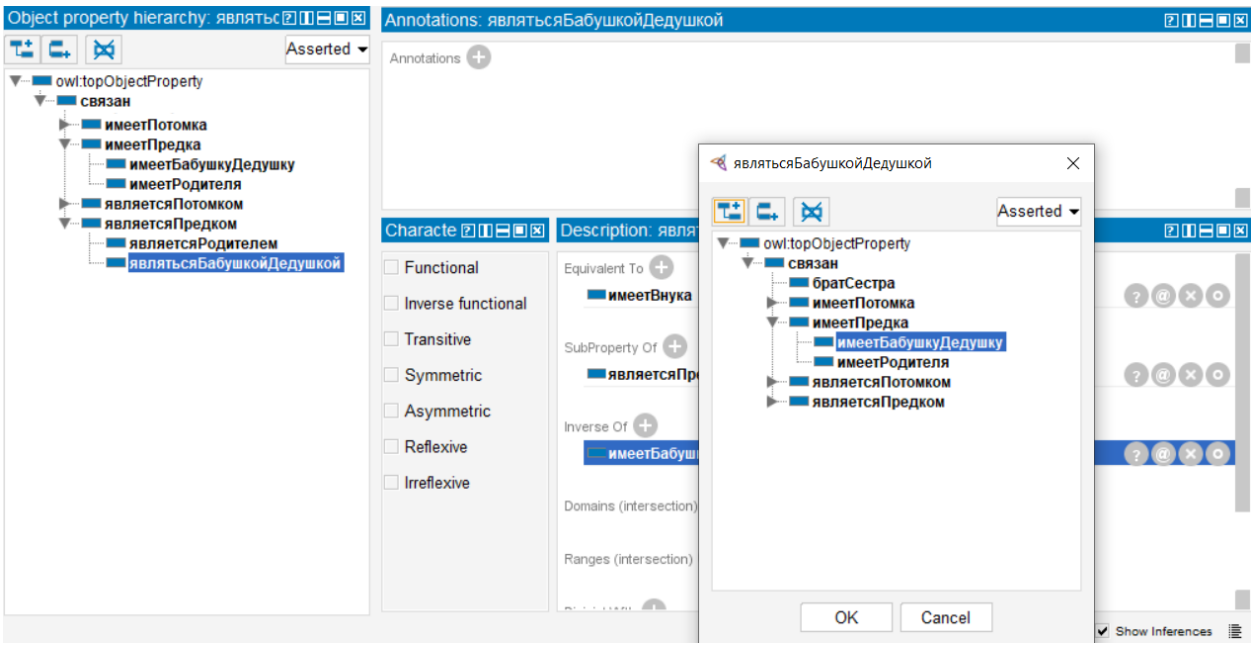
Перейдем в окно Description и здесь внизу есть раздел, который называется SuperProperty Of (Chain), куда и нужно добавить цепочку свойств (вложенная в свойства имеетБабушкуДедушку)



Также добавляем цепочку и для свойства являетсяБабушкойДедушкой

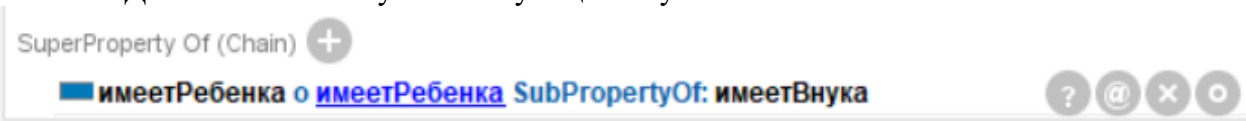


5. Можем логически предположить, что свойства являетсяБабушкойДедушкой будет обратным свойству имеетБабушкуДедушку. Поэтому необходимо добавить сюда инверсию



6. После добавления свойств имеетБабушкуДедушку и явлетсяБабушкойДедушкой, необходимо добавить информацию о внуках. Для этого в свойство имеетПотомка добавляем вложенное свойство имеетВнука, в свойство являетсяПотомком – являетсяВнуком.

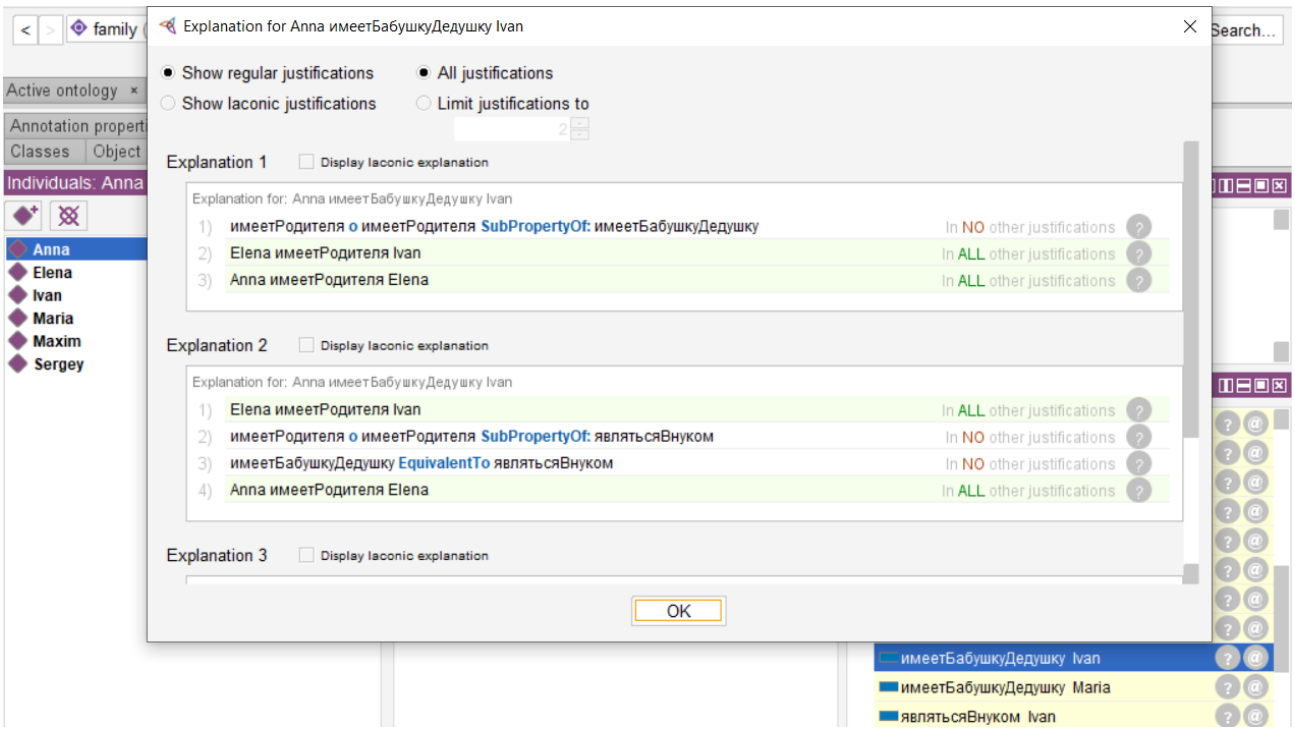
7. Добавляем свойству имеетВнука цепочку



Для свойства являетсяВнуком напишите цепочку самостоятельно

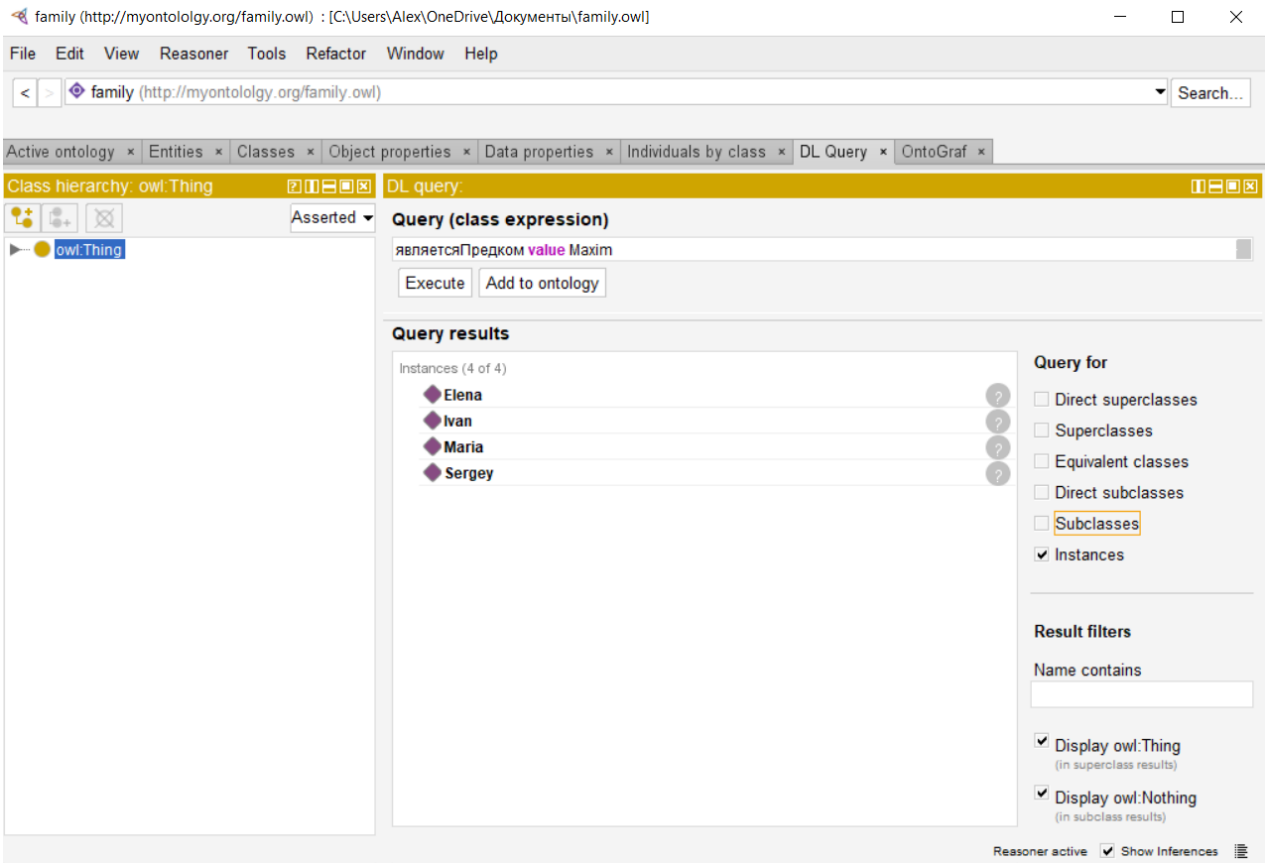
8. Теперь нужно связать понятия внуков и бабушек/дедушек, так как на самом деле сейчас у нас они пока логически не связаны, то есть у нас есть отдельная линия внуков и отдельная линия бабушек/дедушек. Для этого нужно создать перекрестную эквивалентность, иначе говоря нужно показать, что иметьВнука и являтьсяБабушкойДедушкой - это одно и то же, то есть эквивалентное свойство. Поэтому давайте для свойства имеетВнука создадим эквивалентное свойство являтьсяБабушкойДедушкой, а для являтьсяВнуком - имеетБабушкуДедушку

9. Для проверки на ошибки всех изменений, которые вы внесли, необходимо запустить Ризонер. После его запуска появятся все связи, которые вычисляет программа (они выделены желтым). Чтобы просмотреть как программа сделала тот или иной вывод, необходимо нажать на знак вопроса



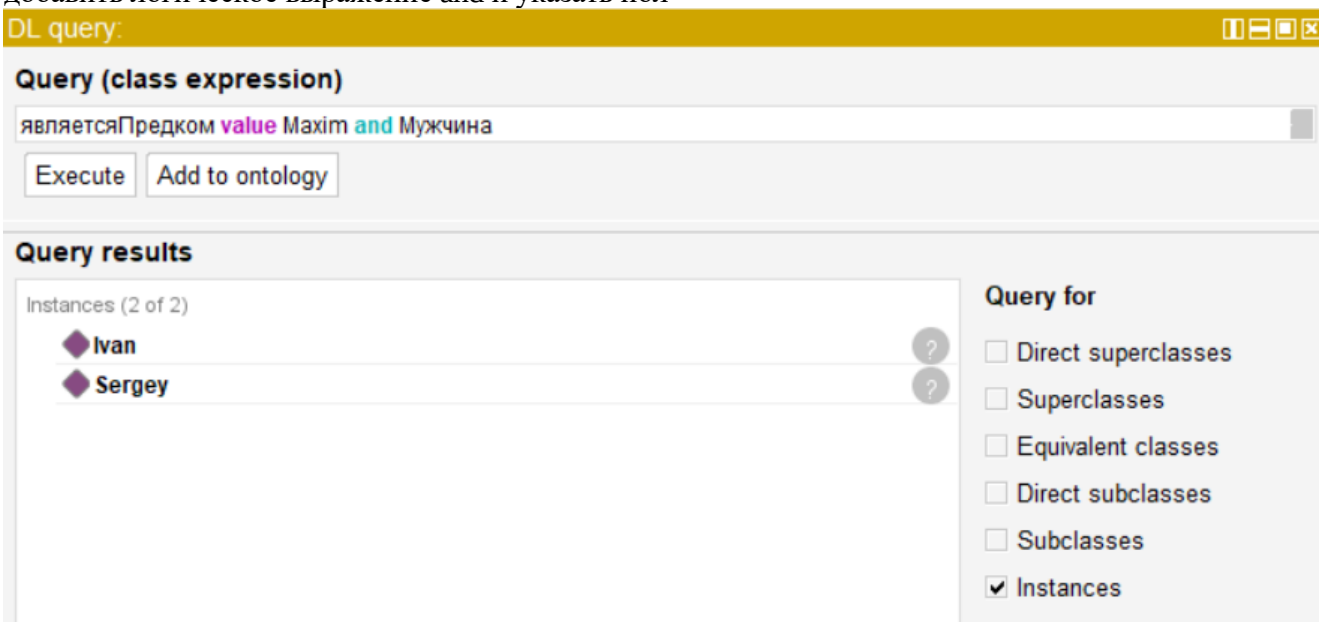
10. Также проверить правильно ли вы сделали связи можно и с помощь DL запросов (DL Query)

Например, нужно узнать, кто является предком Максима. Для этого во вкладке DL Query (если ее нет, то нужно ее отобразить в меню Window > Tabs > DL Query) введите запрос:



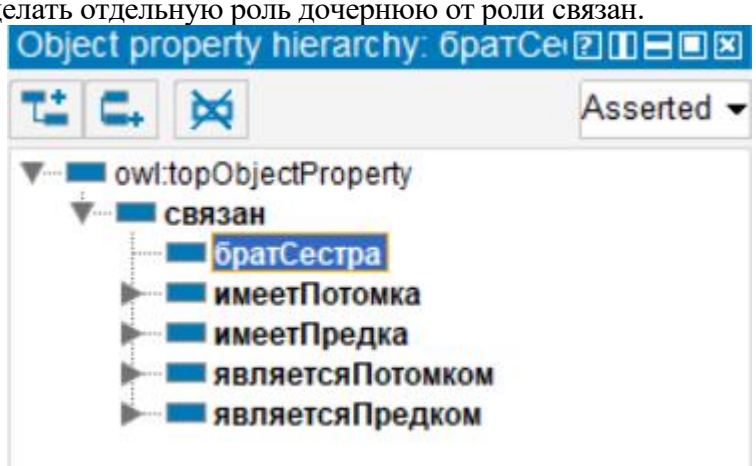
Самостоятельно сделайте запрос, кто является бабушкой и дедушкой Максима

11. Соответственно, чтобы определить только дедушку или папу, надо к запросу добавить логическое выражение and и указать пол

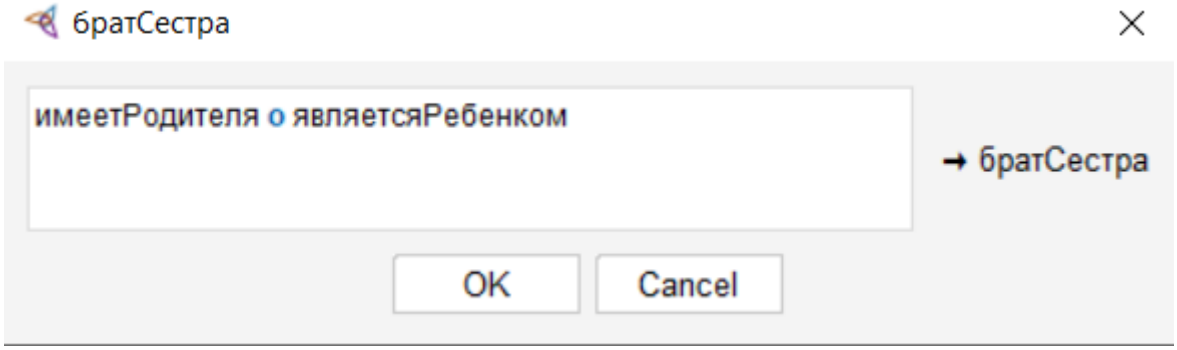


Самостоятельно определите отдельно бабушку и дедушку Максима

12. Если внимательно посмотреть на данную онтологию, то можно заметить, что в ней не хватает отношений между братьями и сестрами, то есть по вертикали все отношения сейчас выстроены, а по горизонтали их пока что нет. Соответственно, нужно добавить данное свойство во вкладке Object properties. Поскольку братья и сестры не является ни предками, ни потомками, поэтому ни в одну из этих ролей не нужно включать братьев и сестер, а нужно сделать отдельную роль дочернюю от роли связан.



Поскольку брат/сестра - это человек, который имеет одного и того же родителя, то нужно установить соответствующую цепочку



Кроме того, данное свойство является транзитивным и симметричным (т.е. двунаправленное отношение). Выставляем эти параметры галочку в панели Charachteristics.

13. Самостоятельно создайте свойство супруги, установите соответствующие связи (не забывайте, что данное свойство является симметричным).

Задание 1

В онтологии, которая была вами разработана в л/р 4, добавьте 2-3 свойства, установите связи. Запустите Ризонер и проверьте созданную онтологию на ошибки. Сделайте несколько DL запросов.

Практическая работа 7.
Разработка учебной экспертной системы

Цель занятия – сформировать представление о разработке и использовании учебной экспертной системы.

Задачи занятия:

- изучить примеры построения экспертных систем;
- научиться создавать и редактировать базу знаний ЭС;
- научиться реализовывать и тестировать базу знаний.

Задания для выполнения и методические рекомендации:

Экспе́ртная систе́ма (ЭС, англ. expert system) — компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации. Современные ЭС начали разрабатываться исследователями искусственного интеллекта в 1970-х годах, а в 1980-х получили коммерческое подкрепление. Предшественниками экспертных систем были предложены в 1832 году С. Н. Корсаковым, создавшим механические устройства, так называемые «интеллектуальные машины», позволявшие находить решения по заданным условиям, например, определять наиболее подходящие лекарства по наблюдаемым у пациента симптомам заболевания.

В информатике экспертные системы рассматриваются совместно с базами знаний (база знаний (БЗ; англ. knowledge base) в информатике и исследованиях искусственного интеллекта — это особого рода база данных, разработанная для оперирования знаниями. База знаний содержит структурированную информацию, покрывающую некоторую область знаний, для использования кибернетическим устройством (или человеком) с конкретной целью. Современные базы знаний работают совместно с системами поиска информации, имеют классификационную структуру (классификация (классифицирование) (от лат. classis — разряд и лат. facere — делать) — особый случай применения логической операции деления объема понятия, представляющий собой некоторую совокупность делений (деление некоторого класса на виды, деление этих видов и т.д.) и формат представления знаний (представление знаний — вопрос, возникающий в информатике — с подбором представления конкретных и обобщённых знаний, сведений и фактов для накопления и обработки информации в ЭВМ. Главная задача в искусственном интеллекте (ИИ) — научиться хранить знания таким образом, чтобы программы могли осмысленно обрабатывать их и достигнуть тем подобия человеческого интеллекта).

Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила вывода, допускающие автоматические умозаключения о вновь вводимых фактах и, как следствие, осмысленную обработку информации. Область наук об искусственном интеллекте, изучающая базы знаний и методы работы со знаниями, называется инженерией знаний) как модели поведения экспертов в определенной области

знаний с использованием процедур логического вывода и принятия решений, а базы знаний — как совокупность фактов и правил логического вывода в выбранной предметной области деятельности. В инструментальную систему входят помимо описанной выше экспертной оболочки программа-редактор баз знаний и программа логического вывода.

Этапы разработки экспертных систем:

1. Идентификация.

Определение участников и их ролей в процессе создания и эксплуатации экспертной системы.

В процессе создания экспертной системы могут участвовать следующие специалисты: инженеры по знаниям, эксперты, программисты, руководитель проекта, заказчики (конечные пользователи). При реализации сравнительно простых экспертных систем программистов может и не быть. Роль инженера по знаниям - выуживание профессиональных знаний из экспертов и проектирование базы знаний экспертной системы и ее архитектуры. Программист необходим при разработке специализированного для данной экспертной системы программного обеспечения, когда подходящего стандартного (например, оболочки для создания экспертной системы) не существует или его возможностей не достаточно и требуются дополнительные модули.

В процессе эксплуатации могут принимать участие конечные пользователи, эксперты, администратор.

На этом этапе разработчик должен ответить на следующие вопросы:

- Какой класс задач должна решать экспертная система?
- Как эти задачи могут быть охарактеризованы или определены?
- Какие можно выделить подзадачи?
- Какие исходные данные должны использоваться для решения?
- Какие понятия и взаимосвязи между ними используются при решении задачи экспертами?
- Какой вид имеет решение, и какие концепции используются в нем?
- Какие аспекты опыта эксперта существенны для решения задачи?
- Какова природа и объем знаний, необходимых для решения задачи?
- Какие препятствия встречаются при решении задач?
- Как эти помехи могут влиять на решение задач?

2. Концептуализация.

На этом этапе разработчик должен ответить на следующие вопросы:

- Какие типы данных нужно использовать?
- Что из данных задано, а что должно быть выведено?
- Имеют ли подзадачи наименования?
- Имеют ли стратегии наименования?
- Имеются ли ясные частичные гипотезы, которые широко используются?

3. Формализация.

4. Реализация прототипной версии.

5. Тестирование.

6. Перепроектирование прототипной версии.

Назначение, возможности и принцип работы байесовской стратегии оценки выводов

Байесовская стратегия оценки выводов - одна из стратегий, применяемых для оценки достоверности выводов (например, заключений продукционных правил) в экспертных системах. Основная идея байесовской стратегии заключается в оценке вероятности некоторого вывода с учетом фактов, подтверждающих или опровергающих этот вывод.

Вероятность осуществления некой гипотезы H при наличии определенных подтверждающих свидетельств E вычисляется на основе априорной вероятности этой

гипотезы без подтверждающих свидетельств и вероятностей осуществления свидетельств при условиях, что гипотеза верна или неверна.

Значения P(E/H) и P(E/неH), подставленные в теорему Байеса, позволяют вычислить апостериорную вероятность исхода, т.е. вероятность, скорректированную в соответствии с ответом пользователя на данный вопрос:

P(H/E) = P(E/H) * P(H)/(P(E/H) * P(H) + P(E/неH) * P(неH))

или

P апостериорная = Py * P/(Py * P + Pn * (1 – P))

Пример применения байесовской стратегии оценки выводов

В экспертных системах для составления прогнозов погоды вероятность дождя на следующий день определяется с учетом трех факторов: ветер, влажность, облачность (в день наблюдения). За 173 дня (из них 53 дождливых) накоплены статистические данные, приведенные в таблице.

Приведенные в таблице данные означают, например, следующее: за период наблюдений (173 дня) слабый ветер наблюдался 71 день (71=19+52). В 19 случаях на следующий день погода была дождливой, в 52 случаях - без осадков.

В некоторый день наблюдается следующая погода: сильный ветер, высокая влажность, облачно. Требуется найти вероятность дождливой погоды на следующий день.

Погода в день наблюдения		Количество случаев дождливой погоды на следующий день	Количество случаев погоды без осадков на следующий день
Ветер	Слабый	19	52
	Умеренный	27	44
	Сильный	7	24
Влажность	Высокая	35	18
	Средняя	12	42
	Низкая	6	60
Облачность	Ясно	5	83
	Облачно	8	27
	Пасмурно	40	10

В данном случае в качестве гипотез рассматриваются состояния погоды на следующий день: H1 - дождь, H2 - погода без осадков. Свидетельством здесь является сочетание трех факторов, характеризующих погоду в день наблюдения: ветер, влажность и облачность (можно сказать, что в данном случае используются три свидетельства); обозначим эти факторы как E1,E2, E3. Обозначим наблюдаемое сочетание факторов (сильный ветер, высокая влажность, облачно) как событие E.

Определим вероятности, необходимые для расчетов по формуле Байеса. Априорные вероятности гипотез (то есть вероятности дождя и погоды без осадков, без учета наблюдаемого состояния погоды):

P(H1)=53/173=0,306;

P(H2)=120/173=0,694.

Наблюдаемое свидетельство (состояние погоды в день наблюдения) представляет собой сочетание трех событий, наблюдаемых вместе: сильного ветра, высокой влажности и облачности. Считая эти события независимыми (то есть считая, например, что

влажность не зависит от облачности, и т.д.), можно найти условные вероятности свидетельства по формуле умножения вероятностей:

P(E/Hi) = P(E1, E2, E3/Hi) = P(E1/Hi) P(E2/Hi)P(E3/Hi), i=1, 2.

Найдем величины, необходимые для применения формулы умножения вероятностей:

P(E1/H1)=7/53=0,132; P(E2/H1)=35/53=0,66; P(E3/H1)=8/53=0,151; P(E1/H2)=24/120=0,2; P(E2/H2)=18/120=0,15; P(E3/H2)=27/120=0,225.

Здесь, например, P(E1/H1)- вероятность того, что в текущий день наблюдается сильный ветер, при условии, что следующий день будет дождливым. Эта величина показывает, насколько часто наблюдается сильный ветер в дни, предшествующие дождливой погоде.

Подставляя найденные величины в формулу умножения вероятностей, получим:

P(E/H1) = 0,132*0,66*0,151 = 0,0132; P(E/H2) = 0,2*0,15*0,225 = 0,00675.

Здесь, например, величина P(E/H1) - вероятность наблюдаемого состояния погоды (сильного ветра, высокой влажности и облачности) при условии, что следующий день будет дождливым.

При построении экспертных систем часто используют вероятностный подход. Составляя базу знаний, применяют экспертную матрицу, имеющую следующий вид:

Наблюдения	Значение «да»	Значение «нет»
1	a	b
2	c	d
…	…	…
10	s	t

Номера 1, 2, …, 10 соответствуют вопросам заданной предметной области, буквы a, b, …, t в массиве означают возможные варианты ответов. Матрица должна содержать все знания эксперта по данному предмету.

Область запросов - та область, в которой экспертная система должна быть экспертом.

Возможные варианты ответов в экспертной матрице носят вероятностный характер. Они задаются числами от 0 до 1. Если событие наверняка должно произойти, то вероятность его появления равна 1. Если событие не имеет шанса произойти когда-либо, то вероятность его появления равна 0. Если же событие равновероятно, то есть может как произойти, так и нет, то вероятность его появления равна 0,5. Все остальные возможные случаи лежат в указанном диапазоне.

Пример:

Завтра будет дождь Завтра дождя не будет

Сыро 0,6 0,4

Сухо 0,45 0,55

Задания для выполнения и методические рекомендации:

Задание 1

Создать простую базу знаний «Ты геймер?», решающей задачу классификации (с четкой логикой).

В этой БЗ будет 2 равновероятных исхода: «геймер» и «не геймер».

Априорные вероятности обоих исходов равны 0.5 (P(H) = 0.5), т.е. гипотезы, что наугад выбранный человек является или не является геймером, равновероятны (при этом сумма априорных вероятностей равна 1, т.е. в базе знаний приведены все возможные исходы).

Каждое утверждение четко относит испытуемого к одной из этих категорий, т.е. если для первого вопроса для исхода «геймер» будет запись 1,1,0, то для исхода «не геймер» запись будет 1,0,1.Таким образом, Py и Pn для каждого вопроса равны либо 1, либо 0, при этом во втором правиле эти значения инвертируются (относительно первого правила). Это приводит к тому, что максимально уверенный ответ («Точно да» или «Точно нет») на любой вопрос однозначно классифицирует пользователя как геймера или не геймера (см. таблицу 1).

(Примечание: не следует указывать Py и Pn равными друг другу, т.к. это означает, что данное свидетельство не влияет на вероятность исхода, т.е. бессмысленно его упоминать).

Представим правила, используемые ЭС «Ты геймер?», в виде таблицы.

Таблица 1

№ п/п	**Утверждение / Вопрос**	**Заключение / Исход**	
		геймер	не геймер
1.	Ты играешь в компьютерные игры по 6 и более часов в сутки	1	0
2.	Ты бросаешь все дела ради новой игры	1	0
3.	От 3D-action игр у тебя кружится голова	0	1
4.	Ты ни за что не потащишь свой компьютер к другу для организации соревнований по сети	0	1
5.	Ты в основном тратишь деньги или на новый игровой диск или на апгрейд компьютера	1	0
6.	Ты не пользуешься "мышью" в 3D-action играх	0	1

В формате базы знаний данные таблицы будут записаны в виде вопросов и исходов, приведенных ниже.

Вопросы:

Ты играешь в компьютерные игры по 6 и более часов в сутки?

Ты бросаешь все дела ради новой игры?

От 3D-action игр у тебя кружится голова?

Ты ни за что не потащишь свой компьютер к другу для организации соревнований по сети?

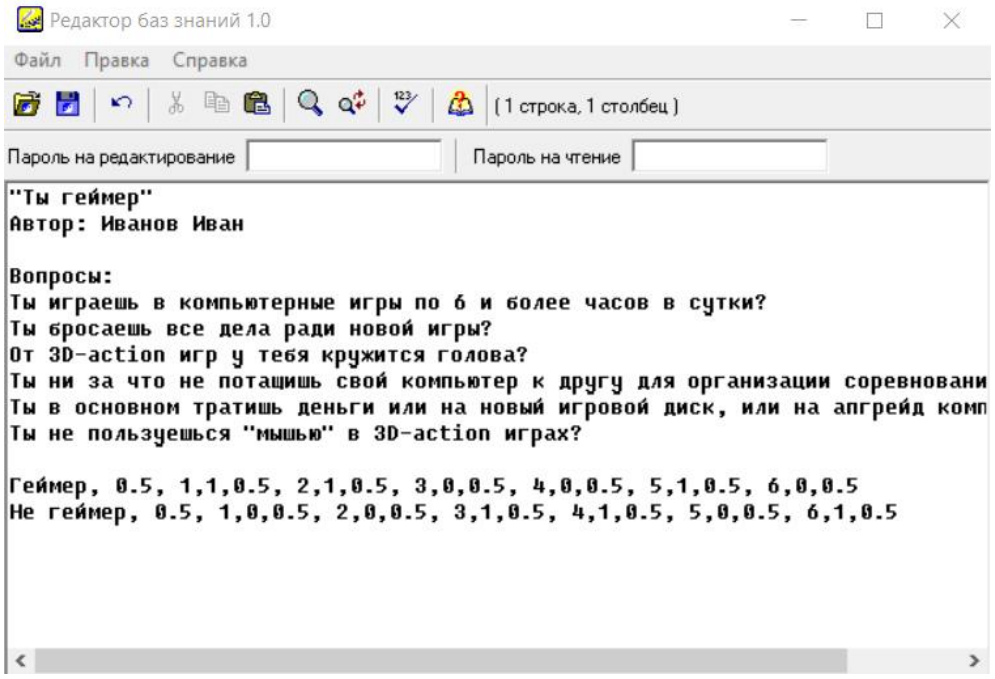
Ты в основном тратишь деньги или на новый игровой диск или на апгрейд компьютера?

Ты не пользуешься "мышью" в 3D-action играх?

Геймер, 0.5, 1,1,0, 2,1,0, 3,0,1, 4,0,1, 5,1,0, 6,0,1

Не геймер, 0.5, 1,0,1, 2,0,1, 3,1,0, 4,1,0, 5,0,1, 6,1,0

Если реализовывать данную БЗ в программе «Малая Экспертная Система», то она будет выглядеть следующим образом:



Задание 2

Загрузите в программу «Малая Экспертная Система» и изучите примеры из папки «Базы знаний». На основание рассмотренных примеров попробуйте составить собственную БЗ, связанную с вашей профессиональной деятельностью.

Рекомендуемая литература:

Перечень основной литературы:

1. Пальмов, С.В.
    Интеллектуальные системы и технологии Электронный ресурс : учебное пособие / С.В. Пальмов. - Самара : Поволжский государственный университет телекоммуникаций и информатики, 2022. - 195 с. - Книга находится в базовой версии ЭБС IPRbooks.

2. Сысоев, Д. В. Введение в теорию искусственного интеллекта : учебное пособие / Д. В. Сысоев, О. В. Курипта, Д. К. Проскурин. — Москва : Ай Пи Ар Медиа, 2021. — 170 с. — ISBN 978-5-4497-1092-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/108282.html

Перечень дополнительной литературы:

1. Аверченков, В. И. Система формирования знаний в среде Интернет : Монография / Аверченков В. И. - Брянск : Брянский государственный технический университет, 2022. - 181 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-89838-328-X

2. Павлов, С. И. Системы искусственного интеллекта : учебное пособие : [16+] / С. И. Павлов. – Томск : Томский государственный университет систем управления и радиоэлектроники, 2021. – Часть 1. – 175 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=208933

3. Сергеев, Н. Е. Системы искусственного интеллекта : учебное пособие : [16+] / Н. Е. Сергеев. – Таганрог : Южный федеральный университет, 2018. – Часть 1. – 123 с. : схем., ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=493307

4. Сотник, С. Л. Проектирование систем искусственного интеллекта : учебное пособие / С. Л. Сотник. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 228 с. — ISBN 978-5-4497-0868-7. —

Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/102054.html

Интернет-ресурсы:

1. Справочная правовая система «Консультант плюс» https://www.consultant.ru/
2. ЭБС «Университетская библиотека онлайн». http://biblioclub.ru
3. Электронно-библиотечная система IPRsmart . https://www.iprbookshop.ru/
4. Российская государственная библиотека https://www.rsl.ru/
5. Электронно-библиотечная система «Знаниум»: https://znanium.com/
6. Электронно-библиотечная система «Лань» https://e.lanbook.com/
7. Электронно-библиотечная система «Консультант студента» https://www.studentlibrary.ru/
8. Электронно-библиотечная система «Национальный цифровой ресурс «Руконт» https://lib.rucont.ru/
9. Библиотека профессионала №1 https://profspo.ru/
10. Русский язык как иностранный РКИ https://www.ros-edu.ru/

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания по организации и проведению самостоятельной работы по дисциплине «Искусственный интеллект в профессиональной сфере» для студентов направления 40.03.01 Юриспруденция профиль Правозащитная деятельность

Ставрополь, 2026 г.

СОДЕРЖАНИЕ

1. **Общие положения**
2. **Цель и задачи самостоятельной работы**
3. **Технологическая карта самостоятельной работы**
4. **Порядок выполнения самостоятельной работы**

4.1. Методические рекомендации по работе с учебной литературой

4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

4.3. Методические рекомендации по самопроверке знаний

4.4. Методические рекомендации по написанию научных текстов (докладов, рефератов, эссе, научных статей и т.д.)

4.5. Методические рекомендации по выполнению исследовательских проектов

4.6. Методические рекомендации по подготовке к экзаменам и зачетам

5. **Контроль самостоятельной работы**
6. **Список литературы для выполнения СРС**

1. Общие положения

Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента.

К основным видам самостоятельной работы студентов относятся:

– формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);

– написание докладов;

– подготовка к семинарам, практическим и лабораторным работам, их оформление;

– составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);

– выполнение учебно-исследовательских работ, проектная деятельность;

– подготовка практических разработок и рекомендаций по решению проблемной ситуации;

– выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;

– компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;

– выполнение курсовых работ (проектов) в рамках дисциплин;

– выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

2. Цель и задачи самостоятельной работы

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию

самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Порядок выполнения самостоятельной работы

3.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим

собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют ***четыре основные установки в чтении научного текста***:

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;
2. Выделите главное, составьте план;
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

Вопросы для собеседования по самостоятельно изученной литературе

1. Краткая история искусственного интеллекта (ИИ), машины и интеллект. Основные направления исследований в области ИИ.
2. Интеллектуальные системы общения: системы обработки текстов естественного языка, системы обучения с базами данных, диалоговые системы решения задач, системы речевого общения.
3. Разработка естественно-языковых интерфейсов и машинный перевод с одного языка на другой.
4. Интеллектуальные роботы.
5. Обучение и самообучение, искусственные нейронные сети.
6. Задачи распознавания образов.
7. Интеллектуальные игры и машинное творчество.
8. Представление задач и стратегии поиска их решения в пространстве состояний (поиск в глубину и ширину, слепой и эвристический поиск, поиск на игровых деревьях, минимаксный алгоритм, альфа-бета алгоритм и др.).
9. Представление знаний - центральная проблема ИИ. Процедурная и декларативная информация. Переход от обработки данных к оперированию со знаниями.
10. Отличительные особенности знаний от данных: внутренняя интерпретируемость, структурированность, связность, активность. Базы знаний. Нечеткие и неточные знания.
11. Открытость знаний в системах ИИ. Основные методы приобретения знаний. Инженерия знаний.
12. Основные модели представления знаний.
13. Формальные логические модели представления знаний. Проблема понимания естественного языка.
14. Сетевые модели представления знаний (семантические сети). Отображение множества информационных единиц во множество типов связей между ними. Отношения типа «абстрактное-конкретное» и «целое-часть». Иерархия наследования.
15. Фреймовые модели представления знаний. Понятия фрейма, терминального слота, протофрейма. Имя фрейма и имя слота, значение слота и тип данных слота. Фреймы-экземпляры. Механизм наследования.
16. Продукционная модель представления знаний. Системы продукций. База знаний (база фактов и правил).
17. Рабочая память. Механизм вывода: прямая и обратная цепочка рассуждений.
18. Достоинства и недостатки продукционной модели представления знаний.
19. Интегрированные модели представления знаний. Языки представления знаний.
20. Общая характеристика программных средств для разработки и реализации систем ИИ. Требования к программному обеспечению систем ИИ.
21. Инструментальные средства для создания систем ИИ.
22. Понятие экспертной системы (ЭС). Назначение, принципы построения и области применения ЭС. Организация знаний в ЭС.
23. Виды ЭС и типы решаемых ими задач. Инструментальные средства разработки ЭС. Оболочковые средства создания прототипов ЭС.
24. Гибридные логические и моделирующие ЭС, ЭС на базе нечеткой логики. Интеллектуальные информационные ЭС.
25. Структурная схема, основные компоненты, архитектура и режимы использования ЭС продукционного типа.
26. Основные этапы разработки ЭС. Жизненный цикл ЭС.
27. Языки программирования для решения задач ИИ.

3.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

4.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

3.4. Методические рекомендации по написанию научных текстов (докладов, рефератов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т, п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Реферат (доклад) - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Реферат не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в реферате должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки реферата студентом.

Выполнение реферата начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания реферата. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста реферата предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки реферат сдается на кафедру для его оценивания руководителем.

Требования к написанию реферата

Написание 1 реферата является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема реферата может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Реферат должен быть написан научным языком.

Объем реферата должен составлять 20-25 стр.

Структура реферата:

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.
- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.
- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса
- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.
- Приложение (при необходимости).

Требования к оформлению:

- текст с одной стороны листа;
- шрифт TimesNewRoman;
- кегль шрифта 14;
- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- реферат должен быть представлен в сброшюрованном виде.

Порядок защиты реферата:

Защита реферата проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту реферата отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите реферата приветствуется использование мультимедиа-презентации.

Оценка реферата

Реферат оценивается по следующим критериям:

• соблюдение требований к его оформлению;

• необходимость и достаточность для раскрытия темы приведенной в тексте реферата информации;

• умение студента свободно излагать основные идеи, отраженные в реферате;

• способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

Темы докладов

1. Моделирование биологических систем.
2. Автоматический компьютерный синтез речи по тексту. Методы синтеза речи.
3. Примеры систем обработки естественного языка.
4. Классификация инструментальных средств ЭС и организация знаний в ЭС.
5. Модели представления знаний: логическая, сетевая, фреймовая, продукционная.
6. Методы озвучивания речи.
7. История возникновения и современные направления исследований в области ИИ.
8. Классификация систем распознавания речи.
9. Распознавание символов. Шаблонные системы. Структурные системы. Признаковые системы.
10. Речевой вывод информации.
11. Машинный интеллект и робототехника.
12. Типы задач решаемые в ЭС.
13. Основные направления исследований в области искусственного интеллекта.
14. Предпосылки возникновения систем понимания естественного языка. Понимание в диалоге.
15. Распознавание рукописных текстов.

3.5. Методические рекомендации по выполнению исследовательских проектов

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая Лабораторная работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Требования по выполнению и оформлению проекта

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

Критерии оценки работы участника проекта.

Для каждого из участников проекта оцениваются:

• профессиональные теоретические знания в соответствующей области;

• умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;

• умение работать с техническими средствами;

• умение пользоваться соответствующими выполняемому проекту информационными технологиями;

• умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;

• умение работать в команде;

• умение публично представлять результаты собственной деятельности;

• коммуникабельность, инициативность, творческие способности.

Критерии выставления оценки участникам проекта

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Отлично»	Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен, допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом.	Технические средства и информационные технологии освоены и использованы для реализации проекта полностью	Студент проявил инициативу, творческий подход, способность к выполнению сложных заданий, навыки работы в коллективе, организационные способности.	Проект представлен полностью и в срок.
«Хорошо»	Работа выполнена на достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.	Обнаруживаются некоторые ошибки в использовании соответствующих технических средств и информационных технологий	Студент достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.	Проект представлен достаточно полно и в срок, но с некоторыми недоработками.
«Удовлетво	Уровень	Обнаруживает	Студент	Проект сдан

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
рительно»	недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.	недостаточное владение навыками работы с техническими средствами и соответствующим и информационным и технологиями	выполнил большую часть возложенной на него работы.	со значительным опозданием (более недели) и не полностью
«Неудовлетворительно»	Работа не выполнена или выполнена на низком уровне. Допущено более 8 фактических ошибок. Ответы на связанные с проектом вопросы обнаруживают непонимание предмета и отсутствие ориентации в материале проекта.	Навыков работы с техническими средствами нет, информационные технологии не освоены	Студент практически не работал, не выполнил свои задачи или выполнил лишь отдельные не существенные поручения в групповом проекте.	Проект не сдан.

Студенты должны: защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (*если проект групповой*).

Примерные темы индивидуальных творческих проектов

1. Проект «Электронный словарь»;
2. Проект «Экспертная система для диагностики здоровья»;
3. Проект «Лингвистическая экспертная система»;
4. Проект «Экспертная система для диагностики устройства»;
5. Проект «Обучающая экспертная система».

3.6. Методические рекомендации по подготовке к экзаменам и зачетам

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент

демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий, особенно по математике - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неутомительные занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо во время ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Правила подготовки к зачетам и экзаменам:

• Лучше сразу сориентироваться во всем материале и обязательно расположить весь материал согласно экзаменационным вопросам (или вопросам, обсуждаемым на семинарах), эта работа может занять много времени, но все остальное – это уже технические детали (главное – это ориентировка в материале!).

• Сама подготовка связана не только с «запоминанием». Подготовка также предполагает и переосмысление материала, и даже рассмотрение альтернативных идей.

• Готовить «шпаргалки» полезно, но пользоваться ими рискованно. Главный смысл подготовки «шпаргалок» – это систематизация и оптимизация знаний по данному предмету, что само по себе прекрасно – это очень сложная и важная для студента работа, более сложная и важная, чем простое поглощение массы учебной информации. Если студент самостоятельно подготовил такие «шпаргалки», то, скорее всего, он и экзамены сдавать будет более уверенно, так как у него уже сформирована общая ориентировка в сложном материале.

• Как это ни парадоксально, но использование «шпаргалок» часто позволяет отвечающему студенту лучше демонстрировать свои познания (точнее – ориентировку в знаниях, что намного важнее знания «запомненного» и «тут же забытого» после сдачи экзамена).

• Сначала студент должен продемонстрировать, что он «усвоил» все, что требуется по программе обучения (или по программе данного преподавателя), и лишь после этого он вправе высказать иные, желательно аргументированные точки зрения.

4. Контроль самостоятельной работы

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка реферата, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

5. Список литературы для выполнения СРС

Рекомендуемая литература:

Перечень основной литературы:

1. Пальмов, С.В.
    Интеллектуальные системы и технологии Электронный ресурс : учебное пособие / С.В. Пальмов. - Самара : Поволжский государственный университет телекоммуникаций и информатики, 2022. - 195 с. - Книга находится в базовой версии ЭБС IPRbooks.

2. Сысоев, Д. В. Введение в теорию искусственного интеллекта : учебное пособие / Д. В. Сысоев, О. В. Курипта, Д. К. Проскурин. — Москва : Ай Пи Ар Медиа, 2021. — 170 с. — ISBN 978-5-4497-1092-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/108282.html

Перечень дополнительной литературы:

1. Аверченков, В. И. Система формирования знаний в среде Интернет : Монография / Аверченков В. И. - Брянск : Брянский государственный технический университет, 2022. - 181 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-89838-328-X

2. Павлов, С. И. Системы искусственного интеллекта : учебное пособие : [16+] / С. И. Павлов. – Томск : Томский государственный университет систем управления и радиоэлектроники, 2021. – Часть 1. – 175 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=208933

3. Сергеев, Н. Е. Системы искусственного интеллекта : учебное пособие : [16+] / Н. Е. Сергеев. – Таганрог : Южный федеральный университет, 2018. – Часть 1. – 123 с. : схем., ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=493307

4. Сотник, С. Л. Проектирование систем искусственного интеллекта : учебное пособие / С. Л. Сотник. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 228 с. — ISBN 978-5-4497-0868-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/102054.html

Интернет-ресурсы:

1.Справочная правовая система «Консультант плюс» https://www.consultant.ru/
2.ЭБС «Университетская библиотека онлайн». http://biblioclub.ru
3.Электронно-библиотечная система IPRsmart . https://www.iprbookshop.ru/
4.Российская государственная библиотека https://www.rsl.ru/
5.Электронно-библиотечная система «Знаниум»: https://znanium.com/
6.Электронно-библиотечная система «Лань» https://e.lanbook.com/
7.Электронно-библиотечная система «Консультант студента» https://www.studentlibrary.ru/
8.Электронно-библиотечная система «Национальный цифровой ресурс «Руконт» https://lib.rucont.ru/
9.Библиотека профессионала №1 https://profspo.ru/
10.Русский язык как иностранный РКИ https://www.ros-edu.ru/